# Cross and Learn: Cross-Modal Self-Supervision

Nawid Sayed[1], Biagio Brattoli[2], and Björn Ommer[2]

Heidelberg University, HCI / IWR, Germany
[1]nawid.sayed42@gmail.com
[2]{firstname.lastname}@iwr.uni-heidelberg.de

**Abstract.** In this paper we present a self-supervised method for representation learning utilizing two different modalities. Based on the observation that cross-modal information has a high semantic meaning we propose a method to effectively exploit this signal. For our approach we utilize video data since it is available on a large scale and provides easily accessible modalities given by RGB and optical flow. We demonstrate state-of-the-art performance on highly contested action recognition datasets in the context of self-supervised learning. We show that our feature representation also transfers to other tasks and conduct extensive ablation studies to validate our core contributions. Code and model can be found at `https://github.com/nawidsayed/Cross-and-Learn`.

## 1 Introduction

In the last decade, Convolutional Neural Networks (CNNs) have shown state of the art accuracy on a variety of visual recognition tasks such as image classification [18], object detection [10] and action recognition [33]. The success of CNNs is based on *supervised learning* which heavily relies on large manually annotated datasets. These are costly to obtain, however unannotated data in form of images and videos have become easily available on a very large scale. Recently this encouraged the investigation of *self-supervised learning* approaches which do not require semantic annotations for the data. Here the general procedure involves pretraining a network on a surrogate task which requires semantic understanding in order to be solved. Although the gap is closing, self-supervised approaches usually cannot compete with feature representations obtained by supervised pretraining on large scale datasets such as ImageNet [31] or Kinetics [15].

In this paper, we use cross-modal information as an alternative source of supervision and propose a new method to effectively exploit mutual information in order to train powerful feature representations for both modalities. The main motivation of our approach is derived from the following observation: Information shared across modalities has a much higher semantic meaning compared to information which is modality specific. We showcase this point in Fig. 1 where we present a tuple of pairs obtained from an action recognition video dataset. We can see that cross-modal information such as the barbell or the baseball bat provide good clues to identify the action. On the other hand modality specific

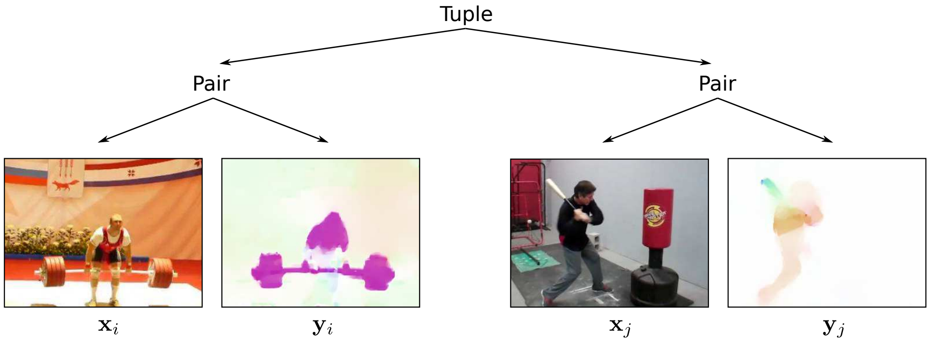

**Fig. 1.** A tuple of pairs obtained from UCF-101 using RGB and optical flow as modalities. The cross-modal information (barbell, bat, person) has a much higher semantic meaning than the modality specific information (background, camera movement, noise)

information such as the background or camera motion do not help to identify the action.

Feature representations which are sensitive to cross-modal information and invariant to modality specific content are therefore desirable. The latter condition is fulfilled if the feature representations of a pair are similar to each other. The former condition is fulfilled if the feature representations are also dissimilar across different pairs. To achieve that we utilize a trainable two stream architecture with one network per modality similar to [33] and propose two different loss contributions which guide the networks to learn the desired feature representations.

For our method we require paired data from different modalities on a large scale. We therefore apply our method to video data which provide very easily accessible modalities RGB and optical flow in practically unlimited quantity. Our choice for the modalities is also motivated by past work where it has been shown that RGB and optical flow complement each other in the context of action recognition [33].

In order to demonstrate the effectiveness of our cross-modal pretraining, we conduct extensive experimental validation and a complete ablation study on our design choices. Our method significantly outperforms state-of-the-art unsupervised approaches on the two highly contested action recognition datasets UCF-101 [34] and HMDB-51 [19] while pretraining for only *6h GPU time* on a single NVIDIA Titan X (Pascal). We also show the transferability of our feature representation to object classification and detection by achieving competitive results on the PASCAL VOC 2007 benchmark.

## 2 Related Work

In recent years, unsupervised deep learning has been an hot topic for the research community due to the abundance of unannotated data. One of the first approaches was the Auto-Encoder [36,16] which learns to encode an image in a lower dimensional features space. The obtained feature representation would mostly focus on low level statistics instead of capturing the semantic content of the input.

A significant step forward in deep unsupervised learning was made by [6,1,40] which introduced the paradigm of self-supervision in the context of deep learning. This family of methods exploits the inherent structure of data in order to train a CNN through a surrogate task. The most recent approaches can generally be divided into two groups, those which make use of static images and those which make use of videos.

As for static images, there are several approaches exploiting color information [42,20] and spatial context [6,26,28]. For the latter an image is tiled into patches and their relative position is exploited as a supervisory signal in order to obtain a good feature representation. These approaches have shown reliable results on the task of object detection and surface normal estimation. There is also a variety of works which specifically aim to learn reliable posture representations without supervision [3,30,23,32,4].

Approaches which use video data mainly exploit temporal context as source of supervision [40,24,4,5,2]. A common idea is to shuffle frames of a video clip and use a network to reconstruct or verify the correct chronological order [24,9,21]. The obtained feature representations generalize well to action recognition. Closely related to our approach is the work of S. Purushwalkam and A. Gupta [30]. They use RGB and optical flow data obtained from videos and learn a feature representation which transfers to the task of pose estimation. Although they achieve good results for action recognition, our method exploits cross-modal information in RGB and optical flow data much more effectively.

There is a large body of work which demonstrates the effectiveness of utilizing multimodal data in order to solve a variety of tasks. For example [33,8] show that RGB and optical flow are complementary modalities in the context of action recognition.

S.Gupta et al. [11] train a feature representation for depth data by distilling the knowledge from a teacher network pretrained on ImageNet [31]. The work of [38] proposes cross-modal ranking in conjunction with structure preserving constrains in order to learn a joint feature representation between image and text data. Both works show successful cross-modal supervision but rely on annotated data, which is not required for our approach.

The work of [25] utilizes a bimodal deep autoencoder in order to learn a shared feature representation for the audio and visual stream of video data. Similarly [41] extends this work to temporal feature representations.

## 3 Approach

In this section we present an efficient method for representation learning utilizing two different modalities. Our goal is to obtain feature representations which are sensitive to cross-modal information while being invariant to modality specific

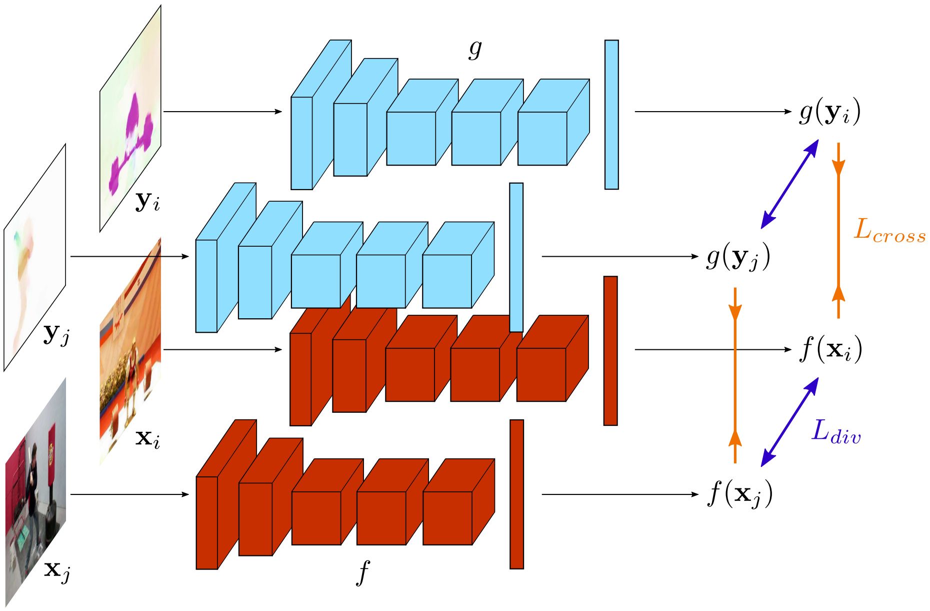

**Fig. 2.** Visualization of our model. Networks of same color share weights and are denoted by $f$ (red, dark) and $g$ (blue, bright) respectively. $f$ and $g$ are trained so that the cross-modal distance (vertical) decreases while the distance between different pairs (horizontal) increases

content. As already explained in the introduction, these conditions are fulfilled by feature representations which are similar for a pair and dissimilar across different pairs. To achieve the former we propose a cross-modal loss $L_{cross}$ and to achieve the latter we utilize a diversity loss $L_{div}$, both of which act directly in feature space thus promising better training signals.

Our method requires paired data from two different modalities $\mathbf{x} \in X$ and $\mathbf{y} \in Y$, which is available in most use cases i.e. RGB and optical flow. We utilize a two-stream architecture with trainable CNNs in order to obtain our feature representations $f(\mathbf{x})$ and $g(\mathbf{y})$. With exception of the first layer, the networks share the same architecture but do not share weights. To calculate both loss contributions we need a tuple of pairs $\mathbf{x}_i, \mathbf{y}_i$ and $\mathbf{x}_j, \mathbf{y}_j$ from our dataset. A visualization of our method is provided in Fig. 2.

**Cross-Modal Loss.** In order to enforce cross-modal similarity between $f$ and $g$ we enforce the feature representations of a pair to be close in feature space via some distance d. Solving this task requires the networks to ignore information which is only present in either $\mathbf{x}$ or $\mathbf{y}$

$$L_{cross}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{x}_j, \mathbf{y}_j) = \frac{1}{2}\left[\mathrm{d}(f(\mathbf{x}_i), g(\mathbf{y}_i)) + \mathrm{d}(f(\mathbf{x}_j), g(\mathbf{y}_j))\right]. \tag{1}$$

We utilize the bounded cosine distance for d, which is given by

$$\mathrm{d}(\mathbf{a},\mathbf{b}) = 1 - \frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{a}|\cdot|\mathbf{b}|} \leq 2. \tag{2}$$

We prevent singularities during training by adding a small $\epsilon$ into the square-root of the euclidean norm. We also experimented with the unbounded euclidean distance for d, as done in [11] but found that it leads to an unstable training.

**Diversity Loss.** We obtain diversity by enforcing the feature representation for both modalities to be distant across pairs with respect to the same distance d as before. This spreads the features of different pairs apart in feature space. Due to the cross-modal loss these features mostly encode cross-modal information, thus ensuring sensitive feature representations for this content. The distance across pairs therefore contributes negatively into the loss

$$L_{div}(\mathbf{x}_i,\mathbf{y}_i,\mathbf{x}_j,\mathbf{y}_j) = -\frac{1}{2}\left[\mathrm{d}(f(\mathbf{x}_i),f(\mathbf{x}_j)) + \mathrm{d}(g(\mathbf{y}_i),g(\mathbf{y}_j))\right]. \tag{3}$$

Here we assume our data to be diverse in the sense that the probability of the two pairs being very similar to each other is sufficiently low. This assumption holds true for most real world data and for modern datasets as they emphasize diversity in order to be challenging to solve. Nevertheless our approach handles two very similar pairs well since our loss is bounded limiting the impact of false information during training.

**Combining Both Loss Contributions.** Our final training objective is given by a weighted sum of both loss contributions which leaves the weighting as an additional hyperparameter of our method. Determining this hyperparameter might sound like a challenging problem but we find that there are not many reasonable options in practice. We make two observations considering the balance of cross-modal and diversity loss. 1) Weighting $L_{cross}$ over $L_{div}$ encourages $f$ and $g$ to collapse their feature representation onto a single point. 2) Weighting $L_{div}$ over $L_{cross}$ significantly slows down the convergence of training and leads to a worse feature representation. It is important to note that these observations only hold true if the distance distributions in both loss contributions can quickly be equalized by the network during training. This is usually the case for modern deep CNN architectures with multiple normalization layers.

Given our observations, we weight both loss contributions equally which yields our final loss

$$L(\mathbf{x}_i,\mathbf{y}_i,\mathbf{x}_j,\mathbf{y}_j) = L_{cross}(\mathbf{x}_i,\mathbf{y}_i,\mathbf{x}_j,\mathbf{y}_j) + L_{div}(\mathbf{x}_i,\mathbf{y}_i,\mathbf{x}_j,\mathbf{y}_j). \tag{4}$$

We obtain our training signal by randomly sampling $B$ tuples ($2B$ pairs) from our dataset and averaging the loss in equation 4 across the tuples. In section 4.5 we further validate our design choices.

## 4 Experiments

In this section we quantitatively and qualitatively evaluate our self-supervised cross-modal pretraining mainly using RGB and optical flow as our modalities of choice. We will therefore refer to $f$ and $g$ as *RGB-network* and *OF-network* respectively. First, we present the implementation details for the self-supervised training. Afterwards, we quantitatively evaluate our feature representation by fine-tuning onto highly contested benchmarks and show a qualitative evaluation of the learned feature representation. Finally, we close with an extensive ablation study where we validate our design choices.

### 4.1 Implementation Details

We use the CaffeNet [14,18] architecture as this is well established for representation learning and extract our feature representation from fc6 after the ReLU activation. We use dropout with $p = 0.5$. If not stated otherwise, we use batch normalization [12] on all convolutional layers in order to be comparable with [21]. We conduct our experiments using pytorch [29].

We use the trainset of UCF-101 for our pretraining in order to be consistent with previous work, the (TV-$L_1$) optical flow data is pre-computed using the OpenCV [13] implementation with default parameters. Due to having temporally ordered data we use a stack of 10 consecutive optical flow frames and its temporally centered RGB frame for each pair. This is motivated by past work [33,30] which show that additional optical flow frames are beneficial for motion understanding. The input of the OF-network therefore has 20 channels.

During training we randomly sample 60 videos ($B = 30$ tuples) and from each video we randomly sample a pair. Similarly to [21,24] the latter sampling is weighted by the average flow magnitudes in a pair which reduces the number of training samples without meaningful information.

The following augmentation scheme is applied independently to both pairs. Given the optical flow stack and the RGB frame we first rescale them to a height of 256 conserving the aspect ratio. We then randomly crop a 224x224 patch and apply random horizontal flips, both of which are done identically across the 10 flow frames and the RGB frame. We further augment the flow stack by applying random temporal flips (augmentation from [30]) and subtract the mean [33]. Additionally we apply channel splitting [21] which yields a color-agnostic input forcing the network to learn a less appearance based representation.

We find that a proper choice for $\epsilon$ is crucial for the stability of our training and recommend using $\epsilon = 1e - 5$ or bigger. We use SGD with momentum of 0.9 and weight decay of $5e-4$ with an initial learning rate of 0.01 which is reduced to 0.001 after 50K iterations. Training is stopped after 65K iterations and requires only *6h GPU time* on a NVIDIA Titan X (Pascal).

### 4.2 Action Recognition

We evaluate our approach on the action recognition datasets UCF-101 and HMDB-51 which emerged as highly contested benchmarks for unsupervised representation learning on videos. To show the versatility of our approach we also utilize the considerably deeper VGG16 architecture, using its fc6 layer as our feature representation. We train on a single GPU using $B = 12$ and a learning rate of 0.005 for a total of 160K iterations, leaving all other hyperparameters unchanged. Pretraining on a single GPU completes in less than two days.

**Datasets.** The action recognition datasets are typically divided in three splits. UCF-101 is composed of 13K video clips per split, 9.5K for training and 3.5K for testing, divided in 101 action categories. HMDB-51 consists of 51 classes with 3.5K videos for training and 1.4K for testing per split. Although both datasets are created in a similar manner, HMDB-51 is more challenging as there can be multiple samples from the same video with different semantic labels (draw sword, sword exercise, sword fight). Samples from different classes can therefore have very similar appearance or posture information and require an understanding of motion to be distinguished properly [19].

**Finetuning Protocol.** We start with a randomly initialized CaffeNet and use the parameters of our pretrained RGB-network to initialize the convolutional and batch normalization layers (FC-layers start from scratch). For training and testing we follow the common finetuning protocol of [33] in order to be comparable with previous work [24,21]. We provide test classification accuracies over 3 splits for both datasets. In order to ensure that the test set of UCF-101 does not contain images which have been previously seen during pretraining we conduct pretraining and finetuning on the same split.

**Table 1.** Test classification accuracies over 3 splits for different pretraining schemes. If not stated otherwise the referenced methods use the CaffeNet architecture. +: indicates only Split 1. Our RGB-network outperforms previous state-of-the-art self-supervised methods especially on HMDB-51, while also being very efficient

| | Dataset | Traintime | UCF-101 | HMDB-51 |
|---|---|---|---|---|
| Random | None | None | 48.2 | 19.5 |
| ImageNet [21] | ImageNet | 3 days | 67.7 | 28.0 |
| Shuffle and Learn [24] | UCF-101 | - | 50.2 | 18.1 |
| VGAN [37] (C3D) | flickr (2M videos) | $> 2$ days | 52.1 | - |
| LT-Motion [22] (RNN) | NTU (57K videos) | - | 53.0 | - |
| Pose f. Action [30] (VGG) | UCF,HMDB,ACT | - | 55.0 | 23.6 |
| OPN [21] | UCF-101 | 40 hours | 56.3 | 22.1 |
| Our | UCF-101 | 6 hours | **58.7** | **27.2** |
| Random (VGG16)+ | None | None | 59.6 | 24.3 |
| Our (VGG16)+ | UCF-101 | 1.5 days | 70.5 | 33.0 |

**Results.** As shown in Table 1, our approach outperforms the previous state-of-the-art method [21] by over 2% and 5% on UCF-101 and HMDB-51 respectively. For the latter dataset our network almost closed the gap to its ImageNet pretrained counterpart. As mentioned before HMDB-51 requires a better understanding of motion and is not easily solved by appearance information. This favors our method because solving our pretraining task requires a good understanding of motion. Compared to past approaches which make use of optical flow [30,22] our method exploits motion information much more effectively. We further expand this point in our qualitative evaluation and ablation studies.

### 4.3 Transfer Learning

In this section, we evaluate the transferability of our learned feature representation to object classification and detection. To this end we use the Pascal VOC 2007 benchmark and compare to other recent unsupervised learning methods.

**Datasets.** The Pascal VOC 2007 dataset [7] provides annotation for 20 classes on 10K images, containing in total 25K annotated objects. The images are split equally between training/validation and testing set. The annotations provide class label and bounding box location for every object.

The ACT [39] dataset is composed of 11K video clips of people performing various actions. The dataset contains 43 different action categories which can be summarized to 16 super-classes. For our pretraining we use the trainset containing 7K videos and do not make use of the semantic labels.

**Pretraining and Finetuning Protocol.** We apply our model onto the training videos from UCF-101, HMDB-51 and ACT [39] which yield about 20K videos during pretraining. We double the number of iterations to 130K (dropping learning rate at 100K) and leave all other hyperparameters unchanged. Similar to section 4.2 we use the Convolutional layers of our RGB-network to initialize a new CaffeNet for finetuning but we do not transfer the batch normalization layers, as suggested by [21]. In order to be comparable to [21] we finetune our RGB-network on Pascal VOC for the multi-class classification task following Krähenbühl et al. [17] protocol and use Fast-RCNN [10] framework when fine-tuning for object detection.

**Results.** Table 2 shows a comparison of our approach to other unsupervised learning methods. Among the methods which do not make use of ImageNet data we improve upon state-of-the-art in both categories. We also show competitive accuracy in object classification among methods which do use ImageNet data. This result is remarkable as the datasets we used for pretraining are very dissimilar to the ImageNet or Pascal VOC dataset. This demonstrates the transferability of the feature representation obtained by our method.

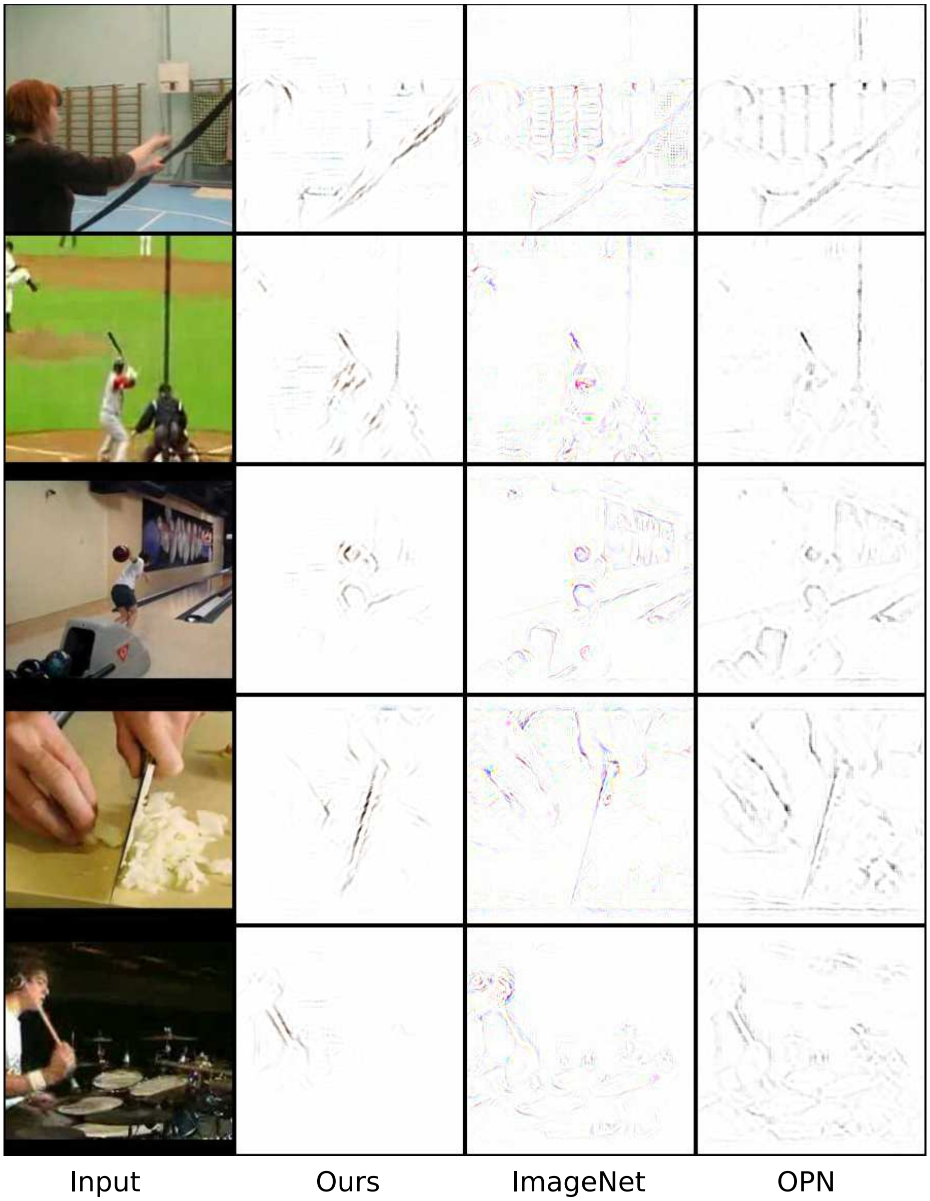

**Fig. 3.** Comparison of the input activations for different models. The first column represents the input sample, second is our RGB-network, third is an ImageNet pretrained network and lastly OPN [21]. The activation are computed using guided backpropagation [35]. In contrast to the other approaches our network is able to identify meaningful objects (Bow, bowling ball, drumstick ...) while also ignoring irrelevant background content

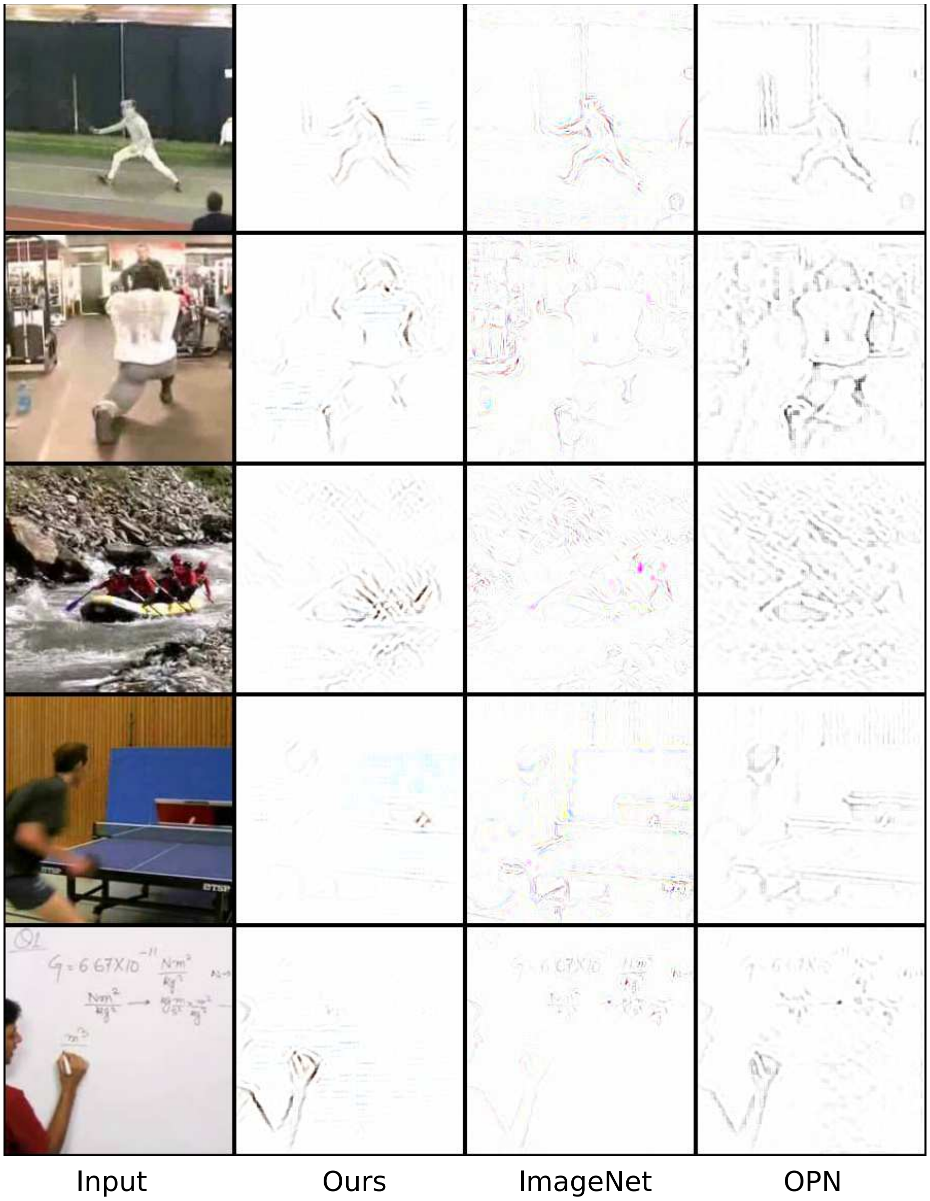

**Fig. 4.** Additional input activations for our RGB-network, ImageNet pretraining and OPN [21]. Our network is also capable of identifying small moving objects (table tennis ball, pen), and ignore most of the background

**Table 2.** Results for Pascal VOC 2007 classification and detection. First segment shows baseline performance for supervised training. Methods in the second segment used large scale image data to pretrain their models whereas methods in the last segment used video data during pretraining

| | Dataset | Traintime | Classification | Detection |
|---|---|---|---|---|
| ImageNet [21] | ImageNet | 3 days | 78.2 | 56.8 |
| Context [6] | ImageNet | 4 weeks | 55.3 | 46.6 |
| Counting [27] | ImageNet | - | 67.7 | 51.4 |
| Jigsaw [26] | ImageNet | 2.5 days | 67.6 | 53.2 |
| Jigsaw++ [28] | ImageNet | - | 72.5 | 56.5 |
| Shuffle and Learn | UCF-101 | - | 54.3 | 39.9 |
| OPN [21] | UCF,HMDB,ACT | $>$ 2 days | 63.8 | 46.9 |
| Our | UCF,HMDB,ACT | 12 hours | **70.7** | **48.1** |

### 4.4 Qualitative Evaluation

In this section we present a qualitative evaluation of our learned feature representations by showing input gradients obtained from high-level layers. We calculate the input layer activations for our pretrained RGB-network and compare to other approaches on the UCF-101 testset. To obtain the results in Fig. 3 and 4 we use guided backpropagation [35] on the 100 strongest activations in pool5 of the networks. We can see that our training leads the network to focus on key moving objects involved in an action (bowling ball, drumstick, pen...) while ignoring irrelevant background information. Contrary, the ImageNet and OPN pretrained networks take the entire appearance of the scene into account indicating a more appearance based feature representation.

### 4.5 Ablation Study

In this section we give a complete ablation study on all of our design choices. We start by investigating the effectiveness of our model without using optical flow as second modality. Afterwards we reveal that our method is mutually beneficial for the used modalities. We close by demonstrating the importance of our cross-modal loss and show that a more mainstream model design yields inferior results. We conduct our evaluation with CaffeNet on split 1 of UCF-101 and HMDB-51 only using UCF-101 for pretraining.

**Using Stack of Frame Differences as Second Modality.** In the works of [9] and [21] reliable results were achieved using stack of frame differences (SOD) for action recognition. Inspired by their findings we train our model using RGB and SOD as modalities (we use the term SOD-network for $g$ in this context). We apply the same data sampling procedure as described in section 4.1 with slight modifications. To obtain a SOD we sample 5 RGB frames and calculate the difference between successive frames yielding us a SOD with 4 frame

**Table 3.** Comparing different choices of modalities. The left segment shows test accuracies obtained by using RGB and optical flow as modalities whereas the right segment reports test accuracies using RGB and SoD. We observe that all modalities benefit significantly from our pretraining

| Dataset | Pretraining | RGB | OF | RGB | SOD |
|---|---|---|---|---|---|
| UCF-101 | Random weights | 49.1 | 76.4 | 49.1 | 64.5 |
| UCF-101 | Our | **59.3** | **79.2** | **55.4** | **66.3** |
| HMDB-51 | Random weights | 19.2 | 47.1 | 19.2 | 30.8 |
| HMDB-51 | Our | **27.7** | **51.7** | **23.5** | **33.3** |

differences. The RGB input is then just the temporally centered RGB frame. We only apply random crops and random horizontal flips to SOD but keep the same augmentation as 4.1 for RGB. All other hyperparameters are kept the same for pretraining and finetuning.

Test classification results on split 1 of UCF-101 and HMDB-51 are reported in the right segment of Table 3. The RGB-network achieves 55.4% and 23.5% test classification accuracy respectively which is still competitive to previous state of the art methods.

**Mutual Benefit Between the Modalities.** In order to demonstrate the mutual benefit of our method we finetune our obtained OF and SOD-network onto split 1 of UCF-101 and HMDB-51. We apply the same scheme already used for RGB to finetune both networks. We compare to randomly initialized networks and report results in Table 3.

We observe that our pretraining increases the classification accuracy on all used modalities. The OF-network acts supervisory on the RGB-network and also vice versa. This result is quite remarkable given the huge performance gap on the classification task and also holds true for RGB and SOD. Although the weaker modality benefits more from the pretraining it does not drag down the results of the stronger modality but instead gives a considerable boost especially for HMDB-51.

**Alternative Design Choices.** As already indicated in section 3 we obtain trivial feature representations if we do not utilize the diversity loss. In the following we also demonstrate the importance of our cross-modal loss. To this end we pretrain our model the same way as before but set the cross-modal loss contribution to zero. We then finetune its RGB-network on UCF-101 and HMDB-51 and present our findings in Table 4. Without the cross-modal loss we perform on par with a randomly initialized network which indicates a failed pretraining and highlights the importance of the cross-modal loss.

To validate our design choices further we now compare to a more mainstream approach of exploiting cross-modal information which is inspired by the work of [30]. We concatenate the fc6 layers of our RGB and OF-network (8192 channels) and put a modified CaffeNet binary classifier (fc7, fc8) on top of it. We

**Table 4.** Ablation study for alternative design choices conducted on split 1 of UCF-101 and HMDB-51. Models were pretrained onto UCF-101. Utilizing $L_{div}$ only performs on par with random weights. A more mainstream model design leads to a worse performance

| Dataset | Pretraining | RGB | OF |
|---|---|---|---|
| UCF-101 | Random weights | 49.1 | 76.4 |
| UCF-101 | Only $L_{div}$ | 49.0 | 75.3 |
| UCF-101 | Concat | 57.6 | 77.8 |
| UCF-101 | Our | **59.3** | **79.2** |
| HMDB-51 | Random weights | 19.2 | 47.1 |
| HMDB-51 | Only $L_{div}$ | 18.0 | 47.2 |
| HMDB-51 | Concat | 24.5 | 49.9 |
| HMDB-51 | Our | **27.7** | **51.7** |

preserve our sampling and augmentation protocol described in 4.1 in order to be as comparable to our approach as possible. Given two pairs $\mathbf{x}_i, \mathbf{y}_i$ and $\mathbf{x}_j, \mathbf{y}_j$ we now exploit the mutual information by posing a binary classification surrogate task. We assign the combinations $\mathbf{x}_i, \mathbf{y}_i$ and $\mathbf{x}_j, \mathbf{y}_j$ positive labels whereas $\mathbf{x}_i, \mathbf{y}_j$ and $\mathbf{x}_j, \mathbf{y}_i$ obtain negative labels. Essentially the network has to figure out if a presented RGB frame and flow stack belong to the same pair or not. We pretrain this model using the Cross-Entropy loss and use the same hyperparameters as our original model. Finetuning is then performed for both modalities in the same way as before. We reference this alternative model as Concat.

Results for our Concat model can be found in Table 4. We observe that the RGB-network of the Concat model is outperforming its randomly initialized counterpart but achieves significantly worse results than our pretraining.

## 5 Conclusion

Recently self-supervised learning emerged as an attractive research area since unannotated data is available on a large scale whereas annotations are expensive to obtain. In our work we reveal cross-modal information as an alternative source of supervision and present an efficient method to exploit this signal in order to obtain powerful feature representations for both modalities. We evaluate our method in the context of action recognition where we significantly outperform previous state-of-the-art results on the two highly contested action recognition datasets UCF-101 and HMDB-51. We also show the transferability of our feature representations by achieving competitive results in object classification and detection. We close out by conducting an extensive ablation study in which we validate all our design choices.

## Acknowledgments

We are grateful to the NVIDIA corporation for supporting our research, the experiments in this paper were performed on a donated Titan X (Pascal) GPU.

## References

1. Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: Proceedings of the IEEE International Conference on Computer Vision (2015)
2. Bautista, M., Fuchs, P., Ommer, B.: Learning where to drive by watching others. Proceedings of the German Conference Pattern Recognition (2017)
3. Bautista, M., Sanakoyeu, A., Sutter, E., Ommer, B.: Cliquecnn: Deep unsupervised exemplar learning. In: Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS) (2016)
4. Brattoli, B., Büchler, U., Wahl, A.S., Schwab, M.E., Ommer, B.: Lstm self-supervision for detailed behavior analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
5. Büchler, U., Brattoli, B., Ommer, B.: Improving spatiotemporal self-supervision by deep reinforcement learning. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
6. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision (2015)
7. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
8. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
9. Fernando, B., Bilen, H., Gavves, E., Gould, S.: Self-supervised video representation learning with odd-one-out networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
10. Girshick, R.B.: Fast R-CNN. CoRR (2015)
11. Gupta, S., Hoffman, J., Malik, J.: Cross modal distillation for supervision transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
12. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
13. Itseez: Open source computer vision library. `https://github.com/itseez/opencv` (2015)
14. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
15. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The kinetics human action video dataset. CoRR (2017)
16. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
17. Krähenbühl, P., Doersch, C., Donahue, J., Darrell, T.: Data-dependent initializations of convolutional neural networks. arXiv preprint arXiv:1511.06856 (2015)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems (2012)
19. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: a large video database for human motion recognition. In: Proceedings of the International Conference on Computer Vision (ICCV) (2011)

20. Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding. arXiv preprint arXiv:1703.04044 (2017)
21. Lee, H.Y., Huang, J.B., Singh, M.K., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: IEEE International Conference on Computer Vision (ICCV) (2017)
22. Luo, Z., Peng, B., Huang, D.A., Alahi, A., Fei-Fei, L.: Unsupervised learning of long-term motion dynamics for videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
23. Milbich, T., Bautista, M., Sutter, E., Ommer, B.: Unsupervised video understanding by reconciliation of posture similarities. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
24. Misra, I., Zitnick, C.L., Hebert, M.: Unsupervised learning using sequential verification for action recognition (2016)
25. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: ICML (2011)
26. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: IEEE European Conference on Computer Vision (ECCV) (2016)
27. Noroozi, M., Pirsiavash, H., Favaro, P.: Representation learning by learning to count. arXiv preprint arXiv:1708.06734 (2017)
28. Noroozi, M., Vinjimoor, A., Favaro, P., Pirsiavash, H.: Boosting self-supervised learning via knowledge transfer. arXiv preprint arXiv:1805.00385 (2018)
29. Paszke, A., Gross, S., Chintala, S., Chanan, G.: Pytorch (2017)
30. Purushwalkam, S., Gupta, A.: Pose from action: Unsupervised learning of pose features based on motion. arXiv preprint arXiv:1609.05420 (2016)
31. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) (2015)
32. Sanakoyeu, A., Bautista, M., Ommer, B.: Deep unsupervised learning of visual similarities. Pattern Recognition (2018)
33. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Conference on Neural Information Processing Systems (NIPS) (2014)
34. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
35. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014)
36. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research (2010)
37. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: Conference on Neural Information Processing Systems (NIPS) (2016)
38. Wang, L., Li, Y., Lazebnik, S.: Learning deep structure-preserving image-text embeddings. CoRR (2015)
39. Wang, X., Farhadi, A., Gupta, A.: Actions ˜ transformations. In: CVPR (2016)
40. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: Proceedings of the IEEE International Conference on Computer Vision (2015)
41. Yang, X., Ramesh, P., Chitta, R., Madhvanath, S., Bernal, E.A., Luo, J.: Deep multimodal representation learning from temporal data. CoRR (2017)
42. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European Conference on Computer Vision (2016)