Self-supervised learning of class embeddings from video

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

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

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

Abstract

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

1. Introduction

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

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

In order to learn the image embedding from a video dataset, the following proxy task is used. Given two frames
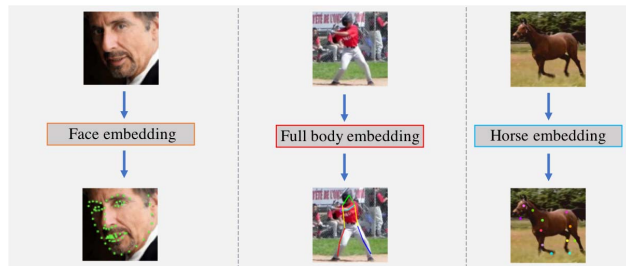

Figure 1. The aim of this work is to obtain a class-specific *image* embedding by self-supervised learning on a large collection of videos. The learned embedding can then be used for a variety of downstream tasks and datasets.

from the same video, their image embeddings are used to warp one of the frames into the other.

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

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

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

In summary, our contributions are as follows.

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

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

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

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

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

2. Related work

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

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

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

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

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

3. A self-supervised representation

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

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

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

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

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

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

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

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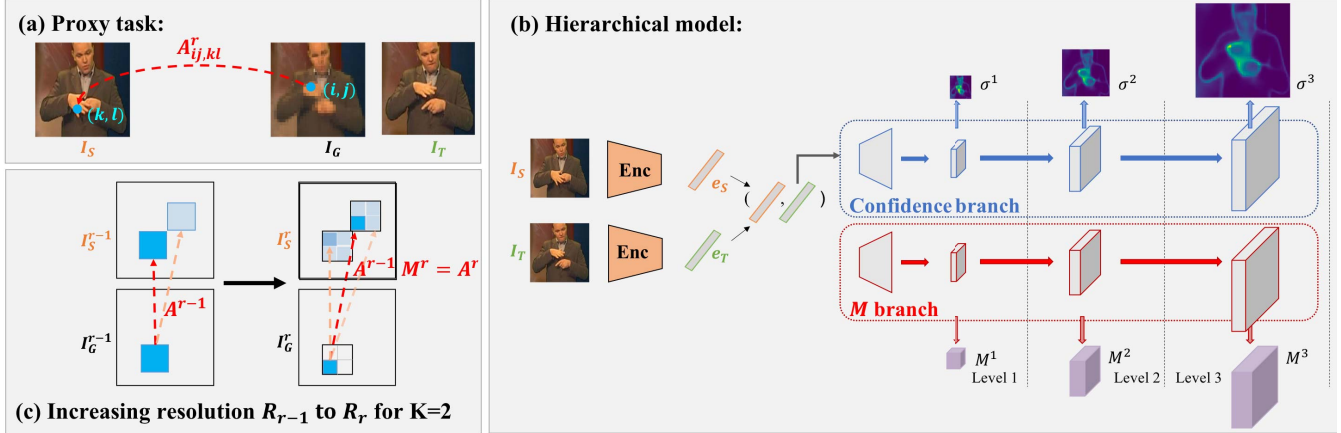
Figure 2. An overview of the approach. (a) The proxy task used to train the model. Given a source frame I_S and a target frame I_T, our model learns a mapping $\mathbf{A}^r$ to warp the source frame into a generated frame I_G. I_G should match I_T. (b) The model in more detail. The two frames are mapped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

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

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

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

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

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

In a traditional CNN, as we decode, we would have to keep track of the probabilities for a pixel (i, j) matching

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

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

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

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

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

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

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

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

3.2. Modelling occlusion and background

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

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

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

3.3. Dealing with multiple modes

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

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

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

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

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

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

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

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

4. Architecture and training

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

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

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

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

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

5. Experiments

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

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

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

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

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

5.1. Predicting landmarks

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

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

5.1.1 Upper body

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

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

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

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

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

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

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

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

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

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

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

in Table 3. For AFLW, we report results on the 5-always visible landmarks (AFLW5) as well as for all 21 landmarks (AFLW21). Qualitative results are shown in Fig. 4.

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

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

5.1.2 Faces

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

Our embedding is used to regress landmarks on the AFLW, 300-W, and MAFL datasets and results are reported

5.1.3 Full body

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

We use the learned embedding to regress landmarks. Quantitative results are reported in Table 4 and qualitative results in Fig. 5. We perform better than the baselines, and approach the performance of methods trained with deep learning on this dataset. Similarly to upper bodies, [53] degenerates to predicting the identity transformation, demonstrating the effectiveness of our method.
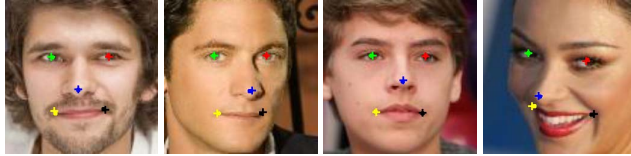
(a) **MAFL**. Crosses are predictions, dots GT.
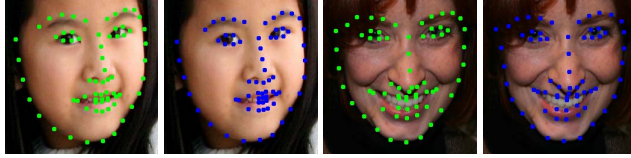
(b) **300W**. Blue is GT, green predictions.
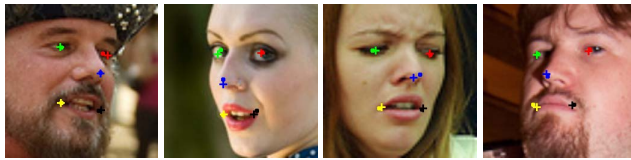
(c) **AFLW5**. Crosses are predictions, dots GT.

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

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

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

5.1.4 Non-human object classes: horses

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

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

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

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

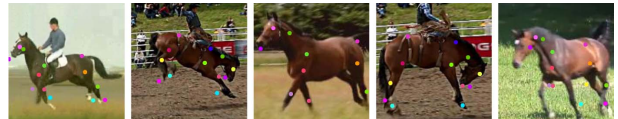Fig. 6, more results are shown in the supplementary material.

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

5.2. Predicting visibility

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

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

6. Conclusion

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

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

Acknowledgements

References

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

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

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

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

[5] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman. Domain adaptation for upper body pose tracking in signed TV broadcasts. In *Proc. BMVC*, 2013. 5, 6

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[55] B. Xiaohan Nie, C. Xiong, and S.-C. Zhu. Joint action recog-

nition and pose estimation from video. In *Proc. CVPR*, 2015. 7

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

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

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

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

[60] Y. Zhang, Y. Guo, Y. Jin, Y. Luo, Z. He, and H. Lee. Unsupervised discovery of object landmarks as structural representations. In *Proc. CVPR*, 2018. 2, 7

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

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

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

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