Sparse-Dense Consistency-based Semi-Supervised Action Recognition

Ishan Rajendrakumar Dave, Jack Vice, Sabaina Haroon
University of Central Florida
4000 Central Florida Blvd, Orlando, FL 32816
ishandave,jack.vice,sabainaharoon@knights.ucf.edu

Abstract

In this paper we propose a self-supervised and semi-supervised consistency based approach to video representation learning which encourages the model to learn a full phase activity representation having both full video context as well as fine grain spatial and temporal feature representation. The human annotation of video for training action recognition requires an exorbitant amount of labor hours. Semi-supervised learning techniques use a combination of labeled and unlabeled data to achieve higher precision while reducing label data requirements. Typically, videos are temporally trimmed into clips or chunks for processing which reduces understanding of the full video context. Our method applies a consistency constraint between the spatio-temporal similarity of RGB space samples to that of embedded space samples, enabling the learning of full video context and fine grain features. We first trained for the unsupervised losses (self-supervised pretext tasks based on the spatio-temporal consistency of the densely and sparsely sampled clips). The trained model is evaluated for the activity classification as a downstream task and achieves a significant improvement in the classification over the randomly initialized model in the initial training phase. We use Mini-Kinetics dataset as unlabeled data and we tested our data on UCF101 validation set.

1. Introduction

The work of Krizhevsky *et al.* [14] showed the Convolution Neural Network (CNN) to be a compelling solution to image classification. Neural network classifiers like CNN's require the training of numerous parameters. The RESNET-50 architecture [11] for example, has 23 million trainable parameters. Large datasets such as video, typically require individual image annotations as ground truth for training deep learning methods of classification. Image annotation is a labor intensive effort and often contains some amount of noise due to human imprecision. As an example labor estimate, if an annotation takes about 10 seconds per video clip, the Kinetics video action dataset [5] with 650,000 labeled video's, represents approximately 1,805 labor hours. Many efforts have sought to mitigate

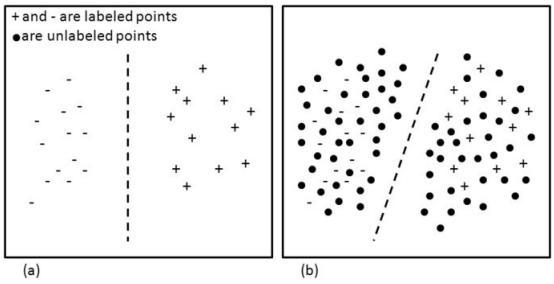

Figure 1. Semi-Supervised Learning

the need for human labeled data with methods such as weakly supervised learning, semi-supervised learning, self-supervised learning, and unsupervised learning. Weakly supervised learning [10] [3] [28] [2] uses data which has weak labels such that each image has one or more labels but no localization bounding boxes. With semi-supervised learning [12] [15] [20] [22] [21] [33], a combination of supervised and unsupervised learning is done using a labeled dataset and unlabeled dataset, respectively. In self-supervised learning [17] [30] [25] [1] [7] [6] [4] [29] [13] [23], supervisor signals for the training dataset are generated by extracting contextual or correlational information and is considered a special case of supervised learning. Unsupervised learning [32] [16] [18] looks for patterns in the dataset and models probability densities over the input samples.

Many machine vision applications have an abundance of unlabeled imagery and far fewer labeled images. Taking advantage of this unlabeled data combined with labeled data to improve training, is the primary goal of semi-supervised learning. As seen in Figure 1, with binary labeled data we discover a classification plane *(a)* which separates the two classes. On the right *(b)*, we have both labeled and unlabeled data and we must find a plane that separates the two classes. The two basic types of semi-supervised training are

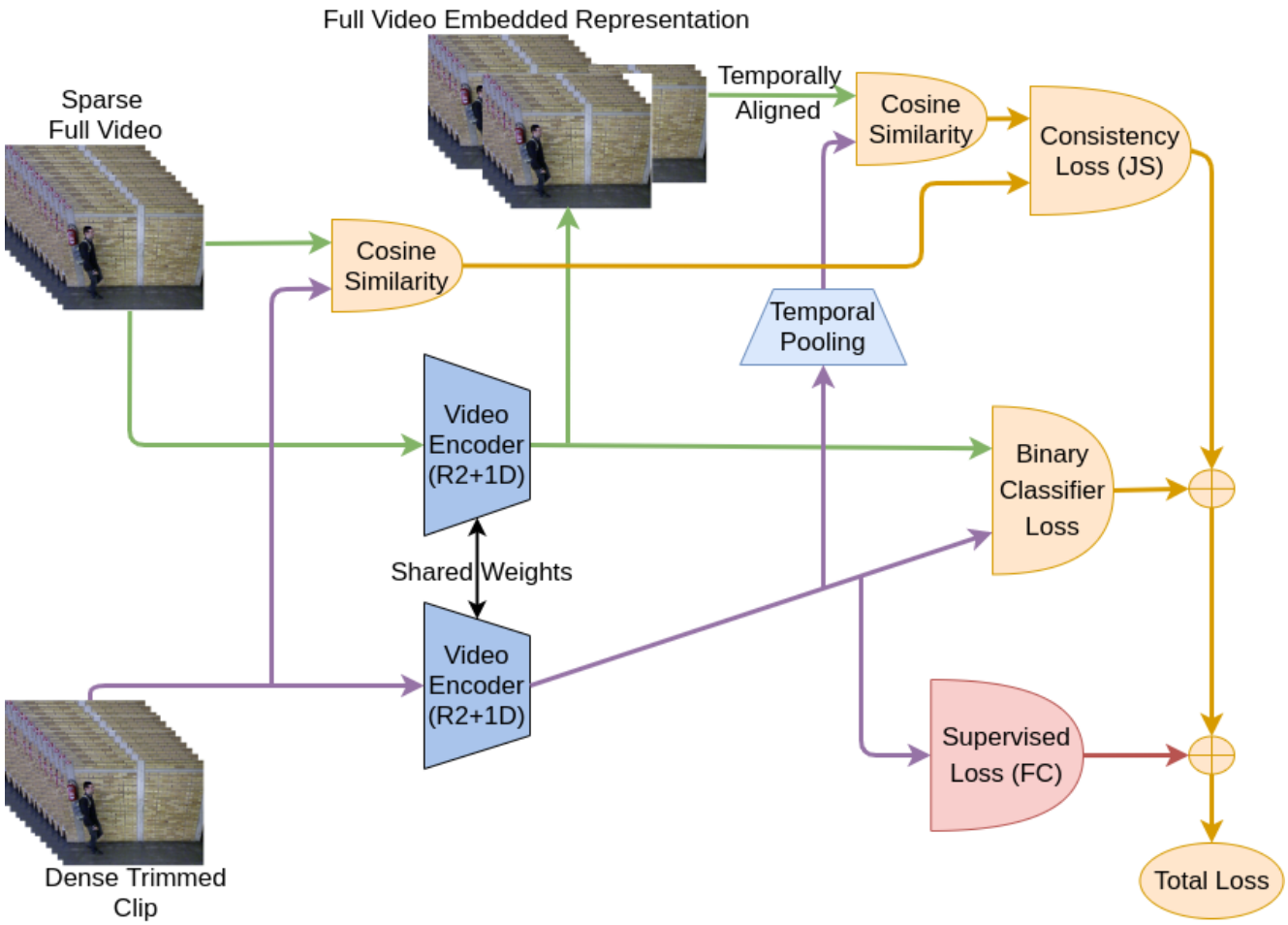

Figure 2. Sparse-Dense consistency Framework

Self-training methods and consistency regularization. The semi-supervised method [33] first trains on labeled data and then does inference on a set of unlabeled data. A prediction threshold allows for instances in which the prediction was above the threshold to be used as training data in a new training cycle and the process is repeated. The performance of self training depends on the data distribution, the prediction threshold value and the number of samples and as such, learning performance can decrease in some cases. Consistency-regularization methods [7] [15] [12] induce random perturbations to the input images and use the difference between the output predictions and the perturbed input as a loss to be minimized.

When designing a machine learning framework, taking into account the hardware capabilities on which the framework will be tested is beneficial and can play a significant role in real world utility. The graphics processing unit on machine learning clusters typically have a specific memory capacity and thus, can only process a specific number of bytes. In the real-time system, we can fit only a limited number of frames in the classifier due to this GPU memory limit. We can either use sparsely sampled frames which has more temporal span or densely sampled frames with fine-grain temporal and spatial information. There are some challenges with using full video sparse sampled frames as it requires large buffer storage, high I/O time, and it might produce less recall because it might miss some activity in between the skipped frames. Therefore, a chunk-based classifier is preferred which requires densely sampled frames from the recent timestamp. However, the chunk-based classifier lacks understanding of the long temporal context which might confuse the classifier, reducing accuracy. In this work, we want to encourage a classifier to learn activity phase understanding, by providing both sparsely sampled clips from the full video which we refer to as the ”Sparse clips” and a densely sampled, temporally trimmed clips which we refer to as the ”Dense clips”.

This paper proposes an efficient Sparse-Dense Consistency based Semi-supervised learning for Action Classification (SD-CSD) which encourages consistency between

sparse and dense representations of an action video. Dense clips are densely sampled (high frame rate), temporally trimmed clips and Sparse clips are sampled at a constant interval dictated by the total frames in the video and ensuring that each Sparse clip spans the entire action video. Our method is informed by the work of Feichtenhofer *et al.* [8] who developed a SlowFast network which conceptually resembles the biological visual processing of motion by primates [9] [27]. Our method enables action classifier training with unlabeled video datasets using the consistency loss between a densely sampled temporally trimmed video clip with the corresponding sparsely sampled section of the full video. The idea is to learn a full action context representation for the entire video clip which also has the fine grained spatial and temporal features most important to accurate classification. The representations from latent encoded space and RGB space for each dense clip and sparse clip, are aligned together for calculating spatio-temporal consistency loss using cosine similarity and Jensen-Shannon divergence. Additional training loss are a binary cross entropy loss between the Dense clip and the Sparse clip representations and finally supervised classification when training with the labeled data. Our proposed framework is tested on the UCF101 dataset [24] which is composed of 101 classes of action videos. Experimental results show our framework outperforms the randomly initialized model. Model trained with our unsupervised losses consistently achieves significant improvement in classification performance on UCF101 in the initial training phase.

2. Background and related work

Unsupervised Learning. By drawing inferences from patterns in the dataset, unsupervised learning is able to learn embeddings for the underlying structure, distribution and correlations in the unlabeled data.

Ahsan et al [1] proposed an unsupervised learning approach to learn spatio-temporal context for video recognition by solving 'Video-Jigsaw' pretext task. Without making use of any optical flow based patches or heavy patch permutation tasks, they used random patches from the 2x2 grid of the video frame tuples, permute them to generate jigsaw puzzle using a novel computationally efficient permutation approach and learned spatio-temporal video representations by solving the jigsaw puzzles. They trained the model on Kinetics dataset without using any model and the using the trained model they showed results on fine-tuning task of action classification on UCF-101, HMDB datasets, also they have shown some interesting transfer learning tasks like Image classification and image retrieval tasks on PASCAL VOC 2007 dataset.

The ActionFlowNet proposed by Ng *et al.* [19] is an unsupervised multitasking CNN based framework which trains on raw pixels to simultaneously estimate optical flow as well as action recognition. The framework takes 16 consecutive frames and uses ResNet-18 as the backbone CNN and does concatenation of the frame representations in the 3D-ResNet layers to build the optical flow prediction output. By predicting the future optical flow of the last frame, the optical flow acts as the supervisor preventing detrimental forgetting with respect to action recognition.

Another unsupervised framework proposed by Li *et al.* [16] combines a method for prediction 3D motion as well as view-adversarial technique which is added to improve the learning of view-invariant features. The method learns view-invariant motion representations and can predict the sequence of motions for multiple views. Additionally, an adversarial discriminator encourages a view-invariant representation of the dynamics in the embedded space by using a view classifier to try to discriminate views.

Self-supervised learning. Self-supervised representation learning is a autonomous training method in which a surrogate task is used to learn feature representations indirectly based on extracting information from the training data distributions. Predicting the correct chronological order of shuffled frames as done in [4] [25] or video clips [30] [6]from an action video can help the model learn spatio-temporal features.

The method proposed by Xu *et al.* [30] shows that because clips contain both spatial and temporal features, learning the correct order of clips improves self-supervised learning over frame order prediction. Non overlapping clips are sampled from the video and shuffled. 3D-CNN's are used to extract the features which are then pairwise concatenated and processed by fully connected layers for order prediction.

Another work [25] that uses correct frame order prediction as the surrogate task, attempts to alleviate the need for a motion feature like optical flow in spatiotempral action learning. In this self supervised approach, multiple branches of a 3D-CNN each take a video clip of either correct ordered frame or randomly shuffled frames and then the branch outputs are concatenated and passed to a fully connected Fusion Net with softmax output and cross entropy loss to train detection of the random ordered clip versus the correct order clips.

In the work done by Buchler *et al.* [4], a Reinforcement Learning (RL) algorithm is proposed to learn a policy for conducting permutations which will improve learning efficiency beyond that of a random permutation. The RL algorithm wraps around a typical backpropagation based CNN and as the CNN learns to infer the correct order permutation for both temporal (frame shuffle) and spatial (jigsaw) permutations, a policy is learned to conduct permutations best suited for learning for the next iteration based on the previous error. The reinforcement learner continually adapts the policy to the changing state of the CNN such that what

might have been a well suited temporal permutation early on, may not be good later in the learning process.

Building on the notion of temporal coherence, Cho el al. [6] design a method for self-supervised action recognition based on temporal group normalization. Multiple video action clips are various playback speeds are frame shuffled and then the model must predict both the correct order and the correct playback speed clip in which each frame belongs. The framework is tested on C3D, R3D and R(2+1)D. Initially, at random starting points in each video clip, frames are sampled and various speeds and in forward and reverse direction(eg. -2x, 1x, 4x). Next, the frames are shuffled and fed into the 3D CNN model which uses layer-dependable temporal group normalization to do feature extraction. Temporal features are divided into multiple groups with each group being normalized by specific appropriate parameters. Features are then concatenated pairwise for passing to the fully connected layers which predict playback speed and frame order.

Another self-supervised learning approach [7] uses temporal correspondences to learn to predict the temporal alignment between videos. Temporal cycle consistency (TCC) is used to learn correspondences over time and across multiple videos. Using 3D convolutions a pair of video's is then represented in embedded space. Using nearest neighbor to match the points in the two videos, the loss encourages the closest neighbor relationship to be mutual, thus the cycle is consistent. This method enables few-shot classification of action phases and achieves performance on par with fully supervised learning methods and can be applied to many temporal alignment tasks.

Lorre *et al.* [17] propose a Contrastive Predictive Coding method for self-supervised video representations which trains on and predicts high level frame information. the framework is trained on multiple video clips which are processed frame by frame by a CNN. In order to aggregate past information, the resulting feature maps from the CNN is passed to an autoregressive model. Future video segments are predicted from context produced by the autoregressor and loss is then calculated from the prediction and target using Noise-Contrastive Estimation (NCE).

In work done by Sermant *et al.* [23], multiple simultaneous video streams from different viewpoints of the same activity are combined and used to learn the action. The model learns attributes that change over time in one viewpoint and attributes that are the same across multiple viewpoints. The method is used to enable a robot arm to reproduce the action task in a self supervised manner. Using triplet loss, an embedding is learned where the information from frames from different viewpoints are consolidated in embedding space thus achieving viewpoint invariance. Likewise, attributes which are temporarily close to each and also visually similar in a give viewpoint are repelled from each other in embedding space thus extracting only the difference over time for that viewpoint.

Using a self supervised learning Kim *et al.* [13] train a 3D-CNN with Space-Time Cubic Puzzles to learn to recognize the action and chronology of a single image taken from a human action video sequence. The method first extracts random 3D spatial-temporal crops from video clips and then trains the 3D CNN to predict the ground truth spatial and temporal order and outperforms 2D CNN's on UCF101 and HMDB51.

Another "puzzle" method which is designed to improve convolution 3D models, [29] shows a new technique which improves video action classification with self-supervised spatio-temporal representation learning. Taking individual frames from an action video, each is divided into various grid patterns and statistical label predictions are made using regression, for each patch in the grid based on the characteristics of the motion and stability or divergence of the colors.

Semi-supervised learning. Next we will look at the consistency based semi-supervised methods developed by Jeong et al [12]. In this work, the authors propose using consistency constraints to enable the use of unlabeled data combined with labeled data during training. The consistency loss is applied to both labeled and unlabeled and is formed by the horizontal flipping of the image and corresponding object bounding boxes. There is a consistency loss method for single stage detectors as well as for two stage detectors. For the single stage detector the bounding box annotation provides a supervised loss and Jensen–Shannon (JS) divergence based consistency loss is computed between the predicted bounding box in the training image and it's flipped counterpart for both labeled and unlabeled data. For a two stage detector, the flipped image is processed through the backbone network and then the RPN predicted bounding box is flipped and applied to the flipped image features and passed to the classifier where the JS consistency loss is calculated between both the original and flipped classifier outputs.

The method proposed by Lain and Aila [15] in 2017 introduces an self-ensembling method for semi-supervised learning. Unknown labels are inferenced by consensus prediction of an ensemble of a single network on different training epochs using varying input augmentations and regularizations. The temporal ensemble uses predictions from multiple dropout conditions as well as over multiple epochs.

Multi-method approaches. Similar to our approach of combining methods, Zhai *et Al.* [31] developed a combined self-Supervised and semi-supervised method. The authors propose two techniques based on the idea that self-supervised methods can be improved by taking advantage of a small amount of labeled data, and in doing so, improve the overall performance such that it is a practical alternative to fully supervised solutions. The two techniques

utilized for testing the self-supervised component of the framework are rotation and exemplar. Rotation methods in un-supervised learning augment the input images by a random rotation transformation and then the model is trained to predict the magnitude of the rotation. Exemplar techniques train a model to prediction transformations such as inception cropping, mirroring and colorspace randomization which encourages the model to learn image representations which are invariant to image augmentations.

SlowFast Network. The research most informative to our work was the SlowFast network for video recognition method proposed by Feichtenhofer *et al.* [8] where spatial and temporal information are learned by combining low and high framed rate inputs. This single stream method uses a fast network for temporal information and a slow network for spatial information similar to the retinal ganglion cells in the visual system of primates [9] [27]. To achieve high temporal resolution, the fast network samples at eight frames for each single frame the slow network. To maintain high spatial resolution, the slow network was tested with eight times the number of channels as the fast network with the overall objective being to capture as much relevant temporal and spatial information as possible at a given computational cost. Using two convolution models in parallel, lateral connections between the two pathways are intended to fuse the information of both. Feature representations from the fast pathways pass over the unidirectional lateral connection and undergo a dimensionality transformation to match the slow pathway thus enabling fusing the features into the slow pathway by concatenation. The three fusion methods tested were time-to-channel, time-strided sampling and time-strided convolution with time-strided convolution showing the best performance. Experiments are conducted on the Kinetics-400, Kinetics-600 and Charades datasets showing precision improvements over eight other methods on Kinetics-400, over two other methods on Kinetics-600 and improvements over five other methods for Charades.

3. Method

In this section we present our combined unsupervised and semi-supervised learning method. The semi-supervised learning approach for the video classification problem is formally defined as taking a joint distribution *p(X,Y)* over videos and labels and then training a classifier with a labeled training set D_l sampled i.i.d from *p(X,Y)* and an unlableled training set D_u sampled i.i.d from the marginal distribution *p(X)*. Defining $\mathcal{L}_l$ as our cross-entropy classification loss for labeled videos, $\mathcal{L}_u$ as our unsupervised loss, *w* as our scalar weight, and θ as the parameters for our model, our learning objective is then:

$$\min_{\theta} \mathcal{L}_l(D_l, \theta) + w\mathcal{L}_u(D_u, \theta) \tag{1}$$

Similar to [8], for each video in the dataset, our framework trains on both sparsely sampled frames from the entire video referred to as the Sparse clip and also temporally trimmed but densely sampled frames referred to as the Dense clip with both clips being of identical dimensions. For the full video sparse sampling, a start frame is randomly selected from the set of possible start frames. So, for example, if we are sampling 10 frames out of 100, our start frame would be randomly selected from the range of zero to nine inclusively. Sampling frames from the full video extracts *N* number of frames sampled at a constant interval and then stacked into the Sparse clip corresponding to the entire video and thus capturing the full temporal context of the video, albeit at relatively low temporal and spacial fidelity. The Dense clip is also randomly selected from the full video and will have the same *N* number of frames thus the same input dimensions as the Sparse clip. We then alternate between randomly sampled Sparse and Dense clips as training input to the model.

Referring to Figure 2, The two input clips are alternately fed to a R(2+1)D video encoders with weights shared between the two. Starting in the pre-processed RGB space, the sparse clip is further segmented into four intervals and cosine similarity [4] [6] is calculated between Dense clip and the four Sparse intervals. After video processing and temporal pooling, a cosine similarity is then calculated for the Dense clip representation and the four interval representations corresponding to the four RGB Sparse intervals. Two sets of four similarity values with each set being used to calculate consistency loss using Jensen-Shannon divergence similar to [12].

We additionally implement a binary classifier loss, as shown in Figure 2, which further encourages the model to learn a different representation for the Dense clip versus the Sparse clip, an inversed binary version of the view classifier in [16]. The Dense clip and the Sparse clip are the same number of frames of the same action class but each contain different but not mutually exclusive information relevant to action classification. Indeed, the Dense clip contains more fine grain spatial and temporal information while the Sparse clip contains more context information about the full video. The model must learn which information it is processing and how to best represent that information for action classification. This 2-class pretext classification task learns spatiotemporal context of a video in depth using two classes which are densely sampled class and sparsely sampled class. We use standard binary cross entropy loss and denote our binary loss as $\mathcal{L}_{bin}$.

The Dense clip representation which had been pooled will contain substantially more fine grain spatial and temporal information compared to the corresponding Sparse clip representation for the given time interval.

As shown in figure 2, the consistency loss is then calculated between these two representations of the same time

interval. For the supervised loss, shown in Figure 2, when using labeled training data, we take the temporally trimmed representation from the video encoder and pass it to fully connected layers for standard cross entropy loss. The third loss during supervised training, is the binary classification loss between the trimmed clip and full clip outputs from the video encoder. During unsupervised training, we just have the consistency and binary losses.

We use the cosine similarity between Sparse and Dense clips for both RGB and embedded representations. This results in calculating four similarity values for each Dense clip. We denote D_r as the RGB Dense clip and D_e for the post pooling Dense clip embedded representation. For k Sparse intervals, we have S_r^k for the k^{th} sparse RGB interval and S_e^k for the k^{th} embedded Sparse interval representation. The k^{th} RGB cosine similarity denoted C_r^k is defined as:

$$C_r^k(D_r, S_r^k) = \frac{D_r \cdot S_r^k}{\|D_r\| \, \|S_r^k\|} \tag{2}$$

The k^{th} embedded representation cosine similarity denoted C_e^k is thus:

$$C_e^k(D_e, S_e^k) = \frac{D_e \cdot S_e^k}{\|D_e\| \, \|S_e^k\|} \tag{3}$$

Consistency loss is then calculate from k interval cosine similarity values using Shannon-Jenson divergence

$$\mathcal{L}_{con} = JS(C_r^k(D_r, S_r^k), C_e^k(D_e, S_e^k)) \tag{4}$$

The final consistency loss $\mathcal{L}_{con}$ is multiplied by a weight scheduling function $w(t)$ and added to our unsupervised binary loss $\mathcal{L}_{bin}$ and our supervised classification loss$\mathcal{L}_{cls}$ to get our final loss $\mathcal{L}$ as:

$$\mathcal{L} = w(t) \cdot \mathcal{L}_{con} + \mathcal{L}_{bin} + \mathcal{L}_{cls} \tag{5}$$

The backbone convolution network and fully connected layers used in our implementation are based on the R(2+1)D [26] classifier shown in Figure 3. R(2+1)D is a convolutional block composed of 2D spatial convolutions and 1D temporal convolutions. The idea is to factorize the 3D convolution into these separate spatial and temporal layers which when compared to a 3D convolution, has the effect of doubling the number of non-linearities with the same number of parameters. Thus, R(2+1)D convolutions can approximate much more complex functions with little impact on training efficiency. Indeed, R(2+1)D has show state-of-art performance on action video datasets such as Kinetics-400 and UCF101 datasets.

For our implementation experiment, our framework first processes on video clips of 16 frames of 112 x 112 as represented by the Dense Trimmed Clip in the lower left of the Figure 2. Thus, the input to the is 16 x 3 x 112 x 112 and the output is 4 x 512 x 7 x 7. After temporal pooling the representation dimension are 512 x 7 x 7.

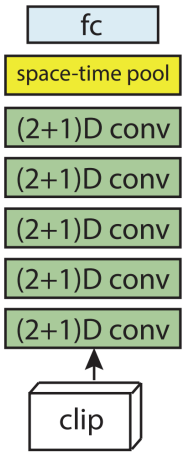

Figure 3. R(2+1)D [26]

4. Experiments

Dataset.UCF101 is a video action dataset [24] which consists of 101 labeled action classes in over 13 thousand unconstrained video clips. The video were recorded in various environments with a range of lighting conditions, frame rates and camera motion. The 101 action classes are divided into the following five types of actions: Human-Human interaction, Human-Object Interaction, Sports, Body-Motion Only and Playing Musical Instruments. Figure 4 shows a small snapshot of some of the classes.

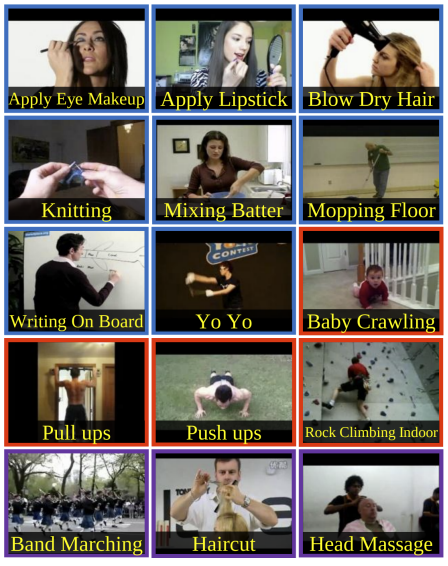

Figure 4. Example of UCF-101 action classes

Training. We train our models from scratch with random initialization and no pre-training. The details of the dataset splits is shown in Table-1. We are dropping 2120 short clips (less than 100 frames) UCF101 training clips, to avoid using very short temporal span in the densely sampled

clip and maintaining the skip rate ratio (at least 3.5) of the densely sampled and sparsely sampled clips.

The framework is first trained on unlabeled data for unsupervised learning. During this period, the models learn the full video context spatio-temperal representations based on the consistency and binary losses. Next the framework is trained on the labeled data with the supervised loss plus the other two losses.

Loss Weight Scheduling We train our network by applying weight scheduling scheme which focuses on self supervision and semi supervised task for starting one third time of the total training process. For a total of 150 epochs, supervised classification starts after 50th epoch taking in input, pre-trained weights from self supervised and semi-supervised classificaiton task.

Validation. Table 2. shows validation results for five different experiments. Proposed method (JSD) calculates semi-supervised loss using the same process as defined in the paper where cosine similarity is calculated between sparse and dense latent space representations and RGB space representations and afterwards their consistency is observed using divergence loss. Proposed method (Cosine) differs from JSD such that instead of calculating divergence, it calculates loss from cosine similarity between representations. In our proposed methods using weight scheduling we make sure to use pre-trained weights from unlabeled data for supervised classification, whereas in Random (1,2,3) experiments we feed randomly initialized weights to network, without any pre-processed learning of unlabeled data. These results are after first epoch and can be taken as a proof of concept for the proposed idea. Keeping in view the time constraint associated with this work and the nature of labeled and unlabeled dataset, our final training results are yet to be completed.

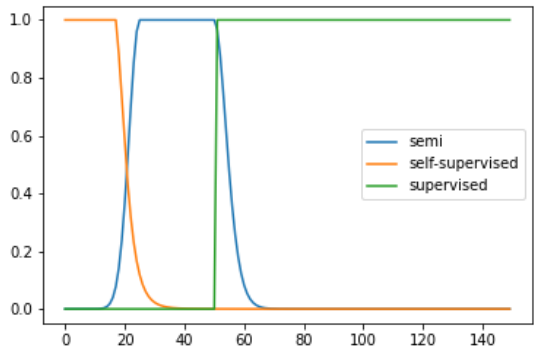

Figure 5. Weight Scheduling where network is trained for self supervised task for first 20 epochs, semi-supervised task for next 40 epochs and supervised task afterwards.

Split	Number of samples	Number of Classes	Dataset
Training (Labeled)	**7417**	101	UCF101
Training (Unlabeled)	**36494**	-	Mini-Kinetics
Validation	3783	101	UCF101

Table 1. Labeled, Unlabeled and Validation split

Initialization method	F1-Score (%)	Validation Accuracy (%)
Proposed method (JSD)	0.09347	1.428
Proposed method (Cosine)	**0.21426**	**1.587**
Random-1	0.02105	1.058
Random-2	0.02533	1.296
Random-3	0.02073	1.058
Random-4	0.01459	0.741
Random-5	0.02329	1.190

Table 2. Validation accuracy at of the epoch-0 with randomly initialized and proposed method

5. Discussion

From our experimental data, it can be seen that both of our proposed unsupervised losses, from Figure 8 and 9, are decreasing overtime. Whereas an interesting insight can be inferred from Figure 10. where results from our proposed method, using one minus Cosine similarity as unsupervised loss outperforms all other experiments despite the fact that loss is not learning. This insight is consistent with findings from [1], where unsupervised loss does not necessarily correlate with network performance, by not decreasing over time but is still contributing towards final accuracy by managing to learn better feature representations using unlabeled data for supervised learning task.

As a framework modification for future work, a generator network could be added to the framework and the temporal pooling removed. The generator network would be used on the post R(2+1)D sparse representation so as to match the dense clip representation dimensions for the given time interval. The consistency loss would then be calculated between each dense representation and the corresponding interval of the generator output. The batch size will have to equal the number of dense clips required to span the entire video such that we achieve a loss for every output in the generator before we can do back-propagation. A binary loss would not be necessary as the generator would be encouraged to predict the fine grain spatial and temporal features for the entire video on unlabeled data and then the labeled data would be used to teach the framework the classes for the embedded features.

6. Conclusion

In this paper we presented a unsupervised consistency and semi-supervised based approach to video representation learning which encourages the model to learn a full phase activity representation having both high temporal fidelity as well as fine grain spatial features. Using a unsupervised consistency loss combined with a binary loss and a supervised loss, our work show improves the activity classification performance for UCF101 dataset. By learning a full action context with fine grained spatial and temporal feature representation combined with efficiency of processing, this method should be appealing for application to many real world video representation learning tasks.

References

[1] Unaiza Ahsan, Rishi Madhok, and Irfan Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 179–189. IEEE, 2019.

[2] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Improving consistency-based semi-supervised learning with weight averaging. *arXiv preprint arXiv:1806.05594*, 2, 2018.

[3] Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars. Weakly supervised object detection with convex clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1081–1089, 2015.

[4] Uta Buchler, Biagio Brattoli, and Bjorn Ommer. Improving spatiotemporal self-supervision by deep reinforcement learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 770–786, 2018.

[5] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019.

[6] Hyeon Cho, Taehoon Kim, Hyung Jin Chang, and Wonjun Hwang. Self-supervised spatio-temporal representation learning using variable playback speed prediction. *arXiv preprint arXiv:2003.02692*, 2020.

[7] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1801–1810, 2019.

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019.

[9] Daniel J Felleman and DC Essen Van. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991.

[10] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Multi-fold mil training for weakly supervised object localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2409–2416, 2014.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. In *Advances in Neural Information Processing Systems*, pages 10758–10767, 2019.

[13] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8545–8552, 2019.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[16] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan Kankanhalli. Unsupervised learning of view-invariant action representations. In *Advances in Neural Information Processing Systems*, pages 1254–1264, 2018.

[17] Guillaume LORRE, Jaonary Rabarisoa, Astrid Orcesi, Samia Ainouz, and Stephane Canu. Temporal contrastive pretraining for video action recognition. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 662–670, 2020.

[18] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016.

[19] Joe Yue-Hei Ng, Jonghyun Choi, Jan Neumann, and Larry S Davis. Actionflownet: Learning motion representation for action recognition. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1616–1624. IEEE, 2018.

[20] Nhu-Van Nguyen, Christophe Rigaud, and Jean-Christophe Burie. Semi-supervised object detection with unlabeled data. In *international conference on computer vision theory and applications*, 2019.

[21] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018.

[22] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019.

[23] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141. IEEE, 2018.

[24] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[25] Tomoyuki Suzuki, Takahiro Itazuri, Kensho Hara, and Hirokatsu Kataoka. Learning spatiotemporal 3d convolution with video order self-supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[26] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[27] David C Van Essen, Jack L Gallant, et al. Neural mechanisms of form and motion processing in the primate visual system. *Neuron*, 13(1):1–10, 1994.

[28] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. C-mil: Continuation multiple instance learning for weakly supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2199–2208, 2019.

[29] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4006–4015, 2019.

[30] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019.

[31] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE international conference on computer vision*, pages 1476–1485, 2019.

[32] Chunjie Zhang, Jian Cheng, and Qi Tian. Unsupervised and semi-supervised image classification with weak semantic consistency. *IEEE Transactions on Multimedia*, 21(10):2482–2491, 2019.

[33] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

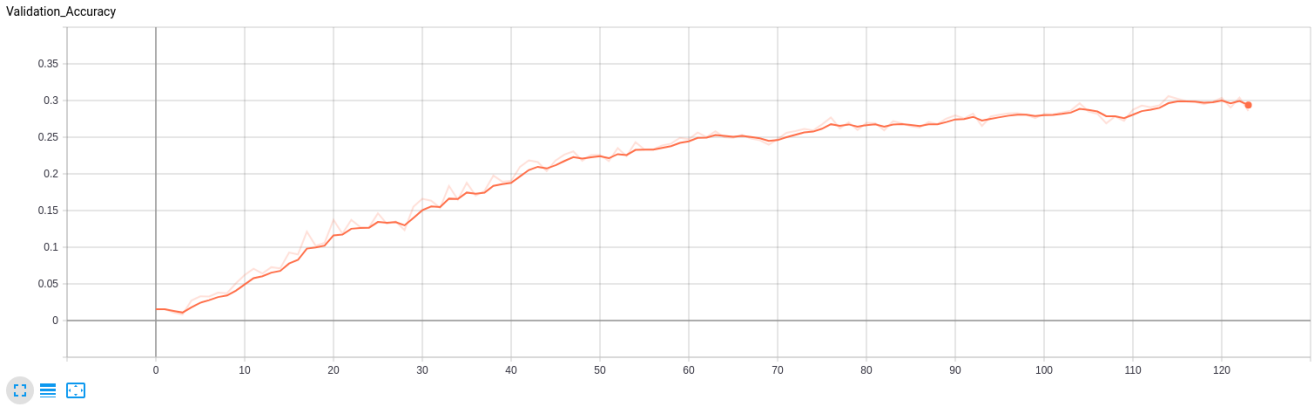

Figure 6. Baseline Validation Accuracy

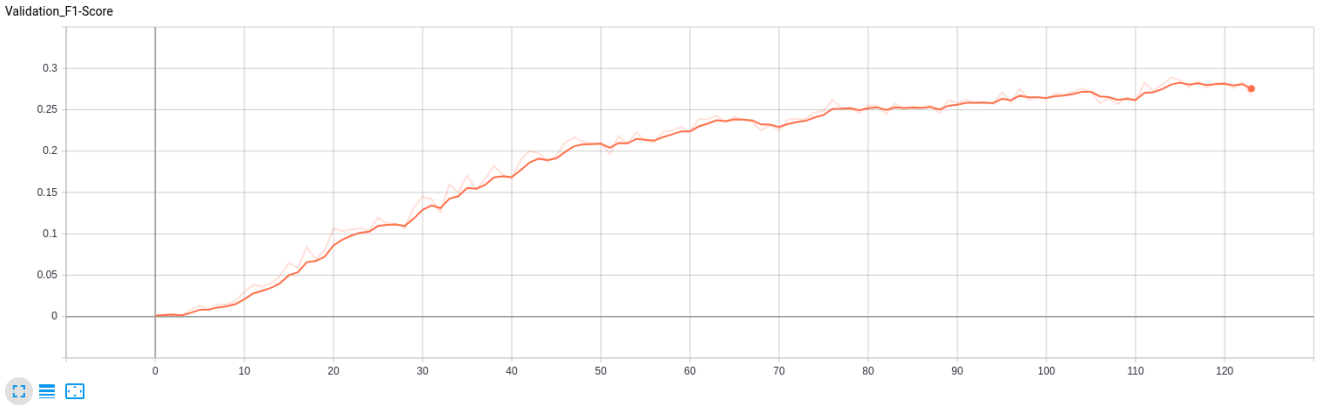

Figure 7. Baseline Validation F1-Score (Classwise Average)

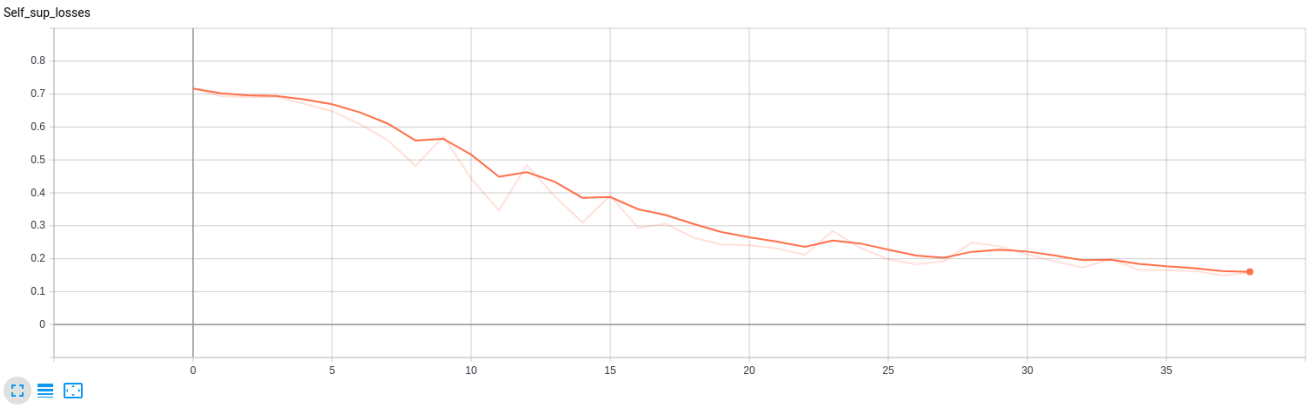

Figure 8. Self supervised loss (training phase)

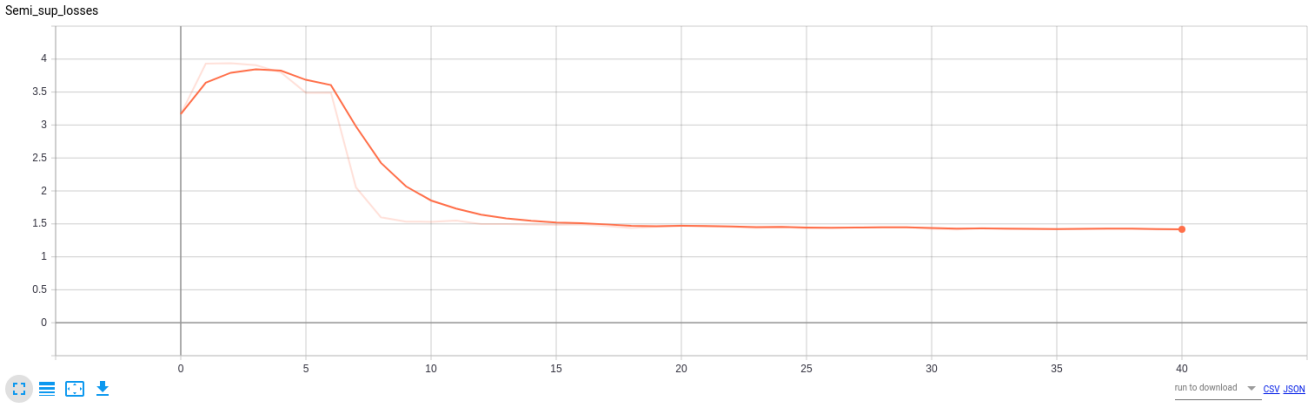

Figure 9. Semi supervised loss (JSD) (training phase)

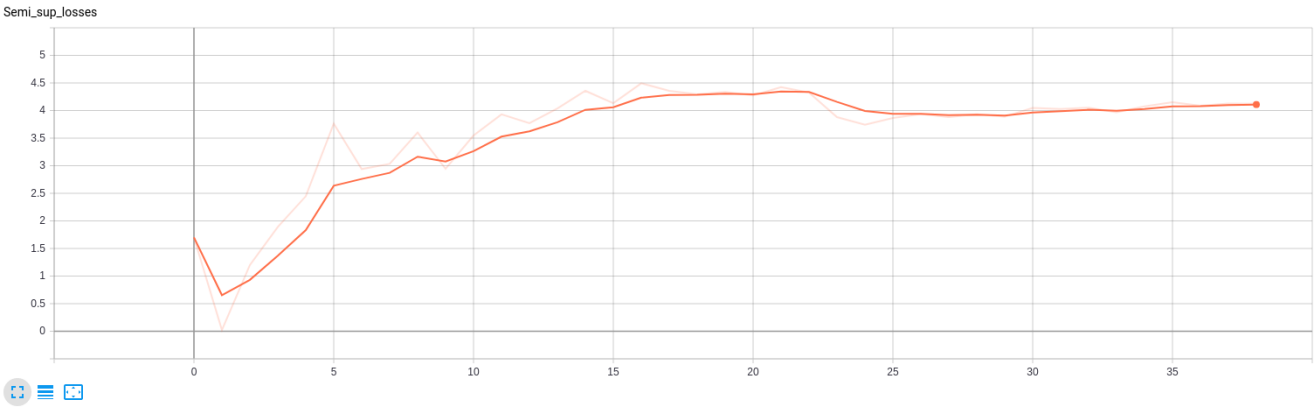

Figure 10. Semi supervised loss (Cosine Similarity) (training phase)