Self-Supervised Segmentation by Grouping Optical-Flow

Aravindh Mahendran   James Thewlis   Andrea Vedaldi

Visual Geometry Group
Dept of Engineering Science, University of Oxford
{aravindh,jdt,vedaldi}@robots.ox.ac.uk

Abstract. We propose to self-supervise a convolutional neural network operating on images using temporal information from videos. The task is to learn a representation of single images and the supervision for this is obtained by learning to group image pixels in such a way that their collective motion is “coherent”. This learning by grouping approach is used as a pre-training as well as segmentation strategy. Preliminary results suggest that the segments obtained are reasonable and the representation learned transfers well for classification.

1 Introduction

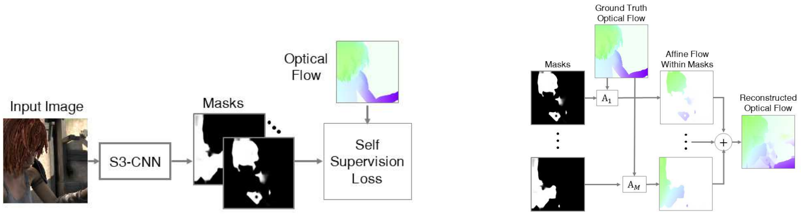
An increasingly popular approach to representation learning is to use proxy tasks that do not require the use of manual annotations. In this paper, we explore using motion cues, represented as optical flow, to formulate a proxy task for self-supervision. Inspired by Gestalt principle of common fate, we develop a framework which groups pixels that constitute “coherent” motion. Crucially this grouping is obtained by looking at a single image only. The optical flow is used only in the loss function. Therefore, at test time, the model can be deployed without video or flow information. The underlying assumption is that a segment containing an object exhibits “coherent” motion. Therefore a segmentation with our objective will learn to segment objects or object-parts. We call this framework *Self-Supervised Segmentation-CNN* or *S3-CNN* . An illustration is provided in fig. 1a.

Our formulation can be easily extended to the case where motion is induced by action / ego-motion. This extension is more expensive to experiment with and hence we restrict ourselves to offline videos.

2 Related Work

Self-Supervised Learning. *S3-CNN* is a self-supervised pre-training scheme to learn a feature extractor that can be fine-tuned for other tasks. We review closely related prior works by grouping them based on the nature of their pre-training loss.

The first group comprises methods that **predict an auxiliary input y given an image x.** For example using RNNs to *predict future frames in videos* [1].



(a) *S3-CNN* framework. (b) Affine Motion Loss

Fig. 1: (a) We propose to learn a neural network operating on images using temporal information contained in videos as supervision. The learning goal is to predict regions that are likely to have "coherent" optical flow. Flow can be observed by the loss, but not by the CNN. It encourages the network to learn about object-part-like regions in images. (b) Affine Motion Loss: Optical flow within each region is approximated using an affine transformation $(A_1, \cdots, A_M)$. These are recombined to give a reconstructed flow which is compared against ground truth.

Similarly, *Colorization* [2,3] predicts colour given grayscale input. A generalization to arbitrary pairs of modalities was proposed in [4]. Recent work has explored the geometric target of surface normals [5]. Closely related to our work is the use of *video segmentation* by [6]. They use an off-the-shelf video segmentation method to construct a foreground-background segmentation dataset to pretrain a CNN. We differ from them in that we do not require a sophisticated pre-existing pipeline to extract video segments, but use optical flow directly.

The second group of self-supervised methods **reconstruct (properties of) the image $\mathbf{x}$ given an incomplete or corrupted version of the same**. For example, [7] solve the inpainting problem, where part of the image is occluded. Alternative low dimensional targets have been explored by the community. For example, [8] learn to predict the global image rotation. [9] predict the relative position of two patches extracted from an image. [10,11] solve a jig-saw puzzle problem. [12] improve upon context based methods. The temporal analog of these are methods that predict the correct ordering of frames [13,14,15] or embed frames using temporal cues [16,17,18,19,20]. [21] train a siamese style convolutional neural network to predict the transformation between two images. [22] use videos along with spatial context pretraining [9] to construct an image graph. Transitivity in the graph is exploited to learn representations.

Our approach borrows from both paradigms. We predict a property of image $\mathbf{x}$ – a grouping of its pixels. At the same time, we supervise these segments using auxiliary data. This adds richer supervision than can be obtained by looking at cues contained in image $\mathbf{x}$ alone.

Segmentation Cues. Our method is based on using various motion cues to evaluate image regions and in this way relates to classical work [23,24,25]. These methods, however, use motion at test/inference time while we use it only at training time for supervision.

3 Method: Self-Supervised Grouping Losses

Our idea is to learn a CNN that predicts a segmentation $\Phi : \mathbf{x} \mapsto \mathbf{m} \in \{1, \dots, L\}^{H \times W}$ of the image. Pixels $u \in [1, \dots, H] \times [1, \dots, W]$ within each region l are assumed to be I.I.D with respect to a simple parametric distribution $p(f_u|\theta_l)$ where f_u is the flow at pixel u. Marginalizing the region parameters $p(\theta_l)$ results in the model:

$$p(\mathbf{f}|\mathbf{m}) = \prod_{l=1}^{L} \int \left[\prod_{u:m_u=l} p(f_u|\theta_l) \right] p(\theta_l)\, d\theta_l. \tag{1}$$

Crucially, due to the marginalization, network Φ is not tasked with predicting the transformation parameters θ, but only the regions $\mathbf{m}$. As a simpler alternative to marginalizing by integration, in the rest of this extended abstract we marginalize the model parameters by maximization and drop the prior on the parameters, so that the probability density for a region is written as:

$$p(\mathbf{f}|\mathbf{m}) = \prod_{l=1}^{L} \max_{\theta_l} \prod_{u:m_u=l} p(f_u|\theta_l). \tag{2}$$

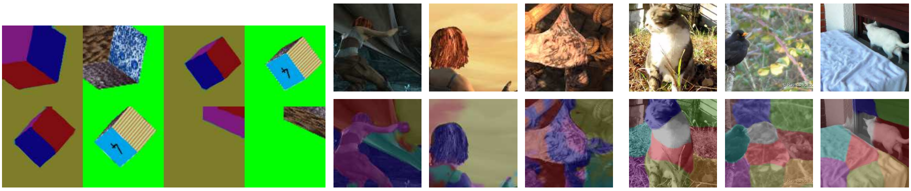
We further adapt the formulation for soft segments $\mathbf{m} \in [0,1]^{H \times W \times L}$. We experiment with two choices of θ_l - Affine transforms and flow-magnitude histograms.

Affine transformations: We fit an affine motion model to the optical flow within each segment. This “fit” corresponds to the max operation in eq. (2) and is computed by solving a weighted least squares problem. As a proxy for the likelihood in eq. (2), our loss function is a robust residual between the affine approximation and the optical-flow $\mathbf{f}$. This is a motion based self-supervision loss which conveys a notion of coherent motion within each segment based on an affine approximation of its optical flow. Computing this loss requires solving a weighted least squares online in the network's forward pass which is a simple combination of matrix arithmetic and a matrix inverse all of which are differentiable.

Low Entropy Motion Loss. Instead of fitting parametric motion models to regions, histograms offer a general non-parametric alternative. We compute a histogram for the flow-magnitude within each segment. The histogram itself constitutes θ_l (eq. (2)) and $\mathbf{f}$ is the flow magnitude rather than 2D flow vectors. The entropy of this histogram is used as a loss, again as a proxy for the likelihood in eq. (2). We assume that a segment straddling different independently movable objects will constitute a high entropy histogram. In other words, we assume a histogram entropy loss encourages the separation of independently movable objects.

4 Experiments

We show qualitative results as sample image segmentations generated by our *S3-CNN* . We then assess its capability to pre-train for image recognition. In



(a) Orthographic projection: Segment cube faces (train set).

(b) Sintel - $L = 5$. Col-1: Train Set, Col-2,3: Val.

(c) Youtube Objects (Val. set) - $L = 10$

Fig. 2: Predicted regions are visualized by a colour map.

these experiments, we use a Fully Convolutional Network [26] FCN-8s model on VGG-16 [27]. FCN scores are mapped to soft segmentation masks as in [28]. Parameter free batch normalization [29] was used after every convolutional and fully connected layer in the pretraining stage.

Qualitative Results: First, we demonstrate our method on a toy problem. The data consists of synthetic videos of a single translating and rotating 3D textured cube (fig. 2a); paired with the corresponding optical flow field. Cubes are imaged under an orthographic camera, so that the affine motion model of section 3 applies to each cube face. We train a network to predict 5 segments with self-supervision from five sequences containing 99 frames each. As seen in fig. 2a, the network learns to correctly group together the pixels in each cube face.

Next we consider Sintel [30], containing videos from an animated 3D movie and use the affine flow model to learn a grouping of image regions. While this model offers only a loose approximation of the complex motions in these videos, informative regions can still be learned as the affine approximation is quite good for body parts and other small objects. The results obtained, on training and validation images, by the model trained using the affine flow loss on 20 training sequence from Sintel are shown in fig. 2b, where several objects and parts are highlighted. Notice in particular that even bodies and heads are picked up despite their non-planar structure.

In the case of real world data, we have large systematic noise in automatically computed optical flow. We find that the histogram entropy loss works best in these cases. Figure 2c shows qualitative results on frames from the Youtube objects dataset[31,32]. These were predicted by our model trained on frames extracted from YFCC100m [33] and supervised using the flow magnitude histogram entropy loss (section 3). The cat boundaries align well with segments in the first column and a bird in the middle is segmented out. Also each segment caters to one spatial region. The teal coloured region is always in the middle left whereas the light green region is always in the top right corner.

Pre-training for Object Recognition: Our approach can also be used as a proxy to pre-train a generic feature extractor. These features can then be fine-tuned for other tasks such as image classification. To test this use, we follow the protocol of [34] to evaluate on Pascal VOC 2007 classification. Batch

Table 1: We fine-tune our model for VOC-07 classification (%mAP on test split).

Method	ImageNet	Random (∼ [2])	k-means [34]	Colorization [2]	*S3-CNN*
% mAP	86.9	59.85	56.5	77.2	76.35

normalization moments are absorbed into convolution filters and biases before fine-tuning.

We first pre-train our *S3-CNN* model on optical flow and frames extracted from videos in the YFCC100m dataset. We use 150k videos and compute optical flow between the first and fifth frame of each using EpicFlow[1] [35] with initial matches given by FlowFields [36]. This yields a dataset of 150k frames.

Table 1 lists methods that report results on VOC-07 classification using a VGG-16 based model. We observe that our *S3-CNN* model performs better than a non pretrained VGG-16. We are competitive to state-of-the-art models for VOC-07 classification: 76.35% mAP compared to 77.2% mAP of [2] despite using only 150k pre-training pairs compared to their pretraining dataset of 3.7M images. Lastly, we trained an AlexNet model akin to that of [6] by constructing an AlexNet FCN *S3-CNN* . We compare with them on VOC-07 classification and obtain 57.37% mAP versus their result of 61% mAP. This is promising given that we use 150k images versus their dataset of 1.6M images.

5 Conclusions

We have presented the *S3* framework, that allows supervising neural network architectures for general-purpose feature extraction using optical flow.

Acknowledgements: The authors gratefully acknowledge the support of ERC 677195-IDIU and AIMS CDT (EPSRC EP/L015897/1).

References

1. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: Proc. ICML. (2015)
2. Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding. In: Proc. CVPR. (2017)
3. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV. (2016)
4. Zhang, R., Isola, P., Efros, A.A.: Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In: Proc. CVPR. (2017)
5. 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)
6. Pathak, D., et al.: Learning features by watching objects move. In: CVPR. (2017)
7. Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proc. CVPR. (2016)
8. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised Representation Learning by Predicting Image Rotations. In: Proc. ICLR. (2018)

[1] Epic flow uses structured edge detection to obtain an edge map. This is trained using manually annotated data. We assume that the influence of this supervision is weak.

9. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proc. ICCV. (2015) 1422–1430
10. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: Proc. ECCV, Springer (2016) 69–84
11. Noroozi, M., Vinjimoor, A., Favaro, P., Pirsiavash, H.: Boosting self-supervised learning via knowledge transfer. In: Proc. CVPR. (2018)
12. Mundhenk, T., Ho, D., Y. Chen, B.: Improvements to context based self-supervised learning. In: Proc. CVPR. (11 2017)
13. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and Learn: Unsupervised Learning using Temporal Order Verification. In: Proc. ECCV. (2016)
14. Wei, D., Lim, J.J., Zisserman, A., Freeman, W.T.: Learning and using the arrow of time. In: Proc. CVPR. (2018) 8052–8060
15. Lee, H.Y., Huang, J.B., Singh, M.K., Yang, M.H.: Unsupervised representation learning by sorting sequence. In: Proc. ICCV. (2017)
16. Mobahi, H., Collobert, R., Weston, J.: Deep learning from temporal coherence in video. In: Proc. ICML, ACM (2009) 737–744
17. Isola, P., Zoran, D., Krishnan, D., Adelson, E.H.: Learning visual groups from co-occurrences in space and time. ICLR Workshop (2015)
18. Jayaraman, D., Grauman, K.: Slow and steady feature analysis: higher order temporal coherence in video. In: Proc. CVPR. (2016) 3852–3861
19. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: Proc. ICCV. (2015) 2794–2802
20. Gao, R., Jayaraman, D., Grauman, K.: Object-centric representation learning from unlabeled videos. In: Proc. ACCV. (2016)
21. Agrawal, P., et al.: Learning to see by moving. In: Proc. ICCV. (2015)
22. Wang, X., He, K., Gupta, A.: Transitive invariance for self-supervised visual representation learning. In: Proc. ICCV. (2017) 2794–2802
23. Isack, H., Boykov, Y.: Energy-based geometric multi-model fitting. IJCV **97** (April 2012) 123–147
24. Delong, A., Osokin, A., Isack, H., Boykov, Y.: Fast approximate energy minimization with label costs. IJCV **96** (Jan. 2012) 1–27
25. Sivic, J., et al.: Object level grouping for video shots. IJCV **67**(2) (2006) 189–210
26. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proc. CVPR. (2015)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
28. Flynn, J., Neulander, I., Philbin, J., Snavely, N.: Deepstereo: Learning to predict new views from the world's imagery. In: Proc. CVPR. (2016) 5515–5524
29. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proc. ICML. (2015)
30. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: Proc. ECCV. (Oct 2012) 611–625
31. Prest, A., et al.: Learning object class detectors from weakly annotated video. In: Proc. CVPR. (2012)
32. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: Proc. ECCV. (2010)
33. Thomee, B., et al.: Yfcc100m: the new data in multimedia research. ACM (2016)
34. Krähenbühl, P., Doersch, C., Donahue, J., Darrell, T.: Data-dependent initializations of convolutional neural networks. ICLR (2016)
35. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Epicflow: Edge-preserving interpolation of correspondences for optical flow. In: Proc. CVPR. (2015)
36. Bailer, C., Taetz, B., Stricker, D.: Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In: Proc. ICCV. (2015)