Deep Learning with Synthetic, Temporal, and Adversarial Supervision

D.Phil Thesis

Visual Geometry Group
Department of Engineering Science
University of Oxford

Ankush Gupta
Balliol College

Supervised by
Professor Andrew Zisserman
Professor Andrea Vedaldi

Michaelmas Term, 2018

Ankush Gupta
Balliol College

Doctor of Philosophy
Michaelmas Term, 2018

Deep Learning with Synthetic, Temporal, and Adversarial Supervision

Abstract

In this thesis we explore alternatives to manually annotated training examples for supervising the training of deep learning models. Specifically, we develop methods for learning under three different supervision paradigms, namely — (1) *synthetic* data, (2) *temporal* data, and (3) *adversarial* supervision for learning from *unaligned* examples. The dominant application domain of our work is *text spotting*, *i.e.* detection and recognition of text instances in images. We learn text localisation networks on synthetic data, and harness an adversarial discriminator for training text recognition networks using no paired training examples. Further, we exploit the changing pose of objects in temporal sequences (videos) to learn object landmark detectors. The unifying objective is to scale deep learning methods beyond manually annotated training data.

We develop a large-scale, *realistic* synthetic scene text dataset. Armed with this large annotated dataset of scene images, we train a novel, fast fully-convolutional text detection network, and show excellent performance on *real* images. This generalisation from synthetic to real images, confirms the verisimilitude of our rendering process. The dataset, *SynthText in the Wild*, has been widely adapted by the research community, and has enabled the development of end-to-end text spotting models.

While synthetic text can be readily generated, it needs to be adapted for the specific application domain. However, *unaligned* examples of text-images, and valid language sentences are abundant. With this in mind, we develop a method for text recognition which learns from such unaligned data. We cast the text recognition problem as one of aligning the conditional distribution of strings predicted from given text images, with lexically valid strings. This alignment is induced through an *adversarial* discriminator which tries to distinguish the predicted and real text strings apart. Our method achieves excellent text recognition accuracy, using no labelled training examples.

Temporal sequences (videos) of objects encode changes in their pose. We develop a method to harness this, and learn object landmark detectors, which *consistently* track object parts across different poses and instances. We achieve this by conditionally generating a future frame given a past frame, and a sparse keypoint like (learnt) representation extracted from the future frame. We demonstrate generality of our method by learning landmarks for human faces (where we outperform existing landmark detectors), articulated human body, and rigid 3D objects, with no modification to the method.

Finally, we propose one-step *inductive* training for improving generalisation in *recurrent neural networks* to longer sequences. We restrict the recurrent state to a spatial memory map which tracks the regions of the image which have been accounted for, and train the network for valid evolution of this map. We show excellent generalisation to much longer sequences on two sequential visual recognition tasks — joint localisation and recognition of multiple lines of text, and counting objects in aerial images.

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfillment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Ankush Gupta
Balliol College
September, 2018

Acknowledgments

I am indebted and grateful to my supervisors and *gurus* (in the truest sense of the word), Andrew Zisserman and Andrea Vedaldi, for their excellent guidance and mentorship over the years. Their enthusiasm and encouragement has been pivotal in making this possible. I am extremely fortunate to have worked with them so closely. I hope to never lose the nuggets of wisdom I have gathered during my time here, for they are most valuable.

I thank my thesis examiners Ingmar Posner and Simon Osindero for kindly reading through and evaluating my work. Their valuable feedback and insightful comments have helped improve the manuscript profoundly.

Max Jaderberg generously shared his research output, which provided direction for some of the initial work on text spotting. Work on learning object landmarks grew out of discussions with Tom Jakab. Olivia Wiles meticulously examined every comma in the introduction. Karel Lenc provided the LaTeX template in which this thesis document is typeset. John Schulman has been a constant source of inspiration. I thank them all.

I am also grateful to the fine scholars of the *Visual Geometry Group (VGG)*, past and present (who are too many to name), for enriching my experience. I gratefully acknowledge the generous financial support of the Clarendon Fund, Eddie Dinshaw Scholarship, and the EPSRC AIMS CDT program. Thanks to Balliol College for providing a quiet and tranquil accommodation.

I shall always be thankful to my parents Nishi Gupta and Dinesh Gupta, and my brother Ajay Gupta, for their unconditional love, support, and understanding. Ajay has been a solid pillar of support during my time here.

Above mentioned are some of the people and entities who have had an immediate influence on this work, however I owe an incalculable debt to the wider research community and thinkers, engineers and researchers who have developed the tools and ideas which are our hammers and chisels, the founders, members and visionaries of the excellent institutions which are our sanctuaries, and each member of the society and the wider consciousness who sustains and nurtures our being with their daily toil and labour.

Contents

1

Introduction

Deep learning [LeCun et al., 2015], learns *representations* that map raw data formats to easily ingestible and compact vectors. This capability has been demonstrated across domains such as images [Krizhevsky et al., 2012], audio [Dahl et al., 2012], and text [Sutskever et al., 2014]. Expressive, multi-step function approximators, or *deep neural networks*, learn increasingly abstract, hierarchical representations by simply minimising a suitable loss function on input-output examples through gradient descent. Such representations replace the pre deep learning era features designed to incorporate suitable invariances and other prior domain knowledge. In the visual domain, *Convolutional Neural Networks* (ConvNets) [LeCun et al., 1998] reign supreme. ConvNets exploit the spatial arrangement of pixels in images and apply small *local* filters *convolutionally* over the input array. Deep convolutional representations have become the workhorse for computer vision methods, yielding ground-breaking results in classic visual perception tasks such as object recognition [Krizhevsky et al., 2012], detection [Girshick, 2015a], and segmentation [Long et al., 2015b]. However, in the absence of explicit feature engineering, deep learning methods depend on a large number (often millions [Krizhevsky et al., 2012]) of manually annotated training examples, to distil and discover patterns. Are there alternatives for manual supervision?

In this thesis, we explore training deep ConvNets under three different supervision paradigms, namely (1) *synthetic* data, (2) *temporal* data, and (3) *adversarial* supervision for learning from *unaligned* examples. The unifying objective is to reduce the reliance on manually-annotated training data. This makes training deep neural models more scalable, cost-effective, and efficient. Further, it opens up the possibility of harnessing the ever growing collection of digital data, laying the ground for future methods free from this *annotation bottleneck*.

The dominant application domain in this thesis is *text spotting*, or detection and recognition of instances of text in images. The goal is to advance the state-of-the-art of text-spotting methods, improving their overall performance, while reducing their dependence on manual annotations. Specifically, we develop methods for harnessing synthetically generated data (chap. 3), and easily harvested *unaligned* data (chap. 4) for training text detection and recognition methods respectively. We further explore improving the generalisation ability of joint text localisation and recognition architectures (chap. 6), thereby utilising the available supervision more effectively. Finally, we exploit temporal data, specifically, a pair of video frames, for learning consistent object landmarks (chap. 5).

1.1 Motivation

1.1.1 Scaling supervision for deep learning

Banko and Brill [2001] analysed the effect of training set size on the performance of a number of machine learning algorithms on a natural language processing task. They found that the performance improves with increasing amounts of training data, regardless of the specific learning algorithm as shown in fig. 1.1. Deep learning methods are no different, and in fact are better suited for such scaling, as the model capacity can be readily increased by adding more layers/increasing the number of parameters, making the models amenable for absorbing more data. This, then, essentially ties performance to the availability of large annotated datasets. The existing methods for computer vision are

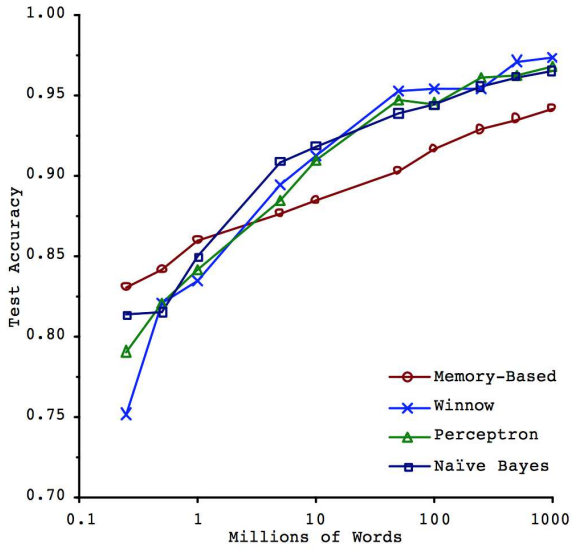


Figure 1.1: Scaling machine learning methods. Performance of various machine learning methods against the training set size, on the *confusion set disambiguation* task (in natural language processing); reproduced from Banko and Brill [2001]. Performance scales with training data, regardless of the specific learning algorithm.

routinely trained on millions of manually-labelled examples [Lin et al., 2014, Russakovsky et al., 2015]. Such manual annotations can be very tedious/expensive to obtain, especially for dense pixel-level tasks. For instance, high-quality semantic pixel labeling required 60 minutes per image for the CamVid dataset [Brostow et al., 2009], and 90 minutes per image for the Cityscapes dataset [Cordts et al., 2016]. Hence, we explore alternative sources of supervision for training deep models, namely synthetic data, temporal sequences, and adversarial supervision for learning from unaligned data.

Synthetic data. Data generated automatically without manual intervention, or *synthetic data*, provides an economical and scalable alternative to manual supervision. With advances in real-time and photo-realistic rendering it has become feasible to render physically-plausible images quickly, which can be used either directly for training deep ConvNets, or after fine-tuning to close the domain gap [Shrivastava et al., 2017]. Synthetic data has multiple advantages: (1) it provides fine control over the content, (2) it is a readily scalable, almost infinite source of annotated data, and perhaps most importantly, (3) it provides access to detailed ground-truth information, *e.g.* dense optical flow, depth maps,

instance segmentation masks, *etc*.. This makes synthetic data ideal for generating large annotated datasets for training deep networks. Further, they can provide annotations that are impossible/expensive to collect manually, such as the style of fonts and the detailed geometric transformations of the various text instances in an image. Finally, synthetic data provides a perfect setting to perform systematic analysis of methods by turning-off various factors and measuring the corresponding impact.

Temporal sequences. Temporal sequences of images or videos encapsulate rich priors and knowledge about the physical world. The temporal order of events encodes cues for cause and effect, and dynamics of objects. Wiskott and Sejnowski [2002] in *Slow Feature Analysis* provide a framework for de-constructing the physical world through identifying the slowly varying underlying causes of high frequency temporal streams. Further, changes in the visual stream due to movement, or *ego-motion*, convey information about the geometric structure of the environment [Kruppa, 1913], and agency of other agents [Sommerville and Woodward, 2005, Sommerville et al., 2005]. Importantly, videos are abundant — for instance, in the year 2017, approximately 300 hours of videos were uploaded every minute to YouTube alone! [1] This makes videos a rich resource for learning visual representations, as also evident from the recent proliferation of large-scale video datasets, *e.g. YouTube-8M* [Abu-El-Haija et al., 2016], *Kinetics* [Kay et al., 2017], and *AVSpeech* [Shillingford et al., 2018]. In our work, we exploit the motion of objects in videos, to learn consistent landmarks for both rigid, and non-rigid objects, without any annotations.

Adversarial supervision for unaligned data. Supervised learning relies on *aligned* pairs of input-output training examples, which are usually obtained through strenuous manual effort. However, *unaligned* data is abundant, and can be mined automatically. For example, while it is difficult to collect text transcriptions for speech tracks, valid text sentences can be mined independently from corpora, and unlabelled speech can be harvested from digital media. The recent work of Zhu et al. [2017] proposes a framework for mapping

[1] https://www.youtube.com/yt/about/press/

elements in one domain to another, given only *unpaired* examples from the two domains. They achieve this by making the predictions indistinguishable from samples from the target domain, using an *adversarial* discriminator [Goodfellow et al., 2014a], paired with a reconstruction objective. While they apply their method to change the *style* of images, this is a general framework, and has been extended to other visual tasks [Tung et al., 2017], and language translation [Zhang et al., 2017]. We extend and adapt this framework for *text recognition*, and learn to recognise text in images, given only valid language strings from the target language. First of all, this relieves the dependence on large aligned datasets for training text recognition methods, as is the current standard practice. Second, this framework presents an opportunity to decode, old historical printed documents and manuscripts, for which labelled data is difficult to obtain.

1.1.2 Text spotting

Text is a concise visual medium for precise communication of ideas. In fact, the advent of *written* historical record, literally splits the time into pre-history and history; text is the only reliable source of exact historical events. Today, text is omnipresent, and is the basis for much of the transfer and communication of knowledge and thoughts. It labels objects, gives precise instructions, and is hence, abundantly represented in the visual media. Recognising text can help disambiguate object identities and extract detailed descriptions from images and videos. Fast automated systems for decoding text from images can be used to index documents, make old manuscripts interpretable, and videos searchable. Further, recognising text signs can help in navigating urban environments, both indoor, and outdoor, which is especially helpful for self-driving cars [Posner et al., 2010].

Detecting text in natural images is very challenging, as text can be easily confused with surrounding texture or clutter. Further, text is different from standard physical objects, as it varies highly in appearance due to variations in font styles, glyph sizes, orientation, colours, and borders which are all determined independently for each instance. Another,

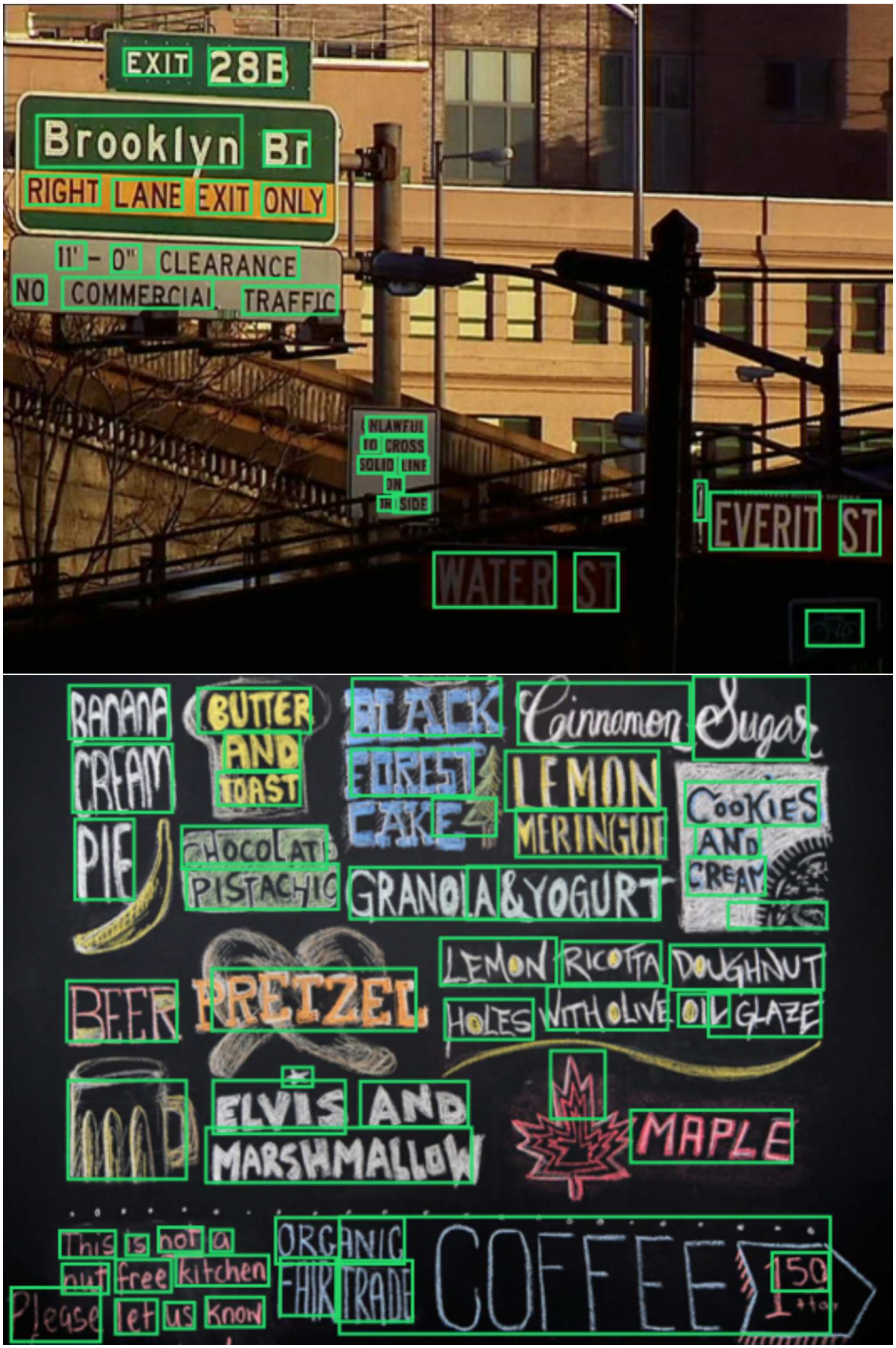


Figure 1.2: Text localisation in the wild. Visualisations of text detections from our detector on two images. Note, in the second image, the method generalises to hand-written text, even when it is trained only on synthetically rendered fonts. View detailed results for these images at our online demo: top, and bottom.

confounding factor is scale. Text is a synthetic construct, so it does not have a standard scale range and can vary drastically in size — from measuring a few pixels in the image, to covering the entire frame. The surrounding context, unlike for physical objects, only provides very weak cues for the scale of text instances. Further, text is quite sensitive to missed/imprecise detections — object detectors for physical objects can miss a few pixels at the boundary, and still be understood to have localised the object successfully; however, missing a few pixels for text could mean that some characters are dropped, which adversely affects the downstream recognition stage. Text *recognition*, in addition to solving the perceptual problem of correctly transducing the pixels to characters, also requires appropriate models to enable generalisation to *unseen* text instances (*e.g.* out of vocabulary words). This is a significant departure from the common *object recognition* setting, where the object classes are pre-determined. Hence, when not restricted to a pre-determined lexicon of words, text recognition becomes an *open-set* problem [Scheirer et al., 2013]. In this thesis we address text localisation (chap. 3), unsupervised text recognition (chap. 4), and joint localisation and recognition of lines of text in images containing blocks of text (multiple-lines/paragraph) (chap. 6). Figure 1.2 visualises two examples images, sourced in the *wild* from the Internet, which demonstrate the above challenges; also visualised is the localisation output of our method.

1.1.3 Object landmarks

Establishing *correspondence* between multiple images of an object lies at the heart of classic vision problems, like structure-from-motion [Fitzgibbon and Zisserman, 1998], and object matching [Sivic and Zisserman, 2003]. Correspondences across multiple views of the *same* scene was achieved by finding discriminative *interest points*, *e.g.* Harris corners [Harris and Stephens, 1988], or SIFT features [Lowe, 1999], which are invariant to geometric transformations (*e.g.* affine, or perspective), changes in viewpoints, illumination, and partial occlusion. However, these points are based on photometric considerations (*e.g.* gradients), and are not *semantically* consistent, *i.e.* they do not consistently detect

parts of objects. Detecting *semantic* parts is a well-studied area, going back to the classic *Pictorial Structures* work of Fischler and Elschlager [1973]. These methods model objects as collections, or *constellations* [Fergus et al., 2003, Weber et al., 2000] of structured *parts*. For example, human faces, can be modelled as collection of eyes, nose, mouth, and ears, and human bodies a collection of torso, head, and limbs. Separate appearance models are learnt for each part, with geometric constraints on their placement [Felzenszwalb and Huttenlocher, 2000]. Detecting parts provides detailed localisation and improves detection performance [Felzenszwalb et al., 2010b]. Further, part-detectors have been successfully learnt for both rigid [Xiang et al., 2014b], and non-rigid objects [Ramanan, 2007]. More recently, ConvNets have been augmented with top-down and bottom-up cues for learning object keypoints in both fully-supervised setting [Newell et al., 2016] and unsupervised setting [Thewlis et al., 2017a, Zhang et al., 2018], given only object category specific images. Learning object keypoints can be seen as an extreme variant of localisation; in fact object recognition, detection, and semantic segmentation are instantiations of the idea of learning detectors at varying levels of spatial locality. Keypoints provide correspondence across different object instances, and are useful for matching, detailed transfer of attributes [Zhou et al., 2015] and recovering 3D structure. Our work (chap. 5) also learns from category specific images given no landmark annotations; we apply our method to learn consistent landmark detectors for human faces, articulated human pose, and rigid 3D objects. Note, although no explicit supervision about object parts is provided during training, these landmarks learn to track geometrically similar locations, *e.g.* corners of lips, or eyes in faces, across different instances. Hence, these landmarks can be associated with *semantic* object parts *a posteriori*, either through direct correlation with manually labeled locations, or by learning regressors from the discovered landmarks to manually labeled ones.

1.2 Contributions and outline

In this section, we summarise the contributions of this thesis, and provide an outline of the chapters.

Synthetic scene text image generation. Jaderberg et al. [2014a] proposed a large-scale dataset of synthetic *cropped*-word level images for training text recognition networks. In chapter 3, we extend this to full *scene-level* images, with synthetically generated text instances embedded in them. The primary aim is to generate a large, fully-annotated dataset for training text detection, and joint localisation-and-recognition ConvNets. We propose a synthetic engine which: (1) produces realistic scene-text images so that the trained models can generalise to real (non-synthetic) images, (2) is fully automated and, (3) is fast. The engine operates on scene images downloaded from the Internet and models the local geometry to perspectively warp the text instances to lie on the surfaces. Further, it restricts the text instances to regions of uniform text and colour, to avoid crossing strong image discontinuities. We demonstrate that generating *realistic*-looking text instances, by modelling the scene geometry and segments, leads to better results than training with a simple dataset with text rendered in fronto-parallel orientation without regard to the scene geometry. Our dataset, *SynthText in the Wild*, has become one of the most widely used datasets for training text spotting models.[2]

Fully-convolutional text detector. In chapter 3, we further propose a fast fully-convolutional text detection network which localises the text instances at the word level. Our network, inspired by *Fully Convolutional Network* (FCN) for image segmentation [Long et al., 2015a], regresses dense text-detection predictions. Differently from FCN, the prediction is not just a binary class label (text/not text), but the parameters of a bounding box enclosing the word centred at that location. The latter idea is borrowed from the *You Only Look Once* (YOLO) technique of Redmon et al. [2016b] but with a crucial difference: the final fully-connected layers are replaced with convolutional re-

[2]The dataset is available at: http://www.robots.ox.ac.uk/~vgg/data/scenetext

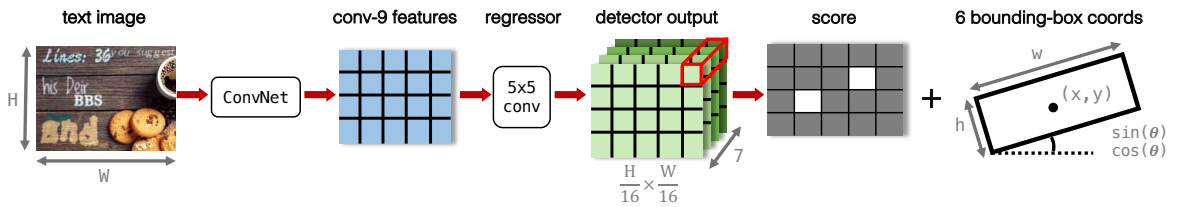


Figure 1.3: Our text localisation network. An input scene-text image of dimensions $H \times W$ (height $\times$ width), is mapped through a stride-16 convolutional network. A convolutional regressor with filters of dimensions 5×5 operates on the conv-9 features and outputs 7 bounding-box parameters at each feature location, namely the confidence score, height (h), width (w), location (x, y), and the orientation angle ($\cos(\theta), \sin(\theta)$). Final predictions are obtained through non-maximal suppression, or optionally further refinement based on the text-recognition scores. See chapter 3 for further details.

gressors, which drastically reduces the parameters (by over 90%), and speeds up learning. Figure 1.3 summarises the architecture of our localisation network. Our detector replaces the multi-stage region proposal based text detection framework of Jaderberg et al. [2015b], which generated a large number of proposals for high-recall, followed by multiple-stages of cascaded filtering and bounding-box refinement. At the time of publication, our network significantly advanced the state-of-the-art, both for text-detection and end-to-end word spotting (when paired with the recognition network of Jaderberg et al. [2014a]). Since then, several improvements have been proposed. Notably, our detector was perhaps one of the first to adapt a fully-convolutional architecture for detection, which was only later popularised by Dai et al. [2016] for general object detection.

Unsupervised text recognition. In chapter 4 we develop a method for text recognition in images without any paired labelled training examples. This enables fully automated and unsupervised learning from just line-level text-images, and unpaired text-string samples, obviating the need for large aligned datasets. This is a significant advance over the current state-of-the-art, which relies on millions of annotated examples for training [Jaderberg et al., 2014a, Tesseract OCR, 1985 – 2018]. We formulate the text recognition problem as one of aligning the conditional distribution of strings predicted from given text images with lexically valid strings sampled from target corpora. This alignment is induced through an *adversarial* discriminator [Goodfellow et al., 2014a], which tries to distinguish

Figure 1.4: Discovering Object Landmarks. In chapter 5 we develop a method for learning object landmarks, given only unlabelled object category specific images/videos. Here we visualise the landmarks discovered by our method on — (1) humans, (2) faces (unsupervised, and regression to annotated landmarks in the first, and second rows respectively), and (3) 3D objects. The landmarks consistently localise object parts across different pose, and instances.

the predicted characters, and real text strings apart. To the best of our knowledge, this is the first work which learns to decode a sequence of discrete symbols from images without any supervision. We present detailed analysis of the impact of various factors on the convergence of the method, and demonstrate excellent text recognition accuracy on both synthetically generated text images and scanned images of real printed books, using no labelled training examples.

Unsupervised discovery of object landmarks. In chapter 5 we develop a method for learning object landmark detectors which consistently track similar geometric locations in object across different instances, given only unlabelled object category specific images/videos. Our approach learns from pairs of images (denoted *source*, *target*) of objects that differ by time and/or viewpoint. At the core, is a *conditional auto-encoder*, which learns to generate the target frame, given only the source frame, and a sparse key-point like representation extracted from the target frame. Since, the two images differ in object pose, the representation is encouraged to distil the spatial location of object parts. The strength of the method resides in the simplicity of the approach and its generalisation ability to complex datasets without modification. Furthermore, we demonstrate learning from synthetically-generated image deformations, or raw videos directly, as unlike other methods our method does not require access to correspondences, optical-flow, or transformations. We also show that our method outperforms the existing state-of-the-art methods for facial landmark detection. Figure 1.4 visualises the unsupervised landmarks discovered by our method on faces, humans, and 3D object categories.

Improving generalisation in RNNs. In chapter 6, we address the problem of poor generalisation of *recurrent neural networks* (RNNs) to sequences of length beyond those present in the training set. RNNs are trained end-to-end on finite sequences and may not learn the correct loop invariant required to generalise to arbitrary sequence lengths. We demonstrate that this is indeed the case, and propose to train them instead for one-step *inductive* updates. The idea is to first decompose the problem into a sequence of inductive steps, and then to explicitly train the RNN to reproduce such steps. Generalisation is achieved as the RNN is not allowed to learn an arbitrary internal state, but is tasked with mimicking the evolution of a valid state. In particular, the state is restricted to a spatial memory map that tracks parts of the image which have been accounted for in previous steps. The RNN is trained to update the memory in addition to producing the desired output. We evaluate our method on two different visual recognition problems involving visual sequences: (1) end-to-end text spotting, *i.e.* joint localisation and recognition of text

in images containing multiple lines (or a block) of text, and (2) sequential counting of objects in aerial images. We show that inductive training of recurrent models enhances their generalisation ability on challenging image datasets.

1.3 Publications

The work presented in this thesis has been published at the following venues:

- Chapter 3: A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- Chapter 4: A. Gupta, A. Vedaldi, and A. Zisserman. Learning to read by spelling: Towards unsupervised text recognition. *11th Indian Conference on Computer Vision, Graphics and Image Processing*, 2018b.
- Chapter 5: T. Jakab*, A. Gupta*, H. Bilen, and A. Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Advances in Neural Information Processing Systems*, 2018. The first two authors contributed equally; see the statement of authorship at the end of the chapter.
- Chapter 6: A. Gupta, A. Vedaldi, and A. Zisserman. Inductive visual localisation: Factorised training for superior generalisation. In *Proceedings of the British Machine Vision Conference*, 2018a.

A note on presentation. This thesis is presented as an *integrated thesis*,[3] where the publications are reproduced in the format they were submitted for publication at the respective venues.

[3] https://www.admin.ox.ac.uk/edc/policiesandguidance/policyonresearchdegrees/annexbintegratedthesesguidancefordivisionalboards/

2

Literature Review

In this chapter, we first review developments in learning under three different supervision paradigms, namely — (1) learning from *synthetic data* (sect. 2.1), (2) self-supervision from *temporal sequences* (sect. 2.2), and (3) *adversarial* supervision for learning from unaligned data (sect. 2.3). Since, the dominant application domain of our work is *text spotting* (sect. 2.4), we review it next in detail, under three sub-sections — (1) methods for detection or localisation of text instances (section 2.4.1), (2) recognition of text from cropped text images (section 2.4.2), and (3) recent methods which combine the above two stages, into an end-to-end unified framework (section 2.4.3).

2.1 Learning from synthetic data

Synthetically generated datasets provide detailed ground-truth annotations, and are cheap and scalable alternatives to manually annotated labels. Further, detailed per-pixel annotations like — optical flow, surface normals, stereo disparity, and segmentation masks — can be extremely expensive (or impossible) to obtain manually, *e.g.* high-quality semantic labeling required 60 minutes per image for the CamVid dataset [Brostow et al., 2009], and 90 minutes per image for the Cityscapes dataset [Cordts et al., 2016]; synthetic

datasets provide an economical alternative. A classic example of the success of synthetic data in computer vision applications is the use of synthetically generated depth-maps to train the body-parts and joints classifiers for *Microsoft Kinect* depth sensor [Shotton et al., 2011]. Instead of generating realistic intensity images using computer graphic techniques, which suffer from domain gap, they re-target motion capture data to pre-defined human mesh models, and render noisy depth-images and corresponding body-part labels; Sharp et al. [2015] similarly train articulated hand-pose estimators and trackers from simulated depth images.

Optical flow. Optical flow has a rich tradition of benchmarking and learning from simulated sequences. The classic work of Barron et al. [1994] evaluates several optical flow methods on real and simulated image sequences, which range from simple sinusoids and moving squares to the famous *Yosemite sequence* rendered synthetically from texture-mapped depth map of the valley. The *Middlebury* dataset [Baker et al., 2011] extends the former with more complex scenes with larger motion ranges, more realistic texture, independent motion, and with more complex occlusions. More recently *MPI-Sintel* [Butler et al., 2012], an optical flow dataset derived from the open source 3D animated short film *Sintel* has become the standard benchmark; it has been extended to include other dense annotations like depth, stereo disparity and bottom-up segmentation. Similar in spirit, is the animated film dataset by Mayer et al. [2016], but is larger to facilitate training of deep ConvNets. Dosovitskiy et al. [2015a] were perhaps the first to train deep ConvNets for optical flow from synthetic non-photorealistic renderings of 3D CAD models of *flying* chairs blended on random background images from Flickr; they demonstrate surprising good generalisation to real datasets, even without fine-tuning.

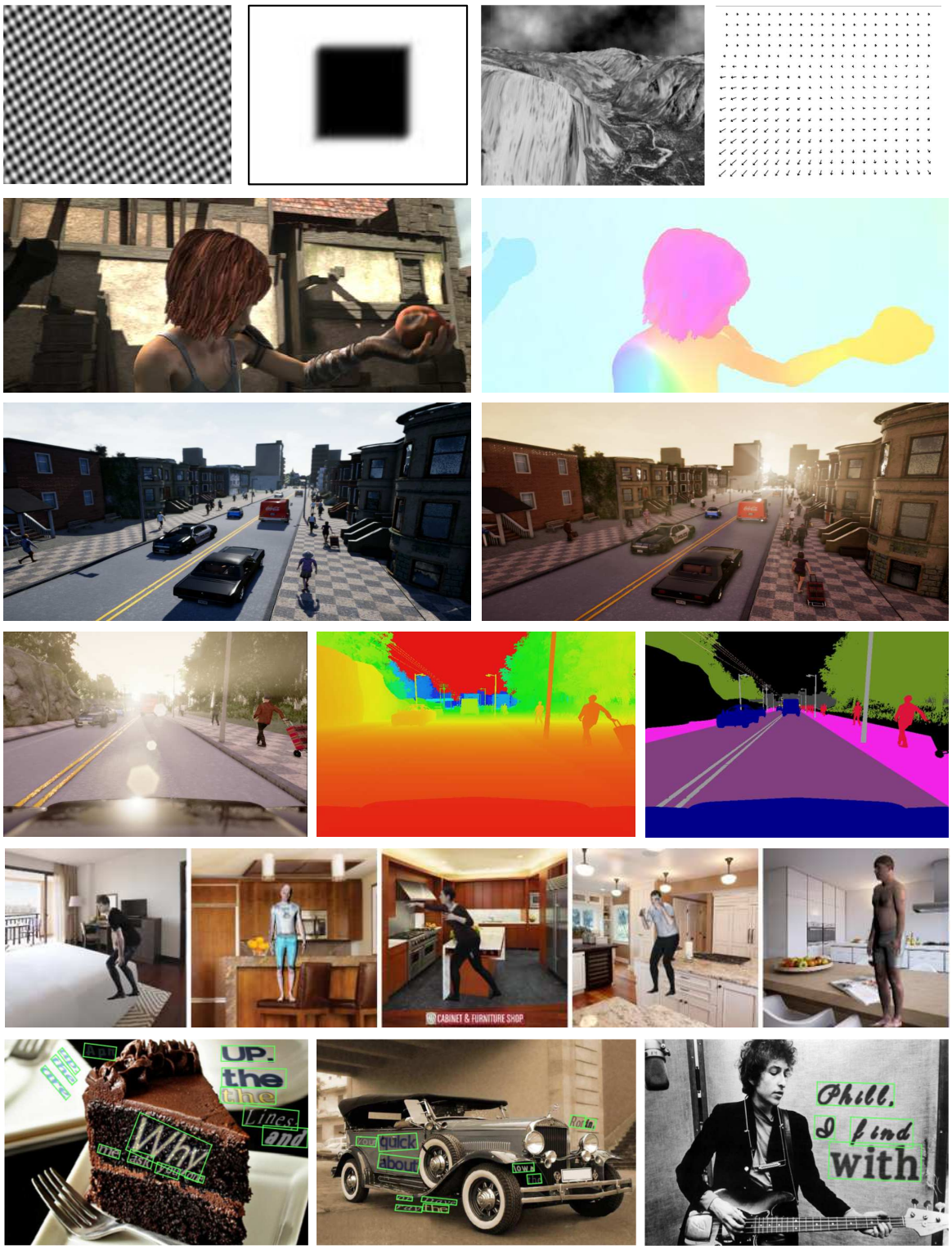


Figure 2.1: Montage of synthetic datasets. Synthetic datasets are a scalable and cheap source of supervision for the data hungry deep learning methods for vision. **[top – bottom]:** Synthetic sequences introduced by Barron et al. [1994] for evaluating optical flow methods — sinusoids, moving square, *Yosemite* sequence with its ground-truth flow. A frame and corresponding optical flow ground-truth from the *Sintel* movie [Butler et al., 2012]. Urban driving scene from the *CARLA* simulator [Dosovitskiy et al., 2017] under different lighting conditions; depth and semantic segmentation annotations for another frame. Synthetic humans from the SURREAL dataset [Varol et al., 2017] for learning 3D human pose. Samples from our synthetic dataset for text spotting [Gupta et al., 2016] (chap. 3). Images reproduced from respective publications.

Human pose and detection. Another domain that has benefitted immensely from synthetic data is human detection and pose estimation. Marin et al. [2010] train pedestrian detectors on renderings from the *Half Life 2* game engine, and evaluate on real world images. Pishchulin et al. [2011] instead use detailed morphable 3D human models acquired using a multi-view motion capture setup, and generate training data by mapping a few pedestrian example images onto articulated 3D meshes. In [Pishchulin et al., 2012] they replace the expensive motion acquisition step with 3D pose recovery from single images. In the follow on work Vazquez et al. [2014] address the domain gap between real and synthetic images by embedding the features from real and synthetic samples into a common space. Park and Ramanan [2015] take videos with human-part annotations only for the first frame, and generate tiny videos for pose estimation by warping the parts in subsequent frames; they demonstrate the effectiveness of simple nearest-neighbour classifiers due to abundance of synthetic data. More recently, Rogez and Schmid [2016] use image-based synthesis to generate images annotated with 3D human-pose information from a dataset of images with 2D keypoints and an independent dataset of 3D motion capture data. This is achieved by finding images with poses closest to the projection of a given 3D pose, and then seamlessly stiching these images together; they train a ConvNet on these image to classify into $k-$pose clusters. Varol et al. [2017] (*SURREAL* dataset) and Chen et al. [2016], directly render a textured 3D human mesh model, and blend on randomly selected background images, in a non-photorealistic manner.

3D object pose and viewpoints. A parallel use of 3D CAD models has been in 3D object detection and viewpoint estimation. Pepik et al. [2015] extend the deformable-parts-model (DPM [Felzenszwalb et al., 2008]), with geometric cues from 3D CAD models and jointly estimate the viewpoint and localise objects in 3D; they also employ the CAD data to enrich the appearance model with rendered images. Gupta et al. [2014] augment RGBD datasets with rendered 3D CAD models and demonstrate impressive improvements in 3D object localisation, instance and semantic segmentation, by using geocenteric embedding for depth images. Peng et al. [2015] explore the impact of various components, like

texture and color by rendering synthetic 3D objects for the object detection task. Su et al. [2015] estimate viewpoint from single 2D images by training on rendered images, and show significant improvements on the *PASCAL 3D+* benchmark [Xiang et al., 2014a].

Simulation engines. A recent trend of exploiting learning in/from completely virtual simulation engines has emerged. The driving forces behind this are twofold — (1) need for large datasets to train deep learning models, and (2) advancements in photo-realisitic (although realism is not a neccessary component), and real-time (which enables fast data generating) rendering. A prime application has been in training perception modules for autonomous driving systems. The recent *CARLA* urban driving simulator of Dosovitskiy et al. [2017] provides a physics engine backed environment with photo-realisitc renderings, complex urban scenes, varied seasons, lighting *etc*., which is much more complex and realistic than the *TORCS* racing car simulator [Wymann et al., 2014]. The *Synthia* dataset [Ros et al., 2016] similarly provides dense pixel-level ground-truth for semantic segmentation tasks in virtual urban environments. In [Richter et al., 2016, 2017] they further extend the gamut of annotations to include semantic *instance* segmentation, 3D scene layout, visual odometry, and optical flow, by injecting a *detouring* middleware between the popular *GTA* game and graphics hardware.

Perhaps, the most successful application of learning in simulation engines is the seminal work on training reinforcement learning agents directly from raw video pixels in *Atari* games [Mnih et al., 2015]. While, *Atari* provides a challenging suite of evironments for control, these environments are perceptually simple. Recently, Xia et al. [2018] have extended the idea to create perceptually challenging *indoor* virtual worlds which are based on real world scene geometry, physics and textures, to enable *embodied learning*. This is similar in spirit to the recent earlier works of Handa et al. [2016], McCormac et al. [2017] who use photorealistic renderings of indoor scenes for scene-recognition and segmentation in RGB-D images.

Other applications. Synthetic data has been used for gaze estimation [Shrivastava et al., 2017, Wood et al., 2016], hand pose estimation [Fanello et al., 2014, Shrivastava et al.,

2017, Supancic et al., 2015, Tompson et al., 2014], shape from shading [Richter and Roth, 2015]. Synthetic data is especially suitable for *analysis by synthesis* methods: Yildirim et al. [2015] combine a generative model based on a realistic 3D graphics engine with a recognition ConvNet model fine-tuned by brief runs of MCMC inference, to learn detailed recognition — shape, texture, pose — of human faces. Dosovitskiy et al. [2015b] train a *generative* neural renderer for 3D chairs from renderings, and learn representations which allow measuring similarity, and interpolation between different instances.

Synthetic data as a diagnostic tool. Synthetic data has been used extensively for the analysis of computer vision methods, specifically their sensitivity and invariance to the various nuisance factors — *e.g.* lighting, texture, pose. Utilising synthetic data provides precise control over the latent factors, which is not practical with real world data, and lends a unique setting for careful analysis of various methods. McCane et al. [2001] study optical flow methods by carrying out *factorial experiments* by systematically varying the amount of — scene complexity, object and camera motion. LeCun et al. [2004] create a curated dataset (*NORB*) of synthetically rendered common objects and study the impact of surrounding clutter on object category recogition methods by placing the objects against synthetic background images. Kaneva et al. [2011] evaluate descriptor performance under controlled changes in viewpoint, scene, and illumination in a photorealistic virtual world. Gaidon et al. [2016] "clone" the real video sequences from the *KITTI* dataset [Geiger et al., 2013] into virtual enviroments, and introduce a *Virtual KITTI* dataset. They study the impact of lighting conditions (*e.g.* fog), and camera angles, which is impractical in real-world conditions, on object detection, tracking, segmentation and optical flow. Aubry and Russell [2015] investigate the features of deep ConvNets through transformations (*e.g.* rotation, style *etc.*), of renderings from a large database of 3D CAD models; they find increasing invariance to viewpoint in deeper layers. Finally, Johnson et al. [2017] introduce the *CLEVR* diagnostic dataset for studying the compositionality of neural architectures for the visual question answering task; they render simple object shapes under complex spatial arrangements, and use the detailed ground-truth information for

framing textual questions and answers.

2.1.1 Synthetic data for text images

Printed text is an artificial construct, which makes it highly amenable to synthetic generation. In fact, modern research both in document OCR (optical character recognition) and scene-text recognition, relies heavily on synthetically rendered text images for training. The popular open-source *document* OCR system, *Tesseract* [Tesseract OCR, 1985 – 2018], renders text sourced from corpora of respective languages in different fonts to generate the training images. For the more challenging setting of *scene* text recognition, Wang et al. [2012] were perhaps the first to render images of single characters, with surrounding clutter from other characters, in different fonts and styles, to train a ConvNet; they augment their training set with real single-character images from the ICDAR 2003 [Karatzas et al., 2013], and the Chars74K [Campos et al., 2009] datasets. Goodfellow et al. [2014b] extend recognition to multiple characters, and train their models on synthetically generated *CAPTCHA* images, achieving excellent accuracy (99.8%). Jaderberg et al. [2014a] train their word-level (multiple characters) recognition models solely on *synthetic* data and achieve excellent accuracy on *real* test images. Note, this is a crucial departure from testing on *CAPTCHA* images, which themselves are synthetically generated. To achieve this significant generalisation from synthetic to real images, they propose a sophisticated synthetic rendering engine: font is sampled from a catalogue of over 1400 fonts; inset border, outset border or shadow widths are randomly determined; foreground and border/shadow colours are selected; the text is distorted through a projective distortion to simulate camera effect, and finally the text is blended against a real image background with gaussian or JPEG compression noise and blur. They generate a dataset of 9 million word images, which has become the standard training set for scene-text recognition methods. In our work, we further extend the idea of synthetic text images to *full scene* images to enable training *localisation* methods. Details are presented in chapter 3; in summary: for a given scene image downloaded from the internet, we first perform bottom-up segmentation to

constrain the text instances to regions of uniform colour and texture; next, we regress depth using single-image depth estimation neural networks to fit local planar regions to these segments — this is to perspectively deform the text according to the local geometry; finally, we render the text in various styles and colours and blend into the base image. Our dataset has become the standard for training text localisation networks, and is used extensively by the community. It has recently been extended by Zhan et al. [2018] to constrain the placement of text to semantically meaningful regions. Finally, synthetic text images have also been employed to train font-recognition networks [Wang et al., 2015].

2.2 Self-supervision from temporal sequences

While single images have been the main application domain for deep learning in vision, the inherent consistency and temporal coherence in videos (consecutive frames) has long been identified as a source of supervision. The classic work of Földiák [1991] learns invariant features from transformation sequences. Wiskott and Sejnowski [2002] in *Slow Feature Analysis* (SFA), identify *slowly* varying underlying structure in quickly varying primary sensory data, as external causes of the input signal. They learn unsupervised features from videos based on a reconstruction loss. Mobahi et al. [2009] employ the contrastive loss framework of Hadsell et al. [2006] and enforce the embeddings of nearby frames to be close, while pushing away those from randomly sampled frames. Zou et al. [2012] learn hierarchical (multi-layer) invariant features in an autoencoder framework using temporal slowness constraint through tracking. Goroshin et al. [2015] enforce sparsity on the learnt features and propose a principled method to trade-off the discriminability and stability of the learnt representations. The above methods only impose smoothness on two consecutive frames; Jayaraman and Grauman [2016] extend this to multiple frames, by imposing the higher-order derivatives to be small. Memisevic [2013], Taylor et al. [2010] train convolutional gated RBMs and Energy models to learn latent representations from pairs of successive images. Srivastava et al. [2015] cast the problem of video auto-encoding

and future-prediction in the recurrent *encoder-decoder* framework [Cho et al., 2014]: video frames are summarised into a single context vector through a recurrent encoder, which is then decoded by two separate recurrent decoders, one for reconstruction of the previous frames, and another for regressing future frames. Instead of collapsing the past context into a single fully-connected vector, Patraucean et al. [2015] learn convolutional features, with a bilinear sampler to deform the past aggregated context into future features; they apply their method for label propagation using optical flow for weakly supervised semantic segmentation.

Another set of works design *pretext tasks* for *self-supervision* in videos to induce meaningful representations. Once trained, such representations are re-purposed for target applications like object classification, tracking and segmentation, by first optionally fine-tuning them on small amount of labelled data. Isola et al. [2015] learn to classify whether two frames belong within a defined time-range (3 seconds); they show application on *shot-detection*, *i.e.* segmenting a movie into clips. Wang and Gupta [2015] track patches, based on motion of SURF [Bay et al., 2006] interest points, and design a Siamese-triplet network with a ranking loss function [Wang et al., 2014] to pull embeddings for related patches closer than embeddings for a randomly selected patch; Gao et al. [2016] do the same but for region proposals. Pathak et al. [2017] learn ConvNet features to segment foreground objects in videos by relying on an off-the-shelf video segmentation method [Faktor and Irani, 2014]; they show transfer to object recognition and segmentation tasks. Fernando et al. [2017], Lee et al. [2017], Misra et al. [2016], Wei et al. [2018] exploit the underlying chronological order of frames and induce features by sorting shuffled video frames; the network must develop an understanding of underlying object dynamics to correctly solve this task.

Inspired by *efference copy* — the internal copy of outflowing motor signals — some works augment the visual stream (videos) with *ego-motion*. Agrawal et al. [2015] induce representations by predicting the *known* transformation between a pair of frames or images. Jayaraman and Grauman [2015] learn embeddings from pairs of images which are

equivariant to the relative motion. More recently, Gupta et al. [2017] and Henriques and Vedaldi [2018] learn differentiable allocentric spatial representations (maps) from egocentric videos and motion information. A related line of works focus on unsupervised learning of single-image depth regression from monocular videos. Zhou et al. [2017a] warp a source image to a target frame (separated by acquisition time) by jointly predicting dense depth-maps and the relative camera motion. Vijayanarasimhan et al. [2017] similarly learn unsupervised depth regression, but additionally learn object motion models, while Zhou et al. [2017a] discount moving objects using an *explainability* mask.

This idea of learning from *cross-modal* data, has recently been explored in the *audio-visual* setting. Aytar et al. [2016] and Harwath et al. [2016] learn to correlate videos and images respectively with audio-streams. They cast this problem in the *student-teacher* framework [Hinton et al., 2015], and learn audio features to align with pre-trained deep visual features. This is similar to the *SyncNet* framework [Chung and Zisserman, 2016], which was extended recently by Arandjelovic and Zisserman [2017] to train both audio and visual features jointly; they demonstrate detailed localisation of objects in both the modalities. In Arandjelovic and Zisserman [2018], they learn a common embedding for audio and visual domains, and show cross-modal retrieval. Owens et al. [2016a] learn a generative model for realistic plausible impact sounds from silent videos. Owens et al. [2016b] show learning to predict a summary for plausible ambient sounds from videos can induce representations which convey information about objects and scenes.

2.2.1 Unsupervised learning of object landmarks.

In chapter 5, we present a method (a conditional autoencoder) for unsupervised learning of object landmarks from a pair of frames, say $\mathbf{x}$ and $\mathbf{x}'$, sampled from object category specific videos. This is achieved by reconstructing $\mathbf{x}'$ from $\mathbf{x}$, given only very constrained information $\Phi(\mathbf{x}')$ about $\mathbf{x}'$. In particular, we constrain $\Phi(\mathbf{x}')$ to be a set of K 2D-keypoints, obtained by marginalising 2D feature-maps which are learnt by the model. We sample $\mathbf{x}$ and $\mathbf{x}'$ either from videos, or generate two randomly warped versions (using Thin Plate

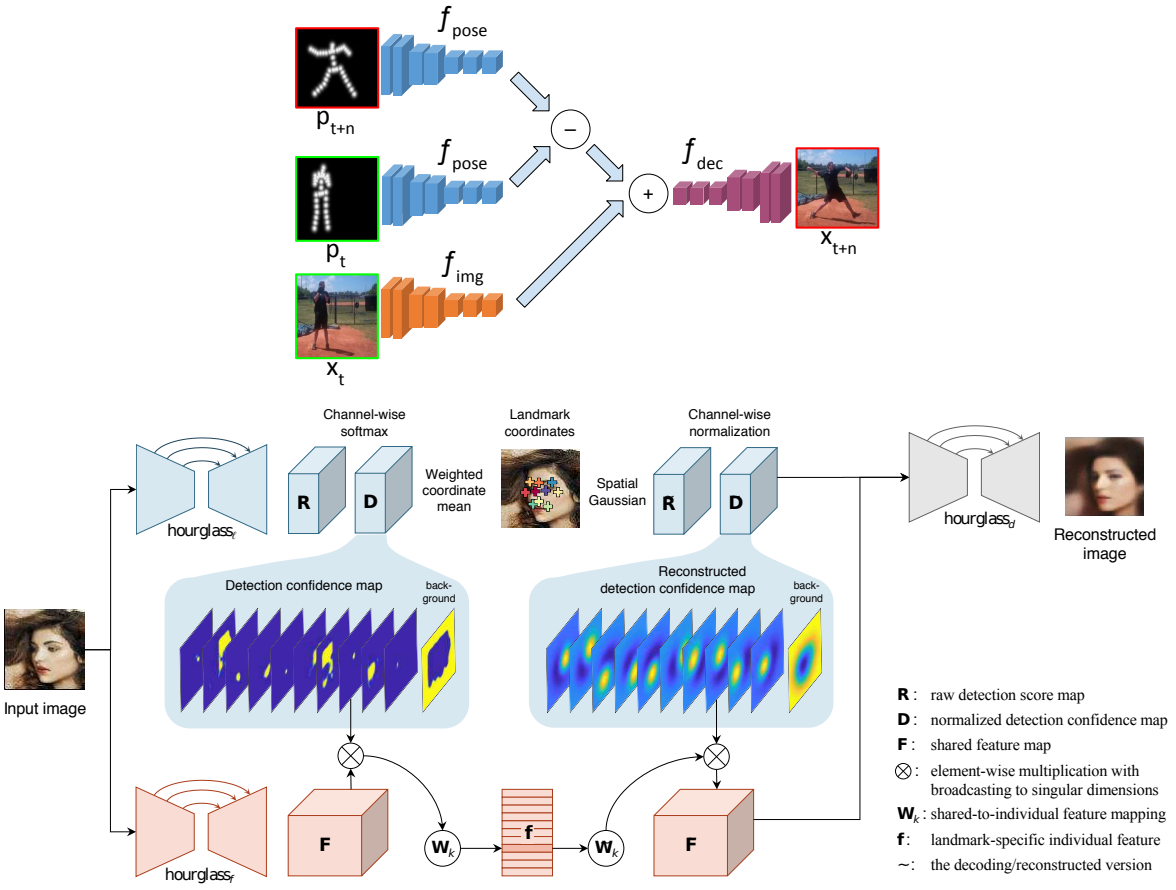


Figure 2.2: Unsupervised learning of landmarks. In chapter 5, we present a method for unsupervised learning of object landmark from a pair of frames. It is based on *analogy networks* [Reed et al., 2015], employed by Villegas et al. [2017] **[top]** who render a future frame ($\mathbf{x}_{t+n}$), given a past frame ($\mathbf{x}_t$), and the past (p_t) and future landmarks (p_{t+n}). We essentially *invert* this, and learn the landmarks as sparse 2D keypoint-like latents by reconstructing the future frame from the past frame. **[bottom]:** Zhang et al. [2018] similarly learn keypoints from image reconstruction but do not condition their generation on a second image. Images reproduced from respective publications.

Splines [Wahba, 1990]) of a single image. We demonstrate learning consistent object landmarks which track similar geometric locations across different instances, for a variety of object categories, namely — faces, humans, 3D objects, and digits.

In spirit, our method is similar to the earlier works of Taylor et al. [2010] and Memisevic [2013], who also learn representation bottlenecks from conditional reconstruction. However, recently several methods have been proposed concurrent to our work, for similar *unsupervised* learning of object landmarks or correspondences. Amongst these, most similar is the work of Zhang et al. [2018]. They also learn object category-specific

keypoint detectors through image reconstruction, however, there is a crucial difference: they do not use a *pair* of images to condition their reconstruction, and instead rely on a single image, as in a conventional auto-encoder framework. This necessitates an elaborate representation bottleneck, and a multi-objective loss function: they extract keypoints from 2D heatmaps as we do, but use them to spatially *transport* features extracted from the source image for reconstruction; we do not require such an explicit feature-transport mechanism. Further, reconstruction loss alone does not suffice, and they impose several constraints to discourage learning arbitrary representations instead of the desired keypoints, namely — (1) *concentration*: encourage point-like concentration of heatmaps; (2) *separation*: discourage keypoints to degenerate to the same location; and, (3) *equivariance*: encourage equivariance to geometric transformations. In ablation experiments, they find the constraints (2) and (3) (borrowed from Thewlis et al. [2017a]) to be crucial for learning keypoints in addition to the image-reconstruction loss; while our work requires only reconstruction for learning.

The idea of conditional image generation is inspired from the work of Reed et al. [2015], who propose a method for *visual analogy making*, which is the task of transforming a query image according to an example pair of related images. To do this, they learn feature embeddings which are transformed through vector algebra [Mikolov et al., 2013] to complete the analogy. The concept of neural analogies was adapted by Villegas et al. [2017] to render future image-frames of human activity videos, given past frames and keypoint representation of the pose in the past and future frames. Our method essentially *inverts* this, and learns keypoints given future and past frames (see fig. 2.2). Wiles et al. [2018a] also propose a similar framework, where they learn a continuous embedding for faces, and regress facial landmarks, expression, and pose from it. There are three crucial differences: (1) they learn a continuous embedding which further needs labelled training samples to regress landmarks, while we directly learn sparse landmarks; (2) they use a *bilinear sampler* to warp the source frame to the target frame (as in Patraucean et al. [2015]), while we learn a neural renderer, which can hallucinate the missing details (*e.g.* due to self-occlusion); and finally (3) their application domain is human faces (however, their

method is applicable to other object categories, without change). Conditional generation has recently been employed in Esser et al. [2018] to generate pose-conditioned images for humans, shoes, and handbags, and in Wiles et al. [2018b] for human faces. The recent work of Vondrick et al. [2018] learn to track, *i.e.* learn correspondences, by transferring colours from a reference frame to a given gray-scale frame, sampled from the same video. This approach is similar to the conditional generation framework of Wiles et al. [2018b], as they also learn feature embeddings for dense correspondence in a pair of video frames; here, the image generation in replaced with colourization.

Finally, our method is related to that of Thewlis et al. [2017a] who learn object keypoints in an unsupervised manner by exploiting geometric equivariance: for a given image $\mathbf{x}$, they deform it using a *known* Thin-Plate Spline transformation $g(\cdot)$ to generate a second image $\mathbf{x}' = g(\mathbf{x})$, and enforce equivariance between the detected keypoints: $g(\Phi(\mathbf{x})) = \Phi(\mathbf{x}')$, where $\mathbf{x}' = g(\mathbf{x})$, and $\Phi(\cdot)$ is the learnt keypoint detector. In Thewlis et al. [2017b], they extend their method to learn dense correspondences / object frames. Recently, Suwajanakorn et al. [2018] extend their idea of equivariance for learning 3D keypoints: they learn keypoint detectors on pairs of images of an object under two different views with a *known* relative transformation. They optimise an ordered list of 3D keypoints which is consistent with both the views. However, their method is *not* fully unsupervised, as — (1) they rely on pairs of views with a *known* relative offset; (2) crucially, they use a *dominant direction* signal to break symmetries, and (3) they explicitly constrain the keypoints to lie inside the object silhouette, using known object segmentation mask.

2.3 Adversarial learning from unaligned data

The seminal work of Goodfellow et al. [2014a] introduced *Generative Adversarial Networks* (GANs), which employ an *adversarial discriminator* to align the distributions of generated and real data samples. They cast this as a two-player minimax game, where the generator G tries to generate samples from noise z which fool the discriminator D, while D learns to

tell them apart from the real samples. In other words, G and D simultaneously optimise the following value function:

$$\min_G \max_D \mathop{\mathbb{E}}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathop{\mathbb{E}}_{z\sim p_z(z)}[\log(1-D(G(z))].$$

Following the original presentation aimed at learning *generative* models, the framework has been successfully applied to learn deep generative models for images [Radford et al., 2015], 3D object shapes [Wu et al., 2016], and future prediction in videos [Mathieu et al., 2015], to name a few. Recently, Karras et al. [2018] generate HD-quality celebrity face images by progressively growing the generator and discriminator networks, starting from a low resolution followed by gradual addition of layers to model finer details.

In chapter 4 we present a method for recognizing text in images without using any labelled data. Hence, of more direct interest to us, is the alternative application of the adversarial loss function in learning from *unaligned* source and target examples. While in the original formulation, the generator acts on random noise, here, it learns to *translate* samples $G : X \rightarrow Y$, from the source domain X, to the target domain Y; the discriminator pushes the generated "predictions" $G(X)$ to be indistinguishable from the target distribution Y. A prime example of this is the *CycleGAN* framework of Zhu et al. [2017] (fig. 2.3). They learn to translate images from one domain to another, *e.g.* they change the texture of photos to look like *van Gogh* paintings, without any paired image examples. They learn a second inverse mapping $F : Y \rightarrow X$ and enforce reconstruction $\mathbf{x} \triangleq F(G(\mathbf{x}))$, to encourage strong correlation between the input and the generated output, in order to avoid the degenerate failure mode of collapsing to the same output instance regardless of the input. The earlier *Pix2Pix* work [Isola et al., 2017] from the same group, which is an instantiation of the *Conditional GAN* framework of Mirza and Osindero [2014], also learnt image-to-image translation, but relied on *paired* examples.

CycleGAN is a general framework for learning from unaligned examples, and has recently been applied to language translation. Lample et al. [2018b] learn a common latent

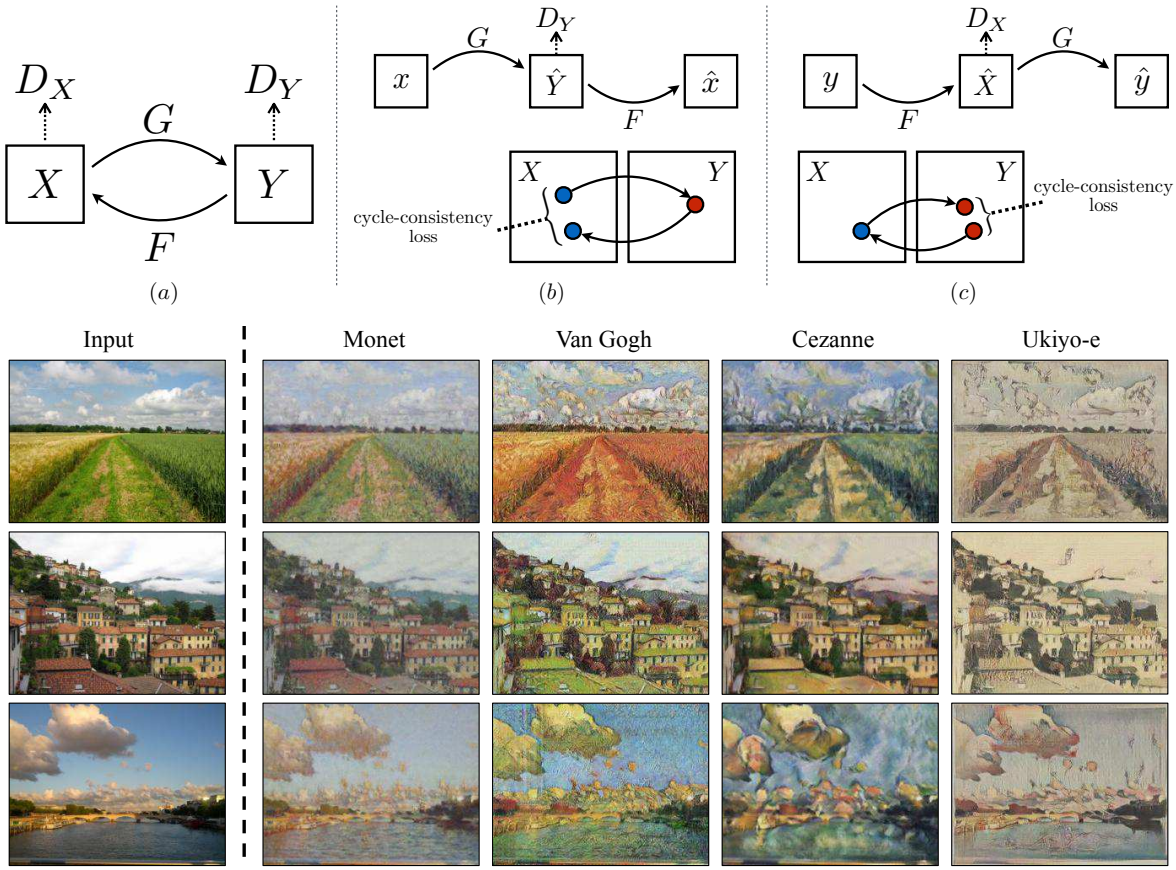


Figure 2.3: Adversarial learning from unaligned data. [top]: *CycleGAN* is a general framework for learning from unaligned examples, where samples from input domain (X) are mapped to (Y) by $G : X \rightarrow Y$, based on an *adversarial* discriminator (D_Y) which makes them indistinguishable from the domain Y. They ground the mappings by enforcing reconstruction $F(G(x)) \approx x$ of the input, where $F : Y \rightarrow X$ is the inverse mapping learnt simultaneously. **[bottom]:** Transfer of photos (X) to paintings (Y). In chapter 4, we similarly learn from unaligned data to recognise text in images. However, we do not reconstruct images from text-strings as it is highly underconstrained, but instead encourage grounding by making the recognition network local (see chap. 4 for further details). Images reproduced from Zhu et al. [2017]

space for the two languages, and learn to translate by reconstruction in the respective domains. They learn denoising auto-encoders for the two languages, but with a shared latent space, which is pulled together using an adversarial loss. Second, they enforce *back-translation* to the source language from a translated version of a sentence, as in *CycleGAN*. Back-translation is also employed by Artetxe et al. [2018] for unsupervised language translation, but they do not use adversarial training. Concurrent work in language translation by Zhang et al. [2017] and Lample et al. [2018a] induce bilingual lexicons from independently pre-trained monolingual word-embeddings (*e.g.* using *Word2Vec* [Mikolov et al., 2013]). They achieve this by mapping embeddings from one language to the other, and aligning them through adversarial training. This mapping is typically simple, *e.g.* linear, and is constrained to be orthogonal, for easy inversion and better generalisation. Another compelling application of *CycleGAN* has been in learning to break *ciphers* given no aligned examples. Gomez et al. [2018] propose *CipherGAN*, which applies *CycleGAN* to text-strings (sequence of one-hot characters), instead of images. Our work for unsupervised text recognition (chapter 4), combines *CycleGAN* and *CipherGAN*, in that it decodes text-strings from images. However, we do *not* learn an inverse mapping from text-strings to text-images, as it is highly underconstrained/ambiguous — rendering a given string requires sampling the background image, font style, font colour, geometry of the glyphs, shadows, noise etc. This ambiguity arises because text recognition requires translating between two very *different* modalities, namely text and images, which is much harder than translating within the *same* modality, *e.g.* between images in *CycleGAN* where only local texture is modified, or between character strings in *CipherGAN* where the characters are permuted. Instead, we ground the predictions to the input image by encouraging *prediction locality* in the recognition network (or "generator"); please refer to section 3 of chapter 4 for further details. Sutskever et al. [2015] also posit such predictor locality as being crucial for recovering the underlying relation between the input and output domains which contain long-range dependencies. They propose a "principled" method for unsupervised learning based on *output distribution matching*, *e.g.* through minimizing

the KL-divergence; the adversarial loss instead optimizes the *Jensen–Shannon divergence* [Goodfellow et al., 2014a].[1]

Using an adversarial loss to constrain the predictions to the prior distribution has been explored for a variety of visual tasks by Tung et al. [2017]. They propose the *adversarial inverse graphics* framework, which is aimed at learning visual recognition and generative models, *e.g.* egomotion estimation, 3D human pose from single image, and facial image generation and transformation. They adapt the *CycleGAN* framework, and only learn *one-arm* of the bi-directional mapping, *i.e.* from the input to output, but not the inverse. They enforce reconstruction to ground the predictions, and constrain the predicted latents using an adversarial loss. Recently, Kanazawa et al. [2018] also apply similar idea to recover dense 3D mesh models from a single image, with only 2D keypoints as the training supervision.

2.4 Text spotting in natural images

The dominant application domain of our work is *text spotting*, *i.e.* both localisation and recognition of text in *natural images*. Hence, we review methods for text localisation or detection (section 2.4.1), recognition (section 2.4.2), and finally the more recent end-to-end text spotting methods (section 2.4.3) which combine the two stages in a unified framework. This review focuses on methods for text spotting in natural images, as opposed to recognition in documents, which is a well-developed area itself.

2.4.1 Text detection methods

Pre-deep-learning methods for text detection can be classified to be based either on connected-components (section 2.4.1.1) or, on the traditional sliding-window approach (section 2.4.1.2); we review these first. Next, in section 2.4.1.3, we review the more recent

[1] The Jensen–Shannon divergence $JSD(P||Q)$ between two distributions P and Q, is a symmetrized and smoothed version of the KL-divergence $D(P||Q)$, where $JSD(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M)$, where $M = \frac{1}{2}(P+Q)$.

methods based on deep features.

2.4.1.1 Detection from connected components

Text detection methods based on connected-components (CCs), first find regions containing text and then connect them into text-lines or words. The initial text-region candidates are generated based, most commonly, on the following two local dense features — (1) *Stroke Width Transform* (SWT) of Epshtein et al. [2010], which groups together neighbouring pixels based on similarity of stroke-width, or (2) *Maximally Stable Extremal Regions* (MSERs) of Matas et al. [2002] which finds regions which are invariant under varying levels of intensity thresholds. Text detection methods of Neumann and Matas [2010, 2011, 2012] classify MSERs into text/no-text using SVMs, or through a cascade and link the neighbouring detections through exhaustive graph search. Chen et al. [2011] enhance the MSERs with edge information, and also use the SWT feature while eliminating non-text candidates. Yin et al. [2013] prune the MSERs by minimising regularised variations, while Huang et al. [2014] use a ConvNet classifier. Zhu and Zanibbi [2016] do a coarse-to-fine detection of character pixels using convolutional features, followed by extracting CCs from characters using edge and colour features, and finally perform a graph-based segmentation of the CCs into words. The more recent *Canny text detector* of Cho et al. [2016], also employs MSERs and props-up low-scoring regions though hysteresis based tracking; their method was successfully used in the ICDAR 2015 Robust Reading competition [Karatzas et al., 2015].

2.4.1.2 Detection by classifying sliding windows

Sliding window methods, examine image windows, or regions of interest, at multiple scales and positions, and score (or classify) them using pre-trained classifiers. Posner et al. [2010] threshold the classifier score heatmap to obtain text bounding-boxes, while Anthimopoulos et al. [2011], Chen and Yuille [2004], Quack [2009] use boosted classifiers and aggregate the positively scored windows. Quack [2009] based their detector on the

popular face-detector of Viola and Jones [2001], where boosted classifiers [Freund and Schapire, 1997] are learnt on top of simple rectangular block features. Wang et al. [2011] use synthetic data to train random-fern classifiers [Ozuysal et al., 2010] on HOG features [Felzenszwalb et al., 2010a] to detect characters, and later group them into words from a fixed lexicon.

2.4.1.3 Deep features for text localisation

Wang et al. [2012] first used ConvNets for localisation by producing a text saliency-map from classifier scores and then separating them into *horizontal* lines using non-maximal suppression. Their work was followed by Jaderberg et al. [2014b], who take this saliency map and also produce *horizontal* text-lines through row-wise run-length-smoothing and Otsu thresholding [Otsu, 1979], as done in traditional OCR methods. Girshick et al. [2014] introduced a generic object detection framework, where object proposals from external methods like Selective Search [Uijlings et al., 2013] are classified into objected categories based on convolutional features. This was then adapted for text localisation in Jaderberg et al. [2015b], where proposals from EdgeBoxes [Zitnick and Dollar, 2014a] and Aggregate Channel Feature detector [Dollar et al., 2014] were used.

In our work (chap. 3), we drop external proposal methods, and instead, directly regress bounding-box parameters from deep-features, following the *YOLO-v1* framework of Redmon et al. [2016b]. Notably, we further adapt the *YOLO-v1* architecture by replacing the fully-connected classification layers at the end with *fully*-convolutional ones, based on the *Fully Convolutional Networks* (FCNs) [Long et al., 2015a]. This has multiple advantages — (1) it dispenses with the dense parameters of fully-connected layers[2] with no drop in performance; (2) it enables application of the detector to images of arbitrary size during evaluation. The idea of fully-convolutional detection architectures was later adapted for general object detection by Dai et al. [2016] in their *Region Fully Convolutional Networks* work (R-FCN).

[2] fully-connected layer parameters typically constitute $\approx 90\%$ of parameters in popular ConvNet architectures, *e.g.* VGGNet [Simonyan and Zisserman, 2015].

Both the methods above [Gupta et al., 2016, Jaderberg et al., 2015b] propose boxes at the word level. However, due to varying size and spacing between characters, it is difficult to find the optimal segmentation between the words, without the top-down information from text recognition. Further, text-boxes have difficulty in capturing non-horizontal instances. Hence, more recent methods, most using variations on the pixel-wise labelling (FCN) framework of Long et al. [2015a], have focused on finding *lines* of text instead of boxes around words. He et al. [2016b] have a sequence of ConvNets to first get a rough estimate of the text-regions and then another one to predict the text centre-line and areas from cropped regions from the first stage; both the stages use the pixelwise FCN framework. Zhang et al. [2016] also use FCN to predict text-regions, next they extract character regions within the predicted text-regions using MSERs, and finally fit a line based on scale information and character centres obtained from a second FCN. Yao et al. [2016], produce two dense per-pixel maps in addition to text-regions — (1) individual character centres and, (2) the 2D direction of text-line at every location. The text-lines are inferred by first defining cliques of characters which lie in the same region, and then segmenting them into lines by inferring a minimum-spanning-tree and using a measure of the "straightness" of lines. Tian et al. [2016] obtain text-regions from pre-defined vertical anchors (as in Faster-RCNN [Ren et al., 2016]), by regressing scores and corrections to the anchors on the hidden state vector of Bidirectional-LSTMs [Graves and Schmidhuber, 2005] acting on conv5 features of VGG-19 [Simonyan and Zisserman, 2015]; they show that anchors help in capturing small text instances.

In recent years (2015-2018), there has been renewed interest in text localisation, mirroring the rapid advances in general visual object detection. He et al. [2017b] builds on Single-Shot Detector (SSD) [Liu et al., 2016], where in addition to word-level bounding-box regression, convolutional features from the intermediate layers are aggregated using a multi-scale *Inception* module [Szegedy et al., 2015], and are provided auxiliary dense (per-pixel) binary text/no-text supervision. Similarly, Shi et al. [2017] distribute text instances of different sizes to convolutional layers with the appropriate receptive fields (as

in SSD), and propose a *segment linking* mechanism to link the instances belonging to the same word together. Wu and Natarajan [2017] classify each pixel as belonging to text or not, with an additional class for pixels at the boundary (between two text instances), as in Ronneberger et al. [2015]; they later find word-level boxes using connected components. Similarly, Xue et al. [2018] further specialize the borders into four types, corresponding to the four sides of a bounding-box, and regress dense labels. Rong et al. [2017] apply the *DenseCap* [Johnson et al., 2016b] framework to scene-text, and learn recurrent embeddings for the surrounding context and text-instance descriptions.

Multi-oriented scene text detection. Axis aligned bounding-boxes employed in general object detections, have poor overlap with oriented (non-horizontal) text lines. Hence, a recent trend in text detection has been to learn more detailed localisation, and go beyond rectangular boxes. He et al. [2017c] regress offsets of *text-line quadrilateral* in a fully-convolutional manner, followed by non-maximal suppression on the predicted scores. Similarly, Liu and Jin [2017] predict general quadrilateral instead of rectangles, using non-axis aligned anchors. Zhou et al. [2017b] regress *rotated-rectangles* from multi-scale features pooled from the intermediate layers of a ConvNet. Lyu et al. [2018] adapt the recent *Mask-RCNN* framework [He et al., 2017a], and learn tight *masks* (contours) for text instances, instead of bounding boxes / quadrilaterals. Similarly, Long et al. [2018] also learn tight contours for text, but instead parametrise each instances as a sequence of ordered, overlapping disks centered at symmetric axes, each with its own radius and orientation. Wang et al. [2018] learn instance (word/line) specific transformations, where in addition to the bounding-box extents, and a confidence score, they regress parameters of a *Spatial Transformer* [Jaderberg et al., 2015d]; this corrects the perspective transformation and simplifies learning of the filters.

Weakly supervised detection. Although, our *synthetic dataset* (chap. 3) is commonly used for training text-detection methods, and provides ample training data for directly training deep ConvNets, yet learning from weakly-annotated, or unlabelled images could be useful for application in specialised domains (*e.g.* license plate localisation). Tian et al.

Author	Year	Venue	IIIT5K			SVT		IC03				IC13	IC15
		Lexicon→	50	1K	None	50	None	50	Full	50K	None	None	None
ABBYY			24.3	-	-	35.0	-	56.0	55.0	-	-	-	-
Wang et al.	2011	ICCV	-	-	-	57.0	-	76.0	62.0	-	-	-	-
Mishra et al.	2012	BMVC	64.1	57.5	-	73.2	-	81.8	67.8	-	-	-	-
Wang et al.	2012	ICPR	-	-	-	70.0	-	90.0	84.0	-	-	-	-
Novikova et al.	2012	ECCV	-	-	-	72.9	-	82.8	-	-	-	-	-
Goel et al.	2013	ICDAR	-	-	-	77.3	-	89.7	-	-	-	-	-
Bissacco et al.	2013	ICCV	-	-	-	90.4	78.0	-	-	-	-	87.6	-
Alsharif and Pineau	2014	ICLR	-	-	-	74.3	-	93.1	88.6	85.1	-	-	-
Almazán et al.	2014	PAMI	91.2	82.1	-	89.2	-	-	-	-	-	-	-
Yao et al.	2014	CVPR	80.2	69.3	-	75.9	-	88.5	80.3	-	-	-	-
Su and Lu	2014	ACCV	-	-	-	83.0	-	92.0	82.0	-	-	-	-
Jaderberg et al.	2014	ECCV	-	-	-	86.1	-	96.2	91.5	-	-	-	-
Jaderberg et al. (DICT)	2014	NIPS	-	-	-	95.4	80.7	98.7	98.6	93.3	-	-	-
(CHAR)		DLW	-	-	-	92.6	68.0	96.7	94.0	89.5	-	-	-
Rodriguez-Serrano et al.	2015	IJCV	76.1	57.4	-	70.0	-	-	-	-	-	-	-
Gordo	2015	CVPR	93.3	86.6	-	91.8	-	-	-	-	-	-	-
Jaderberg et al.	2015	ICLR	95.5	89.6	-	93.2	71.1	97.8	97	93.4	89.6	81.8	-
Shi et al.	2015	arXiv	97.6	94.4	78.2	96.4	80.8	98.7	97.6	95.5	89.4	86.7	-
Lee and Osindero (Full)	2016	CVPR	96.8	94.4	78.4	96.3	80.7	97.9	97.0	-	88.7	90.0	-
CNN only			-	-	-	-	78.9	-	-	-	-	88.5	-
CNN+RNN (no attn.)			-	-	-	-	79.1	-	-	-	-	88.9	-
Shi et al. (Full)	2016	CVPR	96.2	93.8	81.9	95.5	81.9	98.3	96.2	94.8	90.1	88.6	-
RNN (no rectification)			96.5	92.8	79.7	96.1	81.5	-	-	-	-	87.5	-
Poznanski and Wolf	2016	CVPR	97.9	94.2	-	96.6.0	83.6	-	-	-	-	-	-
Cheng et al.	2017	ICCV	99.3	97.5	87.4	97.1	85.9	99.2	97.3	-	94.2	93.3	85.3
Bai et al.	2018	CVPR	99.5	97.9	88.3	96.6	87.5	98.7	97.9	-	94.6	94.4	73.9
Cheng et al.	2018	CVPR	99.6	98.1	87.0	96.0	82.8	98.5	97.1	-	91.5	-	68.2

Table 2.1: Performance on cropped-word text recognition through the years, on several standard benchmarks — IIIT5K [Mishra et al., 2012a], Street View Text (SVT) [Wang and Belongie, 2010], ICDAR {2003, 2013, 2015} [ICDAR 2003 Robust Reading Competition, 2003, Karatzas et al., 2013, 2015]. Complete word recognition accuracy (all-or-nothing) is reported.

[2017] bootstrap starting from a "light" supervised character detector, trained on a small labelled dataset, and gradually expand their training data with high-confidence detections on unlabelled images. Hu et al. [2017] learn *character*-level detectors from *word*/*text-line* level localisation annotations, by exploiting the fact that characters have (approximately) uniform width.

2.4.2 Text recognition methods

Owing to sustained interest over the years in scene text recognition, distinct paradigms have emerged and evolved in scene text recognition. This has lead to a steady, and excellent improvements in text-recognition from cropped word-level images, especially in recent years. Table 2.1 summarises the progress over several standard scene text-recognition benchmark datasets. Below we review methods, organized into three categories, those

based on — (1) recognising characters, (2) word-level methods, and finally, (3) those based on the *encoder-decoder* framework [Cho et al., 2014, Sutskever et al., 2014].

Character level. Traditional *character-level* methods employ a three-step pipeline, consisting of segmentation into, or localisation of characters or character-regions, followed by character-level recognition, and finally using a language model for forming word-level strings. Characters are recognised either using sliding-window classifiers [Jaderberg et al., 2014b, Mishra et al., 2012a, Wang et al., 2011, 2012, Yao et al., 2014], or through under / over segmentation into parts [Alsharif and Pineau, 2013, Bissacco et al., 2013, Neumann and Matas, 2012] followed by grouping through classification. The characters are then strung together into words or sentences, using Viterbi-style energy minimisation over unary (character-level) and higher-order (bi-grams, n-grams or word-lexicons) language model terms [Alsharif and Pineau, 2013, Jaderberg et al., 2014b, Lee et al., 2014, Mishra et al., 2012a,b, Novikova et al., 2012, Shi et al., 2013, Wang and Belongie, 2010, Wang et al., 2011, 2012]. The classifiers employ hand-crafted image features like, histogram of oriented gradients (HOG [Dalal and Triggs, 2005]) [Bissacco et al., 2013, Mishra et al., 2012a, Wang et al., 2011], character-level image similarity (based on colour and font) [Novikova et al., 2012], random ferns ([Ozuysal et al., 2010]) [Wang et al., 2011] or SIFT ([Lowe, 2004]) [Gordo, 2015], and more recently, deep-features [Goodfellow et al., 2014b, Jaderberg et al., 2014b, Wang et al., 2012].

Word level. Another set of methods, unify the traditional three-step character-based methods into an end-to-end system, processing a whole word image. Early works model word recognition as retrieval or matching in a collection of word images from a fixed lexicon [Almazán et al., 2014, Goel et al., 2013, Gordo, 2015, Rodriguez-Serrano et al., 2015]. Jaderberg et al. [2014a] also operating on word-images, propose a large synthetic dataset for learning strong CNN based classifiers at the character, n-gram, and word levels. Jaderberg et al. [2015a] combine the the character and n-gram models into a single end-to-end structured prediction framework. More recently Poznanski and Wolf [2016] extended the n-gram classifier of Jaderberg et al. [2014a] with spatial histograms (PHOC

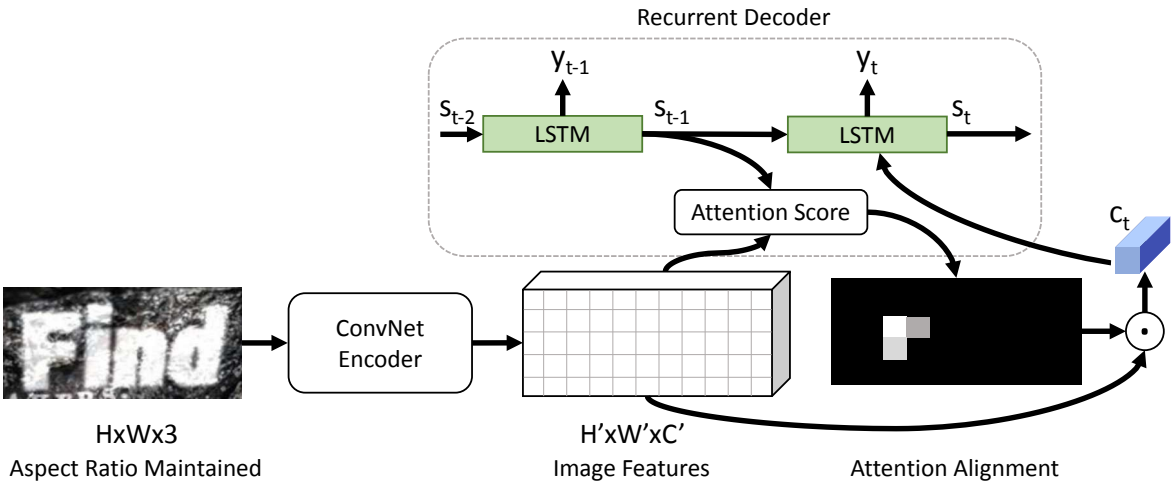


Figure 2.4: Encoder-decoder text recognition architecture. Encoder-decoder recognition architectures encode the image features using convolutional networks, producing either a compact fully-connected vector (such that $H' = W' = 1$), or maintain the spatial extents using fully-convolutional layers. The decoder is a recurrent network (commonly *LSTM* [Hochreiter and Schmidhuber, 1997]), and optionally employs the soft-attention mechanism of Bahdanau et al. [2015] to modulate the attention over the encoded image features. The characters are decoded sequentially from the pooled context c_t.

attributes [Almazán et al., 2014]), providing stronger supervision, resulting in significant improvements.

Encoder-decoder models. More recent methods cast the text-recognition problem as one of sequence prediction in the *encoder-decoder* framework [Cho et al., 2014, Sutskever et al., 2014]. Recurrent (commonly LSTMs [Hochreiter and Schmidhuber, 1997], or Bidirectional-LSTMs [Graves and Schmidhuber, 2005, Schuster and Paliwal, 1997]) encodings of image are obtained to capture long-term dependencies. These are then decoded into output characters sequentially using a recurrent unit, which models the conditional probability of the next character given the decoding so far; fig. 2.4 visualises the architecture. Su and Lu [2014] adopted this framework first, using HOG features with *Connectionist Temporal Classification* (CTC) [Graves et al., 2006] to align the predicted characters with the image features. He et al. [2016a], Shi et al. [2015] replace HOG features with stronger CNN features.

Lee and Osindero [2016] augment the recurrent decoder with *soft-attention* [Bahdanau et al., 2015], to dynamically modulate the image features for decoding each character; they

encode the image using blocks of *recursive* convolutional layers with shared parameters. Similarly, Shi et al. [2016] also employ a similar recurrent attention decoder, but first rectify the input image using a *Spatial Transformer* [Jaderberg et al., 2015c]. Cheng et al. [2017] identify *attention drift*, *i.e.* misalignment of the soft-attention, as one of the causes for incorrect recognition, and enforce correct alignment using explicit character-level location supervision. They show that using a small amount of character-level labelled examples (1%) gives significant improvements. Bai et al. [2018] also tackle the misalignment in attention, and learn to model the *edit probability* (EP), i.e., the probability of generating a string by inserting, deleting or substituting characters (as in Levenshtein distance [Levenshtein, 1966]), conditioned on the image. They achieve this by augmenting the recurrent decoder to estimate the probabilities for the respective edit-operations, in addition to character probabilities. Cheng et al. [2018] extend the conventional bi-directional encoding (left-right, right-left) in the other two directions as well (top-bottom, bottom-top), which is reminiscent of the multi-dimensional LSTMs for handwriting recognition [Graves and Schmidhuber, 2009]. Liu et al. [2018] drive the image-features learnt for "distorted" synthetic images, to be close to the features from the corresponding "clean" images (with no distortions/background noise), and show superior performance.

2.4.3 End-to-end text spotting

A recent trend in text spotting in natural images has been towards learning single, unified networks which jointly localise, as well as recognise the text instances. This has been enabled by progress in general object detection frameworks [Girshick, 2015a, Ren et al., 2015], growing GPU memory, and also our full scene-level synthetic text dataset (chap. 3), which is used in all of the following works. Such models replace the complex multi-stage pipelines for text spotting [Gupta et al., 2016, Jaderberg et al., 2015b], minimizing the downstream compounding of errors.

Bušta et al. [2017] adapt the *Region Proposal Network* [Ren et al., 2015] and use bilinear-sampling to extract the text-windows (as in *DenseCap* [Johnson et al., 2016a]), and decode

using a recurrent decoder trained with CTC. The concurrent work of Li et al. [2017], similarly build on the *Faster-RCNN* [Ren et al., 2015] framework, but modifies the *RoI-pooling* to be sensitive to the width (as opposed to pooling to a fixed width), to accommodate text instances of varying lengths. He et al. [2018] extend the instances to be multi-oriented quadrilaterals, and employ the character-level attention decoders [Cheng et al., 2017] for recognition. Gómez et al. [2018] cast the end-to-end framework in the retrieval framework, and learnt to regress *Pyramidal Histogram of Characters* (PHOC) [Almazán et al., 2014] embeddings from a *YOLO*-like detector [Redmon et al., 2016a]. In chapter 6, we explore joint localisation and recognition of text-lines in images of text-blocks (multiple lines/paragraph of text), in the context of improving generalisation for recurrent networks to sequence lengths beyond those they are trained on. We achieve this by constraining the state of the recurrent module to be an interpretable spatial map of the lines that have been recognised so far, and training the model for one-step "inductive" (one-line at a time) updates.

3

Synthetic Data for Text Localisation in Natural Images

This work was presented as a *spotlight* presentation at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

Synthetic Data for Text Localisation in Natural Images

Ankush Gupta Andrea Vedaldi Andrew Zisserman
Dept. of Engineering Science, University of Oxford
{ankush,vedaldi,az}@robots.ox.ac.uk

Abstract

In this paper we introduce a new method for text detection in natural images. The method comprises two contributions: First, a fast and scalable engine to generate synthetic images of text in clutter. This engine overlays synthetic text to existing background images in a natural way, accounting for the local 3D scene geometry. Second, we use the synthetic images to train a Fully-Convolutional Regression Network (FCRN) which efficiently performs text detection and bounding-box regression at all locations and multiple scales in an image. We discuss the relation of FCRN to the recently-introduced YOLO detector, as well as other end-to-end object detection systems based on deep learning. The resulting detection network significantly out performs current methods for text detection in natural images, achieving an F-measure of 84.2% on the standard ICDAR 2013 benchmark. Furthermore, it can process 15 images per second on a GPU.

1. Introduction

Text spotting, namely the ability to read text in natural scenes, is a highly-desirable feature in anthropocentric applications of computer vision. State-of-the-art systems such as [20] achieved their high text spotting performance by combining two simple but powerful insights. The first is that complex recognition pipelines that recognise text by explicitly combining recognition and detection of *individual* characters can be replaced by very powerful classifiers that directly map an image patch to words [13, 20]. The second is that these powerful classifiers can be learned by generating the required training data synthetically [19, 44].

While [20] successfully addressed the problem of recognising text *given an image patch containing a word*, the process of obtaining these patches remains suboptimal. The pipeline combines general purpose features such as HoG [6], EdgeBoxes [48] and Aggregate Channel Features [7] and brings in text specific (CNN) features only in the later stages, where patches are finally recognised as specific words. This state of affair is highly undesirable for two reasons. First, the performance of the detection pipeline becomes the new bottleneck of text spotting: in [20] recognition accuracy for correctly cropped words is 98% whereas the end-to-end text spotting F-score is only 69% mainly due to incorrect and missed word region proposals. Second, the pipeline is slow and inelegant.

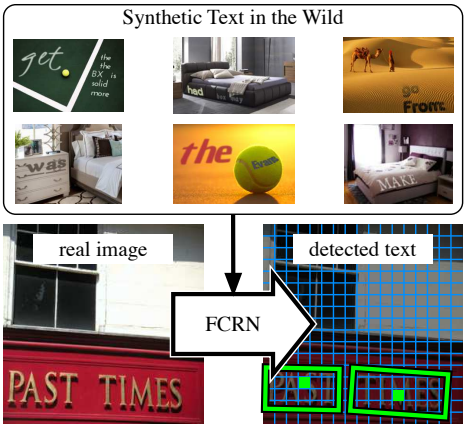


Figure 1. We propose a Fully-Convolutional Regression Network (FCRN) for high-performance text recognition in natural scenes (bottom) which detects text up to 45× faster than the current state-of-the-art text detectors and with better accuracy. FCRN is trained without any manual annotation using a new dataset of synthetic text in the wild. The latter is obtained by automatically adding text to natural scenes in a manner compatible with the scene geometry (top).

In this paper we propose improvements similar to [20] to the complementary problem of *text detection*. We make two key contributions. First, we propose a new method for generating synthetic images of text that *naturally blends text in existing natural scenes*, using off-the-shelf deep learning and segmentation techniques to align text to the geometry of a background image and respect scene boundaries. We use this method to automatically generate a new **synthetic dataset of text in cluttered conditions** (figure 1 (top) and section 2). This dataset, called *SynthText in the Wild* (figure 2), is suitable for training high-performance scene text detectors. The key difference with existing synthetic text datasets such as the one of [20] is that these only contains

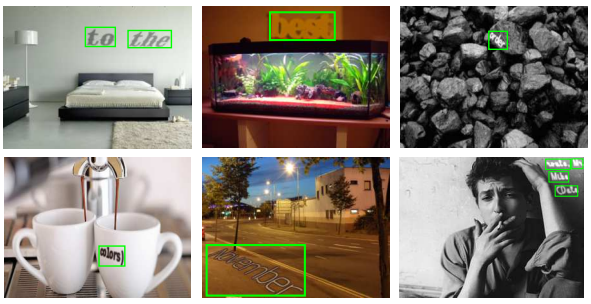


Figure 2. Sample images from our synthetically generated scene-text dataset. Ground-truth word-level axis-aligned bounding boxes are shown.

Dataset	# Images		# Words	
	Train	Test	Train	Test
ICDAR {11,13,15}	229	255	849	1095
SVT	100	249	257	647

Table 1. Size of publicly available text localisation datasets — ICDAR [23, 24, 39], the Street View Text (SVT) dataset [43]. Word numbers for the entry "ICDAR{11,13,15}" are from the ICDAR15 Robust Reading Competition's Focused Scene Text Localisation dataset.

word-level image regions and are unsuitable for training detectors.

The second contribution is a **text detection deep architecture** which is both accurate and efficient (figure 1 (bottom) and section 3). We call this a *fully-convolutional regression network*. Similar to models such as the Fully-Convolutional Networks (FCN) for image segmentation, it performs prediction densely, at every image location. However, differently from FCN, the prediction is not just a class label (text/not text), but the parameters of a bounding box enclosing the word centred at that location. The latter idea is borrowed from the You Only Look Once (YOLO) technique of Redmon *et al*. [36], but with convolutional regressors with a significant boost to performance.

The new data and detector achieve state-of-the-art text detection performance on standard benchmark datasets (section 4) while being an order of magnitude faster than traditional text detectors at test time (up to 15 images per second on a GPU). We also demonstrate the importance of verisimilitude in the dataset by showing that if the detector is trained on images with words inserted synthetically that do *not* take account of the scene layout, then the detection performance is substantially inferior. Finally, due to the more accurate detection step, end-to-end word recognition is also improved once the new detector is swapped in for existing ones in state-of-the-art pipelines. Our findings are summarised in section 5.

1.1. Related Work

Object Detection with CNNs. Our text detection network draws primarily on Long *et al*.'s Fully-Convolutional network [31] and Redmon *et al*.'s YOLO image-grid based bounding-box regression network [36]. YOLO is part of a broad line of work on using CNN features for object category detection dating back to Girshick *et al*.'s Region-CNN (R-CNN) framework [12] combination of region proposals and CNN features. The R-CNN framework has three broad stages — (1) generating object proposals, (2) extracting CNN feature maps for each proposal, and (3) filtering the proposals through class specific SVMs. Jaderberg *et al*.'s text spotting method also uses a similar pipeline for detection [20]. Extracting feature maps for each region *independently* was identified as the bottleneck by Girshick *et al*. in Fast R-CNN [11]. They obtain 100× speed-up over R-CNN by computing the CNN features once and pooling them locally for each proposal; they also streamline the last two stages of R-CNN into a single multi-task learning problem. This work exposed the region-proposal stage as the new bottleneck. Lenc *et al*. [29] drop the region proposal stage altogether and use a constant set of regions learnt through K-means clustering on the PASCAL VOC data. Ren *et al*. [37] also start from a fixed set of proposal, but refined them prior to detection by using a Region Proposal Network which shares weights with the later detection network and streamlines the multi-stage R-CNN framework.

Synthetic Data. Synthetic datasets provide detailed ground-truth annotations, and are cheap and scalable alternatives to annotating images manually. They have been widely used to learn large CNN models — Wang *et al*. [44] and Jaderberg *et al*. [19] use synthetic text images to train word-image recognition networks; Dosovitskiy *et al*. [9] use floating chair renderings to train dense optical flow regression networks. Detailed synthetic data has also been used to learn generative models — Dosovitskiy *et al*. [8] train inverted CNN models to render images of chairs, while Yildirim *et al*. [46] use deep CNN features trained on synthetic face renderings to regress pose parameters from face images.

Augmenting Single Images. There is a large body of work on inserting objects photo-realistically, and inferring 3D structure from single images — Karsch *et al*. [25] develop an impressive semi-automatic method to render objects with correct lighting and perspective; they infer the actual size of objects based on the technique of Criminisi *et al*. [5]. Hoiem *et al*. [15] categorise image regions into ground-plane, vertical plane or sky from a single image and use it to generate "pop-ups" by decomposing the image into planes [14]. Similarly, we too decompose a single image into local planar regions, but use instead the dense depth prediction of Liu *et al*. [30].

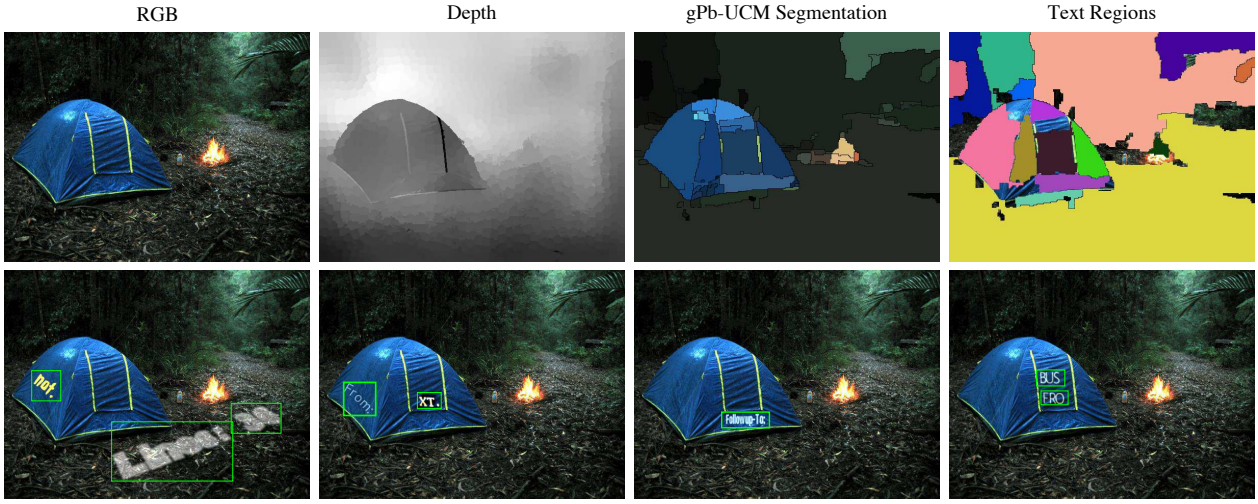


Figure 3. (Top, left to right): (1) RGB input image with no text instance. (2) Predicted dense depth map (darker regions are closer). (3) Colour and texture gPb-UCM segments. (4) Filtered regions: regions suitable for text are coloured randomly; those unsuitable retain their original image pixels. (Bottom): Four synthetic scene-text images with axis-aligned bounding-box annotations at the word level.

2. Synthetic Text in the Wild

Supervised training of large models such as deep CNNs, which contain millions of parameters, requires a very significant amount of labelled training data [26], which is expensive to obtain manually. Furthermore, as summarised in Table 1, publicly available text spotting or detection datasets are quite small. Such datasets are not only insufficient to train large CNN models, but also inadequate to represent the space of possible text variations in natural scenes — fonts, colours, sizes, positions. Hence, in this section we develop a synthetic text-scene image generation engine for building a large annotated dataset for text localisation.

Our synthetic engine (1) produces **realistic** scene-text images so that the trained models can generalise to real (non-synthetic) images, (2) is fully **automated** and, is (3) **fast**, which enables the generation of large quantities of data without supervision. The text generation pipeline can be summarised as follows (see also Figure 3). After acquiring suitable text and image samples (section 2.1), the image is segmented into contiguous regions based on local colour and texture cues [2], and a dense pixel-wise depth map is obtained using the CNN of [30] (section 2.2). Then, for each contiguous region a local surface normal is estimated.

Next, a colour for text and, optionally, for its outline is chosen based on the region's colour (section 2.3). Finally, a text sample is rendered using a randomly selected font and transformed according to the local surface orientation; the text is blended into the scene using Poisson image editing [35]. Our engine takes about half a second to generate a new scene-text image.

This method is used to generate 800,000 scene-text images, each with multiple instances of words rendered in different styles as seen in Figure 2. The dataset is available at: http://www.robots.ox.ac.uk/~vgg/data/scenetext

2.1. Text and Image Sources

The synthetic text generation process starts by sampling some text and a background image. The text is extracted from the Newsgroup20 dataset [27] in three ways — words, lines (up to 3 lines) and paragraphs (up to 7 lines). Words are defined as tokens separated by whitespace characters, lines are delimited by the newline character. This is a rich dataset, with a natural distribution of English text interspersed with symbols, punctuation marks, nouns and numbers.

To favour variety, 8,000 background images are extracted from Google Image Search through queries related to different objects/scenes and indoor/outdoor and natural/artificial locales. To guarantee that all text occurrences are fully annotated, these images *must not contain text of their own* (a limitation of the Street View Text [43] is that annotations are not exhaustive). Hence, keywords which would recall a large amount of text in the images (*e.g.* "street-sign", "menu" etc.) are avoided; images containing text are discarded through manual inspection.

2.2. Segmentation and Geometry Estimation

In real images, text tends to be contained in well defined regions (e.g. a sign). We approximate this constraint by requiring text to be contained in regions characterised by a uniform colour and texture. This also prevents text from crossing strong image discontinuities, which is unlikely to

Figure 4. Local colour/texture sensitive placement. (Left) Example image from the Synthetic text dataset. Notice that the text is restricted within the boundaries of the step in the street. (Right) For comparison, the placement of text in this image does not respect the local region cues.

occur in practice. Regions are obtained by thresholding the gPb-UCM contour hierarchies [2] at 0.11 using the efficient graph-cut implementation of [3]. Figure 4 shows an example of text respecting local region cues.

In natural images, text tends to be painted on top of surfaces (e.g. a sign or a cup). In order to approximate a similar effect in our synthetic data, the text is perspectively transformed according to local surface normals. The normals are estimated automatically by first predicting a dense depth map using the CNN of [30] for the regions segmented above, and then fitting a planar facet to it using RANSAC [10].

Text is aligned to the estimated region orientations as follows: first, the image region contour is warped to a frontal-parallel view using the estimated plane normal; then, a rectangle is fitted to the fronto-parallel region; finally, the text is aligned to the larger side ("width") of this rectangle. When placing multiple instances of text in the same region, text masks are checked for collision against each other to avoid placing them on top of each other.

Not all segmentation regions are suitable for text placement — regions should not be too small, have an extreme aspect ratio, or have surface normal orthogonal to the viewing direction; all such regions are filtered in this stage. Further, regions with too much texture are also filtered, where the degree of texture is measured by the strength of third derivatives in the RGB image.

Discussion. An alternative to using a CNN to estimate depth, which is an error prone process, is to use a dataset of RGBD images. We prefer to estimate an imperfect depth map instead because: (1) it allows essentially any scene type background image to be used, instead of only the ones for which RGBD data are available, and (2) because publicly available RGBD datasets such as NYUDv2 [40], B3DO [22], Sintel [4], and Make3D [38] have several limitations in our context: small size (1,500 images in NYUDv21, 400 frames in Make3D, and a small number of videos in B3DO and Sintel), low-resolution and motion blur, restriction to indoor images (in NYUDv2 and B3DO), and limited variability in the images for video-based datasets (B3DO and Sintel).

2.3. Text Rendering and Image Composition

Once the location and orientation of text has been decided, text is assigned a colour. The colour palette for text is learned from cropped word images in the IIIT5K word dataset [32]. Pixels in each cropped word images are partitioned into two sets using K-means, resulting in a colour pair, with one colour approximating the foreground (text) colour and the other the background. When rendering new text, the colour pair whose background colour matches the target image region the best (using L2-norm in the Lab colour space) is selected, and the corresponding foreground colour is used to render the text.

About 20% of the text instances are randomly chosen to have a border. The border colour is chosen to be either the same as foreground colour with its value channel increased or decreased, or is chosen to be the mean of the foreground and background colours.

To maintain the illumination gradient in the synthetic text image, we blend the text on to the base image using Poisson image editing [35], with the guidance field defined as in their equation (12). We solve this efficiently using the implementation provided by Raskar[1]

3. A Fast Text Detection Network

In this section we introduce our CNN architecture for text detection in natural scenes. While existing text detection pipelines combine several ad-hoc steps and are slow, we propose a detector which is highly accurate, fast, and trainable end-to-end.

Let $\mathbf{x}$ denote an image. The most common approach for CNN-based detection is to propose a number of image regions R that may contain the target object (text in our case), crop the image, and use a CNN $c = \phi(\text{crop}_R(\mathbf{x})) \in \{0, 1\}$ to score them as correct or not. This approach, which has been popularised by R-CNN [12], works well but is slow as it entails evaluating the CNN thousands of times per image.

An alternative and much faster strategy for object detection is to construct a fixed field of predictors $(c, \mathbf{p}) = \phi_{uv}(\mathbf{x})$, each of which specialises in predicting the presence $c \in \mathbb{R}$ and pose $\mathbf{p} = (x-u, y-v, w, h)$ of an object around a specific image location (u, v). Here the pose parameters (x, y) and (w, h) denote respectively the location and size of a bounding box tightly enclosing the object. Each predictor ϕ_{uv} is tasked with predicting objects which occur in some ball $(x, y) \in B_\rho(u, v)$ of the predictor location.

While this construction may sound abstract, it is actually a common one, implemented for example by Implicit Shape Models (ISM) [28] and Hough voting [16]. There a predictor ϕ_{uv} looks at a local image patch, centred at (u, v), and

[1]Fast Poisson image editing code available at: http://web.media.mit.edu/~raskar/photo/code.pdf based on Discrete Sine Transform.

tries to predict whether there is an object around (u, v), and where the object is located relative to it.

In this paper we propose an extreme variant of Hough voting, inspired by the Fully-Convolutional Network (FCN) of Long *et al*. [31] and the You Only Look Once (YOLO) technique of Redmon *et al*. [36]. In ISM and Hough voting, individual predictions are aggregated across the image, in a voting scheme. YOLO is similar, but avoids voting and uses individual predictions directly; since this idea can accelerate detection, we adopt it here.

The other key conceptual difference between YOLO and Hough voting is that in Hough voting predictors $\phi_{uv}(\mathbf{x})$ are local and translation invariant, whereas in YOLO they are not: First, in YOLO each predictor is allowed to pool evidence from the whole image, not just an image patch centred at (u, v). Second, in YOLO predictors at different locations $(u, v) \neq (u', v')$ are different functions $\phi_{uv} \neq \phi_{u'v'}$ learned independently.

While YOLO's approach allows the method to pick up contextual information useful in detection of PASCAL or ImageNet objects, we found this unsuitable for smaller and more variable text occurrences. Instead, we propose here a method which is in between YOLO and Hough voting. As in YOLO, each detector $\phi_{uv}(\mathbf{x})$ still predicts directly object occurrences, without undergoing an expensive voting accumulation process; however, as in Hough voting, detectors $\phi_{uv}(\mathbf{x})$ are local and translation invariant, sharing parameters. We implement this field of translation-invariant and local predictors as the output of the last layer of a deep CNN, obtaining a *fully-convolutional regression network* (FCRN).

3.1. Architecture

This section describes the structure of the FCRN. First, we describe the first several layers of the architecture, which compute text-specific image features. Then, we describe the dense regression network built on top of these features and finally its application at multiple scales.

Single-scale features. Our architecture is inspired by VGG-16 [41], using several layers of small dense filters; however, we found that a much smaller model works just as well and more efficiently for text. The architecture comprises nine convolutional layers, each followed by the Rectified Linear Unit non-linearity, and, occasionally, by a max-pooling layer. All linear filters have a stride of 1 sample, and preserve the resolution of feature maps through zero padding. Max-pooling is performed over 2×2 windows with a stride of 2 samples, therefore halving the feature maps resolution. [2]

Class and bounding box prediction. The single-scale features terminate with a dense feature field. Given that there are four downsampling max-pooling layers, the stride of these features is $\Delta = 16$ pixels, each containing 512 feature channels $\phi^f_{uv}(\mathbf{x})$ (we express uv in pixels for convenience).

Given the features $\phi^f_{uv}(\mathbf{x})$, we can now discuss the construction of the dense text predictors $\phi_{uv}(\mathbf{x}) = \phi^r_{uv} \circ \phi^f(\mathbf{x})$. These predictors are implemented as a further seven 5×5 linear filters (C-7-5×5) ϕ^r_{uv}, each regressing one of seven numbers: the object presence confidence c, and up to six object pose parameters $\mathbf{p} = (x-u, y-v, w, h, \cos\theta, \sin\theta)$ where x, y, w, h have been discussed before and θ is the bounding box rotation.

Hence, for an input image of size $H \times W$, we obtain a grid of $\frac{H}{\Delta} \times \frac{W}{\Delta}$ predictions, one each for an image cell of size $\Delta \times \Delta$ pixels. Each predictor is responsible for detecting a word if the word centre falls within the corresponding cell.[3] YOLO is similar but operates at about half this resolution; a denser predictor sampling is important to reduce collisions (multiple words falling in the same cell) and therefore to increase recall (since at most one word can be detected per cell). In practice, for a 224×224 image, we obtain 14×14 cells/predictors

Multi-scale detection. The limited receptive field of our convolutional filters prohibits detection of large text instances. Hence, we get the detections at multiple downscaled versions of the input image and merge them through non-maximal suppression. In more detail, the input image is scaled down by factors $\{1, 1/2, 1/4, 1/8\}$ (scaling up is an overkill as the baseline features are already computed very densely). Then, the resulting detections are combined by suppressing those with a lower score than the score of an overlapping detection.

Training loss. We use a squared loss term for each of the $\frac{H}{\Delta} \times \frac{W}{\Delta} \times 7$ outputs of the CNN as in YOLO [36]. If a cell does not contain a ground-truth word, the loss ignores all parameters but c (text/no-text).

Comparison with YOLO. Our fully-convolutional regression network (FCRN) has 30× less parameters than the YOLO network (which has ∼90% of the parameters in the last two fully-connected layers). Due to its global nature, standard YOLO must be retrained for each image size, including multiple scales, further increasing the model size (while our model requires 44MB, YOLO would require 2GB). This makes YOLO not only harder to train, but also less efficient (2× slower that FCRN).

4. Evaluation

First, in section 4.1 we describe the text datasets on which we evaluate our model. Next, we evaluate our model on the text localisation task in section 4.2. In section 4.3, to investigate which components of the synthetic data generation pipeline are important, we perform detailed ablation

[2] The sequence of layers is as follows: 64 5×5 convolutional filters + ReLU (CR-64-5×5), max pooling (MP), CR-128-5×5, MP, CR128-3×3, CR-128-3×3-conv, MP, CR-256-3×3, CR-256-3×3, MP, CR-512-3×3, CR-512-3×3, CR-512-5×5.

[3] For regression, it was found beneficial to normalise the pose parameters as follows: $\bar{\mathbf{p}} = ((x-u)/\Delta, (y-v)/\Delta, w/W, h/H, \cos\theta, \sin\theta)$.

	PASCAL Eval												DetEval								
	IC11				IC13				SVT				IC11			IC13			SVT		
	F	P	R	R_M	F	P	R	R_M	F	P	R	R_M	F	P	R	F	P	R	F	P	R
Huang [17]	-	-	-	-	-	-	-	-	-	-	-	-	78	88	71	-	-	-	-	-	-
Jaderberg [20]	77.2	87.5	69.2	70.6	76.2	86.7	68.0	69.3	53.6	62.8	46.8	55.4	76.8	88.2	68.0	76.8	88.5	67.8	24.7	27.7	22.3
Jaderberg (trained on SynthText)	77.3	89.2	68.4	72.3	76.7	88.9	67.5	71.4	53.6	58.9	49.1	56.1	75.5	87.5	66.4	75.5	87.9	66.3	24.7	27.8	22.3
Neumann [33]	-	-	-	-	-	-	-	-	-	-	-	-	68.7	73.1	64.7	-	-	-	-	-	-
Neumann [34]	-	-	-	-	-	-	-	-	-	-	-	-	72.3	79.3	66.4	-	-	-	-	-	-
Zhang [47]	-	-	-	-	-	-	-	-	-	-	-	-	80	84	**76**	80	88	74	-	-	-
FCRN single-scale	60.6	78.8	49.2	49.2	61.0	77.7	48.9	48.9	45.6	50.9	41.2	41.2	64.5	81.9	53.2	64.3	81.3	53.1	31.4	**34.5**	28.9
FCRN multi-scale	70.0	78.4	63.2	64.6	69.5	78.1	62.6	67.0	46.2	47.0	45.4	53.0	73.0	77.9	68.9	73.4	80.3	67.7	**34.5**	29.9	**40.7**
FCRN + multi-filt	78.7	**95.3**	67.0	67.5	78.0	**94.8**	66.3	66.7	56.3	61.5	51.9	54.1	78.0	**94.5**	66.4	78.0	**94.8**	66.3	25.5	26.8	24.3
FCRNall + multi-filt	**84.7**	94.3	**76.9**	**79.6**	**84.2**	93.8	**76.4**	**79.6**	**62.4**	**65.1**	**59.9**	**75.0**	**82.3**	91.5	74.8	**83.0**	92.0	**75.5**	26.7	26.2	27.4

Table 2. Comparison with previous methods on text localisation. Precision (P) and Recall (R) at maximum F-measure (F) and the maximum recall (R_M) are reported.

experiments. In section 4.4, we use the results from our localisation model for end-to-end text spotting. We show substantial improvements over the state-of-the-art in both text localisation and end-to-end text spotting. Finally, in section 4.5 we discuss the speed-up gained by using our models for text localisation.

4.1. Datasets

We evaluate our text detection networks on standard benchmarks: *ICDAR* 2011, 2013 datasets [24, 39] and the Street View Text dataset [43]. These datasets are reviewed next and their statistics are given in Table 1.

SynthText in the Wild. This is a dataset of 800,000 training images generated using our synthetic engine from section 2. Each image has about ten word instances annotated with character and word-level bounding-boxes.

ICDAR Datasets. The *ICDAR* datasets (IC011, IC013) are obtained from the Robust Reading Challenges held in 2011 and 2013 respectively. They contain real world images of text on sign boards, books, posters and other objects with world-level axis-aligned bounding box annotations. The datasets largely contain the same images, but shuffle the test and training splits. We do not evaluate on the more recent *ICDAR* 2015 dataset as it is almost identical to the 2013 dataset.

Street View Text. This dataset, abbreviated *SVT*, consists of images harvested from Google Street View annotated with word-level axis-aligned bounding boxes. *SVT* is more challenging than the *ICDAR* data as it contains smaller and lower resolution text. Furthermore, not all instances of text are annotated. In practice, this means that precision is heavily underestimated in evaluation. Lexicons consisting of 50 distractor words along with the ground-truth words are provided for each image; we refer to testing on *SVT* with these lexicons as *SVT-50*.

4.2. Text Localisation Experiments

We evaluate our detection networks to — (1) compare the performance when applied to single-scale and multiple down-scaled versions of the image and, (2) improve upon the state-of-the-art results in text detection when used as high-quality proposals.

Training. FCRN is trained on 800,000 images from our *SynthText in the Wild* dataset. Each image is resized to a size of 512×512 pixels. We optimise using SGD with momentum and batch-normalisation [18] after every convolutional layer (except the last one). We use mini-batches of 16 images each, set the momentum to 0.9, and use a weight-decay of 5^{-4}. The learning rate is set to 10^{-4} initially and is reduced to 10^{-5} when the training loss plateaus.

As only a small number (1-2%) of grid-cells contain text, we weigh down the non-text probability error terms initially by multiplying with 0.01; this weight is gradually increased to 1 as the training progresses. Due to class imbalance, all the probability scores collapse to zero if such a weighting scheme is not used.

Inference. We get the class probabilities and bounding-box predictions from our FCRN model. The predictions are filtered by thresholding the class probabilities (at a threshold t). Finally, multiple detections from nearby cells are suppressed using non-maximal suppression, whereby amongst two overlapping detections the one with the lower probability is suppressed. In the following we first give results for a conservative threshold of $t = 0.3$, for higher precision, and then relax this to $t = 0.0$ (i.e., all proposals accepted) for higher recall.

Evaluation protocol. We report text detection performance using two protocols commonly used in the literature — (1) *DetEval* [45] popularly used in ICDAR competitions for evaluating localisation methods, and (2) PASCAL VOC style intersection-over-union overlap method (≥ 0.5 IoU for a positive detection).

Single & multi-scale detection. The "FCRN single-scale"

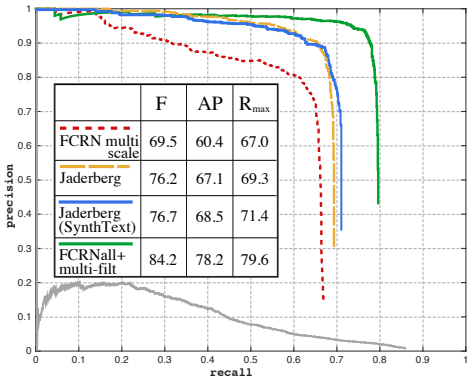


Figure 5. Precision-Recall curves for various text detection methods on IC13. The methods are: (1) multi-scale application of FCRN ("FCRN-multi"); (2) The original curve of Jaderberg *et al*. [20]; (3) Jaderberg *et al*. [20] retrained on the *SynthText in the Wild* dataset; and, (4) "FCRNall + multi-filt" methods. Maximum F-score (F), Average Precision (AP) and maximum Recall (R_{max}) are also given. The gray curve at the bottom is of multi-scale detections from our FCRN network (max. recall = 85.9%), which is fed into the multi-filtering post-processing to get the refined "FCRNall + multi-filt" detections.

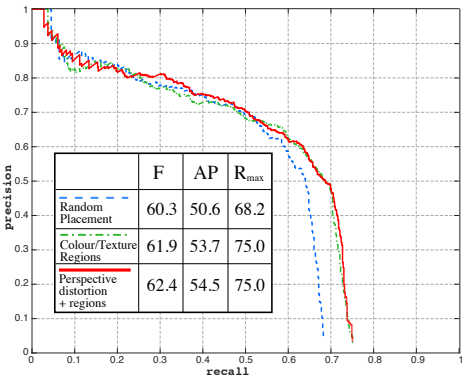


Figure 6. Precision-Recall curves text localisation on the SVT dataset using the model "FCRNall+multi-filt" when trained on increasingly sophisticated training sets (section 4.3).

entry in Table 2 shows the performance of our FCRN model on the test datasets. The precision at maximum F-measure of single-scale FCRN is comparable to the methods of Neuman *et al*. [33, 34], while the recall is significantly worse by 12%.

The "FCRN multi-scale" entry in Table 2 shows performance on multi-scale application of our network. This method improves maximum recall by more than 12% over the single-scale method and outperforms the methods of Neumann *et al*.

Post-processing proposals. Current end-to-end text spotting (detection and recognition) methods [1, 20, 44] boost performance by combining detection with text recognition. To further improve FCRN detections, we use the multi-scale detections from FCRN as proposals and refine them by using the post-processing stages of Jaderberg *et al*. [20]. There are three stages: first filtering using a binary text/no-text random-forest classifier; second, regressing an improved bounding-box using a CNN; and third recognition based NMS where the word images are recognised using a large fixed lexicon based CNN, and the detections are merged through non-maximal suppression based on word identities. Details are given in [20]. We use code provided by the authors for fair comparison.

We test this in two modes — (1) *low-recall*: where only high-scoring (probability > 0.3) multi-scale FCRN detections are used (the threshold previously used in the single- and multi-scale inference). This typically yields less than 30 proposals. And, (2) *high-recall*: where *all* the multi-scale FCRN detections (typically about a thousand in number) are used. Performance of these methods on text detection are shown by the entries named "FCRN + multi-filt" and "FCRNall + multi-filt" respectively in Table 2. Note that the *low-recall* method achieves better than the state-of-the-art performance on text detection, whereas *high-recall* method significantly improves the state-of-the-art with an improvement of 6% in the F-measure for all the datasets.

Figure 5 shows the Precision-Recall curves for text detection on the *IC13* dataset. Note the high recall (85.9%) of the multi-scale detections output from FCRN before refinement using the multi-filtering post-processing. Also, note the drastic increase in maximum recall ($+10.3\%$) and in Average Precision ($+11.1\%$) for "FCRNall + multi-filt" as compared to Jaderberg *et al*.

Further, to establish that the improvement in text detection is due to the new detection model, and not merely due to the large size of our synthetic dataset, we trained Jaderberg *et al*.'s method on our *SynthText in the Wild* dataset – in particular, the ACF component of their region proposal stage.[4] Figure 5 and Table 2 show that, even with $10\times$ more (synthetic) training data, Jaderberg *et al*.'s model improves only marginally (+0.8% in AP, +2.1% in maximum recall).

A common failure mode is text in unusual fonts which are not present in the training set. The detector is also confused by symbols or patterns of constant stroke width which look like text, for example road-signs, stick figures etc. Since the detector does not scale the image up, extremely small sized text instances are not detected. Finally, words get broken into multiple instances or merged into one instance due to large or small spacing between the characters.

4.3. Synthetic Dataset Evaluation

We investigate the contribution that the various stages of the synthetic text-scene data generation pipeline bring to

[4] Their other region proposal method, EdgeBoxes, was not re-trained; as it is learnt from low-level edge features from the Berkeley Segmentation Dataset, which is not text specific.

Model	IC11	IC11*	IC13	SVT	SVT-50
Wang [42]	-	-	-	-	38
Wang & Wu [44]	-	-	-	-	46
Alsharif [1]	-	-	-	-	48
Neumann [34]	-	45.2	-	-	-
Jaderberg [21]	-	-	-	-	56
Jaderberg [20]	76	69	76	53	**76**
FCRN + multi-filt	80.5 (77.8)	75.8 (73.5)	80.3 (77.8)	54.7	68.0
FCRNall + multi-filt	**84.3** (81.2)	**81.0** (78.4)	**84.7** (81.8)	**55.7**	67.7

Table 3. Comparison with previous methods on end-to-end text spotting. Maximum F-measure% is reported. IC11* is evaluated according to the protocol described in [34]. Numbers in parenthesis are obtained if words containing non-alphanumeric characters are not ignored – SVT does not have any of these.

localisation accuracy: We generate three synthetic training datasets with increasing levels of sophistication, where the text (1) is placed at random positions within the image, (2) restricted to the local colour and texture boundaries, and (3) distorted perspectively to match the local scene depth (while also respecting the local colour and texture boundaries as in (2) above). All other aspects of the datasets were kept the same — e.g. the text lexicon, background images, colour distribution.

Figure 6 shows the results on localisation on the *SVT* dataset of our method "FCRNall+multi-filt". Compared to random placement, restricting text to the local colour and texture regions significantly increases the maximum recall (+6.8%), AP (+3.85%), and the maximum F-measure (+2.1%). Marginal improvements are seen with the addition of perspective distortion: +0.75% in AP, +0.55% in maximum F-measure, and no change in the maximum recall. This is likely due to the fact that most text instances in the *SVT* datasets are in a fronto-parallel orientation. Similar trends are observed with the *ICDAR* 2013 dataset, but with more contained differences probably due to the fact that *ICDAR*'s text instances are much simpler than *SVT*'s and benefit less from the more advanced datasets.

4.4. End-to-End Text Spotting

Text spotting is limited by the detection stage, as state-of-the-art cropped word image recognition accuracy is over 98% [19]. We utilise our improvements in text localisation to obtain state-of-the-art results in text spotting.

Evaluation protocol. Unless otherwise stated, we follow the standard evaluation protocol by Wang *et al*. [42], where all words that are either less than three characters long or contain non-alphanumeric characters are ignored. An overlap (IoU) of at least 0.5 is required for a positive detection.

Table 3 shows the results on end-to-end text spotting task using the "FCRN + multi-filt" and "FCRNall + multi-filt" methods. For recognition we use the output of the intermediary recognition stage of the pipeline based on the lexicon-encoding CNN of Jaderberg *et al*. [19]. We improve upon previously reported results (F-measure): $+8\%$ on the *ICDAR* datasets, and $+3\%$ on the *SVT* dataset. Given the high recall of our method (as noted before in Figure 5), the fact that many text instances are unlabelled in *SVT* cause precision to drop; hence, we see smaller gains in *SVT* and do worse on *SVT-50*.

	Total Time	Region Proposal	Proposal Filtering	BB-regression & recognition
FCRN+multi-filt	0.30	0.07	0.03	0.20
FCRNall+multi-filt	2.47	0.07	1.20	1.20
Jaderberg *et al*.	7.00	3.00	3.00	1.00

Table 4. Comparison of end-to-end text-spotting time (in seconds).

4.5. Timings

At test time FCRN can process 20 images per second (of size 512×512px) at single scale and about 15 images per second when run on multiple scales (1,1/2,1/4,1/8) on a GPU. When used as high-quality proposals in the text localisation pipeline of Jaderberg *et al*. [20], it replaces the region proposal stage which typically takes about 3 seconds per image. Hence, we gain a speed-up of about 45 times in the region proposal stage. Further, the "FCRN + multi-filt" method, which uses only the high-scoring detections from multi-scale FCRN and achieves state-of-the-art results in detection and end-to-end text spotting, cuts down the number of proposals in the later stages of the pipeline by a factor of 10: the region proposal stage of Jaderberg *et al*. proposes about 2000 boxes which are quickly filtered using a random-forest classifier to a manageable set of about 200 proposals, whereas the high-scoring detections from multi-scale FCRN are typically less than 30. Table 4 compares the time taken for end-to-end text-spotting; our method is between $3\times$ to $23\times$ faster than Jaderberg *et al*.'s, depending on the variant.

5. Conclusion

We have developed a new CNN architecture for generating text proposals in images. It would not have been possible to train this architecture on the available annotated datasets, as they contain far too few samples, but we have shown that training images of sufficient verisimilitude can be generated synthetically, and that the CNN trained *only* on these images exceeds the state-of-the-art performance for both detection and end-to-end text spotting on real images.

Acknowledgements. We thank Max Jaderberg for generously providing code and helpful advice. We are grateful for comments from Jiri Matas. Financial support was provided by the UK EPSRC CDT in Autonomous Intelligent Machines and Systems Grant EP/L015987/2, EPSRC Programme Grant Seebibyte EP/M013774/1, and the Clarendon Fund scholarship.

References

[1] O. Alsharif and J. Pineau. End-to-end text recognition with hybrid HMM maxout models. *ArXiv e-prints*, Oct 2013. 7, 8

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE PAMI*, 33:898–916, 2011. 3, 4

[3] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proc. CVPR*, 2014. 4

[4] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *Proc. ECCV*, 2014. 4

[5] A. Criminisi, I. D. Reid, and A. Zisserman. Single view metrology. In *Proc. ICCV*, pages 434–442, 1999. 2

[6] N. Dalal and B. Triggs. Histogram of Oriented Gradients for Human Detection. In *Proc. CVPR*, volume 2, pages 886–893, 2005. 1

[7] P. Dollar, R. Appel, and S. Belongie. Fast feature pyramids for object detection. *IEEE PAMI*, 36(8):1532–1545, 2014. 1

[8] A. Dosovitskiy and T. Brox. Inverting visual representations with convolutional networks. In *Proc. CVPR*, 2016. To appear. 2

[9] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. ICCV*, 2015. 2

[10] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395, 1981. 4

[11] R. B. Girshick. Fast R-CNN. In *Proc. ICCV*, 2015. 2

[12] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. CVPR*, 2014. 2, 4

[13] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. In *Proc. ICLR*, 2014. 1

[14] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. In *Proc. ACM SIGGRAPH*, 2005. 2

[15] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *Proc. ICCV*, 2005. 2

[16] P. V. C. Hough. Method and means for recognizing complex patterns. US Patent 3,069,654, 1962. 4

[17] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolution neural network induced mser trees. In *Proc. ECCV*, 2014. 6

[18] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, 2015. 6

[19] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. In *Workshop on Deep Learning, NIPS*, 2014. 1, 2, 8

[20] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *IJCV*, 2015. 1, 2, 6, 7, 8, 11, 14

[21] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In *Proc. ECCV*, 2014. 8

[22] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3-d object dataset: Putting the kinect to work. In *ICCV Workshop on Consumer Depth Cameras in Computer Vision*, 2011. 4

[23] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, et al. ICDAR 2015 robust reading competition. In *Proc. ICDAR*, pages 1156–1160, 2015. 2

[24] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, L. P. de las Heras, et al. ICDAR 2013 robust reading competition. In *Proc. ICDAR*, pages 1484–1493, 2013. 2, 6

[25] K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem. Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics*, 30(6):157, 2011. 2

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. 3

[27] K. Lang and T. Mitchell. Newsgroup 20 dataset, 1999. 3

[28] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, ECCV*, May 2004. 4

[29] K. Lenc and A. Vedaldi. R-CNN minus R. In *Proc. BMVC.*, 2015. 2

[30] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proc. CVPR*, 2015. 2, 3, 4

[31] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, 2015. 2, 5

[32] A. Mishra, K. Alahari, and C. Jawahar. Scene text recognition using higher order language priors. *Proc. BMVC.*, 2012. 4

[33] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *Proc. CVPR*, volume 3, pages 1187–1190, 2012. 6, 7

[34] L. Neumann and J. Matas. Scene text localization and recognition with oriented stroke detection. In *Proc. ICCV*, pages 97–104, December 2013. 6, 7, 8

[35] P. Perez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics*, 22(3):313–318, 2003. 3, 4, 12

[36] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proc. CVPR*, 2016. To appear. 2, 5

[37] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2016. 2

[38] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE PAMI*, 31(5):824–840, 2009. 4

[39] A. Shahab, F. Shafait, and A. Dengel. ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. In *Proc. ICDAR*, pages 1491–1496, 2011. 2, 6

[40] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Proc. ECCV*, 2012. 4

[41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 5

[42] K. Wang, B. Babenko, and S. Belongie. End-to-end scene

text recognition. In *Proc. ICCV*, pages 1457–1464, 2011. 8

[43] K. Wang and S. Belongie. Word spotting in the wild. In *Proc. ECCV*, 2010. 2, 3, 6

[44] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *Proc. ICPR*, pages 3304–3308, 2012. 1, 2, 7, 8

[45] C. Wolf and J. M. Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal on Document Analysis and Recognition*, 8(4):280–296, 2006. 6

[46] I. Yildirim, T. D. Kulkarni, W. A. Freiwald, and J. B. Tenenbaum. Efficient and robust analysis-by-synthesis in vision: A computational framework, behavioral tests, and modeling neuronal representations. In *Annual Conference of the Cognitive Science Society*, 2015. 2

[47] Z. Zhang, W. Shen, C. Yao, and X. Bai. Symmetry-based text line detection in natural scenes. In *Proc. CVPR*, 2015. 6

[48] C. L. Zitnick and P. Dollar. Edge boxes: Locating object proposals from edges. In *Proc. ECCV*, pages 391–405, 2014. 1

A. Appendix

We highlight some components of our synthetic text dataset in sections A.1 and A.2, and show some sample images from the dataset in section A.3. Finally, we compare the detection results from our "FCRNall multi-filt" method and Jaderberg et al. [20] on the ICDAR 2013 dataset in section A.4 and the Street View Text (SVT) dataset in section A.5.

A.1. Variation in Fonts, Colors and Sizes

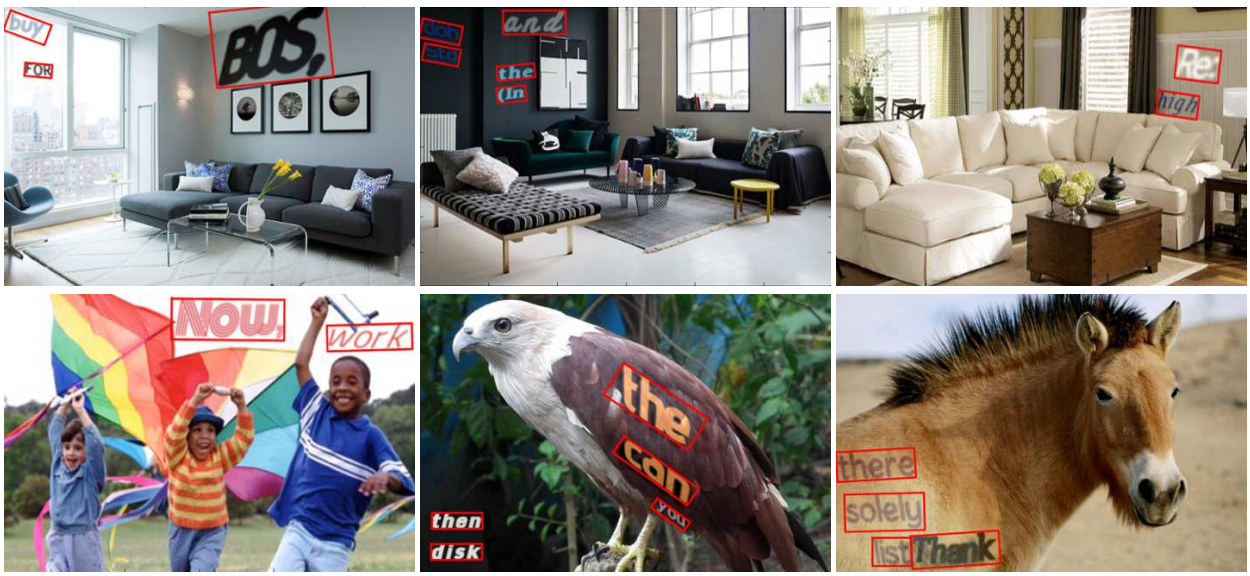
The following images show synthetic text renderings for the same text – "vamos!".
Along the rows, the text is rendered in approximately the same location and against the same background image but in different fonts, colours and sizes.

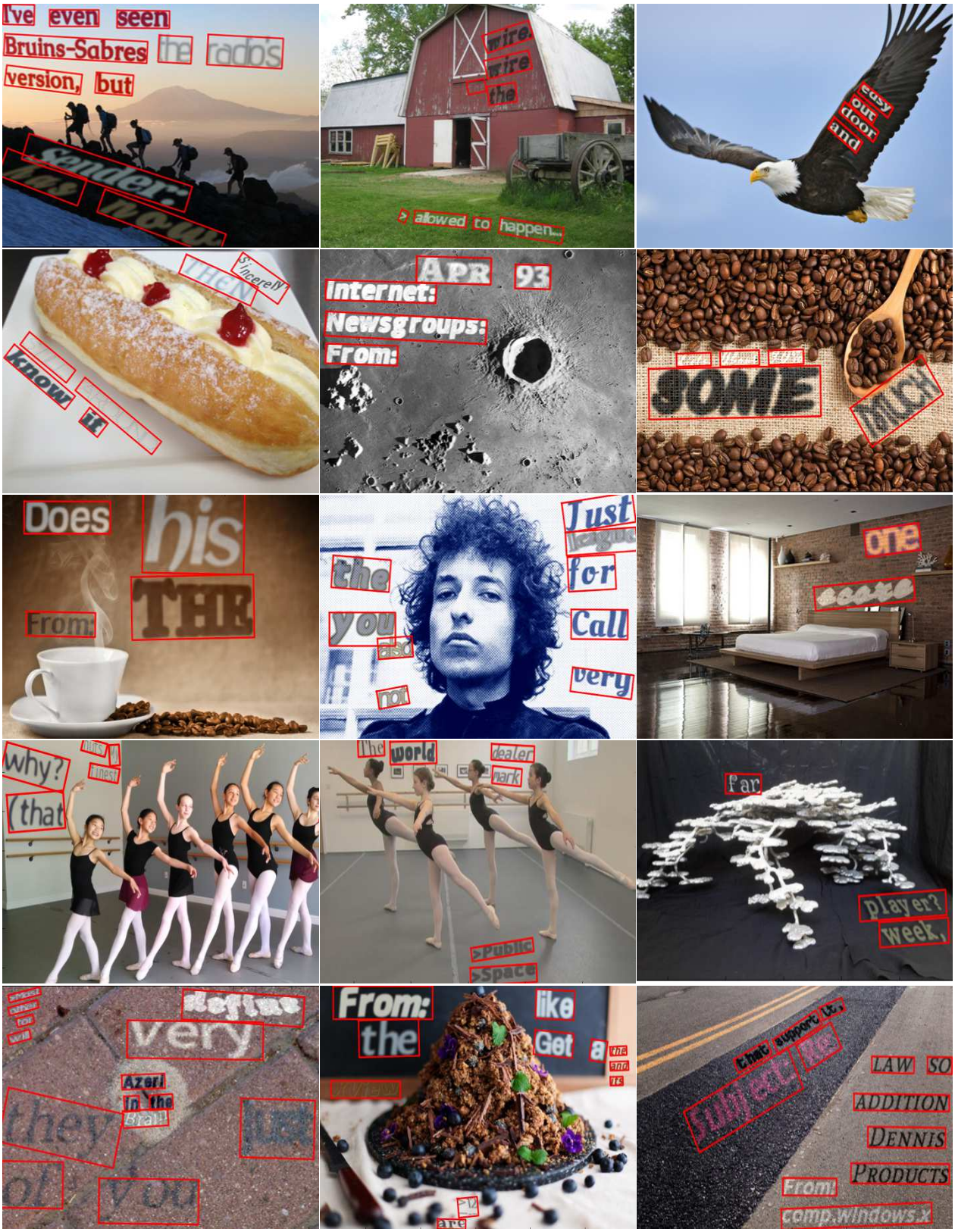
A.2. Poisson Editing vs. Alpha Blending

Comparison between simple alpha blending **(bottom row)** and Poisson Editing [35] **(top row)**.
Poisson Editing preserves local illumination gradient and texture details.

A.3. SynthText in the Wild

Sample images from our synthetic text dataset (continued on the next page).
These images show text instances in various fonts, colours, sizes, with borders and shadows, against different backgrounds, and transformed according to the local geometry and constrained to local contiguous regions of colour and text. Ground-truth word bounding-boxes are marked in red.




I've even seen
Bruins-Sabres the rado's
version, but
wire. wire the
> allowed to happen...
easy our door and
Sincerely
know it
APR 93
Internet:
Newsgroups:
From:
SOME
MUCH
Does his
From:
THE
Just
the
for
you
Call
very
one
why?
(that
The world
dealer
mark
>Public
>Space
far
player?
week,
very
Azeri
in the
they
From:
the
like
Get a
that support it,
LAW SO
ADDITION
DENNIS
PRODUCTS
From:
comp.windows.x

A.4. ICDAR 2013 Detections

Example detections on the ICDAR 2013 dataset from "FCRNall + multi-flit" **(top row)** and those from Jaderberg *et al.* [20] **(bottom row)**. Precision, recall and F-measure values **(P/R/F)** are indicated at the top of each image.

A.5. Street View Text (SVT) Detections

Example detections on the Street View Text (SVT) dataset from "FCRNall + multi-flit" **(top row)** and those from Jaderberg *et al.* [20] **(bottom row)**. Precision, recall and F-measure values **(P/R/F)** are indicated at the top of each image: both the methods have a precision of 1 on these images (except in one case due to missing ground-truth annotation).

4

Learning to Read by Spelling: Towards Unsupervised Text Recognition

This was presented as a *short oral* presentation at the 11th Indian Conference on Computer Vision, Graphics (ICVGIP), 2018.

Learning to Read by Spelling

Towards Unsupervised Text Recognition

Ankush Gupta
Visual Geometry Group
University of Oxford
ankush@robots.ox.ac.uk

Andrea Vedaldi
Visual Geometry Group
University of Oxford
vedaldi@robots.ox.ac.uk

Andrew Zisserman
Visual Geometry Group
University of Oxford
az@robots.ox.ac.uk

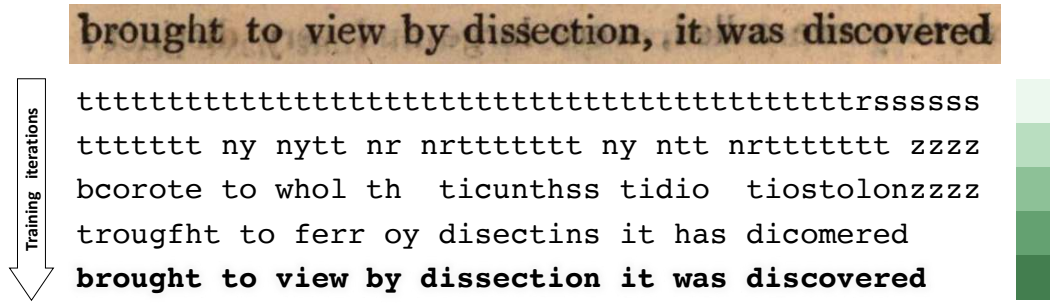


Figure 1: Text recognition from unaligned data. We present a method for recognising text in images without using any labelled data. This is achieved by learning to align the statistics of the predicted text strings, against the statistics of valid text strings sampled from a corpus. The figure above visualises the transcriptions as various characters are learnt through the training iterations. The model first learns the concept of `{space}`, and hence, learns to segment the string into words; followed by common words like `{to, it}`, and only later learns to correctly map the less frequent characters like `{v, w}`. The last transcription also corresponds to the ground-truth (punctuations are not modelled). The colour bar on the right indicates the accuracy (darker means higher accuracy).

ABSTRACT

This work presents a method for visual text recognition without using any paired supervisory data. We formulate the text recognition task as one of aligning the conditional distribution of strings predicted from given text images, with lexically valid strings sampled from target corpora. This enables fully automated, and unsupervised learning from just line-level text-images, and unpaired text-string samples, obviating the need for large aligned datasets. We present detailed analysis for various aspects of the proposed method, namely — (1) impact of the length of training sequences on convergence, (2) relation between character frequencies and the order in which they are learnt, (3) generalisation ability of our recognition network to inputs of arbitrary lengths, and (4) impact of varying the text corpus on recognition accuracy. Finally, we demonstrate excellent text recognition accuracy on both synthetically generated text images, and scanned images of real printed books, using no labelled training examples.

ICVGIP 2018, December 18–22, 2018, Hyderabad, India

ACM ISBN 978-1-4503-6615-1/18/12…$15.00
https://doi.org/10.1145/3293353.3293386

CCS CONCEPTS

• **Computing methodologies** → **Unsupervised learning**; *Image representations*; *Object recognition*; • **Applied computing** → **Optical character recognition**;

KEYWORDS

unsupervised learning, text recognition, adversarial training

ACM Reference Format:
Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. 2018. Learning to Read by Spelling: Towards Unsupervised Text Recognition. In *11th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP 2018), December 18–22, 2018, Hyderabad, India,* Anoop M. Namboodiri, Vineeth Balasubramanian, Amit Roy-Chowdhury, and Guido Gerig (Eds.). ACM, New York, NY, USA, Article 33, 21 pages. https://doi.org/10.1145/3293353.3293386

1 INTRODUCTION

> **read** (riːd) *verb* • Look at and comprehend the meaning of (written or printed matter) by interpreting the characters or symbols of which it is composed.
> **spell** (spɛl) *verb* • Write or name the letters that form (a word) in correct sequence.
> — *Oxford Dictionary of English*

Text recognition, namely the problem of reading text in images, is a classic problem in pattern recognition and computer vision that has enjoyed continued interest over the years, owing to its many practical applications, such as recognising printed [71, 77] or handwritten [12, 45] documents, or more recently, text in natural

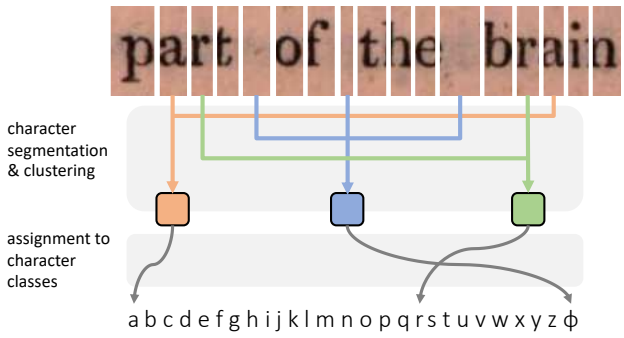


Figure 2: Unsupervised text recognition can be factored into two sub-problems: (1) *visual*: segmentation at the character-level, followed by clustering (or recognition) into a known number of classes, and (2) *linguistic*: determining the character identity of these clusters based on language constraints. Three character classes corresponding to {a, ϕ, r} are visualised above (ϕ stands for {space}). A given text image is mapped to a sequence of characters using a fully-convolutional network; the predicted sequences are compared against linguistically valid text-strings using an *adversarial discriminator*, which guides the mapping of the characters to the correct identity. The two networks trained end-to-end jointly, enable text recognition without any labelled training data.

images [37, 56, 60]. Consequently, many different and increasingly accurate methods have been developed. Yet, all such methods adopt the same *supervised learning* approach that requires example images of text annotated with the corresponding strings.

Annotations are expensive because they must be *aligned* to individual training images. For example, for a text-image of ***cats***, the corresponding annotation is the string {c,a,t,s}. A straightforward but tedious approach is to collect such annotations manually [39, 59, 79]; however, since datasets often comprise several million examples [34, 43], this scales poorly. Another, perhaps more pragmatic, approach is to engineer highly-sophisticated synthetic data generators to mimic real images [26, 34, 81]. However, this requires developing new generators for each new textual domain, and could be problematic for special cases such as text in ancient manuscripts.

We propose instead to develop learning algorithms that can work with *unaligned* annotations. In this paradigm, images containing text can be extracted *e.g.* from scanned documents or by mining online image collections [35]. Independently, strings containing the same *type* of text (but not exactly the same text) can be readily harvested from machine readable text corpora (*e.g.* WMT datasets [1]). Both steps can be implemented economically in a fully-automated manner, making such an approach highly desirable.

More specifically, we demonstrate visual text recognition by only providing examples of valid textual strings, but without requiring them to be aligned to the example images. In this manner, the method is almost unsupervised, as by only knowing how to **spell** correctly, it learns to **read**. The method works by learning a predictor that converts images into strings that *statistically* match the target corpora, implicitly reproducing quantities such as letter and word frequencies, and n-grams. We show empirically that this seemingly weak principle is in fact sufficient to drive learning successfully (section 5).

Text recognition can be factored into two sub-problems (see fig. 2): (1) *visual*: segmenting the text-image into characters and clustering the different characters into a known number of distinct classes, and (2) *linguistic*: assigning these clusters to the correct character identity. Indeed, earlier attempts at unsupervised text recognition proposed two-stage solutions corresponding to the two sub-problems [3, 30, 38, 41]. We address the first problem by exploiting the properties of standard fully-convolutional networks [53] — namely locality and translation invariance of the network's filters. The second problem is equivalent to solving for the correct permutation, or breaking a 1:1–substitution cipher [64]. The latter problem is NP-hard under a bi-gram language model [62]. While several solutions like aligning uni-gram (*i.e.* frequency matching) or n-gram statistics [52, 76] have been proposed traditionally for breaking ciphers [16], we instead adopt an adversarial approach [22]. The result is a compact fully-convolutional sequence (*i.e.* multiple words/text-string) recognition network which is trained against a discriminator in an end-to-end fashion. The discriminator uses as input only unaligned examples of valid text strings.

We study various factors which affect training convergence, and use synthetically-generated data for these controlled experiments. We also show excellent recognition performance on real text images from the Google1000 dataset [23], given *no aligned labelled data.*

The rest of the paper is structured as follows. Section 2 reviews related work, section 3 describes our technical approach, section 4 gives the implementation details, section 5 evaluates the method on the aforementioned data, and section 6 summarises our findings.

2 RELATED WORK

Supervised Text Recognition. Distinct paradigms have emerged and evolved in text recognition. Traditional character-level methods adopt either sliding-window classifiers [35, 56, 80–82], or over / under segment into parts [5, 11, 60], followed by grouping through classification. Words or sentences are then inferred using language models [5, 35, 47, 56, 57, 61, 70, 79–81]. Another set of methods process a whole word image, modelling it either as retrieval in a collection of word images from a fixed lexicon [4, 20, 24, 67] or as learning multiple position dependent classifiers [34, 36, 65]. Our recognition model is similar to these character-sequence classifiers in that we train with a fixed number of output characters; but there is an important difference: we discard their fully-connected (hence, position sensitive) classifier layers and replace them with fully-convolutional layers. This drastically reduces the number of model parameters, and lends generalisation ability to inputs of arbitrary length during inference. More recent methods treat the text-recognition problem as one of sequence prediction in an encoder-decoder framework [14, 75]. [74] adopted this framework first, using HOG features with Connectionist Temporal Classification (CTC) [25] to align the predicted characters with the image features. [28, 68] replaced HOG features with stronger CNN features, while [46, 69] have adopted the soft-attention [9] based recurrent decoders. Note, all these methods learn from labelled training examples.

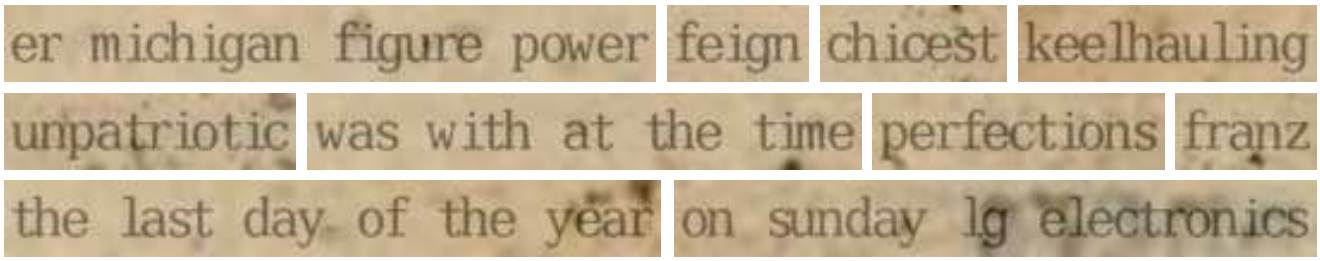


Figure 3: Synthetic text-image samples. A few synthetically generated samples of different lengths, used in the controlled experiments (see section 5). Our model attains ≈99% *character accuracy* and ≈95% *word accuracy* on such samples (section 5.4), after training on only *unaligned* image and text examples.

Unsupervised Text Recognition. Unsupervised methods for text recognition can be classified into two categories. First, category includes generative models for document images. A prime example is the *Ocular* system [10], which jointly models the text content, as well as the noisy rendering process for historical documents, and infers the parameters through the EM-algorithm [15], aided by an n-gram language model. The second category includes methods for automatic decipherment. Decipherment, is the process of mapping unintelligible symbols (ciphertext) to known alphabet/language (plaintext). When the input is visual symbols, it becomes equivalent to text recognition. Some early works [13, 58] for optical character recognition (OCR), indeed model it as such. [29] cluster connected components in binarised document images and assign them to characters by maximising overlap with a fixed lexicon of words based on character frequencies and co-occurrence; [30, 38] also follow the same general approach. [3] break the Borg cipher, a 17th century 408-pages manuscript, by also first clustering symbols but decipher using the noisy-channel framework of [40] through finite-state-machines. [48] learn mappings from hidden-states of an HMM with their transition probabilities initialised with conditional bi-gram distributions. [42] propose an iterative scheme for bootstrapping predictions for learning HMM models, and recognise handwritten text. However, their approach is limited to (1) word images, (2) fixed lexicon (≈44K words) to facilitate exhaustive tree search, whereas, our method is applicable to full *text strings*, does not require a pre-defined lexicon of words.

Unsupervised Learning by Matching Distributions. Output Distribution Matching (ODM) which aligns the *distributions* of predictions with *distributions* of labels was proposed in [76] for "principled" unsupervised learning; although similar ideas for learning by matching statistics have been explored earlier, *e.g.* for decipherment (see above), and also for machine translation [66, 72]. [52] extend ODM to sequences, and apply it to OCR with known character segmentations and pre-trained image features. In essence, ODM [76], or Empirical-ODM [52] minimises the KL-divergence cost between the empirical predicted and ground-truth n-gram distributions. Our learning principle is the same, however, we do not explicitly formulate the matching cost, instead learn it online using an adversary [22]. Recent works [7, 44] have demonstrated unsupervised machine translation using such adversarial losses, however they closely follow the *CycleGAN* framework [83] which learns a bidirectional mapping between the input and target domains to enforce bijection. This framework has also been applied recently in *CipherGAN* to break ciphers [21]. The *CycleGAN* framework learns a bi-directional mapping to enforce strong correlation between the input and the generated output to avoid the degenerate failure mode of collapsing to the same output instance regardless of the input. We, however, dispense with back-translation/reconstruction, and instead enforce correlation directly in the structure of the recogniser by limiting the receptive-field of convolutional layers. Hence, our method is an instantiation of the original (single) generator–discriminator framework of *GANs* [22]. However, our method is perhaps the first to decode sequences of discrete symbols from images using an adversarial framework; these two domains have only been explored independently in *CycleGAN* and *CipherGAN* respectively.

3 METHOD

The aim of text recognition is to predict a sequence of characters given an image of text. Let the image be a tensor $\mathbf{x} \in \mathcal{X} = \mathbb{R}^{H \times W \times C}$, where H, W, C are its height, width, and number of colour channel(s) respectively. Furthermore, let $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in \mathcal{Y}$ denote the corresponding character string where each y_i is a character from an alphabet $\mathcal{A}$ containing K symbols, *i.e.* $|\mathcal{A}| = K$. For later convenience, a character y_i is represented as a K-dimensional one-hot vector. Since such vectors are elements of the K-dimensional simplex Δ^K, we set $\mathcal{Y} = (\Delta^K)^n \subset \mathbb{R}^{K \times n}$. Without loss of generality, we consider strings of a fixed length $n \in \mathbb{N}$. The objective of *unsupervised* text recognition, then, is to learn the mapping $\Phi(\mathbf{x}) = \mathbf{y}$, given only *unpaired* examples from the two domains $\{\mathbf{x}_i\}_{i=1}^N$ where $\mathbf{x}_i \in \mathcal{X}$ and $\{\mathbf{y}_j\}_{j=1}^M$ where $\mathbf{y}_j \in \mathcal{Y}$.

We cast this in an adversarial learning framework based on Goodfellow *et al.* [22]. We view the *text recogniser* $\Phi : \mathcal{X} \to \mathcal{Y}$ as a *conditional generator* of strings $\mathbf{y}$. The recogniser competes against an adversarial discriminator $D_{\mathcal{Y}}$, which aims to distinguish between *real* strings $\{\mathbf{y}\}$ and *generated* strings $\{\Phi(\mathbf{x})\}$. In other words, Φ and $D_{\mathcal{Y}}$ are optimised simultaneously to play the following two-player minimax game [22] $\min_\Phi \max_{D_{\mathcal{Y}}} \mathcal{L}(\Phi, D_{\mathcal{Y}})$ where the value function is given by:

$$\mathcal{L}(\Phi, D_{\mathcal{Y}}) = \mathop{\mathbb{E}}_{\mathbf{y} \sim \mathcal{Y}} [\log D_{\mathcal{Y}}(\mathbf{y})] + \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{X}} [\log(1 - D_{\mathcal{Y}}(\Phi(\mathbf{x}))].$$

The recogniser learns the visual problem of segmenting characters in images, and organising them into distinct categories; while the discriminator, by checking the predicted sequence of characters

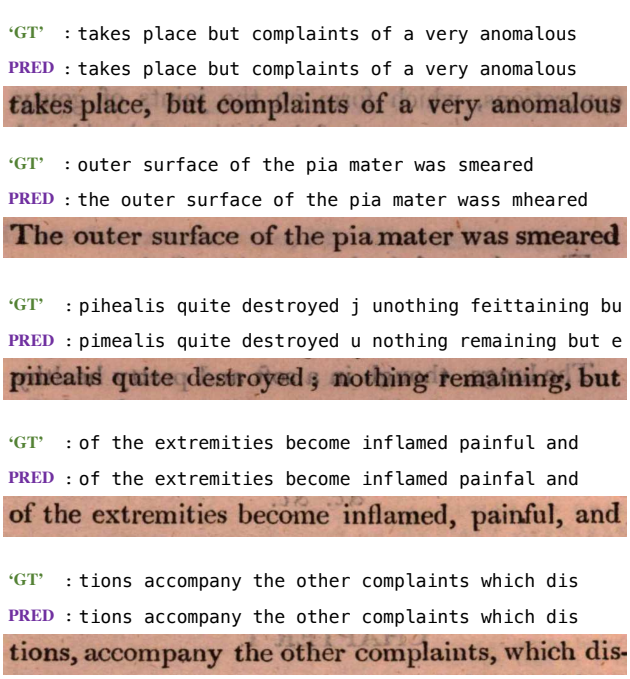


Figure 4: Real text-image samples. Randomly selected samples from a *real* scanned book's test set along with the "ground-truth" (`'GT'`) and the predicted strings (`PRED`); punctuations are not modelled. Our model achieves excellent recognition performance — 96.2% *character*, and 84.8% *word* accuracy (see section 5.6) without using any aligned/labelled training examples. Note the "ground-truth" (`'GT'`) comes from Google's OCR engine output, hence is not perfect (*e.g.* second and third image above).

against linguistically valid strings, guides the assignment of these categories into the respective correct character classes (see fig. 2).

Grounding. A potential pitfall is that the string generator network (or recogniser Φ) may learn to use the input image as a mere source of noise, using it to generate the correct distribution of strings, without learning to recognise the string represented in the image. A useful mapping, instead, must be *grounded*, *i.e.* the generated string $\mathbf{y} = \Phi(\mathbf{x})$ should correspond to the text represented in the input image $\mathbf{x}$.

A possible way to encourage grounding is to ensure that the image $\mathbf{x}$ can be recovered back from the string $\mathbf{y}$. Both *CycleGAN* [83] and *CipherGAN* [21] achieve this by learning a second inverse mapping $\Psi : \mathcal{Y} \rightarrow \mathcal{X}$ from the target domain back to the input and complete the cycle $\Psi(\Phi(\mathbf{x})) \triangleq \mathbf{x}$. However, learning a mapping from character strings to images is highly ambiguous: rendering a given string requires sampling the background image, font style, font colour, geometry of the glyphs, shadows, noise etc. This ambiguity arises because text recognition requires translating between two very different *modalities*, *viz*. text and images, which is much harder than translating within the same modality, *e.g.* between images in *CycleGAN* [83] where only local texture is modified, or between character strings in *CipherGAN* [21], where the characters are permuted.

Instead of enforcing cycle-consistency, we encourage grounding via the following two key architectural modifications in the recogniser Φ (architectural details are given in section 4):

(1) **Prediction Locality.** The character predictor is local, with a receptive field large enough to contain at most two or three characters in the image. While this may sound simple, it embodies a powerful constraint. Namely, such local predictors can generate a string which is globally consistent only if they correctly transduce the structure of the underlying image. Otherwise, local predictors may be able to match local text statistics such as n-grams, but would not be able to match global text statistics, such as forming proper words and sentences (see also section 6.1 of [76] for similar ideas). Global consistency is enforced by the adversarial discrminiator which has a large receptive field over the predicted characters (see section 4).
(2) **Reduced Stochasticity.** We also make the generated strings a deterministic function of the input. We achieve this by removing the noise input from Φ which is normally used in generator networks. Furthermore, we do not use dropout regularization [73].

Training Objective. The discriminator $D_{\mathcal{Y}}$ operates in the domain of *discrete* symbols. While the real symbols are represented as one-hot vectors or *vertices* $\text{Vert}(\Delta^K)$ of the standard simplex, the generated symbols are output of a SoftMax operator over predicted logits, and hence typically belong to the *interior* of the simplex Δ^K. This was identified, as the cause for *uninformative discrimination* in *CipherGAN* [21], where the discriminator distinguishes using this unimportant difference, rather than soundness of the generated strings.

To mitigate this, we adopt their proposed solution and learn a d-dimensional embedding for each of the K symbols in the alphabet, collectively represented by a matrix $W \in \mathbb{R}^{K\times d}$. Furthermore, we replace the log-likelihood loss with a squared difference loss, as proposed by [55]. Hence, we optimise the following revised training objective:

$$\mathcal{L}(\Phi, D_{\mathcal{Y}}, W) = \mathop{\mathbb{E}}_{\mathbf{y}\sim\mathcal{Y}}[D_{\mathcal{Y}}(W^T\mathbf{y})^2] + \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{X}}[(1 - D_{\mathcal{Y}}(W^T\Phi(\mathbf{x})))^2].$$

The embeddings W are trained to aid discrimination among symbols by solving $\min_\Phi \max_{D_{\mathcal{Y}}, W} \mathcal{L}(\Phi, D_{\mathcal{Y}}, W)$. Learning such embeddings improved the speed of convergence and final accuracy, as also noted in [21], while using square differences improved numerical stability.

Discussion: Why is this a feasible learning problem? While learning to recognise visual symbols without any paired data seems unattainable, the tight structure of natural language provides sufficient constraints to enable learning. First, lexically valid text strings form a tiny sub-space of all possible permutations of symbols, *e.g.* there are only $\approx 13k$ valid English words of length 7, as opposed to almost 8 billion permutations of the 26 English letters. Second, the relative frequencies of the characters and their co-occurrence patterns impose further constraints (see section 5.3 for correlation between character-frequency and learning). These constraints combined with strong correlation between the input image and the predicted characters are sufficient to drive learning successfully.

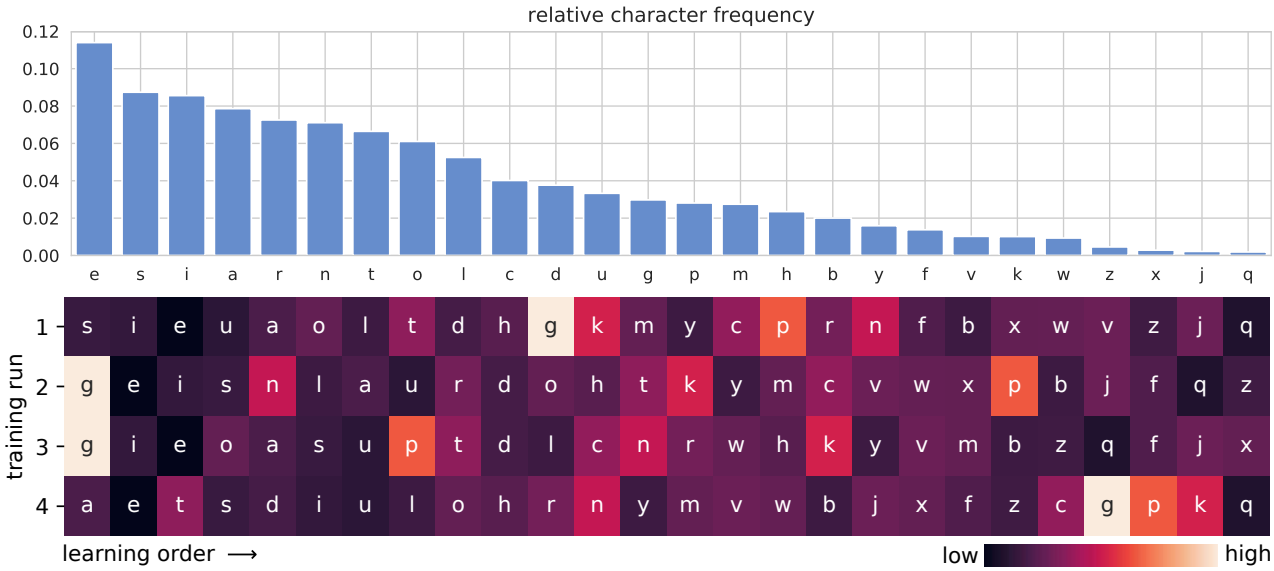


Figure 5: Learning order for different characters. The order in which the various characters are learnt is strongly correlated (Spearman's rank correlation coefficient $\rho = 0.80$, p-value $< 1e{-}5$) to their frequency in the English language **[top]**. Ranking for the learning order is based on the training iteration number at which the model achieves 50% accuracy for a given character. **[bottom]** Rankings from four different training runs are presented to show the variance — bright colours signify high variance in rank across runs, while dark colours correspond to low variance. The character {g} is a curious exception to the trend, as it is sometimes learnt first (runs 2, 3); see section 5.3 for the reason and further discussion.

4 IMPLEMENTATION

Both, the recogniser (Φ) and the discriminator ($D_{\mathbf{y}}$) are implemented as fully-convolutional networks [53]. The recogniser ingests an image of text and produces a sequence of character logits. The discriminator operates instead on character strings represented as sequence of character vectors, and produces a scalar discrimination score as output. The discriminator acts as a *spell-checker*, pointing out the errors in the generated strings. We describe their architecture and optimisation details below.

Recogniser Φ. We train our models for strings of a maximum fixed number of characters = n. To this end, the input image dimensions are held fixed at $32\times(n\cdot 2^4)$ pixels (= height×width). Hence, an image of size $H\times W$ is scaled to $H'\times W' = 32\times\min(\lceil W\cdot\frac{32}{H}\rceil, n\cdot 2^4)$; if the $W' < n\cdot 2^4$, it is padded on the right with the mean channel intensity. The recogniser employs four blocks, each consisting of two convolution layers, followed by a 2×2 max-pooling layer. Each convolutional layer comprises of 32 filters of 3×3 dimensions, and is followed by batch-normalisation [32] and leaky-ReLU activation (slope= 0.2) [54]. Since max-pooling in each block downsamples the input by a factor of two, final output dimensions are $2\times n\times D$ (where, $D = 32$ is the number of features). The height is collapsed using average-pooling, and each of the n D-dimensional feature vectors are mapped to $|\mathcal{A}| = K$ dimensional logits through linear projection, yielding a $K\times n$ dimensional tensor. Note the receptive field of the final (prediction) layer is small to encourage *locality*; specifically, it is 76 pixels wide which corresponds to ≈2.5 characters in the image. Although we train our recogniser on fixed-length strings, yet it generalises to different other lengths due to its fully-convolutional architecture (see section 5.4).

Discriminator (or spell-checker) $D_{\mathbf{y}}$. The input $\mathbf{y}$ to the discriminator are $K\times n$ dimensional tensors of predicted and real strings containing n characters, represented as logits and one-hot vectors, respectively (see fig. 6). The predicted logits are first normalised through SoftMax to a valid probability distribution over

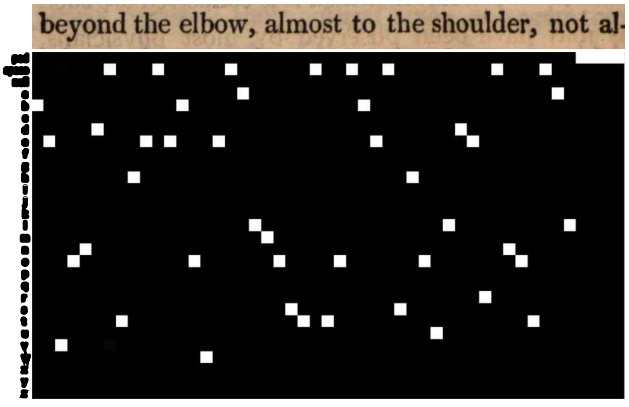


Figure 6: Character sequence representation. Text strings are represented as sequences of n one-hot (for *real* strings) or SoftMax normalised logits (for *predictions*) over $|\mathcal{A}| = K$ character classes. A sample image and the model's prediction are visualised above (one-hot *real* strings look similar); here $K = 29$ and $n = 50$.

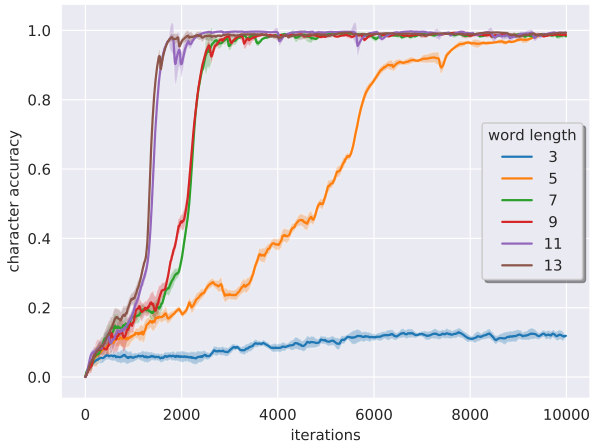


Figure 7: Effect of text length on convergence. Training with longer words leads to faster convergence: the order of convergence {13,11,9,7} mirrors the word lengths (see section 5.2). No convergence is seen for models trained on shorter words of length 3 and 5. For each word-length, the run with largest area-under-curve (AUC) from eight trials is plotted. Shading along the curves represents values within one standard-deviation.

the K characters for each of the n positions. Next, embeddings $\mathbf{y}_e \in \mathbb{R}^{d \times n}$ for both, the real and predicted strings are obtained: $\mathbf{y}_e = W^T \mathbf{y}$, where $W \in \mathbb{R}^{K \times d}$ are the character embeddings ($d = 256$). We adopt the fully-convolutional *PatchGAN* discriminator architecture [33, 50, 51], where patches correspond to substrings here. The embedded input $\mathbf{y}_e$ is fed to a stack of five 1D-convolutional layers, each with 512 filters of size 5. This amounts to a final receptive field of 21 characters which helps to enforce long-range structure. Each layer is followed by layer-normalisation [8] and leaky-ReLU (slope = 0.2); zero padding is used to maintain the size. The resulting $d \times n$ dimensional output is linearly projected to $1 \times n$, and average-pooled to obtain the final scalar score $D_{\mathcal{Y}}(\mathbf{y})$.

Optimization details. Recogniser, discriminator and character embeddings are trained jointly end-to-end. The parameters are initialised with Xavier initialization [19]. We use the RMSProp optimizer [78] with a constant learning rate of 0.001. The two-part discriminator loss objective is multiplied with $\frac{1}{2}$ as in [33]. The models are implemented in TensorFlow [2].

5 EXPERIMENTS

Our experiments have two primary goals. First, is an extensive analysis of various factors which affect the training: we study — (1) the impact of the length of training sequences on convergence (section 5.2); (2) the order in which various characters are learnt and its correlation with their frequencies (section 5.3), (3) generalisation ability of the fully-convolutional recogniser to different sequence lengths (section 5.4), and (4) impact of varying the text corpus on recognition accuracy (section 5.5). For these experiments we use synthetically generated text data as it provides fine control over various nuisance factors. The second objective is to show applicability of the proposed method to *real* document images

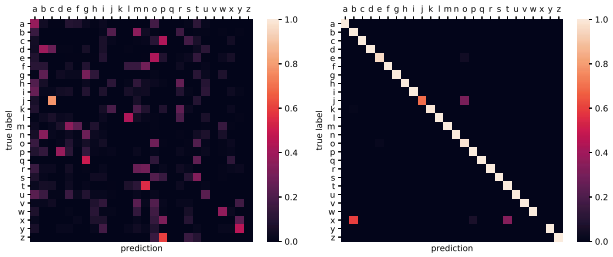


Figure 8: Confusion matrices for models trained on words of length 3 & 7. The model trained on length-3 words does not converge, while the one trained on length-7 words does (see fig. 7 and section 5.2). Further, the accuracy for a character depends on its frequency: the length-7 model **[right]** recognises most characters with high accuracy, yet it confuses the two least common characters {j,x} (see section 5.3).

(section 5.6). We first describe the datasets used in our experiments in section 5.1, and then present the results.

5.1 Datasets

Synthetic data. We generate synthetic text data to simulate old printed documents. Synthetic data aids the controlled ablation studies, as it provides tight control over the various factors, *e.g.* text content, font style and glyph geometry, background, colours, and other noise parameters. We sample the text content from two different sources depending on the experimental setting — (1) *words*: individual English words are sourced from a lexicon of 90K words used in the Hunspell spell-checker [31], and (2) *lines*: these are full valid English language text strings extracted from the 2011 news-crawl corpus provided by WMT [1]. Note, these text sources are used for rendering images, as well as for providing examples of valid strings to the discriminator. To limit the variance in position of characters, we use the `VerilySerifMono` fixed-width font. The background image data is sampled from the margins of historical books [6] to simulate various noise effects. The font colour is sampled from a k-means colour model learnt from the same dataset. The character set consists of the 26 English letters, one space character, and one additional `null` class for padding smaller strings, *i.e.* $|\mathcal{A}| = K = 28$. Punctuations, and other symbols in the text are ignored; lower and upper case letters are mapped to the same class. Different synthetic datasets are generated as required by the experiments; the training sets consist of 100k image samples, while the tests set contain 1k samples. Figure 3 visualises some synthetically generated samples.

Real data. For testing the validity of our method on real text images, we use a scanned historical printed book from the Google1000 dataset [23]. Specifically, we use the book titled *Observations on the Nature and Cure of Gout* by James Parkinson [63]. For simplicity, we discard cover, title and start-of-chapter pages, and pages with significant number of footnotes; we only work with the remaining 140 pages (total 200 pages) which contain text in a relatively uniform font. Nevertheless, this data is still challenging due to: (1) non-fixed-width font which makes character segmentation difficult, (2) varying spacing between words due to fully justified

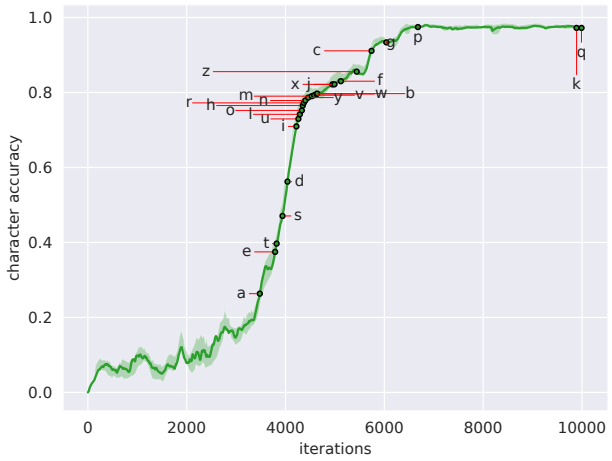


Figure 9: Temporal learning order. Training iterations for a model trained on length-7 words, annotated at the steps when it becomes at least 50% accurate for each character. This curve corresponds to `run-#4` in fig. 5. The characters are learnt in the order of their frequencies (see section 5.3). Shading along the curve represents values within one standard-deviation.

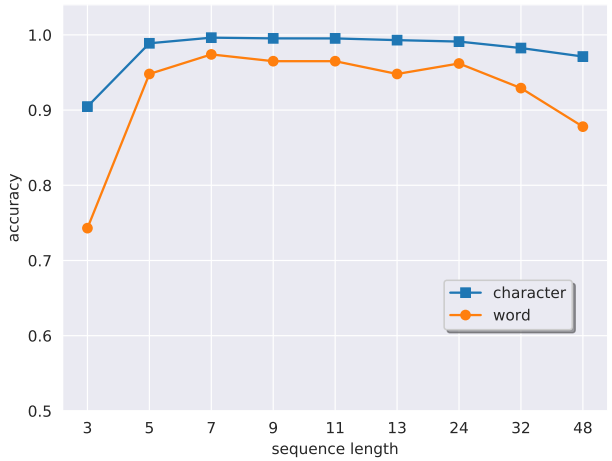


Figure 10: Generalisation to different sequence lengths. A model trained on text-strings of length 24, is evaluated on images containing both shorter and longer strings of lengths — $\{3, 5, 7, 9, 11, 13, 32, 48\}$. Word and character accuracies are plotted. The fully-convolutional architecture of the recognition network enables significant generalisation to lengths not in the training set, with small variance in performance (see section 5.4).

alignment, (3) varying case (lower/upper) and italics, (4) different background colours and textures, (5) show-through from the back of the page, (6) fading and other noise elements, and (7) presence of various punctuations and other symbols. We use the localisation output of the provided OCR engine output to segment the pages into lines; first 300 lines are assigned to the test set, while the remaining 3000 form the training set (no page is shared between the splits). We use the provided OCR text output for lines in the training split, as examples of valid text strings for the discriminator. Note, these strings are sampled uniformly at random during training, and hence, do not have any direct correspondence to images in the training batch. The text lines typically consist of ≈50 characters. The character-set consists of 26 English letters, one space character, one unknown `<UNK>` character, and one `null` class for padding, for a total of $|\mathcal{A}| = K = 29$ characters. We do not distinguish between upper and lower cases; the following symbols and punctuations: `, . ? ! ‘ " * ( )` are suppressed (ignored), and any other character is mapped to `<UNK>`. Figure 4 visualises some sample text-lines.

Metrics. We measure accuracy at the *character* and *word* levels:

- **character accuracy**: this is computed as $1 - \frac{1}{N}\sum_{i=1}^{N} \frac{\text{EditDist}\left(\mathbf{y}_{\text{gt}}^{(i)}, \mathbf{y}_{\text{pred}}^{(i)}\right)}{\text{Length}\left(\mathbf{y}_{\text{gt}}^{(i)}\right)}$, where $\mathbf{y}_{\text{gt}}^{(i)}$ and $\mathbf{y}_{\text{pred}}^{(i)}$ are the i^{th} ground-truth and predicted strings respectively in a dataset containing N strings; EditDist is the *character-level Levenshtein* distance [49]; Length $\left(\mathbf{y}_{\text{gt}}\right)$ is the number of characters in $\mathbf{y}_{\text{gt}}$.
- **word accuracy**: computed as *character accuracy* above, but here the *Levenshtein* distance uses *words* (contiguous strings demarcated by space) as tokens, and is normalized by number of ground-truth words in $\mathbf{y}_{\text{gt}}$.

5.2 Effect of text length on convergence

Although earlier works use low-order, namely uni/bi-gram statistics for alignment [42, 48], higher-order n-grams could be more informative. In this experiment we examine the impact of the *length* of the training text-sequences on convergence. We train separate models on synthetic datasets containing *one* word of a given length, namely — $\{3, 5, 7, 9, 11, 13\}$. Figure 7 tracks *character accuracy* as the training progresses; due to instabilities in training GANs, we train on each word-length eight times, and plot the run with the maximum area-under-curve (AUC) (earliest "take-off"); all eight runs are reproduced in appendix E for completeness. Note, models trained on longer words converge faster, achieving ≈99% *character accuracy*. In detail, the model trained on length-13 words converges the fastest, followed by those trained on 11, 9, 7, and 5 (in order). However, the difference is minor for length 13 and 11, and 9 and 7. No convergence is seen for the short length of 3. This confirms that longer text-sequences impose stronger structural constraints on the possible outputs, leading to faster convergence. Figure 8 visualises the confusion matrices for models trained on lengths 3 and 7; the length-3 model confuses most characters, whereas the length-7 model recognises most characters almost perfectly. The length-3 model does not learn to recognise the image, but instead produces valid 3-grams which fool the discriminator. Note, this is not feasible for longer lengths, as the receptive-field of the recogniser is limited to about 3 characters; see appendix B for experiments on the effect of receptive-field size on convergence.

5.3 Which character is learnt first?

We examine the dynamics of learning, more specifically, we probe the order in which the model learns about different symbols — is there a pattern? Figure 5 visualises the order in which models

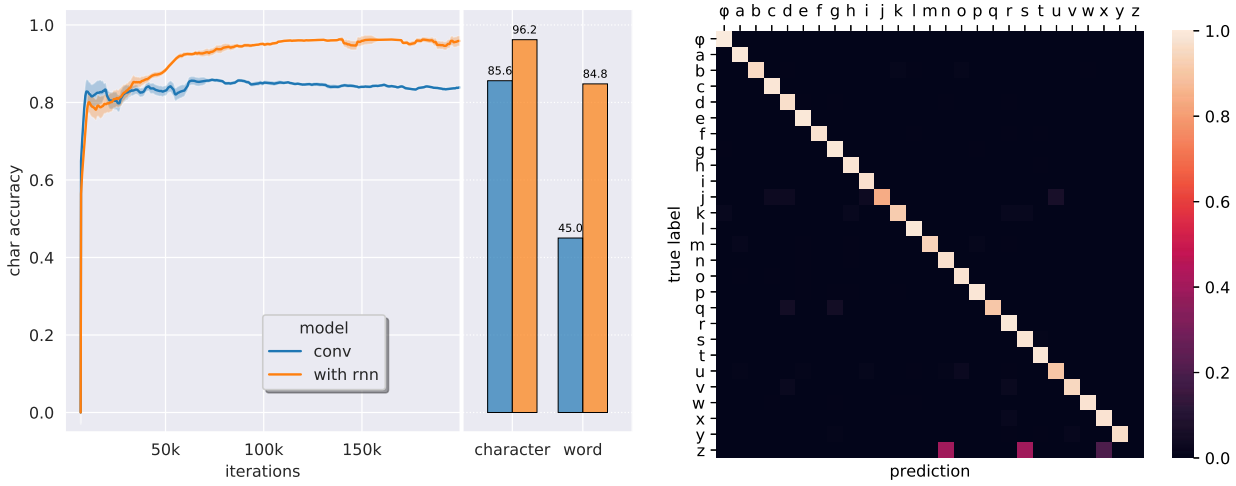


Figure 11: Recognising historical printed books. [left] *Character & word accuracy* on the test split of the *real* dataset (see fig. 4) for both the fully-convolutional and the *skip-RNN* recognition models. *Skip-RNN* dramatically improves: *character accuracy* from 85.6% to 96.2%, and *word accuracy* from 45.0% to 84.8%. *Character accuracy* on the test-set is also visualised against the training iterations. Shading along the curves represents values within one standard-deviation. **[right]** Confusion matrix on the test set: all characters are predicted with high accuracy, except for the low-frequency {z}. {φ} stands for the {space} character (see section 5.6).

(trained on synthetic word images of length 7) achieve an accuracy of at least 50% for each character. We note that this ranking is highly correlated with the frequency of the characters in the English language — Spearman's rank correlation coefficient $\rho = 0.80$, p-value $< 1e{-}5$. It further visualises the variance in the ranking of the characters across multiple runs. The characters at the extremities of the frequency distribution have low variance — common characters (*e.g.* e,s,i,a) are almost always learnt first, and the least common characters (*e.g.* z,x,j,q) are learnt last; while characters in the middle, *viz.* {g,p} show the highest variance. The character {g} is a curious exception as it is sometimes learnt first. This is because of 8.54% of the training (length-7) words end in the suffix '-ing'. Hence, {g} appears at the last position quite frequently, and becomes relatively easy to learn. Figure 9 annotates the training steps at which the accuracy for a character first reaches 50%. After the model becomes confident about the first symbol {a}, it quickly learns the other most commons ones; then it slowly learns the less frequent symbols in the order of their frequencies. Further, fig. 8 visualises the confusion-matrices for models trained on word-length 7. Again, we can note the dependence on character frequencies — the length-7 model is almost perfect at recognising most characters, yet it confuses two of the least common characters {j,x}.

5.4 Generalisation to different lengths

The fully-convolutional architecture of our recognition network generalises to images of lengths significantly different from those it was trained on. To demonstrate this, we train a model on synthetic *text-strings* of length 24 (containing multiple words), and evaluate on synthetic images of different lengths: (1) *shorter* single-word images of lengths — $\{3, 5, 7, 9, 11, 13\}$, and (2) *longer* text-string (multiple words) images of lengths — $\{32, 48\}$. Figure 10 plots the recognition accuracy against the word lengths. We note excellent and consistent *character* (≈99%) and *word* accuracies (≈95%) for both, shorter and longer lengths ($5 - 32$). Note, this demonstrates significant generalisation ability, as the model is never trained on such images. There is a drop in the character accuracy (≈95%) for length-48 text-strings, as the model does not learn a long-range language model. Performance suffers for words of length 3 due to image-edges being close in short images, which is not encountered during training with images of long words.

5.5 Varying the text corpus

We examine the impact of varying the text corpus from which samples of text strings are obtained, on the recognition accuracy. We examine the following three different sources for sampling the strings. The synthetic text-images are held constant across the three settings, and contain up to 24 characters with the text content in them sampled from WMT newscrawl (as before). (1) same strings as in text-images but randomly sampled for each batch, (2) strings from the same corpus (WMT newscrawl) but with no overlap with text-image strings, and (3) strings sampled from a very different corpus, namely, Tolstoy's *War and Peace*. Table 1 summarizes the *character* and *word* accuracies. Training with the completely unrelated lexicon

corpus →	(1) WMT	(2) WMT no overlap	(3) War & Peace
char	98.98	99.13	98.43
word	95.33	96.08	92.53

Table 1: Effect of varying the text corpus on recognition accuracy. Text-strings are sampled from three increasingly distant text corpora. This has a small adverse effect on recognition *word* and *character* accuracies (in %) (see section 5.5).

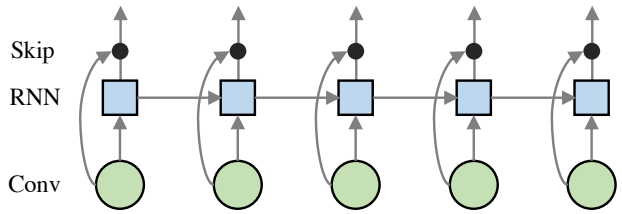


Figure 12: Skip-RNN architecture for real text images. Non-uniform spacing and non-fixed width fonts pose a significant challenge to the fully-convolutional recogniser. We augment the recognition network with a *skip-RNN*, which acts on the convolutional features, and predicts residual updates to the inputs (the residual predictions are *added* to the inputs). This improves the *word accuracy* from ≈45% to ≈85% (see fig. 11).

(#3) does have a small adverse effect (*word* accuracy drops to 92.53% from ≈ 95%), while using a related lexicon (#2) does not have such an effect.

5.6 Recognising a historical printed book

Finally, we apply our model to *real* text-line images extracted from a historical printed book (see section 5.1 for dataset details). As noted in section 5.1, non-fixed width fonts and fully-justified text alignment introduce non-uniform spacing between characters and words. This poses a significant challenge to the fully-convolutional recogniser, making segmentation of the text-image into individual characters difficult (see fig. 4 for example images). Hence, we augment the penultimate layer of the fully-convolutional recogniser with a *skip-RNN* — a *uni*-directional (left to right) RNN (256-dimensional LSTM) with a residual skip-connection [27] (see fig. 12). The RNN lends pliability to the convolutional features, thereby aids character segmentation. All other model parameters are as those used for the synthetic data experiments (see section 4), except: (1) the discriminator filter size is increased from 5 to 11, and (2) number of layers is doubled to 8 to exploit the long-term structure in the much longer text strings (≈50 characters each). Figure 11 visualises the *character* and *word* accuracies for both the fully-convolutional and *skip-RNN* recognition models. *Skip-RNN* dramatically improves the recognition performance: *word accuracy* improves from 45.0% to 84.8%, while *character accuracy* improves from 85.6% to 96.2%. Figure 4 visualises randomly selected examples from the test set and shows the model's predictions; the predictions are comparable to the "ground-truth" annotations obtained from Google's OCR engine. Figure 11 also visualises the confusion matrix for the character classes: all characters are predicted with high accuracy, except for the low-frequency `{z}`. Full page read-outs from our model are visualised in appendix A.

6 CONCLUSION

We have developed a method for training a text recognition network using only *unaligned* examples of text-images and valid text strings. We have presented detailed analysis for various aspects of the proposed method. We have established — (1) positive correlation between the length of the input text and convergence rates; (2) the order in which the characters are learnt is strongly dependent on their relative frequencies in the text; (3) the generalisation ability of our method to input images of different lengths, specifically our recognition model trained on strings of length 24 generalises to both much shorter and longer strings (3 – 48) without drastic degradation in performance; (4) the effect of varying the text corpus used as the source of valid sentences on the recognition accuracy. Finally, we have shown successful recognition on real text images, without using any labelled supervisory data. These results open up a new and promising direction for training sequence recognition models for structured domains (*e.g.* language) given no labelled training data. The proposed method is applicable not just to text images, but other modalities as well, *e.g.* speech and gestures.

ACKNOWLEDGMENTS

We thank Iasonas Kokkinos, Triantafyllos Afouras, and Weidi Xie for insightful discussions, and anonymous reviewers for their detailed feedback. Financial support was provided by UK EPSRC AIMS CDT EP/L015987/2, EPSRC Seebibyte Grant EP/M013774/1, and Clarendon Fund scholarship.

REFERENCES

[1] EMNLP conference on machine translation, 2018.
[2] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.
[3] N. Aldarrab. Decipherment of historical manuscripts. Master's thesis, University of Southern California, 2017.
[4] J. Almazán, A. Gordo, A. Fornés, and E. Valveny. Word spotting and recognition with embedded attributes. *IEEE PAMI*, 36:2552–2566, 2014.
[5] O. Alsharif and J. Pineau. End-to-end text recognition with hybrid HMM maxout models. In *Proc. ICLR*, 2014.
[6] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher. ICDAR 2013 competition on historical book recognition (hbr 2013). pages 1459–1463. IEEE, 2013.
[7] M. Artetxe, G. Labaka, E. Agirre, and K. Cho. Unsupervised neural machine translation. In *Proc. ICLR*, 2017.
[8] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
[9] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*, 2015.
[10] T. Berg-Kirkpatrick, G. Durrett, and D. Klein. Unsupervised transcription of historical documents. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 207–217, 2013.
[11] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. PhotoOCR: Reading text in uncontrolled conditions. In *Proc. ICCV*, 2013.
[12] H. Bunke, S. Bengio, and A. Vinciarelli. Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. *PAMI*, 26(6):709–720, 2004.
[13] R. G. Casey. Text OCR by solving a cryptogram. 1986.
[14] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
[15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. 39 B:1–38, 1977.
[16] J. F. Dooley. *A brief history of cryptology and cryptographic algorithms*. Springer, 2013.

[17] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Proc. NIPS*, 2016.

[18] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proc. CVPR*, 2016.

[19] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[20] V. Goel, A. Mishra, K. Alahari, and C. V. Jawahar. Whole is greater than sum of parts: Recognizing scene text words. In *International Conf. on Document Analysis and Recognition (ICDAR)*, pages 398–402, 2013.

[21] A. N. Gomez, S. Huang, I. Zhang, B. M. Li, M. Osama, and L. Kaiser. Unsupervised cipher cracking using discrete GANs. In *Proc. ICLR*, 2018.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc. NIPS*, 2014.

[23] Google Inc. Book search dataset, Aug 2018. Version V.

[24] A. Gordo. Supervised mid-level features for word image representation. In *Proc. CVPR*, 2015.

[25] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.

[26] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proc. CVPR*, 2016.

[27] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[28] P. He, W. Huang, Y. Qiao, C. Loy, and X. Tang. Reading scene text in deep convolutional sequences, 2016. In *The 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, volume 1, 2016.

[29] T. K. Ho and G. Nagy. OCR with no shape training. In *Proc. ICPR*, 2000.

[30] G. Huang, E. Learned-Miller, and A. McCallum. Cryptogram decoding for optical character recognition. 2007.

[31] Hunspell. https://hunspell.github.io.

[32] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, 2015.

[33] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, 2017.

[34] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. In *Workshop on Deep Learning, NIPS*, 2014.

[35] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In *Proc. ECCV*, 2014.

[36] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Deep structured output learning for unconstrained text recognition. In *International Conference on Learning Representations*, 2015.

[37] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *IJCV*, 116(1):1–20, Jan. 2016.

[38] A. Kae and E. Learned-Miller. Learning on the fly: font-free approaches to difficult OCR problems. 2009.

[39] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, L. P. de las Heras, et al. ICDAR 2013 robust reading competition. In *Proc. ICDAR*, pages 1484–1493, 2013.

[40] K. Knight, A. Nair, N. Rathod, and K. Yamada. Unsupervised analysis for decipherment problems. In *Proceedings of the COLING/ACL*, pages 499–506. Association for Computational Linguistics, 2006.

[41] K. Knight, B. Megyesi, and C. Schaefer. The Copiale cipher. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*. Association for Computational Linguistics, 2011.

[42] M. Kozielski, M. Nuhn, P. Doetsch, and H. Ney. Towards unsupervised learning for handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 549–554. IEEE, 2014.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. NIPS*, pages 1106–1114, 2012.

[44] G. Lample, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. In *Proc. ICLR*, 2017.

[45] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[46] C. Lee and S. Osindero. Recursive recurrent nets with attention modeling for OCR in the wild. In *Proc. CVPR*, 2016.

[47] C. Lee, A. Bhardwaj, W. Di, V. Jagadeesh, and R. Piramuthu. Region-based discriminative feature pooling for scene text recognition. In *Proc. CVPR*, 2014.

[48] D.-S. Lee. Substitution deciphering based on HMMs with applications to compressed document processing. *PAMI*, (12):1661–1666, 2002.

[49] V. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics Doklady*, volume 10, page 707, 1966.

[50] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proc. ECCV*, pages 702–716. Springer, 2016.

[51] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Proc. NIPS*, pages 700–708, 2017.

[52] Y. Liu, J. Chen, and L. Deng. Unsupervised sequence classification using sequential output statistics. In *Proc. NIPS*, pages 3550–3559, 2017.

[53] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, 2015.

[54] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, page 3, 2013.

[55] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *Proc. ICCV*, pages 2813–2821. IEEE, 2017.

[56] A. Mishra, K. Alahari, and C. Jawahar. Scene text recognition using higher order language priors. *Proc. BMVC*, 2012.

[57] A. Mishra, K. Alahari, and C. Jawahar. Top-down and bottom-up cues for scene text recognition. In *Proc. CVPR*, 2012.

[58] G. Nagy. Efficient algorithms to decode substitution ciphers with applications to OCR. In *Proc. ICPR*, pages 352–355, 1986.

[59] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS DLW*, volume 2011, 2011.

[60] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *Proc. CVPR*, volume 3, pages 1187–1190. IEEE, 2012.

[61] T. Novikova, O. Barinova, P. Kohli, and V. Lempitsky. Large-lexicon attribute-consistent text recognition in natural images. In *Proc. ECCV*, pages 752–765. Springer, 2012.

[62] M. Nuhn and H. Ney. Decipherment complexity in 1: 1 substitution ciphers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 615–621, 2013.

[63] J. Parkinson. *Observations on the Nature and Cure of Gout: On Nodes of the Joints; and on the Influence of Certain Articles of Diet, in Gout, Rheumatism, and Gravel*. Symonds, 1805.

[64] S. Peleg and A. Rosenfeld. Breaking substitution ciphers using a relaxation algorithm. *Communications of the ACM*, 22(11):598–605, 1979.

[65] A. Poznanski and L. Wolf. CNN-N-Gram for handwriting word recognition. In *Proc. CVPR*, 2016.

[66] S. Ravi and K. Knight. Attacking decipherment problems optimally with low-order n-gram models. In *proceedings of the conference on*

Empirical Methods in Natural Language Processing, pages 812–819. Association for Computational Linguistics, 2008.

[67] J. A. Rodriguez-Serrano, A. Gordo, and F. Perronnin. Label embedding: A frugal baseline for text recognition. *International Journal of Computer Vision*, 113(3):193–207, 2015.

[68] B. Shi, X. Bai, and C. Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *ArXiv e-prints*, 2015.

[69] B. Shi, X. Wang, P. Lv, C. Yao, and X. Bai. Robust scene text recognition with automatic rectification. In *Proc. CVPR*, 2016.

[70] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang. Scene text recognition using part-based tree-structured character detection. In *Proc. CVPR*, 2013.

[71] R. Smith. An overview of the Tesseract OCR engine. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 629–633. IEEE, 2007.

[72] B. Snyder, R. Barzilay, and K. Knight. A statistical model for lost language decipherment. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1048–1057. Association for Computational Linguistics, 2010.

[73] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *Proc. ICML*, 2015.

[74] B. Su and S. Lu. Accurate scene text recognition based on recurrent neural network. In *Proc. ACCV*, 2014.

[75] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Proc. NIPS*, pages 3104–3112, 2014.

[76] I. Sutskever, R. Jozefowicz, K. Gregor, D. Rezende, T. Lillicrap, and O. Vinyals. Towards principled unsupervised learning. In *ICLR workshop*, 2016.

[77] Tesseract OCR. https://github.com/tesseract-ocr/, 1985 – 2018.

[78] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[79] K. Wang and S. Belongie. Word spotting in the wild. In *Proc. ECCV*. Springer, 2010.

[80] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *Proc. ICCV*, pages 1457–1464. IEEE, 2011.

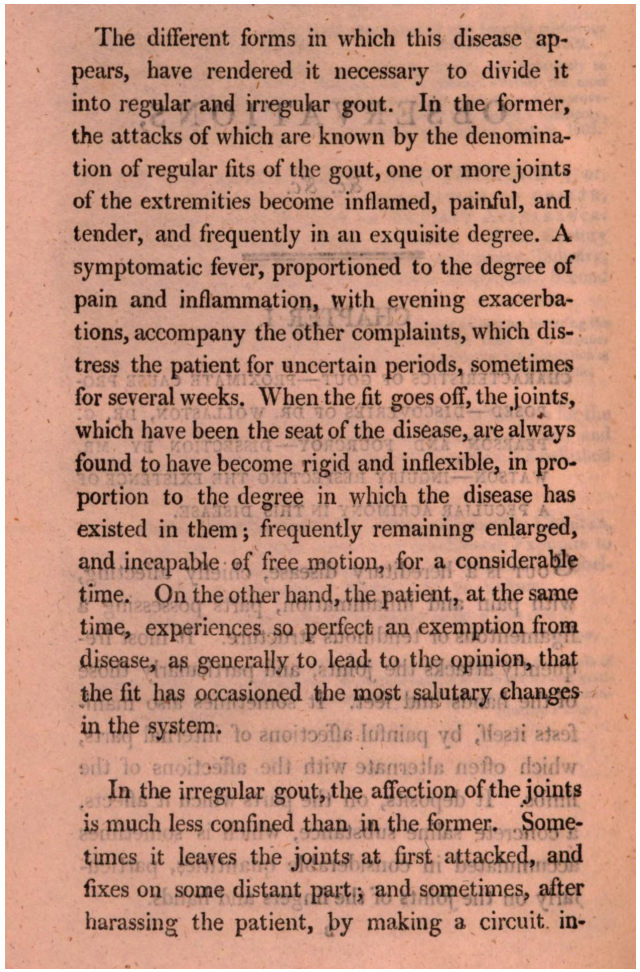
[81] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *Proc. ICPR*, pages 3304–3308. IEEE, 2012.

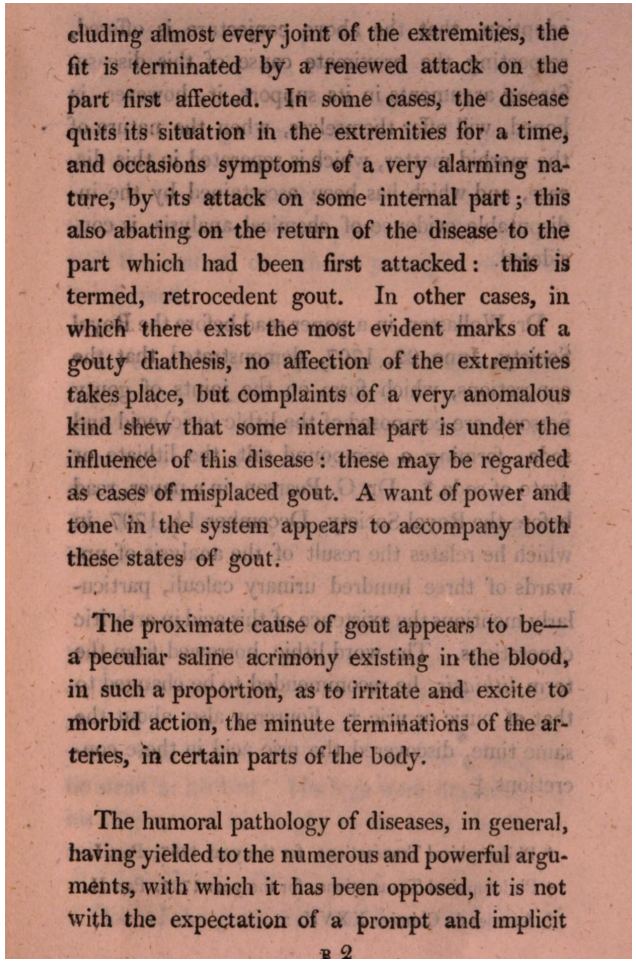
[82] C. Yao, X. Bai, B. Shi, and W. Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *Proc. CVPR*, 2014.

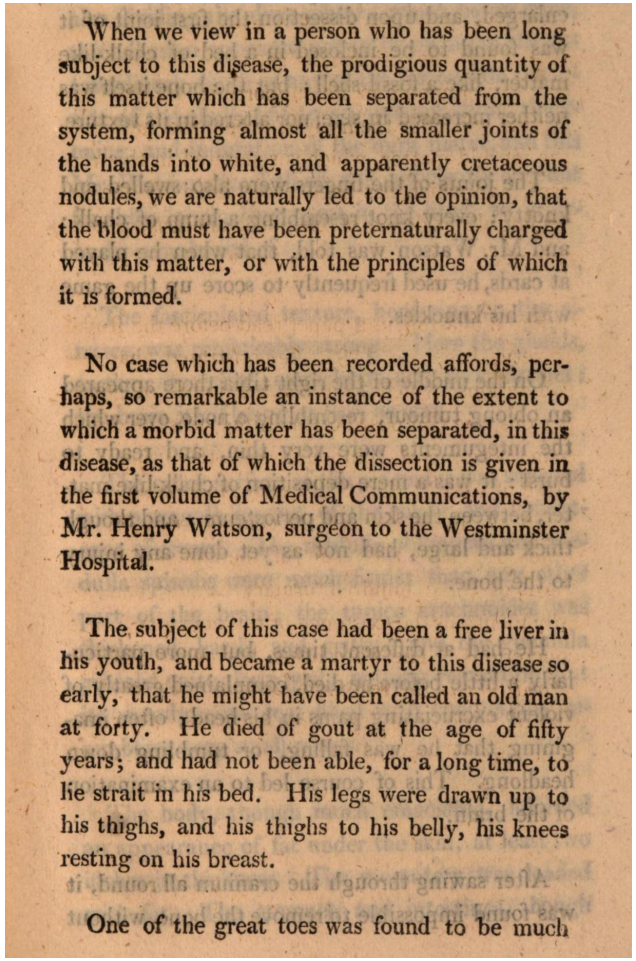
[83] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. ICCV*, 2017.

A VISUALISING REAL BOOK RECOGNITION

In the following pages, we show predictions of our method on a few samples from the *test* set of the *real* book dataset. Pages are first segmented into lines and then fed into our model for recognition (see fig. 4); full page images are shown for visual presentation only. We use the improved *skip-RNN* recognition network described in section 5.6. The "ground-truth" is not perfect, as it itself is output from Google's OCR engine. ■ denotes the <UNK> character class. We note the excellent recognition accuracy of our method, which is trained without any paired/labelled training examples.

Ground Truth	Image	Prediction
the different forms in which this disease ap pears have rendered it necessary to divide it into regular and irregular gout in the former the attacks of which are known by the denomina tion of regular fits of the gout one or more joint of the extremities become inflamed painful and tender and frequently in an exquisite degree a symptomatic fever proportioned to the degree of pain and inflammation with evening exacerba tions accompany the other complaints which dis tress the patient for uncertain periods sometimes for several weeks when the fit goes off the joints which have been the seat of the disease are always found to have become rigid and inflexible in pro portion to the degree in which the disease has existed in them■ frequently remaining enlarged and incapable of free motion for a considerable time on the other hand the patient at the same time experiences so perfect an exemption from disease ag generallyto leadu to the opinion that the fit has occasioned the most salutary changes in the system ■ ■■■ ■ ■i■■		the different forms in which this disease ap pears have rendered it necessary to divide it into regular and aregular gout in the former the attacks of which are known by the denomina tion of regular fits of the got onne o more joints of the extremities become inflamed painful and tender and frequently in an exquisite deqree a symptoneatic fever proportioned to the degree of pain and inflammation with evening exacerba tions accompand the other complaints which dis tress the patient for uncertain perions sometimes for several weeks wo he the fit goes off the piont which have been the seat of the disease are always found to have become rigid and inflexible in pro portion to the degree in which the disease has existed in themuffeequently remaining enlarged and incapable of free motion for a considerable timer cen he other hand the patient at the sas time experiences so perfect an exemption from disease as generally to lead to the opinion that the fit has occasioned the most salutary changes in the sten in ■ne■ e erd y i er
in the irregular gout the affection of the joints is much less confined than in the former yiomeu timcs it leaves the joints at first attacked and fixes on some distant part■ and sometimes after harassing the patient by making a circuit in■		in the irregular gout the affection of the joints is much less confined than in the former some e times it leaves the joints at fort tttached and fixes on some distant part■ and sometimes after harassing the patient by making a circuit in■

Ground Truth	Image	Prediction
eluding almost every joint of the extremities fit is terminated by a renewed attack on the part first affected in some cases the disease qrt its its situation in the extremities for a tim and occasions symptoms of a very alarming na■ tnre by its attack oh some internal part■ this also abating on the return of the disease to the part which had been first attacked■ this is termed retrocedent gout in other cases in which■ there exist the most evident marks of a gouty diathesis no affection of the extremities takes place but complaints of a very anomalous kind shew that some internal part is under the influence of this disease u these may be regarded as cases of misplaced gout a want of power and tone in the system appears to accompany both these states of gout		cluding almost every joint of the extremities fit is terminated by a renewed attack on the part forst affected in some cases the disease quits its situation in tee extremities for a time and occasions symptoms of a very alarming na■ ture by its attack on some internal part■ this also abating on the return of the disease to the part which had bete ffrst attacked■ this is termed retrocedent gout in other cases in which there exist the most evident marks of a gouty diathesis no affection of the extremities takes place but complaints of a very anomalous kind shew that some internal part is under the influenee of this disease■ theese ay be regarded as cases of misplaced gout a ant of power and tone in the system appears to accompany both these states of gout
the proximate cause of gout appears to be■ a peculiar saline acrimony existing in the blood in such a proportion as to irritate and excite to morbid action the minute terminations of the ar teries in certain parts of the body		the proximate cause of gout appears to be■ a peculiar saline acrimony existing in the blood in such a proportion as to irritate and excite to morbid action the minete terminations of the ar teries in certain parts of the boby
the humoral pathology of diseases in general having yielded to the numerous and powerful arg■ ments with which it has been opposed it is not with the expectation of a prompt and implicit		the humeoral pathology of diseases in deeerar having yielded to the numerous and powerful arg■ ments with which it has been opposed it is not with the expectation of a prompt and implicit

Ground Truth	Image	Prediction
when we view in a person who has been long subject to this disease the prodigious quantity of this matter which has been separated from the system forming almost all the smaller joints of the hands into white and apparently cretaceous nodules we are naturally led to the opinion that the blood must have been preternaturally charged with this matter or with the principles of which it is formed	 When we view in a person who has been long subject to this disease, the prodigious quantity of this matter which has been separated from the system, forming almost all the smaller joints of the hands into white, and apparently cretaceous nodules, we are naturally led to the opinion, that the blood must have been preternaturally charged with this matter, or with the principles of which it is formed.	w en we view in a person who has been long subject to this disease the prodigious quantity of this matter which has been separated from the system formeng galmost all the smaller joints of the hands into white and apparently cretaceous nodules we are naturally led to the opinion that the blood must have been preternaturally charged with this matter or with the principles of which it is formed
no case which has been recorded affords per haps so remarkable an instance of the extent to which a morbid matter has been separated in this disease as that of which the dissection is given i the first volume of medical communications by mr henry watson surgeon to the westminster hospital	No case which has been recorded affords, perhaps, so remarkable an instance of the extent to which a morbid matter has been separated, in this disease, as that of which the dissection is given in the first volume of Medical Communications, by Mr. Henry Watson, surgeon to the Westminster Hospital.	be case which has been recorded affords per haps so remartakbe an instance of the extent to which a morbid matter has been separated in this disease as that of which the dissection is diven i the furt solume of medicall communications by krr hency watton surgean to the westminster hospital
the subject of this case had been a free liver ia his youth and became a martyr to this disease so early that he might have been called an old man at forty he died of gout at the age of fifty years■ and had not been able for a long time to lie strait in his bed his legs were drawn up to his thighs and his thighs to his belly his knees resting on his breast	The subject of this case had been a free liver in his youth, and became a martyr to this disease so early, that he might have been called an old man at forty. He died of gout at the age of fifty years; and had not been able, for a long time, to lie strait in his bed. His legs were drawn up to his thighs, and his thighs to his belly, his knees resting on his breast.	the su b ect of this case had been a free liver in his youth and becanme a martyy to this disease so early that he might have been called an old man at forty ind dicd of gout at the age of fift years■ and had not been able for a long time to lie striitiin is bed his legs were drawn up to his thighs and his thighs to his belly his knees resting on his breast
one of the great toes was found to be much	One of the great toes was found to be much	cne of the great toes was found to be much

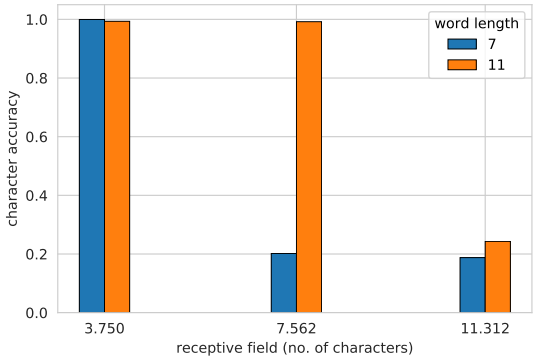
B SENSITIVITY TO RECEPTIVE FIELD SIZE

We examine the impact of the receptive-field size of the recognition network on convergence. For this, we trained three models with filter sizes {3×3, 5×5, 7×7} pixels in all the layers of the recognition network. This corresponds to receptive-fields of {61×61, 121×121, 181×181} pixels in the input image respectively. As the average width of each character is about 16 pixels, this translates to a receptive-fields of {3.75, 7.56, 11.31} characters for the three settings; filter sizes and the corresponding receptive-fields in pixels and character are summarised in the table 2 below.

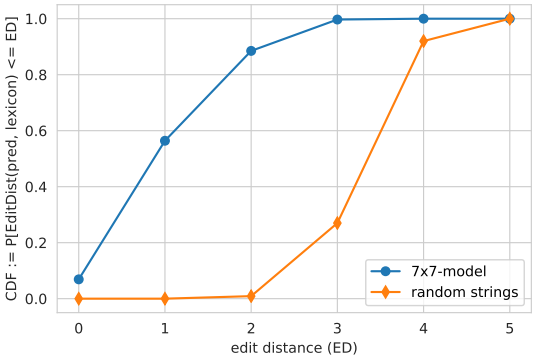
filter size	3×3	5×5	7×7
pixel receptive-field	61×61	121×121	181×181
character receptive-field	3.75	7.56	11.31

Table 2: Receptive field sizes in pixels in the input image, and number of characters (16-pixels per character), for three models with progressively larger filters.

We trained the three models on two synthetic datasets, containing words with 7 and 11 characters respectively. Figure 13a shows the *character accuracies* so obtained. We note that it is crucial to have a *receptive-field which does not cover the entire length of the word image*, i.e. be *local* (see section 3/*Grounding*) for convergence. Concretely, the model with a receptive-field of 7.56-characters does not converge for length-7 words, but does for length-11 words. The model with small receptive-field of 3.75-characters converges for both 7,11-length words, while the larger model with 11.31-characters receptive-field does not converge for the two lengths.



(a) Recognition network receptive-field sensitivity. *Character accuracy* for three models with receptive-fields of {3.75, 7.56.11.31}-characters on datasets containing words with 7 and 11 characters. The recognition network needs to be *local*, *i.e.* unable to synchronise its output over the entire word-length for successful convergence.



(b) Cumulative Density Function (CDF) of the MinEditDistances. *MinEditDistances*-CDF of the 7×7-model (11.31-characters receptive-field) on 7-length words images. This model fails to converge (recognition accuracy ≈19% – see fig. 13a) yet produces strings that look like valid strings (have small edit-distance to real strings). Distribution of random strings of length-7 is given as a baseline.

Figure 13: Sensitivity to receptive-field size.

Models with large receptive-fields fail to converge because they can synchronise the character predictions across the full sequence length, and hence generate valid text-strings to fool the discriminator without decoding the input image. To confirm this we plotted the distribution of *MinEditDistances* of the strings predicted by the 7×7-model (11.31-characters receptive-field) on length-7 words in fig. 13b. Where, the *MinEditDistance* for a given predicted string is the *minimum* edit-distance to any word in the lexicon of valid strings used to train the discriminator. We note that the ≈57% of the predictions are within 1 edit-distance (as compared to 0% for random strings of length-7 generated by sampling characters uniformly at random). This confirms that the "recognition" network in this case learns to fool the discriminator without recognising the input (the character recognition accuracy for this model is ≈19% – see fig. 13a) .

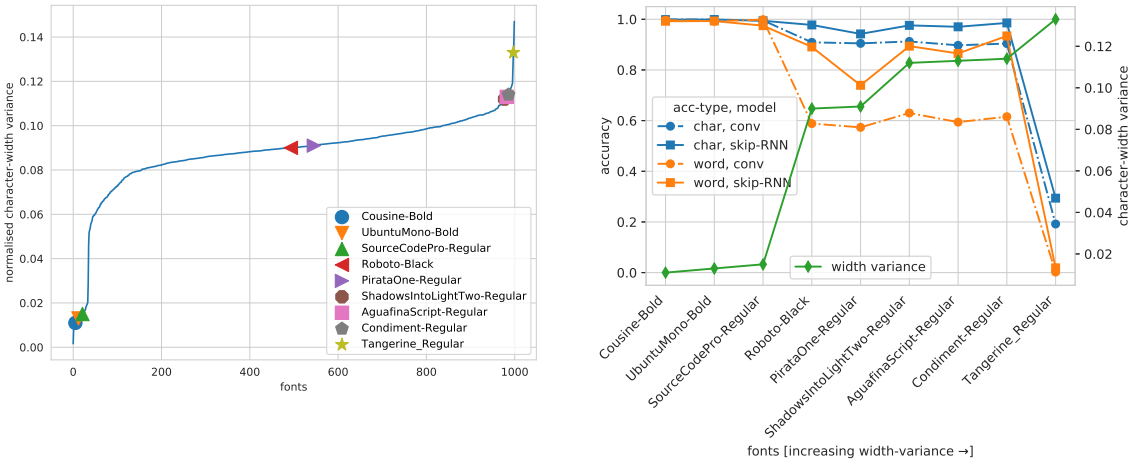
C SENSITIVITY TO VARIANCE IN CHARACTER-WIDTHS IN DIFFERENT FONTS

We examine the impact of the variance in the character-widths on convergence, by training our models on fonts of increasing character-width variance. Varying character widths influence the position of the glyphs in the text-image: fixed-width fonts have constant character widths, which means the glyphs are always positioned at the same position in the image, making the task of segmentation and recognition for the network easier. On the other hand, positions of glyphs is not constant for non fixed-width fonts, which poses a challenge for the recognition network.

The position of characters is influenced by two factors — (1) widths of the preceding characters and hence changes if the font is not fixed-width, and (2) the kerning (*i.e.* spacing between characters). To get a single measure of variance due to these two factors, we defined the *Normalised Character-width Variance (NCV)* of a font f as the following:

$$\text{NCV}(f) = \underset{w\sim\mathcal{L}}{\frac{\sigma}{\mu}}\left(\frac{\text{pixel-width}(f[w])}{\text{\#-characters}(w)}\right)$$

where, $\frac{\sigma}{\mu}$ is the *coefficient-of-variation* operator $\left(i.e., \frac{\sigma}{\mu}(\cdot) = \frac{\sigma(\cdot)}{\mu(\cdot)}\right)$, w is a word sampled from the lexicon $\mathcal{L}$, $f[w]$ is the image of the word w rendered in font f, and #-characters(w) is the number of characters in w. We used randomly chosen $10k$ words from the lexicon of $90k$ words defined in [34] for the lexicon $\mathcal{L}$ above. Then we ranked all the 1000 fonts from [34] by this measure, and selected 9-fonts with `{low, medium, high}` values of *NCV*; the variance of various fonts, and the selected fonts are visualised in fig. 14a. We selected only 9 fonts for this study due to limited resources. Note the fonts with low-variance are fixed-width fonts, *viz.* `{Cousine-Bold, UbuntuMono-Bold, SourceCodePro-Regular}`, while `Tangerine_Regular` with the highest variance is a highly stylised font; the rest of the fonts have varying levels of non fixed-width characters. Figure 15 visualises the glyphs from these nine fonts.



(a) Normalised Character-width Variance (NCV). Character-width variance for 1000 fonts and the 9 fonts with `{low, medium, high}`-variance used in this experiment.

(b) Performance on different fonts. *Word* and *character* accuracies for the *fully-convolutional* and *skip-RNN* models. The accuracies drop with increase in character-width variance. *skip-RNN* is more robust to such variations.

Figure 14: Sensitivity to variance in character-widths.

We train two models — (1) fully-convolutional, and (2) *skip-RNN* (see fig. 12) on different synthetic datasets of rendered in the nine fonts, with images (and text-strings) containing up to 24 characters. Figure 14b visualises the *character* and *word* accuracies of the models. We note that the model is indeed sensitive to the variance in character positions/widths. Accuracy drops with increasing variance — *character* accuracy from ≈100% for `low`-fonts with $NCV \leq 0.015$, to ≈90% for `mid`-fonts with $0.09 \leq NCV \leq 0.114$; the model does not converge for the highly stylised `Tangerine_Regular` font ($NCV = 0.133$). This is also reflected in the word-accuracies where the drop is more dramatic. However, the *skip-RNN* model is able to recover the performance for the `mid`-variance fonts (*word* accuracy: ≈95% vs. ≈60% for the fully-convolutional model), but still fails on `Tangerine_Regular`. This mirrors our results on the real *Google1000* books dataset, where the *skip-RNN* model achieved similar dramatic gains in performance (section 5.6).

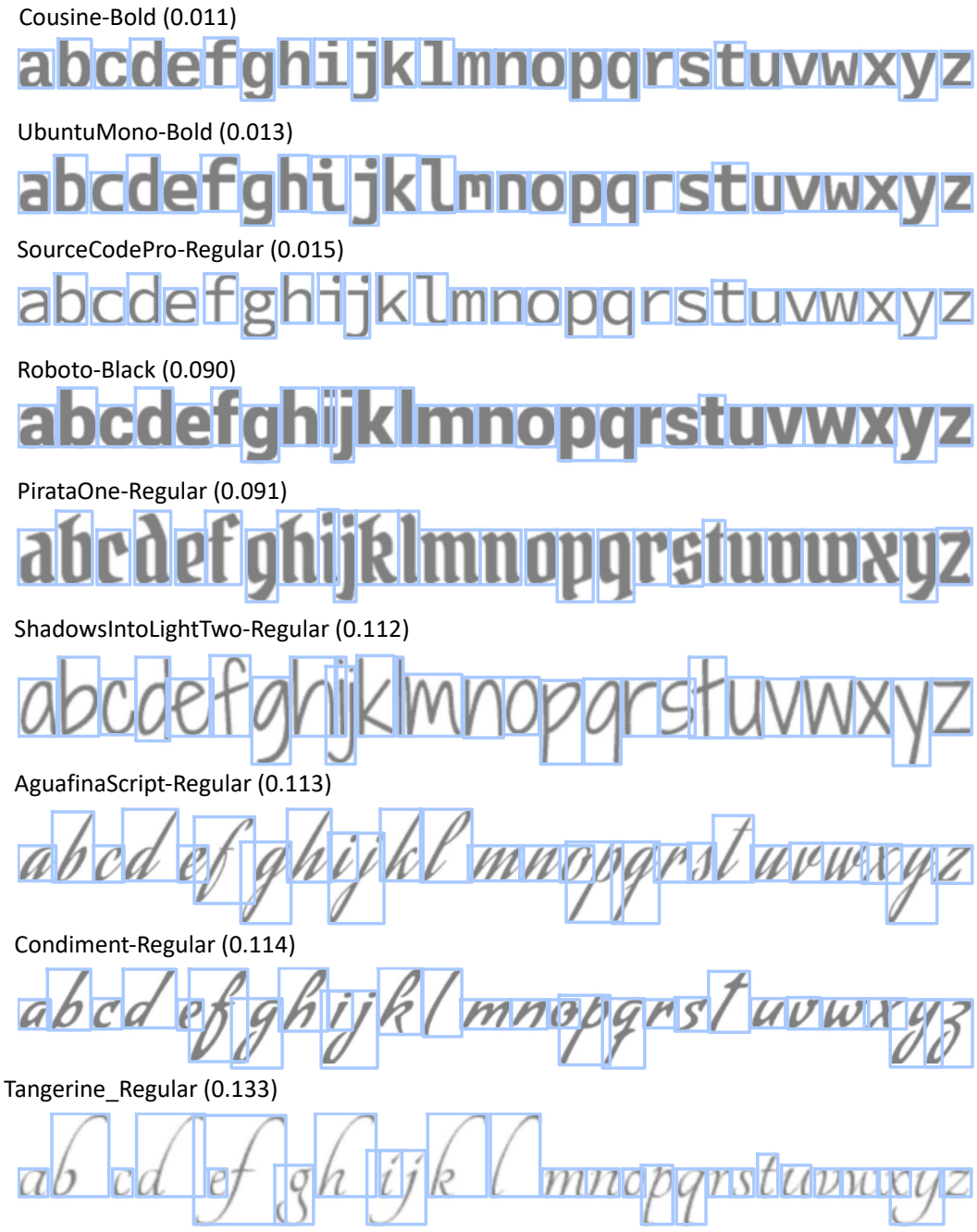
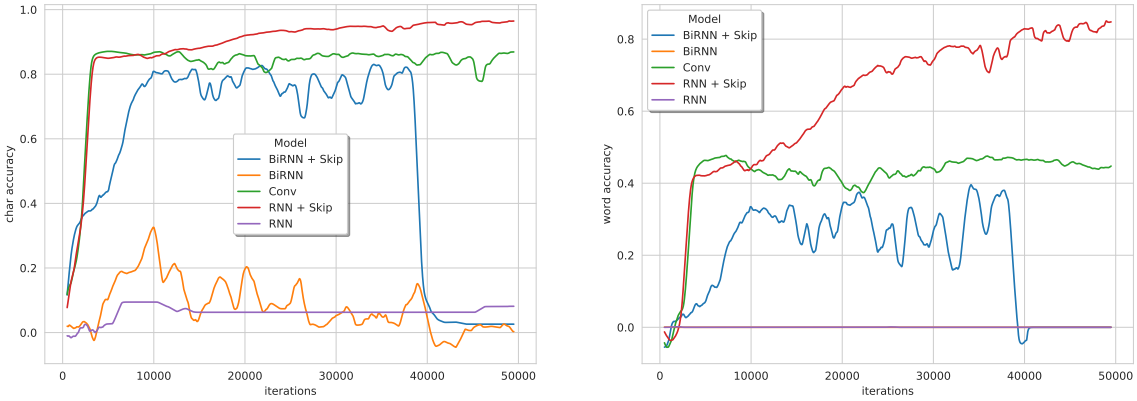


Figure 15: Fonts selected for the experiment. Fonts with increasing NCV variance (in parenthesis).

D RECURRENT ARCHITECTURES ABLATION STUDY

In this experiment we examine if similar gains in performance as obtained by using *skip-RNN* in recognising *real* book scans (section 5.6) can be obtained through other recurrent architectures. To this end, we examine the following five models — (1) fully-convolutional (no recurrent layers) (*Conv*), (2) recurrent neural network (*RNN*), (3) recurrent network with skip-connection (*skip-RNN* or *RNN+Skip*), (4) bi-directional RNN (*BiRNN*), and (5) bi-directional RNN with skip-connection (*BiRNN+Skip*). In all of the recurrent models, the penultimate layer of fully-convolutional model is augmented with *one* layer of the corresponding architecture. LSTM-cell is used for all the recurrent models. As in section 5.6, the output from the previous step is *not* fed back into the recurrent unit (see fig. 12 for an illustration).

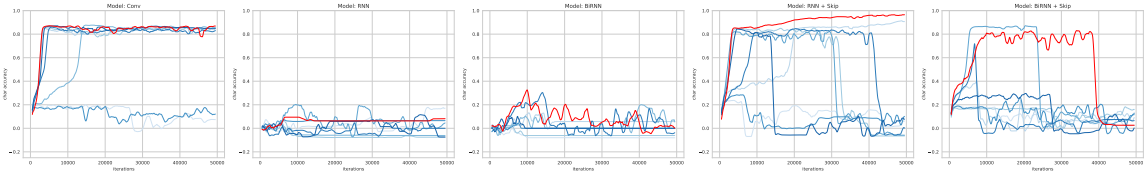
We train all the models on the real book dataset from section 5.6 as this is where *skip-RNN* had advantage over the *Conv* model. Due to instabilities in training GANs, we train on each model 8 times for 50k iterations (due to limited budget), and report the accuracy of the best performing run; training curves from all eight runs for each model are visualised in fig. 17 for completeness.



(a) **(b)**

Figure 16: Recurrent architectures ablation study. *Character* **[left]**and *word* **[right]** accuracies for various models on the *real* book scan dataset from section 5.6. *RNN+Skip* is the best performing model, followed by *Conv* and then *BiRNN+Skip*. Recurrent models without skip-connections, *i.e. RNN* and *BiRNN* fail to take-off.

Figure 16 visualises the best performing curves of each model (in terms of *character* and *word* accuracies). As observed in section 5.6, *skip-RNN* (best model) outperforms *Conv* (second best) model by a large margin (*word* accuracy: ≈85% vs. ≈45%). Recurrent architectures without skip-connections, *i.e. RNN* and *BiRNN* both fail to take-off. This is possibly because the gradients from weak supervision are likely not strong enough to train these recurrent units. Further, *BiRNN+Skip* does start to learn, but performs worse than *Conv*. It is highly unstable to train, and the training collapses mid-training. In our preliminary examination of the predictions generated by *BiRNN+Skip*, *RNN* and *BiRNN*, do not seem to be "valid" sentences which can fool the discriminator (unlike the predictions from large receptive-field fully-convolutional models in appendix B); the training fails to take-off or collapses producing nonsensical predictions. This is likely tied to insufficient signal in the gradients from the weak supervision to train these recurrent units.

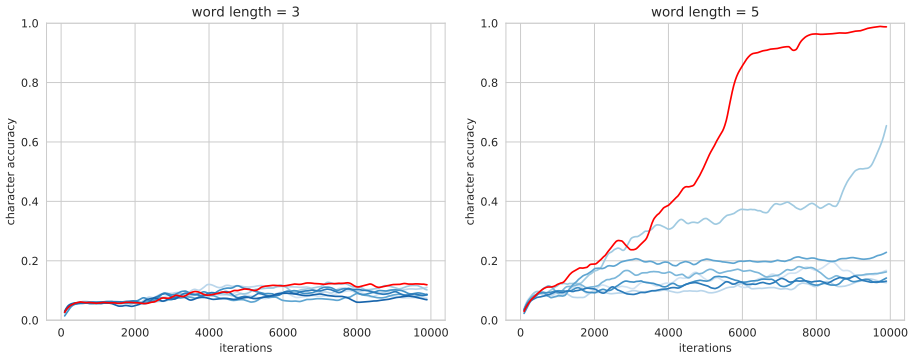


(a) Conv **(b) RNN** **(c) BiRNN** **(d) RNN+Skip** **(e) BiRNN+Skip**

Figure 17: All runs for the various architectures. Due to instabilities in training GANs, *eight* runs for each model were launched, for 50k iterations. The best-performing runs (highest AUC) are highlighted in red and reproduced in fig. 16a. *Character* accuracy is reported.

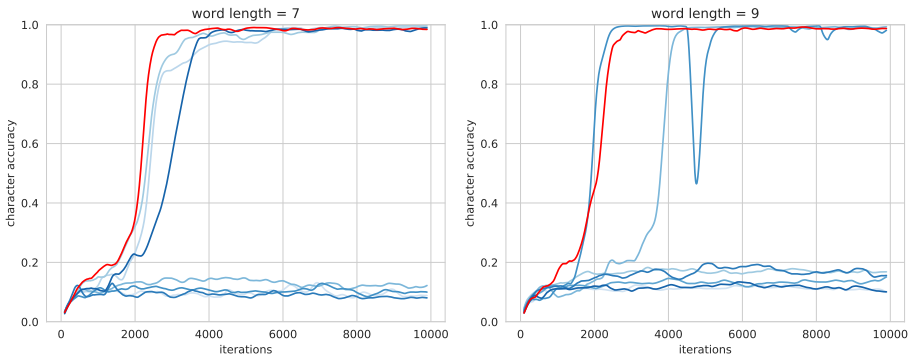
E ALL RUNS FROM THE EXPERIMENT ON THE EFFECT OF TEXT LENGTH ON CONVERGENCE

In section 5.2 we examined the effect of text length on the convergence speed. We found that longer sequences converge faster. Due to instabilities in training GANs, eight training runs for each length — `{3,5,7,9,11,13 }` were conducted, which are reported here for completeness. Due to high-variance in the runs a larger-scale study with more numbers of runs could benefit this analysis; this is unfortunately not feasible with our current resources.



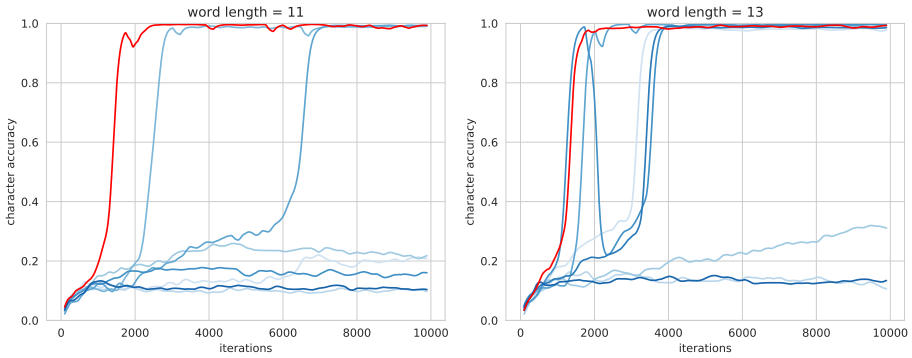
(a) length-3

(b) length-5



(c) length-7

(d) length-9



(e) length-11

(f) length-13

Figure 18: All runs from experiments on the effect of text length on convergence speed. Due to instabilities in training GANs, *eight* runs for each model were launched, for 50k iterations. The best-performing runs (highest AUC / earliest "take-off") in terms of *character accuracy* are highlighted in red and reported in fig. 7.

F GROUNDING PREDICTIONS THROUGH IMAGE RECONSTRUCTION

Our method is similar to *CycleGAN* [83] in that it learns to translate between two domains using only unaligned data. Following *CycleGAN*, to ground the predicted characters in the input image, we employed an image-reconstruction objective to complete the cycle, *i.e.* image → characters → image.

As the character → image mapping is highly ambiguous, we assumed *known* character-level segmentation in the image, to simplify the setting. Hence, the i^{th} (out of the total K) predicted symbol was associated with the i^{th} character image region in the input image. The predicted softmax vector was fed into a *rendering network*[1] which was tasked to reconstruct the associated character-image using a perceptual/content loss [17, 18]. Figure 19 below visualises ground-truth input images, and reconstructions produced by the network. Note, the images were segmented into individual *character*-level patches; these patches are placed side-by-side as complete word images here (image discontinuities are visible at boundaries between the characters).

Figure 19: Grounding predictions through image reconstruction. Following *CycleGAN* our first implementation involved a reconstruction objective to ground the predicted characters in the input image. Visualised here are three samples: **[top]:** ground-truth input image; **[bottom]:** reconstructions produced by the network.

The recognition network achieved perfect accuracy in this simple setting of synthetic images with a fixed-width font. Note in the images above, the rendering network learns to reconstruct the individual characters well, but ignores other details like noise in the background. This is because, this information is not represented in the softmax vector which is input to the rendering network. In fact, this is the *key limitation* of this approach, in that the characters → image mapping is highly ambiguous and is not amenable to pixel-level reconstruction in the most general setting (different fonts, varying glyph locations, background noise *etc.*). We found, even in the absence of this reconstruction objective, limiting the receptive-field is sufficient to ground the predictions in the input image (see appendix B), and hence we discarded image reconstruction to bypass this limitation.

[1] architecture of the rendering network: four (3×3-conv)-ReLU-(2×-bilinearUpsampling) layers, followed by a final 3×3-conv without any non-linearity.

5

Unsupervised Learning of Object Landmarks through Conditional Image Generation

This work was presented at the 32nd Conference on Neural Information Processing Systems (NIPS), 2018.

Unsupervised Learning of Object Landmarks through Conditional Image Generation

Tomas Jakab[1*] **Ankush Gupta**[1*] **Hakan Bilen**[2] **Andrea Vedaldi**[1]

[1] Visual Geometry Group
University of Oxford
{tomj,ankush,vedaldi}@robots.ox.ac.uk

[2] School of Informatics
University of Edinburgh
hbilen@ed.ac.uk

Abstract

We propose a method for learning landmark detectors for visual objects (such as the eyes and the nose in a face) without any manual supervision. We cast this as the problem of generating images that combine the appearance of the object as seen in a first example image with the geometry of the object as seen in a second example image, where the two examples differ by a viewpoint change and/or an object deformation. In order to factorize appearance and geometry, we introduce a tight bottleneck in the geometry-extraction process that selects and distils geometry-related features. Compared to standard image generation problems, which often use generative adversarial networks, our generation task is conditioned on both appearance and geometry and thus is significantly less ambiguous, to the point that adopting a simple perceptual loss formulation is sufficient. We demonstrate that our approach can learn object landmarks from synthetic image deformations or videos, all without manual supervision, while outperforming state-of-the-art unsupervised landmark detectors. We further show that our method is applicable to a large variety of datasets — faces, people, 3D objects, and digits — without any modifications.

1 Introduction

There is a growing interest in developing machine learning methods that have little or no dependence on manual supervision. In this paper, we consider in particular the problem of learning, without external annotations, detectors for the landmarks of object categories, such as the nose, the eyes, and the mouth of a face, or the hands, shoulders, and head of a human body.

Our approach learns landmarks by looking at images of deformable objects that differ by acquisition time and/or viewpoint. Such pairs may be extracted from video sequences or can be generated by randomly perturbing still images. Videos have been used before for self-supervision, often in the context of future frame prediction, where the goal is to generate future video frames by observing one or more past frames. A key difficulty in such approaches is the high degree of ambiguity that exists in predicting the motion of objects from past observations. In order to eliminate this ambiguity, we propose instead to condition generation on two images, a source (past) image and a target (future) image. The goal of the learned model is to reproduce the target image, given the source and target images as input. Clearly, without further constraints, this task is trivial. Thus, we pass the target through a tight bottleneck meant to *distil the geometry of the object* (fig. 1). We do so by constraining the resulting representation to encode spatial locations, as may be obtained by an object landmark detector. The source image and the encoded target image are then passed to a generator network which reconstructs the target. Minimising the reconstruction error encourages the model to learn landmark-like representations because landmarks can be used to encode the *geometry* of the object,

[*] equal contribution.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

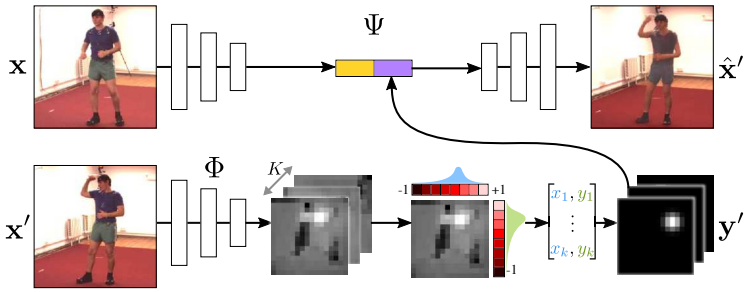


Figure 1: **Model Architecture.** Given a pair of source and target images $(\mathbf{x}, \mathbf{x}')$, the pose-regressor Φ extracts K heatmaps from $\mathbf{x}'$, which are then marginalized to estimate coordinates of keypoints, to limit the information flow. 2D Gaussians $(\mathbf{y}')$ are rendered from these keypoints and stacked along with the image features extracted from $\mathbf{x}$, to reconstruct the target as $\Psi(\mathbf{x}, \mathbf{y}') = \hat{\mathbf{x}}'$. By restricting the information-flow our model learns to detect consistent object keypoints, without any annotations.

which changes between source and target, while the appearance of the object, which is constant, can be obtained from the source image alone.

The key advantage of our method, compared to other works for unsupervised learning of landmarks, is the simplicity and generality of the formulation, which allows it to work well on data far more complex than previously used in unsupervised learning of object landmarks, *e.g*. landmarks for the highly-articulated human body. In particular, unlike methods such as [47, 46, 57], we show that our method can learn from synthetically-generated image deformations as well as raw videos as it *does not* require access to information about correspondences, optical-flow, or transformation between images.

Furthermore, while image generation has been used extensively in unsupervised learning, especially in the context of (variational) auto-encoders [23] and Generative Adversarial Networks (GANs [13]; see section 2), our approach has a key advantage over such methods. Namely, conditioning on both source and target images simplifies the generation task considerably, making it much easier to learn the generator network [18]. The ensuing simplification means that we can adopt the direct approach of minimizing a perceptual loss as in [10], without resorting to more complex techniques like GANs. Empirically, we show that this still results in excellent image generation results and that, more importantly, geometrically consistent landmark detectors are learned without manual supervision (section 4). Project code and details are available at: `http://www.robots.ox.ac.uk/~vgg/research/unsupervised_landmarks/`

2 Related work

The recent approaches of [47, 46] learn to extract landmarks based on the principles of equivariance and distinctiveness. In contrast to our work, these methods are not generative. Further, they rely on known correspondences between images obtained either through optical flow or synthetic transformations, and hence, cannot leverage video data directly. Since the principle of equivariance is orthogonal to our approach, it can be incorporated as an additional cue in our method.

Unsupervised learning of representations has traditionally been achieved using auto-encoders and restricted Boltzmann machines [14, 49, 15]. InfoGAN [6] uses GANs to disentangle factors in the data by imposing a certain structure in the latent space. Our approach also works by imposing a latent structure, but using a *conditional*-encoder instead of an auto-encoder.

Learning representations using conditional image generation via a bottleneck was demonstrated by Xue *et al*. [54] in variational auto-encoders, and by Whitney *et al*. [52] using a discrete gating mechanism to combine representations of successive video frames. Denton *et al*. [8] factor the pose and identity in videos through an adversarial loss on the pose embeddings. We instead design our bottleneck to explicitly shape the features to resemble the output of a landmark detector, without any adversarial training. Villegas *et al*. [48] also generate future frames by extracting a representation of appearance and human pose, but, differently from us, require ground-truth pose annotations. Our method essentially *inverts* their analogy network [38] to output landmarks given the source and target image pairs.

Several other generative methods [44, 42, 39, 50, 34] focus on video extrapolation. Srivastava *et al*. [42] employ Long Short Term Memory (LSTM) [16] networks to encode video sequences into a fixed-length representation and decode it to reconstruct the input sequence. Vondrick *et al*. [50] propose a GAN for videos, also with a spatio-temporal convolutional architecture that disentangles foreground and background to generate realistic frames. Video Pixel Networks [20] estimate the discrete joint distribution of the pixel values in a video by encoding different modalities such as time, space and colour information. In contrast, we learn a *structured embedding* that explicitly encodes the spatial location of object landmarks.

A series of concurrent works propose similar methods for unsupervised learning of object structure. Shu *et al*. [40] learn to factor a single object-category-specific image into an appearance template in a canonical coordinate system, and a deformation field which warps the template to reconstruct the input, as in an auto-encoder. They encourage this factorisation by controlling the size of the embeddings. Similarly, Wiles *et al*. [53] learn a dense deformation field for faces but obtain the template from a second related image, as in our method. Suwajanakorn *et al*. [45] learn 3D-keypoints for objects from two images which differ by a known 3D transformation, by enforcing equivariance [47]. Finally, the method of Zhang *et al*. [57] shares several similarities with ours, in that they also use image generation with the goal of learning landmarks. However, their method is based on generating a single image from *itself* using landmark-transported features. This, we show is insufficient to learn geometry and requires, as they do, to also incorporate the principle of equivariance [47]. This is a key difference with our method, as ours results in a much simpler system that does *not* require to know the optical-flow/correspondences between images, and can learn from raw videos directly.

3 Method

Let $\mathbf{x}, \mathbf{x}' \in \mathcal{X} = \mathbb{R}^{H\times W\times C}$ be two images of an object, for example extracted as frames in a video sequence, or synthetically generated by randomly deforming $\mathbf{x}$ into $\mathbf{x}'$. We call $\mathbf{x}$ the source image and $\mathbf{x}'$ the target image and we use Ω to denote the image domain, namely the $H\times W$ lattice.

We are interested in learning a function $\Phi(\mathbf{x}) = \mathbf{y} \in \mathcal{Y}$ that captures the "structure" of the object in the image as a set of K object landmarks. As a first approximation, assume that $\mathbf{y} = (u_1, \dots, u_K) \in \Omega^K = \mathcal{Y}$ are K coordinates $u_k \in \Omega$, one per landmark.

In order to learn the map Φ in an unsupervised manner, we consider the problem of conditional image generation. Namely, we wish to learn a generator function

$$\Psi : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}, \qquad (\mathbf{x}, \mathbf{y}') \mapsto \mathbf{x}'$$

such that the target image $\mathbf{x}' = \Psi(\mathbf{x}, \Phi(\mathbf{x}'))$ is reconstructed from the *source image* $\mathbf{x}$ and the *representation* $\mathbf{y}' = \Phi(\mathbf{x}')$ of the *target image*. In practice, we learn both functions Φ and Ψ jointly to minimise the expected reconstruction loss $\min_{\Psi,\Phi} E_{\mathbf{x},\mathbf{x}'}\left[\mathcal{L}(\mathbf{x}', \Psi(\mathbf{x}, \Phi(\mathbf{x}')))\right]$. Note that, if we do not restrict the form of $\mathcal{Y}$, then a trivial solution to this problem is to learn identity mappings by setting $\mathbf{y}' = \Phi(\mathbf{x}') = \mathbf{x}'$ and $\Psi(\mathbf{x}, \mathbf{y}') = \mathbf{y}'$. However, given that $\mathbf{y}'$ has the "form" of a set of landmark detections, the model is strongly encouraged to learn those. This is explained next.

3.1 Heatmaps bottleneck

In order for the model $\Phi(\mathbf{x})$ to learn to extract keypoint-like structures from the image, we terminate the network Φ with a layer that forces the output to be akin to a set of K keypoint detections. This is done in three steps. First, K heatmaps $S_u(\mathbf{x}; k), u \in \Omega$ are generated, one for each keypoint $k = 1, \dots, K$. These heatmaps are obtained in parallel as the channels of a $\mathbb{R}^{H\times W\times K}$ tensor using a standard convolutional neural network architecture. Second, each heatmap is renormalised to a probability distribution via (spatial) $\mathrm{Softmax}$ and condensed to a point by computing the (spatial) expected value of the latter:

$$u_k^*(\mathbf{x}) = \frac{\sum_{u\in\Omega} u e^{S_u(\mathbf{x};k)}}{\sum_{u\in\Omega} e^{S_u(\mathbf{x};k)}} \tag{1}$$

Third, each heatmap is replaced with a Gaussian-like function centred at u_k^* with a small fixed standard deviation σ:

$$\Phi_u(\mathbf{x}; k) = \exp\left(-\frac{1}{2\sigma^2}\|u - u_k^*(\mathbf{x})\|^2\right) \tag{2}$$

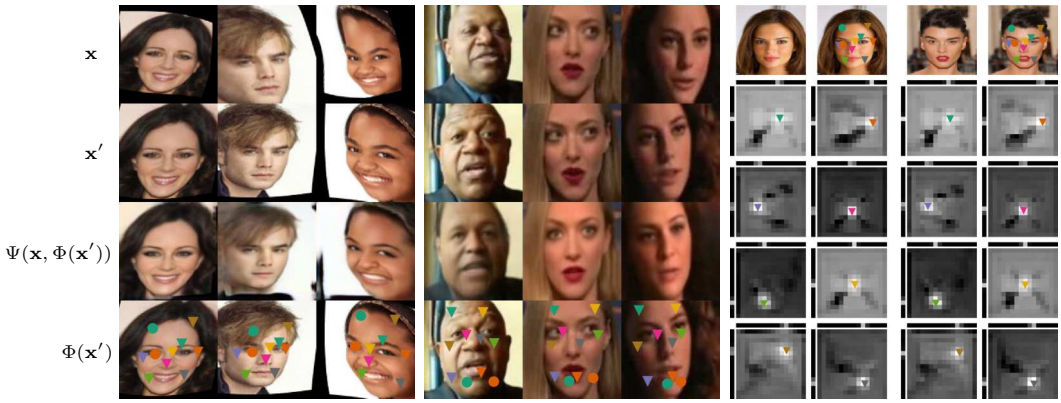


Figure 2: **Unsupervised Landmarks. [left]:** CelebA images showing the synthetically transformed source $\mathbf{x}$ and target $\mathbf{x}'$ images, the reconstructed target $\Psi(\mathbf{x}, \Phi(\mathbf{x}'))$, and the unsupervised landmarks $\Phi(\mathbf{x}')$. **[middle]:** The same for video frames from VoxCeleb. **[right]:** Two example images with selected (8 out of 10) landmarks u_k overlaid and their corresponding 2D score maps $S_u(\mathbf{x}; k)$ (see section 3.1; brighter pixels indicate higher confidence).

The end result is a new tensor $\mathbf{y} = \Phi(\mathbf{x}) \in \mathbb{R}^{H \times W \times K}$ that encodes as Gaussian heatmaps the location of K maxima. Since it is possible to recover the landmark locations exactly from these heatmaps, this representation is equivalent to the one considered above (2D coordinates); however, it is more useful as an input to a generator network, as discussed later.

One may wonder whether this construction can be simplified by removing steps two and three and simply consider $S(\mathbf{x})$ (possibly after re-normalisation, or binarization to limit the information) as the output of the encoder $\Phi(\mathbf{x})$. The answer is that these steps, and especially eq. (1), ensure that very little information from $\mathbf{x}$ is retained, which, as suggested above, is key to avoiding degenerate solutions. Converting back to Gaussian landmarks in eq. (2), instead of just retaining 2D coordinates, ensures that the representation is still utilisable by the generator network. Further, the unimodal Gaussian distribution ensures that only one location is communicated per channel.

We exploit the spatial structure of 2D convolutions in both obtaining the coordinates (by marginalising the heatmaps), and representing the encoded locations (through 2D Gaussians maps). An alternative would be to directly regress K 2D coordinates by flattening the image dimensions and learning a fully-connected regressor. However, our design is more efficient (fewer parameters).

Separable implementation. In practice, we consider a separable variant of eq. (1) for computational efficiency. Namely, let $u = (u_1, u_2)$ be the two components of each pixel coordinate and write $\Omega = \Omega_1 \times \Omega_2$. Then we set

$$u^*_{ik}(\mathbf{x}) = \frac{\sum_{u_i \in \Omega_i} u_i e^{S_{u_i}(\mathbf{x};k)}}{\sum_{u_i \in \Omega_i} e^{S_{u_i}(\mathbf{x};k)}}, \qquad S_{u_i}(\mathbf{x}; k) = \sum_{u_j \in \Omega_j} S_{(u_1,u_2)}(\mathbf{x}; k),$$

where $i = 1, 2$ and $j = 2, 1$ respectively. Figure 2 visualizes the source $\mathbf{x}$, target $\mathbf{x}'$ and generated $\Psi(\mathbf{x}, \Phi(\mathbf{x}'))$ images, as well as $\mathbf{x}'$ overlaid with the locations of the unsupervised landmarks $\Phi(\mathbf{x}')$. It also shows the heatmaps $S_u(\mathbf{x}; k)$ and marginalized separable softmax distributions on the top and left of each heatmap for $K = 10$ keypoints.

3.2 Generator network using a perceptual loss

The goal of the generator network $\hat{\mathbf{x}}' = \Psi(\mathbf{x}, \mathbf{y}')$ is to map the source image $\mathbf{x}$ and the distilled version $\mathbf{y}'$ of the target image $\mathbf{x}'$ to a reconstruction of the latter. Thus the generator network is optimised to minimise a reconstruction error $\mathcal{L}(\mathbf{x}', \hat{\mathbf{x}}')$. The design of the reconstruction error is important for good performance. Nowadays the standard practice is to learn such a loss function using adversarial techniques, as exemplified in numerous variants of GANs. However, since the goal here is not generative modelling, but rather to induce a representation $\mathbf{y}'$ of the object geometry for reconstructing a *specific* target image (as in an auto-encoder), a simpler method may suffice.

Inspired by the excellent results for photo-realistic image synthesis of [4], we resort here to use the "content representation" or "perceptual" loss used successfully for various generative networks [12,

n supervised	Thewlis [47]	Ours selfsup
1	10.82	12.89 ± 3.21
5	9.25	8.16 ± 0.96
† 10	8.49	7.19 ± 0.45
100	—	4.29 ± 0.34
500	—	2.83 ± 0.06
1000	—	2.73 ± 0.03
5000	—	2.60 ± 0.00
All (19,000)	7.15	$2.58 \pm$ N/A

Figure 3: **Sample Efficiency for Supervised Regression on MAFL. [left]:** Supervised linear regression of 5 keypoints (bottom-row) from 10 unsupervised (top-row) on MAFL test set. Centre of the white-dots correspond to the ground-truth location, while the dark ones are the predictions. Both unsupervised and supervised landmarks show a good degree of equivariance with respect to head rotation (columns 2, 4) and invariance to headwear or eyewear (columns 1, 3). **[right]:** MSE ($\pm\sigma$) (normalised by inter-ocular distance (in %)) on the MAFL test-set for varying number (n) of supervised samples from MAFL training set used for learning the regressor from 30 unsupervised landmarks. †: we outperform the previous state-of-the-art [47] with only 10 labelled examples.

1, 9, 19, 28, 31, 32]. The perceptual loss compares a set of the activations extracted from multiple layers of a deep network for both the reference and the generated images, instead of only raw pixel values. We define the loss as $\mathcal{L}(\mathbf{x}', \hat{\mathbf{x}}') = \sum_l \alpha_l \|\Gamma_l(\mathbf{x}') - \Gamma_l(\hat{\mathbf{x}}')\|_2^2$, where $\Gamma(\mathbf{x})$ is an off-the-shelf pre-trained neural network, for example VGG-19 [41], Γ_l denotes the output of the l-th sub-network (obtained by chopping Γ at layer l). As our goal is to have a purely-unsupervised learning, we pre-train the network by using a self-supervised approach, namely colorising grayscale images [26].

We also test using a VGG-19 model pre-trained for image classification in ImageNet. All other networks are trained from scratch. The parameters $\alpha_l > 0, l = 1, \ldots, n$ are scalars that balance the terms. We use a linear combination of the reconstruction error for 'input', 'conv1_2', 'conv2_2', 'conv3_2', 'conv4_2' and 'conv5_2' layers of VGG-19; $\{\alpha_l\}$ are updated online during training to normalise the expected contribution from each layer as in [4]. However, we use the ℓ_2 norm instead of their ℓ_1, as it worked better for us.

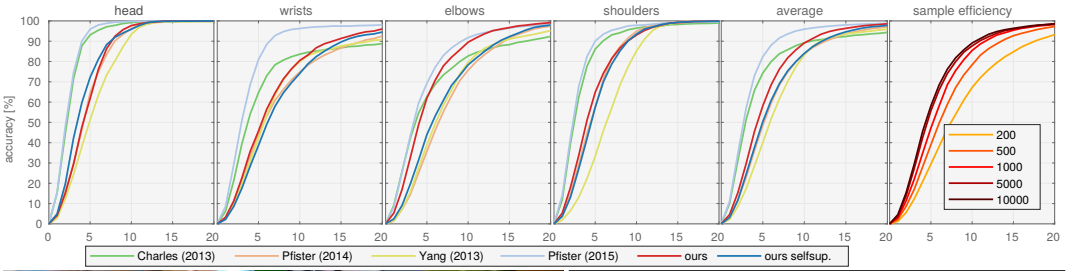
4 Experiments

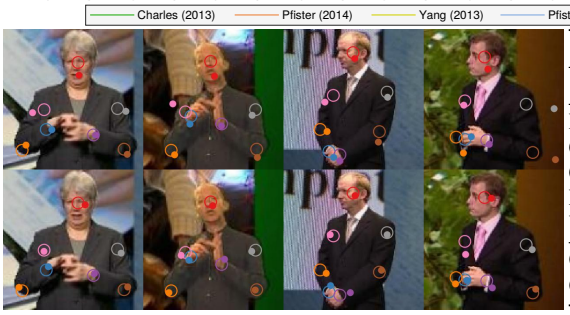
In section 4.1 we provide the details of the landmark detection and generator networks; a common architecture is used across all datasets. Next, we evaluate landmark detection accuracy on faces (section 4.2) and human-body (section 4.3). In section 4.4 we analyse the invariance of the learned landmarks to various nuisance factors, and finally in section 4.5 study the factorised representation of object style and geometry in the generator.

4.1 Model details

Landmark detection network. The landmark detector ingests the image $\mathbf{x}'$ to produce K landmark heatmaps $\mathbf{y}'$. It is composed of sequential blocks consisting of two convolutional layers each. All the layers use 3×3 filters, except the first one which uses 7×7. Each block doubles the number of feature channels in the previous block, with 32 channels in the first one. The first layer in each block, except the first block, downsamples the input tensor using stride 2 convolution. The spatial size of the final output, outputting the heatmaps, is set to 16×16. Thus, due to downsampling, for a network with $n-3, n \geq 4$ blocks, the resolution of the input image is $H\times W = 2^n \times 2^n$, resulting in $16\times16\times(32 \cdot 2^{n-3})$ tensor. A final 1×1 convolutional layer maps this tensor to a $16\times16\times K$ tensor, with one layer per landmark. As described in section 3.1, these K feature channels are then used to render $16\times16\times K$ 2D-Gaussian maps $\mathbf{y}'$ (with $\sigma = 0.1$).

Image generation network. The image generator takes as input the image $\mathbf{x}$ and the landmarks $\mathbf{y}' = \Phi(\mathbf{x}')$ extracted from the second image in order to reconstruct the latter. This is achieved in two steps: first, the image $\mathbf{x}$ is encoded as a feature tensor $\mathbf{z} \in \mathbb{R}^{16\times16\times C}$ using a convolutional network with exactly the same architecture as the landmark detection network except for the final 1×1 convolutional layer, which is omitted; next, the features $\mathbf{z}$ and the landmarks $\mathbf{y}'$ are stacked together (along the channel dimension) and fed to a regressor that reconstructs the target frame $\mathbf{x}'$.





BBC Pose Accuracy (%) at $d = 6$ pixels					
	Head	Wrsts	Elbws	Shldrs	Avg.
Pfister *et al.* [37]	98.00	88.45	77.10	93.50	88.01
Charles *et al.* [3]	95.40	72.95	68.70	90.30	79.90
Chen *et al.* [5]	65.9	47.9	66.5	76.8	64.1
Pfister *et al.* [36]	74.90	53.05	46.00	71.40	59.40
Yang *et al.* [55]	63.40	53.70	49.20	46.10	51.63
Ours (selfsup.)	81.10	49.05	53.05	70.10	60.79
Ours	76.10	56.50	70.70	74.30	68.44

Figure 4: **Learning Human Pose.** 50 unsupervised keypoints are learnt on the BBC Pose dataset. Annotations (empty circles in the images) for 7 keypoints are provided, corresponding to — head, wrists, elbows and shoulders. Solid circles represent the predicted positions; in **[fig-top]** these are raw discovered keypoints which correspond maximally to each annotation; in **[fig-bottom]** these are regressed (linearly) from the discovered keypoints. **[table]:** Comparison against *supervised* methods; %-age of points within d= 6-pixels of ground-truth is reported. **[top-row]:** accuracy-vs-distance d, for each body-part; **[top-row-rightmost]:** average accuracy for varying number of supervised samples used for regression.

The regressor also comprises of sequential blocks with two convolutional layers each. The input to each successive block, except the first one, is upsampled two times through bilinear interpolation, while the number of feature channels is halved; the first block starts with 256 channels, and a minimum of 32 channels are maintained till a tensor with the same spatial dimensions as $\mathbf{x}'$ is obtained. A final convolutional layer regresses the three RGB channels with no non-linearity. All layers use 3×3 filters and each block has two layers similarly to the landmark network.
All the weights are initialised with random Gaussian noise ($\sigma = 0.01$), and optimised using Adam [22] with a weight decay of $5 \cdot 10^{-4}$. The learning rate is set to 10^{-2}, and lowered by a factor of 10 once the training error stops decreasing; the ℓ_2-norm of the gradients is bounded to 1.0.

4.2 Learning facial landmarks

Setup. We explore extracting source-target image pairs $(\mathbf{x}, \mathbf{x}')$ using either (1) synthetic transformations, or (2) videos. In the first case, the pairs are obtained as $(\mathbf{x}, \mathbf{x}') = (g_1\mathbf{x}_0, g_2\mathbf{x}_0)$ by applying two random thin-plate-spline (TPS) [11, 51] warps g_1, g_2 to a given sample image $\mathbf{x}_0$. We use the 200k CelebA [25] images after resizing them to 128×128 resolution. The dataset provides annotations for 5 facial landmarks — eyes, nose and mouth corners, which we *do not* use for training. Following [47] we exclude the images in MAFL [59] test-set from the training split and generate synthetically-deformed pairs as in [47, 57], but the transformations themselves are not required for training. We discount the reconstruction loss in the regions of the warped image which lie outside the original image to avoid modelling irrelevant boundary artefacts.

Figure 5: **Unsupervised Landmarks on Human3.6M. [left]:** an example quadruplet source-target-reconstruction-keypoint (left to right) from Human3.6M. **[right]:** learned keypoints on a test video sequence. The landmarks consistently track the legs, arms, torso and head across frames.

In the second case, $(\mathbf{x}, \mathbf{x}')$ are two frames sampled from a video. We consider VoxCeleb [29], a large dataset of face tracks, consisting of 1251 celebrities speaking over 100k English language utterances. We use the standard training split and remove any overlapping identities which appear in the test sets of MAFL and AFLW. Pairs of frames from the same video, but possibly belonging to different utterances are randomly sampled for training. By using video data for training our models we eliminate the need for engineering synthetic data.

Qualitative results. Figure 2 shows the learned heatmaps and source-target-reconstruction-keypoints quadruplets $\langle \mathbf{x}, \mathbf{x}', \Psi\left(\mathbf{x}, \Phi(\mathbf{x}')\right), \Phi(\mathbf{x}') \rangle$ for synthetic transformations and videos. We note that the method extracts keypoints which consistently track facial features across deformation and identity changes (*e.g.*, the green circle tracks the lower chin, and the light blue square lies between the eyes). The regressed semantic keypoints on the MAFL test set are visualised in fig. 3, where they are localised with high accuracy. Further, the target image $\mathbf{x}'$ is also reconstructed accurately.

Quantitative results. We follow [47, 46] and use unsupervised keypoints learnt on CelebA and VoxCeleb to regress manually-annotated keypoints in the MAFL and AFLW [24] test sets. We freeze the parameters of the unsupervised detector network (Φ) and learn a *linear* regressor (without bias) from our unsupervised keypoints to 5 manually-labelled ones from the respective training sets. Model selection is done using 10% validation split of the training data.

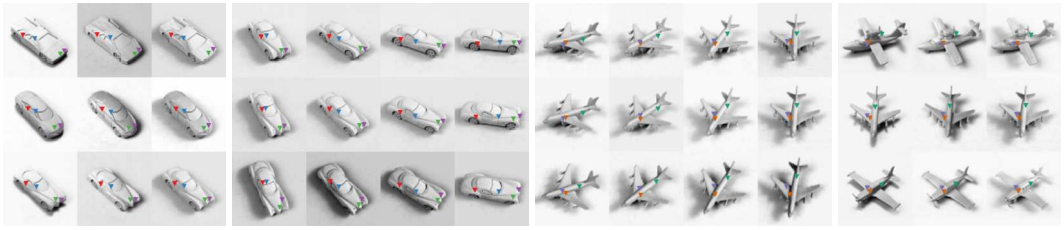
We report results in terms of standard MSE normalised by the inter-ocular distance expressed as a percentage [59], and show a few regressed keypoints in fig. 3. Before evaluating on AFLW, we finetune our networks pre-trained on CelebA or VoxCeleb on the AFLW training set. We do not use any labels during finetuning.

Sample efficiency. Figure 3 reports the performance of detectors trained on CelebA as a function of the number n of supervised examples used to translate from unsupervised to supervised keypoints. We note that $n = 10$ is already sufficient for results comparable to the previous state-of-the-art (SoA) method of Thewlis *et al.* [47], and that performance almost saturates at $n = 500$ (vs. 19,000 available training samples).

Vs. SoA. Table 1 compares our regression results to the SoA. We experiment regressing from $K{=}\{10, 30, 50\}$ unsupervised landmarks, using the self-supervised and the supervised perceptual loss networks; the number of samples n used for regression is maxed out ($= 19000$) to be consistent with previous works. On both MAFL and AFLW datasets, at 2.58% and 6.31% error respectively (for $K = 30$), we significantly outperform all the supervised and unsupervised methods. Notably, we perform better than the concurrent work of Zhang *et al.* [57] (MAFL: 3.16%; AFLW: 6.58%), while using a simpler method. When synthetic warps are removed from [57], so that the *equivariance constraint cannot be employed*, our method is significantly better (2.58% vs 8.42% on MAFL). We are also significantly better than many SoA *supervised* detectors [56, 43, 59] using only $n = 100$ supervised training examples, which shows that the approach is very effective at exploiting the unlabelled data. Finally, training with VoxCeleb video frames degrades the performance due to domain gap; including a bias in the linear regressor improves the performance.

Method	K	MAFL	AFLW
Supervised			
RCPR [2]		–	11.60
CFAN [56]		15.84	10.94
Cascaded CNN [43]		9.73	8.97
TCDCN [59]		7.95	7.65
RAR [43]		–	7.23
MTCNN [58]		5.39	6.90
Unsupervised / self-supervised			
Thewlis [47]	30	7.15	–
	50	6.67	10.53
Thewlis [46](frames)	–	5.83	8.80
Shu † [40]	–	5.45	–
Zhang [57]	10	3.46	7.01
w/ equiv.	30	3.16	6.58
w/o equiv.	30	8.42	–
Wiles ‡ [53]	–	3.44	–
Ours, training set: CelebA			
loss-net: selfsup.	10	3.19	6.86
	30	2.58	**6.31**
	50	**2.54**	6.33
loss-net: sup.	10	3.32	6.99
	30	2.63	6.39
	50	2.59	6.35
Ours, training set: VoxCeleb			
loss-net: selfsup.	30	3.94	6.75
w/ bias	30	3.63	–
loss-net: sup.	30	4.01	7.10

Table 1: **Comparison with state-of-the-art on MAFL and AFLW.** K is the number of unsupervised landmarks. †: train a 2-layer MLP instead of a *linear* regressor. ‡: use the larger VoxCeleb2 [7] dataset for unsupervised training, and include a bias term in their regressor (through batch-normalization). Normalised $\%$-MSE is reported (see fig. 3).



(a) (b) (c) (d)

Figure 6: **Invariant Localisation.** Unsupervised keypoints discovered on smallNORB test set for the *car* and *airplane* categories. Out of 20 learned keypoints, we show the most geometrically stable ones: they are invariant to pose, shape, and illumination. **[b–c]:** elevation-vs-azimuth; **[a, d]:** shape-vs-illumination (y-axis-vs-x-axis).

Ablation study. In table 2 we present two ablation studies: (1) on the keypoint bottleneck, and (2) where we compare against adversarial and other image-reconstruction losses. For both the settings, we take the best performing model configuration for facial landmark detection on the MAFL dataset.

Keypoint bottleneck. The keypoint bottleneck has two functions: (1) it provides a differentiable and distributed representation of the location of landmarks, and (2) it restricts the information from the target image to spatial locations only. When the bottleneck is replaced with a generic low dimensional fully-connected layer (as in a conventional auto-encoder) the performance degrades significantly This is because the continuous vector embedding is not encouraged to encode geometry explicitly.

Reconstruction loss. We replace our content/perceptual loss with ℓ_1 and ℓ_2 losses on generated pixels; the losses are also optionally paired with an *adversarial* term [13] to encourage verisimilitude as in [18]. All of these alternatives lead to worse landmark detection performance (table 2). While GANs are useful for aligning image distributions, in our setting we reconstruct a *specific* target image (similar to an auto-encoder). For this task, it is enough to use a simple content/perceptual loss.

4.3 Learning human body landmarks

Setup. Articulated limbs make landmark localisation on human body significantly more challenging than faces. We consider two *video* datasets, BBC-Pose [3], and Human3.6M [17]. BBC-Pose comprises of 20 one-hour long videos of sign-language signers with varied appearance, and dynamic background; the test set includes 1000 frames. The frames are annotated with 7 keypoints corresponding to head, wrists, elbows, and shoulders which, as for faces, we use only for quantitative evaluation, not for training. Human3.6M dataset contains videos of 11 actors in various poses, shot from multiple viewpoints. Image pairs are extracted by randomly sampling frames from the same video sequence, with the additional constraint of maintaining the time difference within the range 3-30 frames for Human3.6M. Loose crops around the subjects are extracted using the provided annotations and resized to 128×128 pixels. Detectors for $K = 20$ and $K = 50$ keypoints are trained on Human3.6M and BBC-Pose respectively.

Qualitative results. Figure 4 shows raw unsupervised keypoints and the regressed semantic ones on the BBC-Pose dataset. For each annotated keypoint, a maximally matching unsupervised keypoint is identified by solving bipartite linear assignment using mean distance as the cost. Regressed keypoints consistently track the annotated points. Figure 5 shows $\langle \mathbf{x}, \mathbf{x}', \Psi\left(\mathbf{x}, \Phi(\mathbf{x}')\right), \Phi(\mathbf{x}')\rangle$ quadruplets, as for faces, as well as the discovered keypoints. All the keypoints lie on top of the human actors, and consistently track the body across identities and poses. However, the model cannot discern frontal and dorsal sides of the human body apart, possibly due to weak cues in the images, and no explicit constraints enforcing such consistency.

fc-layer (d) $\rightarrow$	10	20	60	ours K=30
MAFL	20.60	21.94	28.96	**2.58**

loss $\rightarrow$	ℓ_1	adv.+ ℓ_1	ℓ_2	adv.+ ℓ_2	content (ours)
MAFL (K=30)	3.64	3.62	2.84	2.80	**2.58**

Table 2: **Abalation Study. [left]:** The keypoint bottleneck when replaced with a low d-dimensional, $d = \{10, 20, 60\}$, *fully-connected* (fc) layer leads to significantly worse landmark detection performance (%-MSE) on the MAFL dataset. **[right]:** Replacing the *content* loss with ℓ_1, ℓ_2 losses on the images, optionally paired with an *adversarial* loss (*adv.*) also degrades the performance.

Figure 7: **Disentangling Style and Geometry.** Image generation conditioned on *spatial* keypoints induces disentanglement of representations for style and geometry in the generator. Source image ($\mathbf{x}$) imparts style (*e.g.* colour, texture), while the target image ($\mathbf{x}'$) influences the geometry (*e.g.* shape, pose). Here, during inference, $\mathbf{x}$ **[middle]** is sampled to have a different *style* than $\mathbf{x}'$ **[top]**, although during training, image pairs with *consistent* style were sampled. The generated images **[bottom]** borrow their style from $\mathbf{x}$, and geometry from $\mathbf{x}'$. **(a) SVHN Digits:** the foreground and background colours are swapped. **(b) AFLW Faces:** pose of the style image $\mathbf{x}$ is made consistent with $\mathbf{x}'$. **(c) Human3.6M:** the background, hat, and shoes are retained from $\mathbf{x}$, while the pose is borrowed from $\mathbf{x}'$. All images are sampled from respective test sets, never seen during training.

Quantitative results. Figure 4 compares the accuracy of localising the 7 keypoints on BBC-Pose against *supervised* methods, for both self-supervised and supervised perceptual loss networks. The accuracy is computed as the the $\%$-age of points within a specified pixel distance d. In this case, the top two supervised methods are better than our unsupervised approach, but we outperform [35, 55] using 1k training samples (vs. 10k); furthermore, methods such as [37] are specialised for videos and leverage temporal smoothness. Training using the supervised perceptual loss is understandably better than using the self-supervised one. Performance is particularly good on parts such as the elbow.

4.4 Learning 3D object landmarks: pose, shape, and illumination invariance

We train our unsupervised keypoint detectors on the SmallNORB [27] dataset, comprising 5 object categories with 10 object instances each, imaged from regularly spaced viewpoints and under different illumination conditions. We train category-specific detectors for $K = 20$ keypoints using image-pairs from neighbouring viewpoints and show results in fig. 6 for *car* and *airplane* (see supplementary material for visualisation of other object categories). Keypoints most invariant to various factors are visualised. These landmarks are especially robust to changes in illumination and elevation angle. They are also invariant to smaller changes in azimuth ($\pm 80^\circ$), but fail to generalise beyond that. Most interesting, they localise structurally similar regions, even when there is a large change in object shape (*e.g.* fig. 6-(d)); such landmarks could thus be leveraged for viewpoint-invariant semantic matching.

4.5 Disentangling appearance and geometry

In fig. 7 we show that our method can be interpreted as disentangling appearance from geometry. Generator/ keypoint networks are trained on SVHN digits [30], AFLW faces, and Human3.6M people. The generator network is capable of retaining the geometry of an image, and substituting the style with any other image in the dataset, including unrelated image pairs never seen during training. For example, in the third column we re-render the number 3 by mixing its geometry with the appearance of the number 5. This generalises significantly from the training examples, which only consist of pairs of digits sampled from the *same* house number instance, sharing a common style.

5 Conclusions

In this paper we have shown that a simple network trained for conditional image generation can be utilised to induce, without manual supervision, a object landmark detectors. On faces, our method outperforms previous unsupervised as well as supervised methods for landmark detection. The method can also extend to much more challenging data, such as detecting landmarks of people, and diverse data, such as 3D objects and digits.

Acknowledgements. We are grateful for the support provided by EPSRC AIMS CDT, ERC 638009-IDIU, and the Clarendon Fund scholarship. We would like to thank James Thewlis for suggestions and support with code and data, and David Novotný and Triantafyllos Afouras for helpful advice.

References

[1] J. Bruna, P. Sprechmann, and Y. LeCun. Super-resolution with deep convolutional sufficient statistics. In *Proc. ICLR*, 2016.

[2] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1513–1520. IEEE, 2013.

[3] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman. Domain adaptation for upper body pose tracking in signed TV broadcasts. In *Proc. BMVC*, 2013.

[4] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *Proc. ICCV*, volume 1, 2017.

[5] X. Chen and A. L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Proc. NIPS*, 2014.

[6] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proc. NIPS*, pages 2172–2180, 2016.

[7] J. S. Chung, A. Nagrani, and A. Zisserman. VoxCeleb2: Deep speaker recognition. In *INTERSPEECH*, 2018.

[8] E. L. Denton and V. Birodkar. Unsupervised learning of disentangled representations from video. In *Proc. NIPS*. 2017.

[9] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Proc. NIPS*, 2016.

[10] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Proc. NIPS*, pages 658–666, 2016.

[11] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*. 1977.

[12] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proc. CVPR*, 2016.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc. NIPS*, 2014.

[14] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[15] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[17] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI*, 2014.

[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, 2017.

[19] J. Johnson, A. Alahi, and F. Li. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, 2016.

[20] N. Kalchbrenner, A. Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. *arXiv preprint arXiv:1610.00527*, 2016.

[21] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Proc. NIPS*, pages 5574–5584, 2017.

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[24] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *ICCV Workshops*, 2011.

[25] Z. L., P. L., X. W., and X. T. Deep learning face attributes in the wild. In *Proc. ICCV*, 2015.

[26] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *Proc. ECCV*, 2016.

[27] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proc. CVPR*, 2004.

[28] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. CVPR*, 2017.

[29] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

[30] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS DLW*, volume 2011, 2011.

[31] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Proc. NIPS*, 2016.

[32] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proc. CVPR*, 2017.

[33] D. Novotny, D. Larlus, and A. Vedaldi. Learning 3d object categories by looking around them. In *Proc. ICCV*, volume 4, 2017.

[34] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *ICLR Workshop*, 2015.

[35] T. Pfister, J. Charles, and A. Zisserman. Large-scale learning of sign language by watching TV (using co-occurrences). In *Proc. BMVC*, 2013.

[36] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In *Proceedings of the Asian Conference on Computer Vision*, 2014.

[37] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *Proc. ICCV*, 2015.

[38] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee. Deep visual analogy-making. In *Proc. NIPS*, 2015.

[39] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In *Proc. NIPS*, pages 217–225, 2016.

[40] Z. Shu, M. Sahasrabudhe, A. Guler, D. Samaras, N. Paragios, and I. Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *Proc. ECCV*, 2018.

[41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[42] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *Proc. ICML*, pages 843–852, 2015.

[43] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proc. CVPR*, 2013.

[44] I. Sutskever, G. E. Hinton, and G. W. Taylor. The recurrent temporal restricted boltzmann machine. In *Proc. NIPS*, pages 1601–1608, 2009.

[45] S. Suwajanakorn, N. Snavely, J. Tompson, and M. Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Proc. NIPS*, 2018.

[46] J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised object learning from dense invariant image labelling. In *Proc. NIPS*, 2017.

[47] J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proc. ICCV*, 2017.

[48] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. *arXiv preprint arXiv:1704.05831*, 2017.

[49] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proc. ICML*, pages 1096–1103. ACM, 2008.

[50] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Proc. NIPS*, pages 613–621, 2016.

[51] G. Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

[52] W. F. Whitney, M. Chang, T. Kulkarni, and J. B. Tenenbaum. Understanding visual concepts with continuation learning. In *ICLR Workshop*, 2016.

[53] O. Wiles, A. S. Koepke, and A. Zisserman. Self-supervised learning of a facial attribute embedding from video. In *Proc. BMVC*, 2018.

[54] T. Xue, J. Wu, K. L. Bouman, and W. T. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Proc. NIPS*, 2016.

[55] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proc. CVPR*, 2011.

[56] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *Proc. ECCV*, 2014.

[57] Y. Zhang, Y. Guo, Y. Jin, Y. Luo, Z. He, and H. Lee. Unsupervised discovery of object landmarks as structural representations. In *Proc. CVPR*, 2018.

[58] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *Proc. ECCV*, pages 94–108. Springer, 2014.

[59] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning Deep Representation for Face Alignment with Auxiliary Attributes. *PAMI*, 2016.

Appendix

We first present more detailed results on MAFL dataset comparing performance of different versions of our method. Then we show extended versions of figures presented in the paper. The sections are organized by the datasets used.

A MAFL

	Training set →	CelebA			VoxCeleb
K landmarks	Regression set	Thewlis [47]	sup.	selfsup.	sup.
10	MAFL	7.95	3.32	3.19	—
30		7.15	2.63	2.58	4.17
50		6.67	2.59	2.54	3.59
10	CelebA	6.32	3.32	3.19	—
30		5.76	2.63	2.57	4.14
50		5.33	2.59	2.53	3.55

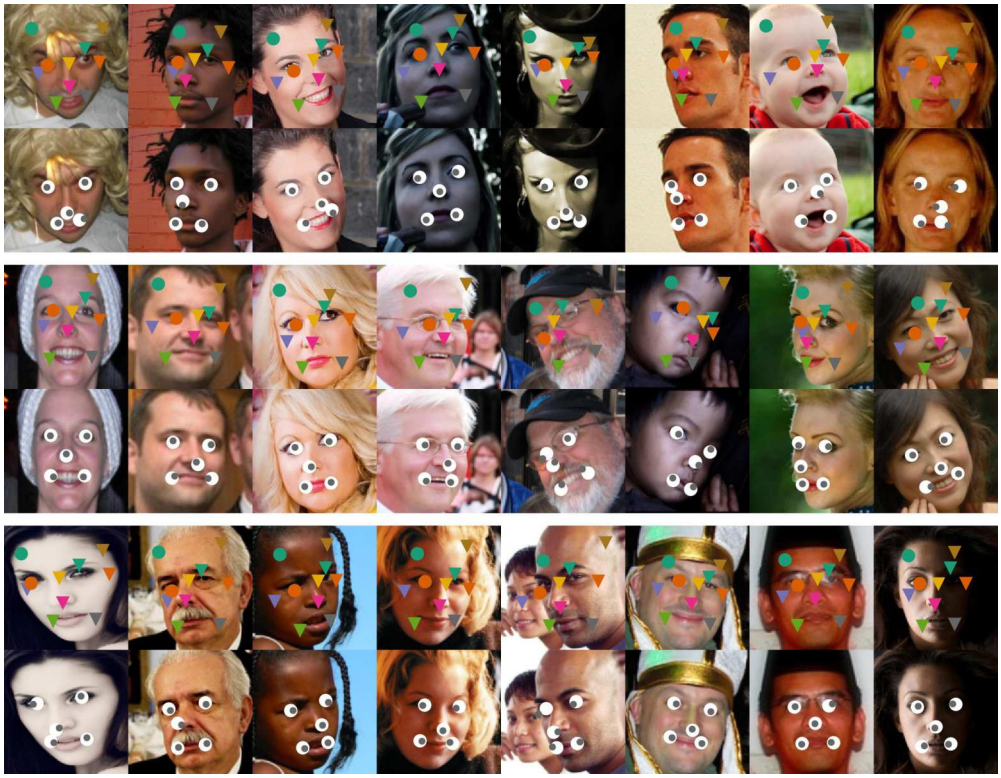
Table 3: **Results on MAFL face-landmarks test-set.** Varying number (K) of unsupervised landmarks are learnt on two training-sets — random-TPS warps on CelebA [25], and face-videos from the VoxCeleb [29]. These landmarks are regressed onto 5 manually-annotated landmarks in the MAFL [59] test set, using either CelebA or MAFL training sets. Mean squared-error (MSE) normalised by the inter-ocular distance is reported.

B Boundary Discounting

When TPS warping is used during training some pixels in the resulting image may lie outside the original image. Since reconstructing these empty pixels is irrelevant we ignore them in the reconstruction loss. We additonaly ignore 10 pixels on the edges of the original image and use a smooth step over the next 20 pixels. This is to further discourage reconstruction of the empty pixels as they can influence the perceptual loss when a convolutional neural network with a large receptive field is used.

C MAFL and AFLW Faces

Figure 8: Supervised linear regression of 5 keypoints (bottom rows) from 10 unsupervised (top rows) on MAFL (above) and AFLW (below) test sets. Centre of the white-dots correspond to the ground-truth location, while the dark ones are the predictions. The models were trained on random-TPS warped image-pairs; self-supervised peceptual-loss network was used.



D VoxCeleb

Figure 9: Training with video frames from VoxCeleb. **[rows top-bottom]:** (1) source image $\mathbf{x}$, (2) target image $\mathbf{x}'$, (3) generated target image $\Psi(\mathbf{x}, \Phi(\mathbf{x}'))$, (4) unsupervised landmarks $\Phi(\mathbf{x}')$ superimposed on the target image. The landmarks consistently track facial features.

E BBCPose

Figure 10: **Learning Human Pose.** 50 unsupervised keypoints are learnt. Annotations (empty circles) for 7 keypoints are provided, corresponding to — head, wrists, elbows and shoulders. Solid circles represent the predicted positions; Top rows show raw discovered keypoints which correspond maximally to each annotation; bottom rows show linearly regressed points from the discovered keypoints. **[above]:** randomly sampled frames for different actors **[below]:** frames from a video track.

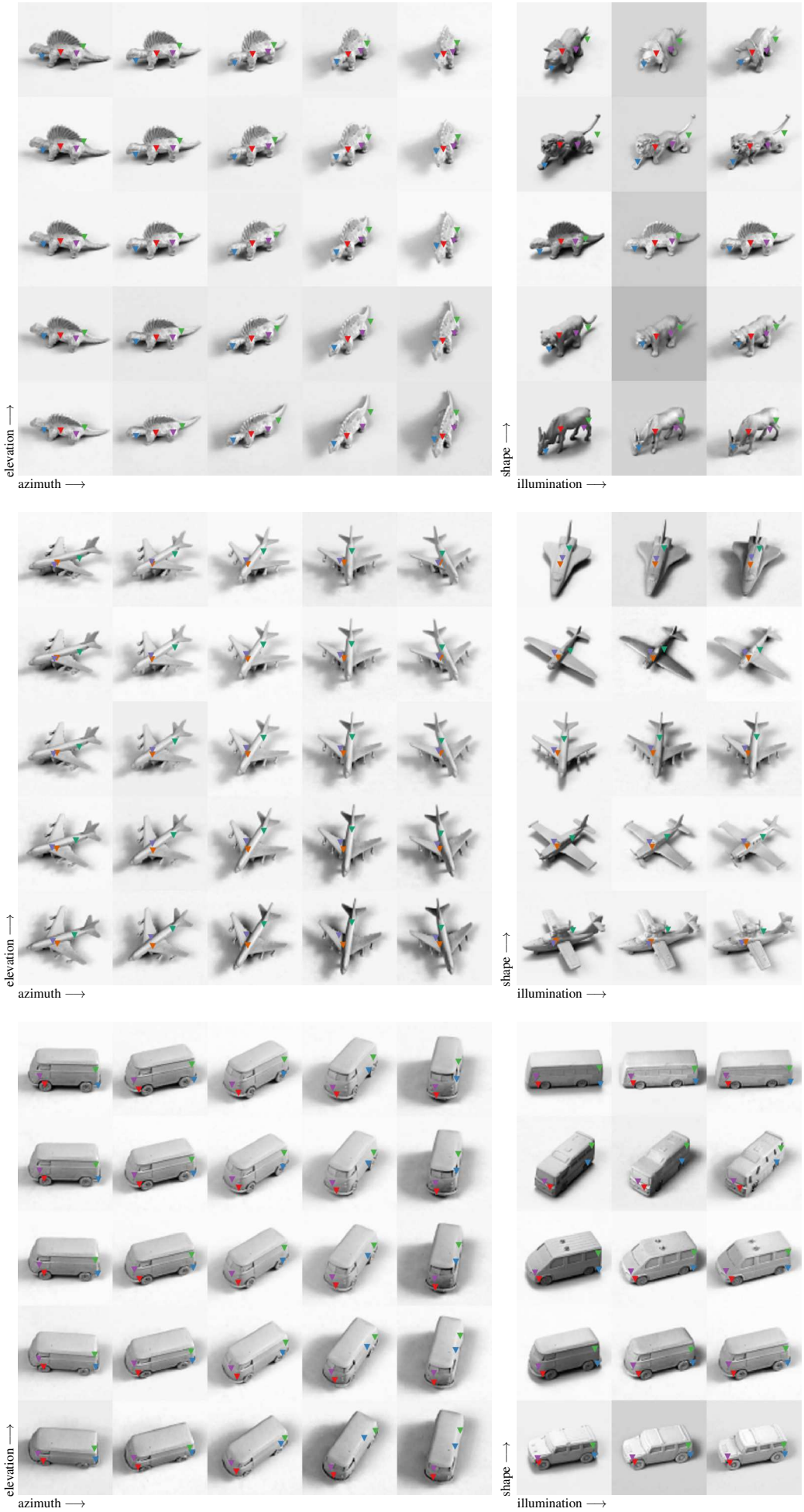
F Human3.6M

Figure 11: **Unsupervised Landmarks on Human3.6M.** Video of two actors (S1, S11) "posing", from the Human3.6M test set. **(rows)** (1) source, (2) target, (3) generated, (4) landmarks, (5) landmarks on frames from a different view, (6–7) landmarks on two views of the second actor. The landmarks consistently track the legs, arms, torso and head across frames, views and actors. However, the model confounds the frontal and dorsal sides.

G smallNORB 3D Objects: pose, shape, and illumination invariance

Object-category specific keypoint detectors are trained on the 5 categories in the smallNORB dataset — *human, car, animal, airplane,* and *truck*. Training is performed on pairs of images, which differ only in their viewpoints, but have the same object instance (or shape), and illumination.

Keypoints invaraint to viewpoint, illumniation, and object shape are visualised for object instances in the test set. The training set consists of only 5 object instances per category, yet the detectors generalise to novel object instances in the test set, and correspond to structurally similar regions across instances.


elevation ⟶
azimuth ⟶
shape ⟶
illumination ⟶
elevation ⟶
azimuth ⟶
shape ⟶
illumination ⟶
elevation ⟶
azimuth ⟶
shape ⟶
illumination ⟶

H Disentangling appearance and geometry

The generator substitutes the appearance of the target image ($\mathbf{x}'$) with that of the source image ($\mathbf{x}$). Instead of sampling image pairs $(\mathbf{x}, \mathbf{x}')$ with *consistent* style, as done during training, we sample pairs with *different* styles at inference, resulting in compelling transfer across different object categories — SVHN digits, Human3.6M humans, and AFLW faces.

Figure 12: **SVHN digits.** Target, source, and generated image triplets $\langle \mathbf{x}', \mathbf{x}, \Psi(\mathbf{x}, \Phi(\mathbf{x}')) \rangle$ from the SVHN test set. The digit shape is swapped out, while colours, shadows, and blur are retained.

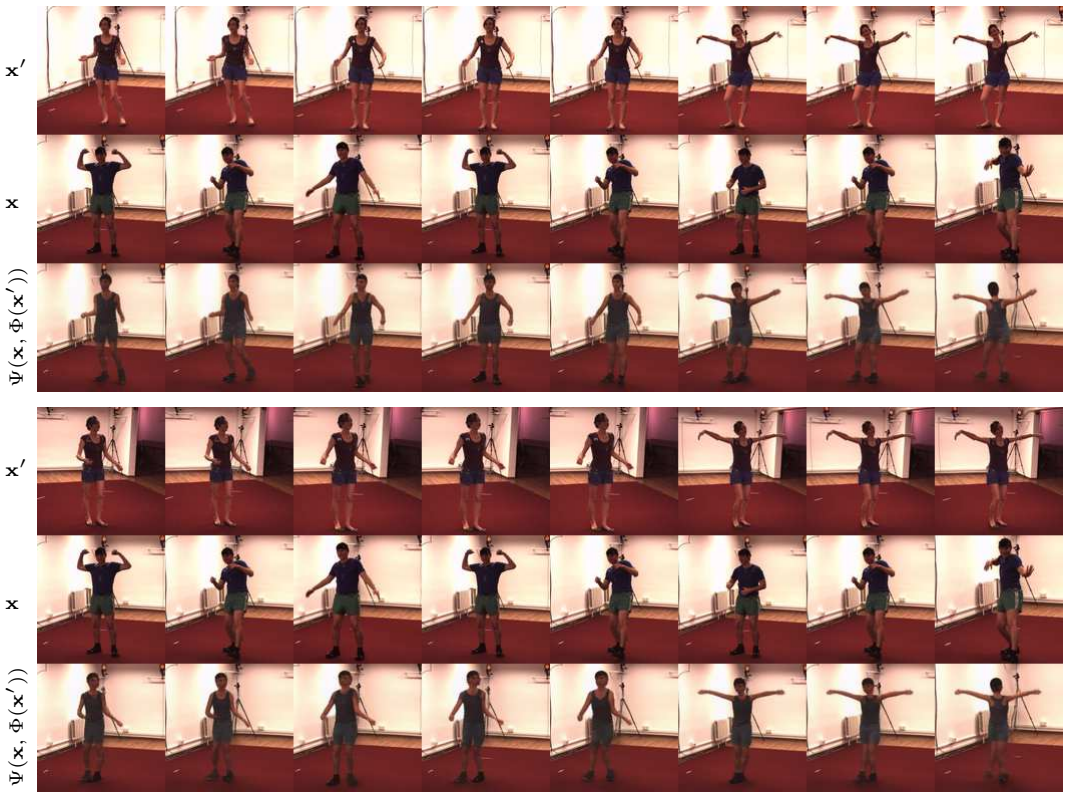


Figure 13: **Human3.6M humans.** Transfer across actors and viewpoints. **[top]:** different actors in various poses, imaged from the same viewpoint; the pose is swapped out, while appearance characteristics like shoes, clothing colour, and hat are retained. **[bottom]:** successful transfer even when the target is imaged from a different viewpoint (same poses as above).

Figure 14: **AFLW Faces.** The source image $\mathbf{x}$ is rendered with the pose from the target image $\mathbf{x}'$; the identity is retained.

I Encoding for changes in the background

Our method learns keypoints to encode for *changes* between the given two images (frames from a video, or a pair of synthetically warped images) to minimise the reconstruction loss. If the only change between the images corresponds to the object of interest (*e.g.* in its pose, or viewpoint), the keypoints will exclusive encode for object geometry / appearance. However, if the background is not static, the keypoints shall encode for these changes as well.

An example where the background is not static is the BBC Pose dataset. Here, half of the frame in the background shows the news stream (on the left side), which changes from frame to frame (see appendix E). Hence, the unsupervised keypoints learnt by our method also encode for these changes (see below). 2D location of the keypoints is used to communicate the content of the background. This is a limitation of our approach.

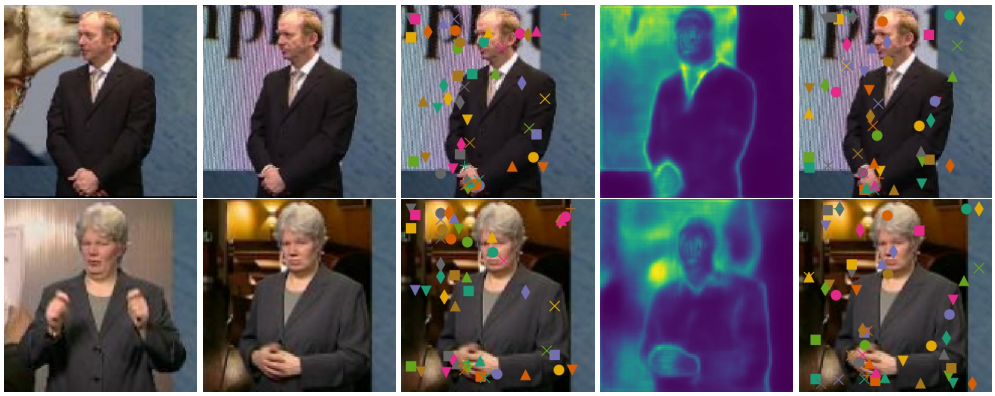


Figure 15: **Non-static background.** The unsupervised "keypoints" encode for changes between the two frames, include those in the background. **[left-to-right]:** (1) past image ($\mathbf{x}$), (2) future image ($\mathbf{x}'$), (3) unsupervised landmarks without confidence prediction, (4) predicted per-pixel confidence map, (5) unsupervised keypoints with confidence prediction.

In an attempt to ameliorate this, we performed some preliminary experiments. We discounted the standard per-pixel ℓ_2-loss with a predicted per-pixel confidence score (or *uncertainty* σ) as in [21, 33]:

$$\mathcal{L}(\mathbf{x}', \hat{\mathbf{x}}') = \sum_{i=1}^{H}\sum_{j=1}^{W} \frac{1}{2\sigma_{ij}^2}||\mathbf{x}'_{ij} - \hat{\mathbf{x}}'_{ij}||^2 + \frac{1}{2}\log\sigma_{ij}^2$$

The model learns to express more uncertainty for the non-static parts of the background (compare the left and right side of the confidence maps above, and the highly articulated fingers). However, it does not focus the keypoints away from the background onto the signers (last column). This is an important problem, and would benefit from further exploration.

Encoding occluded background

Another related problem is when parts of background become disoccluded in the future frame. The network is then tasked to fill in the missing background patch given only the occluded past frame. This is another situation where, the keypoints could be exploited for encoding the disocculded background. However, in our experiments with Human3.6M dataset, where the camera is static, we found that the model "remembers" the background without using any keypoints to encode for it: see fig. 1 in the paper, where the radiator in the background is reconstructed perfectly; the model does not place any keypoints on the background (see fig. 11).

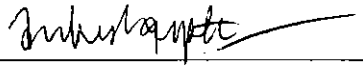
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	**Unsupervised Learning of Object Landmarks through Conditional Image Generation**
Publication Status	□Published ☑Accepted for Publication ☑Submitted for Publication □ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Submitted to Neural Information Processing Systems (NIPS), 2018 T. Jakab*, **A. Gupta***, H. Bilen, A. Vedaldi (* denotes equal contribution)

Student Confirmation

Student Name:	Ankush Gupta		
Contribution to the Paper	▪ Conception of the idea ▪ Implementation and exhaustive experimentation on the various datasets, namely: faces, humans, digits, 3D objects, and birds ▪ Writing the paper ▪ Preparing supplementary material		
Signature		Date	30th August, 2018

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Prof. Andrea Vedaldi		
Supervisor comments		
Signature 	Date	13/9/2018

This completed form should be included in the thesis, at the end of the relevant chapter.

6

Inductive Visual Localisation: Factorised Training for Superior Generalisation

This work was presented as a *spotlight* presentation at the 29th British Machine Vision Conference (BMVC), 2018.

Inductive Visual Localisation: Factorised Training for Superior Generalisation

Ankush Gupta
Andrea Vedaldi
Andrew Zisserman
{ankush,vedaldi,az}@robots.ox.ac.uk

Visual Geometry Group
Department of Engineering Science
University of Oxford

Abstract

End-to-end trained Recurrent Neural Networks (RNNs) have been successfully applied to numerous problems that require processing sequences, such as image captioning, machine translation, and text recognition. However, RNNs often struggle to generalise to sequences longer than the ones encountered during training. In this work, we propose to optimise neural networks explicitly for *induction*. The idea is to first decompose the problem in a sequence of inductive steps and then to explicitly train the RNN to reproduce such steps. Generalisation is achieved as the RNN is not allowed to learn an arbitrary internal state; instead, it is tasked with mimicking the evolution of a valid state. In particular, the state is restricted to a spatial memory map that tracks parts of the input image which have been accounted for in previous steps. The RNN is trained for single inductive steps, where it produces updates to the memory in addition to the desired output. We evaluate our method on two different visual recognition problems involving visual sequences: (1) text spotting, *i.e*. joint localisation and reading of text in images containing multiple lines (or a block) of text, and (2) sequential counting of objects in aerial images. We show that inductive training of recurrent models enhances their generalisation ability on challenging image datasets.

1 Introduction

A key issue in sequence and program learning is to model long-term structure in the data. For example, in language modelling one has two choices: The first is to consider simple models such as character and word n-grams, which generalise well but fail to capture long-term correlations in the data. The second is to switch to models such as Recurrent Neural Networks (RNNs) that, in principle, can capture arbitrarily long correlations. In practice, however, RNNs are trained using back-propagation through time on sequences of limited length and may fail to generalise to longer sequences [1, 2, 3, 4].

This is in contrast with the standard and very familiar notion of mathematical induction, which allows sequences to be analysed or generated ad infinitum. Many problems, such as counting objects or reading text, have an inherent inductive structure: all one needs to do is 1) count the current object or read the current character; and 2) move to the next object or character (or stop when finished). The two steps can then be iterated to process data of arbitrary length. However, as noticed by several authors [1, 2, 3, 4], and confirmed in our

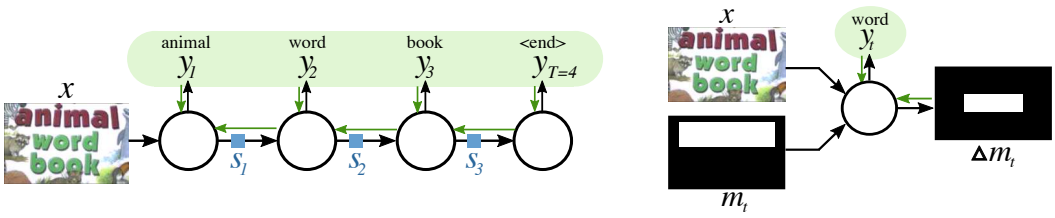


Figure 1: **Inductive visual localisation.** A recurrent neural network (RNN) for sequence recognition is trained end-to-end on sequences $(y_1, \ldots, y_T)$. The internal states ($\mathbf{s}_t$) are learnt from gradients of losses summed over the entire sequence. Without structuring the state space, the recurrent unit may fail to learn the appropriate loop invariant, and thus may be unable to generalise to sequences longer than the length T used for training. We address this problem by decomposing the end-to-end training procedure (left) into one-step inductive updates (right). We achieve this by restricting the recurrent state to a spatial memory map $\mathbf{m}_t$, which keeps track of the progress in processing the sequence. The recurrent network learns to incrementally predict, in addition to the output sequence $\mathbf{y}_t$, updates $\Delta\mathbf{m}_t$ to the memory. Using this inductive decomposition, our network can generalise well to sequences of length far greater than those in the training set.

experiments, RNNs fail to correctly repeat these steps beyond the number of times considered during training.

In this paper we marry the idea of induction to sequence processing using recurrent neural networks (RNN). The inductive approach in this paper can be viewed as an application to the spatial domain of recent approaches for learning programs using recurrent and compositional networks [3, 5].

Our contributions are threefold: 1) we propose to train recurrent networks with the explicit notion of *induction*, where the end-to-end training procedure is decomposed into inductive sub-steps. We show that RNNs trained on one-step inductive updates have superior generalisation ability. 2) We develop a recurrent module with inductive factorisation for recognising multiples lines of text in challenging scene text images, and outperform state-of-the-art methods by combining text localisation and recognition into a single architecture. 3) We apply our approach to sequential visual counting, and validate it on a challenging aerial image dataset, once again demonstrating improved generalisation capabilities.

The rest of the paper is organised as follows: in section 2 we first review related work; next, in section 3 we introduce our inductive decomposition approach; finally, in section 4 we present results on recognising multiple lines of text in images containing blocks of text, and on sequential visual object counting.

2 Related Work

Learning & Composing Programs. Explicit decomposition into the repeated sub-tasks is related to the recent work on neural controllers, which interface with sub-programs or modules. Neural Programmer-Interpreters [3] present a general framework for dispatching functions using a program-stack, which Cai *et al*. [5] augment with the explicit notion of recursion, showing superior generalisation. Zaremba *et al*. [6], learn controllers for operators acting on coarse 2D grids of discrete symbols, while our method works on dense pixel-grids.

RNNs for Scene Text Recognition Scene text recognition is a well-studied problem, with roots in optical and handwritten character recognition [7]. Traditional methods focused on single character recognition [8, 9] with explicit language models (e.g., n-grams or lexicons) to form words or sentences [10, 11, 12, 13, 14, 15, 16, 17]. More recently, the word-level text recognition has been explored extensively [18, 19, 20, 21], primarily using Convolutional Neural Networks (CNNs) [7] to encode images, and RNNs as the decoders [22, 23, 24, 25, 26]. A key component of the RNN decoders is soft-attention [27] which iteratively pools image features at each step of the recurrence [25, 26]. [28] extend this attention to 2D feature-maps for recognising multiple-lines of handwritten text.

RNNs for Visual Object Counting. Counting objects in images [29] has numerous applications *e.g.*, histological analysis of microscopy images [30], parsing medical scans, and population studies from aerial imagery [31]. Sequential counting has been shown to be the primary method of counting in humans [32], and was explored with convolutional-RNNs in [33]. More recently, [34] combine the object density based regression methods [35] with iterative counting using recurrent fully-convolutional networks. We factorise the end-to-end training of such iterative counting methods, and achieve superior generalisation.

3 Method

In order to generalise correctly to sequences of arbitrary lengths, an iterative algorithm must maintain a suitable invariant. For example, an algorithm that counts objects via enumeration maintains as invariant the list of objects visited so far, which must contain no repetition. In order to maintain this invariant, the algorithm must visit at each step a "new" object, or terminate if no more objects are available. However, RNNs are trained without any explicit constraints on the structure of their hidden state, and may not learn such an invariant correctly. For example, a list of objects has no a-priori limitation on its size. While for a human this is obvious, an RNN may be unable to understand it as it cannot experience unbounded lists during training.

In order to address this issue, we design an RNN to update a state that, by construction, has a universal step-independent validity. In particular, we restrict the recurrent state to a *spatial memory map* which keeps track of the parts of the input image which have already been explored. At each step, the model is conditioned on this memory map (in addition to the input image), and predicts as output a token for the sequence, as well as an update to the spatial memory. Such updates for the counting example above amount to adding one more object to the list of visited objects. This is analogous to taking the *inductive step* in mathematical induction.

The rest of the section describes our model in detail. In section 3.1 we discuss encoder-decoder RNNs [36, 37, 38] enhanced with soft-attention [27, 39] as these are suitable for modelling sequences in images. In section 3.2 we describe our inductive decomposition for these recurrent models. Finally, in section 3.3 we detail training and inference.

3.1 Encoder-Decoder Models

Let $\mathbf{x} \in \mathcal{X} = \mathbb{R}^{H \times W \times C}$ be an image, where H, W and C are its height, width and number of color channels. Furthermore, let $\mathbf{y} = (y_1, \ldots, y_T) \in \mathcal{Y}^T$ be the corresponding sequence label, where T is the sequence length. Encoder-decoder methods model the conditional probability

$p(\mathbf{y}|\mathbf{x})$ as a product of conditionals for the next token y_{t+1} given a context vector $\mathbf{c}_t$ and the previously-predicted tokens $y_{1:t}$:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=0}^{T-1} p(y_{t+1}|y_{1:t}, \mathbf{c}_t) \tag{1}$$

These conditional probabilities are modelled using an RNN Φ. We consider in particular an LSTM [4] with hidden state $\mathbf{s}_t$, and write:

$$\Big(p(y_{t+1}|y_{1:t}, \mathbf{c}_t), \quad \mathbf{s}_{t+1}\Big) = \Phi(\mathbf{s}_t, \mathbf{c}_t, y_t).$$

The context vector $\mathbf{c}_t$ injects into the model information extracted from the input image. Context can be kept constant for all steps, or can be dynamically focused on different parts of the image using an attention mechanism. In the first case, exemplified by sequence-to-sequence models [38], the context is extracted[1] by a Convolutional Neural Network (CNN) [40] $\mathbf{c}_t = \mathbf{c} = \Psi(\mathbf{x}) \in \mathbb{R}^{H' \times W' \times C'}$. In the second case, exemplified by [27], the context vector is computed at each step via attention by reweighing the CNN output:

$$\mathbf{c}_t = \sum_{i \in H'} \sum_{j \in W'} \alpha_{ij} \Psi(\mathbf{x})_{ij}, \tag{2}$$

$$\alpha_{ij} = \frac{\exp(v_{ij})}{\sum_{i'} \sum_{j'} \exp(v_{i'j'})}, \tag{3}$$

$$v_{ij} = w^T \tanh(W\mathbf{s}_t + W'\Psi(\mathbf{x})_{ij} + b), \tag{4}$$

where, $v_{ij} \in \mathbb{R}$ is the unnormalised attention score and w, W, W', b are learnable parameters.

The model is trained end-to-end to maximize the log of the posterior probability (1) averaging over example (image $\mathbf{x}$, sequence $\mathbf{y}$) pairs.

3.2 Inductive Decomposition

The RNN models discussed in the previous section may fail to learn a correct inductive decomposition of the problem, and thus fail to generalise properly to sequences of arbitrary length. We propose to address this problem in a simple and yet effective manner: rather than allowing the RNN to learn its own state space, we specify a suitable state space a priori, and train the RNN to make use of it. In more detail, we set the RNN state to be a spatial memory $\mathbf{s}_t = \mathbf{m}_t$ containing a mask covering all the visual objects that have been accounted for up to time t. In this manner, the content of the memory can be derived from the ground-truth data annotations and the step number t. Furthermore, the RNN does not need to learn a new state space from scratch, but only how to generate $\mathbf{m}_{t+1}$ from $\mathbf{m}_t$; for this, training can focus on learning single-step predictions, from t to $t+1$, rather than whole-sequence predictions.

The spatial memory $\mathbf{m} \in \mathbb{R}^{H \times W}$ is implemented as a single 2D map of the same dimensions as the image $\mathbf{x}$. At each step, the model predicts y_t, as well as an update $\Delta\mathbf{m}_t$ to the memory. In this work, we focus on sequence prediction tasks where each token in the sequence corresponds to a 2D location in the image. Hence, $\Delta\mathbf{m}_t$ is trained to encode the 2D location in the image associated with y_t.

Predictions at each step are conditioned on the context vector $\mathbf{c}_t$ and the memory $\mathbf{m}_t$:

$$\Big(p(y_{t+1}|y_{1:t}, \mathbf{c}_t), \quad \Delta\mathbf{m}_{t+1}\Big) = \Phi(\mathbf{c}_t, \mathbf{m}_t). \tag{5}$$

[1] Since in this case no attention is used, the CNN is usually configured so that $H' = W' = 1$.

In practice, at each step, the memory is concatenated with the image to obtain the encoded representation $\Psi([\mathbf{x}||\mathbf{m}])$ instead of being fed into Φ directly; $\mathbf{c}_t$ is obtained from Ψ as detailed in section 3.1. The memory is initialised to all zeros, *i.e.*, $\mathbf{m}_{t=0} = 0^{H\times W}$, and is updated after each step as:

$$\mathbf{m}_{t+1} = \mathbf{m}_t + \Delta\mathbf{m}_t. \tag{6}$$

The exact architecture of Φ and the location representation in $\mathbf{m}_t$ are application dependent, and examples are detailed in sections 4.1.1 and 4.2.1.

3.3 Training and Inference

Training. The model is trained for one-step predictions, where each training sample is a tuple — $(\mathbf{x}, y_t, \mathbf{m}_t, \mathbf{m}_{t+1})$: image, token at time t, ground-truth accumulated location maps $\mathbf{m}_t$ and $\mathbf{m}_{t+1}$. The model is optimised through stochastic gradient descent (SGD) to minimise the following loss:

$$-\log p(y_t|\mathbf{x}_t, \mathbf{m}_t) + \gamma||\mathbf{m}_t + \Delta\mathbf{m}_t - \mathbf{m}_{t+1}||_2^2 \tag{7}$$

where, $\gamma > 0$ balances the terms. The first term maximises the probability of the correct token y_t, while the second is a pixel-wise reconstruction loss for the predicted spatial memory update $\Delta\mathbf{m}_t$. The reconstruction loss provides supervision to the intermediate memory representation, and is similar to auxiliary losses for classification applied to latent features in *Deeply Supervised Nets* [41]. The sequence label for the final step (y_T) indicates the end-of-sequence, *e.g.* through an additional class in the output labels; it is used for terminating the inference loop.

Optimisation. All model parameters are initialised randomly (sampled from a gaussian with 0.01 standard deviation). The model is trained with SGD using the AdaDelta optimiser [42]. γ in eq. (7) is set to 1 for all the experiments.

Inference. The memory is initialised to all zeros. At each step, the memory is concatenated with the image and fed through the image encoder to get the encoded-representation $\Psi([\mathbf{x}||\mathbf{m}])$. Then, the log-probabilities for y_t, and memory updates $\Delta\mathbf{m}_t$ are regressed from the recurrent module Φ. The memory is updated per eq. (6), and fed into the model iteratively, until the end-of-sequence is predicted. An example is shown in fig. 2.

4 Experiments

We evaluate our method on two different tasks – in section 4.1 we present results on recognising lines of text in images containing a block or multiple lines of text, and in section 4.2 we explore counting objects in images through enumeration. We demonstrate superior generalisation ability of recurrent modules when trained with inductive factorisation, and outperform state-of-the-art methods on a text-spotting task.

4.1 Recognising Multiple Lines of Text

We apply our model to the task of spotting, *i.e.* joint localisation and recognition, of text in images containing multiple lines (or a block) of text. We proceed in two steps: first, using synthetically generated text-block images, we show superior generalisation ability of the model trained with inductive factorisation; second, we fine-tune this text-block spotting model

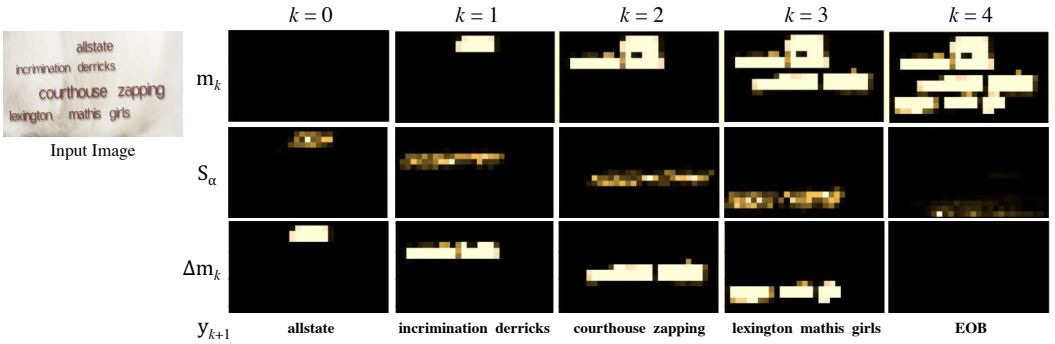


Figure 2: **Multi-line text recognition inference visualisation.** Decoding procedure for an example image containing four lines. Inference runs for five steps, predicting the line characters (y_k) in the first four steps, and indicating the end-of-block (EOB) in the fifth step. The memory-map is initialised (at k=0) to all zeros, and is iteratively updated by adding in the predicted update $\Delta\mathbf{m}_k$, regressed from the accumulated attention maps S_α.

trained on synthetic data, on real text data extracted from the ICDAR 2013 [43] benchmark, and outperform the state-of-the-art word-level text spotting method.

4.1.1 Model Details

Given an image $\mathbf{x}$ containing multiple lines of text (or a text-block; see examples in fig. 3), the corresponding sequence label $\mathbf{y}$ is a sequence of characters in lexicographic order (*i.e.* left-to-right, first-to-last line). We factorise the problem of spotting text in text-blocks at the level of lines, *i.e.* a "token" y_k (in section 3.2), corresponds to the k^{th} line. Hence, y_k itself is a sequence of n characters $\{y_1^k, \ldots, y_n^k\}$ in the k^{th} line. At each inductive step, one full-line is recognised.

Spatial Memory Representation. The spatial memory $\mathbf{m}_k$ at step k represents the location of first k lines which have been recognised so far, by setting the pixel-values inside the corresponding line-level bounding-boxes to 1 (the background is 0). The memory updates $\Delta\mathbf{m}_k$ correspond to the location of k^{th} line (see fig. 2).

Image Encoder (Ψ). We employ the fully-convolutional DRN-C-26 Dilated Residual Network [44] as the image-encoder, which consists of six residual blocks; the encoder downsamples the image by a factor of 8, and has a stride of 32 (details in the reference, and appendix). Hence, an input image, concatenated with memory-map ($\mathbf{m}$) of dimensions $H \times W \times 4$ is encoded as a feature-map of dimensions $\lceil \frac{H}{8} \rceil \times \lceil \frac{W}{8} \rceil \times 512$.

Recurrent Module (Φ). We use an LSTM-RNN with soft-attention over the convolutional features as the line-level character decoder. The state-size is set to 1024, while the attention-embedding dimension is set to 512. The attention weights (α in eq. (3)) corresponding to all the predicted characters in the current line are summed up; this produces an approximate localisation (S_α) of the line (second row in fig. 2). S_α is concatenated with the image-representation Ψ, and convolved with a stack of 2 convolutional+ReLU [45] layers (128 filters each), and a final 1×1 convolutional layer to produce the memory-update ($\Delta\mathbf{m}$).

	# lines→	1	2	3	4	5	6	7	8	9	10
end-to-end	precision	66.69	63.97	59.23	53.70	-	-	-	-	-	-
	recall	69.27	65.50	56.52	39.14	-	-	-	-	-	-
	ED	15.91	17.81	25.08	43.78	-	-	-	-	-	-
inductive	precision	85.13	84.79	85.57	87.25	87.32	86.11	85.41	85.51	84.57	84.41
	recall	84.89	84.74	85.32	86.99	87.22	85.91	84.43	84.03	80.47	76.80
	ED	6.76	7.79	7.09	6.29	5.77	6.96	8.77	9.11	13.18	17.23

Figure 3: **Synthetic Text Blocks results.** (top) Samples with different number of text lines from the Synthetic Text Blocks test set. (bottom) Word-level precision, recall, and character-level normalised edit-distance (ED) are reported (all in %).

4.1.2 Synthetic Text Blocks

Following the success of synthetic data in text-spotting [14, 19, 46], we test the generalisation to number of lines beyond those present in the training set, on synthetically generated text-block images (fig. 3). Using synthetic data enables this study, as it is difficult to collect real-world images of text with a large number of lines.

Dataset. The training set consists of text-block containing 3–5 lines, while the test contains 1–10 lines, with 500000 samples for each number of lines. To generate a synthetic text-block image for a given number of lines, the following procedure is followed: a random number of words (3–5 per line) are selected from a lexicon of approximately 90k words [19]. Then the text-lines are randomly aligned (left, centre, right), resized to potentially different heights (within the same block), separated by random amounts of line-spacing, transformed with a small perspective or affine transformation, and finally rendered against a randomly chosen background image, with a font chosen from over 1200 fonts. Figure 3 shows some samples generated through this procedure.

Evaluation Metrics. We report the word-level precision and recall, computed as the intersection of the predicted words and the ground-truth words, normalised by the total number of predicted or ground-truth words respectively, in addition to the normalised edit-distance.

Results. The results are summarised in fig. 3, and fig. 2 visualises the predicted updates to the mask on a test image. We note consistent levels of precision and recall, even when testing with twice the number of lines than in the training set. We also trained a RNN model (with identical Ψ, Φ) end-to-end without any inductive factorisation. It suffered from two difficulties: (1) block-level end-to-end training did not converge for more than three lines; and (2) the model did not generalise to more than three lines: note, the steep fall in the recall rates, and increase in the edit-distance for four lines.

4.1.3 Real Text Blocks – ICDAR-2013

We fine-tune the inductive block-parser trained on synthetic data on text-blocks extracted from the ICDAR 2013 Focussed Scene Text [43] dataset, and compare with state-of-the-art word-level methods.

Dataset. ICDAR 2013 is a dataset of 229 training, and 233 test scene images containing text, with word-level bounding boxes and text-string annotations. As our model is tailored for

# lines→		1	2	3	4	5	overall
# blocks		169	79	29	6	3	117
# words		270	267	136	34	40	477
He *et al*. [47] and Jaderberg *et al*. [19]	P	**96.15**	**95.59**	**94.59**	**98.92**	**98.74**	**95.94**
	R	61.40	72.22	76.09	64.71	72.50	68.58
	F	74.95	82.28	84.34	**78.23**	83.68	79.98
ours	P	80.31	84.09	86.76	72.97	87.18	82.24
	R	**77.04**	**83.15**	**86.76**	**79.41**	**85.00**	**81.06**
	F	**78.64**	**83.62**	**86.76**	76.06	**86.08**	**81.65**

Figure 4: **ICDAR-2013 Text Blocks results.** (image) Samples with different number of text lines from the ICDAR-2013 Text Blocks test set. (table) Number of block-images with the given number of text lines (*# blocks*), the total number of words in these images (*# words*), and the word-level precision (*P*), recall (*R*), and F-score (*F*) (all in %) are reported.

recognising text in blocks, we extract images of text-blocks, *i.e*. images containing multiple lines of text from the dataset: first, text lines are formed by linking together pairs of words with relative distance at most half, and three times the minimum of their heights, along the vertical and horizontal directions respectively; then, the bounding boxes of these lines are doubled along the height, and those with area of intersection at least a third of the maximum of their areas are merged into text-blocks. Ground-truth text-line masks are used to supervise the spatial memory when fine-tuning the inductive block-parser pre-trained on synthetic text-blocks. Some samples are visualised, and the number of blocks obtained and other statistics are given in fig. 4.

Baseline word level model. We combine the word localisation model of He *et al*. [47] (88% F-score on ICDAR13 Focussed Scene localisation), with the strong lexicon based word-level recognition network of [19] (90.8% on ICDAR13 cropped word recognition).[2] This combination of state-of-the-art models for localisation and recognition, provides a strong baseline to compare our joint model against. To minimise the discrepancy between the test and training setting we run the detector on full scene images and then using the detected word locations, associate them with the text-blocks used in this experiment. Note, we also ran the baseline detector on block images but this lead to worse results: word-spotting F-score = 78.83% (block-images) vs. 79.98% (full-scene images).

Evaluation. As is standard in the benchmark, we report the word-level recall, precision and the F-score. For fair comparison with the lexicon-based word-recognition model [19], we use the same lexicon of 90k words to constrain the predictions of our model.

Results. Figure 4 summarises the results. We note that our inductive block parser, consistently achieves a higher recall and F-score (except for 4-lines) across all lines. Our method combines the stages of both localisation and recognition, and hence avoids downstream error propagation, achieving greater recall. The higher precision of the baseline is due to the detector only producing high-confidence detections.

[2]Implementations were obtained from the authors.

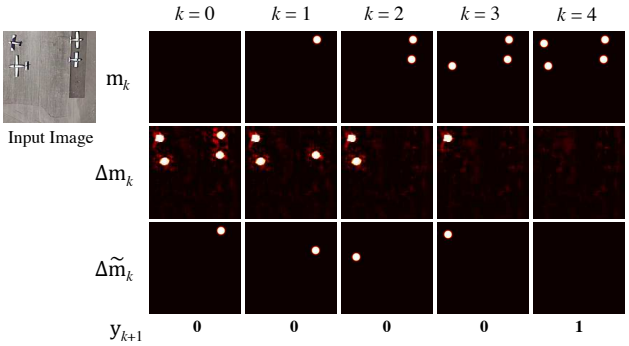


Figure 5: **Enumerative counting visualisation.** Decoding procedure for an example image containing four airplanes. Inference runs for five steps, indicating the end-of-sequence ($y_k = 1$) in the fifth step. At each step, the memory update (Δm_k) regresses peaks at the location of *all* the remaining objects, out of which one is randomly picked ($\Delta \tilde{m}_k$) to update the memory-map for the next step.

4.2 Counting by Enumeration

We further test improvement in generalisation ability induced by inductive training in a different setting: counting objects in images through enumeration, where a recurrent module acts as an enumerator, counting one object in each step, and terminating when done. We test on two datasets: first, images containing randomly coloured shapes, and second: aerial images of airplanes extracted from the recently introduced DOTA dataset [48].

4.2.1 Model Details

In each inductive step one object is counted, and a corresponding label of '0' is produced; the enumeration terminates when all the objects have been accounted for, producing '1' as the output, as in [33]. Hence, for a given image $\mathbf{x}$ containing multiple ($= N \geq 0$) objects of interest, the corresponding sequence label is $(0,\ldots,0,1)$, *i.e.* a sequence containing N zeros, and one 1.

Spatial Memory Representation. The spatial memory $\mathbf{m}_k$ at step k represents the location of first k objects which have been enumerated, by placing a small gaussian peak at their centre. Due to lack of any natural order for counting, it is difficult to assign a particular object to a specific enumeration step. Hence, the memory update $\Delta\mathbf{m}_k$ at the k^{th} step regresses gaussian peaks at the locations of *all* the remaining objects. The memory is updated with the location of one of the remaining objects (selected randomly); fig. 5 visualises the inference steps for a sample image.

Image Encoder (Ψ). A fully-convolutional Dilated Residual Network [44], consisting of three residual-blocks each with two pre-activation residual units [49], is used to encode the images (detailed architecture in appendix). The images are not downsampled, hence, the feature-maps retain the original dimensions of the input.

Recurrent Module (Φ). The memory updates $\Delta\mathbf{m}_k$ are regressed from the image-features, using a 1×1 convolutional layer. The binary valued token at each step y_k is predicted as: $p(y_k = 1) = \text{Sigmoid}\,(w \cdot \text{MaxPool}(\Delta\mathbf{m}_k) + b)$.

Dataset	Model	Number of Objects							
		3	4	5	6	7	8	9	10
Coloured Shapes	end-to-end	99.53	99.53	98.93	0	0	0	0	0
	inductive	100	99.89	99.52	98.93	97.18	98.47	95.48	95.45
DOTA	end-to-end	82.00	70.50	74.80	0	0	0	0	0
	inductive	82.50	79.00	75.50	72.50	69.00	43.81	32.21	29.20

Figure 6: **Enumerative counting results.** (top) Samples from the Coloured Shapes and DOTA Airplane datasets containing different number of objects. (bottom) Accuracy of enumerative counting with and without inductive training (all in %).

4.2.2 Datasets

We evaluate on two datasets described below. Images of size 128×128 were used for both the datasets. The training sets consisted of images containing $\{3,4,5\}$ objects; to test for generalisation beyond training sequence lengths, the test set included 3–10 objects. Figure 6 visualises some samples from the datasets.

Coloured Shapes. We start with a procedurally generated synthetic dataset, which consists of shapes with random colour, position and type (circle, triangle, or square) placed on the canvas. This dataset provides a simplified setting for analysis without confounding complexity. The training set consists of 10k images, while the test set consists of 200 images for each count.

DOTA Airplanes. DOTA [48] is a recently introduced image dataset of high-resolution aerial images, where 15 object categories have been labelled with oriented bounding-boxes. We extract image crops from the DOTA dataset which consist of airplanes. The training set consists of 20k crops extracted from 100 images from the dataset's training set, while the test set consists of 200 image crops for each object count (ranging from 3 to 10) extracted from 70 images from the dataset's validation set.

4.2.3 Results

We compare our inductive model against a soft-attention LSTM-RNN trained end-to-end with the same image encoder Ψ. We report the mean accuracy of prediction, where a test image is evaluated as correct if the predicted count matches the ground-truth count exactly. Figure 6 summarises the results on both the datasets for the two models. On both the datasets, end-to-end trained RNN fails to generalise to object counts beyond those in the training set (> 5), while the one with inductive training does not fail catastrophically at higher counts. Lower accuracy of the inductive model at higher counts can be attributed to crowding of objects, which is not seen in the training set (*e.g.* fig. 6 rightmost image).

5 Conclusions

While RNNs may seem a perfect match for problems with an inductive structure, these networks fail to learn appropriate invariants to allow recursion to extend beyond what is encountered in the training data. We have shown how to repurpose standard RNN models to restrict the recurrent state to a suitable state-representation – which in our application are the image locations corresponding to the predictions at each step – where the correct invariant can be enforced. The result is an iterative visual parsing architecture which generalises well-beyond the training sequence lengths. This idea can be extended to visual problems with a tail-recursive structure, from object tracking to boundary and line tracing.

Acknowledgements. We thank Triantafyllos Afouras for proofreading. Financial support was provided by the UK EPSRC CDT in Autonomous Intelligent Machines and Systems Grant EP/L015987/2, EPSRC Programme Grant Seebibyte EP/M013774/1, and the Clarendon Fund scholarship.

References

[1] A. Joulin and T. Mikolov, "Inferring algorithmic patterns with stack-augmented recurrent nets," in *NIPS*, 2015.

[2] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[3] S. Reed and N. De Freitas, "Neural programmer-interpreters," in *Proc. ICLR*, 2015.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] J. Cai, R. Shin, and D. Song, "Making neural programming architectures generalize via recursion," in *Proc. ICLR*, 2017.

[6] W. Zaremba, T. Mikolov, A. Joulin, and R. Fergus, "Learning simple algorithms from examples," in *Proc. ICML*, 2016.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] d. T. Campos, B. R. Babu, and M. Varma, "Character recognition in natural images," *VISAPP*, 2009.

[9] C. Yao, X. Bai, B. Shi, and W. Liu, "Strokelets: A learned multi-scale representation for scene text recognition," in *Proc. CVPR*, 2014.

[10] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. ICCV*. IEEE, 2011, pp. 1457–1464.

[11] C. Lee, A. Bhardwaj, W. Di, V. Jagadeesh, and R. Piramuthu, "Region-based discriminative feature pooling for scene text recognition," in *Proc. CVPR*, 2014.

[12] O. Alsharif and J. Pineau, "End-to-end text recognition with hybrid HMM maxout models," in *Proc. ICLR*, 2014.

[13] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. ECCV*, 2014.

[14] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. ICPR*. IEEE, 2012, pp. 3304–3308.

[15] A. Mishra, K. Alahari, and C. Jawahar, "Scene text recognition using higher order language priors," *Proc. BMVC.*, 2012.

[16] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang, "Scene text recognition using part-based tree-structured character detection," in *Proc. CVPR*, 2013.

[17] T. Novikova, O. Barinova, P. Kohli, and V. Lempitsky, "Large-lexicon attribute-consistent text recognition in natural images," in *Proc. ECCV*. Springer, 2012, pp. 752–765.

[18] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," in *Proc. ICLR*, 2014.

[19] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," in *Workshop on Deep Learning, NIPS*, 2014.

[20] ——, "Deep structured output learning for unconstrained text recognition," in *International Conference on Learning Representations*, 2015.

[21] A. Poznanski and L. Wolf, "Cnn-n-gram for handwriting word recognition," in *Proc. CVPR*, 2016.

[22] B. Su and S. Lu, "Accurate scene text recognition based on recurrent neural network," in *Proc. ACCV*, 2014.

[23] P. He, W. Huang, Y. Qiao, C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences, 2016," in *The 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, vol. 1, 2016.

[24] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *ArXiv e-prints*, 2015.

[25] C. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for ocr in the wild," in *Proc. CVPR*, 2016.

[26] B. Shi, X. Wang, P. Lv, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. CVPR*, 2016.

[27] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015.

[28] T. Bluche, "Joint line segmentation and transcription for end-to-end handwritten paragraph recognition," in *NIPS*, 2016.

[29] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Interactive object counting," in *Proc. ECCV*, 2014.

[30] ——, "Learning to detect cells using non-overlapping extremal regions," in *International Conference on Medical Image Computing and Computer Assisted Intervention*, ser. Lecture Notes in Computer Science, N. Ayache, Ed., MICCAI. Springer, 2012, pp. 348–356.

[31] C. Arteta, V. Lempitsky, and A. Zisserman, "Counting in the wild," in *Proc. ECCV*, 2016.

[32] S. Dehaene and L. Cohen, "Dissociable mechanisms of subitizing and counting: Neuropsychological evidence from simultanagnosic patients." *Journal of Experimental Psychology: Human Perception and Performance*, vol. 20, no. 5, p. 958, 1994.

[33] B. Romera-Paredes and P. H. S. Torr, "Recurrent instance segmentation," in *Proc. ECCV*, 2016.

[34] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura, "Fcn-rlstm: Deep spatio-temporal neural networks for vehicle counting in city cameras," in *Proc. ICCV*, 2017.

[35] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *NIPS*, 2010.

[36] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[37] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NIPS*, 2014, pp. 3104–3112.

[38] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156–3164.

[39] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. ICML*, 2015.

[40] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[41] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. AISTATS*, 2015, pp. 562–570.

[42] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[43] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, L. P. de las Heras *et al.*, "ICDAR 2013 robust reading competition," in *Proc. ICDAR*, 2013, pp. 1484–1493.

[44] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. CVPR*, 2017.

[45] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. ICML*, 2010.

[46] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. CVPR*, 2016.

[47] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *Proc. ICCV*, 2017.

[48] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "Dota: A large-scale dataset for object detection in aerial images," in *Proc. CVPR*, 2018.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. ECCV*, 2016.

[50] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015.

[51] F. Y. and V. K., "Multi-scale context aggregation by dilated convolutions," in *Proc. ICLR*, 2016.

Appendix

In appendix A we first give detailed architecture of the image-encoder Ψ used for text-recognition in multiple lines (section 4.1 in paper), and for visual object counting (section 4.2). Next, in appendix B we present results on a synthetic shapes dataset for the recognition task, similar to the one used counting; this was excluded from the paper due to lack of space.

A Image Encoder Architecture (Ψ)

Our image encoder is based on the Dilated Residual Network [44]. We give details of the architecture of the encoder used for text-recognition and counting respectively.

A.1 Text Recognition Encoder

The image encoder is based on the DRN-C-26 network of [44]. The network if fully-convolutional, downsamples the input by a factor of 8, and has a stride of 32. The layer-level details are as following (top is first layer):

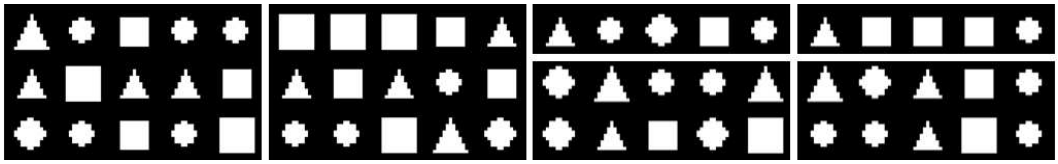
Conv−5×5−F16−D1
Res−3×3−F16−D1−S2
Res−3×3−F32−D1−S2
Res−3×3−F64−D1
Res−3×3−F64−D1−S2
Res−3×3−F128−D1
Res−3×3−F128−D1
Res−3×3−F256−D2
Res−3×3−F256−D2
Res−3×3−F512−D4
Res−3×3−F512−D4
Conv−3×3−F512−D2
Conv−3×3−F512−D2
Conv−3×3−F512−D1
Conv−3×3−F512−D1

Where,

- 'Conv' stands for a convolutional layer, with ReLU activation [45] and batch-normalisation [50]; 'Res' stands for the Pre-activation Residual Unit of He *et al*. [49].
- second term is dimensions of the the filters
- 'Fn' means n filters
- 'Dr' gives the dilation rate of the filters [51].
- if present, 'S2' means a filter stride of 2, otherwise the stride is 1.

A.2 Visual Object Counting Encoder

The image encoder employed for counting is much simpler, and employs six residual [49] layers. The image is not downsampled. Layer-wise details, using the naming scheme given



↓ Model \ Lines→	1	2	3	4	5	6	7	8	9	10	15	20
end-to-end	241.80	54.59	0	28.41	42.66	52.44	59.17	64.20	68.17	71.28	80.9	85.71
inductive	0	0	0	0	0	0	0	0	0	0	0	0.02

Figure 7: (top) Samples with different number of text lines from the Toy Shapes test set. (bottom) Normalised edit-distance rates (%) for the task of recognizing shapes organised in blocks, comparing the generalisation capabilities of a conventional soft-attention RNN trained end-to-end, and with our inductive factorisation. The models were trained on blocks containing three lines, and tested for generalisation on a varying number of lines. Error rates of more than 100% are due to the model always predicting exactly three lines.

above, are as following:

Res−5×5−F32−D1−S1
Res−5×5−F32−D1−S1
Res−5×5−F32−D2−S1
Res−5×5−F32−D2−S1
Res−5×5−F32−D4−S1
Res−5×5−F32−D4−S1

B Recognising Multiple Lines of Text: Toy Example

We evaluate generation ability of sequence recognition models to multiple lines on a synthetic Shapes dataset similar to the Coloured Shapes dataset used for counting. We experimented with the toy task of recognising sequences of shapes organised in multiple lines in an image (see fig. 7). Two models are evaluated: first, our inductive block parser, which is trained at the line-level. The other, a conventional soft-attention encoder/decoder RNN trained at the block level (no factorisation in terms of lines).

Dataset. Binary valued images containing three types of shapes — square, triangle, and circle — each in two different sizes, and organised into lines, are synthetically generated. Each line consists of a sequence of five randomly sampled shapes. The training set consists of 2000 such images all containing *three* lines. The test set consists of 12 different subsets with varying number of lines — {1 to 10, 15, 20}, each containing 200 samples.

Evaluation. We evaluate the models on images containing differing number of lines {1–10,15,20}, to test for generalisation. We report the normalised edit-distance, computed as the total edit-distance between the predicted block-string, and the ground-truth block-string, normalised by the length of the ground-truth string.

Model Architecture. The two models, our inductive block parser and a conventional RNN model have the identical architectures: the image-encoder is a stack of six convolutional layers+ReLU (with 4×16, 2×32 filters, two 2×2 max-pooling after the second and the fourth-layers); the decoder is a soft-attention LSTM-RNN with 128 hidden units; attention embedding is also 128 dimensional. The memory updates $\Delta\mathbf{m}_t$ are regressed using two convolutional+ReLU layers (32 filters each).

Discussion. The results are shown in fig. 7. Both the models achieve perfect recognition accuracy on the test set containing three lines (the same number of lines as in the training set), but only the inductive line parser is able to generalise to different number of lines. The conventional RNN trained end-to-end to produce block-level predictions, always predicts three lines regardless of the number of lines in the test image.

7

Conclusion

In this chapter we first summarise the main contributions and achievements of the work pressented in this thesis (section 7.1). Next, we propose extensions and highlight the scope for further developing some of the work (section 7.2).

7.1 Achievements and impact

Synthetic scene text image generation. In chapter 3, we introduced a large-scale synthetic dataset of full scene-level text images. This dataset embeds synthetic text instances which look realistic, in a fast and scalable manner. We showed that text detection networks trained purely on our *synthetically* generated images generalise to *real* text images, and significantly outperform the existing state-of-the-art methods. This validates the variation and verisimilitude of the visual style represented in the dataset. We have publicly released the pre-generated dataset (800K images)[1], and also open-sourced the synthetic engine.[2] The engine has attracted significant open-source community contributions (with nearly 300 forks on GitHub): while originally designed for rendering horizontally

[1] *SynthText in the Wild* dataset: http://www.robots.ox.ac.uk/~vgg/data/scenetext/

[2] Implementation of the synthetic engine is available at: https://github.com/ankush-me/SynthText

oriented text in the Latin (English) script, it has been extended for rendering vertically oriented text as well, for various different languages, namely — Chinese, Arabic, and Bangla. Patel et al. [2018] have adapted the engine for many different languages[3] to train a multi-language text spotting system. Further, *Facebook Inc.*'s recent text spotting system [Borisyuk et al., 2018], which is routinely applied to billions of images is trained on a variant of our dataset.[4][5] Our dataset has become the standard for training text detection networks, and is widely used by the research community (the publication has received over 250 citations). Further, our dataset has enabled development of network architectures [Bušta et al., 2017, Gómez et al., 2018, He et al., 2018, Li et al., 2017] for end-to-end text spotting, *i.e.*, joint detection and recognition; this was not possible earlier due to lack of sufficiently large training datasets for joint training of the networks.

Fully-convolutional text detector. In chapter 3, we presented a fast, fully-convolutional text detection network. At the time of publication, the network significantly advanced the state-of-the-art for text detection across datasets; since then there has been active development and many extensions. Our network replaced the region proposals designed for general objects (*e.g.*, EdgeBoxes [Zitnick and Dollar, 2014b]) employed in the detection pipeline of Jaderberg et al. [2015b], with those *learned* for specifically text. This significantly improved the detection performance, and also sped-up the region proposal stage by over 45×. Further, the fully-convolutional design significantly reduced the number of model parameters (by over 90% as compared to architectures with fully-connected layers, *e.g.* YOLO-v1 [Redmon et al., 2016b]), and enabled generalisation to images of arbitrary sizes. To the best of our knowledge, we were perhaps the first to propose a *fully-convolutional* architecture for detection. The popular object detection frameworks at the time, namely Fast/Faster R-CNN [Girshick, 2015b, Ren et al., 2016] applied costly per-region subnetwork hundreds of time. However, due to its restricted application

[3] Synthetic data is especially useful for less common languages, which are not well represented in common datasets.

[4] Facebook's blog-post on their text spotting system: https://code.fb.com/ai-research/rosetta-understanding-text-in-images-and-videos-with-machine-learning/

[5] *Wired* article: https://www.wired.com/story/facebook-rosetta-ai-memes/

to text, the impact was limited to the text community; the idea was brought into the conciousness of the broader vision community by Dai et al. [2016] in their *Region Fully Convolutional Networks* work (R-FCN), and is now standard practice. We have setup an online demo for text spotting based on our localisation network;[6] it employs modified versions of networks proposed by Jaderberg et al. [2014a] for recognition. The *live* demo presents a simple interface for the user to upload images and returns results within seconds. Over 10K images have been uploaded to our demo between November, 2016 and September, 2018 (approx. 2 years), winning it the distinction of being the most popular VGG demo.[7] We have also released the pre-compiled binaries for the demo.[8]

Unsupervised text recognition. In chapter 4, we developed a method for text recognition, which learns from *unaligned* samples of text-images, and valid text-strings from the target language, using *no* labelled training data. We applied this to recognise an historic printed book with excellent recognition accuracy (over 96% character-level accuracy). We further analysed various factors which affect the convergence of the method, and found that training with longer text-strings converges faster; strings of length 5 or smaller do not provide enough constraints for successful convergence. We showed that the model learns characters roughly in the order of their uni-gram frequencies. We further demonstrated the excellent generalisation ability of our method to recognise strings of drastically different lengths than those it was trained on. To the best of our understanding, this is the first demonstration of learning to decode sequences of discrete symbols from images given no aligned data. Further, it is perhaps a surprising result, and is a significant advance over existing methods for text recognition, which rely on millions of annotated examples. This result opens up a new and promising direction for training sequence recognition models for structured domains (e.g. language) given no labelled training data.

Unsupervised discovery of object landmarks. In chapter 5, we developed a method for *unsupervised* discovery of object landmarks, given only unlabelled object category

[6] Our text spotting live demo: http://zeus.robots.ox.ac.uk/textspot/
[7] VGG demos: http://www.robots.ox.ac.uk/~vgg/demo/
[8] Demo binaries: http://www.robots.ox.ac.uk/~vgg/software/textspot/

specific images/videos. We demonstrated learning landmark detectors for a variety of object categories — human faces, human body, and 3D objects. We achieved this by inducing a sparse keypoint like representation for reconstructing a *target* image from a *source* image of the same object but in a different pose. Our method drastically simplified the concurrently proposed framework of Zhang et al. [2018] with the key insight of *conditioning* the reconstruction with a second related image. Our method required no modification for application to different object categories. We demonstrated significantly better facial landmark detection than the current state-of-the-art methods, and achieved results comparable to *supervised* methods for detecting articulated human-body landmarks.

Improving generalisation in RNNs. In chapter 6, we addressed the problem of poor generalisation of *Recurrent Neural Networks* (RNNs) to sequence lengths beyond those present in the training set. We achieved this by factoring the training procedure over full sequences into a sequence of *inductive* steps. We restricted the training of RNNs to these single step updates, while encouraging valid state evolution. We further restricted the internal state the RNNs to a spatial memory map which tracks parts of the input image which have been accounted for so far. We demonstrated superior generalisation of RNNs to longer sequences, when trained with the above procedure on two distinct sequential visual recognition tasks, namely — joint localisation and recognition (end-to-end text spotting) of text lines in images, and counting objects in aerial images. Further, we demonstrated that end-to-end text spotting minimises compounding of errors, achieving superior/comparable results to multi-stage solutions.

7.2 Extensions and future work

We next propose a few extensions, and highlight the scope for further development of the work presented in this thesis.

Improving synthetic data generation. While our synthetic dataset is of sufficient verisimilitude to enable generalisation to real images, it can be improved further. Fine-

tuning models which have been pre-trained on our synthetic dataset on small amounts of real data has been shown to improve the detection performance. This exposes a domain gap, which can be bridged further by refining the generated images. Shrivastava et al. [2017] propose a *local* refinement module based on an adversarial discriminator to make synthetically rendered eye and hand images more realistic; they show improvements in performance of models trained on the refined data. This strategy could also be adapted for our domain to remove local artefacts introduced by the synthetic rendering process, *e.g.* aliasing effects, colour saturation, and uneven lighting. However, this only makes local corrections; more long-range parameters like position and scale could also be improved.

A limitation of our method is that the background images used for rendering text were filtered to not contain any text instances to avoid having any unlabelled text-instances in the generated synthetic images. This biases the dataset towards scenes / settings where text is less likely to be found, which is undesirable. Further, the method is agnostic to the semantics of the scene, and hence could render text onto implausible objects / parts of image *e.g.* sky, or animals. This second limitation has recently been addressed in Zhan et al. [2018], where they leverage semantic segmentation datasets, and place text only on plausible object categories. They show improvements in text detection and recognition performance upon training with their dataset with semantically plausible text placement. This should be further investigated to tease apart the contribution to the error from misplaced text / biased background images, and the models. Finally, more such improvements can be made, *e.g.* conditioning the text content on context.

Synthetic data efficiency. While synthetic data provides an inexhaustible source of labelled training data, there are limits to what can be achieved without further improving the synthetic generation process. This upper-bound on the performance, and the more fine-grained trend of how performance scales with the amount of data is yet unknown. These could be established by training models with progressively *larger* datasets, where larger could mean the number of synthetic samples, as well as the "variety" / number of variations of various factors, *e.g.* number of fonts, scales, background images, colours,

noise effects *etc.*. Such a systematic study would likely lead to insights into how to further improve the synthetic generation engine.

Combining generation with inference. The current synthetic engine is based on a number of distributions over the font, size, position, colours etc., which are specified independently; these can be further optimised to learn a more precise *joint* distribution which maximises the final performance of the models trained on the generated data. This can be achieved by using differentiable renderers/probabilistic inference. The *Ocular* system of Berg-Kirkpatrick et al. [2013] jointly models the text content, as well as the noisy rendering process for historical documents, and infers the parameters through the EM-algorithm; similar approach could be extended for the broader category of scene-text, and other visual recognition tasks, *e.g.* human pose estimation and inferring scene geometry. A related direction is to view optimising models on synthetic data as learning a proposal distribution generator for approximate inference in the synthetic generative model. Le et al. [2017] adapt the above approach to break *CAPTCHAs*; this can again be extended for text/other visual domains. Finally, neural generative models [Goodfellow et al., 2014a] can be learnt from real data, and used to generate training samples.

Adversarial learning from unaligned data. Our model for text recognition in images from *unaligned* data exploits long-range structure in the target domain, *i.e. language*. It can be extended to other input modalities, as long as the output is still language. Examples include reading lips, and learning to decode speech, sign-language, and even gestures. However, there are potential challenges to be tackled. Text characters have one-to-one mapping from visual representation to their corresponding character identities. In other domains this relationship between input and output is more tenuous. For example, aligning speech to text characters is challenging due to the variance in speaking styles and speed; perhaps a more feasible task could be to align speech with phonemes. Similarly, sign-language is significantly more challenging due to articulated motion and subtle facial expressions. Finally, the requirement of the output domain being language could be relaxed. Using an adversarial loss to align predictions with labels can be extended to

other visual tasks (*e.g.* Tung et al. [2017] present some preliminary results), and language translation [Lample et al., 2018b, Zhang et al., 2017].

Unsupervised object landmarks. Our framework for unsupervised discovery of landmarks, only scratches the surface of what is possible within this framework. First extension could be to learn 3D keypoints instead of 2D, for both rigid and non-rigid objects. Recent work of Suwajanakorn et al. [2018] presents a similar method for learning 3D landmarks, but has several limitations — it is not completely unsupervised and relies on information like the relative transformation between views, dominant viewpoint direction, and object silhouettes; further, they only demonstrate their method on toy data. Improvements can be made on both of these aspects. A limitation of our method is that it confuses symmetrical parts, *e.g.* it cannot distinguish between the frontal and dorsal sides of the human body. Explicitly modelling object symmetries, or estimating the viewpoint could help alleviate these limitations. Finally, the keypoints are learnt independently of each other and have no explicit geometric constraints among them. The keypoints could be augmented with interconnections, to learn a skeleton or a *constellation* model [Fergus et al., 2003, Weber et al., 2000] ■

Bibliography

S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 4

P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *Proceedings of the International Conference on Computer Vision*, pages 37–45, 2015. 23

J. Almazán, A. Gordo, A. Fornés, and E. Valveny. Word spotting and recognition with embedded attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36: 2552–2566, 2014. 36, 37, 38, 40

O. Alsharif and J. Pineau. End-to-end text recognition with hybrid HMM maxout models. *ArXiv e-prints*, Oct. 2013. 36, 37

M. Anthimopoulos, B. Gatos, and I. Pratikakis. Detection of artificial and scene text in images and video frames. *Pattern Analysis and Applications*, pages 1–16, 2011. 32

R. Arandjelovic and A. Zisserman. Look, listen and learn. In *Proceedings of the International Conference on Computer Vision*, pages 609–617. IEEE, 2017. 24

R. Arandjelovic and A. Zisserman. Objects that sound. In *Proceedings of the European Conference on Computer Vision*, pages 609–617. IEEE, 2018. 24

M. Artetxe, G. Labaka, E. Agirre, and K. Cho. Unsupervised neural machine translation. In *Proceedings of the International Conference on Learning Representations*, 2018. 30

M. Aubry and B. C. Russell. Understanding deep features with computer-generated imagery. In *Proceedings of the International Conference on Computer Vision*, pages 2875–2883, 2015. 20

Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*, pages 892–900, 2016. 24

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, 2015. 38

F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou. Edit probability for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 36, 39

S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1): 1–31, 2011. 16

M. Banko and E. Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th annual meeting on association for computational linguistics*, pages 26–33. Association for Computational Linguistics, 2001. 2, 3

J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994. 16, 17

H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. *Proceedings of the European Conference on Computer Vision*, pages 404–417, 2006. 23

T. Berg-Kirkpatrick, G. Durrett, and D. Klein. Unsupervised transcription of historical documents. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 207–217, 2013. 132

A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. PhotoOCR: Reading text in uncontrolled conditions. In *Proceedings of the International Conference on Computer Vision*, 2013. 36, 37

F. Borisyuk, A. Gordo, and V. Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 71–79. ACM, 2018. 128

G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009. 3, 15

M. Bušta, L. Neumann, and J. Matas. Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In *Proceedings of the International Conference on Computer Vision*, pages 2223–2231. IEEE, 2017. 39, 128

D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *Proceedings of the European Conference on Computer Vision*, 2012. 16, 17

d. T. Campos, B. R. Babu, and M. Varma. Character recognition in natural images. *VISAPP*, 2009. 21

H. Chen, S. Tsai, G. Schroth, D. Chen, R. Grzeszczuk, and B. Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *Proc. International Conference on Image Processing (ICIP)*, pages 2609–2612, 2011. 32

W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen. Synthesizing training images for boosting human 3d pose estimation. In *3D Vision (3DV)*, 2016. 18

X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. In *Proceedings of*

the IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages II–366, 2004. 32

Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou. Focusing attention: Towards accurate text recognition in natural images. In *Proceedings of the International Conference on Computer Vision*, pages 5086–5094. IEEE, 2017. 36, 39, 40

Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou. Aon: Towards arbitrarily-oriented text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5571–5579, 2018. 36, 39

H. Cho, M. Sung, and B. Jun. Canny text detector: Fast and robust scene text localization algorithm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 32

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 23, 37, 38

J. S. Chung and A. Zisserman. Out of time: automated lip sync in the wild. In *Proceedings of the Asian Conference on Computer Vision*, pages 251–263. Springer, 2016. 24

M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 3, 15

G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2012. 1

J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object detection via region-based fully

convolutional networks. In *Advances in Neural Information Processing Systems*, pages 379–387, 2016. 10, 33, 129

N. Dalal and B. Triggs. Histogram of Oriented Gradients for Human Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 886–893, 2005. 37

P. Dollar, R. Appel, and S. Belongie. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014. 33

A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the International Conference on Computer Vision*, pages 2758–2766, 2015a. 16

A. Dosovitskiy, J. Tobias Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1538–1546, 2015b. 20

A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator. In *1st Conference on Robot Learning*, 2017. 17, 19

B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2963–2970. IEEE, 2010. 32

P. Esser, E. Sutter, and B. Ommer. A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8857–8866, 2018. 27

A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *Proceedings of the British Machine Vision Conference*, volume 2, page 8, 2014. 23

S. R. Fanello, C. Keskin, S. Izadi, P. Kohli, D. Kim, D. Sweeney, A. Criminisi, J. Shotton, S. B. Kang, and T. Paek. Learning to be a depth camera for close-range human capture and interaction. *ACM Transactions on Graphics (TOG)*, 33(4):86, 2014. 19

P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 18

P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2241–2248, 2010a. 33

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient matching of pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2066–2073, 2000. 8

P. F. Felzenszwalb, R. B. Grishick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010b. 8

R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271, 2003. 8, 133

B. Fernando, H. Bilen, E. Gavves, and S. Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 23

M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computer*, c-22(1):67–92, Jan. 1973. 8

A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open

image sequences. In *Proceedings of the European Conference on Computer Vision*, pages 311–326. Springer-Verlag, 1998. 7

P. Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2): 194–200, 1991. 22

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997. 33

A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2016. 20

R. Gao, D. Jayaraman, and K. Grauman. Object-centric representation learning from unlabeled videos. In *Proceedings of the Asian Conference on Computer Vision*, pages 248–263. Springer, 2016. 23

A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 20

R. Girshick. Fast R-CNN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1440–1448, 2015a. 1, 39

R. B. Girshick. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision*, 2015b. 128

R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 33

V. Goel, A. Mishra, K. Alahari, and C. V. Jawahar. Whole is greater than sum of parts: Recognizing scene text words. In *International Conf. on Document Analysis and Recognition (ICDAR)*, pages 398–402, 2013. 36, 37

A. N. Gomez, S. Huang, I. Zhang, B. M. Li, M. Osama, and L. Kaiser. Unsupervised cipher cracking using discrete gans. In *Proceedings of the International Conference on Learning Representations*, 2018. 30

L. Gómez, A. Mafla, M. Rusinol, and D. Karatzas. Single shot scene text retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 40, 128

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014a. 5, 10, 27, 31, 132

I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. In *Proceedings of the International Conference on Learning Representations*, 2014b. 21, 37

A. Gordo. Supervised mid-level features for word image representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2956–2964, 2015. 36, 37

R. Goroshin, J. Bruna, J. Tompson, D. Eigen, and Y. LeCun. Unsupervised learning of spatiotemporally coherent metrics. In *Proceedings of the International Conference on Computer Vision*, pages 4086–4093, 2015. 22

A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005. 34, 38

A. Graves and J. Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in Neural Information Processing Systems*, 2009. 39

A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In

Proceedings of the 23rd international conference on Machine learning, pages 369–376. ACM, 2006. 38

A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 17, 34, 39

A. Gupta, A. Vedaldi, and A. Zisserman. Inductive visual localisation: Factorised training for superior generalisation. In *Proceedings of the British Machine Vision Conference*, 2018a.

A. Gupta, A. Vedaldi, and A. Zisserman. Learning to read by spelling: Towards unsupervised text recognition. *11th Indian Conference on Computer Vision, Graphics and Image Processing*, 2018b.

S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 345–360, 2014. 18

S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 24

R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1735–1742. IEEE, 2006. 22

A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Understanding real world indoor scenes with synthetic data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4077–4085, 2016. 19

C. G. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, pages 147–151, 1988. 7

D. Harwath, A. Torralba, and J. Glass. Unsupervised learning of spoken language with visual context. In *Advances in Neural Information Processing Systems*, pages 1858–1866, 2016. 24

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the International Conference on Computer Vision*, pages 2980–2988. IEEE, 2017a. 35

P. He, W. Huang, Y. Qiao, C. Loy, and X. Tang. Reading scene text in deep convolutional sequences, 2016. In *The 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, volume 1, 2016a. 38

P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li. Single shot text detector with regional attention. In *Proceedings of the International Conference on Computer Vision*, volume 6, 2017b. 34

T. He, W. Huang, Y. Qiao, and J. Yao. Accurate text localization in natural image with cascaded convolutional text network. *ArXiv e-prints*, 2016b. 34

T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun. An end-to-end textspotter with explicit alignment and attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 40, 128

W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu. Deep direct regression for multi-oriented scene text detection. In *Proceedings of the International Conference on Computer Vision*, 2017c. 35

J. F. Henriques and A. Vedaldi. Mapnet: An allocentric spatial memory for mapping environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8476–8484, 2018. 24

G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL http://arxiv.org/abs/1503.02531. 24

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997. 38

H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding. Wordsup: Exploiting word annotations for character based text detection. In *Proceedings of the International Conference on Computer Vision*, 2017. 36

W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolution neural network induced mser trees. In *Proceedings of the European Conference on Computer Vision*, 2014. 32

ICDAR 2003 Robust Reading Competition. http://algoval.essex.ac.uk/icdar/datasets.html, 2003. 36

P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Learning visual groups from co-occurrences in space and time. *arXiv preprint arXiv:1511.06811*, 2015. 23

P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 28

M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. In *Workshop on Deep Learning, NIPS*, 2014a. 9, 10, 21, 36, 37, 129

M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In *Proceedings of the European Conference on Computer Vision*, 2014b. 33, 36, 37

M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Deep structured output learning for unconstrained text recognition. In *International Conference on Learning Representations*, 2015a. 36, 37

M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with

convolutional neural networks. *International Journal of Computer Vision*, 2015b. 10, 33, 34, 39, 128

M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015c. 39

M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015d. 35

T. Jakab*, A. Gupta*, H. Bilen, and A. Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Advances in Neural Information Processing Systems*, 2018.

D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. In *Proceedings of the International Conference on Computer Vision*, pages 1413–1421, 2015. 23

D. Jayaraman and K. Grauman. Slow and steady feature analysis: higher order temporal coherence in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3852–3861, 2016. 22

J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016a. 39

J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4565–4574, 2016b. 35

J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1988–1997. IEEE, 2017. 20

A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 31

B. Kaneva, A. Torralba, and W. T. Freeman. Evaluation of image features using a photorealistic virtual world. In *Proceedings of the International Conference on Computer Vision*, pages 2282–2289. IEEE, 2011. 20

D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, L. P. de las Heras, et al. ICDAR 2013 robust reading competition. In *Proc. ICDAR*, pages 1484–1493, 2013. 21, 36

D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, et al. ICDAR 2015 robust reading competition. In *Proc. ICDAR*, pages 1156–1160, 2015. 32, 36

T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *Proceedings of the International Conference on Learning Representations*, 2018. 28

W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 4

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012. 1

E. Kruppa. *Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung*. Hölder, 1913. 4

G. Lample, A. Conneau, L. Denoyer, H. Jégou, et al. Word translation without parallel data. In *Proceedings of the International Conference on Learning Representations*, 2018a. 30

G. Lample, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. In *Proceedings of the International Conference on Learning Representations*, 2018b. 28, 133

T. A. Le, A. G. Baydin, R. Zinkov, and F. Wood. Using synthetic data to train neural networks is model-based reasoning. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 3514–3521. IEEE, 2017. 132

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages II–104, 2004. 20

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015. 1

C. Lee, A. Bhardwaj, W. Di, V. Jagadeesh, and R. Piramuthu. Region-based discriminative feature pooling for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 37

C.-Y. Lee and S. Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. *arXiv preprint arXiv:1603.03101*, 2016. 36, 38

H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the International Conference on Computer Vision*, pages 667–676. IEEE, 2017. 23

V. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics Doklady*, volume 10, page 707, 1966. 39

H. Li, P. Wang, and C. Shen. Towards end-to-end text spotting with convolutional recurrent neural networks. In *Proceedings of the International Conference on Computer Vision*, pages 5238–5246, 2017. 40, 128

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014. 3

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*, pages 21–37. Springer, 2016. 34

Y. Liu and L. Jin. Deep matching prior network: Toward tighter multi-oriented text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3454–3461, 2017. 35

Y. Liu, Z. Wang, H. Jin, and I. Wassell. Synthetically supervised feature learning for scene text recognition. In *Proceedings of the European Conference on Computer Vision*, pages 435–451, 2018. 39

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015a. 9, 33, 34

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015b. 1

S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *Proceedings of the European Conference on Computer Vision*, 2018. 35

D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, pages 1150–1157, Sept. 1999. 7

D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 37

P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *Proceedings of the European Conference on Computer Vision*, pages 67–83, 2018. 35

J. Marin, D. Vázquez, D. Gerónimo, and A. M. López. Learning appearance in virtual scenarios for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 137–144, 2010. 18

J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 384–393, 2002. 32

M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *Proceedings of the International Conference on Learning Representations*, 2015. 28

N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 16

B. McCane, K. Novins, D. Crannitch, and B. Galvin. On benchmarking optical flow. *Computer Vision and Image Understanding*, 84(1):126–143, 2001. 20

J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison. Scenenet RGB-D: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation. In *Proceedings of the International Conference on Computer Vision*, volume 4, 2017. 19

R. Memisevic. Learning to relate images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1829–1846, 2013. 22, 25

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 26, 30

M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 28

A. Mishra, K. Alahari, and C. Jawahar. Scene text recognition using higher order language priors. *Proceedings of the British Machine Vision Conference*, 2012a. 36, 37

A. Mishra, K. Alahari, and C. Jawahar. Top-down and bottom-up cues for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012b. 37

I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *Proceedings of the European Conference on Computer Vision*, pages 527–544. Springer, 2016. 23

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015. 19

H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In *Proceedings of the International Conference on Machine Learning*, pages 737–744. ACM, 2009. 22

L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *Proceedings of the Asian Conference on Computer Vision*, pages 770–783. Springer, 2010. 32

L. Neumann and J. Matas. Text localization in real-world images using efficiently pruned exhaustive search. In *Proc. ICDAR*, pages 687–691. IEEE, 2011. 32

L. Neumann and J. Matas. Real-time scene text localization and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 3, pages 1187–1190, 2012. 32, 37

A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 483–499. Springer, 2016. 8

T. Novikova, O. Barinova, P. Kohli, and V. Lempitsky. Large-lexicon attribute-consistent text recognition in natural images. In *Proceedings of the European Conference on Computer Vision*, pages 752–765. Springer, 2012. 36, 37

N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. 33

A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman. Visually indicated sounds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2405–2413, 2016a. 24

A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba. Ambient sound provides supervision for visual learning. In *Proceedings of the European Conference on Computer Vision*, pages 801–816. Springer, 2016b. 24

M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):448–461, 2010. 33, 37

D. Park and D. Ramanan. Articulated pose estimation with tiny synthetic videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 58–66, 2015. 18

Y. Patel, M. Bušta, and J. Matas. E2e-mlt-an unconstrained end-to-end method for multi-language scene text. *arXiv preprint arXiv:1801.09919*, 2018. 128

D. Pathak, R. B. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 23

V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. *arXiv preprint arXiv:1511.06309*, 2015. 23, 26

X. Peng, B. Sun, K. Ali, and K. Saenko. Learning deep object detectors from 3d models. In *Proceedings of the International Conference on Computer Vision*, pages 1278–1286, 2015. 18

B. Pepik, M. Stark, P. Gehler, and B. Schiele. Multi-view and 3d deformable part models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2232–2245, 2015. 18

L. Pishchulin, A. Jain, C. Wojek, M. Andriluka, T. Thormählen, and B. Schiele. Learning people detection models from few training samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1473–1480, 2011. 18

L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, and B. Schiele. Articulated people detection and pose estimation: Reshaping the future. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3178–3185, 2012. 18

I. Posner, P. Corke, and P. Newman. Using text-spotting to query the world. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010. 5, 32

A. Poznanski and L. Wolf. Cnn-n-gram for handwriting word recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 36, 37

T. Quack. *Large scale mining and retrieval of visual data in a multimodal context.* PhD thesis, ETH Zurich, 2009. 32

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 28

D. Ramanan. Learning to parse images of articulated bodies. In *Advances in Neural Information Processing Systems*, pages 1129–1136, 2007. 8

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016a. 40

J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016b. 9, 33, 128

S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee. Deep visual analogy-making. In *Advances in Neural Information Processing Systems*, pages 1252–1260, 2015. 25, 26

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 39, 40

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2016. 34, 128

S. R. Richter and S. Roth. Discriminative shape from shading in uncalibrated illumination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1128–1136, 2015. 20

S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In *Proceedings of the European Conference on Computer Vision*, pages 102–118. Springer, 2016. 19

S. R. Richter, Z. Hayder, and V. Koltun. Playing for benchmarks. In *Proceedings of the International Conference on Computer Vision*, volume 2, 2017. 19

J. A. Rodriguez-Serrano, A. Gordo, and F. Perronnin. Label embedding: A frugal baseline for text recognition. *International Journal of Computer Vision*, 113(3):193–207, 2015. 36, 37

G. Rogez and C. Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. In *Advances in Neural Information Processing Systems*, pages 3108–3116, 2016. 18

X. Rong, C. Yi, and Y. Tian. Unambiguous text localization and retrieval for cluttered scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3279–3287. IEEE, 2017. 35

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015. 35

G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The Synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016. 19

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, S. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and F. Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015. 3

W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. 7

M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 1997. 38

T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3633–3642. ACM, 2015. 16

B. Shi, X. Bai, and C. Yao. An end-to-end trainable neural network for image-based

sequence recognition and its application to scene text recognition. *ArXiv e-prints*, 2015. 36, 38

B. Shi, X. Wang, P. Lv, C. Yao, and X. Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 36, 39

B. Shi, X. Bai, and S. Belongie. Detecting oriented text in natural images by linking segments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 34

C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang. Scene text recognition using part-based tree-structured character detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 37

B. Shillingford, Y. Assael, M. W. Hoffman, T. Paine, C. Hughes, U. Prabhu, H. Liao, H. Sak, K. Rao, L. Bennett, et al. Large-scale visual speech recognition. *arXiv preprint arXiv:1807.05162*, 2018. 4

J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 16

A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3, 19, 131

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 33, 34

J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, 2003. 7

J. A. Sommerville and A. L. Woodward. Pulling out the intentional structure of action: the relation between action processing and action production in infancy. *Cognition*, 95 (1):1–30, 2005. 4

J. A. Sommerville, A. L. Woodward, and A. Needham. Action experience alters 3-month-old infants' perception of others' actions. *Cognition*, 96(1):B1–B11, 2005. 4

N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *Proceedings of the International Conference on Machine Learning*, pages 843–852, 2015. 22

B. Su and S. Lu. Accurate scene text recognition based on recurrent neural network. In *Proceedings of the Asian Conference on Computer Vision*, 2014. 36, 38

H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for CNN: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the International Conference on Computer Vision*, 2015. 19

J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan. Depth-based hand pose estimation: data, methods, and challenges. In *Proceedings of the International Conference on Computer Vision*, pages 1868–1876, 2015. 20

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014. 1, 37, 38

I. Sutskever, R. Jozefowicz, K. Gregor, D. Rezende, T. Lillicrap, and O. Vinyals. Towards principled unsupervised learning. In *In ICLR Workshop*, 2015. 30

S. Suwajanakorn, N. Snavely, J. Tompson, and M. Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Advances in Neural Information Processing Systems*, 2018. 27, 133

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 34

G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *Proceedings of the European Conference on Computer Vision*, pages 140–153. Springer, 2010. 22, 25

Tesseract OCR. https://github.com/tesseract-ocr/, 1985 – 2018. 10, 21

J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proceedings of the International Conference on Computer Vision*, 2017a. 8, 26, 27

J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *Advances in Neural Information Processing Systems*, pages 844–855, 2017b. 27

S. Tian, S. Lu, and C. Li. Wetext: Scene text detection under weak supervision. In *Proceedings of the International Conference on Computer Vision*, 2017. 35

Z. Tian, W. Huang, T. He, P. He, and Y. Qiao. Multi-oriented text detection with fully convolutional networks. In *Proceedings of the European Conference on Computer Vision*, 2016. 34

J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33 (5):169, 2014. 20

H.-Y. F. Tung, A. W. Harley, W. Seto, and K. Fragkiadaki. Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision. In *Proceedings of the International Conference on Computer Vision*, volume 2, 2017. 5, 31, 133

J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013. 33

G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 17, 18

D. Vazquez, A. M. Lopez, J. Marin, D. Ponsa, and D. Geronimo. Virtual and real world adaptation for pedestrian detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4):797–809, 2014. 18

S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017. 24

R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. In *Proceedings of the International Conference on Machine Learning*, 2017. 25, 26

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001. 33

C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy. Tracking emerges by colorizing videos. In *Proceedings of the European Conference on Computer Vision*, 2018. 27

G. Wahba. *Spline models for observational data*, volume 59. Siam, 1990. 25

F. Wang, L. Zhao, X. Li, X. Wang, and D. Tao. Geometry-aware scene text detection with instance transformation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1381–1389, 2018. 35

J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014. 23

K. Wang and S. Belongie. Word spotting in the wild. In *Proceedings of the European Conference on Computer Vision*, 2010. 36, 37

K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *Proceedings of the International Conference on Computer Vision*, pages 1457–1464, 2011. 33, 36, 37

T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *Proceedings of the International Conference on Pattern Recognition*, pages 3304–3308. IEEE, 2012. 21, 33, 36, 37

X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the International Conference on Computer Vision*, pages 2794–2802, 2015. 23

Z. Wang, J. Yang, H. Jin, E. Shechtman, A. Agarwala, J. Brandt, and T. S. Huang. Deepfont: Identify your font from an image. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 451–459. ACM, 2015. 22

M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proceedings of the European Conference on Computer Vision*, pages 18–32, 2000. 8, 133

D. Wei, J. Lim, A. Zisserman, and W. T. Freeman. Learning and using the arrow of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8052–8060, 2018. 23

O. Wiles, A. Koepke, and A. Zisserman. Self-supervised learning of a facial attribute embedding from video. In *Proceedings of the British Machine Vision Conference*, 2018a. 26

O. Wiles, A. S. Koepke, and A. Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European Conference on Computer Vision*, 2018b. 27

L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002. 4, 22

E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 131–138. ACM, 2016. 19

J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016. 28

Y. Wu and P. Natarajan. Self-organized text detection with minimal post-processing via border learning. In *Proceedings of the International Conference on Computer Vision*, 2017. 35

B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner. TORCS, The Open Racing Car Simulator. http://www.torcs.org, 2014. 19

F. Xia, A. R. Zamir, Z.-Y. He, A. Sax, J. Malik, and S. Savarese. Gibson env: real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 19

Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 75–82, 2014a. 19

Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 75–82. IEEE, 2014b. 8

C. Xue, S. Lu, and F. Zhan. Accurate scene text detection through border semantics

awareness and bootstrapping. In *Proceedings of the European Conference on Computer Vision*, pages 355–372, 2018. 35

C. Yao, X. Bai, B. Shi, and W. Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 36, 37

C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao. Scene text detection via holistic, multi-channel prediction. *ArXiv e-prints*, 2016. 34

I. Yildirim, T. D. Kulkarni, W. A. Freiwald, and J. B. Tenenbaum. Efficient and robust analysis-by-synthesis in vision: A computational framework, behavioral tests, and modeling neuronal representations. In *Annual conference of the cognitive science society*, 2015. 20

X.-C. Yin, X. Yin, and K. Huang. Robust text detection in natural scene images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):970–983, 2013. 32

F. Zhan, S. Lu, and C. Xue. Verisimilar image synthesis for accurate detection and recognition of texts in scenes. In *Proceedings of the European Conference on Computer Vision*, 2018. 22, 131

M. Zhang, Y. Liu, H. Luan, and M. Sun. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1959–1970, 2017. 5, 30, 133

Y. Zhang, Y. Guo, Y. Jin, Y. Luo, Z. He, and H. Lee. Unsupervised discovery of object landmarks as structural representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2694–2703, 2018. 8, 25, 130

Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai. Multi-oriented text detection

with fully convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 34

T. Zhou, Y. Jae Lee, S. X. Yu, and A. A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1191–1200, 2015. 8

T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017a. 24

X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2017b. 35

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the International Conference on Computer Vision*, 2017. 4, 28, 29

S. Zhu and R. Zanibbi. A text detection system for natural scenes with convolutional feature learning and cascaded classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 32

C. L. Zitnick and P. Dollar. Edge boxes: Locating object proposals from edges. In *Proceedings of the European Conference on Computer Vision*, pages 391–405, 2014a. 33

L. Zitnick and P. Dollar. Edge boxes: Locating object proposals from edges. In *Proceedings of the European Conference on Computer Vision*, 2014b. 128

W. Zou, S. Zhu, K. Yu, and A. Y. Ng. Deep learning of invariant features via simulated fixations in video. In *Advances in Neural Information Processing Systems*, pages 3203–3211, 2012. 22