Objects from Motion

James Thewlis
Worcester College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Supervised by Prof. Andrea Vedaldi

Michaelmas 2018

Acknowledgements

I am immensely grateful to my supervisor Andrea Vedaldi for his consistent support and guidance throughout my studies. His drive and intuition for developing new ideas, and encouragement to push forward despite setbacks, can be seen throughout this thesis.

I thank all involved in the EPSRC AIMS CDT, which has funded my studies, in particular founding directors Stephen Roberts and Niki Trigoni for making this program possible and successful, and administrator Wendy Poole whose assistance throughout the CDT has been invaluable.

I thank my collaborators Hakan Bilen, who has contributed to many of the ideas in this thesis, Aravindh Mahendran who spearheaded the work on learning from flow, and Shuai Zheng and Phil Torr who made my first paper possible.

I thank Andrew Zisserman and Victor Prisacariu for valuable feedback in my transfer and confirmation of status, and Iasonas Kokkinos and Natalia Neverova for guidance during my time at FAIR.

I am thankful to all members of the VGG during my time here, it has been a privilege to work with talented people from all over the world, and I very much appreciate those who have put in the extra effort to organise social events.

My time at Oxford has been brightened thanks to the support and friendship of Hillary Shakespeare.

I thank Asal Khoshbin for her caring and patience especially during deadline periods.

I thank my family for their infinite support.

Abstract

This thesis tackles the challenge of learning the abstract structure of object categories without manual supervision. We show that we can learn useful representations based on the motion of objects in videos and even from collections of static images through the use of synthetic warps. An important contribution of this work is the notion of an *Object Frame*, an object-centric frame of reference which can be learned from motion. Objects that appear in images can be affected by complex nuisance factors such as viewpoint changes and deformations, yet our method manages to factorize out these variations and semantically map objects to a common coordinate frame. Importantly, this mapping also works across different object instances despite only being trained on instance-specific correspondences. Two implementations of the *Object Frame* idea are presented. The first learns a sparse, landmark-based representation of structure, simultaneously discovering which landmarks are useful and learning to predict their locations consistently across instances. The second is a dense approach which maps image pixels to a canonical spherical coordinate frame in a semantically consistent manner. We show that the latter formulation has applications in discovering the symmetries of deformable objects, and also explore the relationship between our *Object Frame* and generic, higher dimensional feature descriptors. We also present a trainable method to compute dense matches, and a state of the art self-supervised learning method using optical flow similarity to compute pixel embeddings.

Contents

Chapter 1

Introduction

Many areas in computer vision have recently seen advances thanks to renewed interest in Convolutional Neural Networks (CNNs) [56], initially with image classification [52]. CNNs were subsequently applied to other tasks, such as object detection [35], segmentation [41, 16] and optical flow [31]. CNNs typically require vast amounts of labelled data, in contrast to the ease with which humans learn through observation and interaction with their environment. We propose to use motion information from moving scenes and synthetically warped images to learn the properties of objects, in particular how to consistently label points on an object despite changes in pose and viewpoint. Furthermore, we wish for the labelling to be valid across instances such that, for example, the left wing mirror of two different cars would be given the same label.

The end product is a canonical labelling of points on the surface objects, such that a point on one object instance can be spatially labelled and put into correspondence with the same point on another instance of the same object category.

The rich feedback loop at the disposition of a human is undoubtedly a far cry above the rudimentary mass of labelled images that a machine must use to learn vision. Interacting with a dynamic world and manipulating objects is clearly a huge source of perceptual data. It would be a hard problem to simulate all this, however what we do have available in large quantities is video data. Humans do not perceive the world as single images – cues about depth and shape are derived from motion, and a model trained on isolated images would be deprived of this context. We want to find out how much information can be obtained by a computer vision system observing moving scenes containing camera motion and dynamic objects, with the goal of learning objects from motion.

Initial work focused on optical flow and quasi-dense matching in Chapter 3, which poses the Deep Matching algorithm as an end-to-end trainable CNN.

In Chapters 4 and 5 we then use flow, either predicted between video frames or generated through synthetic warps, as the supervisory signal for how points on objects correspond across images, developing concrete realisations of the idea of an *Object Frame*. Our formulations involve training a CNN in an unsupervised way to produce a representation where semantically analogous pixels from different images are put into correspondence. We explore two different representations. The first in Chapter 4 is based around the idea of identifying and predicting sparse landmarks. The second in Chapter 5 outputs a dense per-pixel descriptor where the distance between descriptors is minimised for points that are in correspondence. However, instead of learning a high dimensional descriptor, we learn a labelling that associates points on the surface of the object with points on a sphere, thereby mapping objects to the same label space across viewpoints, deformations and instances. This notion of learning a coordinate frame specific to an object category, or *Object Frame*, turns out to have useful properties for discovering and evaluating the symmetry of deformable objects, as explored in Chapter 6. We also explore the relationship between our learned "semantic" coordinates and general visual descriptors in Chapter 7, by increasing the descriptor dimensionality while simultaneously compressing the embedding space. Finally, we consider the task of learning an embedding that is consistent with an optical flow field Chapter 8, and show its use as a pre-training task for object detection and segmentation.

1.1 Key Ideas

The main thread running through the projects undertaken is the idea of *Objects from Motion*, *i.e.* learning information about object classes from motion sequences, with the main supervisory signal being the way in which those objects move and deform, as estimated through optical flow or even synthetic warps, rather than requiring explicit labelling.

Although learnt from motion, models trained this way can then be used to predict properties of objects in still images.

This section looks at some key ideas used for taking advantage of this signal and arriving at the concept of an *Object Frame*.

1.1.1 Canonical Coordinate Frames

The initial concept of *Objects from Motion* was a method to consistently and densely identify points of interest on different objects of the same category, relating the pixels

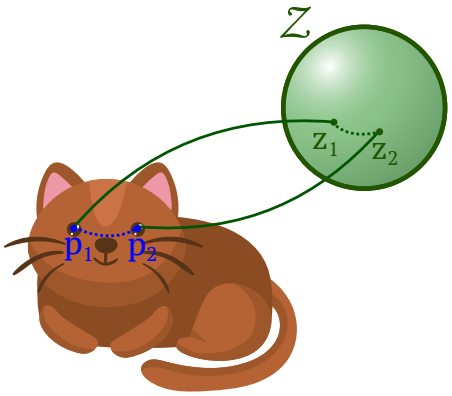


Figure 1.1: Points on the surface of the cat can be semantically related to an underlying space, in this case a sphere.

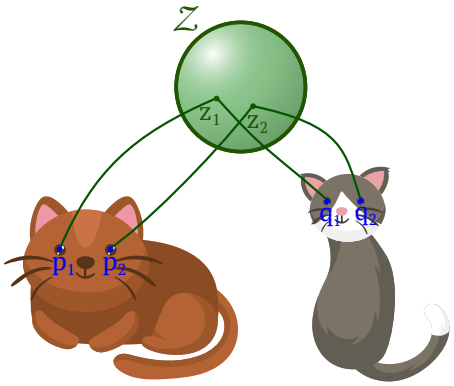


Figure 1.2: By mapping to a canonical underlying label space, points on two different cat instances can be put into correspondence.

of the object to an underlying coordinate system (Figure 1.1) that varies consistently with position on the surface of the object. This would mean, for example, that the left eye of a cat would always be associated with the same label (Figure 1.2), where the label space $\mathcal{Z}$ could be a topological sphere, $\mathcal{Z} = \mathbb{S}^2$, under the assumption that the 3D structure of the cat is homeomorphic to a sphere. If modelled perfectly this would give coverage of the 3D layout of the cat without discontinuities.

We can give a justification for our argument using the formalisms of topology, illustrated in Figure 1.4. This represents an idealised version of our problem where everything is smooth and well behaved, ignoring the complexities of describing arbitrary objects by learning from real world noisy data.

Imagine describing the exterior of an object instance as a smooth surface embedded in Euclidean 3D space, $S \subset \mathbb{R}^3$. Assuming the surface has no "holes", it is

homeomorphic to the sphere, $S \cong \mathbb{S}^2$ which is to say that we can establish a continuous mapping between the two whose inverse is also continuous, i.e. they are topologically identical.

As explained further in Section 1.1.3, we can construct a homeomorphism $p = \pi_S(q)$ mapping points of the sphere $q \in \mathcal{Z}$ to points $p \in S$ of the objects. We can also assume that these mappings are *semantically consistent*, meaning that $\pi_{S'} \circ \pi_S^{-1} : S \to S'$ maps points of object surface S to semantically-analogous points on object surface S'.

This abstract construction shows that we can endow an object category with a canonical, and indeed spherical, reference system $\mathcal{Z}$.

This gives the basic idea of how we might train a CNN to assign some z to each point of the object in the photo, with a loss that minimises the distance between corresponding points in different photos as in Figure 1.3.

This takes the form of a labelling function[1] $\Psi : (\mathbf{x}; u) \to z$ that takes an RGB image $\mathbf{x} : \Lambda \to \mathbb{R}^3$, $\Lambda \subset \mathbb{R}^2$ and a pixel $u \in \Lambda$ to the object point $z \in \mathcal{Z}$.

We can then semantically navigate the visible surface of the object in the photo without needing to consider the true 3D layout. We explore concrete implementations of this idea in Chapters 4 and 5. Chapter 4 employs a discrete system of object landmarks $z_k \in \mathcal{Z}$, whereas Chapter 5 explicitly learn a dense mapping to a spherical coordinate space.

The benefits of a system providing such a labelling can be seen by examining how it would incorporate and combine ideas expressed in the literature review. The work of [68] perhaps comes closest to the idea of spatially indexing an object category using a common coordinate system, illustrating the application to scene alignment. However they explicitly compute 2D deformation maps from the learnt base pose, updating them iteratively during optimisation. We seek to learn the function that maps the object to its label space using the machinery of a CNN. The learnt labelling could also be seen as a dense set of keypoints, giving similar benefits to where keypoints have been used in vision and graphics for faces and human poses [84]. The benefit would extend to arbitrary objects without the need for manually labelling many images, since relevant keypoints for a certain task could be identified post-training on a few reference images. As suggested in FlowWeb [115], a large jointly registered set of images could aid in areas such as edit propagation, co-segmentation, 3D modelling,

[1]Since the published papers reproduced in Chapters 4 and 5 use incompatible notations, in this introduction we are using a compromise notation that matches neither exactly. Consult the method section in each chapter to understand the notation used in each case.

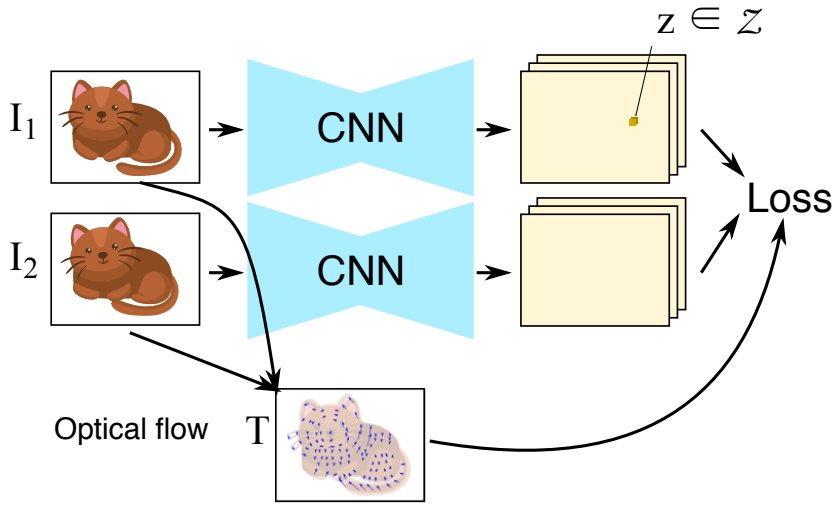


Figure 1.3: A sketch of the training procedure. Two frames related by optical flow are fed through the CNN, which gives a labelling in terms of canonical coordinates. The loss minimises the distance on the sphere between points matched by the flow.

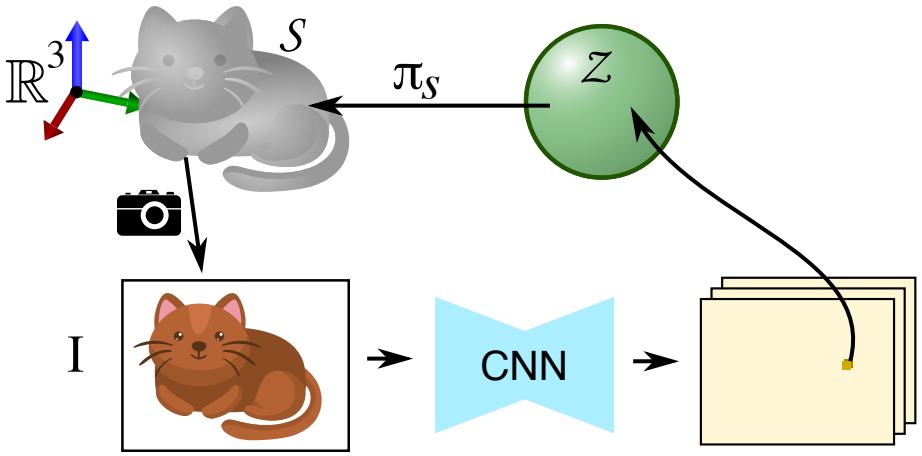


Figure 1.4: Conceptually closing the loop between the spaces involved. Through our CNN we obtain coordinates which relate points on the image to a canonical labelling on the sphere. There exists an (unseen) mapping between the sphere and the surface of the object, and a transformation (for which we do not know the parameters) between the model as it exists in the world and the image.

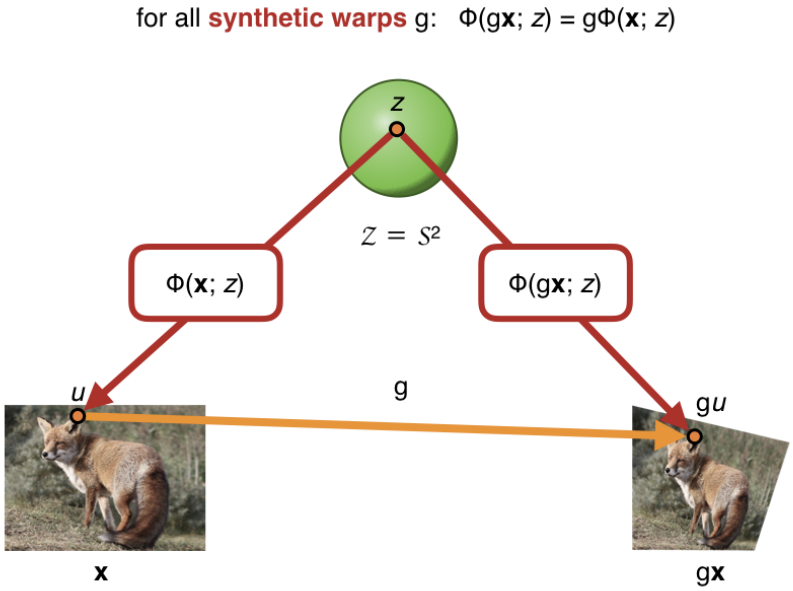


Figure 1.5: **Equivariance.** The learned function from canonical coordinate space $\mathcal{Z}$ to image coordinates is desired to be equivariant with respect to warps g.

3D reconstruction and browsing. The efficiency gains from putting many images into correspondence as in [50], avoiding pairwise flow calculations would also be considerable. Our method is related to learning a descriptor from video in the style of [98], but with a very low dimensional feature. The aim is that the bottleneck forces the network into becoming invariant to object instances, focusing only on the underlying class.

1.1.2 Equivariance

A function f is considered equivariant to another function t if $f(t(x)) = t(f(x))$. That is to say that, if the input changes under some transformation t, the output changes in the same way. Perhaps the most well known example in deep learning is the equivariance of the convolutional layer to translations [36].

It is easy to see how this constraint is useful when it comes to our desired goal of learning a canonical labelling of objects using motion, since the labelling of an object in motion would be expected to behave equivariantly according to that motion. If a

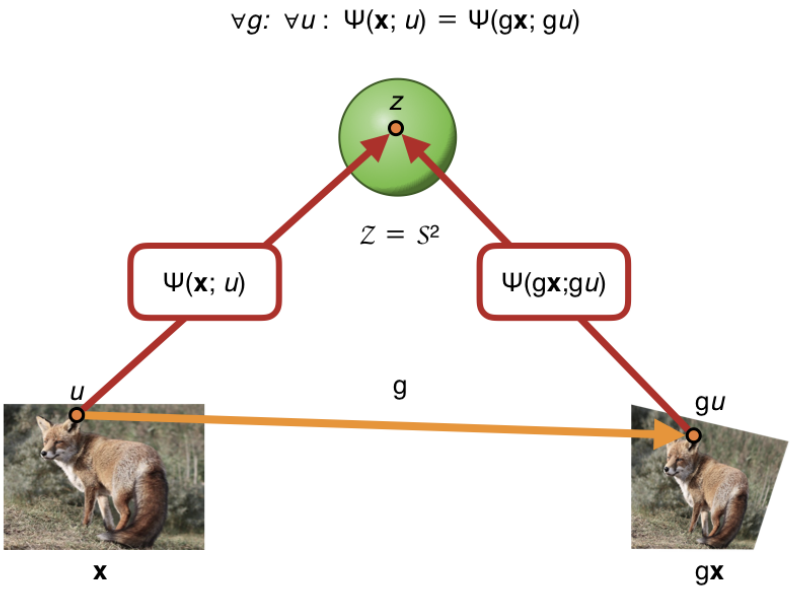


Figure 1.6: **Invariance.** The constraint used can also be seen through the lens of invariance. When mapping from image pixels to canonical coordinate space $\mathcal{Z}$, a semantic point (such as an ear) should live at the *same* position z invariantly to image deformation.

video contains a hand moving up and to the right, whatever labelling system we have used to identify the part "right hand" semantically should also move up and to the right when indexed through the image domain. The equivariance constraint used by our method was introduced by Lenc and Vedaldi in [57] to learn SIFT-like features for image matching.

Since in our setup we assume the presence of two frames and the flow between them, it is easy to incorporate equivariance in our learning formulation, ensuring that our unsupervised learnt mapping is compatible with image deformations.

Letting $\Phi(\mathbf{x}; z)$ be the function that maps from a canonical coordinate z to the image domain for a specific image $\mathbf{x}$, consider a deformation of the image domain $g : \Lambda \to \Lambda$, given by some motion or synthetic warp. As a notational convenience, let $g\mathbf{x} = \mathbf{x} \circ g^{-1}$ be the action of g on the image (obtained by inverse warp).

Pixel $u = \Phi(g\mathbf{x}; z)$ in image $g\mathbf{x}$ must correspond to the same label z as pixel gu in image $\mathbf{x}$, which results in the *equivariance constraint*, as illustrated in Figure 1.5:

$$\forall z \in \mathcal{Z} : \ \Phi(g\mathbf{x}; z) = g(\Phi(\mathbf{x}; z)). \tag{1.1}$$

Equivariance is in some sense a minimal learning principle for our task – rather than defining a complicated set of heuristics, one of the simplest properties that we would expect of a landmark is that, as the object moves, the landmark should move with it in a manner consistent to viewpoint shifts and deformations. Hence equivariance is a very natural fit for solving the problem.

This is the formulation as used in Chapter 4. The constraint can also be seen as an *invariance* when considering the inverse function Ψ from image coordinate to $\mathcal{Z}$, as described in Figure 1.6 and Chapter 5.

1.1.3 Viewpoint Factorization

Imagine an idealized world where all variation present in images of a certain object category can be explained by three factors of variation: viewpoint shifts, object deformation and changes of object instance. Our goal is to factorize out these nuisance factors to obtain a reference frame that is intrinsic to the object category itself. The idea, also present in the work of Novotny *et al.* [70], is to do this without any absolute reference frame given as ground truth in any image. We are only given tuples $(\mathbf{x}, \mathbf{x}', g)$ consisting of an image pair and the relative warp between them.

We can lean on the equivariance constraint Equation (1.1), which implies the existence of some object-centric space $\mathcal{Z}$. As described in Chapter 4, implementing

Φ as a neural network ought to, in an ideal case, learn to bridge automatically across different viewpoints of the same object. This should also extend to deformations of the same object, since (assuming our surface has genus 0) for any surface S that may arise from the family of possible deformations we can introduce a common reference space, such as a sphere, by constructing a homeomorphism $p = \pi_S(q)$ mapping points of the sphere $q \in \mathcal{Z} = \mathbb{S}^2$ to points $p \in S$ of the objects (Figure 1.7, Figure 1.4).

When it comes to different object instances, there is no geometrical reason to suppose that different objects will be mapped in a semantically consistent manner. In theory a completely different mapping could be learned in each case. However, an important finding in Chapters 4 and 5 is that, by sharing the same model Φ and training with many different object instances, semantically meaningful correspondences emerge in practice, mapping for example the left eye of any image of a cat to the same point in $\mathcal{Z}$.

1.2 Object Frame

Fundamental to the approach in question is the concept of an *Object Frame*, an idea introduced in Chapter 4 and further developed in Chapter 5.

Consider an object represented as a 3D surface $S \subset \mathbb{R}^3$, and more abstractly a class of such objects that are homeomorphic to some reference surface $\mathcal{Z}$, typically a sphere $\mathcal{Z} = \mathbb{S}^2$, as in Figure 1.7.

1.2.1 Sparse Landmarks

One way of making concrete the abstract idea of an object frame is to define it in terms of sparse landmarks. Here, a discrete set of object landmarks is represented by a finite set of points $z_k \in \mathcal{Z}$.

The problem to be solved now becomes finding a function $\Phi(\mathbf{x}; z_k)$ that can take an image $\mathbf{x}$ and one of these points, and return its location in the image, as shown in Figure 1.8.

Fortunately, CNNs have proven to be well suited to the task of landmark detection. This is the setup considered in Chapter 4, with a CNN that outputs a series of heatmaps giving the location in the image of each landmark. However, since we do not have labelled ground truth on which semantic point each z_k should represent, this also needs to be learned. In Chapter 4 this is done by taking advantage of the equivariance constraint along with a diversity term to prevent collapse.

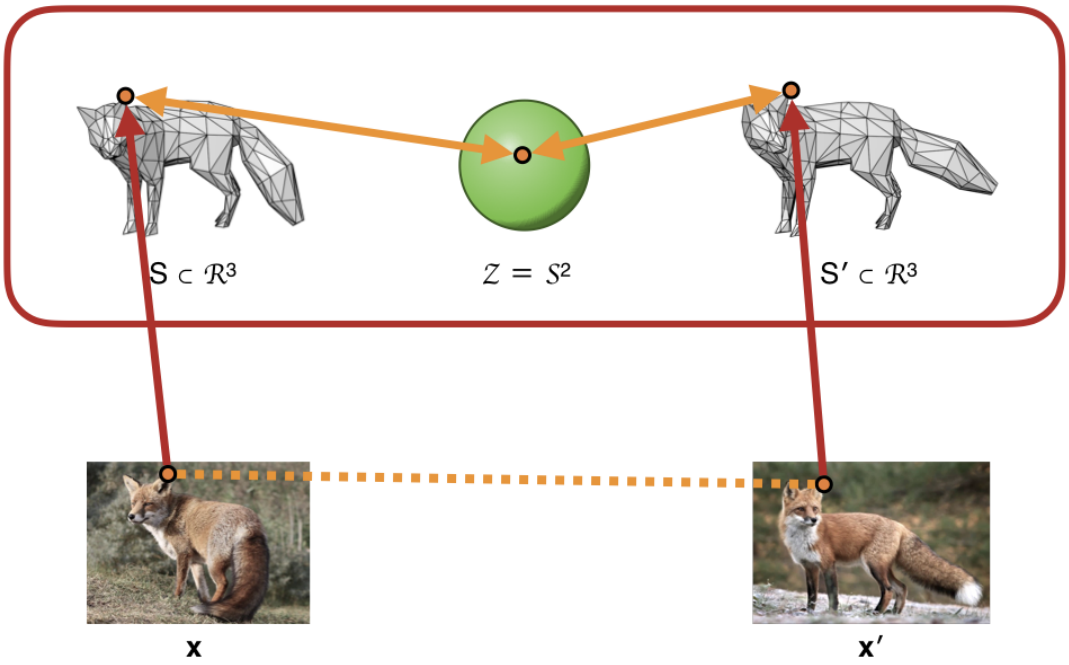


Figure 1.7: Homeomorphism with a sphere

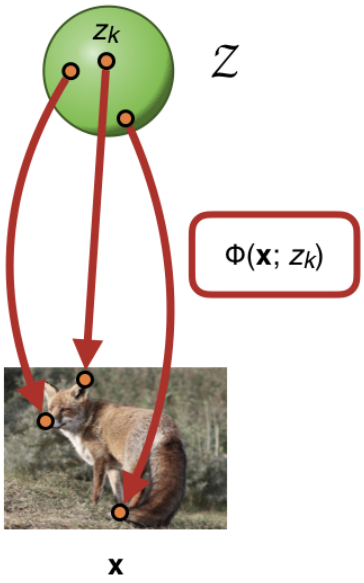


Figure 1.8: **Sparse Landmarks.** This task involves learning a function which maps a discrete set of landmarks $z_1 ... z_n$ to their location in the image. Crucially, neither the meaning of the landmarks nor their ground truth position are given in advance.

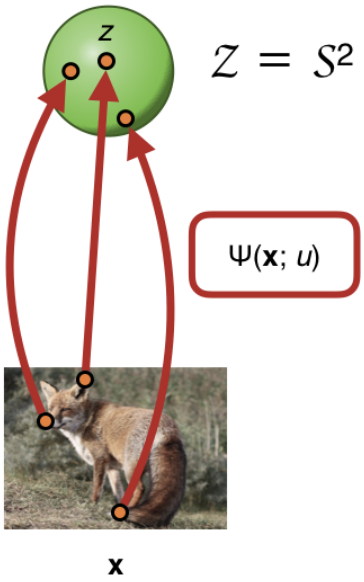


Figure 1.9: **Dense Landmarks.** In this task we learn a function which maps pixel locations u to their position on a sphere. We want this mapping to be semantically consistent across different object instances, without being trained on cross-instance correspondences. This is the inverse function of Figure 1.8.

1.2.2 Dense Mapping

The Sparse Landmark setting can be seen as a particular realization of the idea of an *Object Frame*, however we can obtain a more faithful embodiment of a theoretical *Object Frame* by capturing a dense space intrinsic to the object.

Whereas in the Sparse Landmark setting we sampled the function Φ at a sparse set of points by posing the problem as one of landmark detection, here we are essentially learning the function $\Psi(\mathbf{x}; \cdot) = \Phi(\mathbf{x}; \cdot)^{-1}$ which projects each pixel u to its corresponding $z \in \mathcal{Z}$. This can be understood visually as simply inverting the direction of the arrows in Figure 1.8, giving Figure 1.9.

This dense formulation is developed in Chapter 5, with a loss function based on Section 1.3.2 that implicitly incorporates equivariance and distinctiveness.

1.2.3 Symmetry

So far we have only looked at the basic topological structure of objects. An additional level of structure can be gleaned from the *symmetries* present in objects, in particular the bilateral symmetry present in humans and animals, as well as many man-made objects.

Importantly, when it comes to deformable objects, this sort of symmetry does not normally align with the geometrical concept of symmetry. Humans largely do not spend their time in uncomfortable poses resembling the *Vitruvian Man*.

By taking advantage of the *Object Frame* concept, our aim is to capture the symmetry in the underlying object categories. This can be seen as moving beyond looking for symmetry in the precise physical shape of any given object, and instead examining the whole space of object deformations. These deformations account for object motion, such as rigid or elastic articulations, and as we show in Chapter 6 are an appropriate tool to characterise symmetries.

By building on the dense *Object Frame* framework as realised in Chapter 5 and [89] we show that, by working in the *Object Frame* space, the symmetries we are interested in reduce to a simple transformation group. We can incorporate this knowledge into our unsupervised training formulation by adapting the equivariance constraint to explicitly account for symmetries.

Specifically, we can capture bilateral symmetry such as that of human faces by using the constraint that mirroring the input image should result in a simple reflection of the embedding space across a chosen axis. This axis then gives us a reference frame with respect to bilateral symmetry which can be used to identify symmetric landmarks on an object, even when in image space the object is not symmetric due to variations in viewpoint or deformation.

For bilateral symmetry, one can uniquely identify a left or right side of a person or animal as they appear in an image, and recover an unambiguous pose. This is not the case for rotational symmetry, for example that of a flower, where no particular rotation is canonical. There is an inherent ambiguity which we must address. We can incorporate this ambiguity elegantly into our framework and learning formulation by ensuring that no single orientation of our object frame is special when it comes to our loss function. Instead, for n-fold rotational symmetry, we capture the ambiguity that an image may map to any of the n rotations in intervals of $360°/n$ of the embedding space.

1.2.4 Object Frame vs Visual Descriptors

The formulation of Section 1.2.2, which is expanded upon in Chapter 5, learns to predict per-pixel vectors in a low-dimensional space such as $\mathbb{R}^3$ that are descriptive of the pixel contents. In this way, our *Object Frame* is not that dissimilar from generic visual descriptors such as SIFT [61], which are typically used for image matching, particularly when combined with interest point detectors, such as the corner detection method due to Harris [40].

There are two key differences between our method and traditional descriptors. The first is that of dimensionality – our descriptors have a much lower dimensionality than SIFT, and this bottleneck discourages discriminative yet non-semantic descriptors that can get away with simply describing the image texture. The second difference comes down to the data used. Our images are all of the same object category, and our descriptors are learnt from this data using many instances. These two properties combine such that the only sensible thing for the network to do is learn object-centric coordinates.

This exposes one drawback of our method, which is that the ability of our descriptors to generalize is inherently tied to their low dimensionality, which however harms their precision when it comes to matching, a property verified empirically in Chapter 7.

We need another way to make sure that our descriptors live in the same, small, volume of embedding space, without artificially limiting the amount of information they can convey.

An suitable alternative construction is proposed in Chapter 7, where we develop a technique to control the embedding capacity without fixing the dimensionality of the embedding space to a small number. We instead take full advantage of a high-dimensional embedding space and instead control its capacity with the novel introduction of *Embedding Volume Compression*. This regularises the set of embedding vectors for a category, encouraging embedding vectors extracted from any instance of the same object category to be interchangeable.

A theoretically meaningful outcome of this formulation is that it allows us to blur the boundaries between descriptors and landmarks, two important and usually complementary representations in computer vision. While descriptors are typically computed on top of landmarks, we show that we can reverse the interpretation to see landmarks as a special case of image descriptors.

By extending the learning setup of Chapter 5 to incorporate *Embedding Volume Compression*, we show that we can get the best of both worlds, obtaining embedding

vectors that work well as instance-specific image descriptors *and* intra-category dense landmarks. This formulation learns high dimensional *Object Frames* with no manual supervision, which we can reinterpret as a set of dense landmarks. The resulting embeddings can be simultaneously interpreted as patch descriptors which are good for the task of matching the same object across different views, as well as landmarks for identifying object parts consistently across multiple instances of an object category.

1.3 Flow and Dense Correspondences

The term "flow" can be used in an abstract sense to denote the concept of dense correspondences between images, mapping some useful real world property to a vector field.

Flow is useful to us on two accounts: firstly it gives a supervisory signal to learn from video data, and secondly flow of a semantic nature is similar in spirit to our desire to learn object layout, albeit without an explicit spatial labelling or reference frame.

Optical flow, which specifically deals with the temporal motion of pixels from frame to frame, is relevant in the current work since it gives the "motion" in the "Objects from Motion", providing the supervisory signal relating corresponding points through time. In particular, Chapter 8 learns embeddings that are consistent with estimated optical flow.

There is much work on the estimation of optical flow from pairs of frames, and we explore and extend one such method in Section 1.3.1 and chapter 3. However, we also note that, for many of the object categories in question, such as faces, it is sufficient to create synthetically warped versions of real images using Thin Plate Splines [9]. Here, the dense correspondence field between images is known by default and need not be estimated. Results using these synthetic warps are shown in Chapters 4 to 7. Another way to obtain optical flow "for free" is to use a graphics engine to generate videos of moving objects, storing the pixel motion. We use this approach for toy examples in Chapters 5 to 7.

1.3.1 Deep Matching

Released in 2015, FlowNet [31] was the first attempt to tackle optical flow estimation using a Convolutional Neural Network. However, it underperformed in terms of accuracy relative to state of the art methods that did not learn from data.

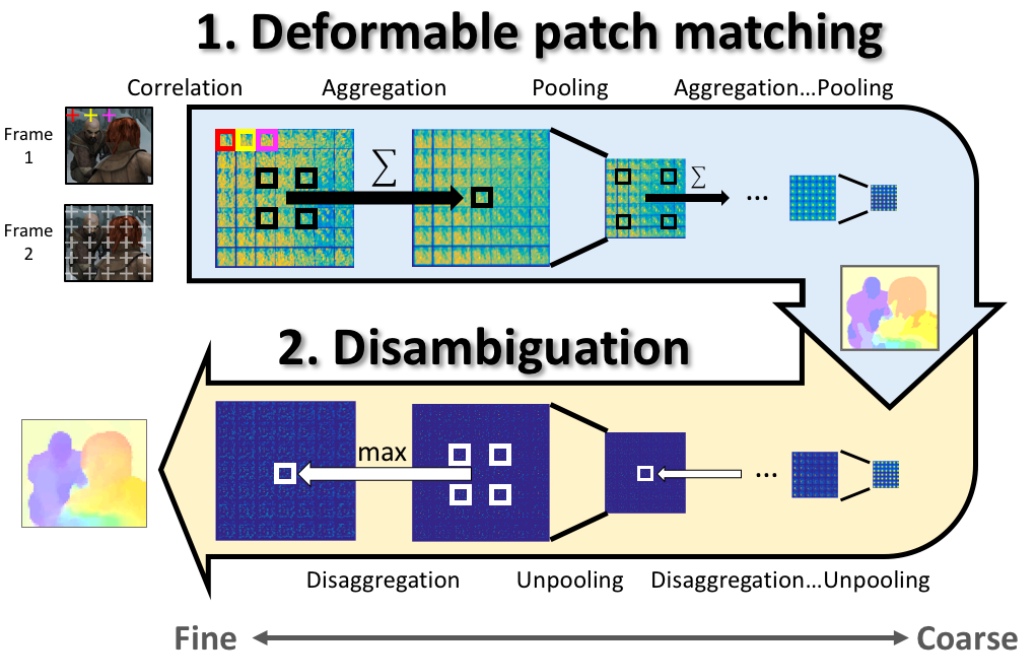


Figure 1.10: Deep Matching schematic

At the time, a leading method to compute optical flow involved using the Deep Matching [79] algorithm to compute quasi-dense matches, before refining these matches to a dense flow field using EpicFlow [80]. Despite using operations inspired by CNNs, Deep Matching is not trainable from data. Therefore, in order to obtain both high performance and the ability to learn from data, in Chapter 3 we convert Deep Matching into an architecture suitable for training using backpropagation.

Deep matching starts by correlating small patches in the source image with all the patches in the target image. This produces a 4D score map, which we can think of as an array of heatmaps giving the location of each source index in the target image.

As visualised in Figure 1.10, in our implementation of Deep Matching, this is followed by a two-stage process of refinement, firstly computing coarser, less ambiguous matches through aggregation and max pooling. The second stage upsamples the disambiguated score maps through a series of unpooling and disaggregations operations, obtaining accurate high resolution score maps.

While the first stage is mostly identical to that in [79], the second stage was originally given as a recursive decoding algorithm. In Chapter 3, we show that it can be implemented instead by convolutional operators, allowing backpropagation through both stages to a feature extraction CNN.

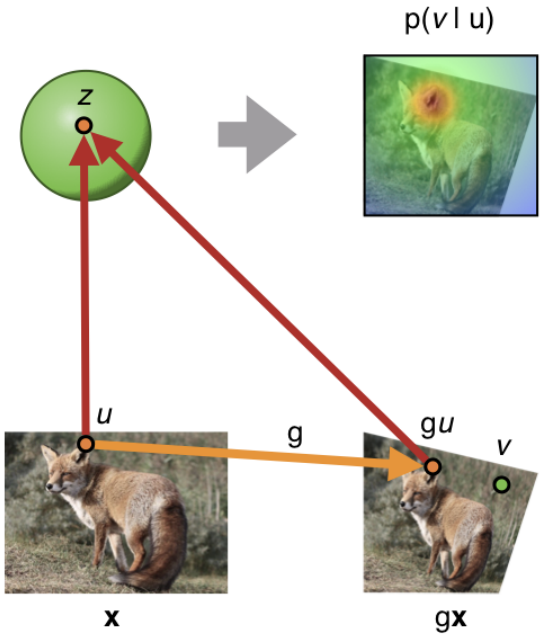


Figure 1.11: The dense probabilistic loss operates on heatmaps between pixels u in the first image and v in the second, warped, image

1.3.2 Probabilistic Matching Losses

The way that the Deep Matching algorithm employs operations on a 4D tensor in order to represent and refine dense correspondences is a useful abstraction. We have used a very similar idea in later work, expressing the loss functions in Chapters 5 to 7 as functions of 4D grids of heatmaps resulting from the correlation between source and target descriptors.

In this setting, as in Figures 1.9 and 1.11, we are typically mapping pixel locations u in an image $\mathbf{x}$ to 3D vectors using some neural network $\Psi(\mathbf{x}; u) \in \mathbb{R}^3$. A 4D conditional heat map expressing the inner product between pixels u in the source image and v in the target image can then be given as $s(v|u) = \langle \Psi(\mathbf{x}; u), \Psi(\mathbf{x}; v) \rangle$. Here, we can use vector length to code for certainty, since a shorter vector will give a lower score.

In order to give a probabilistic interpretation to these heatmaps, we can normalise

them using the softmax function, defining $p(v|u) = \frac{\exp(s(v|u))}{\sum_v \exp(s(v|u))}$.

This then allows us to write loss functions of the form given in Equation (1.2), where g represents the warping function. The distance term then encourages the heatmaps to be compatible with the warp, since for a location to have a high probability while still minimising the loss, the distance must be very small.

$$\mathcal{L} = \sum_u \sum_v \|v - gu\|_2 \, p(v|u). \tag{1.2}$$

1.3.3 Learning from Flow

In Chapter 8, we seek to use self-supervised learning from motion cues given by optical flow estimation. This idea of learning features from the motion of objects was shown to be promising in the work of Pathak *et al.* [73].

While it might seem tempting to simply learn to regress the flow given an input image, this task is inherently ambiguous: we cannot in general tell the motion of objects from a single image, so training a network to perform this impossible task is unlikely to learn useful features, something we verify empirically. Some initial work by the authors towards solving this problem without the need for the preprocessing stages of [73] revolved around predicting motion-consistent segmentations, work that was submitted as a workshop paper [64]. This evolved into the method of Chapter 8, which forgoes the interpretable segmentation in favour of learning a per-pixel embedding, achieving state of the art results. This embedding is encouraged to be consistent with the flow by aligning similarity kernels between the flow vectors and the pixel embeddings.

1.4 Conclusions

In this thesis we have presented an approach to learn the structure of deformable objects without supervision. By formulating this challenge in ways amenable to learning using Convolutional Neural Networks, we have obtained state of the art results. Our contributions include expressing this problem as one of landmark prediction in a manner compatible (equivariant) with synthetic deformations, extending this to a dense geometric embedding, showing the value of this embedding in capturing symmetries, and exploring the relation between semantic and distinctive descriptors. We have also introduced a trainable optical flow method and a method for self-supervision using flow.

Further work on the Objects from Motion idea aims to look at richer classes of objects and motion. The ideal scenario would involve a method that learns from relatively unconstrained video sequences, such as human pose or animals in the wild. This could take advantage sparse tracks on human pose data.

Chapter 2

Literature Review

The project blends the idea of describing objects by their layout with that of describing objects by their relation to other objects of the same category, with the additional challenge of learning from motion. Hence we give a treatment of the literature surrounding "flow" which, connecting points of an object temporally, accounts for its shape and deformation, while connecting objects semantically, gives a frame of reference with which to align different instances. We further describe methods incorporating an estimate of object layout, either as a collection of parts or through an explicit 3D model, and finally examine the literature on learning from video in an unsupervised or weakly supervised fashion.

2.1 Flow

In the seminal work of Horn and Schunck [45] on optical flow, this mapping represents the temporal motion of points in a video, providing for each pixel in frame N the answer to where it ends up in frame $N+1$. The generalised notion of flow can extend to more semantic mappings, such as equivalent points on different object instances of the same category, for example two different cars.

Although flow is most commonly posed as a pairwise problem, taking advantage of the transitivity of flow on a collection of images has been used in the context of graph-based optimisation techniques employing cycle consistency as a constraint.

2.1.1 Pairwise Flow

Optical flow, with the aim of establishing correspondence between spatially and temporally related images, has its origins in the work of [45], which pioneered the field of variational methods that has dominated the optical flow literature since. The original

method employs variational energy minimisation with a local smoothness assumption on the flow field. Although initially losing out in popularity somewhat to sparse, computationally cheaper point tracking such KLT [62] [92], innovations such as [6] improved robustness in face of discontinuities and [12] combined several advances into a coherent, reliable framework.

A more recent addition to the repertoire of optical flow techniques is "Large Displacement Optical Flow" introduced by Brox and Malik [13], which adds a descriptor matching term to the energy to be minimised so as to anchor the flow in the vicinity of long range matches that traditional variational approaches would struggle to find. This has inspired other methods to extract matches such as DeepFlow [101], which uses the CNN-inspired DeepMatching [79] to aggregate matches across multiple scales in a hierarchical manner and refines them with a variational method based on [13]. EpicFlow [80] again builds on DeepMatching, but uses edge-preserving interpolation with a geodesic distance (which better handles motion boundaries) to convert the matches into a dense field before the variational step. Flow Fields [3], one of the best methods on the MPI-Sintel benchmark builds on EpicFlow, but replaces the DeepMatching matches with a novel correspondence fields method to propagate matches initialised from a kd-tree.

These methods, though successful, do not employ any learning from data, and until recently supervised learning of optical flow was comparatively rare. There is some early work by [7] which learns to model flow as a linear combination of basis flows computed using PCA. However, the arrival of large synthetic datasets with flow groundtruth along with the success of "Deep Learning" in fields such as object recognition has made it more attractive to pose optical flow as a learning problem. FlowNet [31] attempts to learn optical flow with a CNN, achieving reasonable performance but not managing to beat the best non-learnt methods.

The idea of flow between semantically related images was introduced by SIFT Flow [58], which finds dense correspondences between images of different scenes that share certain characteristics. A matching image to a query is retrieved from a large database, and dense SIFT [61] descriptors between the two images are matched in a similar way to temporal optical flow, using an energy function that accounts for discontinuities.

2.1.2 Graph-based methods

The benefit of having a collection of images which can all be related to one another by some flow-like property is that the collection as a whole may speak more about

the correspondences than when considering individual pairs. In the case of flows, we may connect images in a graph structure and relate points transitively, since (barring occlusions) the equivalence of the points at any path through the graph will be preserved. This allows for powerful graph based optimisation techniques to refine the flows based on constraints such as preserving cycles.

We are interested in giving a consistent, dense, canonical labelling to visual concepts. One way of viewing this is through the lens of matching, for example if we can consistently find correspondences to a reference image or set of images, this can be seen as a de facto labelling of the concept.

This is the paradigm used by Collection Flow [50] in the context of faces, by projecting each image onto a subspace representing a common neutral expression, using PCA in a similar way to Eigenfaces [95]. The projected images are in correspondence (having zero flow between them), hence it suffices to compute the flow to this neutral reference expression to put any two faces in correspondence through concatenation of flows. This can be seen as providing a dense labelling of faces, and has the advantage of avoiding $O(n^2)$ comparisons to find pairwise flows across n images.

This idea is extended to general objects by FlowWeb [115], which establishes correspondence globally as joint label assignment across an image collection of different object instances, iteratively updating a complete flow graph between images to make it more self-consistent in terms of image triplets. The number of 3-cycles going through a flow is used to measure its quality. Relevant to our goal is their hypothesis that "global correspondences emerge from consistent local correspondences in a bootstrap fashion", since we likewise wish to build upon atomic correspondences, in the hope that it will converge to a solution to our desired problem rather than some failure mode. A pitfall of FlowWeb is that it scales poorly, relying on computing flows on the complete graph of images and hence $O(n^2)$ initial comparisons. It also does not employ learning in the sense of determining the parameters of a model, instead the model is the concrete set of images itself and the flows between them. This means that at no point is the concept of corresponding points distilled into some flexibly reusable form, and establishing correspondence to a new images requires adding it to the graph and re-running the optimisation.

[81] navigates the space of articulations through the use of optical flow on a collection of images. Like FlowWeb they define connections on a graph, but they establish connections in a neighbourhood with respect to a reference image and project the flows to a lower dimensional Euclidean space. A new vector in this low dimensional

space can then be represented as a convex combination of its nearest neighbours, and by applying these weights to the original flows we can navigate to a new image.

[114] builds upon the idea of cycle consistency, using synthetic data as an intermediary to complete the cycle, such that the synthetic-synthetic flow is know but the synthetic-real-real-synthetic edges are predicted. They use a CNN to predict both the flow itself and the "matchability" score indicating whether the corresponding point is visible.

In a similar spirit to Collection Flow, [68] models objects as a combination of low dimensional basis functions, such that geometry can be factored out. Like us, they are interested in manifold learning for natural images in a way that does not discard spatial structure. To model images of objects at a global level while avoiding an intractably large sampling of image space, they propose a compositional latent structure, with shape being the innermost nested component, followed by appearance then colour the outermost, with shape and appearance living in a low-dimensional subspace computed using PCA. While this has applications in domains such as colour transfer and morphing, the main interest for us is the ability of the model to provide a common coordinate system in latent space. Concretely, thanks to the regularisation provided by learning the model from a large set of images, inverting the shape map for two different images will give a corresponding location, hence the flow between objects of the same class can be easily found. However, like FlowWeb [115], it expects the whole image collection to be present in advance, since it involves running an iterative optimisation process to jointly find the three maps per image.

2.1.3 Flow as supervision

Flow has been used as a supervisory or auxiliary cue when training or pretraining neural networks for other tasks. Optical flow has been used as an input signal in two-stream convolutional neural networks for action recognition in video [86]. Alternatively, one may incorporate flow-derived terms in the loss function during training, and in this setting flow is not required at test time. This can give the desirable property of being able to use static images rather than video frames as input. However, since direct prediction of optical flow from a single image is inherently ambiguous, [73] proposes instead to first employ the motion information to generate foreground-background masks using an off-the-shelf pipeline, and use these masks as the prediction target. [64] proposes instead to directly predict a flow-consistent segmentation, by grouping pixels such that they move coherently under certain motion models. In

Chapter 8, which is based on [63], we instead formulate the problem in terms of embedding alignment, encouraging the similarity between pixel embeddings to match that between their optical flow vectors.

2.2 Learning Object Layout

A simplifying principle when dealing with computer vision problems involving objects is to look at them in terms of their constituent parts. This can give a straightforward way of dealing with articulated objects, different viewpoints and occlusion. Although largely used for object detection, such methods can be seen as a more sparse form of correspondence. In the same way that our proposed method could identify equivalent points on objects, the parts learnt to represent, say, the limbs a human would provide a coarse spatial labelling accounting for the relative deformation of different instances.

Modeling object structure is a widely studied computer vision problem including important applications such as facial landmark detection and human body pose estimation. Previous work has proposed the use of PCA-based shape constraints [17], templates [21], pictorial structures such as DPMs [28] and poselets [10]. Recent work [106] benefit from powerful deep representations.

2.2.1 Parts and Templates

The constellation approach used in [29] and based on [14] [100] is one of several part-based models. It uses a feature detector to find regions of interest, and parameters such as the relative locations of regions are learnt and modelled as a Gaussian distribution. The popular Deformable Parts Model (DPM) [28] uses the HOG [21] feature and combines a coarse global "root" template with finer part templates and a spatial prior, while being trained just from object bounding boxes using a latent SVM to optimise the latent part configuration. The location of parts is considered relative to the root, giving a "star" structure. In order to account for the different modes of visual appearance, for example a car seen head on compared to from the side, the DPM models an object category as a mixture of these star models. This explicit accounting for different viewpoints and other major appearance variations gives improved detection accuracy, but means that DPMs do not give a consistent frame of reference, since we cannot put instances belonging to different mixture components into correspondence.

Incorporating knowledge of the 3D structure of an object class has been used to assist many vision tasks, since fitting a 3D model to an image will give a reasonable

estimate of the pose and give a correspondence between parts. An obvious source of 3D information is CAD models, which have been used successfully in areas such as scene understanding with 3D Nearest Neighbors [82], which aligns models from a library to objects in a scene in order to generate hypotheses of the underlying geometry. However, acquiring pre-made models may not be feasible for arbitrary object classes, and aligning models to images can be quite a brittle process, especially in the case of non-rigid objects. 3D LayoutCRF [44] tackles object recognition and segmentation through the use of a rough 3D model obtained by space carving on the ground truth segmentations. Parts are defined on a grid over the 3D surface of the object, with the 3D model used to provide correspondence between the same parts across instances and viewpoints. An appearance model is learnt for these parts using decision forests, which allows them to deform to match each instance better, and global part layout consistency (ensuring neighbouring pairs of parts are ordered consistently with the 3D layout) is incorporated into the energy function of the CRF along with an object instance model (which ensures attributes like colour are consistent across parts). [94] goes further by learning deformable 3D basis shape models, where the modes of deformation are class-specific. Models are learnt from segmentation and keypoint annotations, by using estimated camera projection parameters to give object silhouettes and optimise a shape model consisting of a mean shape and basis vectors of deformation, subject to silhouette and keypoint consistency. The 3D models can then be aligned to images starting from object detection and segmentation based on [39] to try to find a shape that explains the silhouettes of the detected objects. Further high frequency shape detail than then be recovered.

More recently, AnchorNet [71] learns parts that match across object instances using only image-level supervision. Progress has also been made in increasing the quality and density of supervised part prediction, such as DenseReg [38] and DensePose [37] which can densely index human parts in images and map them to a canonical 3D surface.

2.2.2 Deformable Templates

The analysis of deformations owes much to the work of D'Arcy Thompson [91] studying deformations found in nature. Our work can be seen as a case of the deformable template paradigm, which early work expressed through conceptual "rubber masks" [102] and "springs" [32] among many other formulations [2, 107, 8, 47]. In this paradigm an object instance is modelled as a deformation of some reference "template", which may be learned without supervision.

Influential work in this area includes Active Shape Models [17], where a statistical model of shape is learned from a training set of labelled contours. This model captures the modes of variation seen in the training set, and can be iteratively deformed to find the shape of a new object instance. Building on this, Active Appearance Models [18, 66] combine a shape model with a statistical model of appearance, obtained from pixel intensity on aligned training images.

There is also much work on deformable template models that don't use supervision, such as the automatic construction of Active Appearance Models [4, 51]. It is often posed as a problem of joint image set alignment, which can be tackled by Congealing [55, 97] or the construction of diffeomorphic warps [19]. It can also be seen as a form of clustering that is invariant to transformations [33].

Although we do not explicitly construct a template as such, and prediction does not involve the iterative refinement present in some deformable template models, our work can be seen as exploiting the same paradigm through the machinery of a Convolutional Neural Network.

Similarly to Chapter 5, the unsupervised deforming autoencoders of [85] seek to discover a canonical coordinate frame, which they do by prediction a deformation from a canonical template coordinate system.

2.2.3 Landmark Detection

A final approach to determining object layout relies on localising pre-identified landmarks, which may correspond to body parts or facial features. An influential example of this is [84] used in the Kinect gaming platform, which develops a system for human pose estimation from depth images, giving dense body part estimations using a randomised decision forest classifier, trained in part on synthetic data.

There is an extensive literature on landmark detectors, particularly for faces. Examples include Active Appearance Models [18], along with subsequent improvements [66, 20] and others using templates [75] or parts [116]. Other approaches directly regress the landmark coordinates [96, 22, 15, 78]. Deep learning methods use cascaded CNNs [87], coarse-to-fine autoencoders [108], auxiliary attribute prediction [111, 112], learned deformations [106] and LSTMs [104]. Beyond faces, there is work on humans [105, 93], birds [83, 59, 106] and furniture [103]. More general pose estimation including the case of landmarks is explored in [25]. Our method can build on any such detector architecture and can be used as a pretraining strategy to learn landmarks with less or no supervision.

In a more "deep learning" setting, [103] proposes an end-to-end framework for estimating both the position of 2D keypoints and the 3D skeleton connecting them. The Deep Deformation Network of [106] localises landmarks with the help of a "Shape Basis Network" CNN which provides an initial shape (where a shape is a set of points giving the 2D positions of landmarks) as a linear combination of shape PCA bases. This is then refined using a "Point Transformer Network" which generalises the idea of a Spatial Transformer Network (STN) [46], deforming the shape using an affine transform plus a non-linear thin-plate spline transform. Unlike the STN, but like WarpNet [49], it operates on a set of points rather than warping the image or feature maps, and has a geometric ground truth rather than optimising for classification accuracy.

It is worth noting that the STN itself can be seen as mapping an object to a canonical reference frame, and preceding work in this area goes back to Hinton in 1981 [43].

Subsequent to the publication of Chapter 4, [90], several other methods have also looked at the problem of learning landmarks without supervision. The work of [110] uses conditional image generation with landmarks as an intermediary, along with equivariance constraint. [48] use a pair of images to condition their reconstruction and do not require the equivariance constraint. Promising results on landmark discovery have also been shown when moving to three dimensional keypoints [88].

2.2.4 Symmetry

A survey of computational symmetry is given by [60]. Much work [65] looks at finding symmetry as it appears in images themselves, rather than the object-centric symmetry of Chapter 6. Raviv *et al.* [77] looks at intrinsic symmetry of non-rigid shapes with respect to a metric of the surface.

2.3 Learning with less supervision

2.3.1 Unsupervised learning

Classical unsupervised learning methods such as autoencoders [11, 5, 42] and denoising autoencoders aim to learn useful feature representations from an input by simply reconstructing it after a bottleneck. Generative adversarial networks [36] target producing samples of realistic images by training generative models, which can also be used to learn good features [26, 27].

2.3.2 Self-supervised learning

Recently a lot of work has focused on employing pretext tasks to pretrain neural networks. These methods construct some sort of auxiliary pseudo-task as "self-supervision" by truncating or perturbing the input signal, requiring the network to predict the withheld information. Many methods consider jigsaw-style shuffling of patches such as Doersch *et al.* [24] and Noroozi and Favaro [69]. Other self-supervised tasks include colorizing images [109, 53], inpainting [74], ordering video frames correctly [67, 30] and tracking [99]. Agrawal *et al.* [1] use egomotion as supervisory signal by predicting camera transformations and Pathak *et al.* [73] learn to group pixels that move together using segmentation from video. [113, 34] learn depth from video. Our task of aligning warped images can be seen as self-supervision. However, rather than alignment being an unrelated auxiliary task, it is the problem we aim to solve. We discover that the generalisations made to learn an appropriate representation for same-instance alignment lead to the natural emergence of across-instance alignment.

2.3.3 Unsupervised and Weakly Supervised Learning from Video

The idea of using large amounts of unlabelled video data in lieu of labelled static images has been touched upon by numerous papers, since it offers a potential relief to the burden of curating large labelled datasets prevalent in supervised methods. Unsupervised learning from video also gives the possibility of taking advantage of semantically rich motion cues, where per-frame labelling would be impractical.

A motivating example in a deep learning setting is that of [98], which trains a visual descriptor based on triplets of patches from 100K videos, ensuring that patches matching along a track are ranked higher than a negative patch. Using an Alexnet architecture and fine-tuning on Pascal VOC 2012, they achieve 52% mAP, approaching the 54.4% mAP of a network pretrained on the circa 1 million labelled images of ImageNet. They note the role of dynamic sensory inputs in human learning, in particular how infants develop visual tracking prior to semantic representations. Another example of the powerful semantic features that can be learned from "watching" video in an unsupervised fashion is [54], which trains an autoencoder on 10 million images from YouTube videos and manages to capture high level concepts such as human and cat faces. Although not based on temporal information, since only one

frame per video is used, it demonstrates the advantages arising from the sheer scale of large unlabelled datasets.

Prior to the increased popularity of deep learning, there was relatively limited use of video data to directly learn high level concepts. One endeavour is [76], which uses YouTube videos with "weak" (per-video) class annotations in order to learn object detectors. The approach consists of a pipeline constructing candidate "tubes" of objects using optical flow [13], selecting the dominant tube per shot (via energy minimisation over all videos) and training object detectors from frames sampled from the selected tubes. It also tackles the problem of domain adaptation, since stills from video footage can have specific visual characteristics such as motion blur which could harm generalisation to images from other domains.

Discovering object aspects (viewpoint, pose, occlusion) from weakly labelled video is tackled in [72], which finds that foreground-background segmentation from motion followed by computing image descriptors and clustering frames reveals the main visual aspects of tigers and cars, with applications to learning aspect transitions from video. [23] instead uses motion-based "Pairs of Trajectories" features to discover the behaviours of objects, with motion segmentation and clustering of short sequences. Identified sequences of the same behaviour are then aligned by identifying subsequences with common motion and finding a smoothly time-varying set of Thin Plate Splines. The found spatial alignment outperforms SIFT Flow [58], but is limited to sequences showing the same behaviour.

Chapter 3

Fully-Trainable Deep Matching

This work was accepted for Oral Presentation at the British machine Vision Conference (BMVC), York, 2016

This paper re-interprets a quasi-dense matching algorithm as a neural network, allowing it to be trained end-to-end. By backpropagating to the initial patch descriptor CNN, we show that we can improve the accuracy of the obtained matches. These improved matches can also be used with EpicFlow interpolation in order to obtain high quality dense optical flow. As explained in Section 1.3.2, the task of matching comes up in later chapters, and the matching loss introduced in Chapter 5 is very similar to the dense patch correlation performed in this paper.

Fully-Trainable Deep Matching

James Thewlis
jdt@robots.ox.ac.uk
Shuai Zheng
shuai.zheng@eng.ox.ac.uk
Philip H. S. Torr
philip.torr@eng.ox.ac.uk
Andrea Vedaldi
vedaldi@robots.ox.ac.uk

Department of Engineering Science
University of Oxford
Oxford, UK

Abstract

Deep Matching (DM) is a popular high-quality method for quasi-dense image matching. Despite its name, however, the original DM formulation does not yield a deep neural network that can be trained end-to-end via backpropagation. In this paper, we remove this limitation by rewriting the complete DM algorithm as a convolutional neural network. This results in a novel deep architecture for image matching that involves a number of new layer types and that, similar to recent networks for image segmentation, has a U-topology. We demonstrate the utility of the approach by improving the performance of DM by learning it end-to-end on an image matching task.

1 Introduction

Deep Matching (DM) [19] is one of the most popular methods for establishing quasi-dense correspondences between images. An important application of DM is optical flow, where it is used for finding an initial set of image correspondences, which are then interpolated and refined by local optimisation.

The reason for the popularity of DM is the quality of the matches that it can extract. However, there is an important drawback: DM, as originally introduced in [19], is in fact *not* a deep neural network and does not support training via back-propagation. In order to sidestep this limitation, several authors have recently proposed alternative Convolutional Neural Networks (CNN) architectures for dense image matching (Sect. 1.1). However, while several of these trainable models obtain excellent results, they are not necessarily superior to the handcrafted DM architecture in term of performance.

The quality of the matches established by DM demonstrates the strength of the DM architecture compared to alternatives. Thus, a natural question is whether it is possible to obtain the best of both worlds, and construct a trainable CNN architecture which is equivalent to DM. The main contribution of this paper is to carry out such a construction.

In more detail, DM comprises two stages (Fig. 1): In the first stage, DM computes a sequence of increasingly coarse match score maps, integrating information from progressively larger image neighbourhoods in order to remove local match ambiguities. In the second

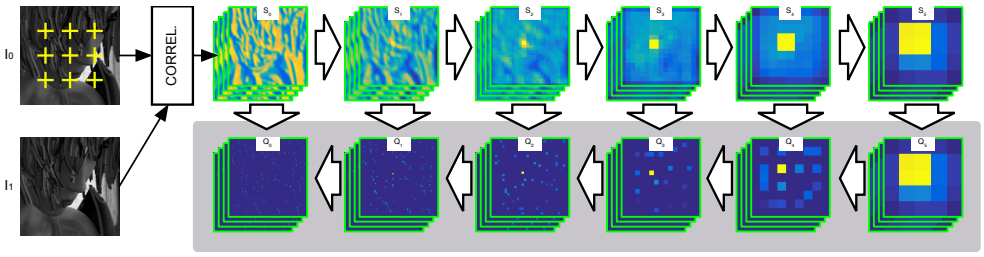


Figure 1: **Fully-Trainable Deep Matching.** Deep matching starts by correlating small patches $\mathbf{p}$ in the reference image I_0 (crosses) with all the patches $\mathbf{q}$ in the target image I_1, producing a 4D score map S_0 (slices $S_0(\cdot|\mathbf{p})$ for varying $\mathbf{p}$ are shown); then, it computes coarser but less ambiguous maps $S_1,\dots,S_L$. In this paper, we formulate the reverse process, reconstructing high resolution matches from coarser ones, as a sequence of *reverse convolutional operators*, producing scores $Q_L,\dots,Q_0$ (shaded area). The result is a deep convolutional network with U-architecture that can be trained using backpropagation.

stage, the coarse information is propagated in the reverse direction, resolving ambiguities in the higher-resolution score maps. While the first stage was formulated as a CNN in [19], the second stage was given as a recursive decoding algorithm. In Sect. 2, we show that this recursive algorithm is equivalent to dynamic programming and that it can be implemented instead by a sequence of new *convolutional operators, that reverse the ones in the first stage of DM*.

The resulting CNN architecture (Fig. 2), which is *numerically equivalent to the original DM*, has a U-topology, as popularized in image segmentation [20], and supports backpropagation. Combined with a structured-output loss (Sect. 2.2), this allows us to perform end-to-end learning of the DM parameters, improving its performance (Sect. 3). Our findings and further potential advantages of the approach are discussed in Sect. 4.

1.1 Related Work

The key reason for the success of CNNs in many computer vision applications is the ability to learn complex systems end-to-end instead of hand-crafting individual components. A number of recent works have applied CNN-based systems to pixel-wise labeling problems such as stereo matching and optical flow. In particular, Fischer *et al*. [5] have shown it is possible to train a fully convolutional network for optical flow. Žbontar *et al*. [28] trained a CNN for stereo matching by using a refined stereo matching cost. Zagoruyko and Komodakis [27] and Han *et al*. [7] have demonstrated learning local image description through a CNN.

Optical flow estimation was tackled mostly by variational approaches [3, 14, 25] since the work of Horn and Schunk [8]. Brox and Malik [2] developed a system that integrates descriptor matching with a variational approach. Recently, leading optical flow approaches such as DeepMatching [19, 26] demonstrated a CNN-like system where feature information is aggregated from fine to coarse using sparse convolutions and max-pooling. However, this approach does not perform learning and all parameters are hand-tuned. EpicFlow [18] has focused on refining the sparse matches from DM using a variational method that incorporates edge information. Fischer *et al*. [5] trained a fully convolutional network FlowNet for optical flow prediction on a large-scale synthetic flying chair dataset. However, the results of FlowNet do not match the performance of DM on realistic datasets. This motivates us to

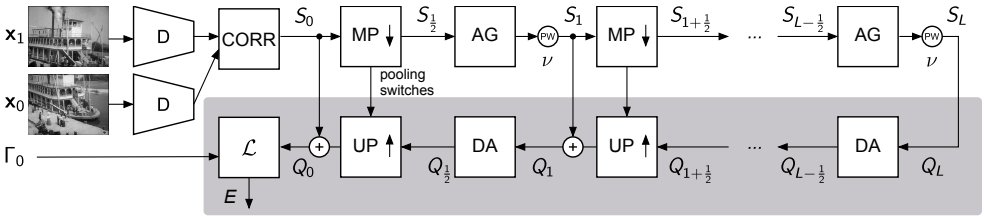


Figure 2: End-to-end deep matching architecture, involving the following layers: descriptor extraction (D), correlation (CORR), max pooling (MP), aggregation (AG), power (PW), disaggregation (DA), unpooling (UP), summation (+), and structured loss ($\mathcal{L}$). The shaded area encloses our contribution, which amounts: formulating the DM decoding algorithm as a sequence of convolutional neural network layers supporting backpropagation.

reformulate DM [19] as an end-to-end trainable neural network.

Beyond CNNs, many authors have applied machine learning techniques to matching and optical flow. Sun *et al*. [23] investigate the statistical properties of optical flow and learn the regularizers using Gaussian scale mixtures, Rosenbaum *et al*. [21] use Gaussian mixture models to model the statistics of optical flow, and Black *et al*. [1] apply the idea of principal components analysis to optical flow. Kennedy and Taylor [9] train classifiers to choose different inertial estimatiors for optical flow. Leordeanu *et al*. [11] obtain occlusion probabilities by learning classifiers. Menze *et al*. [16] formulate optical flow estimation as a discrete inference problem in a conditional random field, followed by sub-pixel refinement. In these works, tuning feature parameters is mostly done separately and manually. In contrast to these works, our work aims to convert the whole quasi-dense matching pipeline into an end-to-end trainable CNN.

2 Method

Our key contribution is to show that the full DM pipeline can be formulated as a CNN with a U-topology (Fig. 2). The fine-to-coarse stage of DM was already given as a CNN in [19]. Here, we complete the construction and show that the DM recursive decoding stage can: (1) be interpreted as dynamic programming and (2) be implemented by convolutional operators which reverse the ones used in the fine-to-coarse stage (Sect. 2.1). The architecture can be trained using backpropagation, for which we propose a structured-output loss (Sect. 2.2).

2.1 Fully-Trainable Deep Matching Architecture

In this section we formulate the *complete* DM algorithm as a CNN. Consider a *reference image* $I_0(\mathbf{p}), \mathbf{p} = (p_1, p_2)$ and a *target image* $I_1(\mathbf{q}), \mathbf{q} = (q_1, q_2)$. The goal is to estimate a *correspondence field* $\Gamma : \mathbb{R}^2 \to \mathbb{R}^2, \mathbf{p} \mapsto \mathbf{q}$ mapping points $\mathbf{p}$ in the reference image to corresponding points $\mathbf{q}$ in the target image. The correspondence field is found as the maximizer

$$\Gamma(\mathbf{p}) = \operatorname*{argmax}_{\mathbf{q}} S_0(\mathbf{q}|\mathbf{p}) \tag{1}$$

of a scoring function $S_0(\mathbf{q}|\mathbf{p})$ that encodes the similarity of point $\mathbf{p}$ in I_0 with point $\mathbf{q}$ in I_1 (the score has of course an implicit dependency on I_0 and I_1).[1]

A simple way of defining the scoring function S_0 is to compare patch descriptors. Thus, let $\phi(\mathbf{q}|I) \in \mathbb{R}^d$ be a visual descriptor of a patch centred at $\mathbf{q}$ in image I; furthermore, assume that ϕ is L^2 normalised. The score of the match $\mathbf{p} \mapsto \mathbf{q}$ can be defined as the cosine similarity of local descriptors, given by the inner product:

$$S_0(\mathbf{q}|\mathbf{p}) = \langle \phi(\mathbf{p}|I_0), \phi(\mathbf{q}|I_1) \rangle. \tag{2}$$

A significant drawback of this scoring function is that it pools information only locally, from the compared patches. Therefore, unless all patches have a highly distinctive local appearance, many of the matches established by eq. (1) are likely to be incorrect.

Correcting these errors requires integrating global information in the score maps. In order to do so, DM builds a sequence of scoring functions $S_l(\mathbf{q}|\mathbf{p}), l = 0,1,2,\ldots,L$ which are increasingly coarse but that incorporate information from increasingly larger image neighborhoods (Fig. 1 top). Given these maps, equation (2) is replaced by a recursive decoding process that extracts matches by analysing $S_L, S_{L-1}, \ldots, S_0$ in reverse order.

While the authors of [19] already showed that maps S_l can be computed by convolutional operators, they did not formulate the decoding stage of DM as a network supporting end-to-end learning. Here we show that the recursive decoding process can be reformulated as the computation of additional score maps $Q_l(\mathbf{q}|\mathbf{p}), l = L, L-1, \ldots, 1$ (Fig. 1 bottom) by reversing the convolutional operators used to compute $S_0, S_1, \ldots, S_L$. The two stages, fine to coarse and coarse to fine, are described in detail below.

Stage 1: Fine to coarse. DM starts with the scoring function S_0, computed by comparing local patches as explained above, and builds the other scores by alternating two operations: max pooling and aggregation.

The **max pooling** step pools scores S_l with respect to the first argument $\mathbf{q}$ in a square of side of $2^{l+1}\eta_0$ pixels, where η_0 is a parameter. This results in an intermediate scoring function $S_{l+1/2}$:

$$S_{l+\frac{1}{2}}(\mathbf{q}|\mathbf{p}) = \max \left\{ S_l(\mathbf{q}'|\mathbf{p}),\ \forall \mathbf{q}' : \|\mathbf{q}' - \mathbf{q}\|_\infty \leq 2^l \eta_0 \right\}. \tag{3}$$

In the following, the locations of the local maxima, also known as *pooling switches*, will be denoted as $\mathbf{q}' = m_l(\mathbf{q}|\mathbf{p})$, where m_l is defined such that $S_{l+1/2}(\mathbf{q}|\mathbf{p}) = S_l(m_l(\mathbf{q}|\mathbf{p})|\mathbf{p})$. Note that max pooling is exactly the same operator as commonly defined in convolutional neural networks. The resulting score $S_{l+1/2}(\mathbf{q}|\mathbf{p})$ can be interpreted as the strength of the best match between $\mathbf{p}$ in the reference image and all points within a distance $2^l\eta_0$ from $\mathbf{q}$ in the target image.

After max pooling, the scores are **aggregated** at the four corners of a square patch of side $2^l\delta_0$ pixels:

$$S_{l+1}(\mathbf{q}|\mathbf{p}) = \left[\frac{1}{4} \sum_{i=1}^{4} S_{l+\frac{1}{2}}(\mathbf{q} + 2^l \boldsymbol{\delta}_i | \mathbf{p} + 2^l \boldsymbol{\delta}_i) \right]^\nu \tag{4}$$

where $\boldsymbol{\delta}_i = (\delta_0/2)\boldsymbol{\varepsilon}_i$, $\delta_0 > 0$ is a parameter, and $\boldsymbol{\varepsilon}_i$ are the unit displacement vectors:

$$\boldsymbol{\varepsilon}_1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad \boldsymbol{\varepsilon}_2 = \begin{bmatrix} -1 \\ +1 \end{bmatrix}, \quad \boldsymbol{\varepsilon}_3 = \begin{bmatrix} +1 \\ +1 \end{bmatrix}, \quad \boldsymbol{\varepsilon}_4 = \begin{bmatrix} +1 \\ -1 \end{bmatrix}.$$

[1] As proposed in DM, matches can be verified by testing whether they maximize the score also when going from the target image I_1 back to the reference image image I_0: $\text{verified}(\mathbf{p}) = [\forall \mathbf{p}' : Q(\Gamma(\mathbf{p})|\mathbf{p}) \geq Q(\Gamma(\mathbf{p})|\mathbf{p}')]$.

The exponent ν (set to 1.4 in DM) monotonically rescales the scores, emphasising larger ones. As detailed in [19], the score $S_l(\mathbf{q}|\mathbf{p})$ can be roughly interpreted as the likelihood that a deformable square patch of side $2^{l+1}\delta_0$ centered at $\mathbf{p}$ in the reference image I_0 matches an analogous deformable patch centered at $\mathbf{q}$ in the target image I_1.

Eq. (4) can be rewritten as the convolution of $S_{l+1/2}$ with a particular 4D filter. Note that most neural network toolboxes are limited to 2+1D or 3+1D convolutions (with 2 or 3 spatial dimension plus one spanning feature channels), whereas here there are four spatial dimensions (given by the join of $\mathbf{p}$ and $\mathbf{q}$) and one feature channel, *i.e.* the convolution is 4+1D. Hence, while implementing aggregation through convolution is more general, for the particular filter used in DM a direct implementation of (4) is much simpler.

Part 2: Coarse to fine. In the original DM, scores $S_0, S_1, \ldots, S_L$ are decoded by a recursive algorithm to obtain the final correspondence field. Here, we give an equivalent algorithm that uses only layer-wise and convolutional operators, with the major advantage of turning DM in an end-to-end learnable convolutional network. Another significant advantage is that the final product is a full, refined score map Q_0 assigning a confidence to all possible matches rather than finding only the best ones.

Since the last operation in the fist stage was to apply aggregation to $S_{L-1/2}$ to obtain S_L, the first operation in the reverse order is **disaggregation**. In general, Q_{l+1} is disaggregated to obtain $Q_{l+1/2}$ as follows:

$$Q_{l+\frac{1}{2}}(\mathbf{q}|\mathbf{p}) = \max\left\{Q_{l+1}(\mathbf{q}-2^l\boldsymbol{\delta}_i|\mathbf{p}-2^l\boldsymbol{\delta}_i),\ i=1,2,3,4\right\}. \tag{5}$$

Disaggregation is similar to *deconvolution* [12, 17, 20, 29] or *convolution transpose* [24] as it reverses a linear filtering operation. However, a key difference is that overlapping contributions are maxed out rather than summed.

Next, Q_l is obtained by **unpooling** $Q_{l+1/2}$ and adding the result to $S_l(\mathbf{q}|\mathbf{p})$:

$$Q_l(\mathbf{q}|\mathbf{p}) = S_l(\mathbf{q}|\mathbf{p}) + \max\left\{Q_{l+\frac{1}{2}}(\mathbf{q}'|\mathbf{p}), \forall \mathbf{q}' : m_l(\mathbf{q}'|\mathbf{p}) = \mathbf{q}\right\} \cup \{-\infty\}. \tag{6}$$

Unpooling is also found in architectures such as deconvnets; however here 1) the result is infilled with $-\infty$ rather than zeros and 2) overlapping unpooled values are maxed out rather than summed. The result of unpooling is summed to $S_l(\mathbf{q}|\mathbf{p})$ to mix coarse and fine grained information.

Next, we discuss the equivalence of these operations to the original DM decoding algorithm. In the fine to coarse stage, through pooling and aggregation, the score $S_0(\mathbf{q}_0|\mathbf{p}_0)$ contributes to the formation of the coarser scores $S_l(\mathbf{q}_1|\mathbf{p}_1), \ldots, S_L(\mathbf{q}_L|\mathbf{p}_L)$ along certain paths $(\mathbf{p}_1,\mathbf{q}_1), \ldots, (\mathbf{p}_L,\mathbf{q}_L)$ restricted to the set:

$$\mathcal{H}(\mathbf{q}_0|\mathbf{p}_0) = \{(\mathbf{p}_0,\mathbf{q}_0,\mathbf{p}_1,\mathbf{q}_1,\ldots,\mathbf{q}_L) : \forall l\ \exists i :\ \mathbf{p}_l = \mathbf{p}_{l+1} - 2^l\delta_i,\ \mathbf{q}_l = m_l(\mathbf{q}_{l+1} - 2^l\delta_i|\mathbf{p}_l)\}.$$

DM associates to the match $\mathbf{q}_0|\mathbf{p}_0$ the sum of the scores along the best of such paths:

$$Q_0(\mathbf{q}_0|\mathbf{p}_0) = \max\left\{\sum_{l=0}^{L} S_l(\mathbf{q}_l|\mathbf{p}_l) :\ (\mathbf{p}_0,\mathbf{q}_0,\mathbf{p}_1,\mathbf{q}_1,\ldots,\mathbf{q}_L) \in \mathcal{H}(\mathbf{q}_0|\mathbf{p}_0)\right\}.$$

DM uses recursion and memoization to compute this maximum efficiently; the disaggregation and unpooling steps given above implement a dynamic programming equivalent of this recursive algorithm. This is easily proved; empirically, the two implementations were found to be numerically equivalent as expected.

2.2 Training and loss functions

Training with DM requires to define a suitable *loss function* for the computed scoring function S. One possibility is to minimise the distance $\mathcal{L}(S,\Gamma_0) = \text{mean}_{\mathbf{p},\mathbf{q}} \|S(\mathbf{q}|\mathbf{p}) - g_\sigma(\mathbf{q} - \Gamma_0(\mathbf{p}))\|^2$ between S and a smoothed indicator function $g_\sigma(z) = \exp(-\|z\|^2/2\sigma^2)$ of the ground truth correspondence field Γ_0. While a similar loss is often used to learn keypoint detectors with neural networks [7, 13], it has two drawbacks: first, it requires scores to attain pre-specified values when only relative values are relevant and, second, the loss must be carefully rebalanced as $g_\sigma(\mathbf{q} - \Gamma_0(\mathbf{p})) \approx 0$ for the vast majority of pairs $(\mathbf{p},\mathbf{q})$.

In order to avoid these issues, we propose to use instead the following *structured output* loss:

$$\mathcal{L}(S,\Gamma_0) = \sum_{\mathbf{p},\mathbf{q}} \max\{0, 1 - g_\sigma(\mathbf{q} - \Gamma_0(\mathbf{p})) + S(\mathbf{q}|\mathbf{p}) - S(\Gamma_0(\mathbf{p})|\mathbf{p})\}.$$

Here, the term $1 - g_\sigma(\mathbf{q} - \Gamma_0(\mathbf{p}))$ defines a variable margin for the hinge loss, small when $\mathbf{q} \approx \Gamma_0(\mathbf{p})$ and close to 1 otherwise. This loss looks at relative scores; in fact $\mathcal{L}(S,\Gamma_0) = 0$ requires the correct matches to have score larger than incorrect ones. Furthermore, it is automatically balanced as each term in the summation involves comparing the score of a correct and an incorrect match.

Note that DM defines a whole hierarchy of score maps $(S_0,\dots,S_L,Q_L,\dots,Q_0)$ and a loss can be applied to each level of the hierarchy. In general, we expect application at the last level Q_L to be the most important, as this reflects the final output of the algorithm, but combinations are possible. For n training image pairs $(\mathbf{x}_0^{(i)},\mathbf{x}_1^{(i)},\Gamma^{(i)})$, and by denoting with $\mathbf{w}$ the parameters of DM, learning reduces to optimizing the objective function:

$$\min_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(Q_L(\mathbf{x}_0^{(i)},\mathbf{x}_1^{(i)};\mathbf{w}),\Gamma_0^{(i)}).$$

We follow the standard approach of optimizing the objective using (stochastic) gradient descent [10]. This requires computing the derivative of the loss and DM function Q_L w.r.t. the parameters $\mathbf{w}$, which can be done using backpropagation. Note that, while derivations are omitted, all layers in the DM architecture are amenable to backpropagation in the usual way.

2.3 Discretization

So far, variables $\mathbf{q}$ and $\mathbf{p}$ have been treated as continuous. However, in a practical implementation these are discretized. By choosing a discretization scheme smartly, we can make the implementation more efficient and simpler. We describe such a scheme here.

For efficiency, DM doubles at each layer the sampling stride of the variable $\mathbf{q}$ and restricts the match $\mathbf{q}$ to be within a given maximum distance of $\mathbf{p}$. Hence, $\mathbf{q}$ is sampled as:

$$\mathbf{q} = 2^l\gamma_0(\mathbf{k}_l - 1 - R_l) + \mathbf{p}, \quad \mathbf{k}_l \in \{1,\dots,2R_l+1\}^2,$$

where $\mathbf{k}_l$ is a discrete index, γ_0 is the sampling stride (in pixels) at level $l = 0$, $\gamma_0 R_0$ the distance to $\mathbf{p}$ at level 0, and $R_{l+1} = \lceil R_l/2 \rceil$ is halved with each layer. In this expression, and in the rest of the section, summing a scalar to a vector means adding it to all its components.

For efficiency, DM is usually restricted to a quasi-dense grid of points $\mathbf{p}$ in the reference image, given by:

$$\mathbf{p} = \alpha_0(\mathbf{i}_l - 1 + \tau_l) + \beta_0, \quad \mathbf{i}_l \in \{1,\dots,H_l\} \times \{1,\dots,W_l\}, \quad \tau_l = \mathbf{1}_{l\geq 1}\left(\frac{\delta_0}{2\alpha_0} - \left\lceil \frac{\delta_0}{2\alpha_0} \right\rceil\right).$$

The parameters α_0 and β_0 are the stride and offset of the patch descriptors extracted from the reference image and they remain constant at all layers; however, there is an additional variable offset τ_l to compensate for the effect of discretization in aggregation, as explained below. Here, the symbol $\mathbf{1}_{l\geq 1}$ is one if the condition $l \geq 1$ is satisfied and zero otherwise.

From these definitions, the discretized score maps, denoted with a bar, are given by $\bar{S}_l(\mathbf{k}_l|\mathbf{i}_l) = S_l(\mathbf{q}|\mathbf{p})$, $\bar{S}_{l+1/2}(\mathbf{k}_{l+1}|\mathbf{i}_l) = S_{l+1/2}(\mathbf{q}|\mathbf{p})$, and similarly for $\bar{Q}_l$.

Simplifications arise by assuming that γ_0 divides exactly the pooling window size η_0, that α_0 divides δ_0, and that γ_0 divides α_0. Under these assumptions, $\bar{S}_{l+1/2}(\mathbf{k}_{l+1}|\mathbf{i}_l)$ is obtained from $\bar{S}_l(\mathbf{k}_l|\mathbf{i}_l)$ by applying the standard CNN max pooling operator with a pooling window size $W = 1 + 2\eta_0/\gamma_0$ and padding $P = \eta_0/\gamma_0 + 2R_{l+1} - R_l$. Note in particular that W is the same at all layers. Since usually $\eta_0 = \gamma_0$, this amounts to 3×3 pooling with a padding of zero or one pixels. The discretized aggregation operator is also simple and given by:

$$\bar{S}_{l+1}(\mathbf{k}_{l+1}|\mathbf{i}_{l+1}) = \frac{1}{4}\sum_{i=1}^{4} \bar{S}_{l+\frac{1}{2}}\left(\mathbf{k}_{l+1}\middle|\mathbf{i}_{l+1} + 2^l \frac{\delta_0}{2\alpha_0}\boldsymbol{\varepsilon}_i - \tau_1 \mathbf{1}_{l=0}\right).$$

Note that, since $\mathbf{q}$ is expressed relatively to $\mathbf{p}$, aggregation reduces to averaging selected slices of the discretized score maps (*i.e.* there is no shift applied to $\mathbf{k}_{l+1}$). Note also that for $l \geq 1$, given that α_0 divides δ_0, the increment applied to the index $\mathbf{i}_{l+1}$ is integer as required. For $l = 0$ and $\alpha_0 = \delta_0$ (as it is usually the case), the shift $\delta_0/2\alpha_0 = 1/2$ is fractional. In this case, however, the additional offset $\tau_1 = -1/2$ restores integer coordinates as needed.

3 Experiments

The primary goal of this section is to demonstrate the benefit of learning the DM parameters using backpropagation compared to hand-tuning. There are several implementations of DM available online; we base ours on the GPU-based version by the original authors[2] [19], except for the decoding stage for which we use their CPU version with memoization removed. We do so because this eliminats a few small approximations found in the original code. This version is the closest, and in fact numerically equivalent, to our implementation using MatConvNet [24] and our new convolutional operators.

Datasets. The **MPI Sintel [4]** dataset contains 1,041 image pairs and correspondence fields obtained from synthetic data (computer graphics). Scenes are carefully engineered to contain challenging conditions. There are two versions: clean and final (with effects such as motion blur and fog). We consider a subset of the Sintel clean training set to evaluate our methodology. This is dubbed SintelMini, and consists of 7 sequences (313 images) for training and every 10th frame from a different set of 5 sequences (25 images) for validation. The **FlyingChair dataset** by Fischer *et al.* [5] contains synthetically-generated data as Sintel, but with abstract scenes consisting of "flying chairs". It consists of respectively 22,232/640 train/val image pairs and corresponding flow fields. These images are generated by rendering 3D chair models in front of random background images from Flickr, while the motions of both the chairs and the background are purely planar. The **KITTI flow 2012 [6, 15]** dataset contains 194/195 training/testing image pairs and correspondence fields for road scenes. The data contains large baselines but only motions arising from driving a car. Ground truth correspondences are obtained using 3D laser scanner and hence are not available at all pixels.

[2] http://lear.inrialpes.fr/src/deepmatching/.

	Patch descr.	Training set	Elements learned expon.	Elements learned features	Acc@2	Acc@5	Acc@10	EPE (matches)	EPE (flow)
(a)	HOG	—	×	×	84.52%	91.89%	94.36%	3.83	1.88
(b)	HOG	Sintel Mini	✓	×	84.59%	92.03%	94.49%	3.73	1.84
(c)	CNN	—	×	×	85.28%	92.25%	94.83%	3.58	1.80
(d)	CNN	Sintel Mini	✓	×	85.30%	92.27%	94.87%	3.70	1.64
(e)	CNN	Sintel Mini	×	✓	**86.81%**	92.52%	94.86%	3.37	1.60
(f)	CNN	Sintel Mini	✓	✓	86.79%	**92.58%**	**94.90%**	3.34	**1.57**
(g)	CNN	Flying Chairs	✓	✓	86.11%	92.47%	94.88%	**3.33**	1.65

Table 1: **Fully-Trainable DM performance.** DM variants evaluated on Sintel Mini (see text) validation and trained on either Sintel Mini training or Flying Chairs. The top row corresponds to the baseline DM algorithm, equivalent to the GPU version of [18].

Furthermore, the flow is improved by fitting 3D CAD models to observed vehicles on the road and using those to compute displacements.

Evaluation metrics. In order to measure matching accuracy, we adopt the **accuracy@T** metric of Revaud *et al*. [19]. Given the ground truth and estimated dense correspondence fields $\Gamma_0, \Gamma : \mathbf{p} \mapsto \mathbf{q}$ from image $I_0 : \Omega_0 \to \mathbb{R}$ to image $I_1 : \Omega_1 \to \mathbb{R}$, accuracy@T is the fraction of pixels in Ω_0 correctly matched up to an error of T pixels, *i.e.* $|\{\mathbf{q} \in \Omega_0 : \|\Gamma_0(\mathbf{q}) - \Gamma_0(\mathbf{q})\| \leq T\}|/|\Omega_0|$.[3] In addition to accuracy@T, we also consider the **end point error** (EPE), obtained as the average correspondence error $\text{mean}_{\mathbf{q}\in\Omega_0} \|\Gamma(\mathbf{q}) - \Gamma_0(\mathbf{q})\|$. In all cases, scores are averaged over all image pairs to yield the final result for a given dataset. If ground truth correspondences are available only at a subset of image locations, Ω_0 is restricted to this set in the definitions above. For the KITTI dataset, we report in particular results for Ω_0 restricted to non-occluded ares (NOC) and all areas (OCC).

Implementation details. For DM, unless otherwise stated we use $L = 6$ layers, $R = 80$ pixels, $\alpha_0 = \delta = 8$, $\beta_0 = 4$, $\gamma_0 = 1$, $\eta_0 = 1.4$. Training uses an NVIDIA Titan X GPU with 12 GBs of on-board memory. Training uses stochastic gradient descent with momentum with mini-batches comprising one image pair at a time (note that an image pair can be seen as the equivalent of a very large batch of image patches).

3.1 Results

End-to-end DM training. In our first experiment (Table 1) we evaluate several variants of DM training. To do so, we consider the smaller and hence more efficient Sintel Mini dataset, a subset of Sintel described above. In Table 1 (a) vs (b) we compare using the default value of $\nu = 1.4$ used to modulate the output of the aggregation layers and learning values $\nu_l, l = 1, \ldots, L$ specific for each layer. Even with this simple change there is a noticeable improvement (+0.13% acc@10). Next, we replace the HOG features with a trainable CNN architecture ϕ to extract descriptors from image patches. We use the first four convolutional layers (conv1_1, conv1_2, conv2_1, conv2_2) of the pre-trained VGG-VD network [22]. Just by replacing the features, we notice a further improvement ((a) vs (c) +0.47% acc@10) of DM, which can be increased by learning the DM exponents (d). Most interestingly, in (f) we obtain a further improvement by back-propagating from DM to the feature extraction

[3] Following [19], the quasi-dense DM matches are first filtered by reciprocal verification and then correspondences are propagated to all pixels by assigning to each point $\mathbf{q}$ the same displacement vector of the most confident available nearest available neighbor $\mathbf{q}'$ within a L^∞-radius of 8 pixels by setting $\Gamma(\mathbf{q}) = \Gamma(\mathbf{q}') - \mathbf{q}' + \mathbf{q}$.

Method	Training	Test	Acc@2	Acc@5	Acc@10	EPE (matches)	EPE (flow)	Err-OCC (flow 3px)
FlowNet S+v [5]	Flying Chairs	KITTI12	-	-	-	-	6.50	-
DM-HOG	—	KITTI12	60.50%	79.34%	84.27%	11.39	**3.59**	16.56%
DM-CNN	—	KITTI12	61.21%	78.81%	84.01%	12.29	4.11	17.78%
DM-CNN	Flying Chairs	KITTI12	**63.90%**	**80.11%**	**84.71%**	**11.12**	3.61	**16.41%**
FlowNet S+v [5]	Flying Chairs	Sintel Final	-	-	-	-	4.76	-
DM [19]	—	Sintel Final	-	-	89.2%	-	4.10	-
DM-HOG	—	Sintel Final	74.37%	85.26%	89.39%	7.08	3.72	11.44%
DM-CNN	—	Sintel Final	75.15%	85.42%	89.48%	7.03	3.63	11.52%
DM-CNN	Flying Chairs	Sintel Final	**76.55%**	**86.22%**	**90.03%**	**6.77**	**3.50**	**11.10%**
FlowNet C+v [5]	Flying Chairs	Sintel Clean	-	-	-	-	3.57	-
DM-HOG	—	Sintel Clean	82.51%	90.18%	92.70%	5.26	2.32	7.00%
DM-CNN	—	Sintel Clean	83.03%	90.24%	92.87%	5.22	2.25	6.85%
DM-CNN	Flying Chairs	Sintel Clean	**84.16%**	**90.85%**	**93.31%**	**4.78**	**2.14**	**6.51%**

Table 2: **Performance comparison.** We train DM variants on large-scale synthetic dataset Flying Chairs, and evaluate on KITTI 12 train and Sintel train. Acc@n [19] assigns each pixel a nearby match, measuring the proportion correct within n pixels. EPE (endpoint error) is the mean euclidean distance between estimated flow vectors and the ground truth (considering just pixels where ground truth is available). EPE (matches) is computed only at the positions where we have our quasi-dense matches. EPE (flow) measures the endpoint error for the flow estimation, where flow is produced by post-processing the matches with EpicFlow [18]. Err-OCC likewise measures the dense flow, giving proportion of flows off by more than 3 pixels. The version excluding occlusions, Err-NOC, is given in the text.

layers and optimizing the features themselves (hence achieving end-to-end training from the raw pixels to the matching result). The last experiment (g) shows that similar improvements can also be obtained by training from completely unrelated datasets, namely Flying Chairs, indicating that learning generalizes well.

Standard benchmark comparisons. To test DM training in realistic scenarios, we evaluate performance on two standard benchmarks, namely the Sintel and KITTI 2012 training sets (Table 2) as these have publicly-available ground truth to compute accuracy. For training, we use Flying Chairs, which is designed to be statistically similar to the Sintel target dataset. Compared to the HOG-DM baseline, training the CNN patch descriptors in DM improves accuracy@10 by +0.44% on KITTI and by +0.64% on Sintel Final.

An application of DM is optical flow, where it is usually followed by interpolation and refinement such as Brox and Malik [2] or EpicFlow [18]. We use EpicFlow to interpolate our quasi-dense matches and compare the EPE results of FlowNet [5]. While there are better methods than FlowNet for optical flow estimation, we choose it for comparison as this was proposed as a fully-trainable CNN for dense image matching; we compare to their results using variational refinement (+v) which is similar to EpicFlow interpolation. We train our method on Flying Chairs to allow a direct comparison with the results reported in [5].

Compared to the pretrained CNN, training further on Flying Chairs gives a notable improvement in EPE, decreasing from 3.63 to 3.50 for Sintel Final and from 4.11 to 3.61 for KITTI. Compared to HOG, the improvement is even greater for Sintel Final, a gap of 0.22 pixels, however for KITTI the CNN is initially worse than HOG. Training on synthetic data improves most metrics on KITTI, with the exception of EPE (flow). We believe the latter result to be due to the fact that the EpicFlow refinement step, which is not trained, is not optimally tuned to the different statistics of the improved quasi-dense matches. The refinement step is in fact known to be sensitive to the data statistics (for example, in [18] different tunings are used for different datasets). If we exclude occlusions in the ground truth for KITTI,

our trained CNN gets EPE-NOC of 1.43 compared to 1.51 for HOG, and Err-NOC falls from 7.84% to 7.41%.

FlowNet EPEs on KITTI12-Train and Sintel Final Train are respectively 6.50 and 4.76, whereas our trained DM-CNN model has EPEs of 3.61 and 3.50 respectively. This confirms the benefit of the DM architecture, which we turn into a CNN in this paper.

4 Summary

In this paper, we have shown that the *complete* DM algorithm can be equivalently rewritten as a CNN with a U-topology, involving a number of new CNN layers. This allows to learn end-to-end the parameters of DM using backpropagation, including the CNN filters that extract the patch descriptors, robustly improving the quality of the correspondence extracted in a number of different datasets.

Once formulated as a modular CNN, components of DM can be easily replaced with new ones. For instance, the max pooling and unpooling units could be substituted with soft versions, resulting in denser score maps, which could result in easier training and in the ability of better expressing the confidence of dense matches. We are currently exploring a number of such extensions.

For the problem of optical flow estimation, it is still required to have EpicFlow as a post-processing step. This type of two-stage approach results a suboptimal solution. In particular, the parameters of EpicFlow are not optimized by end-to-end training with our DM. We would like to explore a solution that allows end-to-end optical flow estimation.

Acknowledgements. This work was supported by the AIMS CDT (EPSRC EP/L015897/1) and grants EPSRC EP/N019474/1, EPSRC EP/I001107/2, ERC 321162-HELIOS, and ERC 677195-IDIU. We gratefully acknowledge GPU donations from NVIDIA.

References

[1] Michael J. Black, Yaser Yacoob, Allan D. Jepson, and David J. Fleet. Learning parameterized models of image motion. In *IEEE CVPR*, 1997.

[2] Thomas Brox and Jitendra Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE TPAMI*, 33(3):500–513, 2011.

[3] Thomas Brox, Andres Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[4] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

[5] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE ICCV*, 2015.

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE CVPR*, 2012.

[7] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *IEEE CVPR*, 2015.

[8] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(3):185–203, 1981.

[9] Ryan Kennedy and Camillo J. Taylor. Optical flow with geometric occlusion estimation and fusion of multiple frames. In *EMMCVPR*, 2015.

[10] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998.

[11] Marius Leordeanu, Andrei Zanfir, and Cristian Sminchisescu. Locally affine sparse-to-dense matching for motion and occlusion estimation. In *IEEE ICCV*, 2013.

[12] Jon Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE CVPR*, 2015.

[13] Jonathan Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *NIPS*, 2014.

[14] Etienne Mèmin and Patrick Pèrez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE TIP*, 7(5):703–719, 1998.

[15] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *IEEE CVPR*, 2015.

[16] Moritz Menze, Christian Heipke, and Andreas Geiger. Discrete optimization for optical flow. In *GCPR*, 2015.

[17] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *IEEE ICCV*, 2015.

[18] Jérôme Revaud, Philippe Weinzaepfel, Zaïd Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *IEEE CVPR*, 2015.

[19] Jérôme Revaud, Philippe Weinzaepfel, Zaïd Harchaoui, and Cordelia Schmid. Deepmatching: Hierarchical deformable dense matching. *IJCV*, 2015.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[21] Dan Rosenbaum, Daniel Zoran, and Yair Weiss. Learning the local statistics of optical flow. In *NIPS*, 2013.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[23] Deqing Sun, Stefan Roth, J.P. Lewis, and Michael J. Black. Learning optical flow. In *ECCV*, 2008.

[24] Andrea Vedaldi and Karel Lenc. MatConvNet: Convolutional neural networks for MATLAB. In *ACM MM*, 2015.

[25] Andreas Wedel, Daniel Cremers, Thomas Pock, and Horst Bischof. Structured motion-adaptive regularization for high accuracy optical flow. In *IEEE ICCV*, 2009.

[26] Philippe Weinzaepfel, Jérôme Revaud, Zaïd Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *IEEE ICCV*, 2013.

[27] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *IEEE CVPR*, 2015.

[28] Jure Žbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *The Journal of Machine Learning Research*, 17 (65):1–32, 2016.

[29] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *IEEE CVPR*, 2010.

Chapter 4

Unsupervised learning of object landmarks by factorized spatial embeddings

This work was accepted for Oral Presentation at the International Conference on Computer Vision (ICCV), Venice, 2017

This paper introduces the idea of an *Object Frame* defined in terms of sparse landmarks. In this setting it is seen as a function, implemented by a CNN, which outputs a fixed number of heatmaps whose peaks correspond to concepts that can be consistently located in all images of the object category. It also introduces the concept of viewpoint factorization used in later work, as well as the training procedure and method for obtaining synthetically warped pairs of images which is largely shared across Chapters 4-7. We show that the proposed technique works qualitatively well for both rigid and deformable objects, and generalises automatically to entire object categories despite only learning from correspondences between warped versions of a single instance. We also show that the predicted landmarks can be used to train a linear regressor with a small amount of manual supervision, proving that the unsupervised landmarks are highly predictive of annotated ones.

Unsupervised learning of object landmarks by factorized spatial embeddings

James Thewlis
University of Oxford
jdt@robots.ox.ac.uk

Hakan Bilen
University of Oxford
University of Edinburgh
hbilen@robots.ox.ac.uk

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

Abstract

Learning automatically the structure of object categories remains an important open problem in computer vision. In this paper, we propose a novel unsupervised approach that can discover and learn landmarks in object categories, thus characterizing their structure. Our approach is based on factorizing image deformations, as induced by a viewpoint change or an object deformation, by learning a deep neural network that detects landmarks consistently with such visual effects. Furthermore, we show that the learned landmarks establish meaningful correspondences between different object instances in a category without having to impose this requirement explicitly. We assess the method qualitatively on a variety of object types, natural and man-made. We also show that our unsupervised landmarks are highly predictive of manually-annotated landmarks in face benchmark datasets, and can be used to regress these with a high degree of accuracy.

1. Introduction

The appearance of objects in images depends strongly not only on their intrinsic properties such as shape and material, but also on accidental factors such as viewpoint and illumination. Thus, learning from images about objects as intrinsic physical entities is extremely difficult, particularly if no supervision is provided.

Despite these difficulties, the performance of object detection algorithms has been rising steadily, and deep neural networks now achieve excellent results on benchmarks such as PASCAL VOC [17] and Microsoft COCO [39]. Still, it is unclear whether these models conceptualise objects as intrinsic entities. Early object detectors such as HOG [13] and DPMs [18] were based on 2D templates applied in a translation and scale invariant manner to images. Recent detectors such as SSD [42] make this even more extreme and learn different templates (filters) for different scales and even different aspect ratios of objects. Hence, these models are likely to capture objects as image-based phenomena, representing them as a collection of weakly-related 2D patterns.

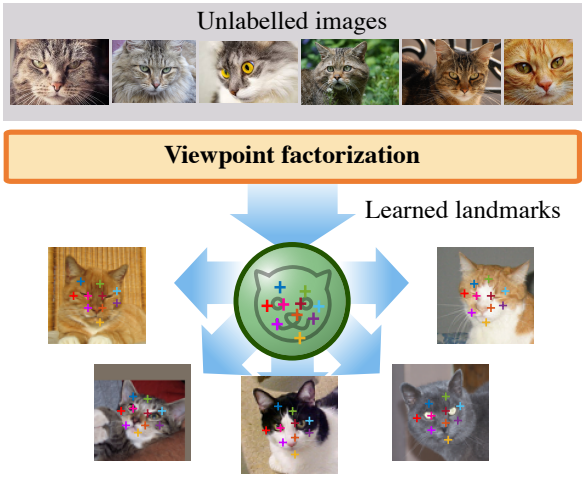


Figure 1. We present a novel method that can learn **viewpoint invariant landmarks without any supervision**. The method uses a process of viewpoint factorization which learns a deep landmark detector compatible with image deformations. It can be applied to rigid and deformable objects and object categories.

Achieving a deeper understanding of objects requires modeling their intrinsic viewpoint-independent structure. Often this structure is defined manually by specifying entities such as landmarks, parts, and skeletons. Given sufficient manual annotations, it is possible to teach deep neural networks and other models to recognize such structures in images. However, the problem of *learning such structures without manual supervision* remains largely open.

In this paper, we contribute a new approach to learn viewpoint-independent representations of objects from images without manual supervision (fig. 1). We formulate this task as a *factorization problem*, where the effects of image deformations, for example arising from a viewpoint change, are explained by the motion of a reference frame attached to the object and independent of the viewpoint.

After describing the general principle (sec. 3.1), we in-

vestigate a particular instantiation of it. In this model, the structure of an object is expressed as a set of *landmark points* (sec. 3.2) detected by a neural network. Differently from traditional keypoint detectors, however, the network is learned without manual supervision. Learning considers pairs of images related by a warp and requires the detector's output to be *equivariant* with the transformation (sec. 3.3). Transformations could be induced by real-world viewpoint changes or object deformations, but we show that meaningful landmarks can be learned even by considering random perturbations only.

We show that this method works for individual rigid and deformable object *instances* (sec. 3.1.1) as well as for object *categories* (sec. 3.1.2). This only requires learning a single neural network to detect the same set of landmarks for images containing different object instances of a category. While there is no *explicit* constraint that forces landmarks for different instances to align, we show that, in practice, this tends to occur automatically.

The method is tested qualitatively on a variety of different object types, including shoes, animals, and human faces (sec. 4). We also show that the unsupervised landmarks are highly predictive of manually-annotated landmarks, and as such can be used to detect these with a high degree of accuracy. In this manner, our method can also be used for unsupervised pretraining of semantic landmark detectors.

2. Related work

Flow. Matching images up to a motion-induced deformation links back to the work of Horn and Schunck [26] on optical flow and to deep learning approaches for its computation [21, 57, 28]. Flow can also be defined semantically rather than geometrically [40, 32, 46, 77, 76]. While our method also establishes geometric and (indirectly) semantic correspondences, it goes beyond that by learning a single set of viewpoint independent landmarks which are valid for *all* images at once.

Parts. A traditional method to describe the structure of objects is to decompose them into their constituent parts. Several unsupervised methods to learn parts exist, from the constellation approach used in [19, 9, 62] to the Deformable Parts Model (DPM) [18] and many others. More recently, AnchorNet [48] successfully learns parts that match different object instances as well as different object categories using only image-level supervision; furthermore, they propose a part orthogonality constraint similar to our own. While the concepts of landmarks and parts are similar, our training method differs substantially from these approaches: rather than learning parts as a byproduct of learning a (deformable) discriminator, our landmark points are trained to fit geometric deformations directly.

Deformation-prediction networks. *WarpNet* [30] learns a neural network that, given two images, predicts a Thin Plate Spline (TPS [6]) that aligns them. While our landmarks can also be seen as a representation of transformations (as matching them between image pairs induces one), learning such landmarks is unique to our method. The Deep Deformation Network of [69] predicts image transformations to refine landmarks using a "Point Transformer Network", but their landmarks are learned using full manual supervision, whereas our method is fully unsupervised. Very recently [53] learn a neural network that also aligns two images by estimating the transformation between them, implicitly learning feature extractors that could be similar to keypoints; however, our work explicitly trains a network to output keypoints that are equivariant to such transformations.

Landmark detection. There is an extensive literature on landmark detectors, particularly for faces. Examples include Active Appearance Models [11], along with subsequent improvements [44, 12] and others using templates [51] or parts [80]. Other approaches directly regress the landmark coordinates [59, 14, 10, 52]. Deep learning methods use cascaded CNNs [56], coarse-to-fine autoencoders [70], auxiliary attribute prediction [73, 74], learned deformations [69] and LSTMs [64]. Beyond faces, there is work on humans [65, 58], birds [55, 41, 69] and furniture [63]. More general pose estimation including the case of landmarks is explored in [16]. Our method can build on any such detector architecture and can be used as a pretraining strategy to learn landmarks with less or no supervision.

Equivariance constraint. A variant of the equivariance constraint used by our method was proposed by [37] to learn feature point detectors for image matching. We build on a similar principle, but use it to learn intrinsic landmarks for object categories instead of generic SIFT-like features with a robust learning objective and learn to detect a set of complementary landmarks rather than a single one at a time.

Unsupervised pretraining. Unsupervised pretraining has received significant interest with the popularization of data-hungry deep networks [5, 24, 23]. Unsupervised learning is based on training a network to solve auxiliary tasks, for which supervision can be obtained without manual annotations. The most common of such tasks is to *generate* the data (autoencoders [7, 4, 25]); or one can remove some information in images and train a network to reconstruct it (denoising [60], ordering patches [15, 47], inpainting [50], analyzing motion [1, 49, 61, 20, 45], and colorizing [71, 35]). Our method can be seen in this light as trying to undo a synthetic deformation applied to an image.

Our method is also related to unsupervised learning for faces, such as alignment based on a face model [78], learning meaningful descriptors [67, 22], and learning a part model [38]. Huang *et al*. [27] learn joint alignment of faces using deep features, and Jaiswal *et al*. [29] use clustering to discover head modes in order to refine manually-defined landmarks in an unsupervised manner, both using genera-

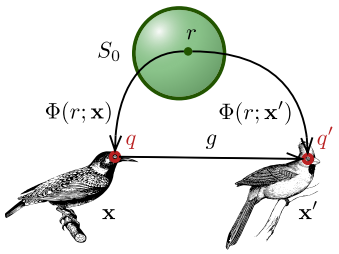


Figure 2. **Modelling the structure of objects.** Points r in the reference space S_0 (conceptually a sphere) index corresponding points in different object instances. Given an image $\mathbf{x}$, the map $\Phi(r;\mathbf{x})$ detects the location q of the reference point r. The map must be compatible with warps g of the objects. For different views of the same (deformable) object instance, the warp g is defined geometrically, whereas for object categories (as shown) it is defined semantically.

tive principles. None of these methods learns landmarks from scratch.

3. Method

Sec. 3.1 introduces the method of viewpoint factorization for learning an intrinsic reference frame for object instances and categories. Then, sec. 3.2 applies it to learn object landmarks and sec. 3.3 discusses the details of the learning formulation.

3.1. Structure from viewpoint factorization

Let $S \subset \mathbb{R}^3$ be the surface of a physical object, say a bird, and let $\mathbf{x} : \Lambda \to \mathbb{R}$ be an image of the object, where $\Lambda \subset \mathbb{R}^2$ is the image domain (fig. 2). The surface S is an intrinsic property of the object, independent of the particular image $\mathbf{x}$ and of the corresponding viewpoint. We consider the problem of learning a function $q = \Phi_S(p;\mathbf{x})$ that maps object points $p \in S$ to the corresponding pixels $q \in \Lambda$ in the image.

We propose a new method to learn Φ_S automatically through a process of viewpoint factorization. To this end, consider a second image $\mathbf{x}'$ of the object seen from a different viewpoint. Occlusion not withstanding, one can write $\mathbf{x}' \approx \mathbf{x} \circ g$ where $g : \mathbb{R}^2 \to \mathbb{R}^2$ is the image warp induced by the viewpoint change. Using the map Φ_S, the warp g can be factorised as follows:

$$g = \Phi_S(\cdot;\mathbf{x}') \circ \Phi_S(\cdot;\mathbf{x})^{-1}. \tag{1}$$

In other words, we can decompose the warp $g : q \mapsto q'$ as first finding the intrinsic object point $p = \Phi_S^{-1}(q;\mathbf{x})$ corresponding to pixel q in image $\mathbf{x}$ and then finding the corresponding pixel $q' = \Phi_S(p;\mathbf{x}')$ in image $\mathbf{x}'$.

The factorization eq. (1) is more conveniently expressed as the following *equivariance constraint*:

$$\forall p \in S : \ \Phi_S(p;\mathbf{x} \circ g) = g(\Phi_S(p;\mathbf{x})). \tag{2}$$

This constraint simply states that the points p must be detected in a manner which is consistent with a viewpoint change.

In order to learn the map Φ_S, we express the latter as a deep neural network and train it to satisfy constraint (2) in a Siamese configuration, supplying triplets $(\mathbf{x}, \mathbf{x}', g)$ to the learning process. Note that, if we are given two views $\mathbf{x}$ and $\mathbf{x}'$ of the same object, the viewpoint transformation g is often unknown. Instead of trying to recover g, inspired by [30], we propose to synthesize transformations g at random and use them to generate $\mathbf{x}'$ from $\mathbf{x}$. While this approach only uses unannotated images of the object, it can still learn meaningful landmarks (sec. 4).[1]

Discussion. While learning only considers deformations of the same image, the model still learns to bridge automatically across moderately different viewpoints (see fig. 5). However we leave very large out-of-plane rotations, which would require to handle partial occlusions of the landmarks, to future work.

3.1.1 Deformable objects

The method developed above extends essentially with no modification to deformable objects. Suppose that the surface S deforms between images according to isomorphisms $w : \mathbb{R}^3 \to \mathbb{R}^3$. We tie the shape variants $wS = \{w(p) : p \in S\}$ together by introducing a common reference space S_0, which we call an *object frame*. Barring topological changes, we can establish isomorphisms π_S mapping reference points $r \in S_0$ to fixed surface points $\pi_S(r) \in S$, in the sense that $\forall w : w(\pi_S(r)) = \pi_{wS}(r)$. Then, by using the substitution $\Phi(r;\mathbf{x}) = \Phi_S(\pi_S(r);\mathbf{x})$, we can rewrite the equivariance constraint (2) as

$$\forall r \in S_0 : \ \Phi(r;\mathbf{x} \circ g) = g(\Phi(r;\mathbf{x})). \tag{3}$$

This simply states that one expects surface points to be detected equivariantly with viewpoint-induced deformations as well as with deformations of the object surface.

3.1.2 Object categories

In addition to deformable objects, our formulation can easily account for shape variations between object instances in the same category. To do this, one simply makes the assumption that all object surfaces S are isomorphic to the same reference shape S_0 (fig. 2).

Differently from the case of deformable objects, geometry alone does not force the mappings π_S for different object

[1] If $\mathbf{x}$ and $\mathbf{x}'$ are given but g is unknown, one can rewrite eq. (2) by expressing the warp g as a function of the predicted landmarks (as the solution of the equation $\forall p : \Phi_S(p;\mathbf{x}') = g\Phi_S(p;\mathbf{x})$), and then by measuring the alignment quality in appearance space as $\|\mathbf{x}' - \mathbf{x} \circ g\|$. However, this approach provides a weaker supervisory signal and is somewhat more complex to implement.

instances S to be related. Nevertheless, we would like to choose such mappings to be *semantically consistent*; for example, if $\pi_S(r)$ is the right eye of face S, then we would like $\pi_{S'}(r)$ to be the right eye of face S'. An important contribution of this work is to show that semantically-meaningful correspondences emerge automatically by simply sharing *the same learned mapping* Φ between all object instances in a given category. The idea is that, by learning a single rule that detects object points consistently with deformations, these points tend to align between different object instances as this is the smoothest solution.

3.2. Landmark detection networks

In this section we instantiate concretely the method of sec. 3.1. First, one needs to decide how to represent the maps $\Phi(\cdot;\mathbf{x}) : S_0 \to \Lambda$ as the output of a neural network or other computational model. Our approach is to *sample* this function at a set of K discrete reference locations $\Phi(\mathbf{x}) = (\Phi(r_1;\mathbf{x}), \dots, \Phi(r_K;\mathbf{x}))$. In this manner, the function $\Phi(\mathbf{x})$ can be thought of as detecting the location $p_k = \Phi(r_k;\mathbf{x})$ of K *object landmarks*. We do not attach particular constraints to the set of landmarks, which can be thought of as an index set $r_k = k$, $k = 1, 2, \dots, K$.

If Φ is implemented as a neural network, one can use any of the existing architectures for keypoint detection (sec. 2). Most such architectures are based on estimating *score maps* $\Psi(\mathbf{x}) \in \mathbb{R}^{H\times W\times K}$, associating a score $\Psi(\mathbf{x})_{uk}$ to each landmark r_k and image location $u \in \{1,\dots,H\} \times \{1,\dots,W\} \subset \mathbb{R}^2$. The score maps can be transformed into probability maps by using the *softmax* operator σ:

$$p(u|\mathbf{x}, r) = \sigma[\Psi(\mathbf{x})]_{ur} = \frac{e^{\Psi(\mathbf{x})_{ur}}}{\sum_v e^{\Psi(\mathbf{x})_{vr}}}.$$

Following [66], it is then possible to extract a landmark location by using the soft argmax operator, which computes the expected value of this density:

$$u^*_r = \sigma_{\text{arg}}[\Psi(\mathbf{x})]_r = \sum_u u\, p(u|\mathbf{x}, r) = \frac{\sum_u u e^{\Psi(\mathbf{x})_{ur}}}{\sum_v e^{\Psi(\mathbf{x})_{vr}}}.$$

The overall network, computing the location of the K landmarks, can then be expressed as

$$\Phi(\mathbf{x}) = \sigma_{\text{arg}}[\Psi(\mathbf{x})]. \tag{4}$$

Discussion. An alternative approach for representing the maps $S_0 \to \Lambda$ is to predict the parameters of a parametric transformation t. Assuming that the reference set $S_0 \subset \mathbb{R}^2$ is a space of continuous coordinates, the transformation t could be an affine one [37] or a thin plate spline (TPS) [30]. This has the advantage of capturing in one step a dense set of object points and can be used to impose smoothness on the map.

However, using discrete landmarks is more robust and general. For example, individual landmarks may be undetectable when occluded, and this model can handle this case more easily without disrupting the estimate of the visible landmarks. Furthermore, one does not need to make assumptions on the family of allowable transformations, which could be difficult in general.

3.3. Learning formulation

In this section, we show how the equivariance constraint (3) can be used to learn Φ from examples. The idea is to setup the learning problem as a *Siamese* configuration, in which the output of Φ on two images $\mathbf{x}$ and $\mathbf{x}'$ is assessed for compatibility with respect to the deformation g and the equivariance constraint (3). We can express this condition as the loss term:

$$\mathcal{L}_{\text{align}} = \frac{1}{K}\sum_{r=1}^{K} \|\Phi(\mathbf{x}\circ g)_r - g(\Phi(\mathbf{x})_r)\|^2. \tag{5}$$

In the rest of the section, we discuss two extensions to eq. (5) that allow the system to train better landmarks: formulating the loss directly in terms of the keypoint probabilities and adding a diversity term.

Probability maps loss. Equation (5) uses the soft argmax operator in order to localise and then compare landmarks. We show here that one can skip this step by writing a loss directly in terms of the probability maps, which provides a more direct and stable gradient signal. The idea is to replace eq. (5) with the loss term

$$\mathcal{L}'_{\text{align}} = \frac{1}{K}\sum_{r=1}^{K}\sum_{uv} \|u - g(v)\|^2 p(u|\mathbf{x}, r) p(v|\mathbf{x}', r) \tag{6}$$

where $p(u|\mathbf{x}, r) = \sigma[\Psi(\mathbf{x})]_{ur}$ and $p(v|\mathbf{x}', r) = \sigma[\Psi(\mathbf{x}')]_{vr}$ are the landmark probability maps extracted from images $\mathbf{x}$ and $\mathbf{x}'$.

Minimizing loss (6) has two desirable effects. First, it encourages the two probability maps to overlap and, second, it encourages them to be highly concentrated. In fact, the loss is zero if, and only if, both p and q are delta functions *and* if the corresponding landmark locations match up to g.

While a naive implementation of (6) requires to visit all pairs of pixels u and v in both images, with a quadratic complexity, a linear-time implementation is possible by decomposing the loss as:

$$\sum_u \|u\|^2 p(u|\mathbf{x}, r) + \sum_v \|g(v)\|^2 p(v|\mathbf{x}', r) - 2\left(\sum_u u\, p(u|\mathbf{x}, r)\right)^\top \cdot \left(\sum_v g(v) p(v|\mathbf{x}', r)\right).$$

Diversity loss. The equivariance constraint eq. (3) and its corresponding losses eqs. (5) and (6) ensure that the network learns at least one landmark aligned with image deformations. However, there is nothing to prevent the network from learning K identical copies of the same landmark.

In order to avoid this degenerate solution, we add a *diversity* loss that requires probability maps of different landmarks to fire in different parts of the image. The most obvious approach is to penalize the mutual *overlap* between maps for different landmarks r and r':

$$\mathcal{L}_{\text{div}}(\mathbf{x}) = \frac{1}{K^2} \sum_{r=1}^{K} \sum_{r'=1}^{K} \sum_{u} p(u|\mathbf{x}, r) p(u|\mathbf{x}, r'). \quad (7)$$

This term is zero only if, and only if, the support of the different probability maps is disjoint.

The disadvantage of this approach is that it is *quadratic* in the number of landmarks. An alternative and more efficient diversity loss is:

$$\mathcal{L}'_{\text{div}}(\mathbf{x}) = \sum_{u} \left(\sum_{r=1}^{K} p(u|\mathbf{x}, r) - \max_{r=1,\dots,K} p(u|\mathbf{x}, r) \right). \quad (8)$$

Just like eq. (7), this loss is zero only if the support of the distributions is disjoint. In fact the sum of probability values at a given point u is always greater than the max unless all but one probability are zero. Note that we can rewrite (8) more compactly as:

$$\mathcal{L}'_{\text{div}}(\mathbf{x}) = K - \sum_{u} \max_{r=1,\dots,K} p(u|\mathbf{x}, r).$$

In practice, we found it beneficial to apply the diversity loss after *downsampling* (by $m \times m$ sum pooling) the probability maps as this encourages landmarks to be extracted farther apart. Thus we consider:

$$\mathcal{L}''_{\text{div}}(\mathbf{x}) = K - \sum_{u} \max_{r=1,\dots,K} \sum_{\delta_u} p(mu + \delta_u|\mathbf{x}, r).$$

where $\delta_u \in \{0, \dots, m-1\}^2$.

Learning objective. The learning objective considers triplets $(\mathbf{x}_i, \mathbf{x}'_i, g_i)$ of images $\mathbf{x}_i$ and $\mathbf{x}'_i$ related by a viewpoint warp g_i and optimizes:

$$\min_{\Psi} \lambda \mathcal{R}(\Psi) + \frac{1}{N} \sum_{i=1}^{N} \Big(\mathcal{L}'_{\text{align}}(\mathbf{x}_i, \mathbf{x}'_i, g_i; \Psi) + \gamma \mathcal{L}''_{\text{div}}(\mathbf{x}_i; \Psi) + \gamma \mathcal{L}''_{\text{div}}(\mathbf{x}'_i; \Psi) \Big), \quad (9)$$

where $\mathcal{R}$ is a regulariser (weight shrinkage for a neural network). As noted before, if triplets are not available, they can be *synthesized* by applying a random transformation g_i to an image $\mathbf{x}_i$ to obtain $\mathbf{x}'_i = \mathbf{x}_i \circ g$. Note that all functions are easily differentiable for backpropagation.

4. Experiments

In this section, we first describe the implementation details (sec. 4.1) and then report both qualitative (sec. 4.2) and quantitative (sec. 4.3) results demonstrating the power of our unsupervised landmark learning method.

4.1. Implementation details

In all the experiments, the detector Φ contains six convolutional layers with 20, 48, 64, 80, 256, K filters respectively, where K is the number of object landmarks. Each convolutional layer is followed by a batch normalization and a ReLU layer. This network is proposed in [74] for supervised facial keypoint estimation. Differently, instead of downsampling the feature map after each convolutional layer, we use only one 2×2 max pooling layer with a stride of 2 after the first convolutional layer (`conv1`). Thus, given an input size of $H \times W \times 3$, the network outputs an $\frac{H}{2} \times \frac{W}{2} \times K$ feature map. We apply a spatial softmax operator to the output of the last convolutional layer to obtain K probability maps, one for each landmark.

During training, we supply a set of triplets of $(\mathbf{x}_i, \mathbf{x}'_i, g_i)$ as input to the network. In order to generate them, given an example image $\mathbf{I}$, one can naively sample a random TPS and warp the image accordingly. However, as the input images are typically centered and at most very slightly rotated, the learned weights can be biased towards such a setting. Instead, we randomly sample two TPS transformations (g_1, g_2) and consecutively warp the given image to generate an image pair *i.e.* $\mathbf{x} = \mathbf{I} \circ g_1$ and $\mathbf{x}' = \mathbf{x} \circ g_2$ (computed using inverse image warping as $\mathbf{x} \circ (g_2 \circ g_1)$). The TPS warps are parametrized as in [6] which can be decomposed into affine and deformation parts. To render realistic and diverse warps, we randomly sample scale, rotation angle and translation parameters within the pre-determined ranges. Examples of the transformations are shown in figs. 3 to 5.

We initialize the weights of convolutions with random gaussian noise and optimize the objective function (eq. (9)) (weight decay $\lambda = 5 \cdot 10^{-4}$, $\gamma = 500$) by using Adam [33] with an initial learning rate 10^{-4} until convergence, then reduce it by one tenth until no further improvement is seen.

4.2. Qualitative results

We train our unsupervised landmarks from scratch on three different domains: shoes (fig. 3), cat faces (fig. 4), and faces (fig. 5), and assess them qualitatively. We train landmark detectors on 49525 shoes from the UT Zappos50k dataset of [68] and 8609 images from the cat heads dataset of [72] and keep the rest for validation. Facial landmarks are learned on the CelebA dataset [43] which contains more than 200k celebrity images for 10k identities with 5 annotated landmarks. We use the provided cropped face images, which are roughly centered and scaled to the same size.

We train an 8 or 10 landmark network for each of the tasks to allow for clearer visualization. In addition, we show

Figure 3. Unsupervised landmarks on shoes (8 landmark network). Top: synthetic TPS deformations (original image leftmost). Bottom: different instances. Note that landmarks are consistently detected despite the significant variation in pose, shape, materials, etc.

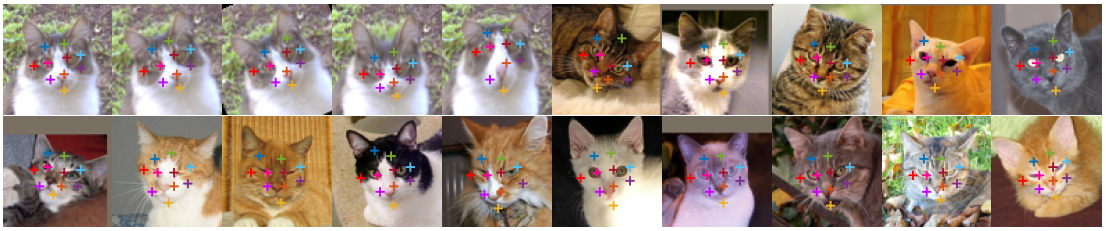


Figure 4. Unsupervised landmarks on cat faces (10 landmark network). Top-left quintuple: synthetic deformations (original image leftmost) transformed by rotation (images 2,3) and TPS warps (images 4,5). Remaining examples: different instances.

n landmarks	Regressor training	Mean error
10	MAFL	7.95
30	MAFL	7.15
50	MAFL	6.67
10	CelebA	6.32
30	CelebA	5.76
50	CelebA	**5.33**

Table 1. Results on MAFL test set in terms of the inter-ocular distance as in [74, 52]. For each setting, n unsupervised landmarks, that is learned on the CelebA training set, are regressed into 5 manually-defined landmarks. The regressor is learnt on CelebA or MAFL training set.

examples of a 30-landmark network for faces in fig. 6. In all cases we observe that: i) landmarks are detected consistently up to synthetic warps (affine or TPS) of the corresponding images and that ii) as a byproduct of learning to be consistent with such transformations, landmarks are very consistent across different object instances as well.

4.3. Quantitative results

In this section we evaluate the performance of our unsupervised landmarks quantitatively by testing how well they correlate with and predict manually-labelled landmarks. To do this, we consider standard facial landmark benchmarks containing manual annotations for semantic landmarks (e.g. eyes, corner of the mouth, etc). We first learn a detector for K landmarks without supervision, freeze its weights, and

Method	Mean Error
TCDCN [74]	7.95
Cascaded CNN [56]	9.73
CFAN [70]	15.84
Our Method (50 points)	**6.67**

Table 2. Comparison to state-of-the-art supervised landmark detectors on MAFL.

Method	Mean Error
RCPR [8]	11.6
Cascaded CNN [56]	8.97
CFAN [70]	10.94
TCDCN [74]	7.65
RAR [64]	7.23
Our Method (51 points)	10.53

Table 3. Comparison to state-of-the-art supervised landmark detectors on AFLW (5 pts) in terms of inter-ocular distance.

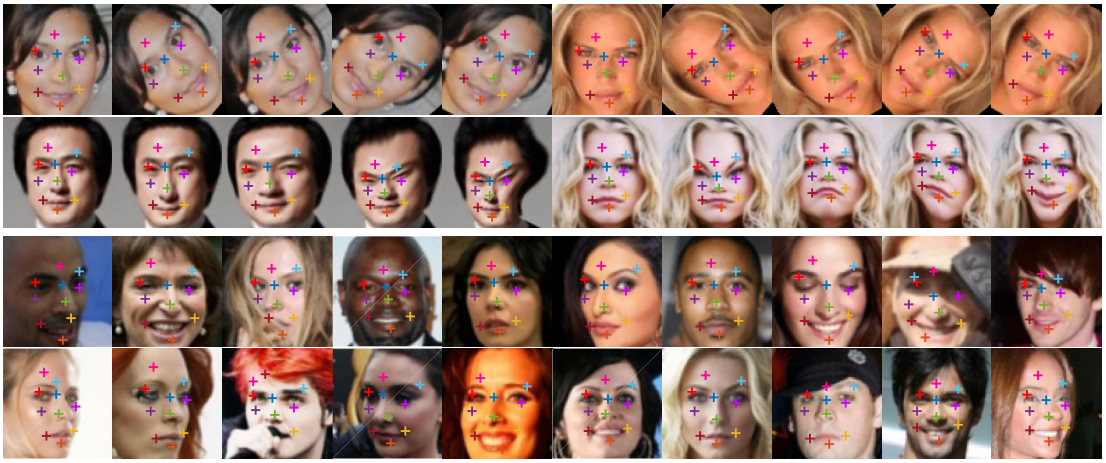


Figure 5. Unsupervised landmarks on CelebA faces (10 landmarks network). Top: synthetic rigid and TPS deformations (original image leftmost). Bottom: different instances. We observe landmarks highly aligned with facial features such as the mouth corners and eyes. Note that, being unsupervised, it needn't prefer the centers of the eyes, but consistently localizes points on the eye boundary.

Figure 6. Regression of supervised landmarks form 30 unsupervised ones (left in each pair) on MAFL. The green dot is the predicted annotation and a small blue dot marks the ground-truth. A failure case is shown to the right.

Supervised training images	Mean Error
All (19,000)	7.15
20	8.06
10	8.49
5	9.25
1	10.82

Table 4. Localization results for different number of training images from MAFL used for supervised training.

Method	Mean Error (68 pts)
DRMF [2]	9.22
CFAN [70]	7.69
ESR [10]	7.58
ERT [31]	6.40
LBF [52]	6.32
CFSS [79]	5.76
cGPRT [36]	5.71
DDN [69]	5.65
TCDCN [74]	5.54
RAR [64]	4.94
Ours (50 landmarks)	9.30
Ours (50 landmarks, finetune)	7.97

Table 5. Comparison to state-of-the-art supervised landmark detectors on 300-W.

then use the supervised training data in the benchmark to learn a linear regressor mapping the unsupervised landmark to the manually defined ones. The regressor takes as input the $2K$ coordinates of the unsupervised landmarks, stacks them in a vector $x \in \mathbb{R}^{2K}$, and maps the latter to the corresponding coordinates of the manually-defined landmarks as $y = Wx$. Learning W can be seen as a fully connected layer with no bias, and is trained similarly to the unsupervised network, using our warps as data augmentation. Note that there is no backpropagation to the unsupervised weights, which remain fixed. W is visualized in fig. 7.

Benchmark data. We first report results on the MAFL dataset [74], a subset of CelebA with 19k training images and 1k test images annotated with 5 facial landmarks (corners of mouth, eyes and nose). We follow the standard evaluation procedure in [74] and report errors in inter-ocular distance (IOD) in table 1. Since the MAFL test set and the CelebA training set overlap partially, we remove the MAFL test images from CelebA when the latter is used for training.

We also consider the more challenging 300-W dataset [54] containing 68 landmarks, obtained by merging and re-annotating other benchmarks. We follow [52] and use 3148 images from AFW [80], LFPW-train [3] and Helen-train [75] as training set, and 689 images from IBUG, LFPW-test and Helen-test as test set.

Finally we use the AFLW [34] dataset, which contains

24,386 faces from Flickr. Although it contains up to 21 annotated landmarks, we follow [74, 64] in only evaluating five and testing on the same 2995 faces cropped and distributed in the MTFL set of [73]. For training we use 10,122 faces that have all five points labelled and whose images are not in the test set.

MAFL results. First, we train the unsupervised landmarks on the CelebA training set and learn a corresponding regressor on the MAFL training set. The accuracy of the regressor on the MAFL test data is reported in table 1 and qualitative results are shown in fig. 6.

Regressing from $K = 10, 30, 50$ unsupervised landmarks improves the results. This can be explained by the fact that more unsupervised landmarks means a higher chance of finding some highly correlated with the five manually-labelled ones and thus a more robust mapping (fig. 7). This can also increase accuracy since our landmarks are detected with a resolution of two pixels (due to the downsampling in the network). Table 2 compares these results to state-of-the-art *fully supervised* landmark localization methods. Encouragingly, our best regressor outperforms the supervised methods (6.67 error rate vs 7.95 of TCDCN [74]). This shows that our unsupervised training method is indeed able to find meaningful landmarks.

Next, in Table 4 we assess how many manual landmark annotations are required to learn the regressor. We consider the problem of regressing from $K = 30$ unsupervised landmarks and we observe that the regressor performs well even if only 10 or 20 images are considered (errors 8.5 and 8.06). By comparison, using all 19,000 training samples reduces the error to 7.15, which shows that most of the required information is contained in the unsupervised landmarks from the outset. This indicates that our method is very effective for **unsupervised pretraining** of manually annotated landmarks as well, and can be used to learn good semantic landmarks with few annotations.

300-W results. We use our best performing model, the 50 point network, trained unsupervised on CelebA, and report results in table 5 for two settings. In the first one, the unsupervised landmarks are learned on CelebA and only the regressor is learned on the 300-W training set; we obtain an error of 9.30. In the second setting, the unsupervised detector is fine-tuned (also without supervision) on the 300-W data to adapt the features to the target dataset. The fine-tuning lowers the error to 7.97 and yields a comparable result with the state-of-the-art supervised methods. This shows another strength of our method: our unsupervised learner can be used to adapt an existing network to new datasets, also without using labels.

AFLW results. Due to tighter face crops, we adapt our 50-landmark CelebA network, fine-tuning it first on similarly cropped CelebA images and then on the AFLW training set. The adapted network has 51 landmarks. We compare against other methods in table 3. Once more, landmarks linearly-regressed from the unsupervised ones are competitive with fully supervised detectors (10.53 vs 7.23). The regressor can be trained with as low as 1 or 5 labelled images almost saturating performance (errors 14.79 and 12.94 respectively). By comparison, the same architecture trained supervised from scratch using 5 and 10 labelled images with TPS data augmentation but no unsupervised pretraining has substantially higher 23.85 and 22.31 errors (achieved essentially by predicting the average landmark locations which has error 24.40).

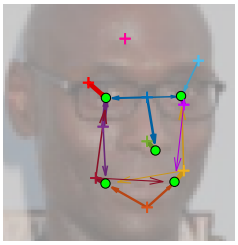


Figure 7. **Unsupervised $\leftrightarrow$ supervised landmark correlation.** The thickness of each arrow from our unsupervised landmarks (crosses) to the supervised ones (circles) represents the averaged magnitude of each contribution in the learned linear regressor.

We also visualize what the regressor learns and which of the source (discovered) landmarks contribute to the target (semantic) ones in fig. 7. To do so, for each target landmark, we take the corresponding column of the regressor, compute the absolute value of its coefficients, ℓ^1 normalize it, remove the entries smaller than 0.2. We show this mapping as a directional graph with arrows between the target landmarks (green circular nodes) and the source ones (colored crosses). We observe that the contributions are proportional to the distance between source and target points. In addition, the landmark on the forehead, not in the convex hull of the target points is ignored, as expected.

5. Conclusions

In this paper we have presented a novel approach to learn the structure of objects in an *unsupervised manner*. Our key contribution is to reduce this problem to the one of learning landmark detectors that are equivariant, i.e. compatible, with image deformations. This can be seen as a particular instantiation of the more general idea of *factorizing deformations* by learning an intrinsic reference frame for the object. We have shown that this technique works for rigid and deformable objects as well as object categories, it results in landmarks highly-predictive of manually annotated ones, and can be used effectively for pretraining.

Acknowledgments: This work acknowledges the support of the AIMS CDT (EPSRC EP/L015897/1) and ERC 677195-IDIU.

References

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

[2] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *Proc. CVPR*, 2013. 7

[3] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. *PAMI*, 2013. 7

[4] Y. Bengio. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2009. 2

[5] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. In *PAMI*, 2013. 2

[6] F. L. Bookstein. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *PAMI*, 1989. 2, 5

[7] H. Bourlard and Y. Kamp. Auto-Association by Multilayer Perceptrons and Singular Value Decomposition. *Biological Cybernetics*, 1988. 2

[8] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion-supp. mat. In *Proc. ICCV*, 2013. 6

[9] M. C. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *Proc. ECCV*, 1998. 2

[10] X. Cao, Y. Wei, F. Wen, and J. Sun. Face Alignment by Explicit Shape Regression. *IJCV*, 2014. 2, 7

[11] T. Cootes, G. Edwards, and C. Taylor. Active Appearance Models. *Proc. ICCV*, 1998. 2

[12] D. Cristinacce and T. Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 2008. 2

[13] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Proc. CVPR*, 2005. 1

[14] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool. Real-time facial feature detection using conditional regression forests. In *Proc. CVPR*, 2012. 2

[15] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised Visual Representation Learning by Context Prediction. In *Proc. ICCV*, 2015. 2

[16] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *Proc. CVPR*, 2010. 2

[17] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88(2), 2010. 1

[18] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *PAMI*, 2010. 1, 2

[19] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, 2003. 2

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

[21] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırba, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *Proc. ICCV*, 2015. 2

[22] M. K. Fleming and G. W. Cottrell. Categorization of faces using unsupervised feature extraction. In *International Joint Conference on Neural Networks*, 1990. 2

[23] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 2

[24] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 2006. 2

[25] G. E. Hinton and R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 2006. 2

[26] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 1981. 2

[27] G. Huang, M. Mattar, H. Lee, and E. G. Learned-Miller. Learning to align from scratch. In *Proc. NIPS*, 2012. 2

[28] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *Proc. CVPR*, 2017. 2

[29] S. Jaiswal, T. R. Almaev, and M. F. Valstar. Guided unsupervised learning of mode specific models for facial point detection in the wild. In *ICCV Workshops*, 2013. 2

[30] A. Kanazawa, D. W. Jacobs, and M. Chandraker. WarpNet: Weakly supervised matching for single-view reconstruction. In *Proc. CVPR*, 2016. 2, 3, 4

[31] V. Kazemi and J. Sullivan. One Millisecond Face Alignment with an Ensemble of Regression Trees. In *Proc. CVPR*, 2014. 7

[32] I. Kemelmacher-Shlizerman and S. M. Seitz. Collection flow. In *Proc. CVPR*, 2012. 2

[33] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015. 5

[34] 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 *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011. 7

[35] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *Proc. ECCV*, 2016. 2

[36] D. Lee, H. Park, and C. D. Yoo. Face alignment using cascade Gaussian process regression trees. In *Proc. CVPR*, 2015. 7

[37] K. Lenc and A. Vedaldi. Learning covariant feature detectors. In *ECCV Workshop on Geometry Meets Deep Learning*, 2016. 2, 4

[38] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang. Probabilistic elastic part model for unsupervised face detector adaptation. In *Proc. ICCV*, 2013. 2

[39] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proc. ECCV*, 2014. 1

[40] C. Liu, J. Yuen, and A. Torralba. SIFT Flow: Dense correspondence across scenes and its applications. *PAMI*, 2011. 2

[41] J. Liu and P. N. Belhumeur. Bird part localization using exemplar-based models with enforced pose and subcategory consistency. *Proc. ICCV*, 2013. 2

[42] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *Proc. ECCV*, 2016. 1

[43] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proc. ICCV*, 2015. 5

[44] I. Matthews and S. Baker. Active Appearance Models Revisited. *IJCV*, 2004. 2

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

[46] H. Mobahi, C. Liu, and W. T. Freeman. A Compositional Model for Low-Dimensional Image Set Representation. *Proc. CVPR*, 2014. 2

[47] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. ECCV*, 2016. 2

[48] D. Novotny, D. Larlus, and A. Vedaldi. Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *Proc. CVPR*, 2017. 2

[49] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning Features by Watching Objects Move. In *Proc. CVPR*, 2017. 2

[50] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context Encoders: Feature Learning by Inpainting. In *Proc. CVPR*, 2016. 2

[51] M. Pedersoli, T. Tuytelaars, and L. Van Gool. Using a deformation field model for localizing faces and facial points under weak supervision. In *Proc. CVPR*, 2014. 2

[52] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 FPS via regressing local binary features. In *Proc. CVPR*, 2014. 2, 6, 7

[53] I. Rocco, R. Arandjelović, and J. Sivic. Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, 2017. 2

[54] 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, 2016. 7

[55] K. J. Shih, A. Mallya, S. Singh, and D. Hoiem. Part Localization using Multi-Proposal Consensus for Fine-Grained Categorization. In *Proc. BMVC*, 2015. 2

[56] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proc. CVPR*, 2013. 2, 6

[57] J. Thewlis, S. Zheng, P. H. S. Torr, and A. Vedaldi. Fully-Trainable Deep Matching. In *Proc. BMVC*, 2016. 2

[58] A. Toshev and C. Szegedy. DeepPose: Human pose estimation via deep neural networks. *Proc. CVPR*, 2014. 2

[59] M. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *Proc. CVPR*, 2010. 2

[60] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proc. ICML*, 2008. 2

[61] X. Wang and A. Gupta. Unsupervised Learning of Visual Representations Using Videos. *Proc. ICCV*, 2015. 2

[62] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *Proc. CVPR*, 2000. 2

[63] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman. Single Image 3D Interpreter Network. In *Proc. ECCV*, 2016. 2

[64] 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. 2, 6, 7

[65] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. *Proc. CVPR*, 2011. 2

[66] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned invariant feature transform. *Proc. ECCV*, 2016. 4

[67] J. Ylioinas, J. Kannala, A. Hadid, and M. Pietikainen. Unsupervised learning of overcomplete face descriptors. In *CVPR Workshops*, 2015. 2

[68] A. Yu and K. Grauman. Fine-Grained Visual Comparisons with Local Learning. In *Proc. CVPR*, 2014. 5

[69] X. Yu, F. Zhou, and M. Chandraker. Deep Deformation Network for Object Landmark Localization. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Proc. ECCV*, Cham, 2016. 2, 7

[70] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *Proc. ECCV*, 2014. 2, 6, 7

[71] R. Zhang, P. Isola, and A. A. Efros. Colorful Image Colorization. In *Proc. ECCV*, 2016. 2

[72] W. Zhang, J. Sun, and X. Tang. Cat head detection - How to effectively exploit shape and texture features. In *Proc. ECCV*, 2008. 5

[73] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *Proc. ECCV*, 2014. 2, 7

[74] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning Deep Representation for Face Alignment with Auxiliary Attributes. *PAMI*, 2016. 2, 5, 6, 7, 8

[75] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *ICCV Workshops*, 2013. 7

[76] T. Zhou, P. Krähenbühl, M. Aubry, Q. Huang, and A. A. Efros. Learning Dense Correspondences via 3D-guided Cycle Consistency. In *Proc. CVPR*, 2016. 2

[77] T. Zhou, Y. J. Lee, S. X. Yu, and A. A. Efros. FlowWeb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *Proc. CVPR*, 2015. 2

[78] J. Zhu, L. Van Gool, and S. C. Hoi. Unsupervised face alignment by robust nonrigid mapping. In *Proc. ICCV*, 2009. 2

[79] S. Zhu, C. Li, C. C. Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *Proc. CVPR*, 2015. 7

[80] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proc. CVPR*, 2012. 2, 7

Chapter 5

Unsupervised learning of object frames by dense equivariant image labelling

This work was accepted for Oral Presentation at the Conference on Neural Information Processing Systems (NeurIPS), Long Beach, 2017

This paper builds on the work presented in Chapter 4, eschewing sparse landmarks in favour of a dense output, relating each image pixel to a point in a spherical coordinate space. In this manner, each semantic concept (eyes, nose, etc) "lives" at some point on the sphere. Synthetic warps are again used for training, but additionally we show successful results on toy datasets of an articulated robotic arm and a 3D sphere rotating out of plane. We again show that the unsupervised output can be predictive of manual annotations, and that intra-category invariance emerges automatically. This dense framework forms the basis for the work in Chapter 6, where constraints are imposed on the learned coordinate frame in order to discover symmetries, and Chapter 7, where higher dimensional embeddings are considered.

Unsupervised learning of object frames by dense equivariant image labelling

James Thewlis[1] **Hakan Bilen**[2] **Andrea Vedaldi**[1]

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

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

Abstract

One of the key challenges of visual perception is to extract abstract models of 3D objects and object categories from visual measurements, which are affected by complex nuisance factors such as viewpoint, occlusion, motion, and deformations. Starting from the recent idea of viewpoint factorization, we propose a new approach that, given a large number of images of an object and no other supervision, can extract a dense object-centric coordinate frame. This coordinate frame is invariant to deformations of the images and comes with a dense equivariant labelling neural network that can map image pixels to their corresponding object coordinates. We demonstrate the applicability of this method to simple articulated objects and deformable objects such as human faces, learning embeddings from random synthetic transformations or optical flow correspondences, all without any manual supervision.

1 Introduction

Humans can easily construct mental models of complex 3D objects and object categories from visual observations. This is remarkable because the dependency between an object's appearance and its structure is tangled in a complex manner with extrinsic nuisance factors such as viewpoint, illumination, and articulation. Therefore, learning the intrinsic structure of an object from images requires removing these unwanted factors of variation from the data.

The recent work of [39] has proposed an unsupervised approach to do so, based on on the concept of *viewpoint factorization*. The idea is to learn a deep Convolutional Neural Network (CNN) that can, given an image of the object, detect a discrete set of object landmarks. Differently from traditional approaches to landmark detection, however, landmarks are neither defined nor supervised manually. Instead, the detectors are learned using only the requirement that the detected points must be equivariant (consistent) with deformations of the input images. The authors of [39] show that this constraint is sufficient to learn landmarks that are "intrinsic" to the objects and hence capture their structure; remarkably, due to the generalization ability of CNNs, the landmark points are detected consistently not only across deformations of a given object instance, which are observed during training, but also across different instances. This behaviour emerges automatically from training on thousands of single-instance correspondences.

In this paper, we take this idea further, moving beyond a sparse set of landmarks to a dense model of the object structure (section 3). Our method relates each point on an object to a point in a low dimensional vector space in a way that is consistent across variation in motion and in instance identity. This gives rise to an object-centric coordinate system, which allows points on the surface of an object to be indexed semantically (figure 1). As an illustrative example, take the object category of a face and the vector space $\mathbb{R}^3$. Our goal is to semantically map out the object such that any point on a face, such as the left eye, lives at a canonical position in this "label space". We train a CNN to learn the function that projects any face image into this space, essentially "coloring" each pixel with its

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

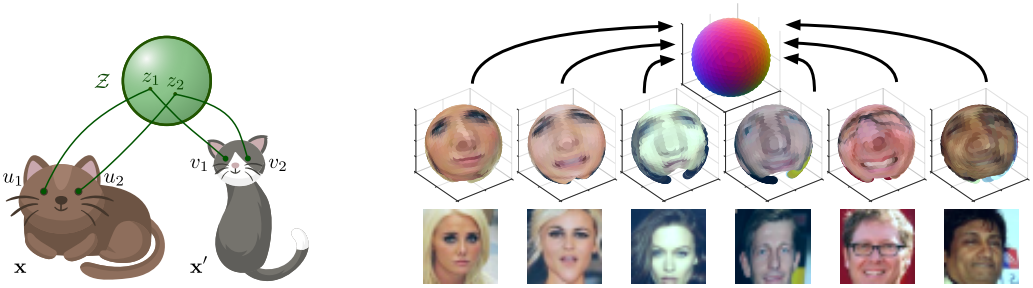


Figure 1: **Dense equivariant image labelling.** *Left:* Given an image $\mathbf{x}$ of an object or object category and no other supervision, our goal is to find a common latent space $\mathcal{Z}$, homeomorphic to a sphere, which attaches a semantically-consistent coordinate frame to the object points. This is done by learning a dense labelling function that maps image pixels to their corresponding coordinate in the $\mathcal{Z}$ space. This mapping function is equivariant (compatible) with image warps or object instance variations. *Right:* An equivariant dense mapping learned in an unsupervised manner from a large dataset of faces. (Results of SIMPLE network, $\mathcal{L}_{dist}, \gamma = 0.5$)

corresponding label. As a result of our learning formulation, the label space has the property of being locally smooth: points nearby in the image are nearby in the label space. In an ideal case, we could imagine the surface of an object to be mapped to a sphere.

In order to achieve these results, we contribute several technical innovations (section 3.2). First, we show that, in order to learn a non-trivial object coordinate frame, the concept of equivariance must be complemented with the one of *distinctiveness* of the embedding. Then, we propose a CNN implementation of this concept that can explicitly express uncertainty in the labelling of the object points. The formulation is used in combination with a probabilistic loss, which is augmented with a robust geometric distance to encourage better alignment of the object features.

We show that this framework can be used to learn meaningful object coordinate frames in a purely unsupervised manner, by analyzing thousands of deformations of visual objects. While [39] proposed to use Thin Plate Spline image warps for training, here we also consider simple synthetic articulated objects having frames related by known optical flow (section 4).

We conclude the paper with a summary of our finding (section 5).

2 Related Work

Learning the structure of visual objects. Modeling the structure of visual objects is a widely-studied (*e.g.* [6, 7, 11, 41, 12]) computer vision problem with important applications such as facial landmark detection and human body pose estimation. Much of this work is supervised and aimed at learning detectors of objects or their parts, often using deep learning. A few approaches such as spatial transformer networks [20] can learn geometric transformations without explicit geometric supervision, but do not build explicit geometric models of visual objects.

More related to our work, WarpNet [21] and geometric matching networks [35] learn a neural network that predicts Thin Plate Spline [3] transformations between pairs of images of an object, including synthetic warps. Deep Deformation Network [44] improves WarpNet by using a Point Transformer Network to refine the computed landmarks, but it requires manual supervision. None of these works look at the problem of learning an invariant geometric embedding for the object.

Our work builds on the idea of *viewpoint factorization* (section 3.1), recently introduced in [39, 32]. However, we extend [39] in several significant ways. First, we construct a *dense* rather than discrete embedding, where all pixels of an object are mapped to an invariant object-centric coordinate instead of just a small set of selected landmarks. Second, we show that the equivariance constraint proposed in [39] is not quite enough to learn such an embedding; it must be complemented with the concept of a *distinctive* embedding (section 3.1). Third, we introduce a new neural network architecture and corresponding training objective that allow such an embedding to be learned in practice (section 3.2).

Optical/semantic flow. A common technique to find correspondences between temporally related video frames is optical flow [18]. The state-of-the-art methods [14, 40, 19] typically employ convolu-

tional neural networks to learn pairwise dense correspondences between the same object instances at subsequent frames. The SIFT Flow method [25] extends the between-instance correspondences to cross-instance mappings by matching SIFT features [27] between semantically similar object instances. Learned-Miller [24] extends the pairwise correspondences to multiple images by posing a problem of alignment among the images of a set. Collection Flow [22] and Mobahi *et al*. [29] project objects onto a low-rank space that allow for joint alignment. FlowWeb [52], and Zhou *et al*. [51] construct fully connected graphs to maximise cycle consistency between each image pair and synthethic data as an intermediary by training a CNN. In our experiments (section 4) flow is known from synthetic warps or motion, but our work could build on any unsupervised optical flow method.

Unsupervised learning. Classical unsupervised learning methods such as autoencoders [4, 2, 17] and denoising autoencoders aim to learn useful feature representations from an input by simply reconstructing it after a bottleneck. Generative adversarial networks [16] target producing samples of realistic images by training generative models. These models when trained joint with image encoders are also shown to learn good feature representations [9, 10]. More recently several studies have emerged that train neural networks by learning auxiliary or pseudo tasks. These methods exploit typically some existing information in input as "self-supervision" without any manual labeling by removing or perturbing some information from an input and requiring a network to reconstruct it. For instance, Doersch *et al*. [8], and Noroozi and Favaro [31] train a network to predict the relative locations of shuffled image patches. Other self-supervised tasks include colorizing images [46], inpainting [34], ranking frames of a video in temporally correct order [28, 13]. More related to our approach, Agrawal *et al*. [1] use egomotion as supervisory signal to learn feature representations in a Siamese network by predicting camera transformations from image pairs, [33] learn to group pixels that move together in a video. [50, 15] use a warping-based loss to learn depth from video. Recent work [36] leverages RGB-D based reconstruction [30] and is similar to this work, showing qualitatively impressive results learning a consistent low-dimensional labelling on a human dataset.

3 Method

This section discusses our method in detail, first introducing the general idea of dense equivariant labelling (section 3.1), and then presenting a concrete implementation of the latter using a novel deep CNN architecture (section 3.2).

3.1 Dense equivariant labelling

Consider a 3D object $S \subset \mathbb{R}^3$ or a class of such objects S that are topologically isomorphic to a sphere $\mathcal{Z} \subset \mathbb{R}^3$ (i.e. the objects are simple closed surfaces without holes). We can construct a homeomorphism $p = \pi_S(q)$ mapping points of the sphere $q \in \mathcal{Z}$ to points $p \in S$ of the objects. Furthermore, if the objects belong to the same semantic category (*e.g*. faces), we can assume that these isomorphisms are *semantically consistent*, in the sense that $\pi_{S'} \circ \pi_S^{-1} : S \to S'$ maps points of object S to semantically-analogous points in object S' (*e.g*. for human faces the right eye in one face should be mapped to the right eye in another [39]).

While this construction is abstract, it shows that we can endow the object (or object category) with a spherical reference system $\mathcal{Z}$. The authors of [39] build on this construction to define a discrete system of object landmarks by considering a finite number of points $z_k \in \mathcal{Z}$. Here, we take the geometric embedding idea more literally and propose to explicitly learn a dense mapping from images of the object to the object-centric coordinate space $\mathcal{Z}$. Formally, we wish to learn a *labelling function* $\Phi : (\mathbf{x}, u) \mapsto z$ that takes a RGB image $\mathbf{x} : \Lambda \to \mathbb{R}^3$, $\Lambda \subset \mathbb{R}^3$ and a pixel $u \in \Lambda$ to the object point $z \in \mathcal{Z}$ which is imaged at u (figure 1).

Similarly to [39], this mapping must be compatible or equivariant with image deformations. Namely, let $g : \Lambda \to \Lambda$ be a deformation of the image domain, either synthetic or due to a viewpoint change or other motion. Furthermore, let $g\mathbf{x} = \mathbf{x} \circ g^{-1}$ be the action of g on the image (obtained by inverse warp). Barring occlusions and boundary conditions, pixel u in image $\mathbf{x}$ must receive the same label as pixel gu in image $g\mathbf{x}$, which results in the *invariance constraint*:

$$\forall \mathbf{x}, u : \quad \Phi(\mathbf{x}, u) = \Phi(g\mathbf{x}, gu). \tag{1}$$

Equivalently, we can view the network as a functional $\mathbf{x} \mapsto \Phi(\mathbf{x}, \cdot)$ that maps the image to a corresponding label map. Since the label map is an image too, g acts on it by inverse warp.[1] Using this, the constraint (1) can be rewritten as the *equivariance relation* $g\Phi(\mathbf{x}, \cdot) = \Phi(g\mathbf{x}, \cdot)$. This can be visualized by noting that the label image deforms in the same way as the input image, as show for example in figure 3.

For learning, constraint (1) can be incorporated in a loss function as follows:

$$\mathcal{L}(\Phi|\alpha) = \frac{1}{|\Lambda|} \int_{\Lambda} \|\Phi(\mathbf{x}, u) - \Phi(g\mathbf{x}, gu)\|^2 \, du.$$

However, minimizing this loss has the significant drawback that a global optimum is obtained by simply setting $\Phi(\mathbf{x}, u) = \text{const}$. The reason for this issue is that (1) is not quite enough to learn a useful object representation. In order to do so, we must require the labels not only to be equivariant, but also *distinctive*, in the sense that

$$\Phi(\mathbf{x}, u) = \Phi(g\mathbf{x}, v) \quad \Leftrightarrow \quad v = gu.$$

We can encode this requirement as a loss in different ways. For example, by using the fact that points $\Phi(\mathbf{x}, u)$ are on the unit sphere, we can use the loss:

$$\mathcal{L}'(\Phi|\mathbf{x}, g) = \frac{1}{|\Lambda|} \int_{\Lambda} \|gu - \text{argmax}_v \langle \Phi(\mathbf{x}, u), \Phi(g\mathbf{x}, v) \rangle\|^2 \, du. \tag{2}$$

By doing so, the labels $\Phi(\mathbf{x}, u)$ must be able to discriminate between different object points, so that a constant labelling would receive a high penalty.

Relationship with learning invariant visual descriptors. As an alternative to loss (2), we could have used a pairwise loss[2] to encourage the similarity $\langle \Phi(\mathbf{x}, u), \Phi(\mathbf{x}', gu) \rangle$ of the labels assigned to corresponding pixels u and gu to be larger than the similarity $\langle \Phi(\mathbf{x}, u), \Phi(\mathbf{x}', v) \rangle$ of the labels assigned to pixels u and v that do *not* correspond. Formally, this would result in a pairwise loss similar to the ones often used to learn invariant visual descriptors for image matching. The reason why our method learns an object representation instead of a generic visual descriptor is that the *dimensionality* of the label space $\mathcal{Z}$ is just enough to represent a point on a surface. If we replace $\mathcal{Z}$ with a larger space such as $\mathbb{R}^d$, $d \gg 2$, we can expect $\Phi(\mathbf{x}, u)$ to learn to extract generic visual descriptors like SIFT instead. This establishes an interesting relationship between visual descriptors and object-specific coordinate vectors and suggests that it is possible to transition between the two by controlling their dimensionality.

3.2 Concrete learning formulation

In this section we introduce a concrete implementation of our method (figure 2). For the mapping Φ, we use a CNN that receives as input an image tensor $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ and produces as output a label tensor $\mathbf{z} \in \mathbb{R}^{H \times W \times L}$. We use the notation $\Phi_u(\mathbf{x})$ to indicate the L-dimensional label vector extracted at pixel u from the label image computed by the network.

The dimension of the label vectors is set to $L = 3$ (instead of $L = 2$) in order to allow the network to express uncertainty about the label assigned to a pixel. The network can do so by modulating the norm of $\Phi_u(\mathbf{x})$. In fact, correspondences are expressed probabilistically by computing the inner product of label vectors followed by the softmax operator. Formally, the probability that pixel v in image $\mathbf{x}'$ corresponds to pixel u in image $\mathbf{x}$ is expressed as:

$$p(v|u; \mathbf{x}, \mathbf{x}', \Phi) = \frac{e^{\langle \Phi_u(\mathbf{x}), \Phi_v(\mathbf{x}') \rangle}}{\sum_z e^{\langle \Phi_u(\mathbf{x}), \Phi_z(\mathbf{x}') \rangle}}. \tag{3}$$

In this manner, a shorter vector Φ_u results in a more diffuse probability distribution.

[1] In the sense that $g\Phi(\mathbf{x}, \cdot) = \Phi(\mathbf{x}, \cdot) \circ g^{-1}$.

[2] Formally, this is achieved by the loss

$$\mathcal{L}''(\Phi|\mathbf{x}, g) = \frac{1}{|\Lambda|} \int_{\Lambda} \max \left\{ 0, \max_v \Delta(u, v) + \langle \Phi(\mathbf{x}, u), \Phi(g\mathbf{x}, v) \rangle - \langle \Phi(\mathbf{x}, u), \Phi(g\mathbf{x}, gu) \rangle \right\} du,$$

where $\Delta(u, v) \geq 0$ is an error-dependent margin.

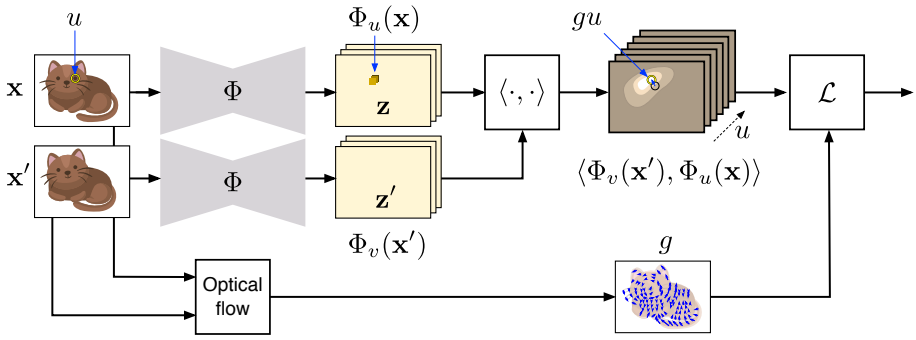


Figure 2: **Unsupervised dense correspondence network.** From left to right: The network Φ extracts label maps $\Phi_u(\mathbf{x})$ and $\Phi_v(\mathbf{x}')$ from the image pair $\mathbf{x}$ and $\mathbf{x}'$. An optical flow module (or ground truth for synthetic transformation) computes the warp (correspondence field) g such that $\mathbf{x}' = g\mathbf{x}$. Then the label of each point u in the first image is correlated to each point v in the second, obtaining a number of score maps. The loss evaluates how well the score maps predict the warp g.

Next, we wish to define a loss function for learning Φ from data. To this end, we consider a triplet $\alpha = (\mathbf{x}, \mathbf{x}', g)$, where $\mathbf{x}' = g\mathbf{x}$ is an image that corresponds to $\mathbf{x}$ up to transformation g (the nature of the data is discussed below). We then assess the performance of the network Φ on the triplet α using two losses. The first loss is the *negative log-likelihood* of the ground-truth correspondences:

$$\mathcal{L}_{\text{log}}(\Phi|\mathbf{x}, \mathbf{x}', g) = -\frac{1}{HW}\sum_{u} \log p(gu|u; \mathbf{x}, \mathbf{x}', \Phi). \tag{4}$$

This loss has the advantage that it explicitly learns (3) as the probability of a match. However, it is not sensitive to the *size* of a correspondence error $v - gu$. In order to address this issue, we also consider the loss

$$\mathcal{L}_{\text{dist}}(\Phi|\mathbf{x}, \mathbf{x}', g) = \frac{1}{HW}\sum_{u}\sum_{v} \|v - gu\|_2^\gamma \, p(v|u; \mathbf{x}, \mathbf{x}', \Phi). \tag{5}$$

Here $\gamma > 0$ is an exponent used to control the robustness of the distance measure, which we set to $\gamma = 0.5, 1$.

Nework details. We test two architecture. The first one, denoted SIMPLE, is the same as [49, 39] and is a chain $(5, 20)_+, (2, \text{mp}), \downarrow_2, (5, 48)_+, (3, 64)_+, (3, 80)_+, (3, 256)_+, (1, 3)$ where (h, c) is a bank of c filters of size $h \times h$, $+$ denotes ReLU, (h, mp) is $h \times h$ max-pooling, $\downarrow_s$ is $s\times$ downsampling. Better performance can be obtained by increasing the support of the filters in the network; for this, we consider a second network DILATIONS $(5, 20)_+, (2, \text{mp}), \downarrow_2$, $(5, 48)_+, (5, 64, 2)_+, (3, 80, 4)_+, (3, 256, 2)_+, (1, 3)$ where (h, c, d) is a filter with $\times d$ dilation [43].

3.3 Learning from synthetic and true deformations

Losses (4) and (5) learn from triplets $\alpha = (\mathbf{x}, \mathbf{x}', g)$. Here $\mathbf{x}'$ can be either generated synthetically by applying a random transformation g to a natural image $\mathbf{x}$ [39, 21], or it can be obtained by observing image pairs $(\mathbf{x}, \mathbf{x}')$ containing true object deformations arising from a viewpoint change or an object motion or deformation.

The use of synthetic transformations enables training even on static images and was considered in [39], who showed it to be sufficient to learn meaningful landmarks for a number of real-world object such as human and cat faces. Here, in addition to using synthetic deformations, we also consider using animated image pairs $\mathbf{x}$ and $\mathbf{x}'$. In principle, the learning formulation can be modified so that knowledge of g is not required; instead, images and their warps can be compared and aligned directly based on the brightness constancy principle. In our toy video examples we obtain g from the rendering engine, but it can in theory be obtained using an off-the-shelf optical flow algorithm which would produce a noisy version of g.

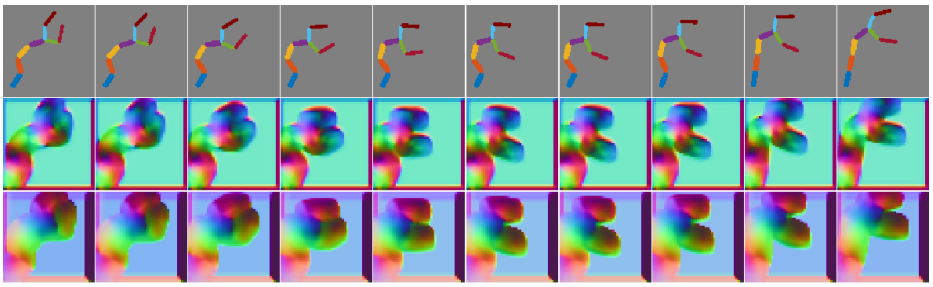


Figure 3: **Roboarm equivariant labelling.** Top: Original video frames of a simple articulated object. Middle and bottom: learned labels, which change equivariantly *with* the arm, learned using $\mathcal{L}_{log}$ and $\mathcal{L}_{dist}$, respectively. Different colors denote different points of the spherical object frame.

4 Experiments

This section assesses our unsupervised method for dense object labelling on two representative tasks: two toy problems (sections 4.1 and 4.2) and human and cat faces (section 4.3).

4.1 Roboarm example

In order to illustrate our method we consider a toy problem consisting of a simple articulated object, namely an animated robotic arm (figure 3) created using a 2D physics engine [38]. We do so for two reasons: to show that the approach is capable of labelling correctly deformable/articulated objects and to show that the spherical model $\mathcal{Z}$ is applicable also to thin objects, that have mainly a 1D structure.

Dataset details. The arm is anchored to the bottom left corner and is made up of colored capsules connected with joints having reasonable angle limits to prevent unrealistic contortion and self-occlusion. Motion is achieved by varying the gravity vector, sampling each element from a Gaussian with standard deviation $15\,\mathrm{m\,s^{-2}}$ every 100 iterations. Frames $\mathbf{x}$ of size 90×90 pixels and the corresponding flow fields $g : \mathbf{x} \mapsto \mathbf{x}'$ are saved every 20 iterations. We also save the positions of the capsule centers. The final dataset has 23999 frames.

Learning. Using the correspondences $\alpha = (\mathbf{x}, \mathbf{x}', g)$ provided by the flow fields, we use our method to learn an object centric coordinate frame $\mathcal{Z}$ and its corresponding labelling function $\Phi_u(\mathbf{x})$. We test learning Φ using the probabilistic loss (4) and distance-based loss (5). In the loss we ignore areas with zero flow, which automatically removes the background. We use the SIMPLE network architecture (section 3.2).

Results. Figure 3 provides some qualitative results, showing by means of colormaps the labels $\Phi_u(\mathbf{x})$ associated to different pixels of each input image. It is easy to see that the method attaches consistent labels to the different arm elements. The distance-based loss produces a more uniform embedding, as may be expected. The embeddings are further visualized in Figure 4 by projecting a number of video frames back to the learned coordinate spaces $\mathcal{Z}$. It can be noted that the space is invariant, in the sense that the resulting figure is approximately the same despite the fact that the object deforms significantly in image space. This is true for both embeddings, but the distance-based ones are geometrically more consistent.

Predicting capsule centers. We evaluate quantitatively the ability of our *object frames* to localise the capsule centers. If our assumption is correct and a coordinate system intrinsic to the object has been learned, then we should expect there to be a specific 3-vector in $\mathcal{Z}$ corresponding to each center, and our job is to find these vectors. Various strategies could be used, such as averaging the object-centric coordinates given to the centers over the training set, but we choose to incorporate the problem into the learning framework. This is done using the negative log-likelihood in much the same way as (4), limiting our vectors u to the centers. This is done as an auxiliary layer with no backpropagation to the rest of the network, so that the embedding remains unsupervised. The error reported is the Euclidean distance as a percentage of the image width.

Results are given for the different loss functions used for unsupervised training in Table 1 and visualized in Figure 5 right, showing that the object centers can be located to a high degree of

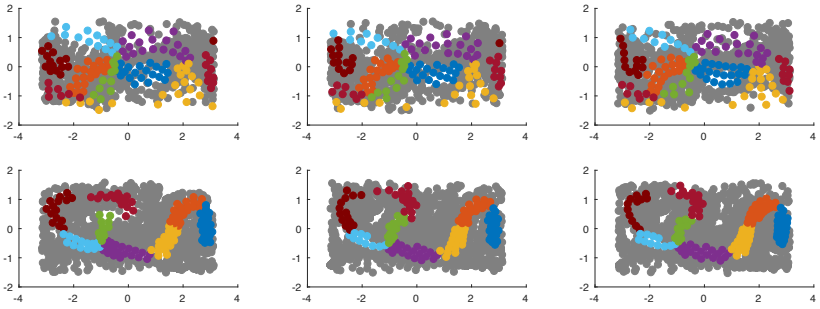


Figure 4: **Invariance of the object-centric coordinate space for Roboarm.** The plot projects frames 3,6,9 of figure 3 on the object-centric coordinate space $\mathcal{Z}$, using the embedding functions learned by means of the probabilistic (top) and distance (bottom) based losses. The sphere is then unfolded, plotting latitude and longitude (in radians) along the vertical and horizontal axes.

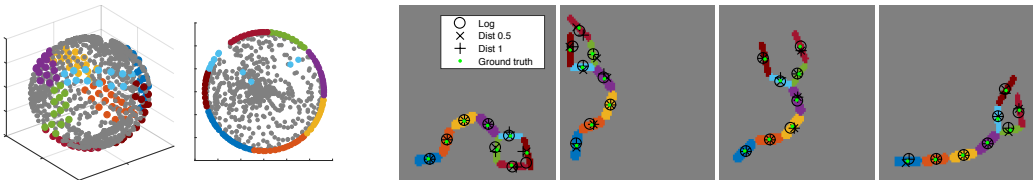


Figure 5: **Left: Embedding spaces of different dimension.** Spherical embedding (from the 3D embedding function $\Phi_u(\mathbf{x}) \in \mathbb{R}^3$) learned using the distance loss compared to a circular embedding with one dimension less. **Right: Capsule center prediction for different losses**.

accuracy. The negative log likelihood performs best while the two losses incorporating distance perform similarly.

We also perform experiments varying the dimensionality L of the label space $\mathcal{Z}$ (Table 2). Perhaps most interestingly, given the almost one-dimensional nature of the arm, is the case of $L = 2$, which would correspond to an approximately circular space (since the length of vectors is used to code for uncertainty). As seen in the right of Figure 5 left, the segments are represented almost perfectly on the boundary of a circle, with the exception of the bifurcation which it is unable to accurately represent. This is manifested by the light blue segment trying, and failing, to be in two places at once.

Unsupervised Loss	Error
$\mathcal{L}_{\text{log}}$	0.97 %
$\mathcal{L}_{\text{dist}}, \gamma = 1$	1.13 %
$\mathcal{L}_{\text{dist}}, \gamma = 0.5$	1.14 %

Table 1: Predicting capsule centers. Error as percent of image width.

Descriptor Dimension	Error
2	1.29 %
3	1.14 %
5	1.16 %
20	1.28 %

Table 2: Descriptor dimension ($\mathcal{L}_{\text{dist}}, \gamma = 0.5$). L>3 shows no improvement, suggesting L=3 is the natural manifold of the arm.

4.2 Textured sphere example

The experiment of Figure 6 tests the ability of the method to understand a complete rotation of a 3D object, a simple textured sphere. Despite the fact that the method is trained on pairs of adjacent video frames (and corresponding optical flow), it still learns a globally-consistent embedding. However, this required switching from from the SIMPLE to the DILATIONS architecture (section 3.2).

4.3 Faces

After testing our method on a toy problem, we move to a much harder task and apply our method to generate an object-centric reference frame $\mathcal{Z}$ for the *category* of human faces. In order to generate an image pair and corresponding flow field for training we warp each face synthetically using Thin Plate Spline warps in a manner similar to [39]. We train our models on the extensive CelebA [26] dataset of over 200k faces as in [39], excluding MAFL [49] test overlap from the given training split. It has annotations of the eyes, nose and mouth corners. Note that we do not use these to train our model. We also use AFLW [23], testing on 2995 faces [49, 42, 48] with 5 landmarks. Like [39] we

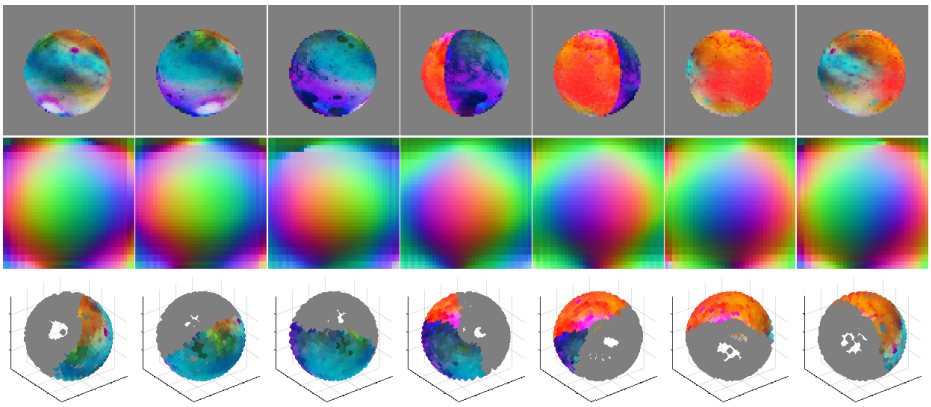


Figure 6: **Sphere equivariant labelling.** Top: video frames of a rotating textured sphere. Middle: learned dense labels, which change equivariantly *with* the sphere. Bottom: re-projection of the video frames on the object frame (also spherical). Except for occlusions, the reprojections are approximately invariant, correctly mapping the blue and orange sides to different regions of the label space

use 10,122 faces for training. We additionally evaluate qualitatively on a dataset of cat faces [47], using 8609 images for training.

Qualitative assessment. We find that for network SIMPLE the negative log-likelihood loss, while performing best for the simple example of the arm, performs poorly on faces. Specifically, this model fails to disambiguate the left and right eye, as shown in Figure 9 (right). The distance-based loss (5) produces a more coherent embedding, as seen in Figure 9 (left). Using DILATIONS this problem disappears, giving qualitatively smooth and unambiguous labels for both the distance loss (Figure 7) and the log-likelihood loss (Figure 8). For cats our method is able to learn a consistent object frame despite large variations in appearance (Figure 8).

Figure 7: **Faces.** DILATIONS network with $\mathcal{L}_{\text{dist}}, \gamma = 0.5$. Top: Input images, Middle: Predicted dense labels mapped to colours, Bottom: Image pixels mapped to label sphere and flattened.

Regressing semantic landmarks. We would like to quantify the accuracy of our model in terms of ability to consistently locate manually annotated points, specifically the eyes, nose, and mouth corners given in the CelebA dataset. We use the standard test split for evaluation of the MAFL dataset [49], containing 1000 images. We also use the MAFL training subset of 19k images for learning to predict the ground truth landmarks, which gives a quantitative measure of the consistency of our *object frame* for detecting facial features. These are reported as Euclidean error normalized as a percentage of inter-ocular distance.

In order to map the object frame to the semantic landmarks, as in the case of the robot arm centers, we learn the vectors $z_k \in \mathcal{Z}$ corresponding to the position of each point in our canonical reference space and then, for any given image, find the nearest z and its corresponding pixel location u. We report the localization performance of this model in Table 3 ("Error Nearest"). We empirically validate that with the SIMPLE network the negative log-likelihood is not ideal for this task (Figure 9) and

Figure 8: **Cats.** DILATIONS network with $\mathcal{L}_{\text{log}}$. Top: Input images, Middle: Labels mapped to colours, Bottom: Images mapped to the spherical object frames.

Figure 9: **Annotated landmark prediction** from the shown unsupervised label maps (SIMPLE network). **Left:** Trained with $\mathcal{L}_{dist}, \gamma = 0.5$, **Right:** Failure to disambiguate eyes with $\mathcal{L}_{log}$. (Prediction: green, Ground truth: Blue)

we obtain higher performance for the robust distance with power 0.5. However, after switching to DILATIONS to increase the receptive field both methods perform comparably.

The method of [39] learns to regress P ground truth coordinates based on $M > P$ unsupervised landmarks. By regressing from multiple points it is not limited to integer pixel coordinates. While we are not predicting landmarks as network output, we can emulate this method by allowing multiple points in our object coordinate space to be predictive for a single ground truth landmark. We learn one regressor per ground truth point, each formulated as a linear regressor $\mathbb{R}^{2M} \rightarrow \mathbb{R}^2$ on top of coordinates from $M = 50$ learned intermediate points. This allows the regression to say which points in $\mathcal{Z}$ are most useful for predicting each ground truth point.

We also report results after unsupervised finetuning of a CelebA network to the more challenging AFLW followed by regressor training on AFLW. As shown in Tables 3 and 4, we outperform other unsupervised methods on both datasets, and are comparable to fully supervised methods.

Network	Unsup. Loss	Error Nearest	Error Regress
SIMPLE	$\mathcal{L}_{\text{log}}$	75.02 %	—
SIMPLE	$\mathcal{L}_{\text{dist}}, \gamma = 1$	14.57 %	7.94 %
SIMPLE	$\mathcal{L}_{\text{dist}}, \gamma = 0.5$	13.29 %	7.18 %
DILATIONS	$\mathcal{L}_{\text{log}}$	11.05 %	5.83 %
DILATIONS	$\mathcal{L}_{\text{dist}}, \gamma = 0.5$	10.53 %	5.87 %
[39]			6.67 %

Table 3: Nearest neighbour and regression landmark prediction on MAFL

Method	Error
RCPR [5]	11.6 %
Cascaded CNN [37]	8.97 %
CFAN [45]	10.94 %
TCDCN [49]	7.65 %
RAR [42]	7.23 %
Unsup. Landmarks [39]	10.53 %
DILATIONS $\mathcal{L}_{\text{dist}}, \gamma = 0.5$	8.80 %

Table 4: Comparison with supervised and unsupervised methods on AFLW

5 Conclusions

Building on the idea of viewpoint factorization, we have introduce a new method that can endow an object or object category with an invariant dense geometric embedding automatically, by simply observing a large dataset of unlabelled images. Our learning framework combines in a novel way the concept of equivariance with the one of distinctiveness. We have also proposed a concrete implementation using novel losses to learn a deep dense image labeller. We have shown empirically that the method can learn a consistent geometric embedding for a simple articulated synthetic robotic arm as well as for a 3D sphere model and real faces. The resulting embeddings are invariant to deformations and, importantly, to *intra-category* variations.

Acknowledgments: This work acknowledges the support of the AIMS CDT (EPSRC EP/L015897/1) and ERC 677195-IDIU. Clipart: FreePik.

References

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proc. ICCV*, 2015.

[2] Yoshua Bengio. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2009.

[3] Fred L. Bookstein. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *PAMI*, 1989.

[4] H Bourlard and Y Kamp. Auto-Association by Multilayer Perceptrons and Singular Value Decomposition. *Biological Cybernetics*, 1988.

[5] Xavier P. Burgos-Artizzu, Pietro Perona, and Piotr Dollár. Robust face landmark estimation under occlusion. In *Proc. ICCV*, 2013.

[6] T F Cootes, C J Taylor, D H Cooper, and J Graham. Active shape models: their training and application. *CVIU*, 1995.

[7] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *Proc. CVPR*, 2005.

[8] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised Visual Representation Learning by Context Prediction. In *Proc. ICCV*, 2015.

[9] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *Proc. ICLR*, 2017.

[10] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. *Proc. ICLR*, 2017.

[11] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part Based Models. *PAMI*, 2010.

[12] Rob Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, 2003.

[13] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proc. CVPR*, 2017.

[14] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *Proc. ICCV*, 2015.

[15] Ravi Garg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Proc. ECCV*, pages 740–756, 2016.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[17] G E Hinton and R R Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 2006.

[18] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 1981.

[19] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. *arXiv preprint arXiv:1612.01925*, 2016.

[20] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial Transformer Networks. In *Proc. NIPS*, 2015.

[21] A. Kanazawa, D. W. Jacobs, and M. Chandraker. WarpNet: Weakly supervised matching for single-view reconstruction. In *Proc. CVPR*, 2016.

[22] Ira Kemelmacher-Shlizerman and Steven M. Seitz. Collection flow. In *Proc. CVPR*, 2012.

[23] Martin Koestinger, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[24] Erik G Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[25] Ce Liu, Jenny Yuen, and Antonio Torralba. SIFT Flow: Dense correspondence across scenes and its applications. *PAMI*, 2011.

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proc. ICCV*, 2015.

[27] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[28] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *Proc. ECCV*, 2016.

[29] Hossein Mobahi, Ce Liu, and William T. Freeman. A Compositional Model for Low-Dimensional Image Set Representation. *Proc. CVPR*, 2014.

[30] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proc. CVPR*, 2015.

[31] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. ECCV*, 2016.

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

[33] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *Proc. CVPR*, 2017.

[34] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context Encoders: Feature Learning by Inpainting. In *Proc. CVPR*, 2016.

[35] I. Rocco, R. Arandjelović, and J. Sivic. Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, 2017.

[36] Tanner Schmidt, Richard Newcombe, and Dieter Fox. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427, 2017.

[37] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *Proc. CVPR*, 2013.

[38] Yuval Tassa. CapSim - the MATLAB physics engine. `https://mathworks.com/matlabcentral/fileexchange/29249-capsim-the-matlab-physics-engine`.

[39] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proc. ICCV*, 2017.

[40] James Thewlis, Shuai Zheng, Philip H. S. Torr, and Andrea Vedaldi. Fully-Trainable Deep Matching. In *Proc. BMVC*, 2016.

[41] Markus Weber, Max Welling, and Pietro Perona. Towards automatic discovery of object categories. In *Proc. CVPR*, 2000.

[42] Shengtao Xiao, Jiashi Feng, Junliang Xing, Hanjiang Lai, Shuicheng Yan, and Ashraf Kassim. Robust Facial Landmark Detection via Recurrent Attentive-Refinement Networks. In *Proc. ECCV*, 2016.

[43] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Proc. ICLR*, 2016.

[44] Xiang Yu, Feng Zhou, and Manmohan Chandraker. Deep Deformation Network for Object Landmark Localization. In *Proc. ECCV*, Cham, 2016.

[45] Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen. Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment. In *Proc. ECCV*, 2014.

[46] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful Image Colorization. In *Proc. ECCV*, 2016.

[47] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection - How to effectively exploit shape and texture features. In *Proc. ECCV*, 2008.

[48] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *Proc. ECCV*, 2014.

[49] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Learning Deep Representation for Face Alignment with Auxiliary Attributes. *PAMI*, 2016.

[50] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. CVPR*, 2017.

[51] Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qixing Huang, and Alexei A. Efros. Learning Dense Correspondences via 3D-guided Cycle Consistency. In *Proc. CVPR*, 2016.

[52] Tinghui Zhou, Yong Jae Lee, Stella X. Yu, and Alexei A. Efros. FlowWeb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *Proc. CVPR*, 2015.

Chapter 6

Modelling and unsupervised learning of symmetric deformable object categories

This work was accepted for Oral Presentation at the Conference on Neural Information Processing Systems (NeurIPS), Montreal, 2018

This paper employs the framework introduced in Chapter 5 in order to tackle the problem of discovering symmetries from a collection of images. Objects that are intuitively symmetric rarely appear so in the image space, due to effects such as perspective and deformations. Instead of attempting to obtain the true underlying 3D shape, we show that we can instead use the *Object Frame* idea in order to find symmetries in the space of object deformations. This allows us to learn a canonical space where symmetries can be trivially represented. As before, learning is unsupervised based on synthetically warped pairs. We show qualitatively the ability to recover bilateral symmetry on human and animal faces, and quantitative results predicting left-right correspondences. We also show examples of radial symmetry on a molecule and flowers, which requires accounting for ambiguities. We introduce a theory of symmetric deformations employing the formalism of group theory.

Modelling and unsupervised learning of symmetric deformable object categories

James Thewlis[1] **Hakan Bilen**[2] **Andrea Vedaldi**[1]

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

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

Abstract

We propose a new approach to model and learn, without manual supervision, the symmetries of natural objects, such as faces or flowers, given only images as input. It is well known that objects that have a symmetric structure do not usually result in symmetric images due to articulation and perspective effects. This is often tackled by seeking the intrinsic symmetries of the underlying 3D shape, which is very difficult to do when the latter cannot be recovered reliably from data. We show that, if only raw images are given, it is possible to look instead for symmetries in the *space of object deformations*. We can then learn symmetries from an unstructured collection of images of the object as an extension of the recently-introduced *object frame* representation, modified so that object symmetries reduce to the obvious symmetry groups in the normalized space. We also show that our formulation provides an explanation of the ambiguities that arise in recovering the pose of symmetric objects from their shape or images and we provide a way of discounting such ambiguities in learning.

1 Introduction

Most natural objects are symmetric: mammals have a bilateral symmetry, a glass is rotationally symmetric, many flowers have a radial symmetry, etc. While such symmetries are easy to understand for a human, it remains surprisingly challenging to develop algorithms that can reliably detect the symmetries of visual object in images. The key difficulty is that objects that are structurally symmetric do not generally result in symmetric images; in fact, the latter occurs only when the object is imaged under special viewpoints and, for deformable objects, with a special poses (Leonardo's Vitruvian Man illustrates this point).

The standard approach to characterizing symmetries in objects is to look not at their images, but at their 3D shape; if the latter is available, then symmetries can be recovered by analysing the *intrinsic geometry* of the shape. However, often only images of the objects are available, and reconstructing an accurate 3D shape from them can be very challenging, especially if the object is deformable.

In this paper, we thus seek a new approach to learn *without supervision and from raw images alone* the symmetries of deformable object categories. This may sound difficult since even characterising the basic geometry of natural objects without external supervision remains largely an open problem. Nevertheless, we show that it is possible to extend the method of [38], which was recently introduced to learn the "topology" of object categories, to do exactly this.

There are three key enabling factors in our approach. First, we do not consider symmetries of a single object or 3D shape in isolation; instead, we seek symmetries shared by all the instances of the objects in a given category, imaged under different viewing conditions and deformations. Second, rather than considering the common concept of intrinsic symmetries, we propose to look at symmetries not of 3D shapes, but of the *space of their deformations* (section 4). Third, we show that the *normalized object frame* of [38] can be learned in such a way that the deformation symmetries are represented by

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

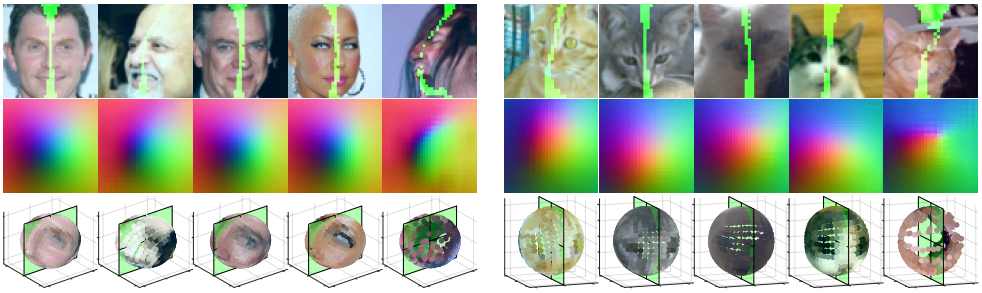


Figure 1: **Symmetric object frame for human (left) and cat (right) faces** (test set). Our method learns a viewpoint and identity invariant geometric embedding which captures the symmetry of natural objects (in this case bilateral) *without manual supervision*. Top: input images with the axis of symmetry superimposed (shown in green). Middle: dense embedding mapped to colours. Bottom: image pixels mapped to 3D representation space with the reflection plane (green).

the obvious symmetry groups in the object frame. The latter also result in a constraint that can be easily added to the self-supervised formulation of [38] to learn symmetries in practice (section 3).

We start by deriving our formulation for the special case of bilateral symmetries (section 3). Then, we propose a theory of symmetric deformation spaces (section 4) that generalises the method to other symmetry groups. An important step in this generalization is to characterise the ambiguities that symmetries induce in recovering the pose of an object from an image of it, or from its 3D shape, which may not occur with bilateral symmetries.

The resulting approach is the first that, to our knowledge, can learn the symmetries of object categories given only raw images as input, without manual annotations. For demonstration, we show that this approach can learn the bilateral symmetry in human and pet faces (fig. 1) as well as in synthetic 3D objects (section 6). To assess the method, we look at how well the resulting representation can detect pairs of symmetric object landmarks (e.g. left and right eyes) even when the object does not appear symmetric.

We also investigate the problem of symmetry-induced ambiguities in learning the geometry of natural objects. For objects such as animals that have a bilateral symmetry, it is generally possible to uniquely identify their left and right sides and thus recover their pose uniquely. On the other hand, for objects such as flowers that may have a radial symmetry, it is generally impossible to say which way is "up", creating an ambiguity in pose recovery. Our framework clarifies why and when this occurs and suggests how to modify the learning formulation to mitigate the effect of such ambiguities (sections 4 and 6.2).

2 Related work

Cross-instance object matching. Our method is also related to the techniques that find dense correspondences between different object instances by matching their SIFT features [25], establishing region correspondences [14, 15] and matching the internal representations of neural networks [24]. In addition, dense correspondences have been generalized between image pairs to arbitrary number of multiple images by Learned-Miller [20]. More recently, RSA [32], Collection Flow [18] and Mobahi *et al*. [28] show that a collection of images can be projected into a lower dimensional subspace before performing a joint alignment among the projected images. Novotny *et al*. [30] train a neural network with image labels that learns to automatically discover semantically meaningful parts across animals.

Unsupervised learning of object structure. Supervised visual object characterization [6, 11, 21, 8, 10] is a well established problem in computer vision and successfully applied to facial landmark detection and human body pose estimation. Unsupervised methods include Spatial Transformer Networks [16] that learn to transform images to improve image classification, WarpNet [17] and geometric matching networks [34] that learn to match object pairs by estimating relative transformations between them. In contrast to ours, these methods do not learn a canonical object geometry and only provide relative mapping from one object to another. More related to ours, Thewlis *et al*. [39, 38] propose to characterize object structure via detecting landmarks [39] or dense labels [38] that are

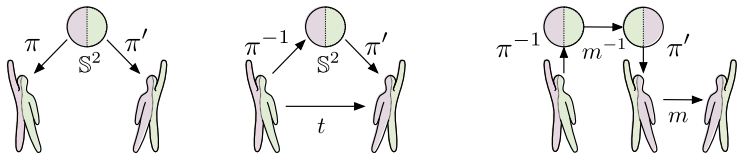


Figure 2: Left: an object category consisting of two poses π, π' with bilateral symmetry. Middle: the non-rigid deformation $t = \pi' \circ \pi^{-1}$ transporting one pose into the other. Right: construction of $t = m\pi m^{-1}\pi^{-1}$ by applying the reflection operator m both in Euclidean space and in representation space $\mathbb{S}^2$. This also shows that the symmetric pose $\pi' = m\pi m^{-1}$ is the "conjugate" of π.

consistent with object deformations and viewpoint changes. In fact, our method builds on [38] and also learns a dense geometric embedding for objects, however, by using a different supervision principle, symmetry.

Symmetry. Computational symmetry [22] has a long history in sciences and played an essential role in several important discoveries including the theory of relativity [29], the double helix structure of DNA [42]. Symmetry is shown to help grouping [19] and recognition [41] in human perception. There is a vast body of computer vision literature dedicated to finding symmetries in images [26], two dimensional [1] and three dimensional shapes [37]. Other axes of variations among symmetry detection methods are whether we seek transformations to map the whole [33] or part of an object [12] to itself; whether distances are measured in the extrinsic Euclidean space [1] or with respect to an intrinsic metric of the surface [33]. In addition to symmetry detection, symmetry is also used as prior information to improve object localization [4], text spotting [47], pose estimation [44] and 3D reconstruction [35]. Symmetry constraints been used to find objects in 3D point clouds [9, 40]. Symmetrization [27] can be used to warp meshes to a symmetric pose. Symmetry cues can be used in segmentation [3, 5]. [2] learns representations that respect a group structure learned from data symmetries.

3 Self-supervised learning of bilateral symmetries

In this section, we extend the approach of [38] to learn the bilateral symmetry of an object category.

Object frame. The key idea of [38] is to study 3D objects not via 3D reconstruction, which is challenging, but by characterizing the correspondences between different 3D shapes of the object, up to pose or intra-class variations.

In this model, an *object category* is a space Π of homeomorphisms $\pi : \mathbb{S}^2 \to \mathbb{R}^3$ that embed the sphere $\mathbb{S}^2$ into $\mathbb{R}^3$. Each possible *shape* of the object is obtained as the (mathematical) image $S = \pi[\mathbb{S}^2]$ under a corresponding function $\pi \in \Pi$, which we therefore call a *pose* of the object (different poses may result in the same shape). The correspondences between a pair of shapes $S = \pi[\mathbb{S}^2]$ and $S' = \pi'[\mathbb{S}^2]$ is then given by $\pi' \circ \pi^{-1}$, which is a bijective deformation of S into S'.

Next, we study how poses relate to images of the object. A (color) image is a function $\mathbf{x} : \Omega \to \mathbb{R}^3$ mapping pixels $u \in \Omega$ to colors $\mathbf{x}_u$. Suppose that $\mathbf{x}$ is the image of the object under pose π; then, a point $z \in \mathbb{S}^2$ on the sphere projects to a point $\pi z \in \mathbb{R}^3$ on the object surface S and the latter projects to a pixel $u = \mathrm{Proj}(\pi z) \in \Omega$, where Proj is the camera projection operator.

The idea of [38] is to learn a function $\psi_u(\mathbf{x})$ that "reverses" this process and, given a pixel u in image $\mathbf{x}$, recovers the corresponding point z on the sphere (so that $\forall u : u = \mathrm{Proj}\left(\pi\psi_u(\mathbf{x})\right)$). The intuition is that z identifies a certain object landmark (e.g. the corner of the left eye in a face) and that the function $\psi_u(\mathbf{x})$ recovers which landmark lands at a certain pixel u.

The way the function $\psi_u(\mathbf{x})$ is learned is by considering pairs of images $\mathbf{x}$ and $\mathbf{x}' = t\mathbf{x}$ related by a *known* 2D deformation $t : \Omega \to \Omega$ (where the warped image $t\mathbf{x}$ is given by $(t\mathbf{x})_u = \mathbf{x}_{t^{-1}u}$). In this manner, pixels u and $u' = tu$ are images of the *same* object landmark and therefore must project on the same sphere point. In formulas, and ignoring visibility effects and other complications, the learned function must satisfy the *invariance constraint*:

$$\forall u \in \Omega : \quad \psi_u(\mathbf{x}) = \psi_{tu}(t\mathbf{x}) \tag{1}$$

In practice, triplets $(\mathbf{x}, \mathbf{x}', t)$ are obtained by *randomly sampling* 2D warps t, assuming that the latter approximate warps that could arise form an actual pose change $\pi' \circ \pi^{-1}$. In this manner, knowledge of t is automatic and the method can be used in an unsupervised setting.

Symmetric object frame. So far the object frame has been used to learn correspondences between different object poses; here, we show that it can be used to establish auto-correspondences in order to model object symmetries as well.

Consider in particular an object that has a *bilateral symmetry*. This symmetry is generated by a reflection operator, say the function $m : \mathbb{R}^3 \to \mathbb{R}^3$ that flips the first axis:

$$m : \qquad \mathbb{R}^3 \to \mathbb{R}^3, \qquad \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \mapsto \begin{bmatrix} -p_1 \\ p_2 \\ p_3 \end{bmatrix}. \tag{2}$$

If S is a shape of a bilaterally-symmetric object, no matter how we align S to the symmetry plane, in general $m[S] \neq S$ due to object deformations. However, we can expect $m[S]$ to still be a valid shape for the object. Consider the example of fig. 2 of a person with his/her right hand raised; if we apply m to this shape, we obtain the shape of a person with the left hand raised, which is valid.

However, reasoning about shapes is insufficient to apply the object frame model; we require instead to work with correspondences, encoded by poses. Unfortunately, even though $m[S]$ is a valid shape, m is *not* a valid correspondence as it flips the left and right sides of a person, which is not a "physical" deformation (why this is important will be clearer later; intuitively it is the reason why we can tell our left hand from the right by looking).

Our key intuition is that we can *learn* the pose representation in such a way that the correct correspondences are trivially expressible there. Namely, assume that m applied to the sphere amounts to swapping each left landmark of the object with its corresponding right counterpart. The correct deformation t that maps the "right arm raised" pose to the "left arm raised" pose can now be found by applying m first in the normalized object frame (to swap left and right sides while leaving the shape unchanged) and then again in 3D space (undoing the swap while actually deforming the shape). This two-step process is visualised in fig. 2 right.

This derivation is captured by a simple change to constraint (1), encoding equivariance rather than invariance w.r.t. the warp m:

$$\forall u \in \Omega : \quad m\psi_u(\mathbf{x}) = \psi_{mu}(m\mathbf{x}) \tag{3}$$

We will show that this simple variant of eq. (1) can be used to learn a representation of the bilateral symmetry of the object category.

Learning formulation. We follow [38] and learn the model $\psi_u(\mathbf{x})$ by considering a dataset of images $\mathbf{x}$ of a certain object category, modelling the function $\psi_u(\mathbf{x})$ by a convolutional neural network, and formulating learning as a Siamese configuration, combining constraints (3) and (1) into a single loss. To avoid learning the trivial solution where $\psi_u(\mathbf{x})$ is the constant function, the constraints are extended to capture not just invariance/equivariance but also distinctiveness (namely, equalities (3) and (1) should *not* hold if u is replaced with a different pixel v in the left-hand side). Following [38], this is captured probabilistically by the loss:

$$\mathcal{L}(\mathbf{x}, m, t) = \int_\Omega \|v - mtu\|_2^\gamma p(v|u)\, dvdu, \quad p(v|u) = \frac{\exp\langle m\psi_u(\mathbf{x}), \psi_v(mt\mathbf{x})\rangle}{\int \exp\langle m\psi_u(\mathbf{x}), \psi_w(mt\mathbf{x})\rangle\, dw} \tag{4}$$

The probability $p(v|u)$ represents the model's belief that pixel u in image $\mathbf{x}$ matches pixel v in image $mt\mathbf{x}$ based on the learned embedding function; the latter is relaxed to span $\mathbb{R}^3$ rather than only $\mathbb{S}^2$ to allow the length of the embedding vectors to encode the belief strength (as shorter vectors results in flatter distributions $p(v|u)$). For unsupervised training, warps $t \sim T$ are randomly sampled from a fixed distribution T as in [38], whereas m is set to be either the identity or the reflection along the first axis with 50% probability.

4 Theory

In the previous section, we have given a formulation for learning the bilateral symmetry of an object category, relying mostly on an intuitive derivation. In this section, we develop the underlying theory in a more rigorous manner (proofs can be found in the supplementary material), while clarifying three important points: how to model symmetries other than the bilateral one, why symmetries such as radial result in ambiguities in establishing correspondences and why this is usually not the case for the bilateral symmetry, and what can be done to handle such ambiguities in the learning formulation when they arise.

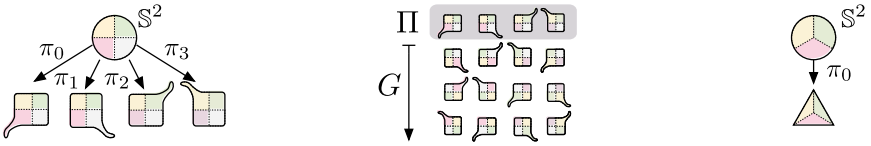


Figure 3: Left: a set $\Pi = \{\pi_0, \dots, \pi_3\}$ of four poses with rotational symmetry group $H = \{h^k, k = 0, 1, 2, 3\}$ where h is a rotation by $\pi/2$. Note that none of the shapes is symmetric; rather, the object, which stays "upright", can deform in four symmetric ways. The shape of the object is then sufficient to recover the pose uniquely. Middle: closure of the pose space Π by rotations $G = H$. Now pose can be recovered from shapes only up to the symmetry group H. Right: an equilateral triangle is represented by a pose π_0 invariant to conjugation by 60 degrees rotations (which are the "ordinary" extrinsic symmetries of this object).

Symmetric pose spaces. A *symmetry* of a shape $S \subset \mathbb{R}^3$ is often defined as an isometry[1] $h : \mathbb{R}^3 \to \mathbb{R}^3$ that leaves the set invariant, i.e. $h[S] = S$. This definition is not very useful when dealing with symmetric but deformable objects, as it works only for special poses (cf. the Vitruvian Man); we require instead a definition of symmetry that is not pose dependent. A common approach is to define *intrinsic symmetries* [33] as maps $h : S \to S$ that preserve the geodesic distance d_S defined on the surface of the object (i.e. $\forall p, q \in S : d_S(hp, hq) = d_S(p, q)$). This works because the geodesic distance captures the intrinsic geometry of the shape, which is pose invariant (but elastic shape deformations are still a problem); however, using this definition requires to accurately reconstruct the 3D shape of objects from images, which is very challenging.

In order to sidestep this difficulty, we propose to study the symmetry not of the 3D shapes of objects, but rather of the space of their deformations. As discussed in section 3, such deformations are captured as a whole by the pose space Π. We define the *symmetries* of the pose space Π as the subset of linear isometries that leave Π unchanged via conjugation:

$$H(\Pi) = \{h \in O(3) : \forall \pi \in \Pi : h\pi h^{-1} \in \Pi \wedge h^{-1}\pi h \in \Pi\}.$$

For example, in fig. 2 we have obtained the "left hand raised" pose π' from the "right hand raised" pose via conjugation $\pi' = m\pi m^{-1}$ via the reflection m (note that $m = m^{-1}$).

Lemma 1. *The set $H(\Pi)$ is a subgroup of $O(3)$.*

The symmetry group $H(\Pi)$ partitions Π in equivalence classes of symmetric poses: two poses π and π' are symmetric, denoted $\pi \sim_{H(\Pi)} \pi'$, if, and only if, $\pi' = h\pi h^{-1}$ for an $h \in H(\Pi)$. In fact:

Lemma 2. *$\pi \sim_{H(\Pi)} \pi'$ is an equivalence relation on the space of poses Π.*

Figure 3 shows an example of an object Π that has four rotationally-symmetric poses $H(\Pi) = \{h^k \pi_0 h^{-k}, k = 0, 1, 2, 3\}$ where h is a clockwise rotation of 90 degrees.

Motion-induced ambiguities. In the example of fig. 3, the object is pinned at the origin of $\mathbb{R}^3$ and cannot rotate (it can only be "upright"); in order to allow it to move around, we can extend the pose space to $\Pi' = G\Pi$ by applying further transformations to the poses. For example, choosing $G = SE(3)$ to be the Euclidean group allows the object to move rigidly; fig. 3-middle shows an example in which $G = H(\Pi)$ is the same group of four rotations as before, so the object is still pinned at the origin but not necessarily upright.

Motions are important because they induce ambiguities in pose recover. We formalise this concept next. First, we note that, if G contains $H(\Pi)$, extending Π by G preserves all the symmetries:

Lemma 3. *If $H(\Pi) \subset G$, then $H(\Pi) \subset H(G\Pi)$.*

Second, consider being given a shape S (intended as a subset of $\mathbb{R}^3$) and being tasked with recovering the pose $\pi \in \Pi$ that generates $S = \pi[\mathbb{S}^2]$. Motions makes this recovery ambiguous:

Lemma 4. *Let the pose space Π be closed under a transformation group G, in the sense that $G\Pi = \Pi$. Then, if pose $\pi \in \Pi$ is a solution of the equation $S = \pi[\mathbb{S}^2]$ and if $h \in H(\Pi) \cap G$, then πh^{-1} is another pose that solves the same equation.*

[1] I.e. $\forall p, q \in \mathbb{R}^3 : d(hp, hq) = d(p, q)$.

Lemma 4 does not necessarily provide a complete characterization of all the ambiguities in identifying pose π from shape S; rather, it captures the ambiguities arising from the symmetry of the object and its ability to move around in a certain manner. Nevertheless, it is possible for specific poses to result in further ambiguities (e.g. consider a pose that deforms an object into a sphere).

In order to use the lemma to characterise ambiguities in pose recovery, given a pose space Π one must still find the space of possible motions G. We can take the latter to be the maximal subgroup $G^* \subset SE(3)$ of rigid motions under which Π is closed[2]

4.1 Bilateral symmetry

Bilateral symmetries are generated by the reflection operator m of eq. (2): a pose space Π has bilateral symmetry if $H(\Pi) = \{1, m\}$, which induces pairs of symmetric poses $\pi' = m\pi m^{-1}$ as in fig. 2.

Even if poses Π are closed under rigid motions (i.e. $G^*\Pi = \Pi$ where $G^* = SE(3)$), in this case there is generally no ambiguity in recovering the object pose from its shape S. The reason is that in lemma 4 one has $G^* \cap H(\Pi) = \{1\}$ due to the fact that all transformations in G^* are orientation-preserving whereas m is not. This explains why it is possible to still distinguish left from right sides in most bilaterally-symmetric objects despite symmetries and motions. However, this is not the case for other types of symmetries such as radial.

Symmetry plane. Note that, given a pair of symmetric poses (π, π'), $\pi' = m\pi m^{-1}$, the correspondences between the underlying 3D shapes are given by the map $m_\pi : \quad S \to m[S], \quad p \mapsto (m\pi m^{-1}\pi^{-1})(p)$. For example, in fig. 2 this map sends the raised left hand of a person to the lowered left hand in the symmetric pose. Of particular interest are the points where m_π coincides with m as they are on the "plane of symmetry". In fact, let $p = \pi(z)$; then:

$$m_\pi(p) = m(p) \quad \Rightarrow \quad m\pi m^{-1}\pi^{-1}(p) = m(p) \quad \Rightarrow \quad m^{-1}(z) = z \quad \Rightarrow \quad z = \begin{bmatrix} 0 \\ z_2 \\ z_3 \end{bmatrix}. \quad (5)$$

4.2 Extrinsic symmetries

Our formulation captures the standard notion of extrinsic (standard) symmetries as well. If $H(S) = \{h \in O(3) : h[S] = S\}$ are the extrinsic symmetries of a geometric shape S (say regular pyramid), we can parametrize S using a single pose $\Pi = \{\pi_0\}$ that: (i) generates the shape ($S = \pi_0[\mathbb{S}^2]$) and (ii) has the same symmetries as the latter ($H(\Pi) = H(S)$).

In this case, the pose π_0 is self-conjugate, in the sense that $\pi_0 = h\pi_0 h^{-1}$ for all $h \in H(\Pi)$. Furthermore, given S it is obviously possible to recover the pose uniquely (since there is only one element in Π); however, as before ambiguities arise by augmenting poses via rigid motions $G = SE(3)$. In this case, due to lemma 4, if $g\pi_0$ is a possible pose of S, so must be $g\pi_0 h^{-1}$. We can rewrite the latter as $(gh^{-1})(h\pi_0 h^{-1}) = (gh^{-1})\pi_0$, which shows that the ambiguous poses are obtained via selected rigid motions gh^{-1} of the reference pose π_0.

5 Learning with ambiguities

In section 3 we have explained how the learning formulation of [38] can be extended in order to learn objects with a bilateral symmetry. The latter is an example where symmetries do not induce an ambiguities in the recovery of the object's pose (the reason is given in section 4.1). Now we consider the case in which symmetries induce a genuine ambiguity in pose recovery.

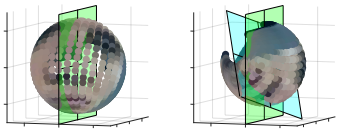
Recall that ambiguities arise from a non-empty intersection of object symmetries $H(\Pi)$ and object motions G^* (section 4). A typical example may be an object with a finite rotational symmetry group (fig. 3). In this case, it is *not* possible to recover the object pose uniquely from an image, which in turn suggests that $\psi_u(\mathbf{x})$ cannot be learned using the formulation of section 3.

[2]Being maximal means that $G^*\Pi = G^* \wedge G\Pi = G \Rightarrow G \subset G^*$. The maximal group can be constructed as $G^* = \langle G \subset SE(3) : G\Pi = \Pi \rangle$, where $\subset$ denotes a subgroup and $\langle \cdot \rangle$ the generated subgroup. This definition is well posed: the generated group G^* contains all the other subgroups G so it is maximal; furthermore $G^*\Pi = \Pi$ because, for any pose $\pi \in \Pi$ and finite combination of other group elements, $g_1^{n_1} \dots g_k^{n_k}\pi \in \Pi$.

Method	Eyes	Mouth
[38]	23.29	15.27
[38] & plane est.	5.17	5.38
Ours	3.21	3.47

(a) Pixel error when using the reflected descriptor from the left eye or left mouth corner to locate its counterpart on the right side of the face, across 200 images from CelebA (MAFL test subset)

(b) Visualisation of fig. 4a. +: ground truth. ○, ●: [38] with no learned symmetry. ○, ●: [38] with mirroring around the plane estimated using annotations. ○, ●: Our method. Where ○, ● is eye, mouth respectively



(c) Difference between us (left) and [38] (right). We learn an axis aligned frame symmetric around a plane (green), [38] has arbitrary rotation and no guaranteed symmetry plane. But we can estimate a plane using annotations (cyan).

Figure 4: Comparing object frames

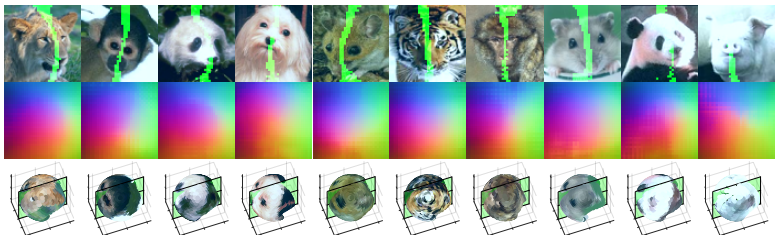


Figure 5: **Bilateral symmetry of animal faces.** The discovered plane of symmetry is shown in green. Top: Inputs, Middle: Colour mapping, Bottom: Embedding (sphere) space

We propose to address this problem by *relaxing* loss (4) in order to discount the ambiguity as follows:

$$\mathcal{L}_{H(\Pi)}(\mathbf{x}, t) = \min_{h \in H(\Pi)} \int_{\Omega} \|v - tu\|_2^{\gamma} p_h(v|u)\, dvdu, \quad p_h(v|u) = \frac{\exp\langle h\psi_u(\mathbf{x}), \psi_v(t\mathbf{x})\rangle}{\int \exp\langle h\psi_u(\mathbf{x}), \psi_w(t\mathbf{x})\rangle\, dw} \quad (6)$$

This loss allows $\psi_u(\mathbf{x})$ to estimate the embedding vector $z \in \mathbb{S}^2$ (or $z \in \mathbb{R}^3$) up to an unknown transformation h.

6 Experiments

We now validate empirically our formulation. To ensure that we have a fair comparison to [38], who introduced learning formulation (4) which our approach extends, we use the same network architecture and hyperparameter values (*e.g.* $\gamma = 0.5$ in eq. (4)). We show that our extension successfully recovers the symmetric structure of bilateral objects (section 6.1) as well as allowing to manage ambiguities arising from symmetries in learning such structures (section 6.2).

6.1 Learning objects with bilateral symmetry

In this section, we apply the learning formulation (4) to objects with a bilateral symmetry. Due to the structure imposed on the embedding function by eq. (3), we expect the symmetry plane of the object to be mapped to the plane $z_1 = 0$ in the embedding space (section 4.1). Once the model is learned, this locus can be projected back to an image for visualisation and qualitative assessment. We also test quantitatively the accuracy of the learned geometric embedding in localising object landmarks and their symmetric counterparts.

Faces. We evaluate the proposed formulation on faces of humans and animals, which have limited out-of-plane rotations. For humans we use the CelebA [23] face dataset, with over 200K images. We use an identical setup to [38, 39], training on 162K images and employing the MAFL [46] subset of 1000 images as a validation set. For cats we use the Cat Head dataset [45], with 8609 training images. We also combine multiple animals in the same training set, with Animal Faces dataset [36] (20 animal classes, about 100 images per class). We exclude birds and elephants since these images have a significantly different appearance, and add additional cat, dog and human faces [45, 31, 23] (but keep roughly the same distribution of animal classes per batch as the original dataset).

In all cases, we do not use any manual annotation; instead, we use learning formulation (4) using the same synthetic transformations $t \sim \mathcal{T}$ as [38]. Additionally, with 50% probability we also apply a left-to-right flip m to both the image and the embedding space, as prescribed by eq. (4).

Results (figs. 1 and 5) show that our method, like [38], learns a geometric embedding of the object invariant to viewpoint and intra-category changes. In addition, our new formulation localises the intrinsic bilateral symmetry plane in the face images and maps it to a plane of reflection in the embedding space. We note that images are embedded symmetrically with respect to the plane (shown in green in fig. 1, bottom row). The plane can also be projected back to the image and, as predicted by eq. (5), corresponds to our intuitive notion of symmetry plane in faces (fig. 1, top row). Importantly, symmetry here is a statistical concept that applies to the category as a whole; specific face instances need not *be* nor *appear* symmetric — the latter in particular means that faces need not be imaged fronto-parallel for the method to capture their symmetry.

To evaluate the learned symmetry quantitatively we use manual annotations (eyes, mouth corners) to verify if the representation can transport landmarks to their symmetric counterparts. In particular, we take landmarks on the left side of the face (*e.g.* left eye), use m (eq. (3)) to mirror their embedding vectors, backproject those to the image, and compare the resulting positions to the ground-truth symmetric landmark locations (*e.g.* right eye). We report the measured pixel error in fig. 4a. As a baseline, we replace our embedding function with the one from [38] which results in much higher error. This is however expected as the mapping m has no particular meaning in this embedding space; for a fairer comparison, we then explicitly estimate an ad-hoc plane of symmetry defined by the nose, mean of the eyes, and mean of the mouth corners, using 200 training images. This still gives higher error than our method, showing that enforcing symmetry during training leads to a better representation of symmetric objects.

In terms of the accuracy of the geometric embedding as such, we evaluate simply matching annotations between different images and obtain similar error to the embedding of [38] (ours 2.60, theirs 2.59 pixel error on 200 pairs of faces, and both 1.63 error for when the second image is a warped version of the first). Hence representing symmetries does not harm geometric accuracy.

We also examine the influence of the synthetic warp intensity, in fig. 6 we train for 5 epochs scaling the original control point parameters by a factor, indicating we are around the sweet spot and unnatural excessive warping is harmful.

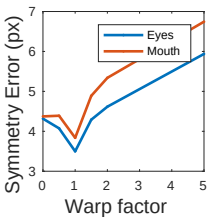


Figure 6: Varying warp intensity

Synthetic 3D car model. A challenging problem is capturing bilateral symmetry across out-of-plane rotations. We use a 3D car, animated with random motion [13] for 30K frames. The heading follows a random walk, eventually rotating 360° out of plane. Translation, pitch and roll are sinusoidal. The back of the car is red to easily distinguish from the front. We use consecutive frames for training, with the ground truth optical flow used for t and image size 75×75. The loss ignores pixels with flow smaller than 0.001, preventing confusion with the solid background. Figure 8 depicts examples from this dataset. Unlike CelebA, the cars are rendered from significantly different views, but our method can successfully localize the bilateral axis accurately.

Synthetic robot arm model. We trained our model on videos of a left-right pair of robotics arms, extending the setup of [38] to a system of two arms.

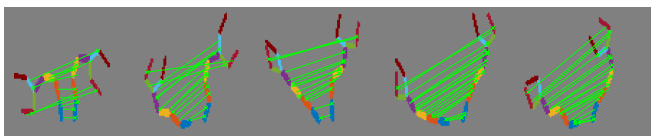


Figure 7: Symmetry in a pair of toy robotics arms

Figure 7 shows the discovered symmetry by joining corresponding points in a few video frames. Note that symmetries are learned automatically from raw videos and ground truth optical flow alone. Note also that none of the images is symmetric in the trivial left-right flip sense due to the object deformations.

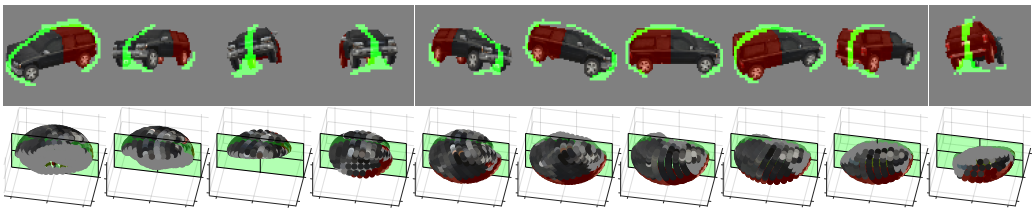


Figure 8: **Bilateral symmetry on synthetic car images**, Top: Input images with the axis of symmetry superimposed (shown in green), Bottom: Image pixels mapped to 3D with the reflection plane (green)

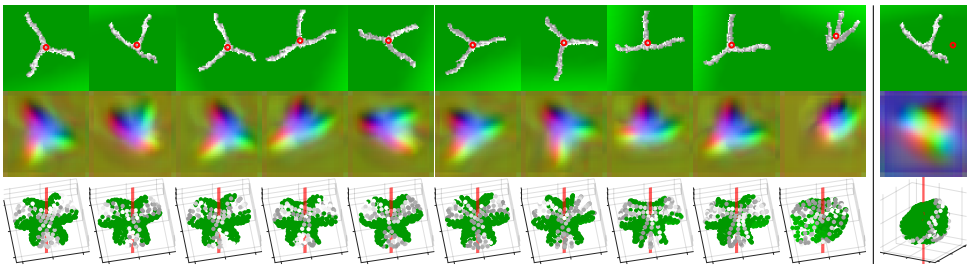


Figure 9: **Rotational symmetry on protein.** Top: Frames, found center of symmetry red. Middle: Colorized object frame, a different colouring is assigned to each leg despite ambiguity. Bottom: Embedding in 3D, it learns to be symmetric around an axis (red). Last column: Without relaxed loss.

6.2 Rotational symmetry

We create an example based on 3-fold rotational symmetry in nature, the Clathrin protein [43]. We use the protein mesh[3] and animate it as a soft body in a physics engine [13, 7], generating 200 400-frame sequences. For each we vary the camera rotation, lighting, mesh smoothing and position. The protein is anchored at its centre. We vary the gravity vector to produce varied motion.

We train using the relaxed loss in eq. (6), where $H(\Pi)$ corresponds to rotating our sphere $0°$, $120°$ or $240°$. The mapping then need only be learned up to this rotational ambiguity. As shown in fig. 9, this maps the protein images onto a canonical position which has rotational symmetry around the chosen axis, whereas without the relaxed loss the object frame is not aligned and symmetrical.

We also show results for rotational symmetry in real images, using flower class Stapelia from ImageNet in fig. 10 which has 5-fold rotational symmetry.

7 Conclusions

In this paper we have developed a new model of the symmetries of deformable object categories. The main advantage of this approach is that it is flexible and robust enough that it supports learning symmetric objects in an unsupervised manner, from raw images, despite variable viewpoint, deformations, and intra-class variations. We have also characterised ambiguities in pose recovery caused by symmetries and developed a learning formulation that can handle them. Our contributions have been validated empirically, showing that we can learn to represent symmetries robustly on a variety of object categories, while retaining the accuracy of the learned geometric embedding compared to previous approaches.

[3] `https://www.rcsb.org/structure/3LVG`

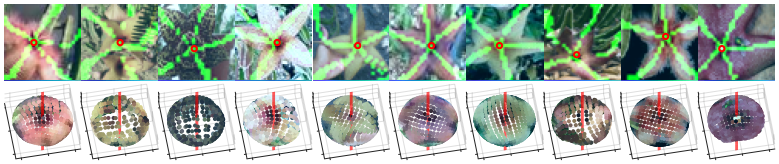


Figure 10: **Rotational symmetry on Stapelia flower.** Superimposed in green, projection into the image of a set of half-planes $72°$ apart in the sphere space. In red, predicted axis of rotational symmetry.

Acknowledgments: This work acknowledges the support of the AIMS CDT (EPSRC EP/L015897/1) and ERC 638009-IDIU. We thank Almut Sophia Koepke for feedback and corrections.

References

[1] Helmut Alt, Kurt Mehlhorn, Hubert Wagener, and Emo Welzl. Congruence, similarity, and symmetries of geometric objects. *Discrete & Computational Geometry*, 3(3):237–256, 1988.

[2] Fabio Anselmi, Georgios Evangelopoulos, Lorenzo Rosasco, and Tomaso Poggio. Symmetry-adapted representation learning. *Pattern Recognition*, 86:201–208, 2019.

[3] Shai Bagon, Oren Boiman, and Michal Irani. What is a good image segment? a unified approach to segment extraction. In *Proc. ECCV*, pages 30–44. Springer, 2008.

[4] Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars. Weakly supervised object detection with posterior regularization. In *Proceedings BMVC 2014*, pages 1–12, 2014.

[5] Oren Boiman and Michal Irani. Similarity by composition. In *Proc. NeurIPS*, pages 177–184, 2007.

[6] T F Cootes, C J Taylor, D H Cooper, and J Graham. Active shape models: their training and application. *CVIU*, 1995.

[7] Erwin Coumans. Bullet physics engine. *Open Source Software: http://bulletphysics. org*, 2010.

[8] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *Proc. CVPR*, 2005.

[9] Aleksandrs Ecins, Cornelia Fermüller, and Yiannis Aloimonos. Cluttered scene segmentation using the symmetry constraint. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 2271–2278. IEEE, 2016.

[10] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part Based Models. *PAMI*, 2010.

[11] Rob Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, 2003.

[12] Ran Gal and Daniel Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics (TOG)*, 25(1):130–150, 2006.

[13] Mike Goslin and Mark R Mine. The Panda3D graphics engine. *Computer*, 37(10):112–114, 2004.

[14] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow. In *Proc. CVPR*, pages 3475–3484, 2016.

[15] Kai Han, Rafael S Rezende, Bumsub Ham, Kwan-Yee K Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. Scnet: Learning semantic correspondence. In *Proc. ICCV*, 2017.

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial Transformer Networks. In *Proc. NeurIPS*, 2015.

[17] A. Kanazawa, D. W. Jacobs, and M. Chandraker. WarpNet: Weakly supervised matching for single-view reconstruction. In *Proc. CVPR*, 2016.

[18] Ira Kemelmacher-Shlizerman and Steven M. Seitz. Collection flow. In *Proc. CVPR*, 2012.

[19] Kurt Koffka. *Principles of Gestalt psychology*, volume 44. Routledge, 2013.

[20] Erik G Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[21] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on statistical learning in computer vision, ECCV*, 2004.

[22] Yanxi Liu, Hagit Hel-Or, Craig S Kaplan, Luc Van Gool, et al. Computational symmetry in computer vision and computer graphics. *Foundations and Trends® in Computer Graphics and Vision*, 5(1–2):1–195, 2010.

[23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proc. ICCV*, 2015.

[24] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2014.

[25] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[26] Giovanni Marola. On the detection of the axes of symmetry of symmetric and almost symmetric planar images. *PAMI*, 11(1):104–108, 1989.

[27] Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. Symmetrization. In *ACM Transactions on Graphics (TOG)*, volume 26, page 63. ACM, 2007.

[28] Hossein Mobahi, Ce Liu, and William T. Freeman. A Compositional Model for Low-Dimensional Image Set Representation. *Proc. CVPR*, 2014.

[29] Gregory L Naber. *The geometry of Minkowski spacetime: An introduction to the mathematics of the special theory of relativity*, volume 92. Springer Science & Business Media, 2012.

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

[31] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *Proc. CVPR*, 2012.

[32] Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *PAMI*, 34(11):2233–2246, 2012.

[33] Dan Raviv, Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Full and partial symmetries of non-rigid shapes. *IJCV*, 89(1):18–39, 2010.

[34] I. Rocco, R. Arandjelović, and J. Sivic. Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, 2017.

[35] Ilan Shimshoni, Yael Moses, and Michael Lindenbaum. Shape reconstruction of 3d bilaterally symmetric surfaces. *IJCV*, 39(2):97–110, 2000.

[36] Zhangzhang Si and Song-Chun Zhu. Learning hybrid image templates (hit) by information projection. *PAMI*.

[37] Changming Sun and Jamie Sherrah. 3d symmetry detection using the extended gaussian image. *PAMI*, 19(2):164–168, 1997.

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

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

[40] Sebastian Thrun and Ben Wegbreit. Shape from symmetry. In *Proc. ICCV*, pages 1824–1831, 2005.

[41] Thomas Vetter and Tomaso Poggio. Linear object classes and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):733–742, 1997.

[42] James D Watson, Francis HC Crick, et al. Molecular structure of nucleic acids. *Nature*, 171(4356):737–738, 1953.

[43] Jeremy D Wilbur, Peter K Hwang, Joel A Ybe, Michael Lane, Benjamin D Sellers, Matthew P Jacobson, Robert J Fletterick, and Frances M Brodsky. Conformation switching of clathrin light chain regulates clathrin lattice assembly. *Developmental cell*, 18(5):854–861, 2010.

[44] Heng Yang and Ioannis Patras. Mirror, mirror on the wall, tell me, is the error small? In *Proc. CVPR*, pages 4685–4693, 2015.

[45] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection - How to effectively exploit shape and texture features. In *Proc. ECCV*, 2008.

[46] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Learning Deep Representation for Face Alignment with Auxiliary Attributes. *PAMI*, 2016.

[47] Zheng Zhang, Wei Shen, Cong Yao, and Xiang Bai. Symmetry-based text line detection in natural scenes. In *Proc. CVPR*, pages 2558–2567, 2015.

Supplementary Material: Modelling and unsupervised learning of symmetric deformable object categories

We show additional qualitative results learning bilateral symmetry on several datasets. Firstly, we try a more challenging setting of the CelebA dataset, by applying rotations with standard deviation of 30 degrees and translations with standard deviation 20% of image width. As shown in fig. 11, our method remains able to learn and recover the axis of symmetry under these conditions.

Secondly, we use an exercise dataset of human pose[1]. Here (fig. 12) the symmetry is recovered accurately with upright pose and certain deformations, but fails in extreme cases.

Finally, we attempt to learn bilateral symmetry on cars, using the CompCars dataset[2]. We observe that, although the symmetry is recovered with frontal images, the plane through the middle of the car seen from side is mistakenly thought to be a symmetry. This is understandable, since we train only using synthetic warps of the same image, so it hard to build up a globally consistent frame. Similarly, the front and back of the car are not disambiguated from each other.

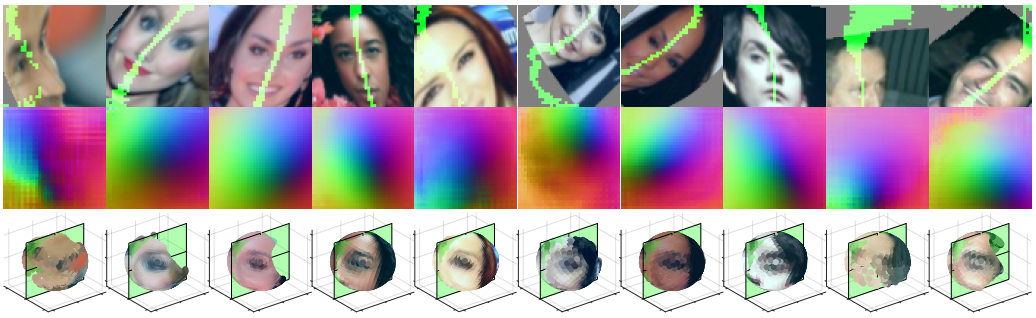


Figure 11: **CelebA trained with large distortions**

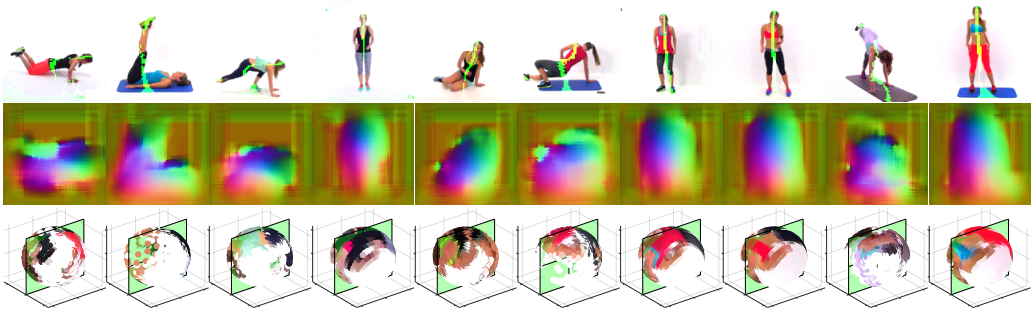


Figure 12: **Bilateral symmetry on humans**

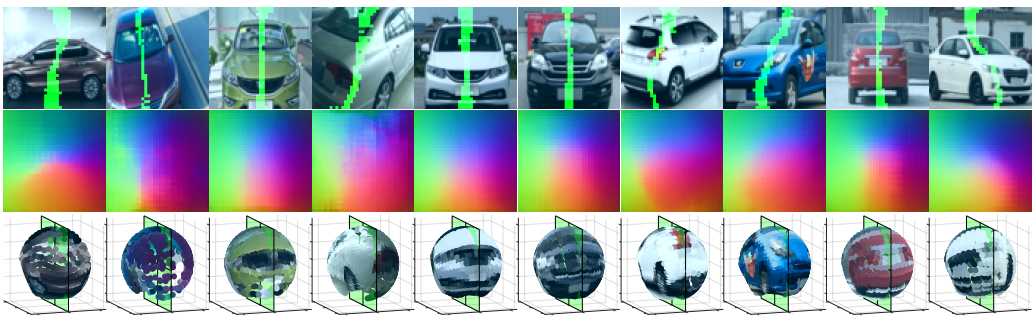


Figure 13: **Bilateral symmetry on cars**

[1]Xue, Tianfan and Wu, Jiajun and Bouman, Katherine L and Freeman, William T. Visual Dynamics: Probabilistic Future Frame Synthesis via Cross Convolutional Networks. In NIPS 2016.

[2]Linjie Yang, Ping Luo, Chen Change Loy, Xiaoou Tang. A Large-Scale Car Dataset for Fine-Grained Categorization and Verification, In CVPR 2015

A Proofs for Section 4 (Theory)

Lemma 1. *The set $H(\Pi)$ is a subgroup of $O(3)$.*

Proof. First, note that, since $O(3)$ is the space of extrinsic symmetries of the sphere $\mathbb{S}^2$, then $\mathbb{S}^2 = h\mathbb{S}^2 = h^{-1}\mathbb{S}^2$. This means that the function composition $h\pi h^{-1}$ is well defined. Furthermore, the identity map $h = 1$ is clearly included in $H(\Pi)$, which is therefore not empty. The set is also closed under composition: if $h_1, h_2 \in H(\Pi)$, then using associativity $(h_1h_2)\pi(h_1h_2)^{-1} = h_1(h_2\pi h_2^{-1})h_1^{-1}$ shows that $h_1h_2 \in H(\Pi)$. It is also closed under inversion: if $h \in H(\Pi)$, then $h^{-1} \in H(\Pi)$ due to the symmetry in the definition. □

Lemma 2. *$\pi \sim_{H(\Pi)} \pi'$ is an equivalence relation on the space of poses Π.*

Proof. The relation is reflexive because $H(\Pi)$ is a group and thus contain the identity element. It is symmetric because $\pi' = h\pi h^{-1} \Rightarrow \pi = h^{-1}\pi' h$ and $h^{-1} \in H(\Pi)$ as a group is closed under inversion. It is transitive because if $\pi'' = h_2\pi' h_2^{-1}$ and $\pi' = h_1\pi h_1^{-1}$ where $h_1, h_2 \in H(\Pi)$, then $\pi'' = h_2\pi' h_2^{-1} = h_2(h_1\pi h_1^{-1})h_2^{-1} = (h_2h_1)\pi(h_2h_1)-1$ since $h_2h_1 \in H(\Pi)$ as a transformation group is closed under composition. □

Lemma 3. *If $H(\Pi) \subset G$, then $H(\Pi) \subset H(G\Pi)$.*

Proof. Let $h \in H(\Pi)$; we need to show that $h \in H(G\Pi)$. To this end, consider the map $r = hgh^{-1}g^{-1}$. We have

$$rg(h\pi h^{-1}) = h(g\pi)h^{-1} \tag{7}$$

By definition, $h\pi h^{-1} \in \Pi$. Furthermore, since $H(\Pi) \subset G$, then $rg = hgh^{-1} \in G$. Hence we conclude that $h(g\pi)h^{-1}$ is contained in $G\Pi$. □

Lemma 4. *Let the pose space Π be closed under a transformation group G, in the sense that $G\Pi = \Pi$. Then, if pose $\pi \in \Pi$ is a solution of the equation $S = \pi[\mathbb{S}^2]$ and if $h \in H(\Pi) \cap G$, then πh^{-1} is another pose that solves the same equation.*

Proof. First, note that the composition πh^{-1} is always well posed since is any orthogonal transformation $h^{-1} \in O(3)$. Hence the range $h^{-1}\mathbb{S}^2$ of h^{-1} is the same as the domain $\mathbb{S}^2$ of π. For the same reason, $\pi h^{-1}\mathbb{S}^2 = \pi\mathbb{S}^2 = S$ have the same shape. To conclude the proof, it remains to show that $\pi h^{-1} \in \Pi$. To this end, note that $\pi h^{-1} = h^{-1}(h\pi h^{-1}) = h^{-1}\pi'$. Since $h \in H(\Pi)$, the map π' belongs to Π by definition of $H(\Pi)$. Since $h \in G$ too, since Π is closed to the action of G, the map $h^{-1}\pi'$ belongs to Π as well. □

Chapter 7

Unsupervised Discovery of Dense Landmarks via Compression of Distinctive Invariant Embeddings

This paper explores the relationship between the low-dimensional semantic embeddings presented in Chapter 5 and more general, high-dimensional SIFT-style descriptors. In particular, we put forward the idea that there is a previously unexplored duality between the two concepts. Our insight is that the generalisation obtained in Chapter 5 functions by constraining the capacity of the embedding (along with the dataset containing a single category). Based on this insight we develop a method to constrain embedding capacity, without having to make the sacrifice of low dimensionality. These high-dimensional, semantic descriptors are then good at both accurate matching and locating semantic concepts (eyes, nose, etc.). We show quantitatively that the learned embeddings can be used to accurately regress annotated landmarks, and we show qualitative results matching analogous parts across animals.

Unsupervised Discovery of Dense Landmarks via Compression of Distinctive Invariant Embeddings

James Thewlis[1], Hakan Bilen[2], and Andrea Vedaldi[1]

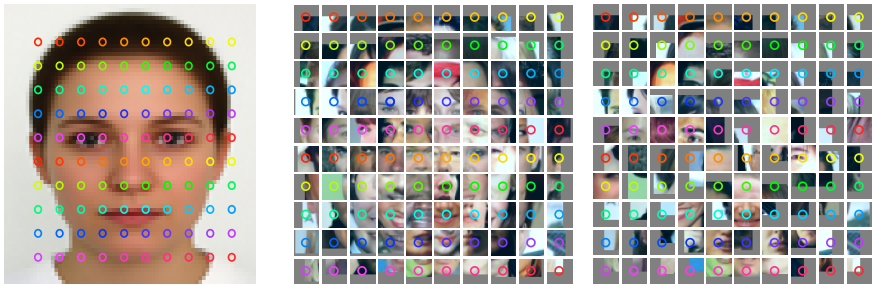
[1] Visual Geometry Group
University of Oxford
{jdt,vedaldi}@robots.ox.ac.uk
[2] School of Informatics
University of Edinburgh
hbilen@ed.ac.uk

Abstract. Recent works have shown that detectors of object landmarks, such as eyes and mouth in faces, emerge without manual supervision by training pixel embeddings that are distinctive and invariant to image deformations. Usually, however, invariance and distinctiveness have been used as defining properties not of landmarks, but of descriptors for image matching. In this paper, we examine more closely this notion and establish for the first time an explicit connection between landmarks and descriptors. We show in particular that landmarks emerge as a limit case of image descriptors as the capacity of the learned embedding is progressively reduced, achieving better generalization to cross-instance appearance variations. We use this insight to propose a new formulation for learning landmarks with no manual supervision. The method is based on the concept of *embedding volume compression*, where embeddings arising from different object instances are encouraged to share the same, small volume of embedding space. We obtain in this manner embeddings that can be simultaneously interpreted as patch descriptors, achieving excellent performance at the task of matching different views of the same object, as well as landmarks, identifying object points consistent for an entire object category.

1 Introduction

In this work, we are interested in the problem of learning in an *unsupervised manner* the structure of object categories such as human or even animal faces. As commonly done in image understanding, we reduce this to the problem to establishing *meaningful correspondences* between different images of the objects.

There are two traditional approaches to establish image correspondences. The first is to extract and match *descriptors* that capture the distinctive appearance of local image patches in a manner which is invariant to nuisance factors such as a viewpoint change. The second is to detect object *landmarks*, i.e. occurrences



Average face With EVC Without EVC

Fig. 1. Visualizations of unsupervised landmark embeddings learned using the proposed EVC method. Left: an average face[4] with a grid of pixels selected. Middle: learned pixel embeddings are sampled from a grid overlaid to an average face image is replaced by a random patch with a similar embedding vector: despite intra-class variations, all substitutions are geometrically sound. Right: without EVC the embedding does not bridge intra-class variations properly.

of characteristic points such as the nose or mouth in a face which are valid for all images of an object category.

Descriptors and landmarks are often thought to be different and complementary notions. Usually, landmarks are used to identify geometrically stable points in images and descriptors to characterise their appearance. For instance, in face verification one may compute descriptors of landmarks such as the nose and the eyes of a person to describe identity. Interest point detectors such Harris corners can also be thought of as a type of landmark detectors which, combined with descriptors such as SIFT [1], can be used to match generic images.

In all such examples, descriptors are computed *on top* of landmarks. Here, we show that *reversing* this relationship allows one to interpret landmarks as a special case of image descriptors instead.

In order to do this, note as in [2] that the landmarks of an object category can be indexed by the unit sphere[5] $\mathbb{S}^2$. For example, fig. 2 shows three sphere points denoting the left eye, nose and mouth corner in human faces. A landmark detector can then be thought of as a function mapping each image pixel to its corresponding landmark label in the spherical embedding. The authors of [2] show that learning an embedding function which is *invariant* to image deformations and *distinctive* for different pixels is sufficient to learn meaningful landmarks automatically, without manual supervision. Remarkably, the resulting landmarks are robust to cross-instance object variations.

Here, we note that a very similar approach has often been used not to learn landmarks, but local image descriptors. In image matching, in fact, it is common to characterise the appearance of a region around each pixel by mapping them to descriptor vectors in an embedding space $\mathcal{E}$. Like the landmark detector function,

[4] Average face from www.beautycheck.de/cmsms/index.php/durchschnittsgesichter
[5] As a sphere is topologically equivalent to all connected 3D surfaces without holes

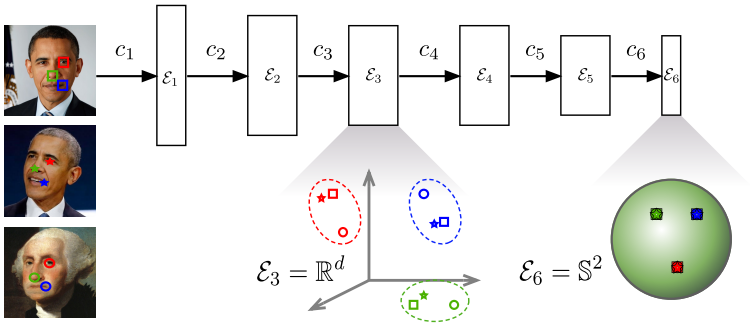


Fig. 2. Descriptors-landmarks duality. A deep neural network computes a hierarchy of dense image descriptors in a hierarchy of neural network embeddings $\mathcal{E}_1, \mathcal{E}_2, \ldots$ of decreasing dimensionality and increasing abstraction. All embeddings are distinctive and invariant, but moving from left to right the lower embedding capacity induces cross-instance alignment between descriptors, until landmarks are learned. We propose a new technique called *embedding volume compression* (EVC) that allows to control capacity directly in a high-dimensional embedding space. The effects is to shrink the dashed ellipses, collapsing embeddings for corresponding landmarks.

this descriptor function must be invariant and distinctive [1, 3] so that pixels in different images can be matched by comparing their descriptors.

We have thus shown that both landmark detectors and descriptor extractors can be characterised as distinctive and invariant pixel embedding functions; if so, what is the difference between them? The answer is in the *generalization properties* of the embedding. Namely, each landmark embedding is valid for all instances of a given object category, whereas a descriptor embedding is valid across a much more limited set of image variations.[6] The key insight of [2] is that generalization can be achieved by explicitly *constraining the capacity of the embedding space* while learning the embedding; by considering the sphere, they in fact set $\mathcal{E} = \mathbb{S}^2$ to be a 2D manifold, whereas descriptor vectors normally use many more dimensions (128D for SIFT). Constraining the capacity in this manner forces the learned embedding function to map similar structures such as the mouth in different faces to the same embedding vector.

This insight is particularly interesting when embeddings are computed by a neural network such as the one of fig. 2. The network in fact produces a *hierarchy of embeddings* $\mathcal{E}_1, \mathcal{E}_2, \ldots$ which, from left to right, increase in abstraction. For the intermediate layers, embeddings are relatively high-dimensional and high-capacity; therefore, they may match points in similar objects well, but may still fail to align points in object instances that differ more. Moving to the last spherical embedding at the end of the network, however, creates a bottleneck

[6] On the other hand, landmarks make sense for a specific category of objects, whereas descriptors may be used for generic image matching as well — generality here refers to the variation between pair of images, not the type of images to which embeddings are applied.

that encourages descriptors to align across object instances, which, as explained before, results in landmarks.

Once the importance of the bottleneck in learning landmarks is understood, constructions alternative to the spherical embedding are apparent. Our key technical contribution is to propose an approach for controlling the embedding capacity which is better than fixing the dimensionality of the embedding space as done by [4, 2]. Our approach is to consider instead a high-dimensional embedding space and control its capacity using a new technique that we call *embedding volume compression* (section 3). The latter builds on the idea that the *set of embedding vectors* extracted from any instance of the same object category should be overall the same and hence interchangeable for the purpose of matching. We show how this constraint can be formulated as a natural extension of [2]. Empirically (fig. 1 and section 4), we show that the key advantage of this formulation is that it produces embedding vectors that simultaneously work well as instance-specific image descriptors *and* intra-category landmarks, capturing in a single representation the advantages of both descriptors and landmarks, and validating our intuition.

2 Related work

General image matching. Image matching based on local features has been an extensively studied problem in the literature with applications to wide-baseline stereo matching [5] and image retrieval [6]. The generic pipeline contains the following steps: i) detecting a sparse set of interest points [7] that are covariant with a class of transformations, ii) extracting local descriptors (*e.g.* [8, 9]) at these points that are invariant to viewpoint and illumination changes, and iii) matching the nearest neighbour descriptors across images with an optional geometric verification. While the majority of the image matching methods rely on hand-crafted detectors and descriptors, recent work show that CNN-based models can successfully be trained to detect covariant detectors [10] and invariant descriptors [3, 11]. We build our method on similar principles, covariance and invariance, but with an important difference that it can learn intrinsic features for object categories in contrast to generic ones.

Cross-instance object matching. The SIFT Flow method [12] extends the problem of finding dense correspondences between same object instances to different instances by matching their SIFT features [8] in a variational framework. This work is further improved by using multi-scale patches [13], establishing region correspondences [14] and replacing SIFT features with CNN ones [15]. In addition, Learned-Miller [16] generalises the dense correspondences between image pairs to arbitrary number of multiple images by continuously warping each image via a parametric transformation. RSA [17], Collection Flow [18] and Mobahi *et al*. [19] project a collection of images into a lower dimensional subspace and perform a joint alignment among the projected images. Anchornet [20] learns semantically meaningful parts across categories, although is trained with image labels.

Transitivity. The use of transitivity to regularise structured data has been proposed by several authors [21–24] in the literature. Earlier examples [21, 22] employ this principle to achieve forward-backward consistency in object tracking and to identify inconsistent geometric relations in structure from motion respectively. Zhou *et al*. [23, 24] enforce a geometric consistency to jointly align image sets or supervise deep neural networks in dense semantic alignment by establishing a cycle between each image pair and a 3D CAD model. Our method also builds on the same general principle of transitivity, however, ours operates in the space of appearance embedding in contrast to verification of subsequent image warps to a composition.

Unsupervised learning of object structure. Visual object characterisation (*e.g*. [25–29]) has a long history in computer vision with extensive work in facial landmark detection and human body pose estimation. A recent unsupervised method that can learn geometric transformations to optimise classification accuracy is the spatial transformer network [30]. However, this method does not learn any explicit object geometry. Similarly, WarpNet [31] and geometric matching networks [32] train neural networks to predict relative transformations between image pairs. These methods are limited to perform only on image pairs and do not learn an invariant geometric embedding for the object. Most related to our work, a recent work [4] characterises objects by learning landmarks that are consistent with geometric transformations without any manual supervision. The same authors extended [4] to extract a dense set of landmarks by projecting the raw pixels on a surface of a sphere in [2]. Similar work [33] leverages frame-to-frame correspondence using Dynamic Fusion [34] as supervision to learn a dense labelling for human images. We build our method on these methods and further extend them in significant ways. First we manage to learn more versatile descriptors that can encode both generic and object-specific landmarks and show that we can gradually learn to move from generic to specific ones. Second we improve the cross-instance generalisation ability by better regularising the embedding space with the use of transitivity. Finally, we show that our method both qualitatively and quantitatively outperforms [4, 2] in facial landmark detection (section 4). A recent work [35] proposes an autoencoder model that automatically discovers landmark coordinates and reconstruct the input image conditioned on them. We compare our method to this work in section 4.

3 Method

This section starts by reviewing and reinterpreting the approach of [2] for learning unsupervised object landmarks, connecting it to the general problem of learning distinctive and invariant embedding functions (section 3.1). It then introduces our extension to control the capacity the embedding space, merging description extraction and landmark detection in a single framework (section 3.2).

3.1 Learning distinctive invariant embeddings

Denote by $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$ an image of an object, by Ω the $H \times W$ lattice forming its domain, and by $u \in \Omega$ an image pixel. In order to recognize the "structure" of the object, consider the problem of mapping pixels u to codes $\mathbf{z}_u = \Phi_u(\mathbf{x}) \in \mathcal{E}$ that identify characteristic points of the object. In [2], the embedding space $\mathcal{E} = \mathbb{S}^2$ is a sphere, as this is sufficient to index all possible landmarks of any object whose shape is a connected surface without holes.

The authors of [2] propose a method to learn the map Φ in an unsupervised manner, using only a dataset $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ of images of a certain object instance or category. They do so by noticing the learned embeddings should be invariant to geometric warps g of the image. Namely, if $g\mathbf{x}$ acts on the image $\mathbf{x}$ by deforming its domain,[7] then the label $\mathbf{z}_u = \Phi_u(\mathbf{x})$ assigned to pixel u of image $\mathbf{x}$ must be the same as label $\mathbf{z}_{gu} = \Phi_{gu}(g\mathbf{x})$ assigned to pixel gu of the deformed image $g\mathbf{x}$. This is because, by construction, pixels u and gu land on the same object point, and thus should be mapped to the same landmark embedding vector. However, this constraint can be trivially satisfied by mapping all pixels to any fixed embedding value, such as the null vector. Hence, one must also make sure that embeddings are informative. The latter can be obtained by enforcing that different pixels are mapped to different embedding vectors (formally, $\Phi_u(\mathbf{x}) = \Phi_v(g\mathbf{x}) \Leftrightarrow v = gu$). We succinctly call the latter property *distinctive invariance*.

The authors of [2] show that the map Φ can be trained by enforcing distinctive invariance for a collection of example images $\mathbf{x} \in \mathcal{X}$ of an object and corresponding *synthetic warps* $g\mathbf{x}$ obtained by applying to these random deformations. Such warps are in fact a proxy to naturally-occurring transformations, as may be induced by a viewpoint change or an object deformation and, since they are synthetic, come with knowledge of the ground-truth correspondence g.

As a concrete formulation of this idea, given images $\mathbf{x}$ and $\mathbf{x}'$, define the probability of pixel u in image $\mathbf{x}$ matching pixel v in image $\mathbf{x}'$ by normalizing the embedding similarity $\langle \Phi_u(\mathbf{x}), \Phi_v(\mathbf{x}')\rangle$:

$$p(v|u;\Phi,\mathbf{x},\mathbf{x}') = \frac{e^{\langle \Phi_u(\mathbf{x}),\Phi_v(\mathbf{x}')\rangle}}{\int_\Omega e^{\langle \Phi_u(\mathbf{x}),\Phi_w(\mathbf{x}')\rangle}\,dw}. \tag{1}$$

Given a warp $g : u \mapsto v$, we can then enforce distinctive invariance by minimizing the loss:

$$\mathcal{L}(\Phi;\mathbf{x},\mathbf{x}',g) = \frac{1}{|\Omega|}\int_\Omega \|v - gu\|\, p(v|u;\Phi)\,du\,dv \tag{2}$$

where $\|v - w\|$ is a suitable distance between pixel coordinates. In order to understand this loss, note that $\mathcal{L}(\Phi;\mathbf{x},\mathbf{x}',g) = 0$ if, and only if, for each pixel $u \in \Omega$, the matching probability $p(v|u;\Phi,\mathbf{x},\mathbf{x}')$ puts all its mass on the corresponding pixel gu. Thus minimizing this loss encourages $p(v|u;\Phi,\mathbf{x},\mathbf{x}')$ to establish correct deterministic correspondences.

[7] Here $g\mathbf{x}$ denotes the image $(g\mathbf{x})_u = \mathbf{x}_{g^{-1}u}$ obtain by inverse warping.

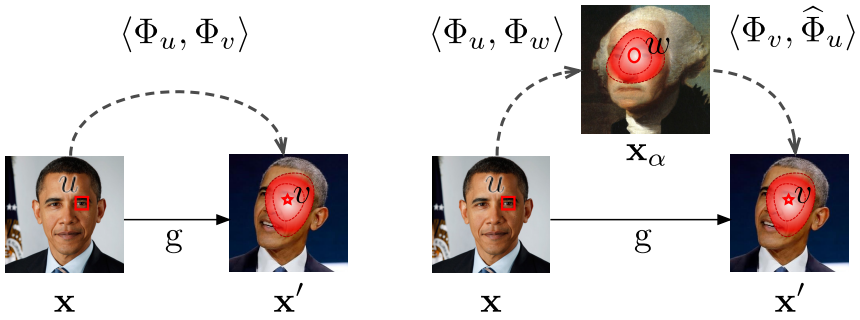


Fig. 3. Embedding volume compression. We learn a dense embedding $\Phi_u(\mathbf{x}) \in \mathbb{R}^d$ of image pixels. The embedding is learned from pairs of images $(\mathbf{x}, \mathbf{x}')$ related by a known warp $v = g(u)$. Left: the approach of [2] directly matches embedding $\Phi_u(\mathbf{x})$ from the left image to embeddings $\Phi_v(\mathbf{x}')$ in the right image. Right: our approach replaces $\Phi_u(\mathbf{x})$ from its reconstruction $\hat{\Phi}_u(\mathbf{x}|\mathbf{x}_\alpha)$ obtained from the embeddings in a third auxiliary image $\mathbf{x}_\alpha$. Importantly, the correspondence with $\mathbf{x}_\alpha$ does not need to be known.

Following [2], learning amounts to optimizeing a Siamese neural network architecture, where $\mathbf{x}$ is randomly sampled for a set of example images $\mathcal{X}$, g from a set of synthetic warps $\mathcal{G}$, and the Siamese embedding functions $\Phi(\mathbf{x})$ and $\Phi(g\mathbf{x})$ are compared based on the loss (2) averaged over such samples.

3.2 Controlling the capacity via embedding volume compression

As suggested above, minimizing eq. (2) should result in embeddings that correctly match a certain image $\mathbf{x}$ up to deformations. The authors of [2] found that the learned embeddings do much more: they correctly align different object instances, despite the fact that this is not explicitly enforced in training. This is imputed to the generalization capability of deep networks and, most importantly, to the fact that the embedding space $\mathbb{S}^2$ is very low dimensional which, as discussed above, encourages embeddings for different objects to merge and align.

More specifically, the embedding space used in [2] is $\mathcal{E} = \mathbb{R}^3$, interpreted as a relaxation of the sphere $\mathbb{S}^2$, where the norm of vectors is used to modulate the peakedness and hence the certainty expressed by probability (1). However, other embedding spaces can also be considered. Here we propose to pick $\mathcal{E}$ to be the d-dimensional Euclidean space $\mathbb{R}^d$ where $d \gg 3$ and propose an alternative approach to control capacity.

A possibility, close in spirit to [2], is to encourage empirical samples $\{\Phi_u(\mathbf{x}) : u \in \Omega, \mathbf{x} \in \mathcal{X}\}$ of the embedding to span a low-dimensional subspace of $\mathcal{E}$. However, such an approach may unduly compress information which is relevant for discriminating different object points and may not properly discard irrelevant instance-specific appearance information, which may prevent the embedding from correctly establishing intra-class correspondences. We address these issues by considering an alternative approach, which we call *embedding volume*

compression (EVC), that encourages the *sets* of embedding vectors extracted from two or more images to be the same *on the whole*.

In more detail, let $(\mathbf{x}, \mathbf{x}', g)$ a warped image pair. Furthermore, let $\mathbf{x}_\alpha$ be an *auxiliary image*, containing an object of the same category as the pair $(\mathbf{x}, \mathbf{x}')$, but not necessarily the same instance as $(\mathbf{x}, \mathbf{x}')$. If the embedding function $\Phi_u(\mathbf{x})$ correctly removes irrelevant differences between object instances, then, up to visibility issues, the set of embedding vectors $\{\Phi_u(\mathbf{x}) : u \in \Omega\}$ and $\{\Phi_u(\mathbf{x}_\alpha) : u \in \Omega\}$ should be the same.

Rather than imposing this set equality constraint explicitly as an additional loss term with a corresponding tunable weight, however, we incorporate it as a natural extension of the loss function (2). The idea is to take the embedding vector $\Phi_u(\mathbf{x})$ extracted from image $\mathbf{x}$ and *replace it in loss* (2) *with a reconstruction of it obtained using embeddings from the auxiliary image* $\mathbf{x}_\alpha$.

In order to obtain such a reconstruction, we start by matching pixels in the source image $\mathbf{x}$ to pixels in the auxiliary image $\mathbf{x}_\alpha$ by using the probability $p(v|u; \Phi, \mathbf{x}, \mathbf{x}_\alpha)$ computed according to eq. (1). Then, we reconstruct the source embedding $\Phi_u(\mathbf{x})$ as the weighted average of the embeddings $\Phi_v(\mathbf{x}_\alpha)$ in the auxiliary image, as follows:

$$\widehat{\Phi}_u(\mathbf{x}|\mathbf{x}_\alpha) = \int \Phi_v(\mathbf{x}_\alpha) p(v|u; \Phi, \mathbf{x}, \mathbf{x}_\alpha)\, dv. \tag{3}$$

Once $\widehat{\Phi}_u$ is computed, we use it to establish correspondences between $\mathbf{x}$ and $\mathbf{x}'$, using eq. (1) as before. This results in the matching probability:

$$p(v|u; \Phi, \mathbf{x}, \mathbf{x}', \mathbf{x}_\alpha) = \frac{e^{\langle \widehat{\Phi}_u(\mathbf{x}|\mathbf{x}_\alpha), \Phi_v(\mathbf{x}') \rangle}}{\int_\Omega e^{\langle \widehat{\Phi}_u(\mathbf{x}|\mathbf{x}_\alpha), \Phi_w(\mathbf{x}') \rangle}\, dw}. \tag{4}$$

This matching probability can be used in the same loss function (2) as before, with the only difference that now each sample depends on $\mathbf{x}, \mathbf{x}'$ as well as the auxiliary image $\mathbf{x}_\alpha$.

While this may seem a round-about way of learning correspondences, it has two key benefits: as eq. (2), it encourages embedding vectors to be invariant and distinctive; in addition to eq. (2), it also requires embedding vectors to be compatible between different object instances. In fact, without such a compatibility, the reconstruction (3) would result in a blurry, unmatchable embedding. This has the important effect of controlling the effective capacity of the embedding space, compressing it along irrelevant intra-category variations. Note that the original formulation of [2] lacks the ability to enforce such a constraint directly.

Relation to cycle consistency. Our approach, involving triplets of images, may on the surface seem related to approaches that enforce cycle consistency [36, 37, 23, 24] for image matching. However, the similarity is limited. While cycle consistency is a form of geometric verification that tests the compatibility of a chain of image warps with respect to composition, our method works in the space of appearance descriptors, requiring the latter to be the same for different instances. In particular, we do not test consistency against composition of transformations and, arguably, the two techniques could be combined.

3.3 Relaxation using auxiliary image sets

A potential issue with eq. (4) is that, while image $\mathbf{x}'$ can be obtained from $\mathbf{x}$ by a synthetic warp so that all pixels can be matched, image $\mathbf{x}_\alpha$ is only weakly related to the two. For example, partial occlusions or out of plane rotations may cause some of all the pixels in $\mathbf{x}$ to not have corresponding pixels in $\mathbf{x}_\alpha$.

In order to overcome this challenge, we take inspiration from the recent method of [38] and consider not one, but a small set $\{\mathbf{x}_\alpha : \alpha \in A\}$ of auxiliary image. Then, the summation in eq. (3) is extended not just over spatial locations, but also over images in this set. The intuition is that, as long as at least one image in the auxiliary image set matches $\mathbf{x}$ sufficiently well then the reconstruction will be reliable.

4 Experiments

Using datasets of human faces (section 4.1), animal faces (section 4.2) and a toy robotic arm (section 4.3), we demonstrate the effectiveness of our EVC technique in two ways. First, we show that the learned embeddings work well as visual descriptors, matching reliably different views of an object instance. Second, we show that they *also* identify a dense family of object landmarks, valid not for one, but for all object instances in the same category. Note that, while the first property is in common with traditional and learned descriptor in the spirit of SIFT, the second clearly sets our embeddings apart from these.

Implementation details. In order to allow for a fair comparison with the literature, in all experiments we adopt the same deep neural network architecture of [4]. This architecture contains seven convolutional layers with 20, 48, 64, 80, 256, and C filters, where $C = 3$ for the spherical embedding of [4] and $C = 64$ for our high-dimensional embedding. A dilation factor of 2, 4, 2 is used in the 2nd, 3rd, 4th layers respectively. All but the last convolutional layers are followed by batch normalisation and ReLU. Features are downsampled after the first convolutional layer using a 2×2 max pooling layer with a stride of 2. As a result, if the input size is $H \times W \times 3$, then the output size is $\frac{H}{2} \times \frac{W}{2} \times C$. The weights of the model are initialised with random Gaussian noise and the model is learnt from scratch by using the Adam optimiser [39].

4.1 Human faces

First, we consider two standard benchmark datasets of human faces: CelebA [40] and MAFL [41], which is a subset of the former. The CelebA [40] dataset contains over 200k faces of celebrities; we use the former for training and evaluate on the smaller MAFL [41] (19,000 train images, 1,000 test images) as the latter provides annotations for the eyes, nose and mouth corners. For training, we follow the same procedure used by [2] and exclude any image in the CelebA training set that is also contained in the MAFL test set. Note that we use MAFL annotations only for evaluation and never for training of the embedding function.

Embedding dimension	Same identity		Different identity	
	[2]	+ EVC	[2]	+ EVC
3	1.59	1.47	2.61	2.48
16	1.28	1.26	6.95	2.18
32	1.20	1.23	6.69	2.21

Table 1. Pixel error matching annotated landmarks across 200 pairs of images from CelebA (MAFL test subset).

We use formulation (4) to learn a dense embedding function Φ mapping an image $\mathbf{x}$ to C-dimensional pixel embeddings, as explained above. Note that loss (2) requires sampling transformations $g \in \mathcal{G}$; in order to allow a direct comparison with [2], we use the same random Thin Plate Spline (TPS) warps as they use, obtaining warped pairs $(\mathbf{x}, \mathbf{x}' = g\mathbf{x})$; we also sample at random one or more auxiliary images $\mathbf{x}_\alpha$ from the training set in order to implement EVC.

We consider several cases; in the first, we set $C = 3$ and sample no auxiliary images, using formulation (1), which is the same as [2]. In the second case, we set $C = 16, 64 \gg 3$ but still do not use EVC; in the last case, we use $C = 3, 16, 64$ and also use compression.

Qualitative results. In fig. 1 we compute embeddings from an average face image and sample corresponding patches in at random in the MAFL dataset. All patch placements are geometrically accurate despite large intra-class variations. On the other hand, if EVC is removed, then the quality of the embedding degrades significantly and the latter is confused by intra-class variations. This figure also illustrates that, by having a category-wide validity, our embeddings identify object landmarks rather than extract mere visual descriptors of the local image appearance.

Matching results. Next, we explore the ability of our embeddings to match face images. For this, we sample 200 different identities in the MAFL dataset and consider two cases.

First, we match images $\mathbf{x}, \mathbf{x}'$ of the *same identity*; since multiple images containing the same identity are not provided in MAFL, we synthetically generate them by applying random warps as before, so that the ground-truth correspondence field g is known. We extract embeddings at the annotated keypoint positions from $\mathbf{x}$ and match them one by one to their closest neighbour embedding in image $\mathbf{x}'$ (searching all pixels in the target image). Second, we match images of *different identities*, again using the annotations. In both cases, we report the mean pixel matching error from the ground truth.

By examining the results in table 1 we can note several facts. When matching the same identities, higher dimensional embeddings work better than low dimensional ones, including in particular [2] (top row). This is expected as high dimensional embeddings can easily capture instance-specific details; also as expected, EVC does not change the results much as in this case there are no intra-class variations. When matching different identities, high-dimensional embeddings are rather poor: these descriptors are too sensitive to instance-specific details and

Method	Unsup.	MAFL	AFLW	300W
CFAN [42]	×	15.84	10.94	
Cascaded CNN [43]	×	9.73	8.97	
TCDCN [41]	×	7.95	7.65	5.54
RAR [44]	×		7.23	4.94
Sparse landmarks [4]	✓	6.67/ –	10.53	7.97
Structural Representations [35]	✓	– /3.15†	(6.58)	–
Dense 3D [2]	✓	5.83/ –	8.80	–
Ours 16D	✓	5.37/3.90†	7.55	6.76
Ours 64D	✓	5.13/3.24†	7.33	5.70

Table 2. Landmark detection results on the MAFL and AFLW datasets. The results are reported as percentage of inter-ocular distance. While we are directly comparable to [4], other work uses different dataset variants, making comparison somewhat complicated (see section 4.1). † denotes the type of data augmentation used for learning the regressor, parentheses denote a different test set.

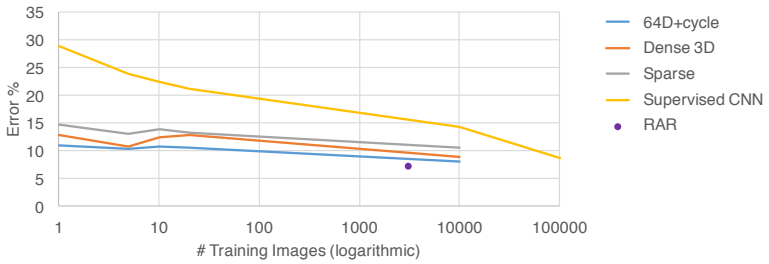


Fig. 4. Presentation of the effect of the number of annotated images used for training different methods, evaluated on AFLW (5-landmark version) incorporating the information in table 3, the Supervised CNN baseline from [4] (supplementary material) and the supervised RAR method [44] trained on 300-W with generated occlusions.

cannot bridge intra-class variations correctly. This justifies the choice of a low dimensional embedding in [2] as the latter clearly generalizes better across instances. However, once EVC is applied, the performance of the high-dimensional embeddings is much improved, and is in fact better than the low-dimensional descriptors even for intra-class matching [2].

Overall, our embeddings have both better intra-class and intra-instance matching performance than [2], validating our hypothesis and demonstrating that our method for regularizing the embedding is preferable to simply constraining the embedding dimensionality.

Landmark regression. Next, as in [2] and other recent papers, we assess quantitatively how well our embeddings correspond to manually-annotated landmarks in faces. For this, we follow the approach of [2] and add on top of our embedding $M = 50$ $1 \times 1 \times C$ filters converting them in the heatmaps of 50 intermediate virtual points; these heatmaps are in turn converted using a softargmax layer to $2C$ x-y pairs which are finally fed to a linear regressor to estimate P

Num. images	Ours 64D	Dense 3D [2]	Sparse [4]
1	11.03	12.79	14.79
5	10.30	10.82	12.94
10	10.72	12.38	13.85
20	10.45	12.79	13.28
10,122	8.04	8.80	10.53

Table 3. Error (% inter-ocular distance) Varying the number of images used for training (AFLW). We compare to the same specific subsets of images as used in [4]. The error decrease is not precisely monotonic with image quantity due to the noisiness of training with such small amounts of data, but the general indication is that most of the information has been encoded in the unsupervised stage. We outperform other unsupervised methods and achieve results close to the best of [4] with just a single image.

manually annotated landmarks. The parameters of the intermediate points and linear regressor are learned using a certain number of manual annotations, but the signal is not back-propagated further so the embeddings remain fully unsupervised.

In detail, after pre-training our 16D and 64D network on the CelebA dataset in a unsupervised manner, we freeze its parameters and only learn the regressors for MAFL [41]. For the more challenging AFLW [45] dataset, and similarly to [2], after training for 50 epochs on CelebA we continue with unsupervised pretraining on 10,122 training images from AFLW for 90 epochs. We also finetune and regress the 68-landmark 300-W dataset [46], with 3158 training and 689 testing images.

We follow the standard evaluation procedure, using the MAFL train and test splits [41] and the $P = 5$ landmark test split of AFLW (see [47]). We report the errors in percentage of inter-ocular distance in table 2 and compare our results on three datasets to state-of-the-art supervised and unsupervised methods. We carefully follow the protocol and data selection used in [2] and related prior work to allow for a direct comparison. We also compare to [35], but note a few differences in their evaluation. For MAFL, this paper reports much better results than [2]; upon investigation, we found that a key difference is in the fact that they do not apply random warp augmentation when learning the landmark regressor (MAFL has little variability, so too much augmentation is harmful — for the other dataset augmentation is better), so we report results for this setting as well, denoted with a dagger † symbol in the table. For AFLW they use a differently cropped variant of the test set, also containing some different individuals, whereas we report results using the images from [47, 2] and others. We also use the same network architecture and data preprocessing of [2] to isolate the impact of our new contribution EVC, whereas [35] use Hourglass, which is significantly larger, and different preprocessing.

We first see that our method outperforms the prior work that either learns sparse landmarks [4] or 3D dense feature descriptors [4], which is consistent with the results in table 1. Additionally, we see that our method achieves comparable results to [35] that use a generative approach and a larger network. Encourag-

Fig. 5. Top: Five landmarks are manually annotated in the top-left image (human) and matched using our unsupervised embedding to a number of animals. Bottom process, but using a cat image (bottom left) as query.

ingly, we also see that our method achieves better or comparable results than the state-of-the-art supervised methods. This indicates that our unsupervised formulation can learn useful information for this task.

We also evaluate how many image annotations our method requires to learn landmark localisation in the AFLW dataset, comparing to [4, 2]. To do so, we vary the number of training images from 1 to the whole training set, 10,122 and report the errors for each setting in table 3 and fig. 4. The results clearly show that our method outperforms [4, 2] and can be used to learn an effective landmark detector with few manual annotations.

4.2 Animal faces

To investigate the generalization capabilities of our method, we consider learning landmarks in an unsupervised manner not just for humans, but for animal faces. To do this, we simply extend the set $\mathcal{X}$ of example image to contain images of animals as well.

In more detail, we consider the Animal Faces dataset [48] with images of 20 animal classes and about 100 images per class. We exclude birds and elephants since these images have a significantly different appearance on average (birds being profile, elephants including the whole body). We then add additional 8609 additional cat faces from [49], 3506 cat and dog faces from [50], and 160k human faces from CelebA (but keep roughly the same distribution of animal classes per batch as the original dataset). We train 3D descriptors using EVC on this data. Here we also found it necessary to use the grouped attention mechanism (section 3.3) which relaxes EVC to project embeddings on a set of auxiliary images rather than just one. In order to do so, we include 16 pairs of images $(\mathbf{x}, \mathbf{x}')$ in each batch and we randomly choose a set of 5 auxiliary images for each pair from a separate pool of 16 images. Note that these images have also undergone synthetic warps.

Results matching human and cat landmarks to other animals are shown in fig. 5. We see that the method achieves localisation of semantically-analogous parts across species, with excellent results particularly for the eyes and general facial region.

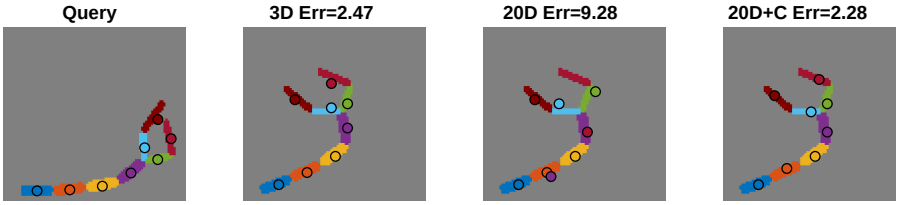


Fig. 6. An example of descriptor matching on a pair from the roboarm dataset, using the blob centers in the first image to locate them in a second image. We show 3D and 20D descriptors (columns 2-3) learned using the loss from [2]. In the 20D case this gives high error, which is corrected by using our EVC (20D+C, last column).

Dimensionality	[2]	+ EVC	- transformations
3	1.42	1.41	1.69
20	10.34	1.25	1.42

Table 4. Results on Roboarm, including an experiment ignoring optical flow (right).

4.3 Roboarm

Lastly, we experimented on the animated robotic arm dataset (fig. 6) introduced in [2] to demonstrate the applicability of the approach to diverse data. This dataset contain around 24,000 images of resolution 90×90 with ground truth optical flow between frames for training. We use the same matching evaluation of section 4.1 using the center of the robot's segments as keypoints for assessing correspondences. We compare models using 3D and 20D embeddings using the formulation of [2], regularizing it using EVC, and finally removing transformation equivariance form the latter (by setting $g = 1$ in eq. (4)).

In this case there are no intra-class variations, but the high-degree of articulation makes matching non-trivial. Since the model is trained using successive frames (so as to obtain optical flow), 3D descriptors work much better than 20D ones (1.43 vs 10.34 error) as they generalize to different rotations properly. With EVC, however, the 20D descriptors (at 1.25 error) outperform the 3D ones (1.41). Interestingly, EVC is effective enough that even removing transformations altogether (by learning from pairs of identical images using $g = 1$) still result in good performance (1.42) – this is possible because matches must hop through the auxiliary image set $\mathbf{x}_\alpha$ which does contain different frames.

5 Conclusions

In this paper, we present a new method that can learn the structure of object instances and categories in *an unsupervised way*. Our key contribution is to formulate this problem in terms of finding correspondences between different instances of same or similar object categories by bridging the gap between two seemingly independent concepts, distinctiveness and invariance. We have shown that relatively high dimensional embeddings can be used to simultaneously match and

align points in similar object instances and categories. We have shown that this method works to find correspondences across different animal species and to predict facial landmarks in the standard computer vision benchmarks.

References

1. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV. (1999)
2. Thewlis, J., Bilen, H., Vedaldi, A.: Unsupervised learning of object frames by dense equivariant image labelling. In: NIPS. (2017)
3. Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: CVPR, IEEE (2015) 4353–4361
4. Thewlis, J., Bilen, H., Vedaldi, A.: Unsupervised learning of object landmarks by factorized spatial embeddings. In: ICCV. (2017)
5. Pritchett, P., Zisserman, A.: Wide baseline stereo matching. In: ICCV, IEEE (1998) 754–760
6. Tuytelaars, T., Van Gool, L., D'haene, L., Koch, R.: Matching of affinely invariant regions for visual servoing. In: ICRA. Volume 2. (1999) 1601–1606
7. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. IJCV **65** (2005) 43–72
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110
9. Tola, E., Lepetit, V., Fua, P.: A fast local descriptor for dense matching. In: CVPR, IEEE (2008) 1–8
10. Lenc, K., Vedaldi, A.: Learning covariant feature detectors. In: ECCV Workshop on Geometry Meets Deep Learning. (2016)
11. Paulin, M., Douze, M., Harchaoui, Z., Mairal, J., Perronin, F., Schmid, C.: Local convolutional features with unsupervised training for image retrieval. In: ICCV. (2015) 91–99
12. Liu, C., Yuen, J., Torralba, A.: SIFT Flow: Dense correspondence across scenes and its applications. PAMI (2011)
13. Hassner, T., Mayzels, V., Zelnik-Manor, L.: On sifts and their scales. In: CVPR, IEEE (2012) 1522–1528
14. Ham, B., Cho, M., Schmid, C., Ponce, J.: Proposal flow. In: CVPR. (2016)
15. Long, J.L., Zhang, N., Darrell, T.: Do convnets learn correspondence? In: Advances in Neural Information Processing Systems. (2014) 1601–1609
16. Learned-Miller, E.G.: Data driven image models through continuous joint alignment. PAMI (2006)
17. Peng, Y., Ganesh, A., Wright, J., Xu, W., Ma, Y.: Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. PAMI **34** (2012)
18. Kemelmacher-Shlizerman, I., Seitz, S.M.: Collection flow. In: CVPR. (2012)
19. Mobahi, H., Liu, C., Freeman, W.T.: A Compositional Model for Low-Dimensional Image Set Representation. CVPR (2014)
20. Novotny, D., Larlus, D., Vedaldi, A.: Learning 3d object categories by looking around them. In: ICCV. (2017)
21. Sundaram, N., Brox, T., Keutzer, K.: Dense point trajectories by gpu-accelerated large displacement optical flow. In: ECCV, Springer (2010) 438–451
22. Zach, C., Klopschitz, M., Pollefeys, M.: Disambiguating visual relations using loop constraints. In: CVPR, IEEE (2010) 1426–1433
23. Zhou, T., Lee, Y.J., Yu, S.X., Efros, A.A.: Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In: CVPR. (2015)
24. Zhou, T., K, P., Aubry, M., Huang, Q., Efros, A.A.: Learning dense correspondence via 3d-guided cycle consistency. In: CVPR. (2016)

25. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models: their training and application. CVIU (1995)
26. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR. (2003)
27. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: ECCV Workshops. (2004)
28. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: CVPR. (2005)
29. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object Detection with Discriminatively Trained Part Based Models. PAMI (2010)
30. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial Transformer Networks. In: NIPS. (2015)
31. Kanazawa, A., Jacobs, D.W., Chandraker, M.: WarpNet: Weakly supervised matching for single-view reconstruction. In: CVPR. (2016)
32. Rocco, I., Arandjelović, R., Sivic, J.: Convolutional neural network architecture for geometric matching. In: CVPR. (2017)
33. Schmidt, T., Newcombe, R., Fox, D.: Self-supervised visual descriptor learning for dense correspondence. IEEE Robotics and Automation Letters **2** (2017) 420–427
34. Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In: CVPR. (2015)
35. Zhang, Y., Guo, Y., Jin, Y., Luo, Y., He, Z., Lee, H.: Unsupervised discovery of object landmarks as structural representations. In: CVPR. (2018)
36. Huang, Q.X., Guibas, L.: Consistent shape maps via semidefinite programming. Eurographics Symposium on Geometry Processing **32** (2013)
37. Zhou, X., Zhu, M., Daniilidis, K.: Multi-image matching via fast alternating minimization. In: CVPR. (2015)
38. Zhuang, B., Liu, L., Li, Y., Shen, C., Reid, I.: Attend in groups: a weakly-supervised deep learning framework for learning from web data. In: CVPR. (2017)
39. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR. (2015)
40. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV. (2015)
41. Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Learning Deep Representation for Face Alignment with Auxiliary Attributes. PAMI (2016)
42. Zhang, J., Shan, S., Kan, M., Chen, X.: Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment. In: ECCV. (2014)
43. Sun, Y., Wang, X., Tang, X.: Deep convolutional network cascade for facial point detection. In: CVPR. (2013)
44. Xiao, S., Feng, J., Xing, J., Lai, H., Yan, S., Kassim, A.: Robust Facial Landmark Detection via Recurrent Attentive-Refinement Networks. In: ECCV. (2016)
45. Koestinger, M., Wohlhart, P., Roth, P.M., Bischof, H.: Annotated facial landmarks in the wild. In: ICCV workshops. (2011)
46. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: The first facial landmark localization challenge. In: CVPR-W. (2013)
47. Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Facial landmark detection by deep multi-task learning. In: ECCV. (2014)
48. Si, Z., Zhu, S.C.: Learning hybrid image templates (hit) by information projection. PAMI (2012)
49. Zhang, W., Sun, J., Tang, X.: Cat head detection - How to effectively exploit shape and texture features. In: ECCV. (2008)
50. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: CVPR. (2012)

Chapter 8

Cross Pixel Optical Flow Similarity for Self-Supervised Learning

This work was accepted for Oral Presentation at the Asian Conference on Computer Vision (ACCV), Perth, 2018

In this paper we consider the task of self-supervision, where a network is trained on a task that does not require manual annotation before being finetuned on some desired task. We note that Optical Flow is a signal that can be easily obtained from video data, yet using it directly for self-supervision is hard, because we perform prediction on a single image but the future motion of objects is inherently ambiguous. Based on this observation, we instead predict an embedding that is consistent in terms of pairwise similarity with the optical flow vectors, and we outperform other motion-based methods on transfer to other tasks.

Contributions of Aravindh Mahendran

1. Prepared large scale version of dataset with 1.6M images.
2. Obtained qualitative and quantitative results. The latter involved transfer learning into a set of benchmarks.
3. Surveyed related work

Contributions of James Thewlis

1. Prepared smaller version of dataset
2. Formulated the cross pixel optical flow similarity loss function
3. Developed the flow normalization method

Cross Pixel Optical-Flow Similarity for Self-Supervised Learning

Aravindh Mahendran[0000−0002−2650−9871], James Thewlis[0000−0001−8410−2570], and Andrea Vedaldi

Visual Geometry Group, University of Oxford
{aravindh,jdt,vedaldi}@robots.ox.ac.uk

Abstract. We propose a novel method for learning convolutional neural image representations without manual supervision. We use motion cues in the form of optical-flow, to supervise representations of static images. The obvious approach of training a network to predict flow from a single image can be needlessly difficult due to intrinsic ambiguities in this prediction task. We instead propose a much simpler learning goal: embed pixels such that the similarity between their embeddings matches that between their optical-flow vectors. At test time, the learned deep network can be used without access to video or flow information and transferred to tasks such as image classification, detection, and segmentation. Our method, which significantly simplifies previous attempts at using motion for self-supervision, achieves state-of-the-art results in self-supervision using motion cues, and is overall state of the art in self-supervised pre-training for semantic image segmentation, as demonstrated on standard benchmarks.

Keywords: Self-Supervised Learning · Motion · Convolutional Neural Network

1 Introduction

Self-supervised learning has emerged as a promising approach to address one of the major shortcomings of deep learning, namely the need for large supervised training datasets. All self-supervised learning methods are based on the same basic premise, which is to identify problems that can be used to train deep networks without the expense of collecting data annotations. In this spirit, an amazing diversity of supervisory signals have been proposed, from image generation to colorization, in-painting, jigsaw puzzle solving, orientation estimation, counting, artifact spotting, and many more (see section 2). Furthermore, the recent work of [9] shows that combining several such cues further helps performance.

In this paper, we consider the case of *self-supervision using motion cues* to learn a convolutional neural network (CNN) for static images. Here, a deep network is trained to predict, from a single video frame, how the image *could change* over time. Since predicted changes can be verified automatically by looking at the actual video stream, this approach can be used for self-supervision. Furthermore, predicting motion may induce a deep network to learn about objects in images. The reason is that objects are a major cause of motion regularity and hence predictability: pixels that belong to the same object are much more likely to "move together" than pixels that do not.

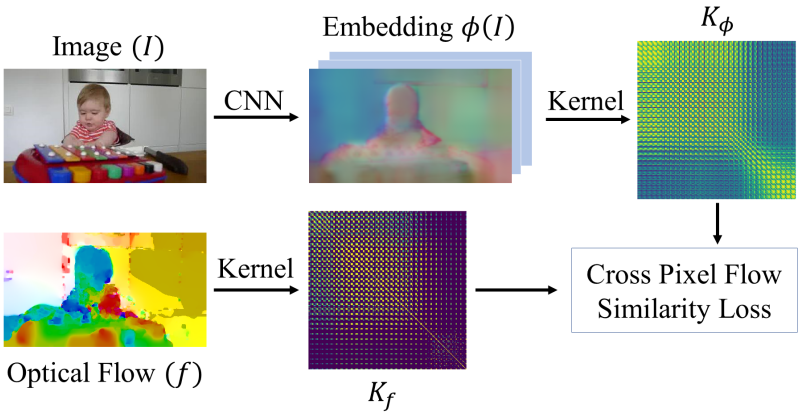


Fig. 1: We propose a novel method to exploit motion information represented as optical-flow, to supervise the learning of deep CNNs. We learn a network that predicts per-pixel embeddings $\phi(I)$ such that the kernel computed over these embeddings (K_ϕ) is similar to that over corresponding optical-flow vectors (K_f). This allows the network to learn from motion cues while avoiding the inherent ambiguity of motion prediction from a single frame.

Besides giving cues about objects, motion has another appealing characteristic compared to other signals for self-supervision. Many other methods are, in fact, based on destroying information in images (e.g. by removing color, scrambling parts) and then tasking a network with undoing such changes. This has the disadvantage of learning the representation on distorted images (e.g. gray scale). On the other hand, extracting a single frame from a video can be thought of as removing information only along the temporal dimension and allows one to learn the network on undistorted images.

There is however a key challenge in using motion for self-supervision: ambiguity. Even if the deep network can correctly identify all objects in an image, this is still not enough to predict the specific direction and intensity of the objects' motion in the video, given just a single frame. This ambiguity makes the direct prediction of the appearance of future frames particularly challenging [54, 59], and overall an overkill if the goal is to learn a general-purpose image representation for image analysis. Instead, the previous most effective method for self-supervision using motion cues [42] is based on first extracting motion tubes from videos (using off-the-shelf optical-flow and motion tube segmentation algorithms) and then training the deep network to predict the resulting per-frame segments rather than motion directly. Thus they map a complex self-supervision task into one of classic foreground-background segmentation.

While the approach of [42] sidesteps the difficult problem of motion prediction ambiguity, it comes at the cost of pre-processing videos using a complex handcrafted motion segmentation pipeline, which includes many heuristics and tunable parameters. In this paper, we instead propose a new method that can ingest cues from optical-flow *directly*, without the need for any complex data pre-processing.

Our method, presented in section 3 and illustrated in fig. 1, is based on a new cross pixel flow similarity loss layer. As noted above, the key challenge is that specific details about the motion, such as its direction and intensity, are usually difficult if not impossible to predict from a single frame. We address this difficulty in two ways. First, we learn to embed pixels into vectors that cluster together when the model believes that the corresponding pixels *are likely to move together*. This is obtained by encouraging the inner product of the learned pixel embeddings to correlate with the similarity between their corresponding optical-flow vectors. This does not require the model to explicitly estimate specific motion directions or velocities. However, this is still not sufficient to address the ambiguity completely; in fact, while different objects may be *able* to move independently, they *may not do so* all the time. For example, often objects stand still, so their velocities are all zero grouping them together in optical-flow. We attempt to address this second challenge by using a contrastive loss.

In section 4 we extensively validate our model against other self-supervised learning approaches. First, we show that our approach works as well or better than [42], establishing a new state-of-the-art method for self-supervision using motion cues. Second, to put this into context, we also compare the results to all recent approaches for self-supervision that use cues other than motion. In this case, we show that our approach has state-of-the-art performance for semantic image segmentation, at the time of submission.

The overall conclusion (section 5) is that our method significantly simplifies leveraging motion cues for self-supervision and does better than existing alternatives for this modality; it is also competitive with self-supervision methods that use other cues, making motion a sensible choice for self-supervision by itself or in combination with other cues [9].

2 Related Work

Self-supervised learning, of which our method is an instance, has become very popular in the community. We discuss here the methods for training generic features for image understanding as opposed to methods with specific goals such as learning object keypoints. We group them according to the supervision cues they use.

Video/Motion Based: LSTM RNNs can be trained to predict future frames in a video [51]. This requires the network to understand image dynamics and extrapolate it into the future. However, since several frames are observed simultaneously, these methods may learn something akin to a tracker, with limited abstraction. On the other hand, we learn to predict properties of optical-flow from a **single input image**, thus learning a static image representation rather than a dynamic one. Closely related to our work is the use of *video segmentation* by [42]. They use an off-the-shelf video segmentation method [15] to construct a foreground-background segmentation dataset in an unsupervised manner. A CNN trained on this proxy task transfers well when fine-tuned for object recognition and detection. We differ from them in that we do not require a sophisticated pre-existing pipeline to extract video segments, but use optical-flow directly. Also closely related to us is the work of [2]. They train a Siamese style convolutional neural network to predict the transformation between two images. The individual base

networks in their Siamese architecture share weights and can be used as feature extractors for single images at test time. This late fusion strategy forces the learning of abstractions, but our **no-fusion approach** pushes the model even further to learn better features. The polar opposite of these is to do early fusion by concatenating two frames as in FlowNet [11]. This was used as a pretraining strategy by [16] to learn representations for **pairs of frames**. This is different from our objective as we aim to learn a **static image representation**. This difference becomes clearer when looking at the evaluation. While we evaluate on image classification, detection, and segmentation; [16] evaluate on dynamic scene and action recognition.

Temporal context is a powerful signal. [35, 57, 32] learn to predict the correct ordering of frames. [24] exploit both temporal and spatial co-occurrence statistics to learn visual groups. [25] extend slow feature analysis using higher order temporal coherence. [55] track patches in a video to supervise their embedding via a triplet loss while [17] do the same but for spatio-temporally matched region proposals. Temporal context is applied in the imitation learning setting by Time Contrastive Networks [49].

Videos contain more than just temporal information. Some methods exploit audio channels by predicting audio from video frames [48, 40]. [3] train a two stream architecture to classify whether an image and sound clip go together or not. Temporal information is coupled with ego-motion in [26]. [56] use videos along with spatial context pretraining [8] to construct an image graph. Transitivity in the graph is exploited to learn representations with suitable invariances.

Colorization: [31, 60] predict colour information given greyscale input and show competitive pre-training performance. [61] generalize to arbitrary pairs of modalities.

Spatial Context: [41] solve the in-painting problem, where a network is tasked with filling-in partially deleted parts of an image. [8] predict the relative position of two patches extracted from an image. In a similar spirit, [37, 38] solve a jigsaw puzzle problem. [38] also cluster features from a pre-trained network to generate pseudo labels, which allows for knowledge distillation from larger networks into smaller ones. The latest iteration on context prediction [36] obtains state-of-the-art on some benchmarks.

Adversarial/Generative: BiGAN based pretrained models [10] show competitive performance on various recognition benchmarks. [27] adversarially learn to generate and spot defects. [45] obtain self-supervision from synthetic data and adapt their model to the domain of real images by training against a discriminator. [5] predict noise-as-targets via an assignment matrix which is optimized on-line. Their approach is domain agnostic. More in general, generative unsupervised layer-wise pretraining was extensively used in deep learning before AlexNet [30]. An extensive review of these and more recent unsupervised generative models is beyond the scope of this paper.

Transformations: [12] create surrogate classes by applying a set of transformations to each image and learn to become invariant to them. [19] do the opposite and try to estimate the transformation (just one of four rotations in their case) given the transformed image. The crop-concatenate transformation is implicit in the learning by counting method of [39].

Others: A combination of self-supervision approaches was explored by [9]. They report results only with ResNet models making it hard to compare with concurrent work, but closely matching ImageNet-pretrained networks in performance on the PASCAL VOC detection task. A widely-applicable trick that helps in transfer learning is the rebalancing method of [29]. Lastly, our optical-flow classification baseline is based on the work of [4]. They learn a sparse hypercolumn model to predict surface normals from a single image and use this as a pretraining strategy. Our baseline flow classification model is the same but with AlexNet for discretized optical-flow.

3 Method

In this section, we describe our novel method, illustrated in fig. 1, for self-training deep neural networks via direct ingestion of optical-flow. Once learned, the resulting image representation could be used for classification, detection, segmentation and other tasks with minimal supervision.

Our goal is to learn the parameters of a neural network that maps a single image or frame to a field of pixel embeddings, one for each pixel. Notation - Let $\Omega \subset \mathbb{R}^2$ be the set of pixels; $I : \Omega \to \mathbb{R}^3$ is an image; Our CNN is the per-pixel mapping $\phi(I, p|\Theta) \in \mathbb{R}^D$ producing D dimensional embeddings. In order to learn this CNN, we require the **similarity** between **pairs** of embedding vectors to align with the similarity between the corresponding flow vectors. This is sufficient to capture the idea that things that move together should be grouped together, popularly known as the **Gestalt principle of common fate** [53].

Formally, given D-dimensional CNN embedding vectors $\phi(I, p|\Theta), \phi(I, q|\Theta) \in \mathbb{R}^D$ for pixels $p, q \in \Omega$ and their corresponding flow vectors $f_p, f_q \in \mathbb{R}^2$, we match the kernel matrices

$$\forall p, q \in \Omega : \quad K_\phi\Big(\phi(I, p|\Theta), \phi(I, q|\Theta)\Big) \approxeq K_f(f_p, f_q) \tag{1}$$

where $K_\phi : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$, $K_f : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ are kernels that measure the similarity of the CNN embeddings and flow vectors, respectively.

In this formulation, in addition to the choice of CNN architecture ϕ, the key design decisions are the choice of kernels K_ϕ, K_f and how to translate constraint eq. (1) into a loss function. The rest of the section discusses these choices.

3.1 Kernels

In order to compare CNN embedding vectors and flow vectors, we choose the (scaled) cosine similarity kernel and the Gaussian/RBF kernel respectively. Using the shorthand notation $\phi_p = \phi(I, p|\Theta)$ for readability, these are:

$$K_\phi(\phi_p, \phi_q) := \frac{1}{4}\frac{\phi_p^T \phi_q}{\|\phi_p\|_2 \|\phi_q\|_2}, \qquad K_f(f_p, f_q) := \exp\left(-\frac{\|f_p - f_q\|_2^2}{2\sigma^2}\right). \tag{2}$$

Note that these kernels, when restricted to the set of pixels Ω, are matrices of size $|\Omega| \times |\Omega|$. Each row or column of this matrix can be thought of as a heatmap capturing

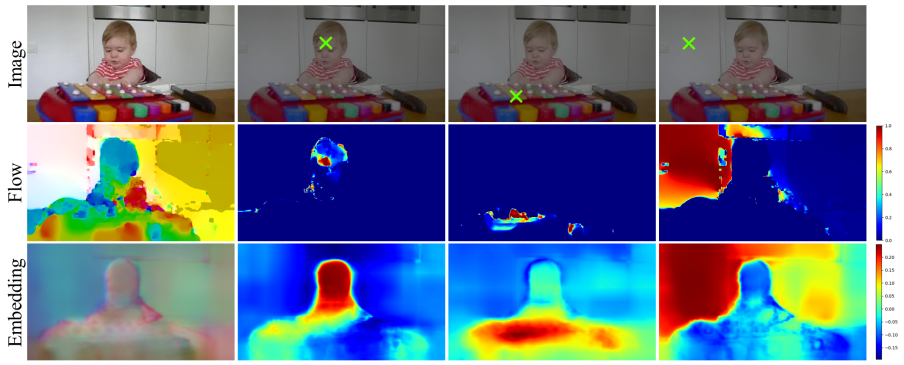


Fig. 2: Visualization of flow and embedding kernels, in the second and third rows respectively. For three pixels p, we plot the row $K_*(p, \cdot)$ reshaped into an image, showing which pixels go together from the kernel's perspective. Note the localized nature of the flow kernel which is obtain by setting a low bandwidth for the RBF kernel. $\sigma^2 = 0.0036$ in this example. In the first column, optical-flow and embeddings (after a random $16D \rightarrow 3D$ projection) are visualized as color images.

the similarity of a given pixel with respect to all other pixels and thus can be visualized as an image. We present such visualizations for both of our kernels in fig. 2.

We use the Gaussian kernel for the flow vectors as this is consistent with the Euclidean interpretation of optical-flow as a displacement. Reducing kernel bandwidth (σ) would result in a localized kernel that pushes our embeddings to distinguish between different movable objects. In some experiments, the value of σ is learned along with the weights of the CNN in the optimization. This localized kernel, with learned $\sigma^2 = 0.0036$, is shown in the second row of fig. 2.

We use the cosine kernel for the learned embedding as the CNN effectively computes a *kernel feature map*, so that in principle it can approximate any kernel via the inner product. However, note that the expression normalizes vectors in L^2 norm, so that this inner product is the cosine of the angle between embedding vectors.

3.2 Cross Pixel Optical-Flow Similarity Loss Function

The constraint in eq. (1) requires kernels K_ϕ and K_f to be similar. We experiment with three loss functions for this task - kernel target alignment, cross-entropy, and cross-entropy reversed.

Kernel Target Alignment (KTA): KTA [7] is a conventional metric to measure the similarity between kernels. KTA for two kernel matrices K, K', is given by

$$\mathcal{L}_{KTA}\left(K, K'\right) = \sum_{pq} K_{pq} K'_{pq} / \sqrt{\sum_{pq} K_{pq}^2 \sum_{pq} {K'_{pq}}^2} \tag{3}$$

Cross-Entropy (CE): Our second loss function treats pixels as classes and kernel values as logits of a distribution over pixels. The cross entropy of these two distributions measures the distance between them. We compute this loss in two steps. First, we re-normalize each column $K_*(\cdot, q)$ of each kernel matrix into a probability distribution $S_*(\cdot, q)$. $S_f(\cdot, q)$ describes which image pixels p are likely to belong to the same segment as pixel q, according to optical-flow. $S_\phi(\cdot, q)$ describes the same but from the CNN embedding's perspective. These distributions, arising from CNN and optical-flow kernels, are compared by using cross entropy, summed over columns:

$$\mathcal{L}_{CE}(\Theta) = -\sum_q \sum_p S_f(p,q) \log S_\phi(p,q). \tag{4}$$

Normalization uses the softmax operator. We reduce the contribution of diagonal terms in the kernel matrix before this normalization because each pixel is trivially similar to itself and would skew the softmax. Formally:

$$S_*(p,q) = \begin{cases} 1/\left(\sum_{q' \neq p} \exp(K_*(p,q')) + 1\right), & \text{if } p = q, \\ \exp(K_*(p,q))/\left(\sum_{q' \neq p} \exp(K_*(p,q')) + 1\right), & \text{if } p \neq q. \end{cases} \tag{5}$$

Cross-Entropy Reversed (CE-rev): Note that the particular ordering of distributions inside the cross entropy loss of eq. (4) treats the distribution induced by the optical-flow kernel (S_f) as ground truth. The embedding is tasked with inducing a kernel such that its corresponding distribution S_ϕ matches S_f. As an ablation study we also experiment with the order of distributions reversed. In other words we use,

$$\mathcal{L}_{CE-rev}(\Theta) = -\sum_q \sum_p S_\phi(p,q) \log S_f(p,q). \tag{6}$$

The intuition here is as follows: For a given pixel p, the distribution $S_\phi(\cdot, q)$ must be a delta distribution around $q' = \operatorname{argmax} S_f(\cdot, q)$. This is the natural effect of a flipped cross entropy loss. This delta distribution can be best approximated by aligning the two embeddings $\phi_p \cong \phi_{q'}$ and making all others anti-correlated $\phi_p \cong -\phi_q \ \forall \ q \neq q'$. Note however that this degenerate solution forces all $\phi_q, \phi_{q''}$ such that $q, q'' \neq q'$ to align as well. This coincidental alignment would, in general, significantly increase the loss function. Thus the embedding is forced to induce a non degenerate distribution S_ϕ. We consider it interesting to experiment with this loss.

Thus we have three cross pixel flow similarity losses - Kernel Target Alignment (CPFS-KTA), Cross Entropy (CPFS-CE) and Cross Entropy reversed (CPFS-CE-rev).

3.3 CNN Embedding Function

Lastly, we discuss the architecture of the CNN function ϕ itself. It maps the image into a per-pixel embedding. Recall that $\forall\ p \in \Omega,\ \phi(I, p|\Theta) \in \mathbb{R}^D$. We design the embedding CNN as a hypercolumn head [22] over a conventional CNN backbone such as AlexNet. The hypercolumn concatenates features from multiple depths so that our embedding can exploit high resolution details normally lost due to max-pooling layers. For training, we

use the sparsification trick of [31] and restrict prediction and loss computation to a few randomly sampled pixels in every iteration. This reduces memory consumption and improves training convergence as pixels in the same image are highly correlated and redundant; via sampling we can reduce this correlation and train more efficiently [4].

In more detail, the backbone is a CNN with activations at several layers: $\{\phi_{c_1}(I|\Theta), \cdots, \phi_{c_n}(I|\Theta)\} \in \mathbb{R}^{H_1 \times W_1 \times D_1} \times \cdots \times \mathbb{R}^{H_n \times W_n \times D_n}$. We follow [31] and interpolate values for a given pixel location and concatenate them to form a hypercolumn $\phi_H(I, p|\Theta) \in \mathbb{R}^{D_1 + \cdots + D_n}$. The hypercolumn is then projected non-linearly to the desired embedding $\phi(I, p|\Theta) \in \mathbb{R}^D$ using a multi-layer perceptron (MLP). Details of the model architecture are discussed in section 4.1.

4 Experiments

We extensively assess our approach by demonstrating its effectiveness in learning features that we show as useful for several tasks. In order to make our results comparable to most of the related papers in the literature, we consider an AlexNet [30] backbone and four tasks: classification in ImageNet [47] and classification, detection, and segmentation in PASCAL VOC [13, 14].

4.1 Backbone Details

We adapt the AlexNet version used by Pathak *et al.* [42]. The modifications are minor (mostly related to padding), to make it suitable to attach a hypercolumn head. Sparse hypercolumns are built from the conv1, pool1, conv3, pool5 and fc7 AlexNet activations. Embeddings are generated using a multi-layer perceptron (MLP) with a single hidden layer and are L2-normalized. The embeddings are $D = 16$ dimensional (this number could be improved via cross validation, although this is expensive). The exact model specification, in the caffe text protocol buffer format (.prototxt), is included in the supplementary material.

4.2 Dataset

We train the above AlexNet model, using various CPFS losses, on a dataset of RGB-optical-flow image pairs. Inspired by the work of Pathak *et al.* [42], we built a dataset from $\sim 204k$ videos in the YFCC100m dataset [52]. The latter consists of Flickr videos made publicly-available under the creative commons license. We extract 8 random frames from each video and compute optical-flow between those at times t and $t + 5$ using the same (handcrafted) optical-flow method of [42, 33]. Overall, we obtain 1.6M image-flow pairs.[1] Example training sample crops along with optical-flow fields are shown in fig. 3. The noisy nature of both the images and optical-flow in such large-scale non-curated video collections makes it all the more challenging for self-supervision.

[1] Optical-flow is stored in fixed point 16bit PNG files similar to KITTI [18] for compression

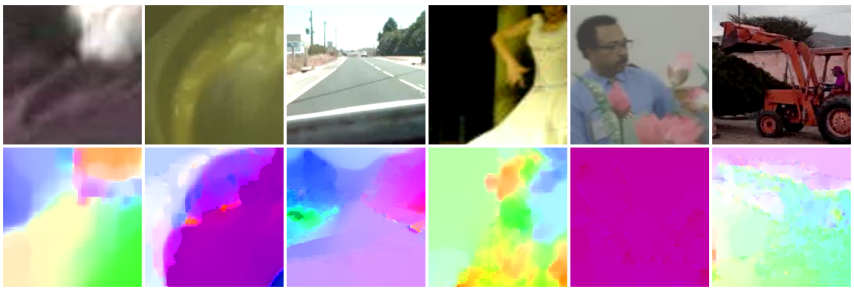


Fig. 3: Image and optical-flow training pairs post scale-crop-flip data augmentation. The noisy nature of both images and optical-flows illustrate the challenges in using motion as a self-supervision signal. Optical-flow is visualized as a colour image using the toolbox of [6].

Optical-flow vectors (f_x, f_y) are normalized logarithmically to lie between $[-1, 1]$ during training, so that occasional large flows do not dominate learning. More precisely, the normalization is given by:

$$f' = \begin{bmatrix} \text{sign}(f_x) \min\left(1, \frac{\log(|f_x|+1)}{\log(M+1)}\right) \\ \text{sign}(f_y) \min\left(1, \frac{\log(|f_y|+1)}{\log(M+1)}\right) \end{bmatrix} \tag{7}$$

where M is a loose upper bound on the flow-magnitude set to 56.0 in our experiments.

Despite the large size of this data and aggressive data augmentation during training, AlexNet overfits on our self-supervision task. We use early stopping to reduce over-fitting by monitoring the loss on a validation set. The validation set consists of 5000 image-flow pairs computed from the YouTube objects dataset [43]. Epic-Flow [46], with initial matches from Deep-Matching [58], was used to compute optical-flow for these frames.

4.3 Learning Details

We use the Adam optimizer [28] with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$ and initial learning rate 10^{-4}. No weight decay is used because it resulted in our model reaching a worse local minima before over-fitting started. Pixels are sampled uniformly at random for the sparse hypercolumns. Sampling more pixels gives more information per image but also consumes more memory and is computationally expensive. We use 512 pixels per image to balance this trade-off. This allows for a large batch size of 96 frames. Scale, horizontal flip and crop augmentation with crop size 224×224 are applied during training. Color augmentation: shifting the hue by up to 0.1, random contrast between $0.2 - 1.8$, random brightness by up to 0.12 (based on $0 - 1$ normalised colours); is also applied. Parameter-free batch-normalization [23] is used throughout the network; the moving average mean and variance are absorbed into convolution kernels after self-supervised training, so that, for evaluation, AlexNet does not contain batch normalization layers. The implementation using TensorFlow [1] will be publicly available on GitHub.

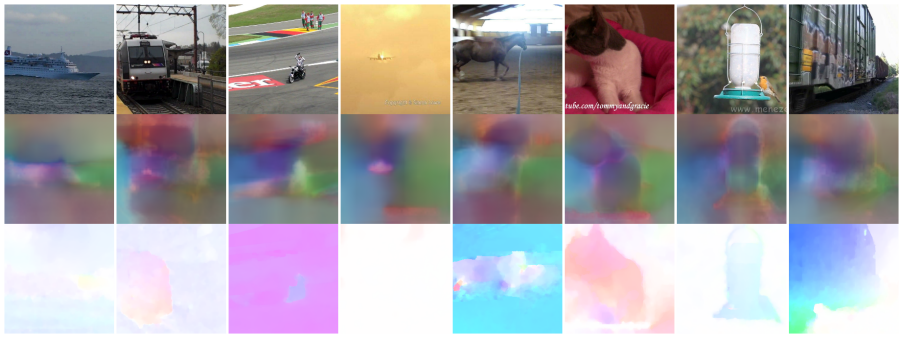


Fig. 4: Per-pixel embeddings are visualized by randomly projecting them to RGB colors. From top to bottom: Validation images, RGB-mapped embeddings, optical-flow.

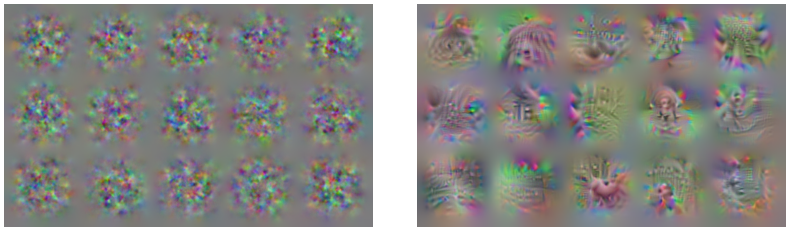


Fig. 5: Neuron maximization results for conv5 features [34]. Left: Neurons in a randomly initialized AlexNet. Right: Neurons in AlexNet trained using our approach: significantly more structure emerges.

4.4 Qualitative Results

In this section we visualize the AlexNet model pre-trained using the CPFS-CE loss function (Equation (4)).

Embedding Visualizations: While our learned pixel embeddings are not meant to be used directly (instead their purpose is to pre-train a neural network parametrization that can be transferred to other tasks), nevertheless it is informative to visualize them. Since embeddings are 16D, we first project them to 3D vectors via random projections and then map the resulting coordinates to RGB space. We show results on the YouTube objects validation set in fig. 4. Note that pixels on a salient foreground object tend to cluster together in the embedding (see, for example, the aircraft in column 4, the motor cyclist in column 3 and the cat in column 6).

Neuron Maximization: We use per-neuron activation maximization [34] to visualize individual neurons in the fifth convolutional layer (fig. 5). This figure presents the estimated optimal stimulus for each of these neurons, made interpretable using a natural image prior. We observe abstract patterns including a human form (row 2, column 4)

Table 1: Pascal VOC Comparison for three benchmarks: VOC2007-classification (column 4) %mAP, VOC2007-Detection (Column 5) %mAP and VOC2012-Segmentation (Column 6) %mIoU. The rows are grouped into four blocks (1) The limits of no-supervision and human supervision, (2) motion/video based self-supervision, (3) Our models and the baseline, (4) others. The third column [ref] indicates which publication the reported numbers are borrowed from. Full table in supplementary material.

	Method	Supervision	[Ref]	Cls.	Detection	Seg.
	Krizhevsky *et al.* [30]	Class Labels	[60]	79.9	56.8	48.0
	Random	-	[41]	53.3	43.4	19.8
Motion cues	Agrawal *et al.* [2]	Egomotion	[10]	63.1	43.9	-
	Jayaraman *et al.* [26]	Egomotion	[26]	-	41.7	-
	Lee *et al.* [32]	Time-order	[32]	63.8	46.9	-
	Misra *et al.* [35]	Time-order	[35]	-	42.4	-
	Pathak *et al.* [42]	Video-seg	[42],Self	61.0	50.2	-
	Wang *et al.* [55]	Track + Rank	[29, 55]	63.1	47.5	-
	CPFS-CE	Optical-flow	Self	64.2	**50.8**	41.4
	CPFS-CE-rev	Optical-flow	Self	63.6	49.9	39.5
	CPFS-KTA	Optical-flow	Self	**65.3**	50.5	**41.5**
	Ours direct cls.	Optical-flow	Self	63.2	46.1	38.8
	Other Cues - State of the art	Varied	[19, 38, 36]	73.3 [19]	55.5 [38]	40.6 [36]

that are obviously not present in a random network, suggesting that the representation may be learning concepts useful for general-purpose image analysis.

4.5 Quantitative Results

We follow standard practice in the self-supervised learning community and fine-tune the learned representation on various recognition benchmarks. We evaluate our features, pre-trained using various CPFS losses, by transfer learning on PASCAL VOC 2007 detection and classification [13], PASCAL VOC 2012 segmentation [14], and ILSVRC12 linear probing [60] (in the latter case, the representation is frozen). We provide details on the evaluation protocol next and compare against other self-supervised models with results reported for AlexNet-like-architectures in tables 1 and 2. Different from other approaches, we did not benefit from the re-balancing trick of [29] and hence we report results without it. This is possibly due to the use of batch normalization layers.

Baseline: Our main hypothesis is that the cross pixel flow similarity matching method, rather than the direct prediction of optical-flow, is more appropriate for exploiting optical-flow as a self-supervisory signal. We validate this hypothesis by comparing against a direct optical-flow prediction baseline, using the same CNN architecture as our method but a different loss function: while we use flow-similarity matching losses,

Table 2: ImageNet LSVRC-12 linear probing evaluation. A linear classifier is trained on the (downsampled) activations of each layer in the pretrained model. Top-1 accuracy is reported on ILSVRC-12 validation set. The column [ref] indicates which publication the reported numbers are borrowed from. We finetune Pathak *et al.*'s [42] model along with ours as they do not report these benchmark in their paper.

	Method	Supervision	[ref]	Conv1	Conv2	Conv3	Conv4	Conv5
	Krizhevsky *et al.* [30]	Class Labels	[61]	19.3	36.3	44.2	48.3	50.5
	Random	-	[61]	11.6	17.1	16.9	16.3	14.1
	Random-rescaled [29]	-	[29]	17.5	23.0	24.5	23.2	20.6
Motion	Pathak *et al.* [42]	Video-seg	Self	**15.8**	23.2	29.0	29.5	25.4
Motion	CPFS-CE	Optical-Flow	Self	14.9	**25.0**	**29.5**	**30.1**	**29.1**
Motion	CPFS-CE-rev	Optical-Flow	Self	15.3	24.8	27.7	27.8	26.3
Motion	CPFS-KTA	Optical-Flow	Self	14.8	24.6	29.2	29.5	28.1
Motion	Ours direct cls.	Optical-Flow	Self	14.0	23.0	26.4	26.7	24.8
Other cues	Doersch *et al.* [8]	Context	[61]	16.2	23.3	30.2	31.7	29.6
Other cues	Gidaris *et al.* [19]	Rotation	[19]	18.8	31.7	**38.7**	38.2	**36.5**
Other cues	Jenni *et al.* [27]	-	[27]	19.5	**33.3**	37.9	**38.9**	34.9
Other cues	Mundhenk *et al.* [36]	Context	[36]	**19.6**	31.4	37.0	37.8	33.3
Other cues	Noroozi *et al.* [37]	Jigsaw	[39]	18.2	28.8	34.0	33.9	27.1
Other cues	Noroozi *et al.* [39]	Counting	[39]	18.0	30.6	34.3	32.5	25.7
Other cues	Noroozi *et al.* [38]	Jigsaw++	[38]	18.2	28.7	34.1	33.2	28.0
Other cues	Noroozi *et al.* [38]	CC+Jigsaw++	[38]	18.9	30.5	35.7	35.4	32.2
Other cues	Pathak *et al.* [41]	In-Painting	[61]	14.1	20.7	21.0	19.8	15.5
Other cues	Zhang *et al.* [60]	Colorization	[61]	13.1	24.8	31.0	32.6	31.8
Other cues	Zhang *et al.* [61]	Split-Brain	[61]	17.7	29.3	35.4	35.2	32.8

this baseline does a standard per-pixel softmax cross entropy across 16 discrete optical-flow classes, once for each spatial dimension — x and y. To this end, since the flow is normalized in $[-1, 1]$ (eq. (7)), this interval is discretized uniformly. Note that direct L^2 regression of flow vectors is also possible, but did not work as well in preliminary experiments. This may be because continuous regression is usually harder for deep networks compared to classification, especially for ambiguous tasks. It was beneficial to use a faster initial learning rate of 0.01 for this baseline model.

VOC2007-detection: We finetune our AlexNet backbone end-to-end using the Fast-RCNN model [20] using code from [44] to obtain results for PASCAL VOC 2007 detection [13]. Finetuning follows the protocol of [29] to use multi-scale training and single-scale testing. We report mean average precision (mAP) in table 1 (col. 5) along with results of other self-supervised learning methods. We achieve the state-of-the-art among methods that use temporal information in videos for self-supervision. This table summarizes the state of the art among methods that use cues other than motion. Please see the supplementary material for a complete table of all relevant methods.

VOC2007 classification: We finetune our pretrained AlexNets to minimize the softmax cross-entropy loss over the PASCAL VOC 2007 *trainval* set for image classification across 20 Pascal classes. The initial learning rate is 10^{-3} and drops by a factor of 2 every 10k iterations for a total of 80k iterations and predictions are averaged over 10 random crops at test time in keeping with [29]. We use the code provided by [31] and report mean average precision (mAP) on VOC2007-test in the fourth column of table 1. We achieve state-of-the-art among methods that derive self-supervision from motion cues; in particular, we outperform [42] by a large margin.

ILSVRC12 linear probing: We follow the protocol and code of [61] to train a linear classifier on activations of our pre-trained network. The activation tensors produced by various convolutional layers (after ReLU) are down-sampled using bilinear interpolation to have roughly 9,000-10,000 elements before being fed into a linear classifier. The CNN parameters are frozen and only the linear classifier weights are trained on the ILSVRC-12 training set. Top-1 classification accuracy is reported on the ILSVRC-12 validation set (table 2). We achieve the state-of-the-art among motion-based self-supervision methods, except for "conv1" features.

VOC2012 segmentation: We use the two staged fine-tuning approach of [31] who finetune their AlexNet for semantic segmentation using a sparse hypercolumn head instead of the conventional FCN-32s head. We do so because it is a better fit for our sparse hypercolumn pre-training, although the hypercolumn itself is built using different layers (conv1 to conv5 and fc6 to fc7). Thus the MLP predicting the embedding ϕ from hypercolumn features is replaced with a new one before fine-tuning for segmentation. Also, our model has a fully convolutional structure but is pre-trained on a non-convolutional proxy task. We obtain a mere 31.3 %mIoU using FCN-32s . The training data consists of the PASCAL VOC 2012 [14] training set augmented with additional annotations from [21]. Thus the training-validation split has 10582 training images and 1449 validation images. Test results are reported as mean intersection-over-union (mIoU) scores on the PASCAL VOC 2012 validation set (Column 6 of table 1). We achieve the state of the art on this benchmark among all self-supervised learning methods, even ones that use other supervisory signals than motion (at the time of submission).

Other Architectures: We experimented with a VGG-16 [50] backbone[2] and followed the protocol of [31] to evaluated on VOC2007-classification and VOC12-segmentation. Our CPFS-CE model achieved 76.4% mAP for VOC2007 classification comparable to Larsson *et al.*'s 77.2% [31]; and 51.7% mIoU for VOC2012-segmentation which fell short of [31]'s 56.0%. VGG-16 has many more parameters than AlexNet. We argue that it may benefit from the extra 2.1M images used by [31] which might explain this performance gap.

4.6 Discussion

We can take home several messages from these experiments. First, in all cases our approach outperforms the baseline of predicting optical-flow directly. This is true for all

[2] Full model: 'pool 1-5', and 'fc7' (projected to 256 channels using a 1×1 convolution for faster training) constitute the hypercolumn head for pre-training on the dataset (Section 4.2).

three cross pixel flow similarity loss functions. This supports our hypothesis that direct single-frame optical-flow prediction is either too difficult due to its intrinsic ambiguity or a distraction from the goal of learning a powerful representation. It also supports our claim that predicting pairwise flow similarities partially addresses this ambiguity and allows us to learn useful CNN representations from optical-flow.

Second, the cross entropy loss is comparable in performance to kernel target alignment (KTA). We know that KTA aligns kernels uniformly and doing so is still highly ambiguous. Thus there is more room for improvement in the loss function design. Also, surprisingly, reversed cross entropy performs well although not as well as the other two.

Third, our method is the state of the art for self-supervision using motion cues. This is notable as our approach is significantly simpler than the previous state of the art [42]. By ingesting optical-flow information directly, it does not require data pre-processing via a video segmentation algorithm.

Finally, we also observe that all video/motion based methods for self-supervised learning are generally not as effective as methods that use other cues; particularly in image classification benchmarks. However, our approach still sets the overall state of the art for semantic image segmentation suggesting that the learned representation may be more suitable for per-pixel applications. Therefore further progress in this area of motion based self-supervision may be possible and is worth seeking. At the same time, authors of [9] find that combinations of different cues may result in the best performance.

5 Conclusion

We have presented a novel method for self-supervision using motion cues based on cross-pixel optical-flow similarity loss functions. We trained an AlexNet model using this scheme on a large unannotated video data-set. Visualizations of individual neurons in a deep layer and of the output embedding show that the representation captures structure in the image. We established the effectiveness of the resulting representation by transfer learning for several recognition benchmarks. Compared to the previous state of the art motion based method [42], our method works just as well and in some cases noticeably better despite a significant algorithmic simplification. We also outperform all other self-supervision strategies in semantic image segmentation (VOC12). This is reasonable as we train on a per-pixel proxy task on undistorted RGB images and use a hypercolumn model for fine-tuning. Finally, we see our contribution as an instance of self-supervision using multiple modalities, RGB and optical-flow, which poses our work as a special case of this broader area of research.

Acknowledgements

The authors gratefully acknowledge ERC IDIU, AIMS CDT (EPSRC EP/L015897/1) and AWS Cloud Credits for Research program. The authors thank Ankush Gupta and David Novotný for helpful discussions, and Christian Rupprecht, Fatma Guney and Ruth Fong for proof reading the paper. We thank Deepak Pathak for help with reproducing some of the results from [42].

References

1. Abadi, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
2. Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: ICCV (2015)
3. Arandjelović, R., Zisserman, A.: Look, listen and learn. In: ICCV (2017)
4. Bansal, A., Chen, X., Russell, B., Gupta, A., Ramanan, D.: Pixelnet: Representation of the pixels, by the pixels, and for the pixels. arXiv:1702.06506 (2017)
5. Bojanowski, P., Joulin, A.: Unsupervised learning by predicting noise. In: ICML (2017)
6. Butler, D.J., et al.: A naturalistic open source movie for optical flow evaluation. In: ECCV (2014)
7. Cristianini, N., et al.: An Introduction to Support Vector Machines. Cambridge: CUP (2000)
8. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV (2015)
9. Doersch, C., et al.: Multi-task self-supervised visual learning. In: ICCV (2017)
10. Donahue, J., et al.: Adversarial feature learning. ICLR (2017)
11. Dosovitskiy, A., et al.: Flownet: Learning optical flow with convolutional networks. In: ICCV (2015)
12. Dosovitskiy, A., et al.: Discriminative unsupervised feature learning with exemplar convolutional neural networks. IEEE PAMI **38**(9), 1734–1747 (Sept 2016)
13. Everingham, M., et al.: The PASCAL Visual Object Classes Challenge 2007 Results (2007)
14. Everingham, M., et al.: The PASCAL Visual Object Classes Challenge 2012 Results (2012)
15. Faktor, A., Irani, M.: Video segmentation by non-local consensus voting. In: BMVC (2014)
16. Gan, C., Gong, B., Liu, K., Su, H., Guibas, L.J.: Geometry guided convolutional neural networks for self-supervised video representation learning. In: CVPR (2018)
17. Gao, R., Jayaraman, D., Grauman, K.: Object-centric representation learning from unlabeled videos. In: Proc. ACCV (2016)
18. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: CVPR (2012)
19. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised Representation Learning by Predicting Image Rotations. In: Proc. ICLR (2018)
20. Girshick, R.B.: Fast R-CNN. In: ICCV (2015)
21. Hariharan, B., et al.: Semantic contours from inverse detectors. In: ICCV (2011)
22. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: CVPR. pp. 447–456 (2015)
23. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
24. Isola, P., Zoran, D., Krishnan, D., Adelson, E.H.: Learning visual groups from co-occurrences in space and time. ICLR Workshop (2015)
25. Jayaraman, D., Grauman, K.: Slow and steady feature analysis: higher order temporal coherence in video. In: CVPR. pp. 3852–3861 (2016)
26. Jayaraman, D., et al.: Learning image representations tied to ego-motion. In: ICCV (2015)
27. Jenni, S., Favaro, P.: Self-supervised feature learning by learning to spot artifacts. In: CVPR (2018)
28. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
29. Krähenbühl, P., et al.: Data-dependent initializations of convolutional neural networks. ICLR (2016)
30. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS. pp. 1106–1114 (2012)

31. Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding. In: CVPR (2017)
32. Lee, H.Y., Huang, J.B., Singh, M.K., Yang, M.H.: Unsupervised representation learning by sorting sequence. In: ICCV (2017)
33. Liu, C.: Beyond Pixels: Exploring New Representations and Applications for Motion Analysis. Ph.D. thesis, Massachusetts Institute of Technology, USA (2009)
34. Mahendran, A., Vedaldi, A.: Visualizing deep convolutional neural networks using natural pre-images. IJCV pp. 1–23 (2016)
35. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and Learn: Unsupervised Learning using Temporal Order Verification. In: ECCV (2016)
36. Mundhenk, T., Ho, D., Y. Chen, B.: Improvements to context based self-supervised learning. In: CVPR (2017)
37. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV. pp. 69–84. Springer (2016)
38. Noroozi, M., Vinjimoor, A., Favaro, P., Pirsiavash, H.: Boosting self-supervised learning via knowledge transfer. In: CVPR (2018)
39. Noroozi, M., et al.: Representation learning by learning to count. In: ICCV (2017)
40. Owens, A., et al.: Ambient sound provides supervision for visual learning. In: ECCV (2016)
41. Pathak, D., et al.: Context encoders: Feature learning by inpainting. In: CVPR (2016)
42. Pathak, D., et al.: Learning features by watching objects move. In: CVPR (2017)
43. Prest, A., et al.: Learning object class detectors from weakly annotated video. In: CVPR (2012)
44. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS (2015)
45. Ren, Z., Lee, Y.J.: Cross-domain self-supervised multi-task feature learning using synthetic imagery. In: CVPR (2018)
46. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Epicflow: Edge-preserving interpolation of correspondences for optical flow. In: CVPR (2015)
47. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. IJCV (2015)
48. de Sa, V.R.: Learning classification with unlabeled data. In: NIPS. pp. 112–119 (1994)
49. Sermanet, P., et al.: Time-contrastive networks: Self-supervised learning from video. In: Proc. Intl. Conf. on Robotics and Automation (2018)
50. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Arxiv 1409.1556 (2014)
51. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: ICML (2015)
52. Thomee, B., et al.: Yfcc100m: the new data in multimedia research. ACM (2016)
53. Todorovic, D.: Gestalt principles. Scholarpedia **3**(12), 5345 (2008), revision #91314
54. Walker, J.: Data-Driven Visual Forecasting. Ph.D. thesis, Carnegie Mellon University (2018)
55. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: ICCV. pp. 2794–2802 (2015)
56. Wang, X., He, K., Gupta, A.: Transitive invariance for self-supervised visual representation learning. In: ICCV. pp. 2794–2802 (2017)
57. Wei, D., et al.: Learning and using the arrow of time. In: CVPR. pp. 8052–8060 (2018)
58. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: DeepFlow: Large displacement optical flow with deep matching. In: ICCV. pp. 1385–1392 (2013)
59. Xue, T., Wu, J., Bouman, K.L., Freeman, W.T.: Visual dynamics: Stochastic future generation via layered cross convolutional networks. In: IEEE PAMI (2018)
60. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016)
61. Zhang, R., Isola, P., Efros, A.A.: Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In: CVPR (2017)

References

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proc. ICCV*, 2015.

[2] Yali Amit, Ulf Grenander, and Mauro Piccioni. Structural image restoration through deformable templates. *Journal of the American Statistical Association*, 86(414):376–387, 1991.

[3] Christian Bailer, Bertram Taetz, and Didier Stricker. Flow Fields: Dense Correspondence Fields for Highly Accurate Large Displacement Optical Flow Estimation. *ICCV*, 2015.

[4] Simon Baker, Iain Matthews, and Jeff Schneider. Automatic construction of active appearance models as an image coding problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10), 2004.

[5] Yoshua Bengio. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2009.

[6] Michael J. Black and P Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 1996.

[7] Michael J. Black, Yaser Yacoob, Allan D. Jepson, and David J. Fleet. Learning parameterized models of image motion. *CVPR*, 1997.

[8] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on pattern analysis and machine intelligence*, 25(9):1063–1074, 2003.

[9] Fred L. Bookstein. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *PAMI*, 1989.

[10] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting people using mutually consistent poselet activations. *Proc. ECCV*, 2010.

[11] H Bourlard and Y Kamp. Auto-Association by Multilayer Perceptrons and Singular Value Decomposition. *Biological Cybernetics*, 1988.

[12] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. *ECCV*, 2004.

[13] Thomas Brox and Jitendra Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[14] Michael C. Burl, Markus Weber, and Pietro Perona. A probabilistic approach to object recognition using local photometry and global geometry. *ECCV*, 1998.

[15] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face Alignment by Explicit Shape Regression. *IJCV*, 2014.

[16] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 40(4):834–848, 2018.

[17] T F Cootes, C J Taylor, D H Cooper, and J Graham. Active shape models: their training and application. *CVIU*, 1995.

[18] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active Appearance Models. *Proc. ICCV*, 1998.

[19] Timothy F Cootes, Stephen Marsland, Carole J Twining, Kate Smith, and Christopher J Taylor. Groupwise diffeomorphic non-rigid registration for automatic model building. In *European conference on computer vision*. Springer, 2004.

[20] David Cristinacce and Tim Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 2008.

[21] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005.

[22] Matthias Dantone, Juergen Gall, Gabriele Fanelli, and Luc Van Gool. Real-time facial feature detection using conditional regression forests. In *Proc. CVPR*, 2012.

[23] Luca Del Pero, Susanna Ricco, Rahul Sukthankar, and Vittorio Ferrari. Behavior Discovery and Alignment of Articulated Object Classes from Unstructured Video. *arXiv*, 2015.

[24] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised Visual Representation Learning by Context Prediction. In *Proc. ICCV*, 2015.

[25] Piotr Dollár, Peter Welinder, and Pietro Perona. Cascaded pose regression. In *Proc. CVPR*, 2010.

[26] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *Proc. ICLR*, 2017.

[27] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. *Proc. ICLR*, 2017.

[28] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[29] Rob Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. *CVPR*, 2003.

[30] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proc. CVPR*, 2017.

[31] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazrba, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning Optical Flow with Convolutional Networks. *ICCV*, 2015.

[32] Martin A Fischler and Robert A Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on computers*, (1):67–92, 1973.

[33] Brendan J. Frey and Nebojsa Jojic. Transformation-invariant clustering using the em algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1), 2003.

[34] Ravi Garg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Proc. ECCV*, pages 740–756, 2016.

[35] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 2014.

[36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[37] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. *arXiv preprint arXiv:1802.00434*, 2018.

[38] Riza Alp Güler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. In *Proc. CVPR*, volume 2, page 5, 2017.

[39] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous Detection and Segmentation. In *ECCV*. 2014.

[40] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244, 1988.

[41] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. ICCV*, pages 2980–2988. IEEE, 2017.

[42] G E Hinton and R R Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 2006.

[43] Geoffrey E. Hinton. A Parallel Computation that Assigns Canonical Object-Based Frames of Reference. *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[44] Derek Hoiem, Carsten Rother, and John Winn. 3D LayoutCRF for multi-view object class recognition and segmentation. In *CVPR*, 2007.

[45] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 1981.

[46] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial Transformer Networks. *Advances in Neural Information Processing Systems*, 2015.

[47] Anil K Jain, Yu Zhong, and Marie-Pierre Dubuisson-Jolly. Deformable template models: A review. *Signal processing*, 71(2):109–129, 1998.

[48] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Conditional image generation for learning the structure of visual objects. *Advances in Neural Information Processing Systems*, 2018.

[49] Angjoo Kanazawa, David W. Jacobs, and Manmohan Chandraker. WarpNet: Weakly Supervised Matching for Single-view Reconstruction. *CVPR*, 2016.

[50] Ira Kemelmacher-Shlizerman and Steven M. Seitz. Collection flow. *CVPR*, 2012.

[51] Iasonas Kokkinos and Alan Yuille. Unsupervised learning of object deformation models. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[52] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 2012.

[53] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *Proc. ECCV*, 2016.

[54] Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S Corrado, Jeff Dean, and Andrew Y. Ng. Building high-level features using large scale unsupervised learning. *ICML*, 2012.

[55] Erik G Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[56] Yan LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 1998.

[57] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *ECCV Workshop on Geometry Meets Deep Learning*, 2016.

[58] Ce Liu, Jenny Yuen, and Antonio Torralba. SIFT Flow: Dense correspondence across scenes and its applications. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2011.

[59] Jiongxin Liu and Peter N. Belhumeur. Bird part localization using exemplar-based models with enforced pose and subcategory consistency. *Proc. ICCV*, 2013.

[60] Yanxi Liu, Hagit Hel-Or, Craig S Kaplan, Luc Van Gool, et al. Computational symmetry in computer vision and computer graphics. *Foundations and Trends® in Computer Graphics and Vision*, 5(1–2):1–195, 2010.

[61] David G Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, volume 2, pages 1150–1157, 1999.

[62] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI*, 1981.

[63] Aravindh Mahendran, James Thewlis, and Andrea Vedaldi. Cross pixel optical flow similarity for self-supervised learning. In *Proc. ACCV*, 2018.

[64] Aravindh Mahendran, James Thewlis, and Andrea Vedaldi. Self-supervised segmentation by grouping optical-flow. In *POCV 2018 Workshop on Action, Perception and Organization, ECCV Workshops*, 2018.

[65] Giovanni Marola. On the detection of the axes of symmetry of symmetric and almost symmetric planar images. *PAMI*, 11(1):104–108, 1989.

[66] I. Matthews and S. Baker. Active Appearance Models Revisited. *IJCV*, 2004.

[67] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *Proc. ECCV*, 2016.

[68] Hossein Mobahi, Ce Liu, and William T. Freeman. A Compositional Model for Low-Dimensional Image Set Representation. *CVPR*, 2014.

[69] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. ECCV*, 2016.

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

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

[72] Anestis Papazoglou, Luca Del Pero, and Vittorio Ferrari. Discovering object aspects from video. *Image and Vision Computing*, 2016.

[73] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning Features by Watching Objects Move. In *Proc. CVPR*, 2017.

[74] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context Encoders: Feature Learning by Inpainting. In *Proc. CVPR*, 2016.

[75] Marco Pedersoli, Tinne Tuytelaars, and Luc Van Gool. Using a deformation field model for localizing faces and facial points under weak supervision. In *Proc. CVPR*, 2014.

[76] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. *CVPR*, 2012.

[77] Dan Raviv, Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Full and partial symmetries of non-rigid shapes. *IJCV*, 89(1):18–39, 2010.

[78] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face alignment at 3000 FPS via regressing local binary features. In *Proc. CVPR*, 2014.

[79] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. DeepMatching: Hierarchical deformable dense matching. *IJCV*, 2015.

[80] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. *CVPR*, 2015.

[81] Aswin C. Sankaranarayanan, Chinmay Hegde, Sriram Nagaraj, and Richard G. Baraniuk. Go with the flow: Optical flow-based transport operators for image manifolds. *Allerton Conference on Communication, Control, and Computing*, 2011.

[82] Scott Satkin, Maheen Rashid, Jason Lin, and Martial Hebert. 3DNN: 3D Nearest Neighbor. *IJCV*, 2015.

[83] Kevin J Shih, Arun Mallya, Saurabh Singh, and Derek Hoiem. Part Localization using Multi-Proposal Consensus for Fine-Grained Categorization. In *Proc. BMVC*, 2015.

[84] Jamie Shotton, Toby Sharp, and A Kipman. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 2013.

[85] Zhixin Shu, Mihir Sahasrabudhe, Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *Proc. ECCV*, 2018.

[86] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[87] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *Proc. CVPR*, 2013.

[88] Supasorn Suwajanakorn, Noah Snavely, Jonathan Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. *arXiv preprint arXiv:1807.03146*, 2018.

[89] J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *Advances in Neural Information Processing Systems*, 2017.

[90] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proc. ICCV*, 2017.

[91] Darcy Wentworth Thompson. On growth and form. *On growth and form.*, 1942.

[92] Carlo Tomasi and T Kanade. Detection and tracking of point features. *Technical Report CMU-CS-91-132*, 1991.

[93] Alexander Toshev and Christian Szegedy. DeepPose: Human pose estimation via deep neural networks. *Proc. CVPR*, 2014.

[94] Shubham Tulsiani, Abhishek Kar, Joao Carreira, and Jitendra Malik. Learning Category-Specific Deformable 3D Models for Object Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[95] Matthew A. Turk and Alex P. Pentland. Face Recognition Using Eigenfaces. *CVPR*, 1991.

[96] Michel Valstar, Brais Martinez, Xavier Binefa, and Maja Pantic. Facial point detection using boosted regression and graph models. In *Proc. CVPR*, 2010.

[97] Andrea Vedaldi and Stefano Soatto. A complexity-distortion approach to joint pattern alignment. In *Advances in Neural Information Processing Systems*, 2007.

[98] Xiaolong Wang and Abhinav Gupta. Unsupervised Learning of Visual Representations Using Videos. *ICCV*, 2015.

[99] Xiaolong Wang and Abhinav Gupta. Unsupervised Learning of Visual Representations Using Videos. *Proc. ICCV*, 2015.

[100] Markus Weber, Max Welling, and Pietro Perona. Towards automatic discovery of object categories. *CVPR*, 2000.

[101] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. DeepFlow: Large Displacement Optical Flow with Deep Matching. *ICCV*, 2013.

[102] Bernard Widrow. The "Rubber-Mask" Technique-II. Pattern Storage and Recognition. In *Learning systems and intelligent robots*, pages 401–421. Springer, 1974.

[103] Jiajun Wu, Tianfan Xue, Joseph J. Lim, Yuandong Tian, Joshua B. Tenenbaum, Antonio Torralba, and William T. Freeman. Single Image 3D Interpreter Network. *arXiv*, 2016.

[104] Shengtao Xiao, Jiashi Feng, Junliang Xing, Hanjiang Lai, Shuicheng Yan, and Ashraf Kassim. Robust Facial Landmark Detection via Recurrent Attentive-Refinement Networks. In *Proc. ECCV*, 2016.

[105] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. *Proc. CVPR*, 2011.

[106] Xiang Yu, Feng Zhou, and Manmohan Chandraker. Deep Deformation Network for Object Landmark Localization. *arXiv*, 2016.

[107] Alan L Yuille. Deformable templates for face recognition. *Journal of cognitive neuroscience*, 3(1):59–70, 1991.

[108] Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen. Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment. In *Proc. ECCV*, 2014.

[109] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful Image Colorization. In *Proc. ECCV*, 2016.

[110] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *Proc. CVPR*, pages 2694–2703, 2018.

[111] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *Proc. ECCV*, 2014.

[112] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Learning Deep Representation for Face Alignment with Auxiliary Attributes. *PAMI*, 2016.

[113] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. CVPR*, 2017.

[114] Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qixing Huang, and Alexei A. Efros. Learning Dense Correspondences via 3D-guided Cycle Consistency. *CVPR*, 2016.

[115] Tinghui Zhou, Yong Jae Lee, Stella X. Yu, and Alexei A. Efros. FlowWeb: Joint image set alignment by weaving consistent, pixel-wise correspondences. *CVPR*, 2015.

[116] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proc. CVPR*, 2012.