Self-supervised and Cross-modal Learning from Videos

Almut Sophia Koepke
Lincoln College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Supervised by Prof Andrew Zisserman

Michaelmas Term 2019

Self-supervised and Cross-modal Learning from Videos

Almut Sophia Koepke

Lincoln College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Michaelmas Term 2019

Deep learning has demonstrated impressive results for tasks where the training of neural networks can be supervised with paired input and manually labelled output data. However, labelling data can be expensive and might not be feasible for some applications. In this thesis, we consider learning from video data using less supervision than standard supervised learning methods. In particular, we focus on self-supervised learning, where the data itself provides supervision without requiring manually annotated labels, and cross-modal audio-visual learning.

Firstly, we propose two self-supervised frameworks: one for controlling face generation, and another for obtaining a meaningful face representation. We use a proxy task to train both frameworks in a self-supervised way which exploits the knowledge that two frames belong to the same video track. It consists of learning to warp one frame into another, and we can leverage a large video dataset of talking faces for training. We demonstrate the effectiveness of this proxy task for driving face generation with a face image of the same or of a different person. This same framework can be used to control face generation with head pose vectors or audio representations through vector arithmetic in the embedding space. Our other proposed framework distils information about facial attributes into a face embedding which can be used for facial landmark prediction, head pose estimation, and facial expression estimation.

Secondly, we consider the task of visual music transcription which aims to generate audio information from visual information alone. This task is similar to lipreading in the speech domain, and it can be particularly useful for enhancing audio information in the presence of noise or when aiming to separate sounds of different instruments. We pose visual music transcription as a cross-modal learning problem where audio information is used to supervise learning from visual inputs, exploiting the natural co-occurrence of audio and visual signals in videos. We present two frameworks for transcribing music from videos of violin and piano playing respectively, which are trained with pseudo ground-truth labels obtained from audio-based pitch estimation methods. Due to the nature of the piano, the starting points of notes (note onsets) give a clear audio signal whose energy decays quickly. This makes it difficult to derive pseudo ground-truth note endings (note offsets) from audio information. Therefore, in the case of piano playing, we focus on predicting note onsets only. For violin playing, on the other hand, we predict not just the starts of notes, but also the note durations. We curate new datasets of violin and piano playing which consist of video recordings in constrained settings and of in-the-wild videos downloaded from YouTube. For violin playing, the in-the-wild videos exhibit significant variation in viewpoints and body pose of the violinists; for piano playing we only consider top-view recordings.

Statement of Originality

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment 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.

Almut Sophia Koepke, Lincoln College

Acknowledgements

Above all, I would like to thank my supervisor, Prof Andrew Zisserman, for his guidance and support during the past four years. A circuitous path of studies via music and mathematics eventually brought me to Oxford, where Andrew gave me the invaluable opportunity to pursue research in the field of computer vision. I am very grateful to Andrew for everything I have learned through working with him. His encouragement to pursue problems that I am passionate about has allowed me to return to music by researching audio-visual problems.

My special thanks goes to Prof Yael Moses for many fruitful discussions, and for hosting me at the IDC in Herzliya. I would also like to thank Dr Daniel Crispell, who was very patient when collaborating with me during the initial phases of my DPhil.

Furthermore, I am fortunate to have had the privilege to learn from and with the members of the Visual Geometry Group, past and present. In particular, I would like to thank Olivia for many hours of discussing and collaborating on projects in the course of my DPhil.

I wish to thank my examiners Prof David Murray and Dr Hakan Bilen for reading this thesis and providing very valuable feedback.

I am extremely grateful for the invaluable support of friends, both in Oxford and from afar.

Finally, I would like to thank my family and especially my parents for their love, support, and everything they have done for me. This would not have been possible without them.

And thank you to you, the curious reader of these acknowledgements (and this thesis). Do not hesitate to get in touch should you have any questions or comments about the work presented in these pages.

Contents

1 Introduction **1**

1.1 Outline and contributions 3

1.2 Publications 9

2 Literature review **10**

2.1 Self-supervised image representation learning 10

2.1.1 Learning from images 11

2.1.2 Learning from videos 12

2.1.3 Learning of class representations 14

2.2 Cross-modal audio-visual learning 15

2.2.1 Joint training of audio and visual networks 17

2.2.2 Audio-visual learning for music 19

3 X2Face: A network for controlling face generation by using images, audio, and pose codes **23**

4 Self-supervised learning of a facial attribute embedding from video **49**

5 Visual pitch estimation for violin playing **72**

6 Sight to sound: An end-to-end approach for visual piano transcription **80**

7 Self-supervised cross-modal learning for music **87**

7.1 Motivation 87

7.2 Method 88
7.2.1 Model 88
7.2.2 Dataset and training details 89
7.3 Experiments 90
7.4 Discussion 90

8 Summary and extensions **92**

A Self-supervised learning of class embeddings from video **99**

Bibliography **110**

List of Abbreviations

AMT	Automatic Music Transcription
AU	(facial) Action Unit
AUC	Area Under the Curve
CNN	Convolutional Neural Network
FAb-Net	Facial Attributes Network
fps	frames per second
GAN	Generative Adversarial Network
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MIDI	Musical Instrument Digital Interface
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristic
SGD	Stochastic Gradient Descent
VMT	Visual Music Transcription
3DMM	3D Morphable Model

Supplementary material

Video examples, datasets, and further details can be found on the following project pages.

Chapter 3	`https://www.robots.ox.ac.uk/~vgg/research/unsup_learn_watch_faces/x2face.html`
Chapter 4	`https://www.robots.ox.ac.uk/~vgg/research/unsup_learn_watch_faces/fabnet.html`
Chapter 5	`https://www.robots.ox.ac.uk/~vgg/research/sighttosound/violinpitch.html`
Chapter 6	`https://www.robots.ox.ac.uk/~vgg/research/sighttosound/`

Chapter 1

Introduction

As humans, we process what we see, hear, taste, touch, or smell, in order to understand the world around us. Very early on, our brains start to learn to convert sensory inputs into representations of the structure around us, which are interpreted in turn to determine our actions.

The following example serves as an illustration of analogies between human learning[1] and learning in computer vision. It showcases the relevance of associating co-occurring audio-visual information and of seeing movement, which is particularly relevant for cross-modal learning from audio-visual data and in self-supervised learning.

A significant amount of learning for infants is done through just observing [Meltzoff, 2007]. If someone shakes a rattle within a baby's field of view, the baby might observe and learn that the sound of the rattle and the rattle itself occur together. This natural co-occurrence of audio and visual information is exploited in cross-modal audio-visual learning.

Watching the movement (of the rattle) is a very strong cue to understand the object structure [Johnson *et al.*, 2003]. Parts moving together on the same trajectory are very likely to be parts of the same object. The baby does not receive any labelled supervisory signal when she learns to track the rattle. However, she still might recognise the rattle the next time she sees it, although she might not be able to call it a "rattle" yet. Humans might use language to describe things to the baby and label the rattle with the word "rattle". Even with a relatively small number of examples, at some point, the baby might learn to associate her internal representation (that allows her to recognise a rattle) with the word and category label ("rattle") [Nazzi and Gopnik, 2001].

[1] This is an active area of research in developmental science. We are aware that there also are other theories as to how babies process information and learn.

A similar way of learning is used in self-supervised learning where proxy tasks which do not require manual labelling (such as tracking the rattle in the above example) are used to learn useful feature representations. The resulting self-supervised representations can then be used for different downstream tasks, by mapping the representation to labelled data (e.g. mapping the representation for the rattle to the label "rattle"). This typically requires fewer (manual) annotations.
Self-supervised learning methods are regarded as being unsupervised, since no manual labels are necessary to train them. Obtaining manual labels can be hard and expensive, for example when it requires experts to annotate data. Instead, proxy tasks are designed to learn invariant properties that will be useful in solving the downstream task. Large datasets (e.g. video data) without annotations can be leveraged for training. Convolutional Neural Network (CNN) frameworks trained fully supervised might exhibit better performances on the specific datasets and tasks that they were trained for than self-supervised methods. However, annotated datasets for specific tasks are limited in size, which can result in overfitting to the particular dataset and the frameworks not being easily transferable to other datasets. Hénaff *et al.* [2020] and He *et al.* [2020] show that their self-supervised representations can achieve better object detection accuracy on PASCAL VOC2007 [Everingham *et al.*, 2007] than supervised representations where both, the self-supervised and the supervised representations, were pre-trained on the ImageNet [Deng *et al.*, 2009] dataset. This shows that self-supervised features can result in more general and transferable representations of the data.

In this thesis, we will explore CNN frameworks for self-supervised learning and cross-modal audio-visual learning. Our work is divided into two parts.
In the first part, we aim to learn a self-supervised face representation which can be used for a variety of tasks. Our proposed self-supervised proxy task, which allows us to leverage a large video dataset of talking faces, consists of learning to warp one frame from a video face track into another frame from the same face track. This is analogous to learning to densely track image points which the baby performs when following a rattle with her eyes. We show that our proposed proxy task indeed results in meaningful face representations which allows us to control and manipulate face generation and to obtain information about facial attributes (such as facial landmarks, head pose, or facial expressions).
In the second part, we aim to extract audio information from silent videos of people playing musical instruments. Obtaining audio information from visual information,

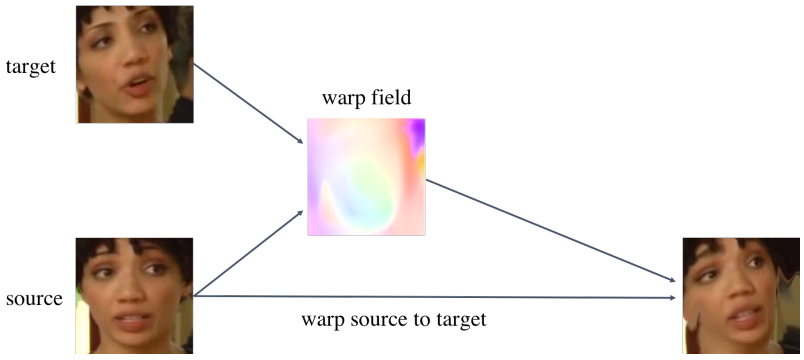


Figure 1.1: Overview of the self-supervised proxy task proposed in chapter 3 and chapter 4. A network learns to predict the warp field to transform a source into a target frame where both frames come from the same video face track. Establishing dense per-pixel correspondences as a proxy task proves to be useful for understanding the object structure.

particularly transcribing music from visual information can be very useful, for instance, when the audio is missing, for polyphonic music (i.e. when more than one note is sounding at the same time), or in the presence of noise. In those situations, audio-based pitch estimation methods fail. We use the natural co-occurrence of audio and visual information to learn to transcribe music from the corresponding visual information of violin playing and piano playing. Finally, we show that we can use self-supervised learning to learn to synchronise audio and visual data for piano playing. This opens many avenues for further research into learning from cross-modal self-supervision.

1.1 Outline and contributions

Self-supervised learning (chapter 3 and chapter 4): Inferring the full object structure from only a still image without prior information is extremely difficult. From a single, frontal view of a human face, it is very challenging to understand the 3D shape of the face very accurately - the shape of the nose, for instance, will be ambiguous from a frontal view only. However, if we are given multiple images of the face from different views, we might be able to better understand the structure of that face. For this, matching reference points across the different views is essential. Watching objects move and tracking relevant reference points amounts to establishing correspondences between multiple views of the same object. Inspired by this observation, we leverage video data of faces of people talking and moving in order to learn

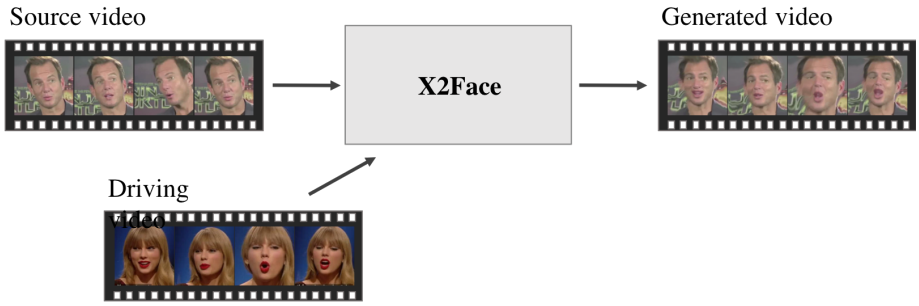


Figure 1.2: Overview of *X2Face* presented in chapter 3. The network learns face puppeteering of a source identity with a driving target video sequence. *X2Face* can also be used for driving face generation with target head poses or corresponding audio.

about the structure of faces.

In chapter 3 and chapter 4, we propose the self-supervised proxy task of learning to warp one face image into another frame from the same video face track. This is done by learning to predict dense per-pixel correspondences between the source and target frame that determine bilinear sampling locations in the source frame to generate the target frame. The proxy task is illustrated in figure 1.1.

In chapter 3, we present *X2Face*, a neural network model that is trained for our proposed proxy task in a self-supervised manner. At test time, *X2Face* can be used to drive the face of one person (source) using the movements of a face of another person (target). The generated frame will have the identity of the source frame with the target's pose and expression (see figure 1.2). This could be especially useful, for instance, to modify a face along with a simultaneous translation of the speech to another language, resulting in a coherent audio-visual translation. In addition to this, our setup allows us to drive face generation with another modality, such as head pose or audio.[2] This is done by a simple vector addition in the embedding space: we linearly map the embedding to a bottleneck of interpretable pose angles, and back to the embedding space. Ideally, this pose angle bottleneck would result in a backward mapped embedding that encodes pose information only. By removing the original pose content and adding on the desired backward mapped pose in the embedding space, we can

[2]This inspired the choice to name this framework *X2Face*. X is mapped onto a face, with $X \in \{\text{driving face image}, \text{head pose}, \text{audio}\}$.

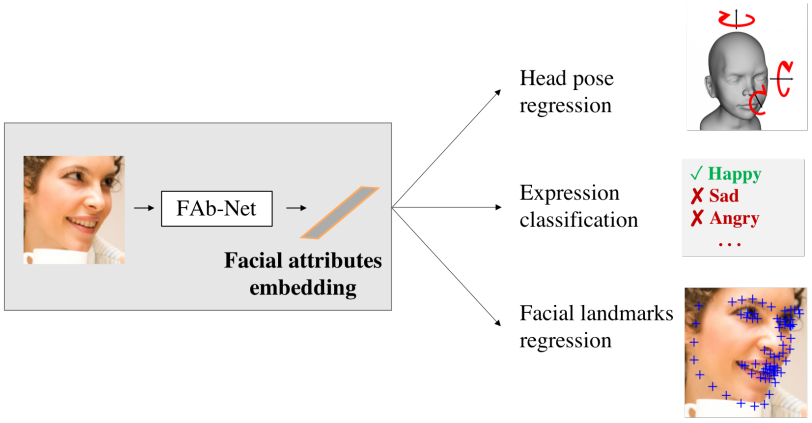


Figure 1.3: Overview of *F*acial *Attri*butes-*Net* (*FAb-Net*) presented in chapter 4. *FAb-Net* learns to map a face image to a meaningful embedding. The embedding can then be used for downstream tasks, such as head pose, facial landmarks, and facial expression estimation. This demonstrates that self-supervised training for the proxy task distils useful information into the embedding space.

drive face generation to our target pose. The same mapping can be done with other modalities, such as for instance with audio representations. Furthermore, our model naturally learns to disentangle the texture from the pose and expression of a face. The obtained texture representation can be modified and used for lightweight video editing.

In chapter 4, we propose *FAb-Net*, a self-supervised framework that learns a face embedding which encodes information about *F*acial *Attri*butes, such as head pose, facial landmarks, and facial expression (see figure 1.3). It is trained for the same proxy task as *X2Face*. The embedding space is the only point in the setup where information from the source and target frames is shared. Therefore, all the information that is necessary to determine how to warp the source into the target frame has to be encoded in the respective face embeddings.

Cross-modal learning (chapter 5 and chapter 6): An early study by Sumby and Pollack [1954] investigated the contribution of visual stimuli to human aural understanding. They found that visual information can enhance aural intelligibility, in particular in the presence of noise. This is relevant not only for speech, but also, for instance, for music performances. The (visible) movement of the body and musical instrument influence the produced sound. We investigate this by training

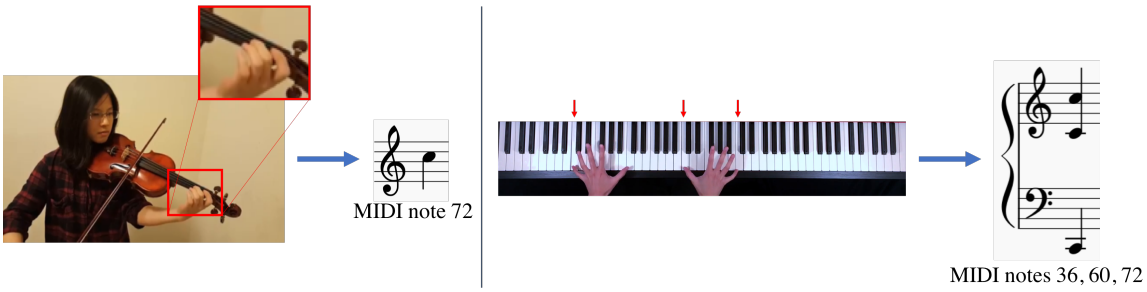


Figure 1.4: Overview over the visual music transcription task for violin and piano playing presented in chapter 5 and chapter 6 respectively. Red arrows mark the piano keys that are being pressed in the displayed frame.

neural networks that predict the audio information given the corresponding visual information only.

In chapter 5 and chapter 6, we present frameworks that predict sound data from visual input of violin and piano playing respectively (see figure 1.4). Attributes such as loudness, pitch, and timbre are commonly used to describe musical sounds. In our works we focus on estimating pitch from visual information (visual music transcription). One can consider our frameworks as imitations of perfect pitch in humans, using visual input instead of audio input. The pitch of a musical sound reflects how "high" or "low" a perceived sound is. For string instruments, the sounding pitch corresponds to the lowest frequency at which a string vibrates and is also referred to as fundamental frequency. However, the sound produced by a musical instrument usually consists of many related frequency components (so-called harmonics) which are integer multiples of the fundamental frequency. The presence of harmonics can make estimating pitch from audio information alone difficult, particularly in cases where multiple notes sound simultaneously. We treat the pitch estimation problem as a classification problem with discrete pitch classes by representing the pitch with its associated MIDI note.

In chapter 5, we present a framework that learns to predict pitch from video frames of violin playing. The violin belongs to the family of bowed string instruments. Commonly, sound is produced by drawing a bow across the strings resulting in their vibration; alternatively, the strings can be plucked. For a detailed study on how sound is produced on a violin, we refer the interested reader to [Chaigne and Kergomard, 2016]. Each of the four open strings of the violin has a different pitch and is controlled with a tuning peg. The fingers of the violinist's left hand can be placed on the strings

Figure 1.5: Challenging example for visual music transcription for piano playing. The fingers occlude the relevant keys that are being played.

to shorten the sounding strings, resulting in higher pitches. The sounding pitch is determined by the combination of which string(s) the bow, controlled by the violinist's right hand, is playing on, and the position of the fingers of the violinist's left hand. It is not enough to look at the left hand, as fingers could be resting on strings that are not producing any sound and therefore are not relevant for pitch estimation. Our proposed framework is learnt using a teacher-student training strategy which distils information from a network that predicts pitch from audio information to a visual network that predicts the same from visual information only. Additional loss functions encourage the features from the visual student network to be close to the ones from the audio teacher network. The visual network is used at test time to predict pitch from visual information alone. Pseudo ground-truth pitch information to train these networks is obtained from an audio pitch estimation method. Our framework is trained and tested in both very constrained settings, where there is little variation in the pose and appearance of the violinist, and also on videos of various musicians in a wider range of settings and poses. For this, we curated a dataset of solo violin playing from YouTube and we recorded a smaller set of videos in constrained settings (i.e. with less variation in viewpoint and with one violinist only). Deriving the pitch from just watching a person playing the violin can be very challenging, especially for more extreme viewpoints, where it is not clear which of the strings is being played and whether the fingers of the left hand are active in producing the sound or just resting.

In chapter 6, we address the problem of obtaining note onsets for videos of piano playing. The nature of the piano results in rather clear starting points of sounding notes (note onsets) but makes it hard to determine when notes are ending. For an acoustic piano, a note onset occurs when a piano hammer strikes a string. Each hammer is controlled by a key on the keyboard, and note onsets can therefore be

inferred from the finger movements on the keyboard. However, the duration of a note is more difficult to determine, since the energy of a sounding note decays quickly and its duration can additionally be affected by using a sustain pedal. In that case, a key does not need to be continued to be pressed by a finger to sustain a note. For a detailed account of the physics of the piano, we refer to [Giordano, 2010]. We consider the task of predicting note onsets from top-view piano video recordings. As there was no dataset readily available to learn this task, we curated a dataset of top-down piano recordings from YouTube. As those do not have note annotations, we used an audio-based method to estimate note onsets from the audio data. Those weak pseudo-labels served as supervision to train our networks. In order to test our networks, we additionally recorded a smaller set of videos with ground-truth MIDI data that was obtained using a digital piano with MIDI output. Visual music transcription can be difficult, in particular when the hands occlude the keys. We show one such example in figure 1.5.

Self-supervised cross-modal learning for music (chapter 7): The natural co-occurrence of audio and visual signals provides a strong learning signal that can be leveraged for self-supervision. In chapter 7, we present our ongoing work on using cross-modal self-supervision from videos of piano playing. We propose a framework that learns to detect synchronisation between audio and visual data without requiring any manual annotations.

1.2 Publications

This thesis is an integrated thesis, which (according to the University's guidelines) can consist of conventional chapters and scientific papers, or be fully paper-based. The following papers are included in this thesis and presented in the format of their original publication. * denotes equal contribution, i.e. both authors contributed equally to conception, implementation, experiments, and paper writing.

> O. Wiles*, A. S. Koepke*, and A. Zisserman. X2Face: A network for controlling face generation by using images, audio, and pose codes. In *European Conference in Computer Vision,* 2018. [Wiles *et al.*, 2018b]

> O. Wiles*, A. S. Koepke*, and A. Zisserman. Self-supervised learning of a facial attribute embedding from video. In *British Machine Vision Conference*, 2018. [Wiles *et al.*, 2018a]

> A. S. Koepke, O. Wiles, and A. Zisserman. Visual pitch estimation. *Sound and Music Computing Conference*, 2019. [Koepke *et al.*, 2019]

> A. S. Koepke, O. Wiles, Y. Moses, and A. Zisserman. Sight to sound: An end-to-end approach for visual piano transcription. In *International Conference on Acoustics, Speech, and Signal Processing*, 2020. [Koepke *et al.*, 2020]

The following publication is included in the appendix to this thesis.

> O. Wiles, A. S. Koepke, and A. Zisserman. Self-supervised learning of class embeddings from video. In *International Conference in Computer Vision Workshops*, 2019. [Wiles *et al.*, 2019]

Chapter 2

Literature review

In this chapter, we review related work in the areas of self-supervised and cross-modal learning.
We first give an overview of methods using self-supervised learning from image and video data in chapter 2.1.1 and chapter 2.1.2 respectively. Those works are closely related to our proposed self-supervised proxy task (used in chapter 3 and chapter 4), and in particular to our work on controlling face generation (chapter 3). Next, we consider self-supervised methods for learning class-specific representations in chapter 2.1.3, which are very relevant to our work on learning class-specific representations for faces in chapter 4 and for representations for other classes, such as humans and horses in appendix A. For a more detailed survey of self-supervised learning with deep learning, we refer the reader to the survey by Jing and Tian [2019].
Furthermore, we summarise audio-visual cross-modal learning methods in chapter 2.2.1 with a particular focus on music applications in chapter 2.2.2; these are most relevant to our work on estimating sound from visual information alone, which is presented in chapter 5 and chapter 6.
In chapter 8, we discuss related work that was published since our papers.

2.1 Self-supervised image representation learning

Self-supervised learning commonly consists of learning auxiliary tasks that use information contained in the data itself as supervision. As no manual annotations are used, self-supervised learning methods can be regarded as unsupervised learning methods. Those auxiliary tasks are typically referred to as *proxy* or *pretext tasks*. We believe that the term *proxy* better describes the tasks we are considering than the

word *pretext*[1].

The aim of learning to solve the proxy tasks is to learn feature representations that capture meaningful information about the data. Useful feature representations should encode an input in such a way that statistical dependencies in the representation and thereby redundancy is reduced [Barlow and others, 1961; Barlow, 1989; Olshausen and Field, 1997]. Representations resulting from task-specific supervision may not generalise well to other tasks. Self-supervised training has the advantage of producing more general representations of the data which can be used for different tasks and domains [Hénaff *et al.*, 2020; He *et al.*, 2020]. Furthermore, large-scale data can be leveraged without requiring costly manual annotations. Self-supervised representations are commonly evaluated on downstream tasks by learning a linear classifier from the self-supervised representation to labels of an annotated dataset. In particular, features learned using large quantities of available data can be transferred to tasks/domains with only smaller annotated datasets, for instance in areas where annotations are difficult and expensive to obtain (for example, when labelling requires expert knowledge). In the following, we distinguish, similar to Jing and Tian [2019], between *generation-based* and *context-based* self-supervised methods.

2.1.1 Learning from images

Generation-based methods: One example of learning in a self-supervised way is the use of autoencoders which encode an input to a feature representation and then decode the same to reconstruct the input [Hinton and Salakhutdinov, 2006]. However, the learnt representation can be improved by tasking the network to perform other proxy tasks that are more difficult than to simply reconstruct the input. This can be done, for example, by removing or modifying some parts of the input data and tasking the network to reconstruct the perturbed parts.

In the following we give an incomplete summary of different methods that use self-supervised proxy tasks to learn useful representations from images.

Image data itself can provide supervision, for example for generative tasks, such as inpainting images [Pathak *et al.*, 2016]. Zhang *et al.* [2016] learn to colourise greyscale images. Zhang *et al.* [2017] build on this and propose the *split-brain autoencoder* which, in addition to solving the colourisation task (which in the *Lab* colour space consists of predicting the a and b channels from the L channel), also performs the

[1] *Pretext* is described in the Oxford University Press *Lexico* dictionary as "A reason given in justification of a course of action that is not the real reason", *proxy* as "A figure that can be used to represent the value of something in a calculation" [OUP, Oxford University Press 2019].

opposite (predicting the L channel from the a and b channels). We compare our self-supervised face representations learned with *FAb-Net* in chapter 4 to the *split-brain autoencoder* representation.

Context-based methods: Several proxy tasks have been proposed that exploit the spatial structure and context in images, such as predicting the relative location of an image patch in the image [Doersch *et al.*, 2015] or solving a jigsaw puzzle [Noroozi and Favaro, 2016].
Another effective image-level task is to transform images via a synthetic rotation or translation and to train a CNN to classify the transformation [Gidaris *et al.*, 2018; Agrawal *et al.*, 2015]. A series of frameworks apply synthetic warps to images in order to learn equivariant pixel embeddings [Novotny *et al.*, 2018; Thewlis *et al.*, 2017b,a] which can serve as class representations. Similar to these methods, our self-supervised proxy task also must also learn to model deformations of an object (a face) between two input images. However, our two input images are simply taken from the same video track and therefore the transformation from one image to another is not given by a known synthetic deformation. Instead, we formulate our proxy task as a generative task, where the network has to determine how to warp the source frame to obtain the target frame. As a result, the learned representation is more powerful when transferred to prediction tasks which require knowledge of these deformations (e.g. pose/expression prediction for faces). We will review methods which aim at learning class representations in more detail in chapter 2.1.3.

2.1.2 Learning from videos

Video data is available in large amounts (e.g. online on platforms like YouTube) and can be leveraged to train visual representations without the need for manual labelling. The use of dynamic input for learning is closer to the visual input used by humans for learning than static labelled image data. Video data has inherent spatial and temporal structure that can be leveraged when learning without manual annotations.

Context-based methods: Common proxy tasks for self-supervised learning from video data make use of the temporal ordering of frames. Földiák [1991] learns features that are invariant to the spatial movement of a pattern using the assumption that features should be stable across time. Becker [1993] also uses the assumption of temporal coherence in image sequences to expand the transformation-invariance to

more challenging object recognition by maximising mutual information between class predictions at neighbouring time steps.

Misra *et al.* [2016] consider the correctness of the temporal ordering of video frames as a binary classification problem which serves as their proxy task. Fernando *et al.* [2017] also exploit the arrow of time and propose *Odd-One-Out* networks trained with a multi-way classification loss to recognise video sequences with wrong temporal order. Their setup exploits the temporal consistency across nearby video frames as a training signal. Lee *et al.* [2017] learn to not only recognise incorrect temporal ordering in a video sequence, but they also learn to order a shuffled video sequence. This results in a rich image representation that can be used for both, video and image tasks. Xu *et al.* [2019a] learn to order shuffled clips instead of shuffled frames by using 3D CNNs to extract features for the clips from which the actual ordering of the video clips is predicted. Their self-supervised pre-training beats ImageNet pre-training when finetuning for action recognition. Dwibedi *et al.* [2019] introduce *temporal cycle consistency learning*, a self-supervised proxy task which learns to align video sequences of the same action. Learning is achieved by minimising a cycle consistency error which consists of finding nearest-neighbour representations of frames from pairs of frame sequences in the embedding space. The cycle consistency error is minimised when cycling from a frame representation in the first sequence to the nearest-neighbour representation in another sequence and back to the nearest-neighbour in the first sequence brings us back to the frame representation in the first sequence that we started from, closing the cycle.

Another method that leverages video data to learn image representation is [Wang and Gupta, 2015], which uses a triplet ranking loss to learn to track patches in the video by enforcing that tracked patches have more similar feature representations than random patches.

Generation-based methods: However, more related to our proposed proxy task are generative tasks that use frame synthesis to learn useful representations. Srivastava *et al.* [2015] learn to generate both the input sequence and a future sequence using a LSTM encoder-decoder framework. Denton and Birodkar [2017] learn to predict future frames by factorising frame representations into content and pose representations using adversarial loss functions for training. Future frames are generated from the pose and content representations using a U-Net [Ronneberger *et al.*, 2015] style LSTM decoder. Villegas *et al.* [2017] also separate content and motion representations. They use image differences as inputs for obtaining the motion representation

which is combined with a content representation from a single frame to generate the next frame. However, these approaches are deterministic, which might not be the most suitable choice for the prediction of uncertain future events.
Another line of work that addresses frame synthesis as a self-supervised task or proxy task builds on the Spatial Transformer module proposed by Jaderberg *et al.* [2015]. This learns to transform feature maps conditioned on the input without requiring direct supervision for the transformation itself. Similar to our *X2Face* and *FAb-Net* frameworks, Finn *et al.* [2016] predict affine image transformations which determine a warping field to generate future frames, encouraging their model to focus on motion rather than object appearances. However, they consider solely the task of action-conditioned frame synthesis and do not evaluate their learnt representations on other downstream tasks. Even more closely related to our proposed self-supervised proxy task is the framework by Pătrăucean *et al.* [2016] which learns a dense transformation map that warps the current frame into the next frame. They extend the Spatial Transformer module to a dense grid such that it consists of per-pixel transformations instead of a single transformation for the full image. Their motion prediction is conditioned on a number of past frames. In contrast, our proxy task of warping one frame into another frame from the same video track does not leverage the temporal ordering present in a video. In contrast, [Xue *et al.*, 2016; Jia *et al.*, 2016] explicitly predict the motion between frames as a convolutional kernel.

2.1.3 Learning of class representations

A useful class representation should ideally capture as much information about the class as possible. We use the term *class* to connote an object category (such as the face, human body, or hose categories). For the face class, we would like to learn a representation which contains not merely information about face identity, but also about facial attributes, such as head pose, or facial expression. A first step towards obtaining a useful class representation is to be able to establish correspondences between relevant locations in different face images.
Correspondences between images are essential for numerous problems in computer vision, such as, for instance, 3D reconstruction from multiple images. Classical methods for obtaining 3D reconstructions from multiple images, which use for instance bundle adjustment [Triggs *et al.*, 1999], rely on minimising a photometric reprojection error between observed and projected image points. This does not necessarily result in establishing semantically meaningful correspondences between different objects. In the context of object recognition, extracted image features can be compared to features

of a stored model or reference image. Felzenszwalb and Huttenlocher [2005] introduce *pictorial structure* models for part-based object recognition. Such a model is given by a collection of parts and their connections between one another, and it is matched to an image by matching parts to image data while respecting the deformable model's properties. One way to represent the face class is a 3D morphable model (3DMM) [Blanz and Vetter, 1999]. The 3DMM can be fitted to an image end-to-end using facial landmarks or a photometric error as supervision [Bas *et al.*, 2017; Tewari *et al.*, 2017]. Facial landmarks describe salient parts of a face, such as the eyes, nose, mouth, and the jawline. They allow the establishment of correspondences between images of different face identities and can therefore already be considered as face representations.

A line of work learns landmarks in an unsupervised fashion. Thewlis *et al.* [2017b] apply synthetic warps to images in order to learn a sparse set of unsupervised object landmarks. Their loss function enforces that a fixed number of detected points must be equivariant with respect to the applied warps. This approach is extended to learning dense, pixel-wise representations of the object structure in [Thewlis *et al.*, 2017a]. Each object point is mapped to a canonical location in the latent space that is homeomorphic to a sphere. Both methods use the known correspondences from the applied warps.

Zhang *et al.* [2018] and Jakab *et al.* [2018] also build on Thewlis *et al.* [2017b] and use the discovered landmarks to reconstruct the input image. The discovered landmarks are passed into the decoder along with appearance features to generate the original input. Jakab *et al.* [2018], similar to our *X2Face* and *FAb-Net* frameworks, use different images of the same video tracks to obtain image pairs and learn using a pixel-wise photometric loss. Unlike *X2Face*, which can be used to drive face generation by using audio, head pose or another face, their method can only be used to drive a face with another face or with target landmarks. Furthermore, our *FAb-Net* framework does not aim solely to obtain facial landmarks like [Jakab *et al.*, 2018] but also results in more generic representations that can be used for predicting facial expression and head pose.

2.2 Cross-modal audio-visual learning

Humans with no hearing or visual impairments make use of audio-visual associations in many everyday situations. One modality can provide cues about the other, as,

for instance, when humans perceive speech. This has been studied experimentally in depth and has influenced the design of automatic speech recognition systems.
In 1976, McGurk and MacDonald [1976] described the so-called McGurk effect which demonstrates the role of vision for speech perception. They performed a study which consisted of two parts: in the first part, the subjects had to listen to auditory information and then repeat what they heard; in the second part, they watched a dubbed video while listening to auditory information containing either matching or non-matching spoken syllables. They found that hearing and seeing were intertwined with one another and that perceived lip movements had a significant impact on the perception of speech for people with no hearing impairment. They observed a stronger influence of visual information on audio perception in adults than in younger children. However, Kuhl and Meltzoff [1982] demonstrated that 4-5 months old infants are already able to detect the correspondence between visually and auditorily perceived speech.
Several studies [Erber, 1969; Dodd, 1977; Summerfield, 1979] showed that this bimodal correspondence can be very useful when noise is obstructing the auditory perception of speech, since the visual information can then provide complementary signals. Lipreading (i.e. understanding speech from mouth movements) is used not only by humans with a hearing impairment, but also by humans with good hearing. However, most people are not trained to fully understand speech from the lip movements alone, but rather process correlated visual information along with the audio information. This is useful for instance at a noisy party (also known as the cocktail party problem [Cherry, 1953]) where lipreading can help to focus attention on the voice that matches the lip movements of one person while ignoring other voices and noise.
Human sensitivity to the synchronisation of auditory and visual signals is very strong. This also allows the localisation of sound sources which can result in the so-called ventriloquism effect. This occurs when the perceived sound source is known to be false, e.g. a television screen [Thomas, 1941; Bertelson and Radeau, 1981].
These observations inspire the use of co-occurring visual and audio information as a rich signal to train neural networks for obtaining useful visual representations. Video data containing audio provides audio-visual correspondence for free, without the need for manual annotations. However, curating videos which provide a strong enough training signal can be challenging, since visible sound sources are sometimes not audible as they can be obfuscated by other background noise from sources that are outside the field of view. There has been a series of works exploiting the correlation

of audio and visual information in video data. We will give an overview of some representative works in the following.

2.2.1 Joint training of audio and visual networks

de Sa [1994a,b] proposed learning jointly from matching audio and visual information where one modality provides the labelling signal for the other one. The introduced algorithm is based on the minimisation of the disagreement error between the correlated audio and visual representations.
The idea of using the knowledge of audio-visual synchronisation to jointly embed audio and visual information has been further explored for learning visual and audio embeddings. These have proven useful for sound source localisation and separation, cross-modal retrieval, sound classification, and action recognition.

Joint audio-visual embeddings

When given a sound, [Hershey and Movellan, 2000; Kidron *et al.*, 2005, 2007; Arandjelović and Zisserman, 2017, 2018; Zhao *et al.*, 2018, 2019; Owens and Efros, 2018] all *localise* the *sound* by highlighting the parts in an image that are associated with audio sources. Hershey and Movellan [2000] maximise the mutual information between audio and visual features to find the image regions that are correlated with audio, e.g. faces of people speaking in videos. [Kidron *et al.*, 2005, 2007]'s "Pixels that sound" considers a broader range of applications beyond talking faces, and imposes spatial sparsity on canonical correlation analysis for regularisation. Owens and Efros [2018] propose an early-fusion CNN model that, given audio and visual input, learns whether the inputs are synchronised or shifted. Their learnt representation can be used for sound source localisation; they show that it also serves as a good pre-training for action recognition tasks. Furthermore, their network can be adapted for *sound source separation*. Zhao *et al.* [2018, 2019] propose frameworks to separate sounds of musical instruments into a set of components that represent the sound from each pixel. During training, sound tracks are mixed together and the network is then tasked to separate them based on a visual input that corresponds to one of the original audio signals. While Zhao *et al.* [2018] seem mainly to rely on image semantics to separate sounds, Zhao *et al.* [2019] consider the motion information in the video explicitly. This results in their framework being able to separate sounds from two semantically similar sound sources. Rouditchenko *et al.* [2019] also build on [Zhao *et al.*, 2018] with the goal of using the image and the audio networks independently after training.

Unlike approaches that mix sound tracks together and learn to separate them to obtain the original sounds, Gao *et al.* [2018] use non-negative matrix factorisation to obtain object category bases, leveraging a ImageNet pretrained network to get weak labels for the objects present in a visual frame. The disentangled object bases can then be used for audio source separation. Gao and Grauman [2019] build on this and introduce a *co-separation* framework that is trained by learning to separate shared object sounds in two videos from different background sounds in the respective videos. Xu *et al.* [2019b] separate sounds recursively by removing sounds in descending energy order until a sound mixture is empty or consists of noise only. This allows for more flexibility as the number of sound sources does not have to be specified a priori. Other methods use more supervision to localise sound in images; Senocak *et al.* [2018] learn to predict an attention mask on the image for sounds from the Flickr-SoundNet dataset [Aytar *et al.*, 2016] which is refined by using manually annotated sound source bounding boxes.

Mapping audio and visual information to the same embedding space enables *cross-modal retrieval* [Arandjelović and Zisserman, 2018] and *sound classification* [Arandjelović and Zisserman, 2017; Korbar *et al.*, 2018]. Korbar *et al.* [2018] build on [Arandjelović and Zisserman, 2017, 2018] and improve the sound classification significantly. Their curriculum learning scheme includes hard negative samples which are chosen from the same video, encouraging the network to not only focus on image semantics, but on the precise synchronisation of visual and audio events. In addition to this, they show that their training serves as a useful pre-training for subsequent *action recognition*.

Other works have used one modality to learn a better representation for the other modality. For instance, Owens *et al.* [2016b] use ambient sound as supervision to learn better visual representations. Soundnet [Aytar *et al.*, 2016], on the other hand, learn a deep representation of natural sounds without using any ground truth sound labels by distilling knowledge from a visual recognition network into a network that takes sound as input. The visual network serves as a teacher to the sound network.

Audio-visual learning for speech

The availability of audio-visual data of talking heads, along with advances in computer vision for the face domain (e.g. face detection), have resulted in audio-visual speech lending itself as a natural first step towards more general audio-visual learning. Yuhas *et al.* [1989] presented a neural network framework that generates the corresponding acoustic spectrum from images of a person forming vowels alone. They demonstrated

that using visual information in addition to noisy audio improves vowel recognition performance. This is done by adding a vowel recogniser after combining information from the different modalities.
Audio-visual alignment for speech is also explored as a task in itself. Synchronisation frameworks can be used to align speech to the corresponding video [Chung and Zisserman, 2016a; Chung *et al.*, 2019]. SyncNet [Chung and Zisserman, 2016a] is set up to learn a joint embedding from a sequence of face crops and corresponding audio signal. It is trained using a contrastive loss to minimise the distance between matching audio and visual features and to maximise the same for non-matching features. The learnt embedding can be used for active speaker detection and lip reading. Lipreading is further explored in depth in [Chung and Zisserman, 2016b]. Chung *et al.* [2019] build on [Chung and Zisserman, 2016a] by changing their training objective to a multi-way matching loss. Jha *et al.* [2018] address a related task, learning to spot words in silent videos instead of learning to recognise words. Several works have used visual information to address the cocktail party problem where the aim is to isolate the relevant speech and to suppress other noise sounds [Ephrat *et al.*, 2018; Afouras *et al.*, 2018; Owens and Efros, 2018].

Sound from vision beyond lipreading

Davis *et al.* [2014] use high framerate silent video recordings of object vibrations to recover the sound that caused the vibration of the object (e.g. the sound of hitting the object). Their method turns everyday objects, such as a glass of water, into *visual microphones*. In contrast, Owens *et al.* [2016a] record a video dataset of people hitting, scratching, and prodding different objects with a drumstick. Their proposed framework uses this dataset to learn to generate plausible sounds that correspond to a visible action. Other works have considered the problem of generating in-the-wild sounds (e.g. ambient sounds but also sounds from people) that match visual frames [Zhou *et al.*, 2018].

2.2.2 Audio-visual learning for music

Even though music is often seen as an aural art form, there are many factors that make music an audio-visual art. In conventional music performances, sound is the result of the movement of a human performer. In addition to motion causing the produced sound (for example, when a pianist hits keys of the keyboard), the perceiver's experience of a music performance can also be influenced by auxiliary visual information and movement (for example, if the pianist is smiling or frowning, this might

change how the music is perceived). Davidson [1993] conducted studies which demonstrated that vision is in some cases more informative than sound for the perception of the expressive intentions of a musician.

In order to better understand the interplay of audio and visual information, Li *et al.* [2019a] recorded the URMP video dataset which consists of video recordings of western classical chamber music with a variety of string and wind instruments. The videos of individual musicians were recorded separately and then added together to give full performances. Li *et al.* [2017a] train and test on the URMP dataset to perform audio source association with the musicians that are visible. They exploit the bowing motion of string instrument players using optical flow and match bowing onsets with note onsets in the aligned music score. Li *et al.* [2017b], on the other hand, analyse vibrato-patterns in the audio and video frames for audio-visual association.

Chen *et al.* [2017] go beyond this and address the problem of cross-modal audio-visual generation for the URMP dataset. They train conditional Generative Adversarial Networks (GANs) to generate a spectrogram when given an input image and to generate an image when given an input sound. Hao *et al.* [2018] consider the same problem and present an end-to-end trainable framework for the generation in both directions which is based on CycleGAN [Zhu *et al.*, 2017]. However, both frameworks leave room for improvement in terms of the image quality of the generated frames. A single video frame is taken as input and generation target, which results in a generated output that does not contain temporal information. The results seem to only reflect the semantic classes. Instead of generating images corresponding to music, several methods have been proposed to predict human body keypoints. Yamamoto *et al.* [2010] predict finger motion for piano players given the music score, and Shlizerman *et al.* [2018] learn to predict arm and hand keypoints for violin and piano players from audio input. Li *et al.* [2018] focus on the pianist's expressive body motion rather than the precise hand movements using MIDI data as input.

A slightly easier problem than generating spectrograms from visual information alone is the task of visually informed audio inpainting which consists of synthesising missing audio that corresponds to given video frames. Zhou *et al.* [2019b] introduce a method that inpaints 0.8 second audio segments using video frames and optical flow as input. Whilst their audio inpainting results sound convincing, they only show results for cases where the inpainting task consists of elongating notes that start or end outside the missing segment. In particular, no notes are generated that would be fully contained (i.e. start and end) within the missing 0.8 second segment. In order to generate new notes that are fully contained in the missing segment, the network

would have to use very fine-grained information (e.g. from the fingers of the musician, or the vibration of the strings). This would result in a problem that is more similar to visual music transcription.

Visual music transcription

Visual music transcription is the task of obtaining a symbolic music representation, such as MIDI or music notation, from visual information alone. Methods for a variety of musical instruments have been proposed, but they only work in extremely constrained settings, e.g. requiring recordings from high framerate cameras. Gómez Gutiérrez *et al.* [2017] and Bazzica *et al.* [2017] propose to transcribe music from clarinet playing videos using the hand movements. Zhang *et al.* [2007] detect the strings of a violin and recognise finger events (e.g. their position and string pressing events) in order to transcribe violin music. Their method requires tracking the fingers and the strings, and makes assumptions about the length of the fingerboard. This requires perfect visual alignment of the data. We propose a method for visual pitch estimation for violin playing in chapter 5 that does not make such assumptions and can be applied to in-the-wild data downloaded from YouTube.
Goldstein and Moses [2018] present a framework that extracts the vibration of guitar strings from high framerate camera recordings of a camera that is mounted on the guitar to transcribe guitar music from silent video. This has similarities to the earlier mentioned work by Davis *et al.* [2014] that recovers sound from visible object vibrations. However, these methods require high framerate video recordings.
Several methods have addressed the problem of visual piano transcription from top-down views onto a keyboard. Suteparuk [2014] and Akbari and Cheng [2015] use RGB images and require difference images between the background and current video frame to detect hands and piano keys. However, this is difficult to obtain when the illumination changes across the video or when shadows appear. Akbari and Cheng [2015] add an illumination correction step in their pipeline, but the authors also report limitations for drastic light changes or vibrations of the camera or piano. Deb and Rajwade [2016] predict per-frame key presses even under illumination changes; however, their set-up is quite constrained and can only predict a single key press per frame. Nisbet and Green [2017] additionally use depth information, which enables finger and key velocity prediction. Possible use cases for visual piano transcription include piano tutoring systems. Rho *et al.* [2014] present such a system which uses depth cameras to identify key presses. Gorodnichy and Yogeswaran [2006] consider the hands and in particular the fingers to detect and teach piano fingering. Oka and

Hashimoto [2013] also estimate the fingering, but use a dictionary dataset to find nearest neighbours at test time. It is not clear whether this method would generalise were it applied to an unseen pianist.

Akbari *et al.* [2018] present a multi-step pipeline that requires significant preprocessing. Given a processed crop of a single key, their CNNs predict whether the key has been pressed. Akbari *et al.* [2018] relies on key presses that are clearly visible from video frame differences. Again, this may not be the case when there is video jitter, instrument vibrations, low-resolution video data, or video recorded from directly above the keyboard.

Lee *et al.* [2019] present a deep learning approach that uses both audio and visual information, to detect key presses. They only demonstrate their method on high-quality videos (recorded at 60 fps) and simple pieces (e.g. piano exercises that have at most one note per hand at the same time).

Our method for visual music transcription for piano playing (see chapter 6) does not require high framerate videos and works in fairly unconstrained recording settings with different musicians, pianos, and lighting variation.

Chapter 3

X2Face: A network for controlling face generation by using images, audio, and pose codes

This work was presented as a *poster* at the European Conference on Computer Vision (ECCV), 2018.

We introduce a self-supervised framework that allows to drive face generation with another face image. The same framework can be used to control face generation with another modality, such as head pose or audio. Furthermore, this setup can be used for lightweight video editing.

X2Face: A network for controlling face generation using images, audio, and pose codes

Olivia Wiles*, A. Sophia Koepke*, Andrew Zisserman

Visual Geometry Group,
University of Oxford
{ow,koepke,az}@robots.ox.ac.uk

Abstract. The objective of this paper is a neural network model that controls the pose and expression of a given face, using another face or modality (e.g. audio). This model can then be used for lightweight, sophisticated video and image editing.
We make the following three contributions. First, we introduce a network, X2Face, that can control a *source* face (specified by one or more frames) using another face in a *driving* frame to produce a *generated* frame with the identity of the *source* frame but the pose and expression of the face in the *driving* frame. Second, we propose a method for training the network fully self-supervised using a large collection of video data. Third, we show that the generation process can be driven by other modalities, such as audio or pose codes, without any further training of the network.
The generation results for driving a face with another face are compared to state-of-the-art self-supervised/supervised methods. We show that our approach is more robust than other methods, as it makes fewer assumptions about the input data. We also show examples of using our framework for video face editing.

1 Introduction

Being able to animate a still image of a face in a controllable, lightweight manner has many applications in image editing/enhancement and interactive systems (e.g. animating an on-screen agent with natural human poses/expressions). This is a challenging task, as it requires representing the face (e.g. modelling in 3D) in order to control it and a method of mapping the desired form of control (e.g. expression or pose) back onto the face representation. In this paper we investigate whether it is possible to forgo an explicit face representation and instead implicitly learn this in a self-supervised manner from a large collection of video data. Further, we investigate whether this implicit representation can then be used directly to control a face with another modality, such as audio or pose information.
To this end, we introduce X2Face, a novel self-supervised network architecture that can be used for face puppeteering of a *source* face given a *driving vector*.

* Denotes equal contribution.

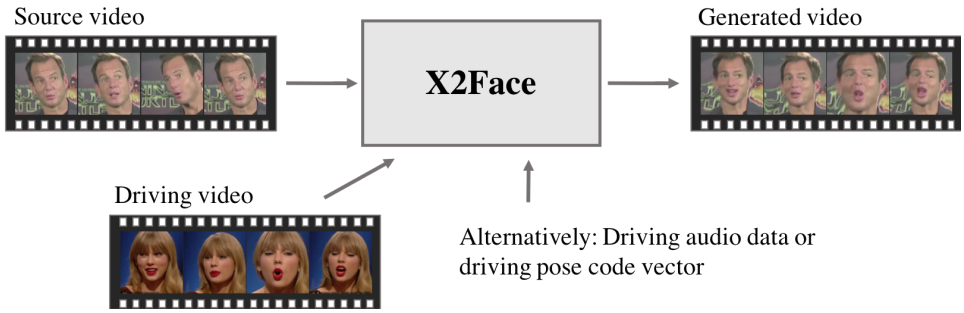


Fig. 1: Overview of X2Face: a model for controlling a *source* face using a *driving* frame, audio data, or specifying a pose vector. X2Face is trained without expression or pose labels.

The *source* face is instantiated from a single or multiple *source* frames, which are extracted from the same face track. The *driving vector* may come from multiple modalities: a *driving* frame from the same or another video face track, pose information, or audio information; this is illustrated in Fig. 1. The *generated* frame resulting from X2Face has the identity, hairstyle, etc. of the *source* face but the properties of the *driving vector* (e.g. the given pose, if pose information is given; or the *driving* frame's expression/pose, if a *driving* frame is given).
The network is trained in a self-supervised manner using pairs of *source* and *driving* frames. These frames are input to two subnetworks: the *embedding network* and the *driving network* (see Fig. 2). By controlling the information flow in the network architecture, the model learns to factorise the problem. The *embedding network* learns an *embedded* face representation for the *source* face – effectively face frontalisation; the *driving network* learns how to map from this *embedded* face representation to the *generated* frame via an embedding, named the *driving vector*.
The X2Face network architecture is described in Section 3.1, and the self-supervised training framework in Section 3.2. In addition we make two further contributions. First, we propose a method for linearly regressing from a set of labels (e.g. for head pose) or features (e.g. from audio) to the *driving vector*; this is described in Section 4. The performance is evaluated in Section 5, where we show (i) the robustness of the generated results compared to state-of-the-art self-supervised [45] and supervised [1] methods; and (ii) the controllability of the network using other modalities, such as audio or pose. The second contribution, described in Section 6, shows how the *embedded* face representation can be used for video face editing, e.g. adding facial decorations in the manner of [31] using multiple or just a single *source* frame.

2 Related work

Explicit modelling of faces for image generation. Traditionally facial animation (or puppeteering) given one image was performed by fitting a 3DMM and then modifying the estimated parameters [3]. Later work has built on the

fitting of 3DMMs by including high level details [34,41], taking into account additional images [33] or 3D scans [4], or by learning 3DMM parameters directly from RGB data without ground truth labels [39,2]. Please refer to Zollhöfer et. al. [46] for a survey.

Given a driving and source video sequence, a 3DMM or 3D mesh can be obtained and used to model both the driving and source face [43,40,10]. The estimated 3D is used to transform the expression of the source face to match that of the driving face. However, this requires additional steps to transfer the hidden regions (e.g. the teeth). As a result, a neural network conditioned on a single driving image can be used to predict higher level details to fill in these hidden regions [25].

Motivated by the fact that a 3DMM approach is limited by the components of the corresponding morphable model, which may not model the full range of required expressions/deformations and the higher level details, [1] propose a 2D warping method. Given only one source image, [1] use facial landmarks in order to warp the expression of one face onto another. They additionally allow for fine scale details to be transferred by monitoring changes in the driving video.

An interesting related set of works consider how to frontalise a face in a still image using a generic reference face [14], transferring expressions of an actor to an avatar [35] and swapping one face with another [20,24].

Learning based approaches for image generation. There is a wealth of literature on supervised/self-supervised approaches; here we review only the most relevant work. Supervised approaches for controlling a given face learn to model factors of variation (e.g. lighting, pose, etc.) by conditioning the generated image on known ground truth information which may be head pose, expression, or landmarks [44,21,42,5,12,30]. This requires a training dataset with known pose or expression information which may be expensive to obtain or require subjective judgement (e.g. in determining the expression). Consequently, self-supervised and unsupervised approaches attempt to automatically learn the required factors of variation (e.g. optical flow or pose) without labelling. This can be done by maximising mutual information [7] or by training the network to synthesise future video frames [29,11].

Another relevant self-supervised method is CycleGAN [45] which learns to transform images of one domain into those of another. While not explicitly devised for this task, as CycleGAN learns to be cycle-consistent, the transformed images often bear semantic similarities to the original images. For example, a CycleGAN model trained to transform images of one person's face (domain A) into those of another (domain B), will often learn to map the pose/position/expression of the face in domain A onto the generated face from domain B.

Using multi-modal setups to control image generation. Other modalities, such as audio, can control image generation by using a neural network that learns the relationship between audio and correlated parts in corresponding images. Examples are controlling the mouth with speech [8,38], controlling a head with audio and a known emotional state [16], and controlling body movement with music [36].

Our method has the benefits of being self-supervised and the ability to control the generation process from other modalities without requiring explicit modelling of the face. Thus it is applicable to other domains.

3 Method

This section introduces the network architecture in Section 3.1, followed by the curriculum strategy used to train the network in Section 3.2.

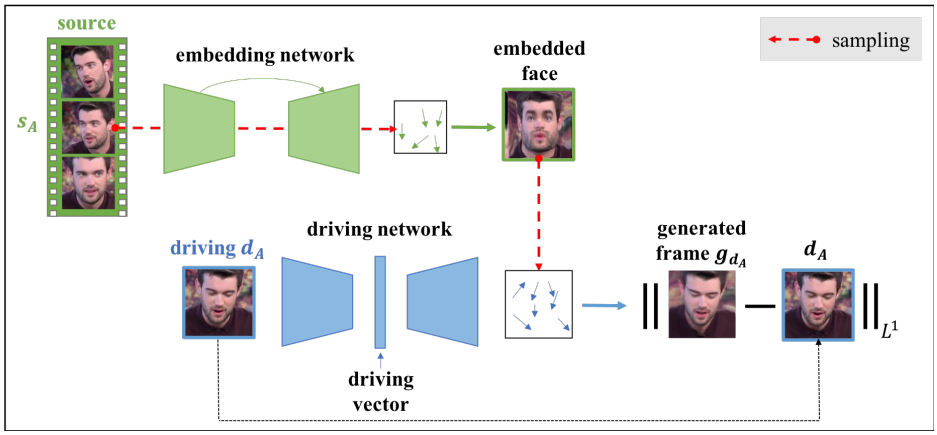


Fig. 2: An overview of X2Face during the initial training stage. Given multiple frames of a video (here 4 frames), one frame is designated the *source* frame and another the *driving* frame. The *source* frame is input to the *embedding network*, which learns a sampler to map pixels from the *source* frame to the *embedded* face. The *embedded* face effectively results in a rolled out frontalised face image (but we do not enforce that this should be a frontalised or neutral face). The *driving* frame is input to the *driving network*, which learns to map pixels from the *embedded* face to the *generated* frame. The *generated* frame should have the identity of the *source* frame and the pose/expression of the *driving* frame. In this training stage, as the frames are from the same video, the *generated* and *driving* frames should match. However, at test time the identities of the *source* and *driving* face can differ.

3.1 Architecture

The network takes two inputs: a *driving* and a *source* frame. The *source* frame is input to the *embedding network* and the *driving* frame to the *driving network*. This is illustrated in Fig. 2. Precise architectural details are given in the supplementary material in Section A.1 and Section A.2.

Embedding network. The *embedding network* learns a bilinear sampler to determine how to map from the *source* frame to a face representation, the *embedded* face. The architecture is based on U-Net [32] and pix2pix [15]; the output is a

2-channel image (of the same dimensions as the *source* frame) that encodes the flow $\delta x, \delta y$ for each pixel.
While the *embedding network* is not explicitly forced to frontalise the *source* frame, we observe that it learns to do so for the following reason. Because the *driving network* samples from the *embedded* face to produce the *generated* frame without knowing the pose/expression of the *source* frame, it needs the *embedded* face to have a common representation (e.g. be frontalised) across *source* frames with differing poses and expressions.
Driving network. The *driving network* takes a *driving* frame as input and learns a bilinear sampler to transform pixels from the *embedded* face to produce the *generated* frame. It has an encoder-decoder architecture. In order to sample correctly from the *embedded* face and produce the *generated* frame, the latent embedding (the *driving vector*) must encode pose/expression/zoom/other factors of variation.

3.2 Training the network

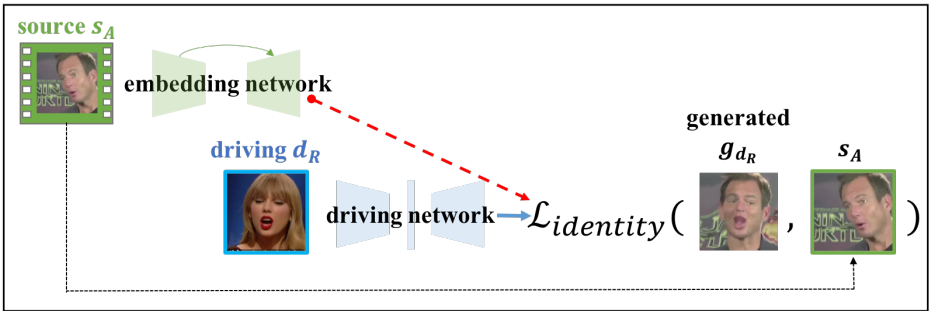


Fig. 3: The identity loss function when the *source* and *driving* frames are of different identities. This loss enforces that the *generated* frame has the same identity as the *source* frame.

The network is trained with a curriculum strategy using two stages. The first training stage (**I**) is fully self-supervised. In the second training stage (**II**), we make use of a CNN pre-trained for face identification to add additional constraints based on the identity of the faces in the *source* and *driving* frames to finetune the model following training stage (**I**).
I. The first stage (illustrated in Fig. 2) uses only a pixelwise $L1$ loss between the *generated* and the *driving* frames. Whilst this is sufficient to train the network such that the *driving* frame encodes expression and pose, we observe that some face shape information is leaked through the *driving vector* (e.g. the *generated* face becomes fatter/longer depending on the face in the *driving* frame). Consequently, we introduce additional loss functions – called identity loss functions – in the second stage.
II. In the second stage, the identity loss functions are applied to enforce that the identity is the same between the *generated* and the *source* frames irrespective

of the identity of the *driving* frame. This loss should mitigate against the face shape leakage discussed in stage **I**. In practice, one *source* frame s_A of identity A, and two *driving* frames d_A,d_R are used as training inputs; d_A is of identity A and d_R a random identity. This gives two generated frames g_{d_A}, g_{d_R} respectively, which should both be of identity A. Two identity loss functions are then imposed: $\mathcal{L}_{\text{identity}}(d_A, g_{d_A})$ and $\mathcal{L}_{\text{identity}}(s_A, g_{d_R})$. $\mathcal{L}_{\text{identity}}$ is implemented using a network pre-trained for identity to measure the similarity of the images in feature space by comparing appropriate layers of the network (i.e. a content loss as in [13,6]). The precise layers are chosen based on whether we are considering g_{d_A} or g_{d_R}:

1. $\mathcal{L}_{identity}(d_A, g_{d_A})$. g_{d_A} should have the same identity, pose and expression as d_A so we use the photometric $L1$ loss and a $L1$ content loss on the Conv2-5 and Conv7 layers (i.e. layers that encode both lower/higher level information such as pose/identity) between g_{d_A} and d_A.
2. $\mathcal{L}_{identity}(s_A, g_{d_R})$ *(Fig. 3)*. g_{d_R} should have the identity of s_A but the pose and expression of d_R. Consequently, we cannot use the photometric loss but only a content loss. We minimise a $L1$ content loss on the Conv6-7 layers (i.e. layers encoding higher level identity information) between g_{d_A} and s_A.

The pre-trained network used for these losses is the 11-layer VGG network (configuration A) [37] trained on the VGG-Face Dataset [26].

4 Controlling the image generation with other modalities

Given a trained X2Face network, the *driving vector* can be used to control the *source* face with other modalities such as audio or pose.

4.1 Pose

Instead of controlling the generation with a *driving* frame, we can control the head pose of the *source* face using a pose code such that when varying the code's pitch/yaw/roll angles, the *generated* frame varies accordingly. This is done by learning a forward mapping $f_{p \to v}$ from head pose p to the *driving vector* v such that $f_{p \to v}(p)$ can serve as a modified input to the *driving network*'s decoder. However, this is an ill-posed problem; directly using this mapping loses information, as the *driving vector* encodes more than just pose.
As a result, we use vector arithmetic. Effectively we drive a *source* frame with itself but modify the corresponding *driving vector* v_{emb}^{source} to remove the pose of the *source* frame p_{source} and incorporate the new driving pose $p_{driving}$. This gives:

$$v_{emb}^{driving} = v_{emb}^{source} + v_{emb}^{\Delta pose} = v_{emb}^{source} + f_{p \to v}(p_{driving} - p_{source}). \quad (1)$$

However, the VoxCeleb dataset [23], which is used for training the framework, does not contain ground truth head pose, so an additional mapping $f_{v \to p}$ is needed to determine $p_{source} = f_{v \to p}(v_{emb}^{source})$.

$f_{v\to p}$. $f_{v\to p}$ is trained to regress p from v. It is implemented using a fully connected layer with bias and trained using an L1 loss. Training pairs (v, p) are obtained using an annotated dataset with image to pose labels p; v is obtained by passing the image through the encoder of the *driving network*.

$f_{p\to v}$. $f_{p\to v}$ is trained to regress v from p. It is implemented using a fully-connected linear layer with bias followed by batch-norm. When $f_{v\to p}$ is known, this function can be learnt directly on VoxCeleb by passing an image through X2Face to get the *driving vector* v and $f_{v\to p}(v)$ gives the pose p.

4.2 Audio

Audio data from the videos in the VoxCeleb dataset can be used to drive a *source* face in a manner similar to that of pose by driving the *source* frame with itself but modifying the *driving vector* using the audio from another frame. The forward mapping $f_{a\to v}$ from audio features a to the corresponding *driving vector* v is trained using pairs of audio features a and driving vectors v. These can be directly extracted from VoxCeleb (so no backward mapping $f_{v\to a}$ is required). a is obtained by extracting the 256D audio features from the neural network in [9] and the 128D v by passing the corresponding frame through the *driving network*'s encoder. Ordinary least squares linear regression is then used to learn $f_{a\to v}$ after first normalising the audio features to $\sim N(0, 1)$. No normalisation is used when employing the mapping to drive the frame generation; this amplifies the signal, visually improving the generated results.
As learning the function $f_{a\to v} : \mathbb{R}^{1\times 256} \to \mathbb{R}^{1\times 128}$ is under-constrained, the embedding learns to encode some pose information. Therefore, we additionally use the mappings $f_{p\to v}$ and $f_{v\to p}$ described in Section 4.1 to remove this information. Given driving audio features $a_{driving}$ and the corresponding, non-modified *driving vector* v_{emb}^{source}, the new *driving vector* $v_{emb}^{driving}$ is then

$$v_{emb}^{driving} = v_{emb}^{source} + f_{a\to v}(a_{driving}) - f_{a\to v}(a_{source}) - f_{p\to v}(p_{audio} - p_{source}),$$

where $p_{source} = f_{v\to p}(v_{emb}^{source})$ is the head pose of the frame input to the *driving network* (i.e. the *source* frame), $p_{audio} = f_{v\to p}(f_{a\to v}(a_{driving}))$ is the pose information contained in $f_{a\to v}(a_{driving})$, and a_{source} is the audio feature vector corresponding to the *source* frame.

5 Experiments

This section evaluates X2Face by first performing an ablation study in Section 5.1 on the architecture and losses used for training, followed by results for controlling a face with a *driving* frame in Section 5.2, pose information in Section 5.3, and audio information in Section 5.4.
Training. X2Face is trained on the VoxCeleb video dataset [23] using dlib [18] to crop the faces to 256×256. The identities are randomly split into train/val/test

identities (with a split of 75/15/10) and frames extracted at one fps to give 900,764 frames for training and 125,131 frames for testing.
The model is trained in PyTorch [27] using SGD with momentum 0.9 and batch-size of 16. First, it is trained just with $L1$ loss, and a learning rate of 0.001. The learning rate is decreased by a factor of 10 when the loss plateaus. Once the loss converges, the identity losses are incorporated and are weighted as follows: (i) for same identities to be as strong as the photometric $L1$ loss at each layer; (ii) for different identities to be 1/10 the size of the photometric loss at each layer. This training phase is started with a learning rate of 0.0001.
Testing. The model can be tested using either a single or multiple *source* frames. The reasoning for this is that if the *embedded* face is stable (e.g. different facial regions always map to the same place on the *embedded* face), we expect to be able to combine multiple *source* frames by averaging over the *embedded* faces.

5.1 Architecture studies

To quantify the utility of using additional views at *test* time and the benefit of the curriculum strategy for training the network (i.e. using the identity losses explained in Section 3.2), we evaluate the results for these different settings on a left-out test set of VoxCeleb. We consider 120K *source* and *driving* pairs where the *driving* frame is from the same video as the *source* frames; thus, the *generated* frame should be the same as the *driving* frame. The results are given in Table 1.

Table 1: $L1$ reconstruction error on the test set, comparing the *generated* frame to the ground truth frame (in this case the *driving* frame) for different training/testing setups. Lower is better for $L1$ error. Additionally, we give the percentage improvement over the $L1$ error for the model trained with only training stage **I** and tested with a single *source* frame. In this case, higher is better

Training strategy	**# of *source* frames at test time**	$L1$ **error**	**% Improvement**
Training stage **I**	1	0.0632	0%
Training stage **II**	1	0.0630	0.32%
Training stage **I**	3	0.0524	17.14%
Training stage **II**	3	0.0521	17.62%

The results in Table 1 confirm that both training with the curriculum strategy and using additional views at *test* time improve the reconstructed image. Section A.3 in the supplementary material includes qualitative results and shows that using additional *source* frames when testing is especially useful if a face is seen at an extreme pose in the initial *source* frame.

5.2 Controlling image generation with a *driving* frame

The motivation of our architecture is to be able to map the expression and pose of a *driving* frame onto a *source* frame *without* any annotations on expression or

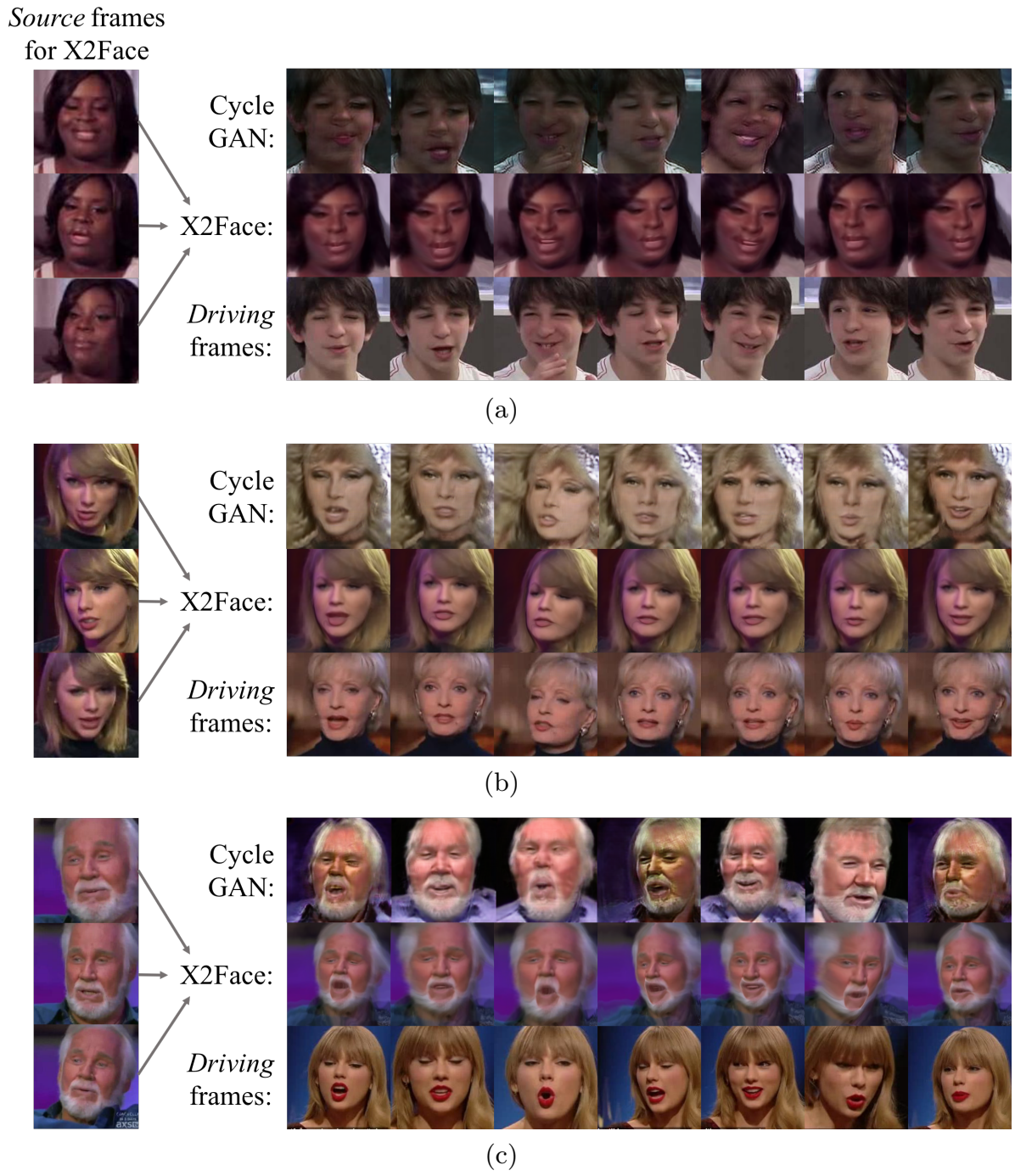


Fig. 4: Comparison of X2Face's *generated* frames to those of CycleGAN given a driving video sequence. Each example shows from bottom to top: the *driving* frame, our *generated* result and CycleGAN's generated result. To the left, *source* frames for X2Face are shown (at test time CycleGAN does not require *source* frames, as it is has been trained to map between the given *source* and *driving* identities). These examples demonstrate multiple benefits of our method. *First*, X2Face is capable of preserving the face shape of the source identity (top row) whilst driving the pose and expression according to the *driving* frame (bottom row); CycleGAN correctly keeps pose and expression but loses information about face shape and geometry when given too few training images as in example (a) (whereas X2Face requires no training samples for new identities). *Second*, X2Face has temporal consistency. CycleGAN samples from the latent space, so it sometimes samples from different videos resulting in jarring changes between frames (e.g. in example (c)).

pose. This section demonstrates that X2Face does indeed achieve this, as a set of *source* frames can be controlled with a driving video and generate realistic results. We compare to two methods: CycleGAN [45] which uses *no* labels and [1] which is designed top down and demonstrates impressive results. Additional qualitative results are given in Fig. 12 in the supplementary material and in the accompanying video[1].

Comparison to CycleGAN [45]. CycleGAN learns a mapping from a given domain (in this case a given identity A) to another domain (in this case another identity B). To compare to their method for a given pair of identities, we take all images of the given identities (so images may come from different video tracks) to form two sets of images: one set corresponding to identity A and the other to B. We then train their model using these sets. To compare, for a given *driving* frame of identity A, we visualise their *generated* frame from identity B which is compared to that of X2Face.

The results in Fig. 4 illustrate multiple benefits. First, X2Face generalises to unseen pairs of identities at test time given only a *source* and *driving* frame. CycleGAN is trained on pairs of identities, so if there are too few example images, it fails to correctly model the shape and geometry of the *source* face, producing unrealistic results. Additionally, our results have better temporal coherence (i.e. consistent background/hair style/etc. across *generated* frames), as X2Face transforms a given frame whereas CycleGAN samples from a latent space.

Comparison to Averbuch-Elor et. al. [1]. We compare to [1] in Fig. 5. There are two significant advantages of our formulation over theirs: first, we can handle more significant pose changes in the driving video and *source* frame (Fig. 5b-c). Second, ours has fewer assumptions: (1)[1] assumes that the first frame of the driving video is in a frontal pose with a neutral expression and that the *source* frame also has a neutral expression (Fig. 5d). (2) X2Face can be used when given a single *driving* frame whereas their method requires a video so that the face can be tracked and the tracking used to expand the number of correspondences and to obtain high level details.

While this is not the focus of this paper, our method can be augmented with the ideas from these methods. For example, as inspired by [1], we can perform simple post-processing to add higher level details (Fig. 5a, X2Face+p.p.) by transferring hidden regions using Poisson editing [28].

We note that our method results in artefacts in the generated background. A dynamic background in the driving video does not animate the background in the generated frames. Lighting changes across a driving video do not result in colour changes in the generated frames, since we are sampling from the same embedded face representation of the source identity for each generated frame.

[1] http://www.robots.ox.ac.uk/~vgg/research/unsup_learn_watch_faces/x2face.html

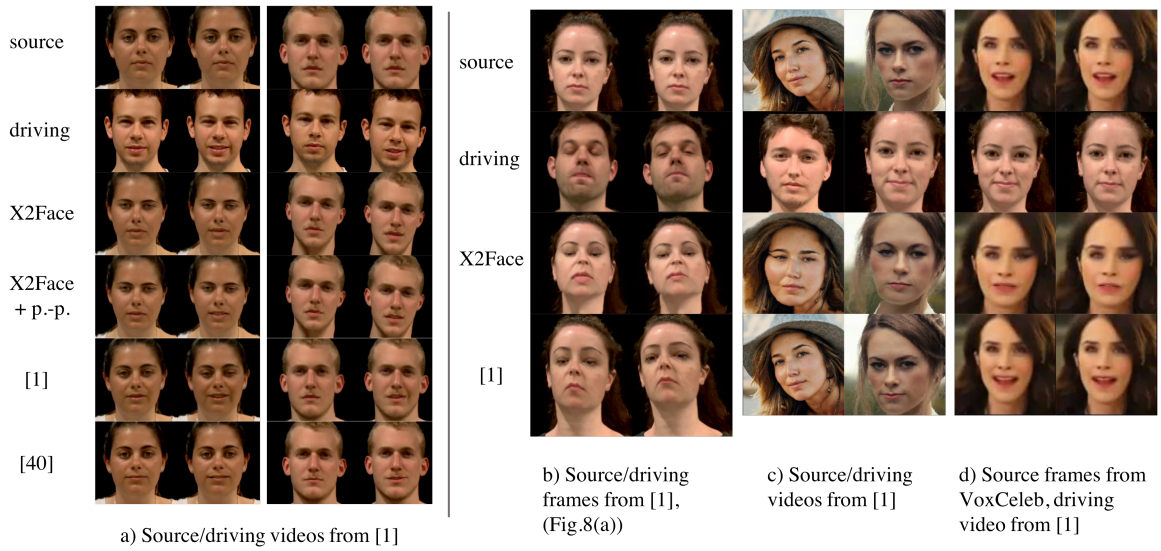


Fig. 5: Comparison of X2Face to supervised methods. In comparison to [1]: X2Face matches (b) pitch, and (c) roll and yaw; and X2Face can handle non-neutral expressions in the *source* frame (d). As with other methods, post-processing (X2Face + p.-p.) can be applied to add higher level details (a).

5.3 Controlling the image generation with pose

Before reporting results on controlling the *driving vector* using pose, we validate our claim that the *driving vector* does indeed learn about pose. To do this, we evaluate how accurately we can predict the three head pose angles – yaw, pitch and roll – given the 128D *driving vector*.

Pose predictor. To train the pose predictor which also serves as $f_{v\to p}$ (Section 4.1), the 25,993 images in the AFLW dataset [19] are split into train/val set, leaving out the 1,000 test images from [22] as test set. The results on the test set are reported in Table 2 confirming that the *driving vector* learns about head pose without having been trained on pose labels, as the results are comparable to those of a network directly trained for this task.
We then use $f_{v\to p}$ to train $f_{p\to v}$ (Section 4.1) and present generated frames for different, unseen test identities using the learnt mappings in Fig. 6. The *source* frame corresponds to p_{source} in Section 4.1 while $p_{driving}$ is used to vary one head pose angle while keeping the others fixed.

5.4 Controlling the image generation with audio input

This section presents qualitative results for using audio data from videos in the VoxCeleb dataset to drive the *source* frames. The VoxCeleb dataset consists of videos of interviews, suggesting that the audio should be especially correlated with the movements of the mouth. [9]'s model, trained on the BBC-Oxford 'Lip

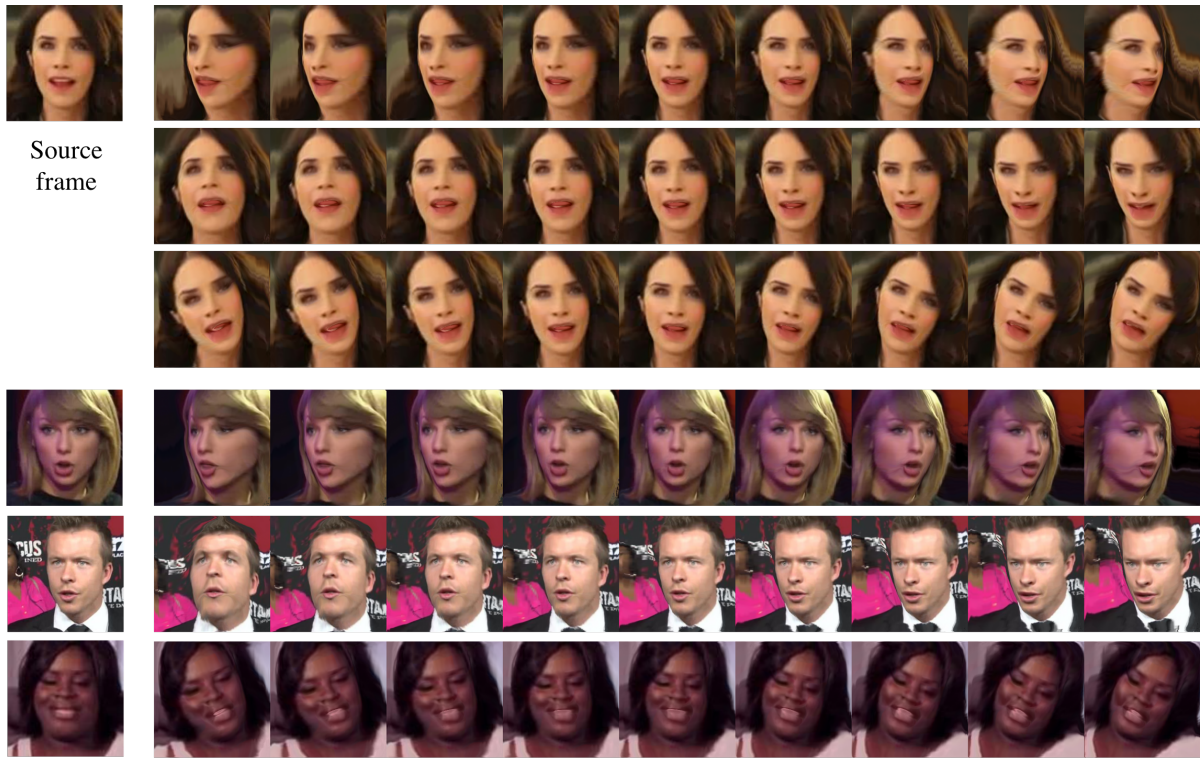


Fig. 6: Controlling image generation with pose code vectors. Results are shown for a single *source* frame which is controlled using each of the three head pose angles for the same identity (top three rows) and for different identities (bottom three rows). For further results and a video animation, we refer to Fig. 15 in the supplementary material and to the accompanying video. Whilst some artefacts are visible, the method allows the head pose angles to be controlled separately.

Table 2: MAE in degrees using the *driving vector* for head pose regression (lower is better). Note that the linear pose predictor from the *driving vector* performs only slightly worse than a supervised method [22], which has been trained for this task. Surprisingly, the yaw angle error reported in [22] is lower than their MAE.

Method	Roll	Pitch	Yaw	MAE
X2Face	5.85	7.59	14.62	9.36
KEPLER [22] (supervised)	8.75	5.85	6.45	7.02

Reading in the Wild' dataset (LRW), is used to extract audio features. We use the 256D vector activations of the last fully connected layer of the audio stream (FC7) for a 0.2s audio signal centred on the *driving* frame (the frame occurs half way through the 0.2s audio signal).

A potential source of error is the domain gap between the LRW dataset and VoxCeleb, as [9]'s model is not fine-tuned on the VoxCeleb dataset which contains much more background noise than the LRW dataset. Thus, their model has not

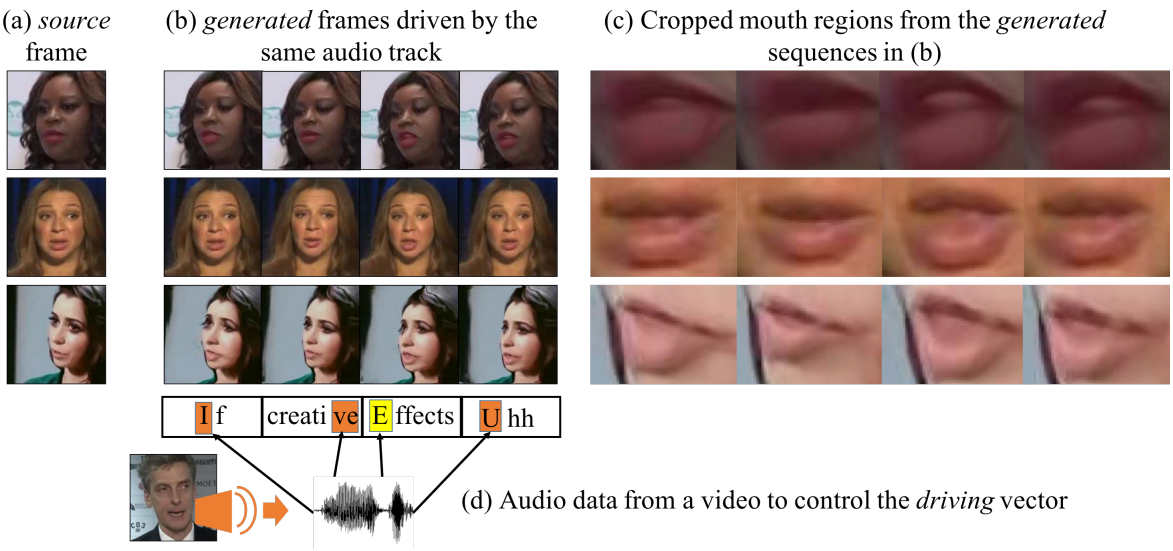


Fig. 7: Controlling image generation with audio information. We show how the same sounds affect various *source* frames; if our model is working well then the *generated* mouths should behave similarly. (a) shows the *source* frames. (b) shows the *generated* frames for a given audio sound which is visualised in (d) by the coloured portion of the word being spoken. As most of the change is expected to be in the mouth region, the cropped mouth regions are additionally visualised in (c). The audio comes from a native British speaker. As can be seen, in all generated frames, the mouths are more closed at the "ve" and "I" and more open at the "E" and "U". Another interesting point is that for the "Effects" frame, the audio is actually coming from an interviewer, so while the frame corresponding to the audio has a closed mouth, the *generated* results still open the mouth.

necessarily learnt to become indifferent to this noise. However, our model is relatively robust to this problem; we observe that the mouth movements in the *generated* frames are reasonably close to what we would expect from the sounds of the corresponding audio, as demonstrated in Fig. 7. This is true even if the person in the video is not speaking and instead the audio is coming from an interviewer. However, there is some jitter in the generation.

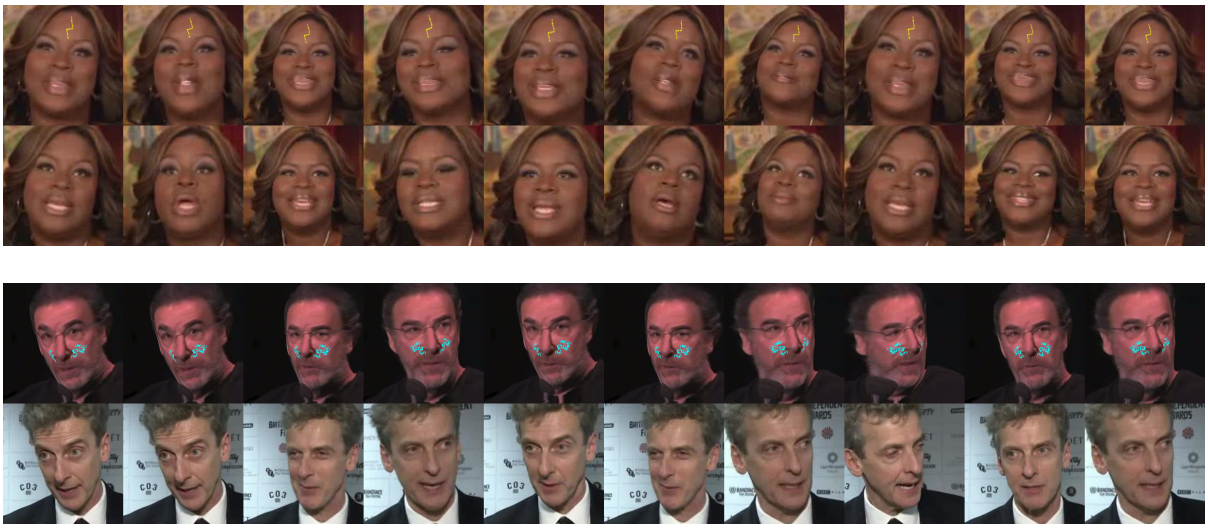
6 Using the embedded face for video editing

We consider how the *embedded* face can be used for video editing. This idea is inspired by the concept of an unwrapped mosaic [31]. We expect the *embedded* face to be pose and expression invariant, as can be seen qualitatively across the example *embedded* faces shown in the paper. Therefore, the *embedded* face can be considered as a UV texture map of the face and drawn on directly.

This task is executed as follows. A *source* frame (or set of *source* frames) is extracted and input to the *embedding network* to obtain the *embedded* face. The *embedded* face can then be drawn on using an image or other interactive tool. A video is reconstructed using the modified *embedded* face which is driven by a set

of *driving* frames. Because the *embedded* face is stable across different identities, a given edit can be applied across different identities. Example edits are shown in Fig. 8 and in Fig. 13 in the supplementary material.

(a) *Source* frames are input to extract the *embedded* face which is drawn on. The modified *embedded* face is used to *generate* the frames below.



(b) An example sequence of *generated* frames (top row) from the modified *embedded* face controlled using a sequence of *driving* frames (bottom row).

Fig. 8: Example results of the video editing application. (a) For given *source* frames, the *embedded* face is extracted and modified. (b) The modified *embedded* face is used for a sequence of *driving* frames (bottom) and the result is shown (top). Note how for the second example, the blue tattoo disappears behind the nose when the person is seen in profile and how, as above, the modified *embedded* face can be driven using the same or another identity's pose and expression. Best seen in colour. Zoom in for details. Additional examples using the blue tattoo and Harry Potter scar are given in the accompanying video and in Fig. 13 in the supplementary material.

7 Conclusion

We have presented a self-supervised framework X2Face for driving face generation using another face. This framework makes no assumptions about the pose, expression, or identity of the input images, so it is more robust to unconstrained settings (e.g. an unseen identity). The framework can also be used with minimal alteration *post* training to drive a face using audio or head pose information. Finally, the trained model can be used as a video editing tool. Our model has achieved all this without requiring annotations for head pose/facial landmarks/depth data. Instead, it is trained self-supervised on a large collection of videos and learns itself to model the different factors of variation.
While our method is robust, versatile, and allows for generation to be conditioned on other modalities, the generation quality is not as high as approaches specifically designed for transforming faces (e.g. [1,17,40]). This opens an interesting avenue of research: how can the approach be modified such that the versatility, robustness, and self-supervision aspects are retained but with the generation quality of these methods that are specifically designed for faces. Finally, as no assumptions have been made that the videos are of faces, it is interesting to consider applying our approach to other domains.

Acknowledgements The authors are grateful to Hadar Averbuch-Elor for helpfully running their model on our data and to Vicky Kalogeiton for suggestions/comments. This work was funded by an EPSRC studentship and EPSRC Programme Grant Seebibyte EP/M013774/1.

References

1. Averbuch-Elor, H., Cohen-Or, D., Kopf, J., Cohen, M.F.: Bringing portraits to life. ACM Transactions on Graphics (Proceeding of SIGGRAPH Asia 2017) (2017)
2. Bas, A., Smith, W.A.P., Awais, M., Kittler, J.: 3D morphable models as spatial transformer networks. In: Proc. ICCV Workshop on Geometry Meets Deep Learning (2017)
3. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: Proc. ACM SIGGRAPH (1999)
4. Booth, J., Roussos, A., Ponniah, A., Dunaway, D., Zafeiriou, S.: Large scale 3d morphable models. IJCV **126**(2-4), 233–254 (Apr 2018)
5. Cao, J., Hu, Y., Yu, B., He, R., Sun, Z.: Load balanced gans for multi-view face image synthesis. arXiv preprint arXiv:1802.07447 (2018)
6. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: Proc. ICCV (2017)
7. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: NeurIPS (2016)
8. Chung, J.S., Senior, A., Vinyals, O., Zisserman, A.: Lip reading sentences in the wild. In: Proc. CVPR (2017)
9. Chung, J.S., Zisserman, A.: Out of time: automated lip sync in the wild. In: Workshop on Multi-view Lip-reading, ACCV (2016)

10. Dale, K., Sunkavalli, K., Johnson, M.K., Vlasic, D., Matusik, W., Pfister, H.: Video face replacement. ACM Transactions on Graphics (TOG) (2011)
11. Denton, E.L., Birodkar, V.: Unsupervised learning of disentangled representations from video. In: NeurIPS (2017)
12. Ding, H., Sricharan, K., Chellappa, R.: Exprgan: Facial expression editing with controllable expression intensity. In: Proc. AAAI (2018)
13. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proc. CVPR (2016)
14. Hassner, T., Harel, S., Paz, E., Enbar, R.: Effective face frontalization in unconstrained images. In: Proc. CVPR (2015)
15. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proc. CVPR (2017)
16. Karras, T., Aila, T., Laine, S., Herva, A., Lehtinen, J.: Audio-driven facial animation by joint end-to-end learning of pose and emotion. ACM Transactions on Graphics (TOG) (2017)
17. Kim, H., Garrido, P., Tewari, A., Xu, W., Thies, J., Nießner, M., Pérez, P., Richardt, C., Zollhöfer, M., Theobalt, C.: Deep video portraits. Proc. ACM SIGGRAPH (2018)
18. King, D.E.: Dlib-ml: A machine learning toolkit. The Journal of Machine Learning Research **10**, 1755–1758 (2009)
19. Koestinger, M., Wohlhart, P., Roth, P.M., Bischof, H.: Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization. In: Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies (2011)
20. Korshunova, I., Shi, W., Dambre, J., Theis, L.: Fast face-swap using convolutional neural networks. In: Proc. ICCV (2017)
21. Kulkarni, T.D., Whitney, W.F., Kohli, P., Tenenbaum, J.: Deep convolutional inverse graphics network. In: NeurIPS (2015)
22. Kumar, A., Alavi, A., Chellappa, R.: KEPLER: keypoint and pose estimation of unconstrained faces by learning efficient H-CNN regressors. In: Proc. Int. Conf. Autom. Face and Gesture Recog. (2017)
23. Nagrani, A., Chung, J.S., Zisserman, A.: VoxCeleb: a large-scale speaker identification dataset. In: INTERSPEECH (2017)
24. Nirkin, Y., Masi, I., Tran, A.T., Hassner, T., Medioni, G.: On face segmentation, face swapping, and face perception. In: Proc. Int. Conf. Autom. Face and Gesture Recog. (2018)
25. Olszewski, K., Li, Z., Yang, C., Zhou, Y., Yu, R., Huang, Z., Xiang, S., Saito, S., Kohli, P., Li, H.: Realistic dynamic facial textures from a single image using gans. In: Proc. ICCV (2017)
26. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: Proc. BMVC. (2015)
27. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch (2017)
28. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Transactions on graphics (TOG) (2003)
29. Pătrăucean, V., Handa, A., Cipolla, R.: Spatio-temporal video autoencoder with differentiable memory. In: NeurIPS (2016)
30. Qiao, F., Yao, N., Jiao, Z., Li, Z., Chen, H., Wang, H.: Geometry-contrastive generative adversarial network for facial expression synthesis. arXiv preprint arXiv:1802.01822 (2018)

31. Rav-Acha, A., Kohli, P., Rother, C., Fitzgibbon, A.: Unwrap mosaics: A new representation for video editing. In: ACM Transactions on Graphics (TOG) (2008)
32. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Proc. MICCAI (2015)
33. Roth, J., Tong, Y., Liu, X.: Adaptive 3D face reconstruction from unconstrained photo collections. In: Proc. CVPR (2016)
34. Saito, S., Wei, L., Hu, L., Nagano, K., Li, H.: Photorealistic facial texture inference using deep neural networks. In: Proc. CVPR (2017)
35. Saragih, J.M., Lucey, S., Cohn, J.F.: Real-time avatar animation from a single image. In: Proc. Int. Conf. Autom. Face and Gesture Recog. (2011)
36. Shlizerman, E., Dery, L., Schoen, H., Kemelmacher-Shlizerman, I.: Audio to body dynamics. Proc. CVPR (2018)
37. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
38. Suwajanakorn, S., Seitz, S.M., Kemelmacher-Shlizerman, I.: Synthesizing Obama: learning lip sync from audio. ACM Transactions on Graphics (TOG) (2017)
39. Tewari, A., Zollhöfer, M., Kim, H., Garrido, P., Bernard, F., Perez, P., Theobalt, C.: Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In: Proc. ICCV (2017)
40. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In: Proc. CVPR (2016)
41. Tran, A.T., Hassner, T., Masi, I., Paz, E., Nirkin, Y., Medioni, G.: Extreme 3D face reconstruction: Seeing through occlusions. In: Proc. CVPR (2018)
42. Tran, L., Yin, X., Liu, X.: Disentangled representation learning gan for pose-invariant face recognition. In: Proc. CVPR (2017)
43. Vlasic, D., Brand, M., Pfister, H., Popović, J.: Face transfer with multilinear models. ACM Transactions on Graphics (TOG) (2005)
44. Worrall, D.E., Garbin, S.J., Turmukhambetov, D., Brostow, G.J.: Interpretable transformations with encoder-decoder networks. In: Proc. ICCV (2017)
45. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. Proc. ICCV (2017)
46. Zollhöfer, M., Thies, J., Garrido, P., Bradley, D., Beeler, T., Pérez, P., Stamminger, M., Nießner, M., Theobalt, C.: State of the art on monocular 3D face reconstruction, tracking, and applications. In: Proc. Eurographics (2018)

Supplementary Material

We provide additional details about X2Face's architecture and the training and testing setup in Section A, and more qualitative results in Section B.

A Additional details on architectures and training.

Section A.3 provides qualitative results on using additional views at *test* time and the curriculum strategy. Section A.1 and Section A.2 give additional details about X2Face's two subnetworks: the *embedding network* and the *driving network*.

A.1 Embedding network

The *embedding network* is based on the pix2pix [15] version of the U-Net architecture [32], except that the last layer is changed to a 2-channel layer. The resulting $2 \times 256 \times 256$ gives the sampler which determines how to sample from the *source* frame to obtain the *embedded* face. The exact architecture is given in Fig. 9.

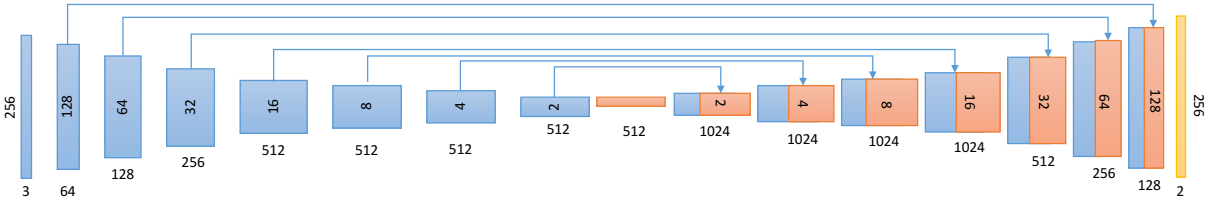


Fig. 9: *Embedding Network*: In the encoder, each convolution is followed by a leaky ReLU with factor 0.2 and then a batch-norm layer (except for the first which has no batch-norm). The convolutional filter sizes are 4×4 and the stride/padding is 2/1. In this decoder, the layers from the encoder are concatenated using skip connections, doubling the channel size. In the decoder, the sequence of executions following each convolutional layer is: ReLU, bilinear upsampling, batch-norm, concatenation (with the corresponding encoder layer). The last layer (yellow) does not have a concatenation. The final $2 \times 256 \times 256$ result is passed through a tanh layer to give X2Face's prediction for how to sample from the *source* frames. The convolutional filter sizes in the decoder are 3×3 and the stride/padding is 1/1.

A.2 Driving network

The *driving network* is an encoder-decoder network that uses the same layer sizes as the *embedding network* but does not have any skip connections. The *driving vector* corresponds to the vector resulting from the encoder portion of the network, which is a 128D vector. The architecture is shown in Fig. 10.

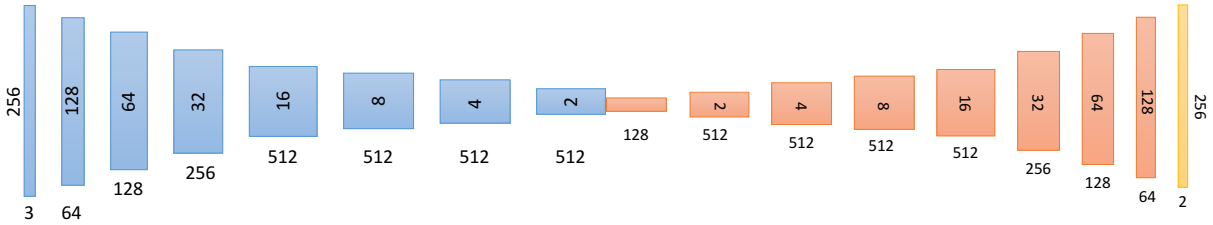


Fig. 10: *Driving Network*: The encoder is similar to the encoder of the *embedding network*; each convolution is followed by a leaky ReLU (factor 0.2) and then a batch-norm layer (except for the first which has no batch-norm); the convolutional filter sizes are 4×4 and the stride/padding is 2/1. In this decoder, there are no skip connections. The *driving vector* has channel size 128. In the decoder, the sequence of executions following each convolutional layer is: ReLU, bilinear upsampling, batch-norm. The final $2 \times 256 \times 256$ result is passed through a tanh layer to give X2Face's prediction for how to sample from the *embedded* face. The convolutional filter sizes in the decoder are 3×3 and the stride/padding is 1/1.

A.3 Qualitative results for different training/testing setups

We consider the following 4 different training/testing settings:

1. **Training stage I, Single source (S):** Training with photometric $L1$ loss only and testing with a single *source* frame.
2. **Training stage I, Multi-source (M):** Training with photometric $L1$ loss only and testing with multiple *source* frames.
3. **Training stage II, S:** Training with photometric $L1$ loss and identity losses as described in Section 3.2 in the paper and testing with a single *source* frame.
4. **Training stage II, M:** Training with photometric $L1$ loss and identity losses as described in Section 3.2 in the paper and testing with multiple *source* frames.

As can be seen in Fig. 11, testing with multiple *source* frames makes our method more robust, as the *generated* frame is not so unstable with respect to the choice of the *source* frame. This improves the quality and sharpness of the *generated* frames.
Although the results get better quantitatively when adding the identity loss functions (as discussed in the paper), we observe that our method produces visually very similar and convincing results when trained with the photometric $L1$ loss only. The use of the identity content losses is the only point where we make use of some form of labelled supervision, as we use a pre-trained network that was trained for identity classification. Using only the photometric loss results in a *completely self-supervised* framework.

B Additional qualitative results

We present more qualitative results comparing X2Face to CycleGAN [45] in Fig. 12 and show additional results for controlling the head pose of a *source*

(a) Single *source* frame

(b) *Embedded face* from a single *source* frame

(c) Multiple *source* frames

(d) *Embedded* face from multiple *source* frames

(e) Frame sequences *generated* for the same *driving* frames with different loss function/architecture choices.

Fig. 11: A comparison of different training/testing settings. (a) The *source* frame used when testing using a single *source* frame. (b) The *embedded* face for this single *source* frame. (c) The *source* frames used when testing with multiple *source* frames. (d) The *embedded* face from these multiple *source* frames. (e) *Generated* frames for different training stages and testing strategies. As the *driving* frame is from the same video, the *driving* frame corresponds to the ground truth and is shown in the bottom row. When using a single *source* frame for testing, the model is sensitive to the quality/pose of the *source* frame (see rows for *Training stage I, Single source (S)* and *Training stage II, S*). Using additional *source* frames and averaging over the resulting *embedded* face produces a better *embedded* face representation and higher quality results (see rows for *Training stage I, Multi-source (M)* and *Training stage II, M*).

frame using a pose vector in Fig. 15. Finally, additional qualitative results for using X2Face for video editing are given in Fig. 13 and Fig. 14.
For further demonstrations of X2Face in action, we refer to the accompanying video which can be found at http://www.robots.ox.ac.uk/~vgg/research/unsup_learn_watch_faces/x2face.html.

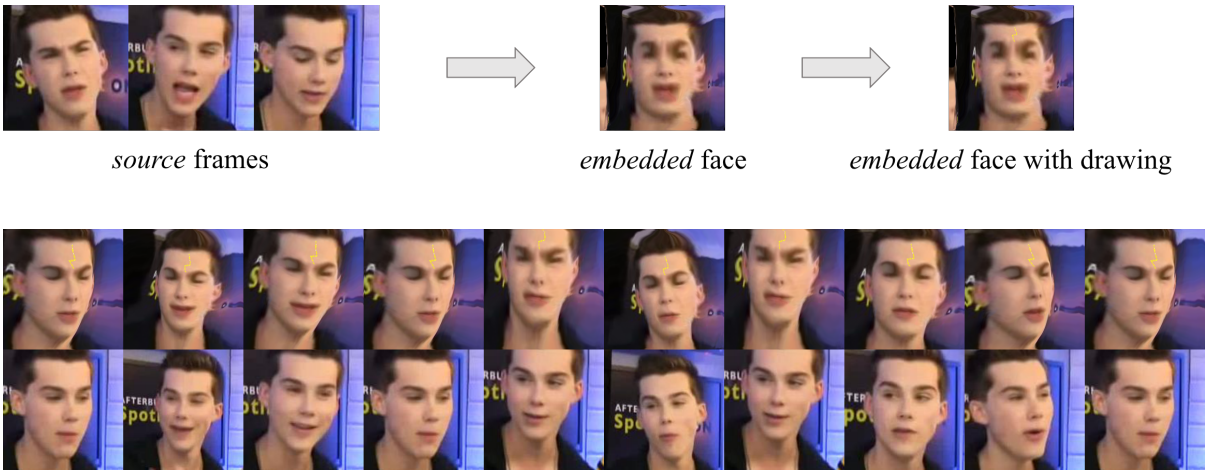
(a) *source* frames

(b)

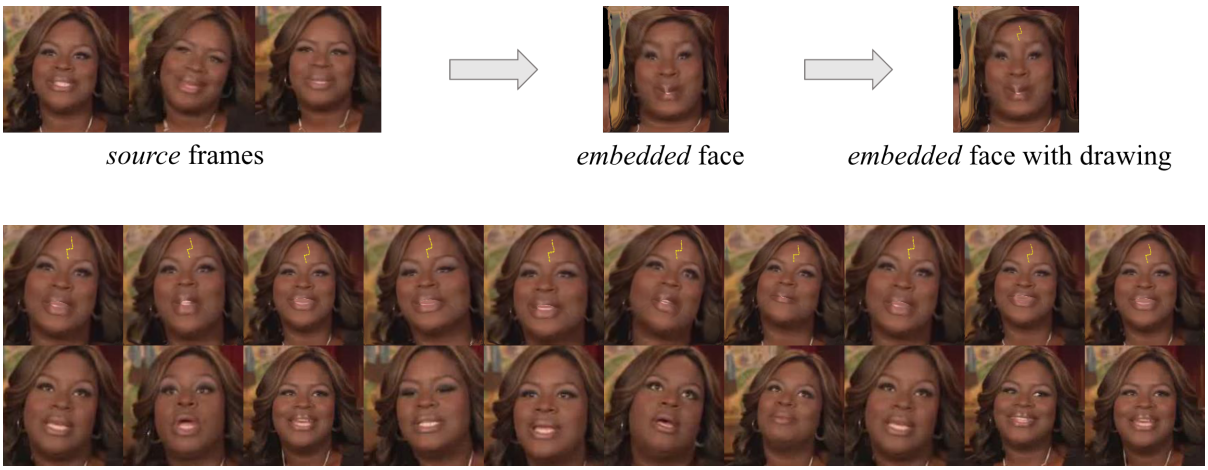
(c) *source* frames

(d)

Fig. 12: More qualitative results comparing X2Face to CycleGAN. Note how X2Face preserves the hair/background setting from the *source* frames, resulting in temporal coherence across the *generated* frames.

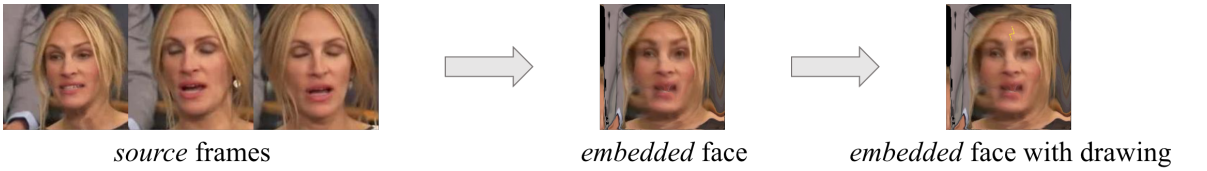


(b)



(d)

Fig. 13: Additional results of the video editing application using the Harry Potter scar. (a) For the given *source* frames, the *embedded* face is extracted and modified. (b) The modified *embedded* face is controlled using a sequence of *driving* frames (bottom) and the result is shown (top). Best viewed in colour. Zoom in for details.



(b)

(d)

Fig. 14: Additional results of the video editing application using the Harry Potter scar. (a) For the given *source* frames, the *embedded* face is extracted and modified. (b) The modified *embedded* face is controlled using a sequence of *driving* frames (bottom) and the result is shown (top). Best viewed in colour. Zoom in for details.

Fig. 15: Controlling the three head pose angles. This example can also be found in the accompanying video.

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	X2Face: A network for controlling face generation by using images, audio, and pose codes.
Publication Status	Published
Publication Details	Olivia Wiles, Almut Sophia Koepke, and Andrew Zisserman: *X2Face: A network for controlling face generation by using images, audio, and pose codes.* European Conference in Computer Vision (ECCV), 2018.

Student Confirmation

Student Name:	ALMUT SOPHIA KOEPKE		
Contributions to the Paper	Developing the initial ideas for the network architecture, self-supervised pretext task, and the controlling of face generation. Preparing the training dataset and the datasets for controlling face generation. Developing and implementing the frameworks including those for controlling image generation with head pose and audio. Running experiments and writing the paper.		
Signature	A. T. Koepke	Date	19/12/19

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 ANDREW ZISSERMAN		
Supervisor comments This was a beneficial, joint, and equal collaboration.		
Signature	Date	19/12/19

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

Chapter 4

Self-supervised learning of a facial attribute embedding from video

This work was accepted for *oral presentation* at the 29th British Machine Vision Conference (BMVC), 2018.

In this paper we introduce *FAb-Net*, a framework that embeds a face image to an embedding space that captures useful information, such as head pose, facial landmarks, and facial expression. This is learnt through the self-supervised proxy task of transforming one face image into another face image from the same video face track.

Self-supervised learning of a facial attribute embedding from video

Olivia Wiles*
ow@robots.ox.ac.uk
A. Sophia Koepke*
koepke@robots.ox.ac.uk
Andrew Zisserman
az@robots.ox.ac.uk

Visual Geometry Group
University of Oxford
Oxford, UK

Abstract

We propose a self-supervised framework for learning facial attributes by simply watching videos of a human face speaking, laughing, and moving over time. To perform this task, we introduce a network, **F**acial **A**ttri**b**utes-**Net** (FAb-Net), that is trained to embed multiple frames from the same video face-track into a common low-dimensional space. With this approach, we make three contributions: first, we show that the network can leverage information from multiple source frames by predicting confidence/attention masks for each frame; second, we demonstrate that using a curriculum learning regime improves the learned embedding; finally, we demonstrate that the network learns a meaningful face embedding that encodes information about head pose, facial landmarks and facial expression – i.e. facial attributes – *without* having been supervised with any labelled data. We are comparable or superior to state-of-the-art self-supervised methods on these tasks and approach the performance of supervised methods.

1 Introduction

Babies and children are highly perceptive to the facial expressions of the people they interact with [14, 18]. The ability to understand and respond to changes in people's emotional state is similarly important for computer vision systems and affective systems when interacting with a human user. Thus being able to predict head pose and expression is of vital importance.

Recently, leveraging deep learning has led to state-of-the-art results on a variety of tasks such as emotion recognition and facial landmarks detection. Despite these advances, supervised methods require large amounts of labelled data which may be expensive or difficult to obtain in realistic, unconstrained settings, or necessitate assigning data to ill-defined categories. For example, categorising emotions with three human annotators leads to only 46% agreement [2], and labelling pose in the wild is notoriously difficult. Moreover, performing each task independently does not leverage the fact that detecting landmarks requires understanding pose and facial features, which in turn correspond to expression.

Consequently, we consider the following question: is it possible to learn an embedding of facial attributes that encodes landmarks, pose, emotion, etc. in a self-supervised manner without *any* hand labelling? The learned embedding can then be used for another task

* Equal contribution.

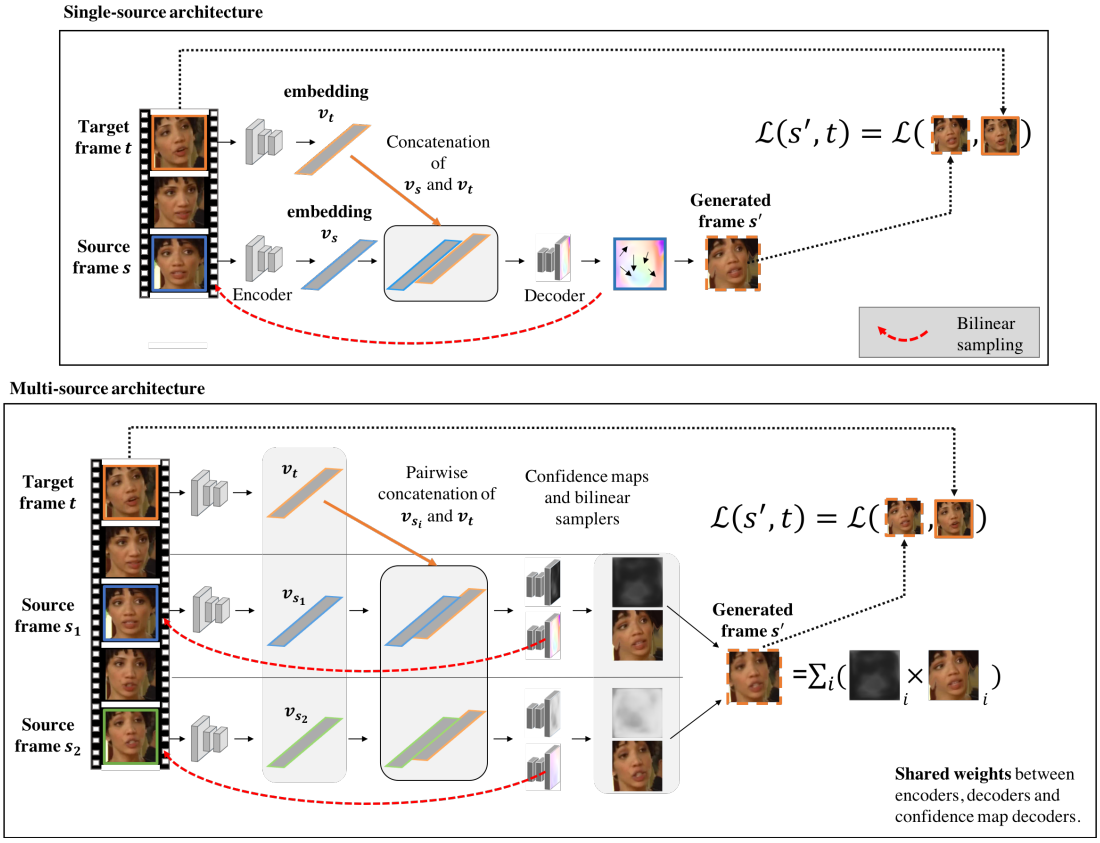


Figure 1: An overview of FAb-Net. In the single-source case (top), the encoder-decoder architecture takes one source frame and a target frame as inputs and learns to generate the target frame. The 256-dimensional outputs of the encoders – the source and target attribute embeddings – are concatenated and input to the decoder. The decoder predicts the point-to-point correspondence from the source frame to the target frame, and RGB values for the generated frame are then obtained from the source frame using a bilinear sampler. The network is trained with an $L1$ loss between the generated frame and the target frame. In the multi-source case (bottom), the decoder also predicts a confidence map, and the confidence maps are used to weight the contributions of the different source frames.

(e.g. landmark, pose, and expression prediction) using a linear layer. To do this, we contribute FAb-Net, a self-supervised framework for learning a low-dimensional face embedding for facial attributes (Section 3). We take advantage of video data which contains a large collection of images of the same person from different viewpoints and with varied expressions. Given only the embeddings corresponding to a source and target frame, the network is tasked to map the source frame to the target frame by predicting a flow field between them. This proxy task forces the network to distil the information required to compute the flow field (e.g. the head pose and expression) into the source and target embeddings. After explaining the setup for a single source frame in Section 3.1, we introduce our additional contributions: a method for leveraging multiple frames in order to improve the learned embedding in Section 3.2; and how a curriculum strategy for training FAb-Net in Section 4 can be used to improve performance.

The learned embedding is extracted and used for a variety of tasks such as landmark

detection, pose regression, and expression classification in Section 5 by simply learning a linear layer. Our results on these tasks are comparable or superior to other self-supervised methods, approaching the performance of supervised methods. These experiments verify the hypothesis that our self-supervised framework learns to encode facial attributes which are useful for a variety of tasks. Finally, the method is tested qualitatively by using the learned embedding to retrieve images with similar facial attributes across different identities.

2 Related Work

Self-supervised learning. Self-supervised methods such as [12, 38, 43, 62] require no manual labelling of the images; instead, they directly use image data to provide proxy supervision for learning good feature representations. The benefit of these approaches is that the features learned using large quantities of available data can be transferred to other tasks/domains which have less or even no annotated data. To provide further supervision from image data, the images themselves can be transformed via a synthetic warp or rotation and the network trained to recognise the rotation [19] or to learn equivariant pixel embeddings [40, 51, 52].

Of more direct relevance to our training framework are self-supervised frameworks that use video data ([1, 8, 11, 16, 17, 21, 22, 30, 34, 44, 55, 56, 58, 61]). Our approach builds in particular on those that use frame synthesis [8, 11, 22, 44, 56, 58], though for us synthesis is a proxy task rather than the end goal. Note, unlike [16, 30, 34, 55], we do not make use of the temporal ordering information inherent in a video; nor do we predict future frames conditioned on a number of past frames [44], or explicitly predict the motion between frames as a convolutional kernel [22, 58], or condition the generation on another modality (e.g. voice [8]). Instead, we treat the frames as an unordered set, and propose a simple formulation: that by embedding the source and target frames in a common space and conditioning the transformation from source to target frame on these embeddings, the learned embeddings must learn to encode the relevant modes of variation necessary for the transformation.

Concurrent to our work, Zhang *et al*. [64] and Jakab *et al*. [21] build on [51] by using the discovered landmarks to reconstruct the original image. However, unlike these works, we do not place any constraints on the learned representation – such as an explicit representation that encodes landmarks as heatmaps.

Supervised learning of face embeddings. Given *known* (labelled) attribute information, e.g. for pose or expression, the embedding can be learned by training in a supervised manner to directly predict the attribute [28, 32, 46], or to generate images (of faces, cars, or other classes) at a new, *known* pose, expression, etc. [13, 25, 53, 59, 68]. Another way of supervising a face embedding is to explicitly learn the parameters of a 3D morphable model (3DMM) [7]. As fitting a 3DMM is relatively expensive, [3, 50] learn this end-to-end using either landmarks or a photometric error as supervision. However, unlike our method, these methods require either ground truth labels or a morphable model which fixes the modes of variation and the embedding.

An interesting half-way point is weak supervision, where the learned object or face embedding is conditioned for instance on object labels [39] or weather/geo-location information [31] respectively. This requires additional meta-data, but results in embeddings that can represent attributes such as age and expression for faces or keypoints for objects.

3 Method

The aim is to train a network to learn an embedding that encodes facial attributes in a self-supervised manner, without any labels. To do this, the network is trained to generate a target frame from one or multiple source frames by learning how to transform the source into the target frame. The source and target frames are taken from the same face-track of a person speaking, i.e. the frames are of the same identity but with different expressions/poses. An overview of the architecture is given in Fig. 1 and further described for a single source frame in Section 3.1 and for multiple source frames in Section 3.2 (additional details are given in the supplementary material).

3.1 Single-source frame architecture

The input to the network is a source frame s and a target frame t from the same face-track. These are passed through encoders with shared weights which learn a mapping f from the input frames to a 256-dimensional vector embedding (as shown in Fig. 1). The embeddings corresponding to the target and source frames are $v_t = f(t)$ and $v_s = f(s)$ respectively. The source and target embeddings are concatenated to give a 512-dimensional vector which is upsampled via a decoder. The decoder learns a mapping g from the concatenated embeddings to a bilinear grid sampler, which samples from the source frame to create a new, generated frame $s' = g(v_t, v_s)(s)$. Precisely, g predicts offsets $(\delta x, \delta y)$ for each pixel location (x, y) in the target frame; the generated frame s' at location (x, y) is obtained by sampling from the source frame s according to these offsets : $s'(x, y) = s(x + \delta x, y + \delta y)$. The network is trained to minimise the $L1$ loss between the generated and the target frame: $\mathcal{L}(s', t) = ||t - s'||_1$.

This setup enforces that the embeddings v_s and v_t represent facial attributes of the source and target frames respectively since the decoder maps from the source frame s to generate the frame s' (i.e. it uses pixel RGB values from the source frame s to create the generated s' – a similar formulation has been proposed concurrently to this work by [54]). As the decoder is a function of the target and source attribute embeddings, and the decoder is the only place in the network where information is shared, the target attribute embedding must encode information about expression and pose in order for the decoder to know where to sample from in the source frame and where to place this information in the generated frame.

3.2 Multi-source frames architecture

While using two frames for training enforces that the network learns a high-quality embedding, additional source frames can be leveraged to improve the learned embedding. This is achieved by also predicting a confidence heatmap – a 1 channel image – for each source frame via an additional decoder. The heatmaps denote how confident the network is of the flow at each pixel location – e.g. if the source frame has a very different pose than the target frame, the confidence heatmap would have low certainty. Moreover, it can express this for sub-parts of the image; if the mouth is closed in the source but open in the target frame, the confidence heatmap can express uncertainty in this region. The confidence heatmaps C_i are combined pixel-wise for each source frame s_i using a soft-max operation. For n source frames, the loss function to be minimised is given as $\mathcal{L} = ||t - \frac{\sum_{i=1}^{n} e^{C_i} * (g(v_t, v_{s_i})(s_i))}{\sum_{i=1}^{n} e^{C_i}}||_1$.

4 Curriculum Strategy

The training of the network is divided into stages, so that knowledge can be built up over time as the examples given become progressively more difficult, as inspired by [5, 27]. The loss computed by a forward pass is used to rank samples (i.e. source and target frame pairs) in the batch according to their difficulty in a manner similar to [33, 37, 48, 49]. However, these methods use only the most difficult samples, which was found to stop our network from learning. Similarly to [37], using progressively more difficult samples proved crucial for the strategy's success.

Given a batch size of N randomly chosen samples, i.e. source and target frame pairs, a forward pass is executed and the loss for each sample computed. The samples are ranked and sorted according to this loss. Initially the loss is back-propagated only on the samples in the batch which are within the 1st to 50th percentile (i.e. the $0.5N$ samples with the lowest loss computed by the forward pass). These are assumed to be easier samples. When the loss on the validation set plateaus, the subset to be back-propagated on is shifted by 10 (e.g. the samples in the 10th-60th percentile range). This is repeated 4 times until the samples being back-propagated on fall into the 40th-90th percentile range. At this point the curriculum strategy is terminated, as it is assumed that the samples in the 90-100th range are too challenging or may be problematic (e.g. there is a large shift in the background which is too challenging to learn).

5 Experiments

In this section, we evaluate the network and the learned embedding. In Section 5.1, the performance of using FAb-Net's learned representation is compared to that of state-of-the-art self-supervised and supervised methods on a variety of tasks: facial landmark prediction, head pose regression and expression classification. Section 5.2 discusses the benefit of using additional source frames, and Section 5.3 shows how the learned representation can be used for retrieving images with similar facial attributes.

Training. The model is trained on the VoxCeleb1 and VoxCeleb2 video datasets [9, 36]; we refer to the combined datasets as VoxCeleb+. The VoxCeleb+ dataset consists of videos of interviews containing more than 1 million utterances of around 7,000 speakers. The frames are extracted at 1 fps. The frames are cropped, resized to 256×256, and the identities are randomly split into train/val/test (with a split of 75/15/10).

The models are trained in PyTorch [42] using SGD, an initial learning rate of 0.001, and momentum 0.9. When using the curriculum strategy described in Section 4, the batchsize is $N = 32$, else $N = 8$. The learning rate is divided by a factor of 10 when the loss on the validation set plateaus. (If the curriculum strategy is used, the learning rate is updated only when the 40-90th percentile is considered.) This is repeated until the loss converges. Further details about the training can be found in the supplementary material.

5.1 Using the embedding for regression and classification

First, we investigate the representation learned in our embedding and evaluate whether it indeed encodes facial attributes by challenging it to predict three different attributes: landmarks, pose, and expression.

Setup. Given a network trained on VoxCeleb+, a linear regressor or classifier is trained from the learned embedding to the output task. The linear regressor/classifier consists of two layers: batch-norm [20] followed by a linear fully connected layer with no bias. The regression tasks are trained using a MSE loss. The classification tasks are trained with a cross-entropy loss. The parameters of the encoder are fixed while the two additional layers are trained on the training set of the target dataset using Adam [23], a learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

5.1.1 Baselines

Self-supervised. There are prior publications on using self-supervision for landmark prediction on the datasets we evaluate on, but none for predicting emotion on standard datasets. Consequently, we implement an autoencoder and a set of state-of-the-art self-supervised methods [19, 63] for object detection and segmentation. The baselines are trained using the same architecture as FAb-Net but with their associated loss functions and training objectives. For [63], the regression loss for both the L and ab channels is used. These models are trained on VoxCeleb+ until convergence, with the same training parameters and data augmentation as FAb-Net. More details are given in the supplementary material.

VGG-Face descriptor. We additionally compare to the *VGG-Face descriptor* which is obtained from the 4096-dimensional FC7 features from a VGG-16 network trained on the VGG-Face dataset [41]. It has recently been shown that a network trained for identity does, contrary to popular belief, retain information about other facial attributes [10, 15]. We use the VGG-Face descriptor to learn a linear regression/classification layer to the desired attribute task. This provides a strong baseline, and the results obtained confirm the finding that a network trained for identity does indeed encode expression and to some extent also pose information. However, note that unlike our method, this face descriptor requires a large dataset of *labelled* face images for training.

5.1.2 Results

Facial landmarks. Facial landmark locations are regressed from the learned embedding and compared to state-of-the-art methods on MAFL [66] and the more challenging 300-W [47] datasets. The evaluation is performed as outlined in [51, 66], and the errors given in inter-ocular distance. For MAFL, 5 facial landmarks are regressed for 19k/1k train/test images. For 300-W, 68 landmarks are regressed for 3148/689 train/test images which are obtained (as described in [51]) from combining multiple datasets [4, 67, 70].

The results are reported in Table 1 and some qualitative results are visualised in Fig. 2 and Fig. 3. These results first demonstrate that fine-tuning with additional views and our curriculum strategy improve the embedding learned by FAb-Net. Second, these results show that our method performs competitively or better than state-of-the-art unsupervised landmark detection methods, better than the VGG-Face descriptor baseline and competitively with state-of-the-art supervised methods. This is achieved even though the other self-supervised methods [21, 51, 52, 64] are explicitly engineered to detect landmarks whereas our method is not. In addition to that, our method is able to bridge the domain gap between VoxCeleb+ and CelebA [32] – the other self-supervised methods that we compare to are pre-trained on CelebA.

Pose. The learned embedding is used for pose prediction and compared to a supervised method [26] and to using the VGG-Face descriptor. To perform the evaluation, the linear

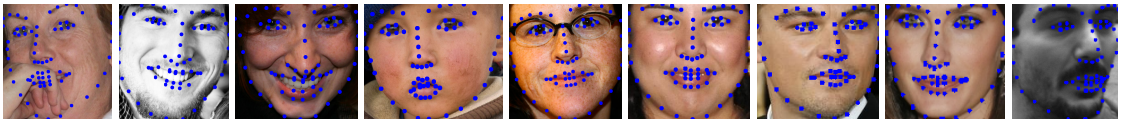
Method	300-W	MAFL
Self-supervised		
Trained on VoxCeleb+		
FAb-Net	6.31	3.78
FAb-Net w/ curric.	5.73	3.49
FAb-Net w/ curric., 3 source frames	**5.71**	3.44
Trained on CelebA		
Jakab *et al.* [21] (2018)	–	**3.08**
Zhang *et al.* [64] (2018)	–	3.15
Thewlis *et al.* [51] (2017)	9.30	6.67
Thewlis *et al.* [52] (2017)	7.97	5.83
Supervised		
Trained on CelebA		
MTCNN [65] (2014)	–	**5.39**
LBF [45] (2014)	6.32	–
CFSS [69] (2015)	5.76	–
cGPRT [29] (2015)	5.71	–
DDN [60] (2016)	5.65	–
TCDCN [66] (2016)	5.54	–
RAR [57] (2016)	**4.94**	–
VGG-Face descriptor [41]	11.16	5.92

Table 1: Landmark prediction error on 300-W and MAFL datasets. Lower is better.

Method	Roll	Pitch	Yaw	MAE
Self-supervised				
FAb-Net	5.54°	7.84°	12.93°	8.77°
FAb-Net w/ curric.	5.33°	7.21°	11.34°	7.96°
FAb-Net w/ curric., 3 source frames	**5.14°**	**7.13°**	**10.70°**	**7.65°**
Supervised				
VGG-Face descriptor [41]	**8.24°**	8.36°	18.35°	11.65°
KEPLER [26] (2017)	8.75°	**5.85°**	**6.45°**	**7.02°**

Table 2: Pose prediction error on the AFLW test set from [26]. Lower is better.

Figure 2: Landmark prediction visualisation for FAb-Net on the MAFL dataset. A dot denotes ground truth and the cross FAb-Net's prediction. A failure case is shown to the right.



(a) Ground truth landmarks

(b) FAb-Net's predicted landmarks

Figure 3: Landmark prediction visualisation for FAb-Net on the 300-W dataset.

regression is trained from the given embedding to head pose labels using the AFLW dataset [24], but after leaving out the 1,000 images of the AFLW test set from [26]. As can be seen in Table 2, FAb-Net performs better in predicting the roll angle, and the MAE is comparable to [26] which is supervised with head pose labels. Furthermore, our embedding outperforms the VGG-Face descriptor which is trained on identities; i.e. our learned embedding encodes more information about head pose.

Expression. We evaluate the performance of our learned embedding for expression estimation on two datasets: AffectNet [35] and EmotioNet [6], which both contain over 900,000 images. These datasets are taken 'in-the-wild' as opposed to in a constrained environment. AffectNet contains 8 facial expressions (neutral, happy, sad, surprise, fear, disgust, anger, contempt) and EmotioNet contains 11 action units (AUs) (combinations of AUs correspond

	AUC for different AUs											
	1	2	4	5	6	9	12	17	20	25	26	avg.
Self-supervised												
FAb-Net	72.0	68.9	73.2	69.4	88.2	78.6	89.5	71.0	75.9	81.4	72.0	76.4
FAb-Net w/ curriculum	73.4	71.8	75.3	67.8	90.4	78.8	91.9	72.4	74.5	**83.7**	73.3	77.6
FAb-Net w/ curriculum, 3 source frames	**74.1**	**72.3**	**75.8**	**68.8**	**90.7**	**81.8**	**92.5**	**73.7**	**77.2**	83.6	**73.6**	**78.6**
Gidaris *et al*. [19]	68.6	64.0	72.8	70.0	83.9	78.1	83.8	68.4	72.6	73.1	67.2	72.9
SplitBrain [63]	65.5	59.8	66.7	60.8	71.8	65.8	73.3	64.5	57.4	68.1	61.1	65.0
Autoencoder	67.2	60.5	70.1	65.1	79.6	70.4	80.1	68.3	66.5	70.5	64.1	69.3
Supervised												
VGG-Face descriptor [41]	**81.8**	**83.0**	83.5	81.8	92.0	**90.9**	95.7	**80.6**	85.2	86.5	73.0	84.9
VGG-11 (from scratch)	74.7	77.2	**85.8**	**83.7**	**93.8**	89.7	**97.5**	78.3	**86.9**	**96.4**	**81.5**	**86.0**

Table 3: Expression classification results for state-of-the-art self-supervised and supervised methods on EmotioNet [6] for multiple facial action units (AUs). Higher is better for AUC.

	AUC								
	Neutral	Happy	Sad	Surprise	Fear	Disgust	Anger	Contempt	avg.
Self-supervised									
FAb-Net	70.0	87.6	68.8	75.5	76.5	70.0	73.2	71.2	74.2
FAb-Net w/ curric.	71.5	90.0	70.8	78.2	77.4	72.2	75.7	72.1	76.0
FAb-Net w/ curric., 3 source frames	**72.3**	**90.4**	**70.9**	**78.6**	**77.8**	**72.5**	**76.4**	**72.2**	**76.4**
Gidaris *et al*. [19]	67.8	84.9	69.0	73.9	75.7	69.8	71.5	68.7	72.7
SplitBrain [63]	63.9	74.7	64.2	61.3	68.3	58.6	68.2	62.8	65.3
Autoencoder	65.8	80.0	64.7	66.1	70.6	63.4	68.3	65.0	68.0
Supervised									
VGG-Face descriptor [41]	75.9	92.2	80.5	81.4	82.3	81.4	81.2	77.1	81.5
AlexNet [35]	–	–	–	–	–	–	–	–	**82**

Table 4: Expression classification results for state-of-the-art self-supervised and supervised methods on AffectNet [35]. Higher is better for AUC.

to facial expressions).

Both of these datasets were organised for challenges with a held out, unreleased test set. Therefore, the train set is subdivided into two subsets; one is used for training and the other for validation. The validation set of the original dataset is used to test the different models. The linear classifier for EmotioNet is trained with a binary cross-entropy loss for each AU, whereas for AffectNet, a cross-entropy loss is used. Both training datasets are highly imbalanced. As a result, the examples from the under-represented classes are re-weighted inversely proportionally to the class frequencies to penalise the loss more heavily for mis-classifying images of the under-represented classes.

The embedding learned by FAb-Net is compared to a number of self-supervised and supervised methods by measuring the Area Under the ROC curve (AUC). For each class (e.g. emotion or AU), the AUC is computed independently and the result is averaged over all classes. The results are reported in Table 3 and Table 4 for EmotioNet and AffectNet respectively showing that our network performs better than other self-supervised methods over both metrics when given the same training data. This is supposedly due to the fact that the network must learn to transform the source frame in order to generate the target frame. As parts of the face move together (e.g. an eyebrow raise or the lips when the mouth opens), the embedding must learn to encode information about facial features and thereby encode expression. Interestingly, the autoencoder performs well, presumably due to the restricted nature of this domain.

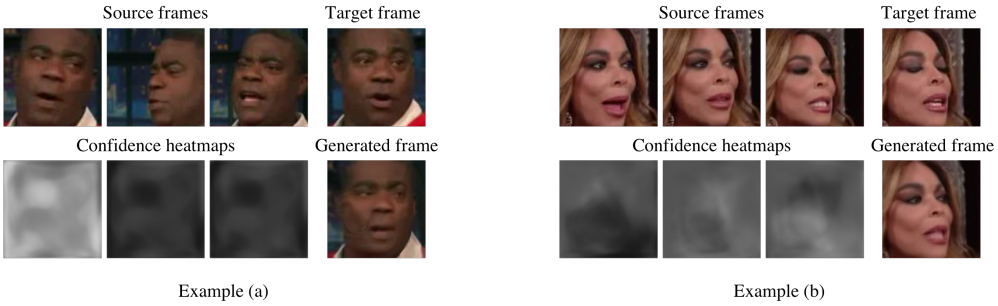


Figure 4: Confidence heatmaps learned by FAb-Net. Higher intensity corresponds to higher confidence. The network chooses the frames with most similar poses to draw from and ignores those with less similar poses (see Example (a)). In Example (b), the mouth has higher confidence in the third source frame allowing the network to re-construct the teeth that are present in the target frame. More examples can be found in the supplementary material.

FAb-Net is also not far off supervised methods despite the domain shift; VoxCeleb+ consists only of people being interviewed (and consequently with mostly neutral/smiling faces), so it does not include the range/extremity of expressions found in AffectNet or EmotioNet. Finally, it can be observed that the VGG-Face descriptor trained to predict identities does surprisingly well at predicting emotion.

Discussion. FAb-Net has achieved impressive performance, as most self-supervised methods when transferred to another task have a large gap in comparison to supervised methods. There is no gap or a small gap for the smaller datasets (landmarks/pose) and the model approaches supervised performance for the larger datasets (expression).

5.2 What is the benefit of additional source frames?

The previous sections have shown that using additional source frames improves performance. This is at the expense of performing additional forward passes through the encoder (in this case two). Given enough GPU memory, these forward passes can be done in parallel, affecting only the memory requirements and not the computational speed.

Using multiple source frames is further investigated by visualising confidence heatmaps for a given set of source frames in Fig. 4. The confidence heatmaps allow images with more similar pose to be used for creating the generated frame. Furthermore, the network can focus on one frame for generating a part of the face (e.g. the mouth) and on another one for a different part.

5.3 Image retrieval

This section considers an application of the learned embedding: retrieving images with similar facial attributes (e.g. pose) but across different identities. To perform this task, a subset of 10,000 randomly sampled test images from VoxCeleb+ is obtained. For a given query image, all other images (the gallery) are ranked based on their similarity to the query image using the cosine similarity metric between the corresponding embeddings. For a given query image Q, the embedding x_q is extracted by performing a forward pass through the network. Similarly, the embedding x_i is extracted for each image I_i in the gallery. Each image I_i is then ranked according to the cosine similarity between x_q and x_i. If the network does indeed

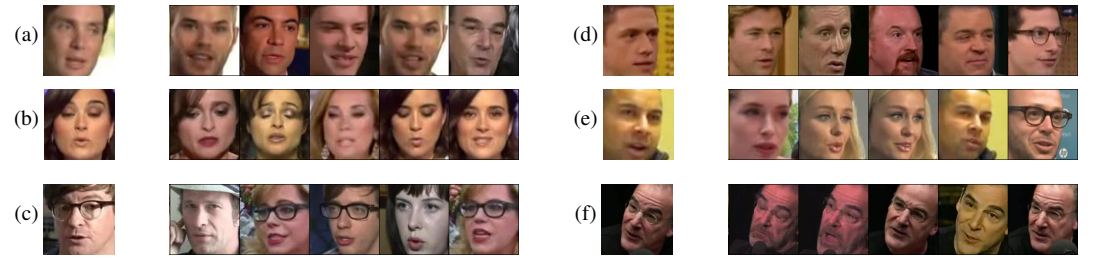


Figure 5: Retrieval results using the embedding learned by FAb-Net. The embedding captures similar visual attributes since gallery images with similar facial attributes are retrieved for a given query image. The retrieved images have similar pose to the query in all cases, and the expression similarity can be seen for example in (b) with the eyes shut, and (a) with the mouth slightly open. Please refer to the supplementary material for additional examples.

encode salient information about facial attributes, the cosine similarity can be used to identify images with similar poses and facial attributes. For a set of query images, the results are visualised in Fig. 5. From these results it is again affirmed that our embedding encodes information about facial attributes, as the retrieved images have poses and expressions similar to those of the query images. Note, the embedding is largely unaffected by facial decorations (e.g. glasses) and identity, as these do not change within a face-track and so do not need to be learned in order to predict the transformation.

6 Conclusion

We have introduced FAb-Net: a self-supervised framework for learning facial attributes from videos. Our method learns about pose and expression by watching faces move and change over a large number of videos without *any* hand labels. The features of our trained network can then be used to predict pose, landmarks, and expression on other datasets (despite the domain shift) by just training a linear layer on top of the learned embedding. The features have been shown to be comparable or superior performance to self-supervised and supervised methods on a variety of tasks. This is impressive as generally the performance of self-supervised methods has been found to be worse than that of supervised methods, yet our method is indeed competitive/superior to supervised methods for pose regression and facial landmark detection, and approaches supervised performance on expression classification.

7 Acknowledgements

The authors would like to thank James Thewlis for helpfully sharing code and datasets. This work was funded by an EPSRC studentship and EPSRC Programme Grant Seebibyte EP/M013774/1.

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *Proc. ICCV*, 2015.

[2] E. Barsoum, C. Zhang, C. C. Ferrer, and Z. Zhang. Training deep networks for facial expression recognition with crowd-sourced label distribution. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 279–283. ACM, 2016.

[3] A. Bas, P. Huber, W. A. P. Smith, M. Awais, and J. Kittler. 3D morphable models as spatial transformer networks. In *Proc. ICCV Workshop on Geometry Meets Deep Learning*, 2017.

[4] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. *IEEE PAMI*, 2013.

[5] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proc. ICML*, 2009.

[6] C. F. Benitez-Quiroz, R. Srinivasan, and A. M. Martinez. EmotioNet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In *Proc. CVPR*, 2016.

[7] V. Blanz, S. Romdhani, and T. Vetter. Face identification across different poses and illumination with a 3D morphable model. In *Proc. AFGR*, 2002.

[8] J. S. Chung, A. Jamaludin, and A. Zisserman. You said that? In *Proc. BMVC.*, 2017.

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

[10] F. Cole, D. Belanger, D. Krishnan, A. Sarna, I. Mosseri, and W. T. Freeman. Synthesizing normalized faces from facial identity features. In *Proc. CVPR*, 2017.

[11] E. L. Denton and V. Birodkar. Unsupervised learning of disentangled representations from video. In *NeurIPS*, 2017.

[12] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proc. ICCV*, 2015.

[13] A. Dosovitskiy, J. T. Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *Proc. CVPR*, 2015.

[14] P. Ekman and H. Oster. Facial expressions of emotion. *Annual review of psychology*, 30(1):527–554, 1979.

[15] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *Proc. ACM SIGGRAPH*, 2018.

[16] B. Fernando, H. Bilen, E. Gavves, and S. Gould. Self-supervised video representation learning with odd-one-out networks. In *Proc. CVPR*, 2017.

[17] C. Gan, B. Gong, K. Liu, H. Su, and L. J. Guibas. Geometry guided convolutional neural networks for self-supervised video representation learning. In *Proc. CVPR*, 2018.

[18] F. C. Gerull and R. M. Rapee. Mother knows best: effects of maternal modelling on the acquisition of fear and avoidance behaviour in toddlers. *Behaviour research and therapy*, 40(3):279–287, 2002.

[19] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *Proc. ICLR*, 2018.

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

[21] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi. Conditional image generation for learning the structure of visual objects. *arXiv preprint arXiv:1806.07823*, 2018.

[22] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool. Dynamic filter networks. In *NeurIPS*, 2016.

[23] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *Proc. ICLR*, 2015.

[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 *Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[25] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *NeurIPS*, 2015.

[26] A. Kumar, A. Alavi, and R. Chellappa. KEPLER: keypoint and pose estimation of unconstrained faces by learning efficient H-CNN regressors. In *Proc. Int. Conf. Autom. Face and Gesture Recog.*, 2017.

[27] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *NeurIPS*. 2010.

[28] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Describable visual attributes for face verification and image search. *IEEE PAMI*, 33(10):1962–1977, 2011.

[29] D. Lee, H. Park, and C. D. Yoo. Face alignment using cascade gaussian process regression trees. In *Proc. CVPR*, 2015.

[30] H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang. Unsupervised representation learning by sorting sequences. In *Proc. ICCV*, 2017.

[31] Y. Li, R. Wang, H. Liu, H. Jiang, S. Shan, and X. Chen. Two birds, one stone: Jointly learning binary code for large-scale face image retrieval and attributes prediction. In *Proc. ICCV*, 2015.

[32] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proc. ICCV*, 2015.

[33] I. Loshchilov and F. Hutter. Online batch selection for faster training of neural networks. *ICLR Workshops*, 2016.

[34] I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *Proc. ECCV*, 2016.

[35] A. Mollahosseini, B. Hasani, and M. H. Mahoor. AffectNet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 2017.

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

[37] A. Nagrani, S. Albanie, and A. Zisserman. Learnable pins: Cross-modal embeddings for person identity. *arXiv preprint arXiv:1805.00833*, 2018.

[38] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. ECCV*, 2016.

[39] D. Novotny, D. Larlus, and A. Vedaldi. Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *Proc. CVPR*, 2017.

[40] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi. Self-supervised learning of geometrically stable features through probabilistic introspection. In *Proc. CVPR*, 2018.

[41] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proc. BMVC.*, 2015.

[42] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. 2017.

[43] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proc. CVPR*, 2016.

[44] V. Pătrăucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *NeurIPS*, 2016.

[45] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *Proc. CVPR*, 2014.

[46] E. M. Rudd, M. Günther, and T. E. Boult. Moon: A mixed objective optimization network for the recognition of facial attributes. In *Proc. ECCV*, 2016.

[47] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: Database and results. *Image and Vision Computing*, 47:3–18, 2016.

[48] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *Proc. CVPR*, 2016.

[49] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proc. ICCV*, 2015.

[50] A. Tewari, M. Zollhöfer, H. Kim, P. Garrido, F. Bernard, P. Perez, and C. Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proc. ICCV*, 2017.

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

[52] J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *NeurIPS*, 2017.

[53] L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proc. CVPR*, 2017.

[54] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy. Tracking emerges by colorizing videos. In *Proc. ECCV*, 2018.

[55] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *Proc. ICCV*, 2015.

[56] O. Wiles, A. S. Koepke, and A. Zisserman. X2Face: A network for controlling face generation using images, audio, and pose codes. In *Proc. ECCV*, 2018.

[57] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, and A. Kassim. Robust facial landmark detection via recurrent attentive-refinement networks. In *Proc. ECCV*, 2016.

[58] T. Xue, J. Wu, K. Bouman, and B. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NeurIPS*, 2016.

[59] J. Yang, P. Ren, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. In *Proc. CVPR*, 2017.

[60] X. Yu, F. Zhou, and M. Chandraker. Deep deformation network for object landmark localization. In *Proc. ECCV*, 2016.

[61] A. R. Zamir, T. Wekel, P. Agrawal, C. Wei, J. Malik, and S. Savarese. Generic 3d representation via pose estimation and matching. In *Proc. ECCV*, 2016.

[62] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *Proc. ECCV*, 2016.

[63] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proc. CVPR*, 2017.

[64] 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.

[65] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *Proc. ECCV*, 2014.

[66] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning deep representation for face alignment with auxiliary attributes. *IEEE PAMI*, 2016.

[67] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *ICCVW*, 2013.

[68] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. In *Proc. ECCV*, 2016.

[69] S. Zhu, C. Li, C. C. Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *Proc. CVPR*, 2015.

[70] X. Zhu and D. Ramanan. Face detection, pose estimation and landmark localization in the wild. *Proc. CVPR*, 2012.

Supplementary Material: Self-supervised learning of a facial attribute embedding from video

Olivia Wiles*
ow@robots.ox.ac.uk
A. Sophia Koepke*
koepke@robots.ox.ac.uk
Andrew Zisserman
az@robots.ox.ac.uk

Visual Geometry Group
University of Oxford
Oxford, UK

We provide additional details about FAb-Net's architecture and training in Section 1, the self-supervised baselines in Section 2, pre-processing of the datasets in Section 3 and additional qualitative results in Section 4.

1 Additional details on architectures and training

Section 1.1 and Section 1.2 give additional details about the encoder-decoder architecture used and about the training respectively.

1.1 Architecture

The architecture of the encoders and decoders is based on pix2pix [12] but without skip-connections (see Fig. 1). It consists of encoders (with shared weights) and corresponding symmetrical decoders (also with shared weights). The face embedding vectors which are the outputs of the encoder have channel size 256. At the centre of the network, the source embedding vectors are concatenated pairwise with the target embedding vector. This 512-channel vector serves as input to the decoders.

The sampler decoders predict a $2 \times 256 \times 256$ output which determines how to sample from the source frame. When using multiple source frames, the network is augmented with confidence map decoders whose architecture is identical to the sampler decoders, except that the last layer outputs a 1-channel image. The confidence decoders also have shared weights for the different source frames.

1.2 Training and data augmentation

The images are first scaled to size 256×256. Because VoxCeleb1/VoxCeleb2 have different crops, VoxCeleb2 is re-cropped by padding the images by $[20, 80, 20, 30]$ pixels to the left, top, right, and bottom respectively and then taking a centre crop of size 190×190. Given the re-cropped VoxCeleb2 images and the VoxCeleb1 images, the images are augmented by

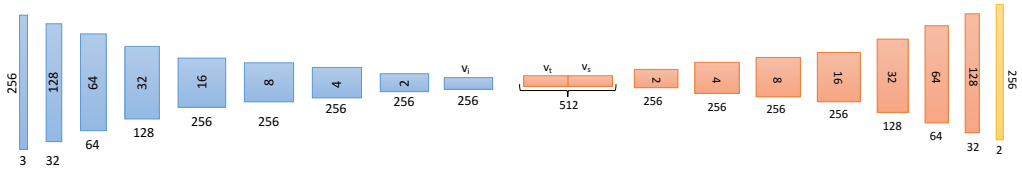


Figure 1: *Encoder-decoder*: In the encoder, each convolution is followed by a leaky ReLU (factor 0.2) and a batch-norm layer (except for the first which has no batch-norm); the convolutional filter sizes are 4×4 and the stride/padding is $2/1$. The face embedding vector has channel size 256. In the decoder, the face embedding vectors corresponding to the source/target frame (v_s, v_t) are concatenated giving a 512-dimensional embedding vector. In all decoder layers, the sequence of executions is: ReLU, bilinear upsampling, batch-norm. The convolutional filter sizes in the sampler decoder and the confidence decoder are 3×3, the stride/padding is $1/1$. The final $2 \times 256 \times 256$ result (or $1 \times 256 \times 256$ for the confidence decoders) is passed through a tanh layer to give FAb-Net's prediction for how to sample from the source frame.

taking random square crops with width in the range $[170, 190]$ for VoxCeleb2 and $[180, 200]$ for VoxCeleb1.

The models are trained using SGD with learning rate 0.001, momentum 0.9 and batch size $N = 8$ (unless the curriculum strategy for FAb-Net is used in which case $N = 32$). The learning rate is divided by a factor of 10 when the loss plateaus (unless the curriculum strategy for FAb-Net is used, in which case this occurs only after the curriculum strategy terminates).

2 Self-supervised baselines

FAb-Net is compared to an autoencoder and to two state-of-the-art self-supervised methods [3, 10]. FAb-Net's architecture but the appropriate loss functions and setups are used. More detail is given below. The baselines are trained with the same training hyperparameters and data augmentation as FAb-Net (please refer to Section 1.2).

Autoencoder. FAb-Net's encoder-decoder architecture are used. The autoencoder is trained to recreate the source frame via a 256-dimensional bottleneck vector, which is used later for evaluation. (The 256 dimensional vector output from the encoder serves as input to the decoder.)

Gidaris et al. [3]. Gidaris et al. apply a rotation $\theta \in \{0°, 90°, 180°, 270°\}$ to an image and train a CNN to predict the rotation that has been applied. To implement this baseline, FAb-Net's encoder (illustrated in Fig. 1) is used. A linear layer (with input channel size 256 and output channel size 4) is appended followed by a softmax layer. The network is trained with a cross-entropy loss. The 256-dimensional embedding is used for evaluation.

Zhang et al. [10]. Zhang et al. split an input image into L and ab channels (e.g. grey and colour channels) and then learn to reconstruct the ab channels from the L channel and the L channel from the ab channels. To do this, they effectively split the network into two smaller networks, each with half the capacity of the original network. To implement this baseline,

the exact same encoder-decoder structure as FAb-Net is used, except that it is divided into two subnetworks, each with half the capacity of the original network at each layer (e.g. 16 channels in the first layer and 128 channels in the embedding for each sub-network). The MSE loss is used as the reconstruction loss. The concatenation of the two 128-dimensional embeddings gives a 256-dimensional embedding, which is used for evaluation.

3 Datasets

For our models and baselines the input image size is $256/times256$ except for the VGG-Face descriptor which requires input images of size $224/times224$.

300-W and MAFL. FAb-Net is evaluated on 300-W and MAFL using the procedure outlined in [8]. In order to make the images more similar to those of VoxCeleb+, the images are re-cropped to make a tighter crop around the face region (this is a fair comparison as [8, 9, 11] re-crop to make the images more similar to those in CelebA [6] which they use as their training set).

AFLW. The images in AFLW [5] are resized to 256×256 for our models and to 224×224 for the evaluation of the VGG-Face descriptor.

EmotioNet. For EmotioNet [1], the classifier is trained on the subset of the dataset that is automatically annotated. We divide this subset into a training and a validation set (with a 80/20 split). The faces are detected using dlib [4] and cropped to 256×256. This results in $743,033$ images for training and $185,759$ images for validation. The independent set of about 25k images is used as test set. After detecting faces, this gives $25,517$ images for testing. This evaluation is done for the 11 AUs used in track 1 of the EmotioNet challenge 2017 [2].

AffectNet. For our experiments on AffectNet [7], we use the manually annotated subset of the dataset. As the AffectNet test set is not released, we use the released validation set to test on and randomly divide the training set into a training and a validation subset (with a 85/15 split). The faces are detected using dlib [4] and cropped to 256×256. Furthermore, images annotated as 'non-face', 'none' or 'uncertain' are discarded. This results in $287,055$ images for training, $57,411$ for validation and 3989 images for testing. The test set is balanced across the different emotion categories whereas the training data is not.

4 Additional qualitative results.

Additional examples of learned confidence heatmaps are visualised in Fig. 2 and additional examples for retrieval are visualised in Fig. 3.

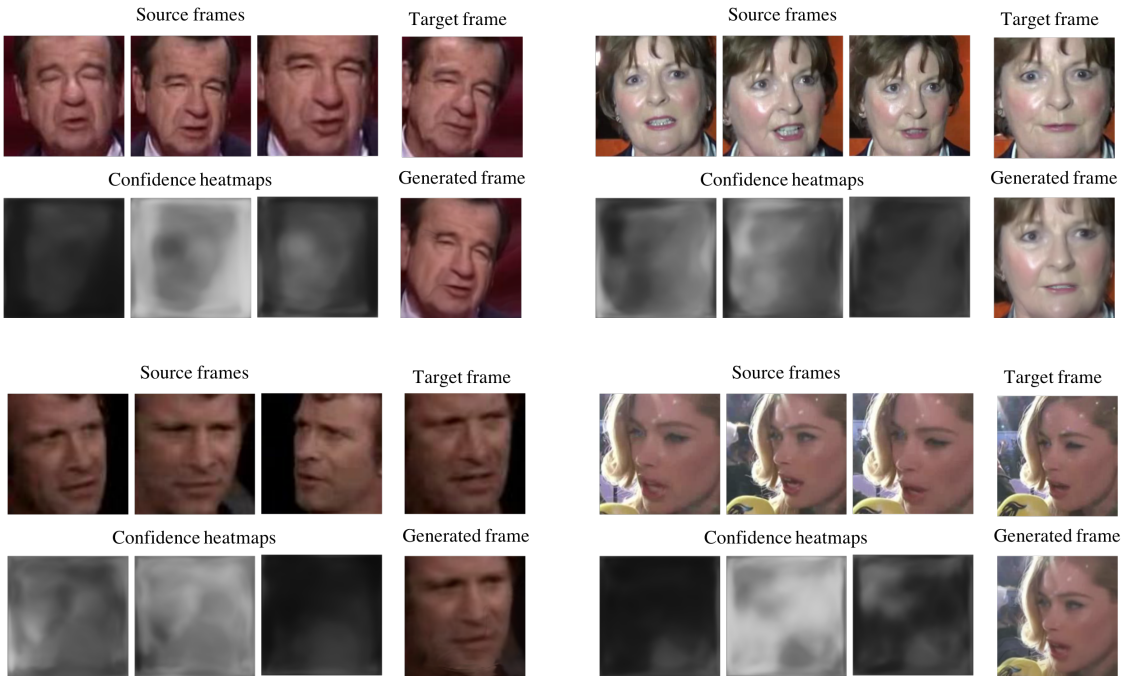


Figure 2: Additional examples of confidence heatmaps predicted by FAb-Net for the given source and target frames. These examples demonstrate how the confidence maps allow the network to focus on certain source frames or parts of different source frames. In the top left example, the network chooses one eye from the rightmost source frame even though its pose is quite different as compared to that of the target frame. In the bottom two examples, the source frames with pose or zoom dissimilar to that of the target frame are discarded.

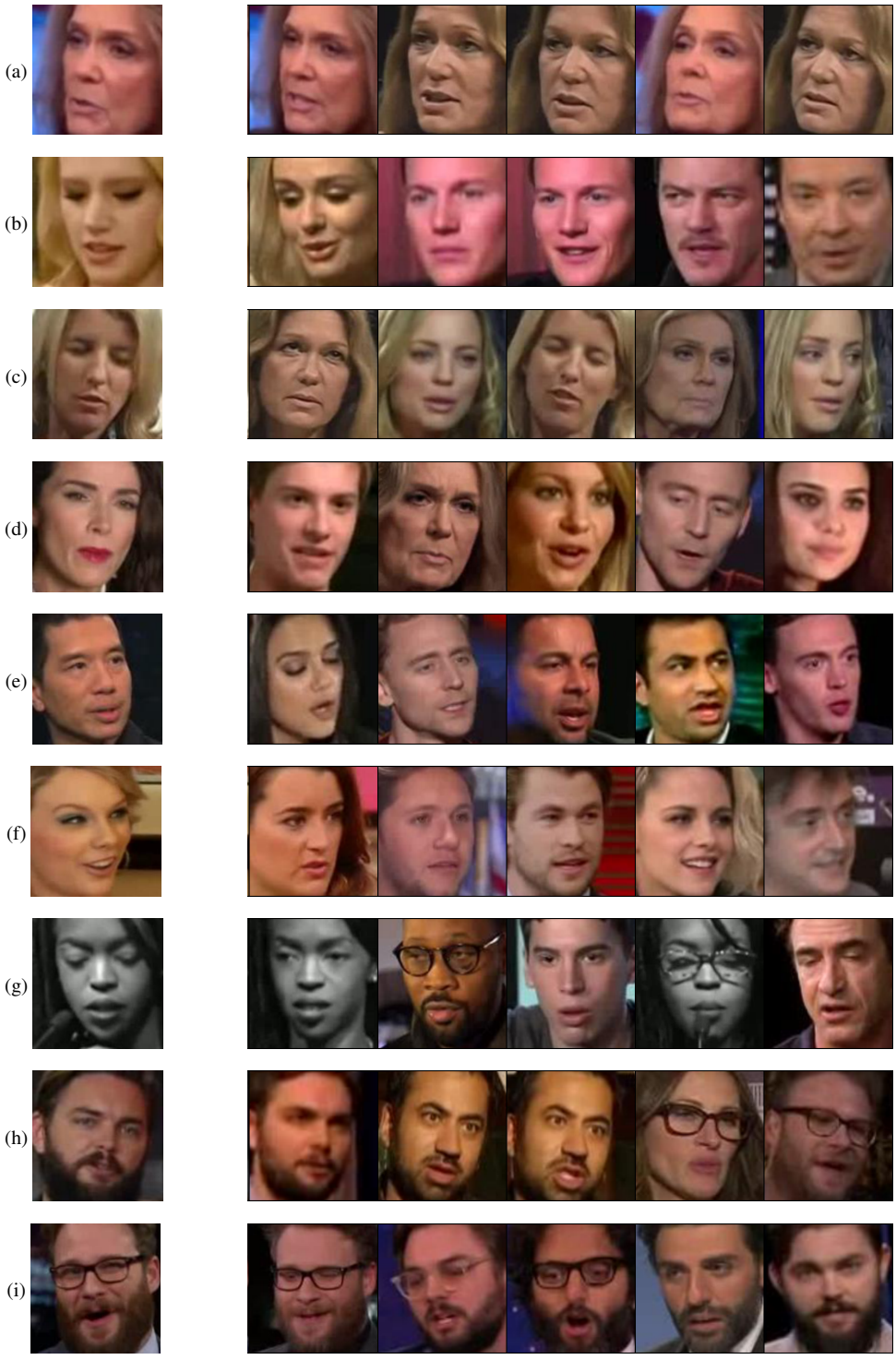


Figure 3: Additional results for nearest neighbour retrieval using the embedding learned by FAb-Net. The embedding captures facial attributes, e.g. pose and expression, as it succeeds in retrieving images with similar facial attributes given a query image.

References

[1] C. F. Benitez-Quiroz, R. Srinivasan, and A. M. Martinez. EmotioNet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In *Proc. CVPR*, 2016.

[2] C. F. Benitez-Quiroz, R. Srinivasan, Q. Feng, Y. Wang, and A. M. Martinez. Emotionet challenge: Recognition of facial expressions of emotion in the wild. *arXiv preprint arXiv:1703.01210*, 2017.

[3] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *Proc. ICLR*, 2018.

[4] D. E. King. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758, 2009.

[5] 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 *Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[6] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proc. ICCV*, 2015.

[7] A. Mollahosseini, B. Hasani, and M. H. Mahoor. AffectNet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 2017.

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

[9] J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *NeurIPS*, 2017.

[10] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proc. CVPR*, 2017.

[11] 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.

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

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	Self-supervised learning of a facial attributes embedding from video
Publication Status	Published
Publication Details	Olivia Wiles, Almut Sophia Koepke, and Andrew Zisserman: *Self-supervised learning of a facial attributes embedding from video*. British Machine Vision Conference (BMVC), 2018.

Student Confirmation

Student Name:	ALMUT SOPHIA KOEPKE		
Contributions to the Paper	Developing the ideas for the network architecture, self-supervised pretext task, and for learning a useful face embedding. Preparing the training dataset and the datasets for evaluating head pose and facial expression estimation. Developing and implementing the frameworks including those for evaluating the face embeddings and training the linear classifiers/regressions. Running experiments and writing the paper.		
Signature	A. T. Koepke	Date	19/12/19

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 ANDREW ZISSERMAN			
Supervisor comments This was a beneficial, joint, and equal collaboration.			
Signature	A. Zisserman	Date	19/12/19

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

Chapter 5

Visual pitch estimation for violin playing

This work was presented as a *poster* at the 16th Sound & Music Computing Conference (SMC), 2019.

In this paper, we present a student-teacher framework that allows to estimate pitch for videos of violin playing from visual information alone.

Visual Pitch Estimation

A. Sophia Koepke
University of Oxford
koepke@robots.ox.ac.uk

Olivia Wiles
University of Oxford
ow@robots.ox.ac.uk

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

ABSTRACT

In this work, we propose the task of automatically estimating pitch (fundamental frequency) from video frames of violin playing using *vision* alone. Here, we consider only monophonic violin playing (where only one note is being played at a time).

In order to investigate this task, we curate a new dataset of monophonic violin playing. We propose a Convolutional Neural Network (CNN) architecture that is trained using a student-teacher strategy to distil knowledge from the audio domain to the visual domain. At test time, our network takes video frames as input and directly regresses the pitch. We train and test this architecture on different subsets of our new dataset.

We show that this task (i.e. pitch prediction from vision) is actually possible. Furthermore, we verify that the network has indeed learnt to focus on salient parts of the image, e.g. the left hand of the violin player is used as a visual cue to estimate pitch.

1. INTRODUCTION

Humans can obtain some understanding of music simply by watching instruments being played, even without access to audio recordings of the music itself. Indeed, a trained musician might be able to transcribe an entire video purely from visual cues alone, although with great painstaking manual effort. The movement and position of the instrument and body (specifically the movement of the arms, hands and fingers) have a direct correlation with the sound produced. In this work, we investigate the following question: is it possible for a trained neural network to identify the pitch of played notes, simply from the frames of a silent video?

Our approach is a valuable first step towards the task of *complete* visual music transcription. While *audio* based music transcription is a widely studied and successful field, the task of *visual* music transcription has not been explored to a great extent. Performing this task from standard frame-rate visual information alone can be extremely useful in instances when the audio is of poor quality, missing, or mixed with information from other audio sources, e.g. in the case of polyphonic music. These scenarios are challenging for purely audio-based pitch estimation methods.

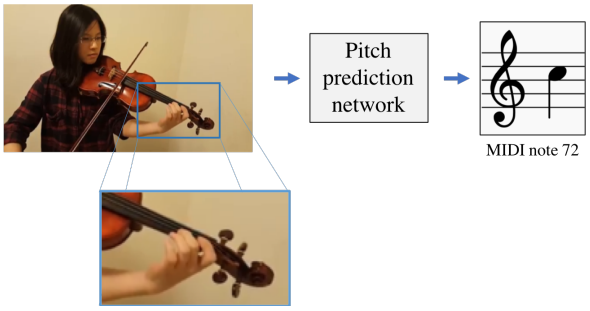


Figure 1. Pitch estimation from visual information. Given video frames, the network is tasked to predict pitch using *only* the visual information.

We investigate this by training a network to predict pitch information from video frames of monophonic solo violin recordings using only the visual image data (see Figure 1). Given a set of video frames, the network learns to regress the corresponding pitch. In order to perform this challenging task, our method makes use of two insights. First, using a teacher-student strategy (i.e. training one network using another network [1]) is important to enforce that the network learns the visual cues that are correlated with the corresponding sound. Intermediate audio features are crucial in providing an additional training signal for the visual pitch estimation network. Second, using multiple frames as input (i.e. a short silent video clip) is preferable to using a single still frame. This is because the additional frames resolve ambiguities such as which string is vibrating (i.e. the string that is being played on with the bow). These insights inform our architecture choices, described in Section 3.

The models are trained and evaluated on a new dataset (Section 4) of violin playing. This dataset is divided into three subsets which vary in difficulty. The first two subsets are recordings of a single player photographed by a fixed mobile phone camera. The third subset consists of 'in-the-wild' videos downloaded from YouTube.

On all of these datasets, our method demonstrates that regressing pitch directly from video frames is indeed possible (Section 5). Finally, we verify that the method is making sensible predictions by investigating what regions of the image are most salient for the prediction. We find that our method focusses on the movement and location of the musician's arms, hands and fingers; this is similar to how

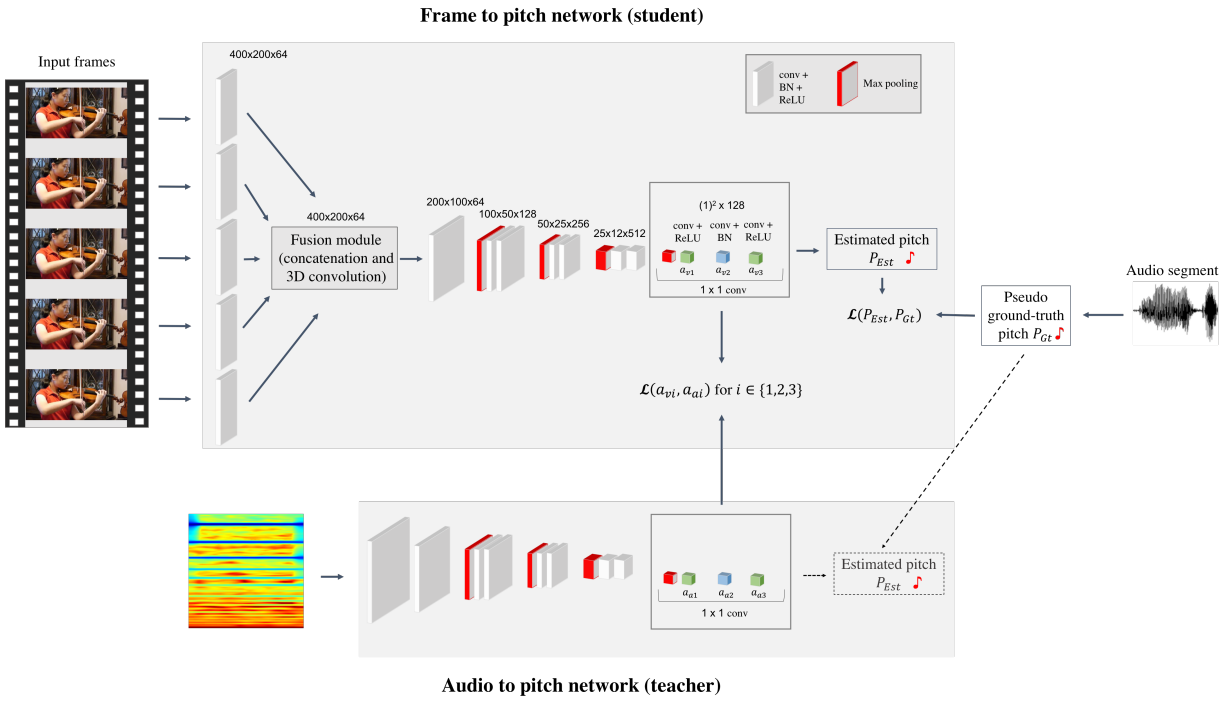


Figure 2. An overview of the visual pitch estimation method. We train our framework with a student-teacher strategy by distilling discriminative knowledge from a teacher network to a student network. The teacher network regresses pitch from audio whereas the student network is trained to regress pitch from visual information alone. Both networks are trained using pseudo ground-truth pitch information which is automatically extracted using an audio-based method and thereby does not require any manual annotation. The audio to pitch network is trained first and then used to train the frame to pitch network (student) by minimising the distance between the activations of the final three fully-connected network layers of the student and the teacher network. At test time, given one or multiple visual input frames, the student network is used by itself to regress pitch. In the case of multiple input frames, the outputs of the first convolutional layers of the student network are concatenated and fused through a 3D convolutional layer (Fusion module) before being input to the next convolutional layer.

a human would approach this task.

2. RELATED WORK

Here, we only consider directly related work on cross-modal information transfer between audio and visual information.

Multi-modal audio-visual representations. Training strategies that encourage synchronisation between the audio and visual streams have been used successfully for speech synchronisation [2]. More generally the correspondence between the audio and visual streams (though not their strict synchronisation) has proven very useful for obtaining meaningful features for sound localisation and separation [3, 4]. In these works, the natural synchronisation in videos can be leveraged in a self-supervised manner to obtain useful image and audio representations. Aytar et al. [5] also exploit this natural synchronisation property in order to transfer knowledge from visual recognition networks into sound networks. However, we propose a framework that transfers knowledge the other way round, i.e. from audio to visual information.

Cross-modal audio-visual generation. Related to our framework are methods that generate audio from visual information, e.g. spectrograms or other sound features from visual information [6–10], or localise sound in video in order to separate different sounds [11, 12], or analyse vibrato-patterns for audio-visual association [13].

The URMP dataset by Li et al. [14] is targeted at cross-modal audio-visual generation. However, it poses two limitations for our task. Firstly, the image resolution of the released dataset is not very high which makes it difficult to actually recognise pitch from the visual information alone, as the key parts of the image (e.g. the fingers of the left hand) are too small. Secondly, the dataset was recorded in constrained settings with only a limited number of musicians which limits the generalisability of models trained on this data to other settings. Therefore, in addition to training and evaluating our models on the URMP dataset, we gathered a new dataset to train and test our framework that is of higher resolution and which also contains 'in-the-wild' videos.

Music transcription from silent video. More closely related to ours is the work by Gomez et al. [15] which proposes to leverage visual information to transcribe clarinet videos using the hand movements in recorded video sequences. However, unlike their method, we do not require any manual tracking or labelling (i.e. finger/hole positions)

as supervision in order to train our network.

Zhang et al. [16] addressed a similar task to ours of visually obtaining pitch for violin by detecting the strings of a violin and by recognising finger events (such as their position and whether they are pressing on a string). However, their method is quite constrained; it involves tracking the fingers and the strings, and makes assumptions about the length of the fingerboard which requires the image data to always be perfectly aligned. In contrast, our method gives convincing results for different viewpoints and requires no manual labelling.

Another related method is the physics-based approach for recovering pitch from silent guitar video by Goldstein and Moses [17]. However, their method requires mounting a camera, that allows recording with high frame rates, on the guitar itself in order to use the actual string vibrations to predict pitch. Unlike their method, our set-up only requires the use of a normal camera and it learns to localise the left-hand position of the musician (relative to the instrument) in order to infer pitch. Our method can thus be applied retroactively to videos that have already been recorded.

3. MODEL

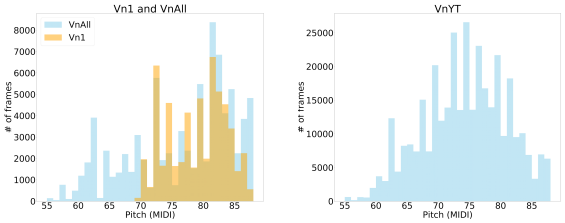
In this section, we describe the training and testing framework used to regress pitch from video frames. We treat this as a classification task. The network takes video frames as input and estimates the pitch as a MIDI number. An overview is given in Figure 2.

Teacher-student strategy. We found that directly regressing pitch from the video frames did not generalise at test time. This is presumably because the visual information relevant for the pitch prediction task occupies only a small part in the video frames.

As a result, we train two networks – a teacher and student network – such that the activations of the student network are similar to those of the teacher. The teacher network regresses pitch from audio and the student regresses pitch from video frames. The rationale for using this strategy is that, in order to contain relevant information about pitch, the high-level representation of the visual information (encoded in the student) should be close to that of the audio information (encoded in the teacher). This strategy proved crucial to obtain a network that generalises at test time.

The teacher network is first trained using STFT spectrograms as input to regress the pseudo ground-truth pitch (the method for obtain this pseudo ground-truth is described in section 5). The student network is then trained to regress the pitch with an additional loss that enforces that the activations of the higher level layers are similar to those in the teacher network. For this, we use an L1 loss which is weighed so that the contribution for each of the three fully-connected layers is as big as the pitch classification loss. Both networks are trained to predict pitch with a cross-entropy loss.

Neural Network Architecture. The teacher and student network architectures are loosely based on the VGG-M network architecture [18] and can be seen in more detail in Figure 2. For the student network, in the case of multiple input frames, the outputs of the first convolutional layers are concatenated and fused using a 3D convolutional layer to combine the information from the frames with spatial kernel size 3×3 followed by batch normalization and ReLU. The output serves as input to the second convolutional layer. For just a single input frame, the output of the first (2D) convolutional layer is directly input to the second convolutional layer. The first two convolutional layers consist of 7×7 convolutions, whereas the subsequent ones are 3×3 convolutions and the last three are 1×1 convolutions. All convolutional layers have stride 1 except for the second one, which has stride 2.



(a) *Vn1* and *VnAll* datasets. (b) *VnYT* dataset.

Figure 3. Pitch distribution over the number of frames in the three subsets of our dataset; the constrained single-string data *Vn1*, the data on all strings *VnAll*, and the in-the-wild data *VnYT*. Pitch is shown in MIDI numbers. The subsets cover the chosen full pitch ranges.

4. DATASETS

We curate a new violin playing dataset which consists of three subsets (*Vn1*, *VnAll* and *VnYT*) that differ in difficulty and size. The most challenging subset *VnYT* consists of in-the-wild violin solo videos downloaded from YouTube [1] . These are largely comprised of recordings of solo recitals, etudes, and orchestra auditions. Both *Vn1* and *VnAll* are recorded in simpler conditions: all videos are of a single violinist, have similar backgrounds and are taken from similar angles.

The datasets vary in terms of the range of pitches. Both *Vn1* and *VnAll* consist of recordings of a violinist playing in a practise-like set-up (but without the thousandfold repetitions of the same phrases). The easiest subset *Vn1* consists of videos that only contain violin playing on a single string, resulting in a range of 20 semitones (MIDI numbers between 68 and 88). Estimating the pitch is easier in this case, as there is no ambiguity concerning which string is being played. *VnAll* contains videos played in the full pitch range of the violin without being restricted to playing just on one string. For both *VnAll* and the most difficult subset *VnYT*, we consider a range of 33 semitones (MIDI numbers between 55 and 88).

All subsets are split into train/val/test sets. Disjoint parts of the same videos are used for training and validation. The test sets consist of frames that were not seen during train-

[1] Example videos: https://youtu.be/-ccYdhQAn10, https://youtu.be/YGCYelAHdaU

Dataset *Vn1*			
	Train	Val	Test
# of videos	5		1
# of frames	25308	2791	9875
Dataset *VnAll*			
	Train	Val	Test
# of videos	9		1
# of frames	54865	4107	6373
Dataset *VnYT*			
	Train	Val	Test
# of videos	122		10
# of frames	303391	33063	20067

Table 1. Dataset statistics. Details of the three different subsets, the controlled setting data on a single string *Vn1*, the controlled setting data on all strings *VnAll*, and the in-the-wild data on all strings *VnYT* and their respective train/val/test splits.

ing (from left-out unseen videos). The precise numbers of frames and videos are given in Table 1.

Finally, for all three subsets, we extract frames and pseudo ground-truth pitch using the spectral domain YIN algorithm [19] using the implementation in the *aubio* library (yinfft) [20].

In addition to the above datasets, we consider the subset of the URMP dataset that contains videos of violin playing. We loosely crop the frames around the violinist and leave out 4 videos (single-instrument tracks) for testing and take the remaining violin videos for training and validation. This results in about 35000 frames for training and 12761 for testing. This dataset contains ground-truth pitch information. Therefore, we can train with actual ground-truth pitch. We consider a range of 33 semitones (MIDI numbers between 55 and 88). All models are trained and tested with the same train/val/test split on each dataset.

Furthermore, we generate STFT spectrograms for all mentioned datasets in order to train the audio to pitch network.

5. EXPERIMENTS

In this section, we evaluate both the audio to pitch (teacher), and the video frame to pitch (student) models. We consider using a single versus multiple input video frames. We first train the audio to pitch network to regress pitch from spectrograms. This network then serves as the teacher network when training the single frame to pitch network or the multi-frame to pitch network.

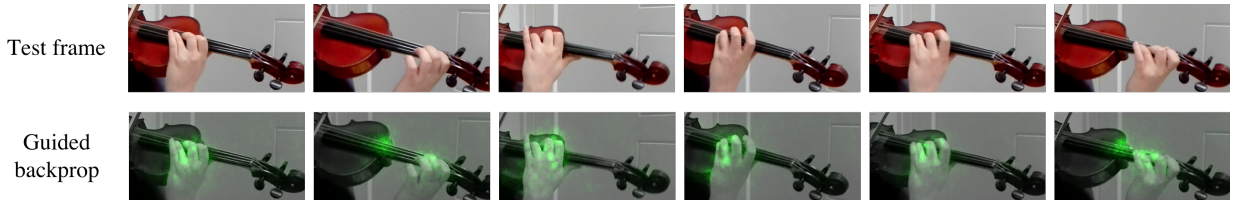
The models are trained in PyTorch [21] using the Adam optimiser [22] with an initial learning rate of 0.001. The learning rate is divided by a factor of 10 when the loss on the validation set plateaus. The batchsize is $N = 64$ for the single frame architecture and the audio to pitch network, and $N = 24$ for the architecture with 5 input frames. The frames are resized to 400×200. For the datasets *Vn1* and *VnAll*, the frames are consistently more tightly cropped around the instrument whereas there is much more variation of the location and relative size of the instrument in *VnYT*.

Network	RPA	RPA tol	PA	ACA	ACE
	Dataset *Vn1*				
Audio to pitch	98.30	99.14	96.74	86.03	0.06
Frame to pitch	89.98	91.57	62.41	51.64	0.45
5 fr. to pitch (3D conv)	93.8	94.91	66.7	58.75	0.43
	Dataset *VnAll*				
Audio to pitch	94.26	94.40	90.87	94.33	0.06
Frame to pitch	74.17	75.55	47.48	33.3	2.50
5 fr. to pitch (3D conv)	77.24	78.98	50.33	41.66	1.65
	Dataset *VnYT*				
Audio to pitch	98.30	99.14	96.74	86.03	0.06
Frame to pitch	44.3	51.37	33.18	45.2	2.50
5 fr. to pitch (3D conv)	62.5	67.89	48.44	51.77	2.34
	Dataset URMP				
Audio to pitch	98.28	98.5	96.73	98.88	0.07
Frame to pitch	53.11	58.3	42.71	39.86	2.73
5 fr. to pitch (3D conv)	57.3	62.04	45.26	41.79	2.43

Table 2. Evaluation of our models determining the accuracy in predicted pitch for the *Vn1*, *VnAll*, *VnYT* , and URMP test sets. Higher is better for Raw Pitch Accuracy (RPA), Raw Pitch Accuracy with a tolerance of one frame (RPA tol), Pitch Accuracy (PA), and Average Class Accuracy (ACA). Lower is better for Average Class Error (ACE). Using multiple input frames improves the performance.

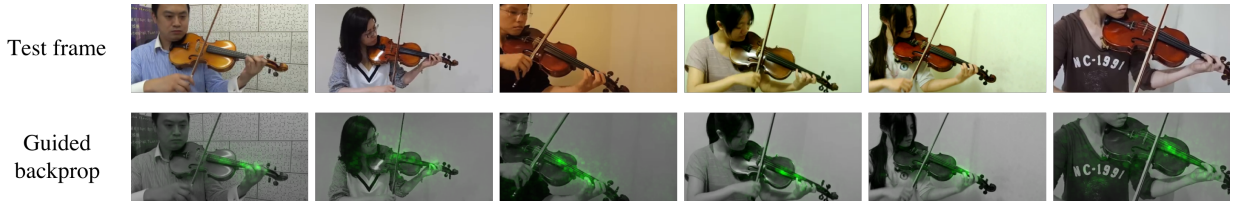
Evaluation measures. We report the performance of our models in Table 2. For Raw Pitch Accuracy (*RPA*), a predicted pitch is counted as correctly estimated if it lies within one semitone of the ground truth pitch. *RPA tol* additionally allows the prediction to be off by at most one frame. Furthermore, we report Pitch Accuracy (*PA*) and Average Class Accuracy (*ACA*). ACA gives the averaged per-pitch-class accuracy. The ACE describes the average error between the predicted and the ground truth pitch class (ACE of 1 corresponds to an average error of one semitone).

Video to pitch performance. The audio to pitch teacher networks reach an RPA of above 90% on the test sets. This serves as a very good starting point to train the student frame to pitch networks. We note that direct training of the visual pitch estimation network without distilling information from the pitch teacher network did not converge. It can be observed that our method performs best when trained and tested on the simpler datasets with minimal ambiguities *Vn1* and *VnAll*. This corresponds with the intuition that this set-up is easier, as the fingers and therefore the pitch is more clearly visible at higher resolution as compared to *VnYT* or URMP. Nevertheless, a RPA of 62.5% for the frame to pitch network on the in-the-wild YouTube video dataset *VnYT* means that the pitch is estimated within a semitone of the ground-truth on average in 62.5% of the test cases; this verifies that our method generalises to unseen videos and people at test time on challenging 'in-the-wild' videos. When allowing for an offset of one frame in the predictions, we achieve an accuracy of 67.89% (RPA tol). This accounts for the case that the alignment between audio and visual information might not be perfect in the data which is the case for some of the downloaded videos. The reported lower performance on the URMP dataset may be due to the lower resolution size of the frames in the dataset and the limitations in terms of



(a) Gradient visualisation for the *Vn1* test set.

(b) Gradient visualisation for the *VnAll* test set.



(c) Gradient visualisation for the *VnYT* test set.

Figure 4. Discriminative information visualisations using guided backpropagation [23] for the test sets of the *Vn1* subset in (a), the *VnAll* subset in (b), and the *VnYT* subset in (c). Heat maps are overlaid in the second rows of (a), (b), and (c). As demonstrated, the networks focus on the left hand across all the test frames even though the hands are in different positions relative to the frame. In (c), it can be seen that the network also seems to be focussing on the strings implying that it may be using vibrations or the movement of the strings in order to estimate pitch. The location of the instrument and strings relative to the left hand might serve as a further cue for estimating pitch.

its dataset size which confirms the benefits of using our datasets to address this task. These results are impressive, given that our method estimates the pitch from visual information only and in unconstrained recording conditions. However, our method can only predict one pitch playing at a time and cannot identify chords as it has been trained only on monophonic data.

Another interesting point is that there is a consistent improvement when using multiple frames as opposed to a single one as input to the frame to pitch network. This can be seen very clearly for the *VnYT* dataset (RPA tol of 67.89% vs. 51.37%). This is presumably due to the fact that visually it can be hard to determine just from the fingers of the left hand which string a note is played on. To solve this problem, the network needs to determine which string is active (using for example information from the bowing hand / bow or from the vibration of the strings). While the placement of the hand should be visible from a single image, the vibration of the string is unlikely to be visible (unless there is significant motion blur) without taking into account more frames.

Visualizing what has been learnt. To gain an insight into what the networks have learnt and how they infer the pitch from a given frame, we apply guided backpropagation [23] to our trained networks to determine which parts of the images are most discriminative. As demonstrated in Figure 4, the networks have learnt that the fingers of the left hand and the left hand itself are most relevant for predicting the pitch given a still frame. Potentially the network also makes use of some information about the vibration of the played strings (e.g. by recognising motion blur around strings that are vibrating). This confirms that the networks do not simply memorise parts of a video, but instead learn to localise the left hands/fingers in the image in order to estimate pitch. However, the image regions which the networks focus on are actually quite small relative to the image size.

6. CONCLUSION

We have presented a method for addressing monophonic visual pitch estimation; given video frames of violin playing, our method can automatically estimate the pitch being played using *vision* alone. The presented task is extremely challenging, as it requires making use of subtle visual cues (such as the placement of the hand or string vibrations over the course of multiple frames), yet our network shows convincing results in three different scenarios: when only one string is played or all strings are played but the person and environment remains the same, and in unconstrained 'in-the-wild' videos. Moreover, our method is generalisable, as training the networks did not require any manual annotations; instead, the pseudo ground-truth pitch information

was extracted automatically from the audio data. It will be interesting to use this framework to improve pitch prediction using both visual and audio information. This could prove useful when the audio is of poor quality. In addition to that, estimating pitch from vision might help the task of sound source separation when similar instruments are played on. Furthermore, this method could be pushed further to estimating polyphonic violin music played on the same instrument.

Acknowledgments

This work is supported by the EPSRC programme grant Seebibyte EP/M013774/1: Visual Search for the Era of Big Data. We are very grateful to Yael Moses for insightful discussions. We thank Arsha Nagrani for feedback.

7. REFERENCES

[1] C. Bucilua, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *2th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006.

[2] J. Chung and A. Zisserman, "Out of time: automated lip sync in the wild," in *Workshop on Multi-view Lip-reading, ACCV*, 2016.

[3] R. Arandjelović and A. Zisserman, "Objects that sound," in *Proc. ECCV*, 2018.

[4] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba, "The sound of pixels," in *Proc. ECCV*, 2018.

[5] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *NeurIPS*, 2016.

[6] L. Chen, S. Srivastava, Z. Duan, and C. Xu, "Deep cross-modal audio-visual generation," in *Proceedings of the on Thematic Workshops of ACM Multimedia*, 2017.

[7] A. Davis, M. Rubinstein, N. Wadhwa, G. Mysore, F. Durand, and W. T. Freeman, "The visual microphone: Passive recovery of sound from video," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 2014.

[8] W.-L. Hao, Z. Zhang, and H. Guan, "Cmcgan: A uniform framework for cross-modal visual-audio mutual generation," *CoRR*, 2018.

[9] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman, "Visually indicated sounds," in *Proc. CVPR*, 2016.

[10] A. Ephrat, T. Halperin, and S. Peleg, "Improved speech reconstruction from silent video," in *ICCV workshop*, 2017.

[11] T. Afouras, J. S. Chung, and A. Zisserman, "The conversation: Deep audio-visual speech enhancement," in *INTERSPEECH*, 2018.

[12] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation," *Proc. ACM SIGGRAPH*, 2018.

[13] B. Li, C. Xu, and Z. Duan, "Audiovisual source association for string ensembles through multi-modal vibrato analysis," *Proc. Sound and Music Computing (SMC)*, 2017.

[14] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications," *IEEE Transactions on Multimedia*, 2019.

[15] E. Gómez Gutiérrez, P. Arias Martínez, P. Zinemanas, and G. Haro Ortega, "Visual music transcription of clarinet video recordings trained with audio-based labelled data," in *ICCV workshop*, 2017.

[16] B. Zhang, J. Zhu, Y. Wang, and W. K. Leow, "Visual analysis of fingering for pedagogical violin transcription," in *Proceedings of the 15th ACM international conference on Multimedia*, 2007.

[17] S. Goldstein and Y. Moses, "Guitar music transcription from silent video," in *Proc. BMVC.*, 2018.

[18] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. BMVC.*, 2014.

[19] P. Brossier, "Automatic annotation of musical audio for interactive systems," Ph.D. dissertation, Ph. D. thesis, Queen Mary University of London, London, UK, 2006.

[20] P. Brossier, M. Hermant, E. Müller, N. Philippsen, T. Seaver, H. Fritz, and S. Alexander, "aubio/aubio: 0.4.6," Oct 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1002162

[21] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," 2017.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization." *Proc. ICLR*, 2015.

[23] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *ICLR workshop*, 2015.

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	Visual Pitch Estimation
Publication Status	Published
Publication Details	Almut Sophia Koepke, Olivia Wiles, and Andrew Zisserman: *Visual Pitch Estimation*. Sound and Music Computing Conference (SMC), 2019.

Student Confirmation

Student Name:	ALMUT SOPHIA KOEPKE		
Contributions to the Paper	Developing the idea for the task and network architecture. Curating the violin video datasets. Running all experiments and writing the paper.		
Signature	A. S. Koepke	Date	19/12/19

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 ANDREW ZISSERMAN		
Supervisor comments Sophia originated the idea and was the driving force for this paper with Olivia providing some support.		
Signature	Date	19/12/19

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

Chapter 6

Sight to sound: An end-to-end approach for visual piano transcription

This work was accepted for *oral presentation* at the 45th International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2020.

In this work, we present a framework that estimates MIDI onsets for videos of piano playing from visual information only.

SIGHT TO SOUND: AN END-TO-END APPROACH FOR VISUAL PIANO TRANSCRIPTION

A. Sophia Koepke[†], *Olivia Wiles*[†], *Yael Moses*[‡], *Andrew Zisserman*[†]

[†]VGG, Department of Engineering Science, University of Oxford
[‡]The Interdisciplinary Center, Herzliya

ABSTRACT

Automatic music transcription has primarily focused on transcribing audio to a symbolic music representation (e.g. MIDI or sheet music). However, audio-only approaches often struggle with polyphonic instruments and background noise. In contrast, visual information (e.g. a video of an instrument being played) does not have such ambiguities. In this work, we address the problem of transcribing piano music from visual data alone. We propose an end-to-end deep learning framework that learns to automatically predict note onset events given a video of a person playing the piano. From this, we are able to transcribe the played music in the form of MIDI data. We find that our approach is surprisingly effective in a variety of complex situations, particularly those in which music transcription from audio alone is impossible. We also show that combining audio and video data can improve the transcription obtained from each modality alone.

***Index Terms*—** visual music transcription, automatic music transcription, music information retrieval, deep learning

1. INTRODUCTION

Automatic music transcription (AMT) describes the process of automatically transcribing raw data – typically audio information – into a symbolic music representation (e.g. music notation or MIDI data). Such technology can be used to transcribe music when improvising or deliberately composing, making it easily reproducible. However, AMT from audio alone is challenging in multiple situations, such as in the presence of multiple notes or instruments or when there is background noise. While digital instruments automatically transcribe music using sensors rather than audio (e.g. a digital piano uses keypress sensors to write MIDI data), acoustic instruments are typically not equipped with such sensors.

In this paper, we propose an end-to-end deep learning approach that uses only visual information for transcribing piano music while ignoring audio cues, i.e. visual music transcription (VMT). We obtain pseudo ground-truth data to train our framework using an audio-based method.

Using visual cues alone for music transcription is possible because simply watching a pianist play reveals information about the notes being produced. For example, the positioning of the hand and keys reveals information about the keys being pressed. Furthermore, the motion between frames provides localisation information about the onset of notes. Thus, it is reasonable to expect that musical audio information can be extracted from purely visual data. Using video removes the ambiguities that arise from relying on audio alone when multiple notes sound simultaneously. However, this is a challenging task since the fingers may move without pressing a key and keypresses can be occluded by the hands.

Given a video of a pianist playing, we automatically predict the pitch onset events (i.e. which and when keys are being pressed) in each video frame. We can then stitch onset events together to extract a music transcription for an entire video.

This enables a number of possible applications. An obvious use case is to transcribe silent piano videos. Also, in a similar manner to lip reading in the speech domain improving speech recognition, note onset estimation from visual information can improve audio music transcription compared to using audio alone.

2. RELATED WORK

Music transcription from visual information. Most previous methods for transcribing piano music from visual data alone are designed for constrained settings; they rely on detecting pressed keys and do not make use of temporal information in terms of hand and finger motion. [1, 2] use RGB images and require difference images between the background and current video frame to detect hands and piano keys. This is difficult to obtain when the illumination changes across the video or when shadows appear. [2] add an illumination correction step in their pipeline, but the authors report limitations for drastic light changes or vibrations of the camera or piano. [3] adds depth information, which enables velocity prediction. [4] also use depth cameras to identify key presses for a piano tutoring system. [5] predicts per-frame key presses; however, their set-up is quite constrained and can only predict a single key press per frame. Our method on the other hand, does not require depth information or background images and can deal with illumination, shadow changes, and vibration of the camera or piano. A few works have tackled VMT for other instruments (e.g., guitar [6], violin [7, 8], and clarinet [9, 10]). [11, 12] fuse visual and audio information

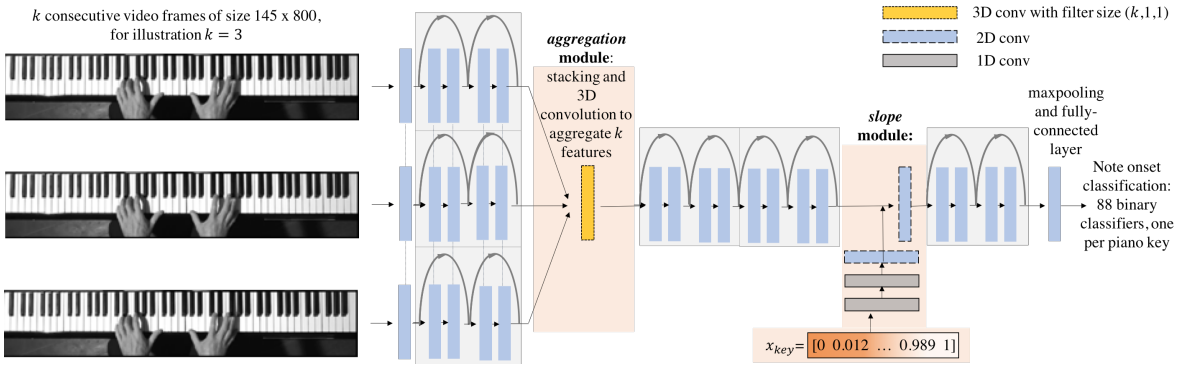


Fig. 1: An overview of our network architecture. Note onset prediction from k video frames of piano playing. Our models use $k = 5$. The network architecture is based on the ResNet18 architecture. The activations from k consecutive input frames are aggregated using a 3D convolution (*aggregation* module). x_{key} is a vector that is passed into the *slope module* which encourages the network to preserve spatial information at later stages.

together for guitar and piano transcription respectively.

Our work is most similar to [13] and [12], both of which use learning-based approaches to VMT. [13] presents a multi-step pipeline that requires significant preprocessing: given a processed crop of a single key, their Convolutional Neural Networks (CNNs) predict whether it has been pressed. [13] relies on key presses that are clearly visible from video frame differences. This is not the case when there is video jitter, instrument vibrations, low-resolution video data, or video recorded from directly above the keyboard. We cannot compare our method on [13]'s data as their provided MIDI and video information is not aligned. However, we evaluate our data on more challenging and varied music pieces and settings. [12] present a deep learning approach that uses both audio and visual information to detect key presses. They only demonstrate their method on high-quality videos (recorded at 60 fps) and simple pieces (e.g., piano exercises that have at most one note per hand at the same time). We compare our visual only performance to [12]'s on their data in the experiments.

Music transcription from audio information. [14] provides a detailed review of AMT methods, including those for piano. We use [15]'s *Onsets and Frames* framework to obtain pseudo ground-truth to train our networks.

3. NETWORK ARCHITECTURE

In this section we introduce a spatio-temporal model architecture (see Fig. 1) for performing VMT. The model is tasked to predict note onsets (a note onset is the start of a note – for piano playing this coincides with the pressing of a piano key). It uses temporal information by way of the *aggregation* module and maintains spatial information through the *slope* module.

Overview. Our model is based on the ResNet18 architecture [16]. Given 5 consecutive grayscale video frames, the model is tasked to predict all note onsets occurring within ± 1 frame around the middle video frame (i.e. within $\pm \frac{1}{\text{video frame rate}}$ sec). For a video frame rate of 25 fps, 5 frames cover $0.2s$. The 5 input frames are each passed through the first ResNet18 block (with shared weights).

Aggregation module. This module allows the model to make use of temporal information (e.g. the motion of the hand between frames) to determine whether a note is being pressed down (an onset). The output features of size $64 \times 73 \times 400$ ($d \times h \times w$) from the first ResNet18 block are aggregated by stacking them and passing them through a $5 \times 1 \times 1$ 3D convolution, resulting in a channel-wise temporal weighted average of the activations corresponding to the input frames.

Slope module. Classification CNNs are designed to be invariant to spatial positioning. However, in our case spatial localisation is essential, as the location of the hand within the image gives a large amount of information as to the octave and thereby the actual note being played. To allow the network to preserve spatial information, we also pass as an input a slope vector $x_{key} \in [0, 1]^{88}$, which contains 88 linearly spaced valued between 0 and 1 to represent the relative position of a key on the keyboard. This constant slope vector x_{key} is passed through two 1D convolutional layers, with filter size 3 and padding of 1, spatially cloned to expand to size $64 \times 10 \times 50$, such that it matches the output of the third ResNet18 block of size $256 \times 10 \times 50$, before being concatenated to the same. The concatenated activations are then passed through another convolutional layer with filter size (3×3) and padding of 1 resulting in features of size $256 \times 10 \times 50$ before being passed through the rest of the ResNet18 model.

Loss function. The outputs of the final fully-connected layer for our model are 88 probabilities, one for each of the MIDI notes that the piano covers. The models are trained by minimising a binary cross-entropy loss function for each note.

4. DATASETS AND TRAINING

4.1. Datasets

We curated two new datasets of piano playing (PianoYT and MIDI test set) for training our model and to test its generalisation capabilities. The PianoYT and MIDI datasets are available at https://www.robots.ox.ac.uk/~vgg/research/sighttosound/. We also test our models on the Two Hands Hanon test set from [12].

PianoYT: This dataset contains over 20 hours of piano playing videos uploaded to YouTube. All videos are recorded from a top view. We split the data into 209 training/validation videos and 19 test videos. 172 of the training videos and all test videos were recorded by Paul Barton[1]. We obtain pseudo MIDI ground-truth from the audio in the video using the *Onsets and Frames* framework [15].

MIDI test set: In order to evaluate how robust our method is, we also test on 8 recorded videos of an amateur pianist that does not appear in the training set. For this, we recorded data with actual MIDI ground truth using a phone camera. The MIDI was recorded with a digital piano and then aligned with the audio of the recorded video. It consists of a variety of piano pieces (e.g. BWV 778, Schumann Op.15 No.1, Hanon exercises 1 and 5).

Two Hands Hanon test set: The third evaluation dataset is the *Two Hands Hanon test set* in [12] (i.e. Hanon exercises 1 and 5). This dataset contains less challenging pieces with fewer chords and the notes are within a smaller range than those in the MIDI test set.

4.2. Training details

The models are trained on the PianoYT training set using pseudo ground-truth MIDI. Video frames were extracted at their native frame rate. We performed a visual registration procedure resulting in a resized crop of 145×800 pixels such that the keyboard is fully visible and roughly in the same location within the crops.

Because of the relative sparsity of onset events, we reweight training examples in each batch (i.e. class balancing) such that the weight of onset events is equal to that of non-onset events. The models are trained in PyTorch [17] using the Adam optimizer [18] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and batchsize of 24. The initial learning rate is set to 0.001 and training was stopped when the validation loss plateaued. The classification threshold was set to 0.4 using the validation set for all models. For data augmentation, we resize the crops to 150×805 and randomly crop 145×800 pixel regions. In addition to spatial jitter, we jitter the brightness and add Gaussian noise with a factor of 1% of the mean value of the image to 40% of the training images.

[1]https://www.youtube.com/user/PaulBartonPiano

5. EXPERIMENTS

We evaluate our model in multiple settings. First, we demonstrate that we can indeed extract onsets from visual information alone (Section 5.1). We then demonstrate that this is useful in the case of corrupted audio (Section 5.2) and that it can be used to produce MIDI and thereby the audio corresponding to the entire video (Section 5.3).

Metrics: We report precision, recall, accuracy, and F_1 scores for the onset estimation on different test sets. For details about the calculation of these metrics, see [19]. For the PianoYT and the MIDI test set, we report note-level metrics. For the Two Hands Hanon test set, we (similar to the authors of [12]) report frame-level metrics after finetuning to not just predict onsets, but also sustained notes.

5.1. Visual pitch onset estimation

We test how well our model (*ResNet + aggregation + slope*) can extract onsets from visual information alone. We also perform a model ablation study to demonstrate the utility of the aggregation and slope modules by comparing to two baselines: (i) *ResNet* is a ResNet18 model that takes as input 5 frames; the network's first layer is modified to have 5 channels. This model has neither the temporal aggregation nor the slope modules. (ii) *ResNet + aggregation* is a ResNet18 model with temporal aggregation after the first ResNet block but without the slope module.

Results: For the test set of the PianoYT dataset, the estimated MIDI prediction is compared to the pseudo-ground truth. In addition to that, we test our models on videos with actual MIDI ground truth (MIDI test set). Finally, in order to compare to [12], we report results on their Two Hands Hanon test set. Results are given in Table 1.

We see that our custom additions to the ResNet18 backbone architecture (e.g. temporal aggregation and slope module) improve our model's performance. Furthermore, there is a large difference in results when testing on the MIDI test set as opposed to [12]'s Two Hands Hanon test set. The MIDI test set contains more difficult music pieces than [12]'s Two Hands Hanon test set, as more chords are played and a much wider range of notes is covered. In order to bridge the domain gap between our training data (PianoYT dataset) and [12]'s Two Hands Hanon test set (after removing radial distortion of the images), we finetuned our models on their training set. We obtain a frame-level note accuracy (pressed key accuracy) of 87.43%, outperforming their best model that uses audio and visual information. Our results demonstrate the generalisability of our model both to unseen pianists and to more challenging pieces as compared to other methods.

5.2. Audio-visual pitch onset estimation for noisy audio

In Table 2, we demonstrate that using audio and visual information together can be useful especially when the audio is

Model	Prec	Rec	Acc	F_1-score
PianoYT test set				
ResNet	61.40	67.59	50.29	63.72
ResNet+aggregation	**63.86**	67.87	52.20	65.26
ResNet+aggregation+slope	62.23	**73.00**	**53.33**	**66.63**
MIDI test set				
ResNet	65.26	42.82	36.86	49.94
ResNet+aggregation	72.83	52.44	44.97	59.57
ResNet+aggregation+slope	**74.76**	**73.08**	**59.59**	**72.91**
Two Hands Hanon test set from [12]				
ResNet †	92.36	86.25	80.27	88.53
ResNet+aggregation †	92.55	**93.69**	86.96	92.76
ResNet+aggregation+slope †	**93.07**	93.40	**87.43**	**93.00**
[12]‡ 2-stream w/ Multi-Task	-	-	75.37	-

Table 1: Precision, recall, accuracy and F_1-score for pitch onset estimation on the PianoYT test set, the MIDI test set and [12]'s Two Hands Hanon test set for our model (ResNet + aggregation + slope) and two baselines (ResNet, ResNet + aggregation). † fine-tuned on the training set from [12] (after removing radial distortion). ‡ Pressed key accuracy taken from [12] for their best performing model that takes both, audio and visual information as input.

mixed with other sounds or noise.

The performance of the Onsets and Frames framework [15] decreases drastically when the audio is noisy. Mixing the PianoYT test set with other piano audio with a signal-to-noise ratio of 1 results in an F_1 score of 67.52% compared to the pseudo-ground truth obtained for the clean audio. In order to see how we can improve the pitch onset estimation using audio and visual information together, we train a 3-layer perceptron that combines the visual and audio information. It takes the concatenatation of [15]'s 512-dimensional final feature vector from the noisy PianoYT audio with our final feature vector (ResNet+aggregation+slope) as input. As can be seen in Table 2, this results in a significantly improved F_1 score of 81.82%, outperforming the audio-based and our visual based method which achieved an F_1 score of 66.73% on this data. This demonstrates that using our model to leverage visual information is beneficial for obtaining note onsets.

5.3. Producing MIDI

We combine the outputs that our model produces for every 5-frame window to re-create the audio of a full video. For a given video, we pass all outputs from our model through a Gaussian filter ($\sigma = 5$) to add temporal smoothing and threshold the smoothed signal resulting in a binary signal for every note which can be saved as MIDI data. Fig. 2 shows spectrograms for one of the test videos in our MIDI test set computed from the generated and the ground-truth audio (synthesized from MIDI) respectively.

In this example, we notice that the generated audio is able to capture the rough structure of the piece and to correctly predict most of the notes. More example results can be found at https://www.robots.ox.ac.uk/~vgg/research/sighttosound/. There remains room for improvement as the timing of the note onset predictions sometimes is slightly off. Our model was trained to only predict note onset events. However, it tends to predict multiple onset events (e.g. not just at the very beginning of a note) as it is not trained to learn the notion of a note ending.

Model	Prec	Rec	Acc	F_1-score
Noisy audio PianoYT test set, SNR $= 1$				
Audio to note onsets [15]	63.10	80.12	58.93	67.52
Audio-visual MLP (ours + [15])	**87.32**	**78.20**	**71.97**	**81.82**

Table 2: Precision, recall, accuracy and F_1-score for pitch onset estimation for the noisy PianoYT test set where the clean audio is mixed with other piano audio. The performance of the audio to onset estimation suffers with added noise. Audio and visual features are ingested by the MLP which combines visual features from our model with audio results from [15], and results in a significant improvement.

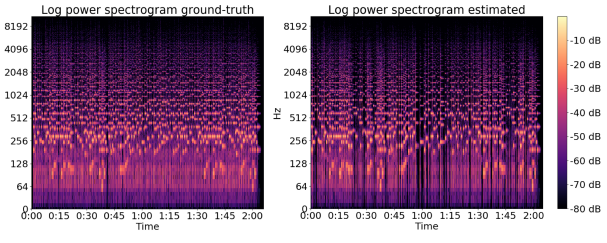


Fig. 2: Spectrogram comparison for generated audio from our MIDI prediction (right) and ground truth (left) for a test video. Our model's MIDI prediction captures the structure of the music piece and predicts most of the ground truth notes correctly. Note that our model gives sparser predictions than the ground truth (e.g. darker vertical lines appear around 1:00 min in the spectrogram on the right).

6. DISCUSSION

We proposed an end-to-end deep-learning framework to tackle the problem of transcribing piano music from visual data alone. Our system predicts note onset events given a top-view video of a person playing the piano. We demonstrated this on different test sets which vary in difficulty. Here, we focussed on piano data but our method could be extended to any other instrument with a spatial layout similar to the piano (e.g. organ, harpsichord, marimba, harp, etc.). We trained our frameworks with pseudo ground-truth data but it would be interesting to use actual ground truth data for training.

Further work should be done to allow for different viewpoints and to better exploit the temporal information between distant frames.

Acknowledgements

This work is supported by the EPSRC programme grant Seebibyte EP/M013774/1: Visual Search for the Era of Big Data. We thank Ruth Fong for help with smoothing the output.

7. REFERENCES

[1] Potcharapol Suteparuk, "Detection of piano keys pressed in video," *Dept. of Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep*, 2014.

[2] Mohammad Akbari and Howard Cheng, "Clavision: visual automatic piano music transcription.," in *NIME*, 2015.

[3] Albert Nisbet and Richard Green, "Capture of dynamic piano performance with depth vision," https://albertnis.com/resources/2017-05-10-piano-vision/Nisbet_Green_Capture_of_Dynamic_Piano%20Performance_with_Depth_Vision.pdf.

[4] Seungmin Rho, Jae-In Hwang, and Junho Kim, "Automatic piano tutoring system using consumer-level depth camera," in *International Conference on Consumer Electronics (ICCE)*. IEEE, 2014.

[5] Souvik Sinha Deb and Ajit Rajwade, "An image analysis approach for transcription of music played on keyboard-like instruments," in *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing*. ACM, 2016.

[6] Shir Goldstein and Yael Moses, "Guitar music transcription from silent video," in *BMVC*, 2018.

[7] Bingjun Zhang, Jia Zhu, Ye Wang, and Wee Kheng Leow, "Visual analysis of fingering for pedagogical violin transcription," in *Proceedings of the 15th ACM international conference on Multimedia*. ACM, 2007.

[8] A. Sophia Koepke, Olivia Wiles, and Andrew Zisserman, "Visual pitch estimation," in *Sound and Music Computing Conference*, 2019.

[9] Alessio Bazzica, J.C. van Gemert, Cynthia C.S. Liem, and Alan Hanjalic, "Vision-based detection of acoustic timed events: a case study on clarinet note onsets," in *Proc. of the First Int. Workshop on Deep Learning and Music*, 2017.

[10] Pablo Zinemanas, Pablo Arias, Gloria Haro, and Emilia Gomez, "Visual music transcription of clarinet video recordings trained with audio-based labelled data," in *ICCV Workshop*, 2017.

[11] Christian Dittmar, Andreas Männchen, and Jakob Abeber, "Real-time guitar string detection for music education software," in *14th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2013.

[12] Jangwon Lee, Bardia Doosti, Yupeng Gu, David Cartledge, David Crandall, and Christopher Raphael, "Observing pianist accuracy and form with computer vision," in *Proc. WACV*, 2019.

[13] Mohammad Akbari, Jie Liang, and Howard Cheng, "A real-time system for online learning-based visual transcription of piano music," *Multimedia Tools and Applications*, vol. 77, no. 19, 2018.

[14] Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, 2018.

[15] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck, "Onsets and frames: Dual-objective piano transcription," in *Proc. ISMIR*, 2018.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. ICCV*, 2016.

[17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in PyTorch," in *NeurIPS Autodiff Workshop*, 2017.

[18] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] Mert Bay, Andreas F. Ehmann, and J. Stephen Downie, "Evaluation of multiple-f0 estimation and tracking systems.," in *Proc. ISMIR*, 2009.

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	Sight to sound: An end-to-end approach for visual piano transcription
Publication Status	Under submission
Publication Details	Almut Sophia Koepke, Olivia Wiles, Yael Moses, and Andrew Zisserman: *Sight to sound: An end-to-end approach for visual piano transcription*. Under submission

Student Confirmation

Student Name:	ALMUT SOPHIA KOEPKE		
Contributions to the Paper	Developing the idea for the task and network architecture. Curating the piano video datasets. Running all experiments and writing the paper.		
Signature	A. S. Koepke	Date	19/12/19

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 ANDREW ZISSERMAN			
Supervisor comments Sophia originated the idea and was the driving force for the paper with Olivia and Yael providing some support.			
Signature	A. Zisserman	Date	19/12/19

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

Chapter 7

Self-supervised cross-modal learning for music

In this chapter, we give a brief summary of our work in progress on self-supervised learning from audio-visual music data which connects our work presented in chapter 3 and chapter 4 with our work on cross-modal learning. We include experimental results for audio-visual synchronisation and lay out directions for further research.

7.1 Motivation

The primary idea for most frameworks that use self-supervised learning from audio-visual data (as described in chapter 2) is to leverage the temporal synchronisation of audio and visual information as a training signal by distinguishing temporally matching pairs of audio and visual information from non-matching pairs. We propose to similarly use the temporal co-occurrence of audio and visual information for videos of piano playing for training. An example of temporally matching and non-matching audio-visual pairs is shown in figure 7.1.

Previous work on self-supervised training for synchronisation has shown to produce rich audio and visual features [Arandjelović and Zisserman, 2017, 2018; Korbar *et al.*, 2018]. Similar to their observations, we expect that training with an audio-visual synchronisation task would produce good features for subsequent training on a downstream task (e.g. visual music transcription). Korbar *et al.* [2018] observe that a careful curriculum for training allows them to learn significantly better features. However, piano playing synchronisation is also very useful as a task in itself, since exact alignment of audio and visual information is desired and can be tedious when done manually post recording. We propose to train a framework to address audio-visual synchronisation similar to Chung and Zisserman [2016a] to align the pianists'

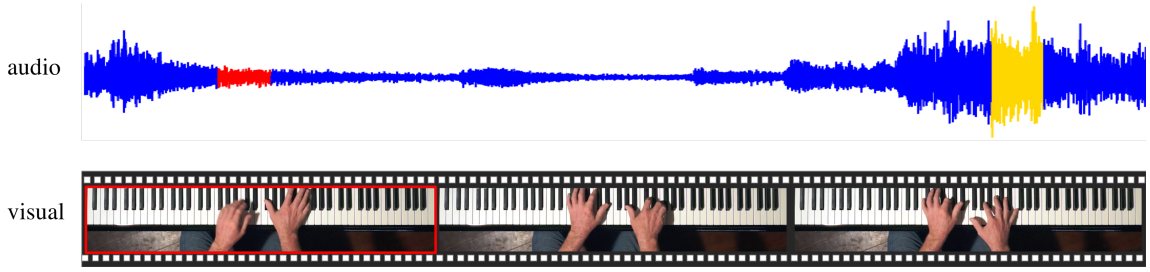


Figure 7.1: Visualisation of the audio and visual streams for a video of piano playing. Matching audio-visual information is highlighted in the same colour (i.e. red video paired with red audio); an example of a negative pair would be realised by differently coloured audio-visual clips (i.e. red visual frame and audio segment highlighted in yellow).

finger movements with the corresponding audio. We claim that this task can be challenging since, in addition to the movement of the fingers to hit a key, the hands themselves move across the keyboard and are not stationary. In the case of lip-speech synchronisation, the use of face detections results in the mouth being in the same position within the input frame where only the shape of the mouth varies. Furthermore, when playing the piano, movements of fingers can occur without hitting a key. However, relying on visible keypresses instead of finger motion alone is not feasible either, because keys can be occluded by the hands or not visible directly from above.

7.2 Method

7.2.1 Model

Our model consists of a visual stream and an audio stream (see figure 7.2). The visual stream is based on the ResNet18 architecture [He *et al.*, 2016] and takes 5 consecutive greyscale video frames as input. The audio stream is based on the CREPE network architecture [Kim *et al.*, 2018] for pitch estimation from audio which takes 0.2 second raw audio segments as input. The network is trained to align features of matching audio and visual inputs and to separate non-matching ones.
Following Chung and Zisserman [2016a], we minimise a contrastive loss function l_{con} [Chopra *et al.*, 2005] for training. For a pair of audio and visual features, a and v,

$$l_{con}(v, a, y) = \frac{1}{2}\left((1-y)d^2 + y(\max(0, m-d))^2\right), \tag{7.1}$$

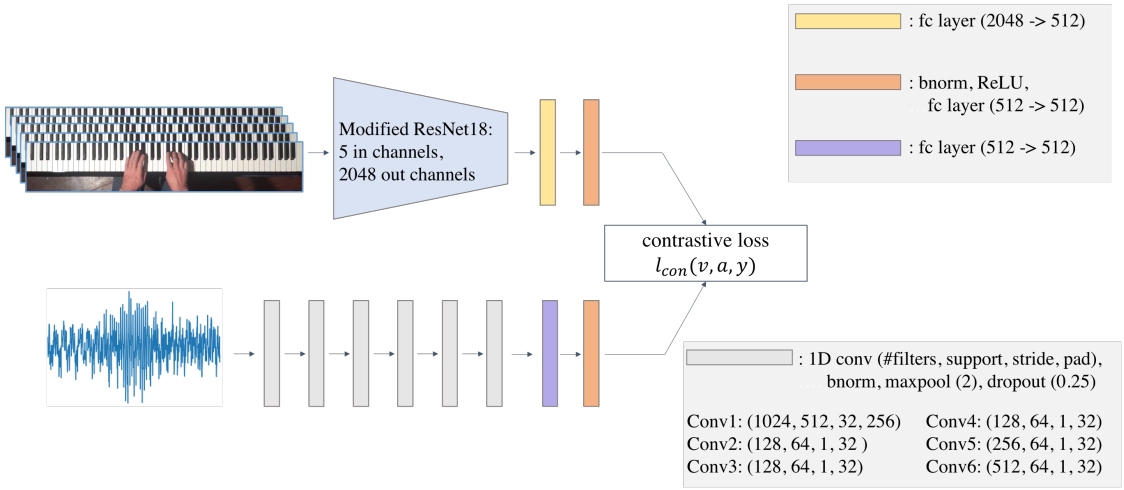


Figure 7.2: Network architecture for audio-visual synchronisation. Our network consists of a visual and an audio stream. 5 consecutive video frames are input to the visual stream. It consists of a slightly modified ResNet18 architecture which takes as input 5 channels (5 concatenated greyscale images) and outputs a 2048-dimensional feature vector. This is passed through two fully-connected layers to give the visual feature v. A corresponding or non-corresponding $0.2s$ raw audio segment is input to the audio stream which consists of 6 1D-convolutional layers. Its output goes through two fully-connected layers to give the audio feature a. The network is trained using a contrastive loss function l_{con} (see equation 7.1).

where $d = \|v - a\|_2$, m is a margin which we choose to be 1, and y is the label which is 0 for non-matching and 1 for matching audio-visual pairs.

7.2.2 Dataset and training details

Our framework is trained using the PianoYT dataset introduced in chapter 6 which consists of spatially aligned top-view piano videos. The visual frames are cropped and resized to 145×800 pixels. The model is trained in PyTorch [Paszke *et al.*, 2019] using the Adam optimizer [Kingma and Ba, 2015] ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with a batchsize of 24. The learning rate was set to 0.0001 and training was stopped when the validation loss plateaued. Positive audio-visual pairs consist of matching audio and visual frames for a temporal segment of $0.2s$. Negative pairs can be from the same or from a different video, but the temporal windows of the audio and visual inputs do not overlap. No curriculum strategy is used for training.

7.3 Experiments

We visualise the distance between visual and audio features in figure 7.3. We show the averaged L_2 distance over all clips in the test set between each visual feature and the audio features within a time window of ± 2 seconds around the true corresponding audio. An offset of 0 corresponds to the true match which is where the distance is minimal as desired. As can be seen in figure 7.3, our model learns to embed audio and visual features closer together for matching audio and visual inputs as compared to taking audio input that is shifted. The results show that our model could be used for temporal audio-visual synchronisation of videos of piano playing.

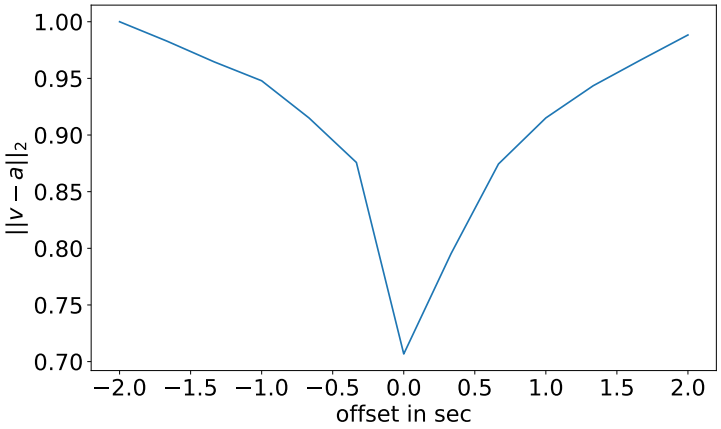


Figure 7.3: Visualisation of synchronisation results on the PianoYT test set. L_2 distance between audio and visual features for 0.2 second audio features within a window of ± 2 seconds of the true match.

7.4 Discussion

We proposed a self-supervised framework for automatic temporal alignment of audio with its corresponding video for videos of piano playing. We have presented preliminary results which show that the network learns to align audio and visual features. However, our results leave room for improvement, for instance, by adding a curriculum for training and data augmentation. The exploration of the usefulness of the resulting features beyond audio-visual synchronisation remains to be shown.
As a next step, we will use the features obtained by training for audio-visual synchronisation as an initialisation to train for and evaluate visual music transcription.

To further aid the learning of synchronisation, we will consider using motion information more explicitly than just using multiple input frames, by leveraging optical flow. Furthermore, we will explore spatial attention modules similar to the framework by Arandjelović and Zisserman [2018] which might improve the learnt features. Taking longer time segments as input might be useful for the network to exploit more information about the context.

Zhao *et al.* [2019] have recently shown promising results for coarse sound source localisation which go beyond cross-modal semantic matching. Similar to other audio-visual frameworks mentioned above, they employ a curriculum training strategy and are then able to separate the sounds of the same kind of instruments (e.g. two violins). Adding another curriculum step to the training would ideally allow us to identify and separate out sounds being played on the keyboard by individual fingers. However, this requires very fine-grained attention to the correspondence between pitch and finger localisation on the keyboard. If we were able to achieve this from self-supervision, we could directly read off music transcription from the sound localisation because of the alignment of the piano in our data. This would remove the need to rely on pseudo-ground truth from audio to train for visual music transcription. However, it is unclear how to supervise this, since we do not have separate audio and visual data for individual fingers. Zhao *et al.* [2019] are able to separate instruments at test time by learning to separate synthetically mixed separate instruments' recordings at training time.

Chapter 8

Summary and extensions

We conclude by summarising the contributions of the work presented in this thesis and by discussing closely related recent work that has been published concurrently or since our publications, some of which builds on our work. Furthermore, we propose directions for future work.

Self-supervised proxy task: In chapter 3 and chapter 4, we proposed the self-supervised proxy task of learning to warp one face image into another image from the same video face track in order to be able to control face generation (*X2Face*) and to learn a useful face embedding (*FAb-Net*). This is done by learning to predict a per-pixel bilinear sampling location requiring pairs of face images from the same video face track.

Whilst this self-supervised setup has demonstrated many benefits, it could potentially be improved by making use of the temporal content and relation between the source and target frames. Currently, our frameworks rely on pairs of video frames without exploiting their temporal relation more explicitly. Similar to the framework by Wang *et al.* [2019], which makes use of temporal cycle-consistency, we could learn to track along time by tracking forward and backward. The discrepancy between start and end points can then serve as a learning signal.

It would be interesting to explore incorporating audio information along with the visual information in order to learn good class representations. On the other hand, additionally being able to train on image collections instead of requiring video data would open new possibilities. Our frameworks are optimised using a photometric loss; alternatively, using perceptual loss functions that capture more semantic content could enable training on image collections.

X2Face: The *X2Face* setup presented in chapter 3 allows us to control the pose and expression of a given face by using another face image or modality (i.e. head pose, or audio) and results in the disentangling of texture and shape. The same network can be used for lightweight video editing of the face texture by, for instance, adding glasses.

Several works have built on this framework since it was published. Siarohin *et al.* [2019a,b] animate images of objects according to a driving video by incorporating unsupervised keypoint detection in their framework. The unsupervised keypoints are used to predict dense motion. Siarohin *et al.* [2019a] use an U-Net architecture to generate images and do not rely on a bilinear sampler. Image details that were not present in the source image can be hallucinated (e.g. teeth that show when smiling or speaking). Their method, unlike *X2Face*, generalises to other object classes, such as humans. Siarohin *et al.* [2019b] improve this further by representing the transformation as a set of learnt displacements of unsupervised landmarks with corresponding local affine transformations. Grigorev *et al.* [2019] learn to complete texture maps which enables the generation of new views that are very different from the input view. Zhou *et al.* [2019a] learn a GAN model for talking faces that disentangles identity and speech-specific information by aligning the shared information from the visual and audio inputs. Training requires word and identity labels; at test time either audio or visual information can be used to drive the face image generation. However, the head pose of the input face is not modified when frames are generated (even the ones that are generated using another video). There have been several other recent methods that allow the generation of talking heads given input text [Fried *et al.*, 2019; Doukas *et al.*, 2019], audio [Chen *et al.*, 2019], facial landmarks [Songsri-in and Zafeiriou, 2019], or facial boundary maps [Qian *et al.*, 2019]. Those methods, unlike *X2Face*, do not suffer from the problem of identity information from the driving video leaking through the model to the source identity.

Other recent methods fit a 3D face model to an image which is then used to generate arbitrary face expressions for the same person [Geng *et al.*, 2019; Pumarola *et al.*, 2018; Ververas and Zafeiriou, 2019; Kim *et al.*, 2019]. Tripathy *et al.* [2019] introduce ICface, a GAN framework which builds on the *X2Face* setup by adding an adversarial loss on the synthesised neutral image and the generated image and by using head pose and facial action unit ground-truth information during training. As a result, controlling face generation does not require learning additional mappings to annotated datasets, as is the case for our *X2Face* framework. As can be seen in figure 8.1, ICface

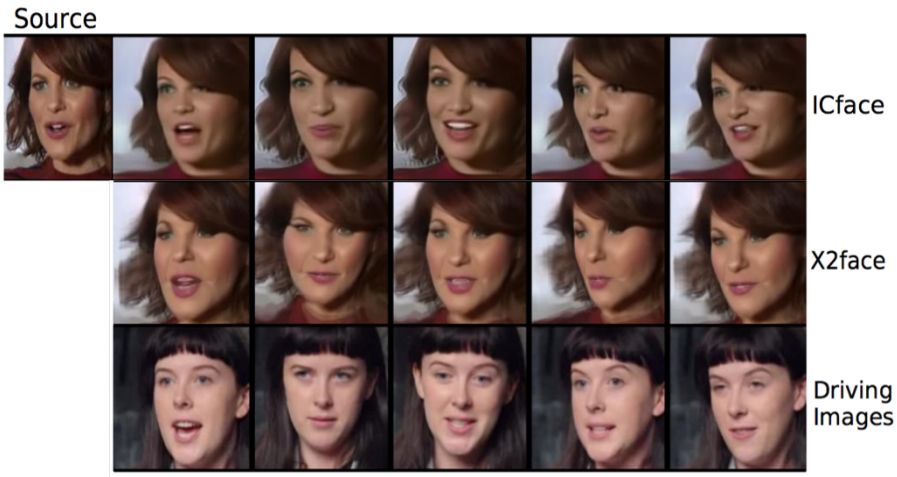


Figure 8.1: Comparison of generation results from ICface [Tripathy *et al.*, 2019] and *X2Face* taken from Figure 3 in [Tripathy *et al.*, 2019]. ICface better preserves the identity information during generation than *X2Face*.

does not have the problem of identity information from the target frame getting mapped onto the generated frame of a different source identity.

Gu *et al.* [2019] combine both warping and appearance-based generation of talking heads with adversarial loss functions in FLNet. Facial landmark information is used to select diverse expressions in the input source frame set and as input to the network which learns to warp the source frames. FLNet can generate unseen details (such as teeth) through an appearance stream (see figure 8.2). It generates much finer unseen details (such as eyelids and teeth) than the GANimation framework by Pumarola *et al.* [2018], which requires fitting a 3D morphable model.

Zakharov *et al.* [2019] propose another GAN framework that uses facial landmark information along with multiple frames from the same face track to disentangle pose-specific information from appearance-based information. In order to generate face images with new poses and expressions, a target landmark representation is input into an encoder-decoder network which gets appearance-based information through adaptive instance normalisation between landmark-specific features and appearance-based features. While their results seem visually convincing, subtle details that are not captured in the landmark representation cannot be generated.

One of the main advantages of other, more recent methods for controlling face image generation seems to result from incorporating adversarial losses to prevent identity information from leaking from the target image to the source image and to improve the image quality. It would be interesting to incorporate those findings into the

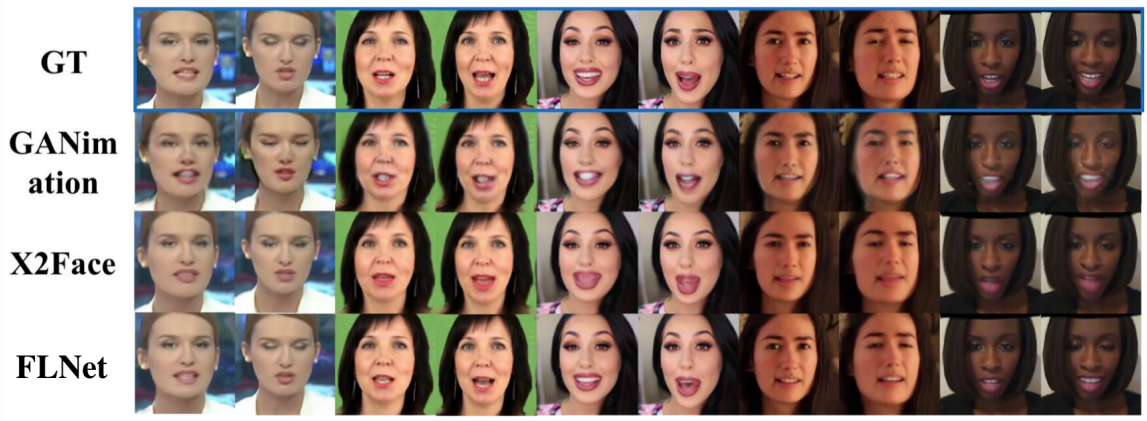


Figure 8.2: Comparison of generation results from FLNet [Gu *et al.*, 2019], GANimation [Pumarola *et al.*, 2018] and *X2Face* taken from Figure 3 in [Gu *et al.*, 2019] using 5 source frames. Unlike *X2Face*, GANimation and FLNet can generate unseen regions like teeth, with FLNet giving much more fine-grained details than GANimation.

X2Face framework.
However, whilst the quality of face generation results is impressive, the easy accessibility of such tools also raises concerns regarding their abuse. As a research community we should thrive for societal awareness of the existence and capabilities of such systems. Furthermore, we should aim for the introduction of regulations that will minimise their misuse.

FAb-Net: In chapter 4, we presented the *FAb-Net* framework which learns a self-supervised facial attribute embedding that contains useful information for head pose estimation, facial landmarks prediction and facial expression estimation. The embedding was trained using our proposed self-supervised proxy task.
Both, our *X2Face* and *FAb-Net* frameworks make use of per-pixel bilinear sampling similar to the Spatial Transformer Networks introduced by Jaderberg *et al.* [2015] to generate a target frame. However, bilinear interpolation is a very local operation; it considers only the four direct pixel neighbours of a selected pixel. This makes it difficult to learn big transformations. Ideally, the transformation would be determined by computing a cost volume that models the probability distribution of where each pixel in the target frame maps to in the source frame. However, a cost volume at high resolution is expensive to obtain. We introduced a framework [Wiles *et al.*, 2019] (see appendix A) which builds on the work presented in chapter 4. It uses a hierarchical upsampling mechanism that has to learn a full cost volume only at a low resolution. The cost volume at low resolution constrains the predicted sampling

locations at higher resolutions. Because it is able to model bigger transformations, this framework can learn better class representations, including for more difficult classes, such as human bodies and horses. Other recent methods have addressed the problem of local gradients by improving the sampling mechanism itself. Jiang *et al.* [2019] use auxiliary sampling locations around a chosen pixel and obtain the intensities at those auxiliary locations using bilinear sampling. The estimated point intensities and coordinates are used to linearly approximate the chosen pixel. These linear approximations are used as pixel representations in the transformed image providing bigger context than simple bilinear interpolation.

A concurrent line of work [Jakab *et al.*, 2018, 2019] relies on unsupervised keypoints discovery as object class representations. This is achieved by factorising appearance and geometry information through a differentiable bottleneck which produces landmarks as centres of Gaussians with fixed standard deviation. The recent work by Jakab *et al.* [2019] additionally uses a discriminator on the discovered keypoints to enforce that the resulting skeleton maps (in the case of humans) appear similar to skeleton maps on a different keypoint annotated dataset.

Another recent framework [Thewlis *et al.*, 2019] improves the unsupervised keypoint learning method of Thewlis *et al.* [2017a]. It extends the dense equivariant labelling by going via a third face image of a different identity to achieve better generalisation across different face identities.

Li *et al.* [2019b] also learn a facial action unit representation in a self-supervised way by disentangling movements into those related to facial expression and those related to head motion. However, they do not compare on the same datasets as we did in our *FAb-Net* work; instead, they re-train our framework and report results on other datasets which they claim are worse for *FAb-Net* than using the split-brain proxy task (contrary to our observations presented in our paper).

Bertasius *et al.* [2019] build on the *FAb-Net* architecture in order to propagate sparse human pose labels across videos. Their *PoseWarper* network learns to predict the pose in an unlabelled frame by using the known pose information in another frame and by learning to relate frames. Two frames are related to one another by considering the difference of sparse pose heatmaps which can be used at inference time to propagate labels across times.

The *FAb-Net* framework gives a low-dimensional embedding that encodes useful information about facial attributes, as the embedding space is the only part of the setup where the source and target frames can share information. Possible future directions include considering denser representations (the *FAb-Net* embedding is a

256-dimensional vector) taking inspiration from the recent work by Han *et al.* [2019]. Furthermore, it would be interesting to combine our proxy task with other proxy tasks that encourage learning about connected components (e.g. semantics - in the case of faces, face parts) in the image, such as colourisation. In addition to that, instead of predicting a warp field and optimising a loss function that compares warped and ground-truth RGB image pixel values, our framework could be tasked to estimate optical flow obtained from other (self-supervised) methods.

Visual pitch estimation for violin playing: In chapter 5, we presented a framework for estimating musical notes from videos of violin playing using visual information alone. For this, we curated a dataset which consists of monophonic (only one note sounding at a time) violin videos recorded in constrained conditions, and YouTube videos of solo violin playing.
Our visual pitch estimation method for violin playing works only for a limited range of poses. There is a lot of variation in how violin playing videos are recorded and in how a violinist holds the instrument. However, our current method does not generalise to extreme poses - it has been trained only on poses where the hands and instruments are visible in a way that would allow a trained musician to recognise the played pitch from the visual information. Furthermore, our current framework can only estimate monophonic music because the audio-based method that generates the pseudo ground-truth pitch labels does not work for polyphonic music. It would be interesting to extend our framework to also work in those cases, for instance, by aligning the video's audio with the underlying sheet music in order to obtain ground-truth information. In the case of polyphonic music, visual information could be especially useful for resolving ambiguities arising when considering audio information alone.
Another application where visual information might be very useful would be when incorporating hand pose estimation methods to discover the relevant finger of the left hand that is producing the sounding note. This would enable the retrieval of information about the fingering used from videos.

Visual pitch estimation for piano playing: In chapter 6, we presented a framework for estimating note onsets from silent top-view videos of piano playing. Our estimated note onsets from visual information alone replicate the structure of a music piece fairly well. However, there are several directions to pursue in order to improve this.

Our current framework takes 5 visual frames as input and outputs per-key MIDI predictions for that temporal segment. However, the network seems to focus on the visible key presses rather than the motion of the fingers, the latter of which could give a much more accurate signal about onsets (starts of key presses as opposed to ongoing key presses). This results in false positive onset predictions. If the network focused more on the actual finger motion, note predictions for cases where keys are occluded by the hands (see figure 1.5) could be improved. Furthermore, such an improved framework could be used to generate MIDI from videos of hand and finger motion on any background, e.g. a table top. Such a video of playing the piano on a table top could be pasted onto a keyboard background before inputting it into our network.
Furthermore, our current setup learns from pseudo ground-truth information obtained from the audio which does not always give reliable information, in particular about note endings. Curating a dataset with recorded MIDI ground-truth information would allow us to additionally learn note durations and not only the note onsets.
Another avenue to pursue would be to train on data of a single pianist only, so that variations arising from different people's hands and instrument setups would be minimised. This could result in a model that gives very accurate predictions for the person and instrument it is trained for.
Furthermore, it would be interesting to lift the constraint on the crop to be directly from above with the keyboard being fully visible. This could be done by, firstly, allowing for different crops of the keyboard, and, secondly, by making the method useful for a broader variety of viewpoints (such as side views). The same method could be extended to be used with other keyboard-like instruments which have a similar setup as the piano, such as the harpsichord, organ, or the marimba.

Appendix A

Self-supervised learning of class embeddings from video

This work was presented as a *poster* at the ICCV workshop on Compact and Efficient Feature Representation and Learning in Computer Vision, 2019.

In this paper, we build on the *FAb-Net* architecture by introducing a hierarchical upsampling mechanism that allows to model bigger transformations. This framework enables us to not only improve the learnt face embedding, but it can also be trained on human bodies and animals and it gives convincing class representations for those.

Self-supervised learning of class embeddings from video

Olivia Wiles
University of Oxford
ow@robots.ox.ac.uk

A. Sophia Koepke
University of Oxford
koepke@robots.ox.ac.uk

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

Abstract

This work explores how to use self-supervised learning on videos to learn a class-specific image embedding that encodes pose and shape information. At train time, two frames of the same video of an object class (e.g. human upper body) are extracted and each encoded to an embedding. Conditioned on these embeddings, the decoder network is tasked to transform one frame into another. To successfully perform long range transformations (e.g. a wrist lowered in one image should be mapped to the same wrist raised in another), we introduce a hierarchical probabilistic network decoder model. Once trained, the embedding can be used for a variety of downstream tasks and domains. We demonstrate our approach quantitatively on three distinct deformable object classes – human full bodies, upper bodies, faces – *and show experimentally that the learned embeddings do indeed generalise. They achieve state-of-the-art performance in comparison to other self-supervised methods trained on the same datasets, and approach the performance of fully supervised methods.*

1. Introduction

How much information is needed to learn a representation of an object class? Do we require separate representations for different aspects: e.g. one representation for 3D, another for pose, another for 2D landmarks? We investigate how to learn a single representation for a given object class that encodes multiple properties in a self-supervised manner. This representation can be used for further downstream tasks and domains with minimal additional effort.

We learn this representation – which we call an *image embedding* – in a self-supervised manner from a large collection of videos of that object class (e.g. human upper bodies, or talking heads). The principal assumption is that of *temporal coherence* – that frames of the video contain the object class, but *no* additional prior auxiliary information is required.

In order to learn the image embedding from a video dataset, the following proxy task is used. Given two frames from the same video, their image embeddings are used to warp one of the frames into the other.

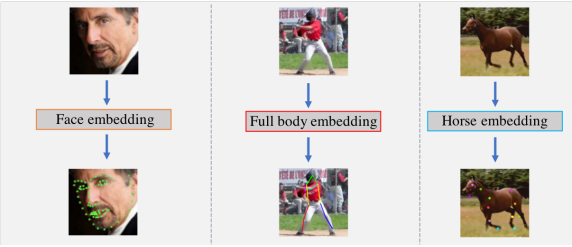


Figure 1. The aim of this work is to obtain a class-specific *image* embedding by self-supervised learning on a large collection of videos. The learned embedding can then be used for a variety of downstream tasks and datasets.

We want to model long range dependencies at high resolutions, for example large hand movements. In order to do this, we instantiate the warp probabilistically – for every pixel in one frame, we would like to predict the probability that that pixel corresponds to every other pixel in the other frame. Doing this naively is computationally prohibitive above a small (e.g. 32×32) resolution.

As a result, we use a hierarchical approach to perform this operation. The model first learns the probabilities at a low resolution, before refining the probabilities at successive layers while conditioning on the lower resolution predictions. While solving the proxy task at a small resolution may seem trivial, in fact low resolution images encode important salient information such as spatial layout and context [46]. This approach is inspired by the classical (i.e. pre deep learning) multi-resolution methods employed for optical flow and stereo matching [1, 4, 26, 29].

The embedding, trained using only pairs of video frames, is then used for the tasks of predicting landmarks and their visibility on a variety of datasets which may differ substantially from the initial dataset. Our paradigm is useful in applications, as it requires only one large network per class and one additional small network per down stream task.

In summary, our contributions are as follows.

1. A self-supervised class embedding (Section 3) that can

model complex large movements, e.g. the movement of arms or hands.

2. A hierarchical probabilistic network that allows us to estimate the probability that a given pixel in a given frame matches each pixel in another frame of the same video for high resolution images.

3. Two additional losses for learning this embedding. The confidence loss (Section 3.2) allows the model to express what portions of the target image can be reliably predicted from the source and what portions cannot. The cyclic loss (Section 3.3) enforces that the model does not degenerate into a trivial solution.

4. We demonstrate that the method learns a useful representation that can be used for downstream tasks on the same or different domains for a variety of object classes. Our method achieves state-of-the-art performance in comparison to other self-supervised methods trained on the same datasets. Finally, we show qualitative examples of using our approach for a non-human class, that of horses.

2. Related work

Here, we focus on self-supervised learning from video. We also cover class specific modelling, where a model of the object is extracted using auxiliary information and then applied to novel images.

Self-supervised learning on video collections. Learning from video [2, 10, 15, 17, 21, 22, 30, 31, 35, 40, 42, 47, 52, 62, 64] is a powerful paradigm, as unlike with image collections, there is additional temporal and sequential information. The aim of self-supervised learning from video can be to learn to predict future frames [47], or to learn to predict depth [12, 14, 62]. However, we are interested in learning a set of useful features (e.g. frame representations).

One approach is to use the temporal ordering or coherence as a proxy loss in order to learn the representation [10, 17, 22, 24, 30, 31, 49, 52, 64]. Other approaches use egomotion [2, 21] in order to enforce equivariance in feature space [21]. In contrast, [23] predicts the transformation applied to a spatio-temporal block. Instead of enforcing constraints on the features, one can learn features using a generative task of future or input frame prediction [15, 40, 42]. Another approach is to use colourisation to learn features and to track objects [48].

Unlike these works, our focus is to learn a feature representation for a specific class, which can be used to predict class-specific attributes. Most similar to our method is [53] which uses video to learn a representation of faces. However, they do not consider other object classes.

Self-supervised learning of landmarks. Instead of using proxy tasks to learn useful features, another line of self-supervised learning is to explicitly learn a set of landmarks. This can be done by conditioning image generation on the image landmarks [19, 60]. Another approach is to recover object structure by enforcing equivariance to image transformations [43, 44].

3. A self-supervised representation

This section introduces our self-supervised model and architecture (Fig. 2). The model is trained for the proxy task of transforming one frame into another frame in a hierarchical manner (Section 3.1). We allow the model to express uncertainty (Section 3.2) and use additional cyclic constraints (Section 3.3) to stop the learned transformation from degenerating. This gives the final training objective. We introduce the framework for the case of human upper bodies, but the same framework is used for the other classes considered in this paper (full human body, talking faces, horses).

3.1. Proxy task to train the network: Modelling the transformation between images

A source frame I_S and a target frame I_T are randomly selected from the same video. The proxy task to train the model consists of learning how to warp the source frame I_S into the target frame I_T. Both frames are mapped, using a convolutional encoder with shared weights, to image embeddings e_S and e_T respectively.

Conditioned on these embeddings, the model predicts the probability of a pixel in the generated frame I_G matching each pixel in I_S. These probabilities are used to generate the colour of a pixel by taking the weighted average. To introduce our notation, let $I_{S_{kl}}$ and $I_{T_{ij}}$ be the colours for pixel locations (k, l) and (i, j) in the source and target frame respectively. The network predicts the colour in the generated frame I_G at pixel location (i, j) as a linear combination of pixels in the source frame

$$I_{G_{ij}} = \sum_{k,l} \mathbf{A}_{ij,kl} I_{S_{kl}}, \tag{1}$$

where $\mathbf{A}_{ij,kl}$ is the probability that a pixel $I_{T_{ij}}$ in the target frame matches a pixel $I_{S_{kl}}$ in the source frame. We explicitly predict the match similarity $\mathbf{M}_{ij,kl}$ between a pixel $I_{S_{kl}}$ and $I_{T_{ij}}$ and normalise the $\mathbf{M}_{ij,kl}$ to give $\mathbf{A}_{ij,kl}$ (see Eqs. (3)-(5)). I_G should match the target frame I_T (Fig. 2a), which we enforce using a photometric L1 loss

$$\mathcal{L}_{ph} = |I_G - I_T|_1. \tag{2}$$

While using the naive weighted sum works for smaller resolution images, for larger images this becomes computationally prohibitive. To deal with this problem, we introduce our hierarchical approach (Fig. 2b). Learning in a

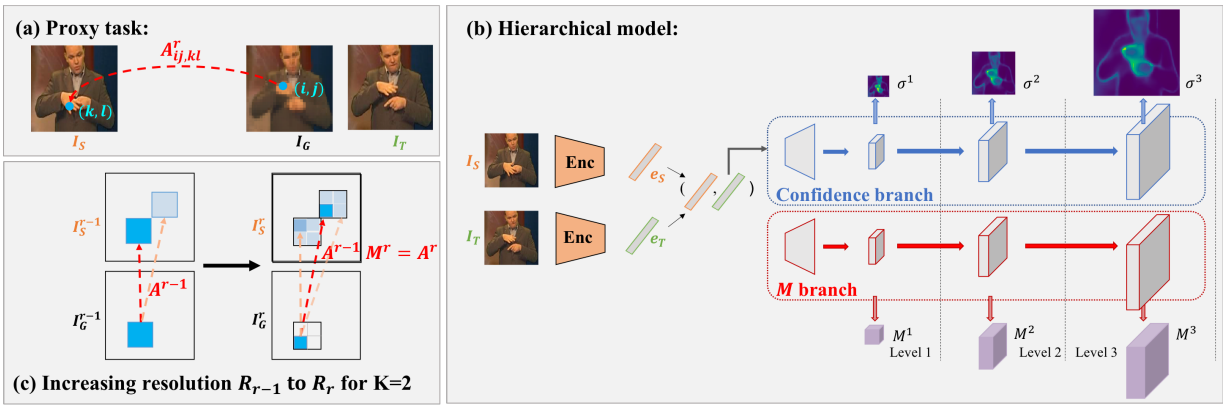


Figure 2. An overview of the approach. (a) The proxy task used to train the model. Given a source frame I_S and a target frame I_T, our model learns a mapping $\mathbf{A}^r$ to warp the source frame into a generated frame I_G. I_G should match I_T. (b) The model in more detail. The two frames are mapped to embeddings e_S, e_T. Conditioned on these embeddings, the model predicts the warp at an initial resolution $R_1 = (32 \times 32)$ as well as a confidence σ^1 for each pixel. These predictions are then refined at successively higher resolutions. (c) Illustration of how the predicted $\mathbf{M}^r$ at each resolution R_r are used to determine the warp $\mathbf{A}^r$.

hierarchical manner has been found to be useful in a number of tasks [9, 27, 28, 50, 51]. In our case, the network learns to determine roughly how to transform points (e.g. bigger parts of the image, like the arms) at a low resolution ($\mathbf{M}^1$ at level 1). This transformation is refined progressively at higher resolutions ($\mathbf{M}^r$ at level r). At these higher levels, the network can learn to focus on the details (e.g. the placement of the wrists). This can be regarded as a form of curriculum learning [3] where the decoder is progressively expanded in levels which increase the resolution of the generated image.

Probabilistic prediction at a low resolution (Training level 1). At the lowest resolution, $R_1 = (W_1 \times W_1) = (32 \times 32)$, we explicitly predict the probability $\mathbf{M}^1_{ij,kl}$ that each point (k, l) in the source frame matches each point (i, j) in the target frame. We then take the weighted average to obtain the probability distribution $\mathbf{A}^1_{ij,kl}$:

$$\mathbf{A}^1_{ij,kl} = \exp(\mathbf{M}^1_{ij,kl}) / \sum_{m,n} \exp(\mathbf{M}^1_{ij,mn}). \quad (3)$$

Using the computed probability distribution, we obtain the generated frame (Eq. (1)).

Refining the prediction at a higher resolution (Training level r.) Given the generated frame I^{r-1}_G at resolution R_{r-1}, we seek to refine I^{r-1}_G to obtain I^r_G at a higher resolution R_r. For a given location (i, j), the highest $\mathbf{A}^1_{ij,kl}$ give the most likely locations that (i, j) points to in the source frame. We will use this to limit the locations we consider at the higher resolution (see Fig. 2c).

In a traditional CNN, as we decode, we would have to keep track of the probabilities for a pixel (i, j) matching every pixel (k, l) in the source frame at that resolution. So doubling the resolution of the generated image at each layer requires quadrupling the number of predicted probabilities. Our insight is that keeping track of all of these probabilities is unnecessary. For a given pixel (i, j), we can throw away the unlikely matches at lower resolutions (effectively setting them to 0) while keeping track of the top K matches. Then when we double the resolution at the next layer, we only need to predict $4K$ values (if the width and height of the generated image has doubled, then one pixel at the lower level corresponds to four pixels at the higher level as illustrated in Fig. 2 c)). Instead of using these predicted $4K$ values as raw probabilities we use them to re-weight the probabilities predicted at the lower resolution to make the process differentiable. This leads to a sparser representation that grows quadratically.

The M-branch decoder is used to obtain the $4K$ values $\mathbf{M}^r$. These are multiplied by the probabilities at the lower resolution and a softmax normalisation is performed to obtain the final probability distribution $\mathbf{A}^r$:

$$\mathbf{P}^r_{ij,kl} = \mathbf{A}^{r-1}_{\lfloor \frac{i}{2} \rfloor \lfloor \frac{j}{2} \rfloor, \lfloor \frac{k}{2} \rfloor \lfloor \frac{l}{2} \rfloor} \mathbf{M}^r_{ij,kl} \quad (4)$$

$$\mathbf{A}^r_{ij,kl} = \exp(\mathbf{P}^r_{ij,kl}) / \sum_{m,n} \exp(\mathbf{P}^r_{ij,mn}). \quad (5)$$

Discussion. Our aim is to compute a cost volume that models the probability distribution of where a pixel in the target frame maps to in the source frame. [11] introduced using a cost volume in a deep learning framework for optical flow by computing the similarity between features. This idea has been leveraged in many recent works [45, 48]. However, naively comparing features at a $W \times W$ resolution requires computing W^4 values which quickly becomes prohibitively

large. As a result these methods are forced to use a small cost volume or a tiny batch size.

The grid sampler introduced in [18] provided another way to model the transformation between images by explicitly learning the warp field. This was used effectively by [53] in order to learn meaningful embeddings for faces. However, gradients only occur in the local neighbourhood of a point. As a result if the point needs to travel a large distance between images and there is no smooth colour transition (as is common in most images), then these gradients will be useless and the model will fail to learn.

Our hierarchical approach gives a way to address the limitation of both approaches. We can grow the cost volume to image resolutions of the same size as the original image with minimal overhead. We additionally do not suffer from the problem of local gradients. Finally, the hierarchical approach enforces the spatial constraint – that pixels in a local neighbourhood move together.

3.2. Modelling occlusion and background

When modelling the transformation between frames it is possible for part of an object to become occluded (e.g. the hand moving in front of the face) or un-occluded. Additionally, there may be parts of the scene that are not visible in the previous frame (e.g. for the signing videos the background is a video itself and constantly changing).

To allow the model to express uncertainty due to these challenges, we use an additional decoder which explicitly models the confidence σ^r at resolution R_r for the transformation at each location in I_G. Following [33], we assume that the pixel-wise confidence measure is Laplace distributed and use it to reweight the photometric loss $\mathcal{L}^r_{ph}$ at each pixel:

$$\mathcal{L}^r_{con} = \sum_{i,j} -\ln \frac{\sqrt{2}}{2\sigma^r_{ij}} \exp\left(-\frac{\sqrt{2}|I_{G_{ij}} - I_{T_{ij}}|_1}{\sigma^r_{ij}}\right). \quad (6)$$

3.3. Dealing with multiple modes

One of the degeneracies that can occur when using the probabilistic approach is a non-injective mapping due to multiple colour modes (e.g. the three skin regions – two hands and the head). For example, a point on the left hand can be mapped to either hand or the head; the model is not forced to choose correctly between them. In practice, the model cheats and maps all these modes to the one that moves the least (the head).

The key idea here is to use a cyclic loss [41, 63] and normalisation to enforce uniqueness in order to avoid this problem. If pixels are transformed from I^1_S to I^1_T and back to I^1_S, then they should end up at their original location. If they do not, then it means multiple points in one image are mapped to the same point in another.

The cyclic loss enforces that pixels should return to their original location. It is formulated as the log likelihood of the expectation that a point in the source frame will end up back at the same point at level 1 of the hierarchical model,

$$\mathcal{L}_{cyc} = \frac{\sum_{kl} - \ln(\sum_{ij}(A^1_{kl,ij}A^1_{ij,kl}))}{W_1 W_1}. \quad (7)$$

The loss is minimised when each pixel (k, l) in the source frame maps with probability 1 to a point in the target frame and that same point in the target maps with probability 1 to the original point in the source, i.e. when $A^1_{ij,kl} = A^1_{kl,ij} = 1$.

To enforce uniqueness of the pixel transformation (e.g. that not all points in the source frame are mapped to the same point in the target), we perform a normalisation step before applying the cyclic loss. Points that map to many others in either the source or the target frame are down-weighted to give $A^1_{ij,kl}$:

$$A^1_{ij,kl} = \min\left(\frac{\mathbf{A}^1_{ij,kl}}{\sum_{m,n}\mathbf{A}^1_{mn,kl}}, \frac{\mathbf{A}^1_{ij,kl}}{\sum_{m,n}\mathbf{A}^1_{ij,mn}}\right). \quad (8)$$

The matches that still have a high probability are unique in both target and source, as required.

4. Architecture and training

All self-supervised models are trained using 3 levels with the lowest resolution $R_1 = (32 \times 32)$ which is increased to resolution $R_3 = (128 \times 128)$ (as we found additional levels led to marginal improvements). They are trained with $K = 9$, $\lambda = 1$, a learning rate of 0.001 and the Adam optimizer [25]. When sampling frame pairs from the video, we sample within a distance of 50 frames from the initial frame for upper body and horses, 20 frames for full human body and the whole face track for faces.

Architecture. We use a convolutional architecture similar to that of [53]. A 256×256 image is passed through 8 convolutional layers (interleaved with leaky ReLUs and batch-normalization) to give a $256D$ embedding. The confidence and M-decoder branches have the same structure but different weights. The concatenated embeddings are passed through 7 upsampling layers (composed of a ReLU, bilinear upsampler, convolution and batch-norm) to give a 128×128 resolution result. The intermediary outputs (e.g. $\mathbf{M}^r, \sigma^r$) are obtained by taking the feature map of resolution R_r and performing a 5×5 convolution to compress the number of channels.

Curriculum training strategy. The final training objective is the sum of the confidence loss at all layers and the cyclic loss weighted by a hyperparameter λ, $\mathcal{L} = \sum_i \mathcal{L}^r_{con} + \lambda\mathcal{L}_{cyc}$.

These losses are trained in a curriculum strategy. As the predictions of the higher layers depend on those of the

lower layers, we train the lower layers to a good local minimum before training the higher layers. We start at the lowest resolution R_1 and incorporate new layers when the loss plateaus. The model can first learn a rough estimation of how to transform the source frame into the target before iteratively refining at successively higher resolutions.

5. Experiments

We apply the learning framework of Section 3 to three distinct human object classes – *upper bodies*, *faces*, and *full human bodies* – to demonstrate its utility by modelling a variety of classes with different challenges. In addition to that, we show that our framework is useful for other, non-human object classes by presenting qualitative results for horses. The question we are seeking to answer here is whether the embedding that we learn from a large set of videos for each object class has encoded useful information about pose and shape of the object.

Downstream learning setup. Given an embedding learnt using self-supervision on one of the large video datasets, a regressor is trained to map this embedding to the downstream task (e.g. landmark prediction). This regressor is trained and then evaluated on the given train and test sets of the given dataset. For the regressor we consider a linear layer or a multi-layer perceptron containing two layers. While our embedding should learn about pose and expression, there is no reason to expect that the explicit landmarks should be linearly related to the embedding (this is unlike [19], which explicitly encode landmarks in their latent representation). Note that we are *not* training our encoder/embedding but *only* this regressor.

Training datasets. The upper body embedding is trained on the Extended BBC Pose dataset [5, 37] of people signing. The face embedding is trained on the VoxCeleb2 dataset [7] consisting of faces of people being interviewed. The full body embedding is trained on the Penn Action dataset [59] of people performing sporting actions. The horse embedding is trained on the horse subset of the TigDog dataset [8]. As our task is not to perform the detection but to learn a representation of the object class, we use the crops provided by the dataset or, if this is not available, a rough crop based on the provided information.

Baselines. We compare to two baselines. The first is using our encoder with random weights; this baseline shows how well our self-supervised training improves over random initialisation. The second baseline is [53] which uses a similar proxy task and capacity but a different loss function/architecture to learn the image embedding and a bilinear sampling for the transformation. They do not use a hierarchical approach or confidence predictions. We retrain [53] on upper body pose and fully body pose datasets using the authors' code provided online.

Other methods. We also report the results of other self-supervised and supervised methods on these datasets. These approaches vary in terms of how they pre-process their training data and assumptions made about the downstream task. We give these numbers to benchmark our approach against recent progress but note that these setups are not precisely the same.

5.1. Predicting landmarks

We consider the downstream task of predicting landmarks from our learnt embedding.

Evaluation metric. In order to evaluate the landmarks on upper body and full human body, we use the PCK metric [57]. This metric reports the percentage of correct keypoints within a normalised distance of the ground truth. The normalised distance depends on the dataset. In the case of BBC Pose, we use $d = 6$ pixels as is customary on this dataset. For FLIC we use a threshold of 0.2α where α is the torso diameter [38]. For Penn Action we use a threshold of $0.2 \max(s_w, s_h)$ where s_w, s_h are the width and height of the bounding box. For faces, we report the root mean squared error normalised by the interocular distance.

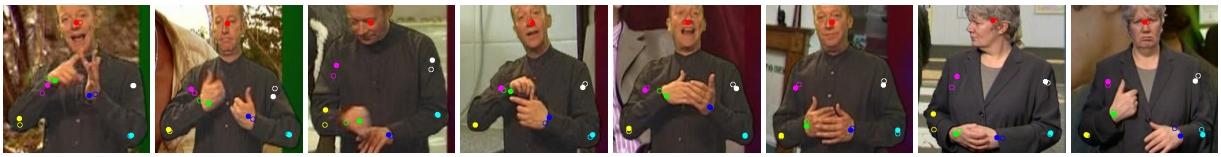
5.1.1 Upper body

We use the embedding trained on the BBC Pose dataset to predict upper body landmarks on the same dataset and on the FLIC dataset [38]. Quantitative results are discussed below, and qualitative results are shown in Fig. 3.

BBC Pose. The results on BBC Pose are given in Table 1. We first ablate our approach, demonstrating the utility of predicting confidences, and of using the cyclic loss $\mathcal{L}_{cyc}$. Each addition improves the average results and the results on the most challenging joint, the wrists. Using three levels as opposed to one improves performance, demonstrating the utility of the hierarchical approach.

In comparison to other self-supervised methods, our approach exhibits strong performance. It performs better than the baseline methods and [19], which was engineered to extract landmarks. [53] fails on this dataset due to the problem of local gradients – the movement between frames (e.g. of the hand) during training is too large, and it degenerates to predicting the identity transformation. Our approach is also better or competitive with most of the supervised methods. Clearly our embedding has indeed learned a semantically meaningful representation.

FLIC. Given that our approach outperforms the state-of-the-art on the BBC Pose dataset, we consider how well the embedding generalises to a new domain, the FLIC dataset, which consists of the upper body of people in film. The background and people are very different from the BBC



(a) **BBC**. Filled dots are GT, empty predictions.

(b) **FLIC**. Predicted poses.

Figure 3. Qualitative results on the upper body pose datasets. More examples are given in the supplementary material.

Table 1. Upper body landmark prediction on BBC Pose. Results reported are the PCK for $d < 6$. Higher is better. † denotes training with Extended BBC Pose, else with BBC Pose. The column *Loss* specifies the training losses used, $\mathcal{L}_{ph}$(ph), $\mathcal{L}_{cyc}$ (cyc) and $\mathcal{L}^{r}_{con}$ (con). r denotes the level/resolution at which training is stopped. $r = 1$ corresponds to a generated image of size 32×32, $r = 3$ to a generated image of size 128×128.

Method	Loss	Rg.	Hd	Wrt	Elb	Shldr	Avg
Ours							
r=1†	ph,cyc,con	lin	93.7	35.8	72.3	81.6	67.7
r=1†	ph,cyc,con	2 lr	94.2	51.2	78.7	82.4	74.1
r=3	ph,cyc,con	lin	**98.0**	30.7	78.9	71.3	65.6
r=3	ph,cyc,con	2 lr	96.5	41.0	82.4	73.2	69.9
r=3†	ph	2 lr	94.3	54.1	79.1	83.2	75.3
r=3†	ph,con	2 lr	96.0	58.3	**83.5**	**83.7**	78.1
r=3†	ph,cyc,con	2 lr	96.8	**62.1**	82.1	82.8	**78.7**
Self-supervised							
FAb-Net [53]†		2 lr	73.8	21.8	64.7	61.	52.9
Rand. init†		2 lr	73.2	23.2	64.5	54.7	51.1
Jakab *et al.* [19]		lin	81.1	49.1	53.1	70.1	60.7
Supervised							
Yang and Ramanan [56]			63.4	53.7	49.2	46.1	51.6
Pfister *et al.* [37]			74.9	53.1	46.0	71.4	59.4
Chen and Yuille [6]			65.9	47.9	66.5	76.8	64.1
Charles *et al.* [5]			95.4	73.9	68.7	90.3	79.9
Pfister *et al.* [36]			98.0	88.5	77.1	93.5	88.0

Pose dataset. As can be seen in Table 2, our approach generalises well to this new domain, achieving high performance. Again, using three levels as opposed to one improves performance.

5.1.2 Faces

The second class we consider is faces. As this model is trained on VoxCeleb2, which has no annotated keypoints, we test the learned embedding by predicting landmarks on a variety of other datasets. This additionally tests the embedding's generalisability.

Our embedding is used to regress landmarks on the AFLW, 300-W, and MAFL datasets and results are reported in Table 3. For AFLW, we report results on the 5-always visible landmarks (AFLW5) as well as for all 21 landmarks (AFLW21). Qualitative results are shown in Fig. 4.

Table 2. Upper body landmark prediction at PCK0.2 (as defined in [32]) on FLIC using the embedding trained on Extended BBC Pose. Higher is better. †The entire model is fine-tuned on the FLIC dataset, whereas we regress *only* two layers from the embedding.

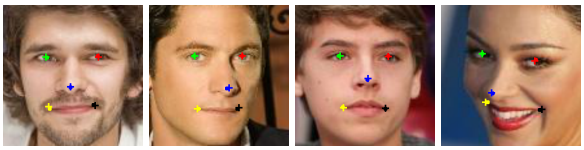
Method	Rg.	Hd	Shldr	Elb	Wrt	Avg
Ours						
r=1	2 lr	94.2	95.7	82.5	62.6	82.3
r=3	2 lr	97.2	97.1	84.8	65.2	84.5
Self-supervised						
Random init	2 lr	85.5	90.9	77.9	65.1	79.0
S&L [30]†		98.1	93.8	87.1	69.7	86.2
Supervised						
Newell *et al.* [32]		–	–	99.0	97.0	–

Our approach performs better than the baseline methods and other methods designed for predicting landmarks when trained with similar data. Our method even performs better than full frameworks trained (self-supervised or supervised) on the given dataset.

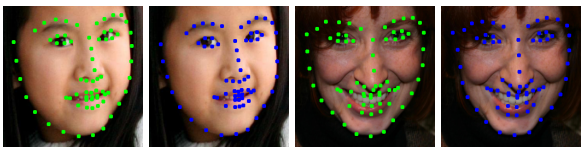
5.1.3 Full body

Finally, we test our method on full bodies using the Penn Action dataset [59]. The person may be seen from the front or back and performing a large variety of deformations which results in an extremely challenging dataset.

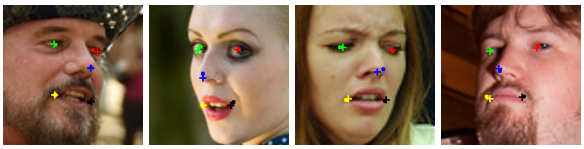
We use the learned embedding to regress landmarks. Quantitative results are reported in Table 4 and qualitative results in Fig. 5. We perform better than the baselines, and approach the performance of methods trained with deep learning on this dataset. Similarly to upper bodies, [53] degenerates to predicting the identity transformation, demonstrating the effectiveness of our method.



(a) **MAFL**. Crosses are predictions, dots GT.



(b) **300W**. Blue is GT, green predictions.



(c) **AFLW5**. Crosses are predictions, dots GT.

Figure 4. Qualitative results on the face datasets. More examples are given in the supplementary material.

Table 3. Face landmark prediction error on the 300-W and MAFL, AFLW datasets. Lower is better. † denotes trained on VoxCeleb 1/2, ‡ on VoxCeleb 1. Note that MAFL is a subset of CelebA and models trained on CelebA are fine-tuned on AFLW when reporting results on this dataset. Our embedding is never fine-tuned on these datasets; *only* the regressor is trained.

Method	Regr.	**300-W**	**MAFL**	**AFLW5**	**AFLW21**
Self-supervised					
Trained on VoxCeleb2					
Ours					
r=3	lin	4.93	3.21	6.73	**7.16**
r=1	2 lr	5.42	3.55	7.30	7.84
r=3	2 lr	**4.70**	**2.98**	**6.64**	7.28
FAb-Net [53]†	lin	5.71	3.44	7.52	8.08
Jakab *et al.* [19]‡	lin	–	3.63	6.75	–
Jakab *et al.* [20]‡	lin	5.37	–	–	–
Trained on CelebA					
Jakab *et al.* [19]	lin	–	**2.54**	**6.33**	–
Zhang *et al.* [60]	lin	–	3.16	6.58	–
Thewlis *et al.* [44]	lin	9.30	6.67	10.53	–
Thewlis *et al.* [43]	lin	7.97	5.83	8.80	–
Supervised					
MTCNN [61]		–	**5.39**	6.90	–
TCDCN [58]		5.54	–	7.65	–
RAR [54]		**4.94**	–	7.23	–

5.1.4 Non-human object classes: horses

A big advantage of our self-supervised framework is that we can get embeddings for any object class, provided we have video data to train with. To show this, we obtain a horse embedding by training on the horse subset of the TigDog dataset. We train a 2-layer regressor from the embedding to the provided keypoints. Example results can be seen in Fig. 6, more results are shown in the supplementary material.

Figure 5. Full body 2D landmarks results on the Penn Action dataset.

Table 4. Full body landmark prediction at PCK0.2 (as defined in [39]) on rhe Penn Action dataset. Higher is better.

Method	Regr.	Hd	Shldr	Elb	Wrt	Hip	Knee	Ankl	Mean
Ours									
r=1	2 lr	80.7	76.4	66.3	54.2	79.3	76.3	76.5	72.6
r=3	2 lr	**83.0**	**78.8**	**71.0**	**58.3**	**80.9**	**78.6**	**76.9**	**75.1**
Self-supervised									
FAb-Net [53]	2 lr	69.3	59.1	50.2	34.0	68.8	62.2	57.5	56.4
Random init	2 lr	70.5	60.4	50.4	35.1	70.9	63.5	53.9	56.8
Supervised									
Park and Ramanan [34]		62.8	52.0	32.3	23.3	53.3	50.2	43.0	45.3
Nie *et al.* [55]		64.2	55.4	33.8	24.4	56.4	54.1	48.0	48.0
Iqbal *et al.* [16]		89.1	86.4	73.9	72.0	85.3	79.0	80.3	81.1
Gkioxari *et al.* [13]		95.6	93.8	90.4	90.7	91.8	90.8	91.5	91.8
Song *et al.* [39]		97.6	96.8	95.2	95.1	97.0	96.8	96.9	96.4

Figure 6. 2D landmarks results on horses from the TigDog dataset.

5.2. Predicting visibility

While we have extensively investigated and demonstrated the high quality of the learned embedding by using it to regress landmarks, here we investigate whether the embedding has learned something beyond landmarks. In particular, we consider whether our embedding can be used to predict whether a landmark is or is not visible. Self-supervised methods for detecting landmarks, such as [19] cannot perform this task, as they explicitly use the landmarks in their representation.

Both the Penn Action and AFLW datasets have visibility annotations. We train a 2-layer multi-layer perceptron from the embedding to predict visibility for each landmark using a binary-cross entropy loss. We compute the area under the curve (AUC) and average over each landmark. For AFLW, we obtain 89.0 AUC and for Penn Action 77.4 AUC. A network with random initialisation achieves 63.3 AUC for PennAction and 76.6 for AFLW. This demonstrates that our method has learned something beyond just 2D positioning.

6. Conclusion

We have introduced a novel method for learning an embedding which encodes high-fidelity 2D landmarks using self-supervision on video. Because our method is self-supervised, we can incorporate an unlimited amount of data from varied domains to improve the learned embedding and only use a small set of training data in order to learn the mapping from the embedding to downstream tasks or domains. We explore further in the supplementary material how the downstream performance varies with the size of this downstream training set. We have demonstrated the method for four distinct deformable or articulated classes, but it is equally applicable to rigid classes (e.g. cars).

There are many interesting future directions. The embedding can be learnt for more animal classes and used for other downstream tasks. Also, the embedding could be extended to incorporate the temporal component implicit in the video in order to summarise multiple frames.

Acknowledgements

This work is supported by the EPSRC programme grant Seebibyte EP/M013774/1: Visual Search for the Era of Big Data.

References

[1] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA engineer*, 1984. 1

[2] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *Proc. ICCV*, 2015. 2

[3] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proc. ICML*, 2009. 3

[4] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *Proc. CVPR*, 2009. 1

[5] 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. 5, 6

[6] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NeurIPS*, 2014. 6

[7] J. S. Chung, A. Nagrani, and A. Zisserman. VoxCeleb2: Deep speaker recognition. In *INTERSPEECH*, 2018. 5

[8] L. Del Pero, S. Ricco, R. Sukthankar, and V. Ferrari. Behavior discovery and alignment of articulated object classes from unstructured video. *IJCV*, 2016. 5

[9] E. L. Denton, S. Chintala, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *NeurIPS*, 2015. 3

[10] B. Fernando, H. Bilen, E. Gavves, and S. Gould. Self-supervised video representation learning with odd-one-out networks. In *Proc. ICCV*, 2017. 2

[11] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. Van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. ICCV*, 2015. 3

[12] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Proc. ECCV*, 2016. 2

[13] G. Gkioxari, A. Toshev, and N. Jaitly. Chained predictions using convolutional neural networks. In *Proc. ECCV*, 2016. 7

[14] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proc. CVPR*, 2017. 2

[15] R. Goroshin, J. Bruna, J. Tompson, D. Eigen, and Y. LeCun. Unsupervised learning of spatiotemporally coherent metrics. In *Proc. CVPR*, 2015. 2

[16] U. Iqbal, M. Garbade, and J. Gall. Pose for action-action for pose. In *Proc. Int. Conf. Autom. Face and Gesture Recog.*, 2017. 7

[17] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Learning visual groups from co-occurrences in space and time. In *Proc. ICLR*, 2015. 2

[18] M. Jaderberg, K. Simonyan, and A. Zisserman. Spatial transformer networks. In *NeurIPS*, 2015. 4

[19] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *NeurIPS*, 2018. 2, 5, 6, 7

[20] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi. Learning landmarks from unaligned data using image translation. *arXiv preprint arXiv:1907.02055*, 2019. 7

[21] D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. In *Proc. ICCV*, 2015. 2

[22] D. Jayaraman and K. Grauman. Slow and steady feature analysis: higher order temporal coherence in video. In *Proc. CVPR*, 2016. 2

[23] L. Jing and Y. Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2018. 2

[24] D. Kim, D. Cho, and I. S. Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, 2018. 2

[25] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *Proc. ICLR*, 2014. 4

[26] R. Koch. 3-d surface reconstruction from stereoscopic image sequences. In *Proc. ICCV*, 1995. 1

[27] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proc. CVPR*, 2017. 3

[28] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE PAMI*, 2018. 3

[29] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. IJCAI*, 1981. 1

[30] I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *Proc. ECCV*, 2016. 2, 6

[31] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In *Proc. ICML*, 2009. 2

[32] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *Proc. ECCV*, 2016. 6

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

[34] D. Park and D. Ramanan. N-best maximal decoders for part

models. In *Proc. ICCV*, 2011. 7

[35] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *Proc. CVPR*, 2017. 2

[36] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *Proc. ICCV*, 2015. 6

[37] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In *Proc. ACCV*, 2014. 5, 6

[38] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Proc. CVPR*, 2013. 5

[39] J. Song, L. Wang, L. Van Gool, and O. Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *Proc. CVPR*, 2017. 7

[40] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *Proc. ICML*, 2015. 2

[41] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *Proc. ECCV*, 2010. 4

[42] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *Proc. ECCV*, pages 140–153, 2010. 2

[43] J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *NeurIPS*, 2017. 2, 7

[44] J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proc. ICCV*, 2017. 2, 7

[45] J. Thewlis, S. Zheng, P. H. S. Torr, and A. Vedaldi. Fully-trainable deep matching. In *Proc. BMVC*, 2016. 3

[46] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE PAMI*, 2008. 1

[47] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating visual representations from unlabeled video. In *Proc. CVPR*, 2016. 2

[48] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy. Tracking emerges by colorizing videos. In *Proc. ECCV*, 2018. 2, 3

[49] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *Proc. ICCV*, 2015. 2

[50] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers. A fully progressive approach to single-image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018. 3

[51] Y.-X. Wang, D. Ramanan, and M. Hebert. Growing a brain: Fine-tuning by increasing model capacity. In *Proc. CVPR*, 2017. 3

[52] D. Wei, J. Lim, A. Zisserman, and W. T. Freeman. Learning and using the arrow of time. In *Proc. CVPR*, 2018. 2

[53] O. Wiles, A. S. Koepke, and A. Zisserman. Self-supervised learning of a facial attribute embedding from video. In *Proc. BMVC*, 2018. 2, 4, 5, 6, 7

[54] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, and A. Kassim. Robust facial landmark detection via recurrent attentive-refinement networks. In *Proc. ECCV*, 2016. 7

[55] B. Xiaohan Nie, C. Xiong, and S.-C. Zhu. Joint action recognition and pose estimation from video. In *Proc. CVPR*, 2015. 7

[56] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proc. CVPR*, 2011. 6

[57] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE PAMI*, 2013. 5

[58] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *Proc. ECCV*, 2016. 7

[59] W. Zhang, M. Zhu, and K. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *Proc. ICCV*, 2013. 5, 6

[60] 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. 2, 7

[61] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *Proc. ECCV*, 2014. 7

[62] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. CVPR*, 2017. 2

[63] T. Zhou, P. Krähenbühl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3D-guided cycle consistency. In *Proc. CVPR*, 2016. 4

[64] Y. Zhou and T. L. Berg. Temporal perception and prediction in ego-centric video. In *Proc. ICCV*, 2015. 2

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	Self-supervised learning of class embeddings from video
Publication Status	Published
Publication Details	Olivia Wiles, Almut Sophia Koepke, and Andrew Zisserman: *Self-supervised learning of class embeddings from video*. International Conference on Computer Vision Conference Workshop (ICCVW), 2010.

Student Confirmation

Student Name:	ALMUT SOPHIA KOEPKE		
Contributions to the Paper	Developing ideas for improving and extending our BMVC paper. Preparing the training dataset for faces and animals and the datasets for evaluating facial landmark estimation. Running experiments for faces and animals and writing the paper.		
Signature	A. S. Koepke	Date	19/12/19

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 ANDREW ZISSERMAN		
Supervisor comments Olivia was the driving force on this project with Sophia providing support as explained in the contributions.		
Signature	Date	19/12/19

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

Bibliography

T. Afouras, J. S. Chung, and A. Zisserman. The conversation: Deep audio-visual speech enhancement. In *INTERSPEECH*, 2018.

P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *Proc. ICCV*, 2015.

M. Akbari and H. Cheng. Clavision: visual automatic piano music transcription. In *NIME*, 2015.

M. Akbari, J. Liang, and H. Cheng. A real-time system for online learning-based visual transcription of piano music. *Multimedia Tools and Applications*, 77(19), 2018.

R. Arandjelović and A. Zisserman. Look, listen and learn. In *Proc. ICCV*, 2017.

R. Arandjelović and A. Zisserman. Objects that sound. In *Proc. ECCV*, 2018.

Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *NeurIPS*, 2016.

H. B. Barlow et al. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1:217–234, 1961.

H. B. Barlow. Unsupervised learning. *Neural computation*, 1(3):295–311, 1989.

A. Bas, W. A. P. Smith, M. Awais, and J. Kittler. 3D morphable models as spatial transformer networks. In *ICCW Workshop*, 2017.

A. Bazzica, J. van Gemert, C. C. Liem, and A. Hanjalic. Vision-based detection of acoustic timed events: a case study on clarinet note onsets. *arXiv preprint arXiv:1706.09556*, 2017.

S. Becker. Learning to categorize objects using temporal coherence. In *NeurIPS*, 1993.

G. Bertasius, C. Feichtenhofer, D. Tran, J. Shi, and L. Torresani. Learning temporal pose estimation from sparsely-labeled videos. *arXiv preprint arXiv:1906.04016*, 2019.

P. Bertelson and M. Radeau. Cross-modal bias and perceptual fusion with auditory-visual spatial discordance. *Perception & Psychophysics*, 29(6):578–584, 1981.

V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proc. ACM SIGGRAPH*, 1999.

A. Chaigne and J. Kergomard. *Acoustics of musical instruments*. Springer, 2016.

L. Chen, S. Srivastava, Z. Duan, and C. Xu. Deep cross-modal audio-visual generation. In *Proceedings of the on Thematic Workshops of ACM Multimedia*, 2017.

L. Chen, R. K. Maddox, Z. Duan, and C. Xu. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *Proc. CVPR*, 2019.

E. C. Cherry. Some experiments on the recognition of speech, with one and with two ears. *The Journal of the Acoustical Society of America*, 25(5):975–979, 1953.

S. Chopra, R. Hadsell, Y. LeCun, et al. Learning a similarity metric discriminatively, with application to face verification. In *Proc. CVPR*, 2005.

J. S. Chung and A. Zisserman. Out of time: automated lip sync in the wild. In *ACCV Workshop on Multi-view Lip-reading*, 2016.

J. S. Chung and A. Zisserman. Lip reading in the wild. In *Proc. ACCV*, 2016.

S.-W. Chung, J. S. Chung, and H.-G. Kang. Perfect match: Improved cross-modal embeddings for audio-visual synchronisation. In *Proc. ICASSP*, 2019.

J. W. Davidson. Visual perception of performance manner in the movements of solo musicians. *Psychology of Music*, 21(2):103–113, 1993.

A. Davis, M. Rubinstein, N. Wadhwa, G. Mysore, F. Durand, and W. T. Freeman. The visual microphone: Passive recovery of sound from video. In *Proc. ACM SIGGRAPH*, 2014.

V. R. de Sa. Learning classification with unlabeled data. In *NeurIPS*, 1994.

V. R. de Sa. Minimizing disagreement for self-supervised classification. In *Proceedings of the 1993 Connectionist Models Summer School*, page 300. Psychology Press, 1994.

S. S. Deb and A. Rajwade. An image analysis approach for transcription of music played on keyboard-like instruments. In *Proc. ICVGIP*, 2016.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009.

E. L. Denton and V. Birodkar. Unsupervised learning of disentangled representations from video. In *NeurIPS*, 2017.

B. Dodd. The role of vision in the perception of speech. *Perception*, 6(1):31–40, 1977.

C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proc. ICCV*, 2015.

M. C. Doukas, V. Sharmanska, and S. Zafeiriou. Video-to-video translation for visual speech synthesis. *arXiv preprint arXiv:1905.12043*, 2019.

D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman. Temporal cycle-consistency learning. In *Proc. CVPR*, 2019.

A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *Proc. ACM SIGGRAPH*, 2018.

N. P. Erber. Interaction of audition and vision in the recognition of oral speech stimuli. *Journal of Speech and Hearing Research*, 12(2):423–425, 1969.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2007.

P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.

B. Fernando, H. Bilen, E. Gavves, and S. Gould. Self-supervised video representation learning with odd-one-out networks. In *Proc. CVPR*, 2017.

C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *NeurIPS*, 2016.

P. Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991.

O. Fried, A. Tewari, M. Zollhöfer, A. Finkelstein, E. Shechtman, D. B. Goldman, K. Genova, Z. Jin, C. Theobalt, and M. Agrawala. Text-based editing of talking-head video. *arXiv preprint arXiv:1906.01524*, 2019.

R. Gao and K. Grauman. Co-separating sounds of visual objects. In *Proc. ICCV*, 2019.

R. Gao, R. Feris, and K. Grauman. Learning to separate object sounds by watching unlabeled video. In *Proc. ECCV*, 2018.

Z. Geng, C. Cao, and S. Tulyakov. 3D guided fine-grained face manipulation. In *Proc. CVPR*, 2019.

S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *Proc. ICLR*, 2018.

N. J. Giordano. *Physics of the Piano*. Oxford University Press, 2010.

S. Goldstein and Y. Moses. Guitar music transcription from silent video. In *Proc. BMVC.*, 2018.

E. Gómez Gutiérrez, P. Arias Martínez, P. Zinemanas, and G. Haro Ortega. Visual music transcription of clarinet video recordings trained with audio-based labelled data. In *ICCV workshop*, 2017.

D. O. Gorodnichy and A. Yogeswaran. Detection and tracking of pianist hands and fingers. In *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*, pages 63–63. IEEE, 2006.

A. Grigorev, A. Sevastopolsky, A. Vakhitov, and V. Lempitsky. Coordinate-based texture inpainting for pose-guided human image generation. In *Proc. CVPR*, 2019.

K. Gu, Y. Zhou, and T. Huang. Flnet: Landmark driven fetching and learning network for faithful talking facial animation synthesis. *arXiv preprint arXiv:1911.09224*, 2019.

T. Han, W. Xie, and A. Zisserman. Video representation learning by dense predictive coding. In *ICCV workshop*, 2019.

W. Hao, Z. Zhang, and H. Guan. Cmcgan: A uniform framework for cross-modal visual-audio mutual generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016.

K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. CVPR*, 2020.

O. J. Hénaff, A. Razavi, C. Doersch, S. Eslami, and A. v. d. Oord. Data-efficient image recognition with contrastive predictive coding. In *Proc. ICLR*, 2020.

J. R. Hershey and J. R. Movellan. Audio vision: Using audio-visual synchrony to locate sounds. In *NeurIPS*, 2000.

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

M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NeurIPS*, 2015.

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*, pages 4016–4027, 2018.

T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi. Learning human pose from unaligned data through image translation. In *CVPR workshop*, 2019.

A. Jha, V. Namboodiri, and C. V. Jawahar. Word spotting in silent lip videos. In *Proc. WACV*, 2018.

X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool. Dynamic filter networks. In *NeurIPS*, 2016.

W. Jiang, W. Sun, A. Tagliasacchi, E. Trulls, and K. M. Yi. Linearized multi-sampling for differentiable image transformation. In *Proc. ICCV*, 2019.

L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *arXiv preprint arXiv:1902.06162*, 2019.

S. P. Johnson, D. Amso, and J. A. Slemmer. Development of object concepts in infancy: Evidence for early learning in an eye-tracking paradigm. *Proceedings of the National Academy of Sciences*, 100(18):10568–10573, 2003.

E. Kidron, Y. Y. Schechner, and M. Elad. Pixels that sound. In *Proc. CVPR*, 2005.

E. Kidron, Y. Y. Schechner, and M. Elad. Cross-modal localization via sparsity. *IEEE Transactions on Signal Processing*, 55(4):1390–1404, 2007.

J. W. Kim, J. Salamon, P. Li, and J. P. Bello. Crepe: A convolutional representation for pitch estimation. In *Proc. ICASSP*, 2018.

H. Kim, M. Elgharib, M. Zollhöfer, H.-P. Seidel, T. Beeler, C. Richardt, and C. Theobalt. Neural style-preserving visual dubbing. *ACM Transactions on Graphics (TOG)*, 38(6):178, 2019.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *Proc. ICLR*, 2015.

A. S. Koepke, O. Wiles, and A. Zisserman. Visual pitch estimation. In *Proc. Sound and Music Computing (SMC)*, 2019.

A. S. Koepke, O. Wiles, Y. Moses, and A. Zisserman. Sight to sound: An end-to-end approach for visual piano transcription. In *Proc. ICASSP*, 2020.

B. Korbar, D. Tran, and L. Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018.

P. K. Kuhl and A. N. Meltzoff. The bimodal perception of speech in infancy. *Science*, 218(4577):1138–1141, 1982.

H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang. Unsupervised representation learning by sorting sequences. In *Proc. ICCV*, 2017.

J. Lee, B. Doosti, Y. Gu, D. Cartledge, D. Crandall, and C. Raphael. Observing pianist accuracy and form with computer vision. In *Proc. WACV*, 2019.

B. Li, K. Dinesh, Z. Duan, and G. Sharma. See and listen: Score-informed association of sound tracks to players in chamber music performance videos. In *Proc. ICASSP*, 2017.

B. Li, C. Xu, and Z. Duan. Audiovisual source association for string ensembles through multi-modal vibrato analysis. *Proc. Sound and Music Computing (SMC)*, 2017.

B. Li, A. Maezawa, and Z. Duan. Skeleton plays piano: Online generation of pianist body movements from midi performance. In *ISMIR*, 2018.

B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma. Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications. *IEEE Transactions on Multimedia*, 2019.

Y. Li, J. Zeng, S. Shan, and X. Chen. Self-supervised representation learning from videos for facial action unit detection. In *Proc. CVPR*, 2019.

H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264(5588):746, 1976.

A. N. Meltzoff. Infants' causal learning: Intervention, observation, imitation. *Causal learning: Psychology, philosophy, and computation*, 2007.

I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *Proc. ECCV*, 2016.

T. Nazzi and A. Gopnik. Linguistic and cognitive abilities in infancy: When does language become a tool for categorization? *Cognition*, 80(3):B11–B20, 2001.

A. Nisbet and R. Green. Capture of dynamic piano performance with depth vision. `https://albertnis.com/resources/2017-05-10-piano-vision/Nisbet_Green_Capture_of_Dynamic_Piano%20Performance_with_Depth_Vision.pdf`, 2017. [Online; accessed 12-Feb-2020].

M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. ECCV*, 2016.

D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi. Self-supervised learning of geometrically stable features through probabilistic introspection. In *Proc. CVPR*, 2018.

A. Oka and M. Hashimoto. Marker-less piano fingering recognition using sequential depth images. In *The 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision*, pages 1–4. IEEE, 2013.

B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311–3325, 1997.

OUP. Lexico: Oxford English Dictionary, Oxford University Press, 2019.

A. Owens and A. A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proc. ECCV*, 2018.

A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman. Visually indicated sounds. In *Proc. CVPR*, 2016.

A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba. Ambient sound provides supervision for visual learning. In *Proc. ECCV*, 2016.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proc. CVPR*, 2016.

V. Pătrăucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *NeurIPS*, 2016.

A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proc. ECCV*, 2018.

S. Qian, K.-Y. Lin, W. Wu, Y. Liu, Q. Wang, F. Shen, C. Qian, and R. He. Make a face: Towards arbitrary high fidelity face manipulation. In *Proc. ICCV*, 2019.

S. Rho, J.-I. Hwang, and J. Kim. Automatic piano tutoring system using consumer-level depth camera. In *International Conference on Consumer Electronics (ICCE)*. IEEE, 2014.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. MICCAI*, 2015.

A. Rouditchenko, H. Zhao, C. Gan, J. McDermott, and A. Torralba. Self-supervised audio-visual co-segmentation. In *Proc. ICASSP*, 2019.

A. Senocak, T.-H. Oh, J. Kim, M.-H. Yang, and I. So Kweon. Learning to localize sound source in visual scenes. In *Proc. CVPR*, 2018.

E. Shlizerman, L. Dery, H. Schoen, and I. Kemelmacher-Shlizerman. Audio to body dynamics. In *Proc. CVPR*, 2018.

A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe. Animating arbitrary objects via deep motion transfer. In *Proc. CVPR*, 2019.

A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe. First order motion model for image animation. In *NeurIPS*, 2019.

K. Songsri-in and S. Zafeiriou. Face video generation from a single image and landmarks. *arXiv preprint arXiv:1904.11521*, 2019.

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

W. H. Sumby and I. Pollack. Visual contribution to speech intelligibility in noise. *The Journal of the Acoustical Society of America*, 26(2):212–215, 1954.

Q. Summerfield. Use of visual information for phonetic perception. *Phonetica*, 36(4-5):314–331, 1979.

P. Suteparuk. Detection of piano keys pressed in video. *Dept. of Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep*, 2014.

A. Tewari, M. Zollhöfer, H. Kim, P. Garrido, F. Bernard, P. Perez, and C. Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proc. ICCV*, 2017.

J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *NeurIPS*, 2017.

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

J. Thewlis, S. Albanie, H. Bilen, and A. Vedaldi. Unsupervised learning of landmarks by descriptor vector exchange. In *Proc. ICCV*, 2019.

G. J. Thomas. Experimental study of the influence of vision on sound localization. *Journal of Experimental Psychology*, 28(2):163, 1941.

B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

S. Tripathy, J. Kannala, and E. Rahtu. Icface: Interpretable and controllable face reenactment using gans. *arXiv preprint arXiv:1904.01909*, 2019.

E. Ververas and S. Zafeiriou. Slidergan: Synthesizing expressive face images by sliding 3D blendshape parameters. *arXiv preprint arXiv:1908.09638*, 2019.

R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017.

X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *Proc. ICCV*, 2015.

X. Wang, A. Jabri, and A. A. Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019.

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

O. Wiles, A. S. Koepke, and A. Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proc. ECCV*, 2018.

O. Wiles, A. S. Koepke, and A. Zisserman. Self-supervised learning of class embeddings from video. In *ICCV workshop*, 2019.

D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proc. CVPR*, 2019.

X. Xu, B. Dai, and D. Lin. Recursive visual sound separation using minus-plus net. In *Proc. ICCV*, 2019.

T. Xue, J. Wu, K. Bouman, and B. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NeurIPS*, 2016.

K. Yamamoto, E. Ueda, T. Suenaga, K. Takemura, J. Takamatsu, and T. Ogasawara. Generating natural hand motion in playing a piano. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems*, 2010.

B. P. Yuhas, M. H. Goldstein, and T. J. Sejnowski. Integration of acoustic and visual speech signals using neural networks. *IEEE Communications Magazine*, 27(11):65–71, 1989.

E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky. Few-shot adversarial learning of realistic neural talking head models. *arXiv preprint arXiv:1905.08233*, 2019.

B. Zhang, J. Zhu, Y. Wang, and W. K. Leow. Visual analysis of fingering for pedagogical violin transcription. In *Proc. ACMM*, 2007.

R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *Proc. ECCV*, 2016.

R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proc. CVPR*, 2017.

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.

H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba. The sound of pixels. In *Proc. ECCV*, 2018.

H. Zhao, C. Gan, W.-C. Ma, and A. Torralba. The sound of motions. In *Proc. ICCV*, 2019.

Y. Zhou, Z. Wang, C. Fang, T. Bui, and T. L. Berg. Visual to sound: Generating natural sound for videos in the wild. In *Proc. CVPR*, 2018.

H. Zhou, Y. Liu, Z. Liu, P. Luo, and X. Wang. Talking face generation by adversarially disentangled audio-visual representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

H. Zhou, Z. Liu, X. Xu, P. Luo, and X. Wang. Vision-infused deep audio inpainting. In *Proc. ICCV*, 2019.

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.