CS-E4875
Semi-Supervised Video Action Recognition

Jaakko Kainulainen

April 19, 2023

Abstract

Semi-supervised learning (SSL) is an approach used to leverage unlabeled data to improve the accuracy of models. Although several SSL methods have been used to improve the performance of image classification models, they have not been used widely in video classification tasks. Transformer-based models have achieved state-of-the-art results in video classification tasks, but these models require a lot of data to train. In this paper, we trained a Video Vision Transformer model using a FixMatch SSL algorithm on a mini-Something-Somethingv2 dataset. We show that the model trained using SSL can slightly outperform the model trained with supervised learning if 30 % of videos are labeled. Since the performance improvement is small, a transformer-based video classification model might require a more complex SSL algorithm. Although using SSL does not improve the accuracy much, the results suggest that SSL can be used in video classification tasks to improve the results when only a small amount of training data is labeled.

1 Introduction

Until recently, deep convolutional neural networks have been the dominant architecture in machine vision. In natural language processing (NLP), state-of-the-art architectures utilize pure transformer [1] based models. Inspired by the NLP architectures, transformers have been integrated into convolutional network architectures, and transformer-based models have been implemented for vision classification. Recently pure transformer-based image recognition models [2] have outperformed their convolutional neural network counterparts, and using a similar architecture, Arnab et al. [3] achieved state-of-the-art results in video classification. These results were achieved using supervised learning, which requires large labeled datasets. Labeling datasets for vision problems is often done manually and, thus, can be expensive and time-consuming. The labeled video datasets are often smaller than the datasets used in image classification. Dostovitskiy et al. observed that these pure transformer-based networks require more data [2] than the convolutional networks, and even though Arnab et al. [3] managed to achieve great results on comparatively smaller video datasets with regularization and leveraging pre-trained image models, supervised training requires large enough high-quality video datasets.

Semi-supervised learning (SSL) is an approach for training models that requires less labeled data. It utilizes unlabeled data to increase the accuracy of the models, and therefore the training requires smaller labeled dataset. A popular class of methods used with SSL produces artificial labels for the unlabeled data, and the model is trained to predict the artificial label for the unlabeled data. Fixmatch [4] is an SSL algorithm that combines two methods that produce artificial labels, pseudo-labeling [5], and consistency regularization [6, 7, 8]. The algorithm produces artificial labels for weakly augmented images and then uses the prediction as the target class of a strongly augmented version of the same image if the model is confident in the artificial label. FixMatch managed to achieve state-of-the-art results on multiple commonly studied SSL benchmarks, and recently Jing et al. achieved promising accuracies with a CNN-based video classification model using SSL [9].

In this paper, we implement a vision-transformer model proposed in [3] and train it using both supervised learning and SSL. FixMatch algorithm [4] adapted to video data is used for the SSL.

Dataset: https://developer.qualcomm.com/software/ai-datasets/something-something
Dataloader adapted from: https://github.com/IBM/action-recognition-pytorch
FixMatch adapted from: https://github.com/kekmodel/FixMatch-pytorch

2 Related Work

SSL algorithms utilize several different methods to leverage unlabeled data. We concentrate on the methods used in FixMatch.

Consistency regularization is a common SSL method that enforces that the model output stays the same even if the input is perturbed. This approach was first proposed in [6], and was later referred to as "Regularization With Stochastic Transformations and Perturbations" in [8], and as Π-Model in [7]. The most common perturbation method is to apply data augmentations to the input data [6, 8, 7, 10], but other methods, including adversarial perturbations [11] and stochastic regularization [12], are also used. Using strong data augmentations has been shown to improve the results of consistency regularization [10, 13].

Pseudo-labeling [5], a variant of self-training [14, 15, 8], is an SSL method, that uses the model to generate labels for the unlabeled images. The generated labels with a higher probability than a predefined confidence threshold are used in the training.

Several SSL algorithms, such as ReMixMatch [13], UDA [10], and FixMatch [4] have combined these methods to increase the accuracy of the models. These algorithms use weakly augmented images to generate the pseudo-labels and then enforce the model consistency against a strongly augmented version of the same image. To ensure the quality of the pseudo-labels FixMatch and UDA both use fixed thresholds, which can lead to low data utilization. Recent algorithms have shown that a dynamic [16, 17] or a class-specific threshold [18] can increase the data utilization and improve the results.

Transformers proposed by Vaswani et al. [1] have become the state-of-the-art method in NLP architectures [19, 20]. The transformer replaced convolution and recurrent networks with a network consisting of self-attention, layer normalization, and multilayer perception operations. Pure transformed-based models have recently achieved state-of-the-art results in image classification tasks [2]. Inspired by the results Arnab et al. [3] used a similar architecture to achieve great results on video classification tasks.

Even though the SSL algorithms have achieved great results on image classification tasks, semi-supervised algorithms for video classification have not received much attention. Recently pseudo-label-based SSL algorithms have achieved impressive performances with a small amount of labeled data on video classification tasks [9, 21]. These works were trained on CNN-based models. Even though pure transformer-based models have recently outperformed their CNN-based counterparts using supervised learning[3, 22], to the best of my knowledge, SSL algorithms have not been used with transformer-based video classification models.

3 Video Vision transformer

This section summarizes the ViViT-model [3] used in this paper. We first explain a tubelet embedding method used to extract tokens for video in Sec 3.1. Then, we give the details of the transformer layout used in the model in Sec. 3.2 and the model architecture in Sec 3.3. Finally, we go through the methods used to initialize the model with a pre-trained image classification model In Sec 3.4.

3.1 Tubelet embedding

Arnab et al. [3] suggested two methods for embedding the video into tokens. Since tubelet embedding outperformed patch embedding in [3], we use tubelet embedding, and the rest of section 3 will only consider an input embedded with this method.

The tubelet embedding extracts non-overlapping tubelets of shape $t \times h \times w$ from the video. From a video with length T, and frame size $H \times W$, $n_t \times n_h \times n_w$ tubelets are extracted, where $n_h = \frac{T}{nt}$, $n_h = \frac{H}{h}$, $n_w = \frac{W}{w}$. The frames are sampled with stride n_t, to get non-overlapping tubelets.

The embedding maps a video $V \in R^{T \times H \times W \times C}$ to the sequence of tokens $\tilde{z} \in R^{n_t \cdot n_h \cdot n_w \times d}$. A classification token is prepended and positional embedding is added to $\tilde{z}$, which is then reshaped to $R^{N \times d}$ to get the transformer input z.

3.2 Transformer layout

ViViT [3] adapts Vision Transformer [2] architecture to process video data instead of image data. The videos are mapped to non-overlapping tubelets. These are flattened and mapped to 1D by linear projection. A class token z_{cls} is prepended to the sequence of input tokens and 1D positional embeddings are added to all the inputs to retain the positional information. The transformer encoder consists of L layers, and each layer consists of MultiHeaded-Self Attention (Eq. 2) and MLP blocks (Eq. 3), and Layernorm is applied before each block. The MLP block contains two linear projections separated by GELU non-linearity. The prepended class token z_{cls} is extracted from the output, and forwarded through a linear classifier to classify the input.

With 2D image patches, the ViViT architecture is identical to the ViT architecture. With 3D tubelets, the linear projection with E is equivalent to a 3D convolution instead of a 2D convolution.

$$z = [z_{cls}, Ex_1, Ex_2, \ldots, Ex_N] + p, \qquad E \in R^{t \times h \cdot w \times d}, p \in R^{(N+1) \times d} \tag{1}$$

$$y^l = MSA(LN(z^l)) + z^l \tag{2}$$

$$z^{(l+1)} = MLP(LN(y^l)) + y^l \tag{3}$$

$$\tag{4}$$

3.3 Transformer Model for Video

We implemented the Factorized Encoder ViViT-model [3]. The model consists of two separate transformer encoders, spatial and temporal encoders. The spatial encoder models interactions between tubelets extracted from the same temporal index, and the temporal encoder models the interactions between the temporal index representations extracted from the outputs of the spatial encoder. A spatial classification token z_{cls}^{Ls} is prepended to the input of the spatial transformer, and extracted from the output after Ls layers, and the extracted tokens are concatenated into $H \in R^{n_t \times d}$. A temporal classification token is prepended into H, which is forwarded through a temporal transformer consisting of Lt layers. The temporal classification token is extracted from the transformer output and used to predict the class of the input video.

3.4 Initialization using pretrained weights

The accuracy of ViViT [3] increased when initialized with a pre-trained ViT [2] model, and therefore we chose to follow the initialization strategy with my models. We will now summarize the initialization strategies used with ViViT.

Positional embedding, p

Positional embedding token p is added to the embedded input tokens. In ViT models the shape of the p is $R^{n_w \cdot n_h \times d}$ and in the ViViT models the shape is $R^{n_t \cdot n_w \cdot n_h \times d}$. The pre-trained p is repeated n_t times in the temporal dimension to expand it to the correct shape

Embedding weights, E

Arnab et al. [3] suggested two ways of initializing the embedding weights E with the tubelet embedding method. The first method inflates the 2D convolutional filters to 3D by replicating them along the temporal dimension and averaging them.

$$E = \frac{1}{t}[E_{image}, \ldots, E_{image}]$$

The second method is central frame initialization, where center index $\frac{t}{2}$ of E is initialized with E_{image}, and all the other temporal indices are initialized with zero.

$$E = \frac{1}{t}[\ldots, 0, E_{image}, 0, \ldots]$$

4 FixMatch

FixMatch [4] algorithm was used for training the model using Semi-Supervised learning (SSL). Next, we will summarize the FixMatch algorithm and the augmentations used with the training.

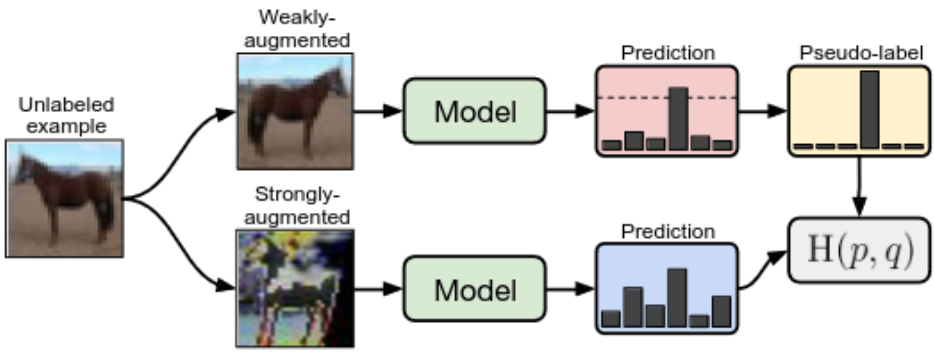

Figure 1: Diagram of FixMatch [4]. If a probability of a prediction made from a weakly augmented image is higher than the threshold(dotted line in the red box), the predicted label is used as a pseudo label. A prediction made from a strongly augmented version of the same image is compared to the pseudo label. We use the same strategy with videos instead of images.

4.1 Algorithm

FixMatch calculates separate loss functions for labeled and unlabeled data and combines them into a single loss function. For the labeled data, the loss function l_s is a standard cross-entropy loss for supervised learning.

$$l_s = \frac{1}{B}\sum_{b=1}^{B} H(p_b, p_m(y \mid \alpha(x_b)))$$

With unlabeled data, FixMatch combines pseudo-labeling and consistency regulation. The pseudo labels $\hat{q}$ are created by predicting the class of the weakly augmented data. Given threshold τ, the pseudo labels with $\hat{q} > \tau$ are used as the true labels when the unsupervised loss l_u is calculated with a cross-entropy loss for the predictions made for the strongly augmented images.

$$l_s = \frac{1}{\mu B}\sum_{b=1}^{\mu B} 1(max(q_b) \geq \tau) H(\hat{q}, p_m(y \mid \alpha(u_b)))$$

The loss function in FixMatch is the sum $l = l_s + \lambda l_u$, where λ is a hyperparameter denoting the weight of the unlabeled loss. The hyperparameter stays constant during training.

4.2 Augmentations

We use the suggested augmentation methods in [4] for weak and strong augmentations. The weak augmentation flips the video frames horizontally with a probability of 50 % and translates them horizontally and vertically by up to 12.5 %. The strong augmentation uses RandAugment [23] by randomly selecting two augmentations with randomly selected magnitude. Finally, a Cutout [24] is applied to the strongly augmented frames.

5 Experimental Evaluation

5.1 Dataset

We used a mini-Something-Something v2 (Mini-ssv2) dataset [25] to evaluate the models. Mini-ssv2 is a subset of Something-Something v2 dataset [26] and contains 87 classes and 81663 training, 11799

validation, and 12967 test videos. The duration of the videos ranges from 2 to 6 seconds.

5.2 Visual Encoding

The input to the models is 32 frames sampled with stride 2. The pre-trained model trained with supervised learning was initialized with the pre-trained ViT-model using both the filter inflation and central frame methods. From Table 1, we see that the accuracy, when trained from scratch, is significantly lower than when the model is initialized with the pre-trained image classification model. The filter inflation method slightly outperforms the central frame method and, is, therefore, also used with the semi-supervised model. Since the central frame method was better in A. Arnab et. al results with tubelet length 4 [3], the filter inflation method seems to work better with short tubelets and the central frame better with longer tubelets.

Table 1: The top1-accuracies with supervised learning

	Top1-accuracy
From scracth	28.1%
With Pre-trained weights	
Filter inflation	37.3%
Central frame	36.8%

5.3 Model settings

The model architecture follows the ViVit-B/16x2-architecture introduced in [3]. The model consists of $L_s = 12$ spatial transformer layers, $L_t = 4$ temporal transformer layers, each with a self-attention block of $N_H = 12$ heads, and hidden dimension $d = 768$. The tubelet size of the model is $t \times w \times h = 2 \times 16 \times 16$. The pre-trained model is initialized with a ViT image model trained with ImageNet-21K.

Since the dataset is small, and pure transformer-based models require large training datasets, we utilized a few regularization methods suggested in [3] to avoid overfitting with supervised learning. The supplementary contains the details of the used regularisers.

5.4 Results

We use 300 labeled videos per class with semi-supervised learning, which is approximately 30 % of the dataset. The labeled videos are randomly sampled from the dataset.

FixMatch [4] used an additional measurement called mask rate, which measures the number of unlabeled images with confidence above the threshold.

$$\text{mask rate} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} 1(max(q_b) \geq \tau)$$

Sohn et al. [4] achieved a mask rate of 98.13 % on a threshold $\tau = 0.95$ that they used in their image classification experiments. My model is less confident in the predictions it makes, and in Table 2 we can see that with the higher confidence threshold levels, the model is effectively trained only with supervised learning on a smaller dataset. The smaller confidence threshold utilizes the unlabeled data more in the training but it might impede the training with noisy pseudo-labels. Since transformer-based models are prone to overfit on small datasets, the fixed confidence threshold used in Fixmatch might not work well, and algorithms such as Dash [16] or AdaMatch[17] that use dynamic confidence thresholds might work better. The distribution of the labels in the mini-ssv2 dataset is also unbalanced, and class-specific thresholds like in [18] could improve the mask rate.

In order to balance the quality and quantity of the pseudo-labels, we use the confidence threshold $\tau = 0.25$ in the rest of the experiments. Although using $\tau = 0.25$ only utilizes 46 % of the unlabeled data, the quality of the pseudo-labels decreased with $\tau = 0.1$, which decreased the accuracy of the model. Table 3 shows that leveraging the unlabeled videos with SSL improves the top1-accuracy by 3.7% and top5-accuracy by 5.3%.

Table 2: The mask rate with different confidence thresholds

τ	mask rate
0.1	94 %
0.25	46 %
0.5	13 %
0.7	3 %
0.95	0 %

Table 3: Test accuracies using SSL and supervised learning with 300 labeled videos per class

	Top1-accuracy	Top5-accuracy
SSL	23.8%	55.7%
Supervised	20.1 %	50.4%

5.5 Comparison

Table 4: Top1- and top5-accuracies of the models. The supervised and SSL models were trained with the mini-ssv2 dataset and the ViViT-L/16x2 model was trained with the full Something-Something dataset.

	Top1-accuracy	Top5-accuracy
Supervised	37.3 %	69.8 %
SSL	23.8%	55.7%
ViViT-L/16x2 FE [3]	65.9 %	89.9%

We compare the supervised learning model, and SSL models with each other and with the state-of-the-art ViViT model. The state-of-the-art ViViT model ViViT-L/16x2 FE [3] is a larger version of the model architecture we used. It was trained with the full Something-Something v2 dataset instead of the mini-ssv2. Table 4 shows that the model outperforms my supervised model by a significant margin, top1-accuracy by 28.3 % and top5-accuracy by 20.1 %. Since the pure transformer-based models require a significant amount of data, the smaller dataset combined with fewer transformer layers might explain the margin in the results. The supervised model trained with 300 labeled videos per class had similar drop in accuracy compared to the model that was trained with full mini-ssv2 dataset. Although the model trained with supervised learning using the full mini-ssv2 dataset outperforms the model trained with SSL by a large margin, table 3 shows that using SSL improves the accuracy when only a small part of the data is labeled.

6 Conclusion

FixMatch[4] achieved state-of-the-art results on many SSL image classification benchmarks. we show in this work that SSL can be utilized with training pure-transformer-based video classification models to improve classification accuracy. The transformer-based models are prone to overfit and require large datasets, and the results suggest that semi-supervised learning can be used to improve the performance when the model is trained with smaller labeled dataset. Although using SSL improves the performance, since the model's confidence is so low, the improvement is not great. The mini-ssv2 dataset is comparatively small, and using a larger dataset could lead to a more impressive improvement in performance. The fixed confidfence interval that FixMatch uses requires the model to be confident in the predictions it makes to maximize the quantity and quality of the pseudo-labels. Too low a confidence threshold increases the quantity but decreases the quality, and too high a confidence threshold increases the confidence while decreasing the quantity. Since the prediction confidence of the model is low, an algorithm with a dynamic confidence threshold could improve the quality of the pseudo-labels.

7 Acknowledgment

I thank my advisor Selen Pehlivan for her supervision and feedback on this paper.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[3] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6836–6846, 2021.

[4] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in neural information processing systems*, vol. 33, pp. 596–608, 2020.

[5] D.-H. Lee, "Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks," *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.

[6] P. Bachman, O. Alsharif, and D. Precup, "Learning with pseudo-ensembles," *Advances in neural information processing systems*, vol. 27, 2014.

[7] L. Samuli and A. Timo, "Temporal ensembling for semi-supervised learning," in *International Conference on Learning Representations (ICLR)*, vol. 4, p. 6, 2017.

[8] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," 2016.

[9] L. Jing, T. Parag, Z. Wu, Y. Tian, and H. Wang, "Videossl: Semi-supervised learning for video classification," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1110–1119, 2021.

[10] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," *Advances in neural information processing systems*, vol. 33, pp. 6256–6268, 2020.

[11] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[13] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," *arXiv preprint arXiv:1911.09785*, 2019.

[14] H. Scudder, "Probability of error of some adaptive pattern-recognition machines," *IEEE Transactions on Information Theory*, vol. 11, no. 3, pp. 363–371, 1965.

[15] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," in *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, vol. 1, pp. 29–36, 2005.

[16] Y. Xu, L. Shang, J. Ye, Q. Qian, Y.-F. Li, B. Sun, H. Li, and R. Jin, "Dash: Semi-supervised learning with dynamic thresholding," in *International Conference on Machine Learning*, pp. 11525–11536, PMLR, 2021.

[17] D. Berthelot, R. Roelofs, K. Sohn, N. Carlini, and A. Kurakin, "Adamatch: A unified approach to semi-supervised learning and domain adaptation," *arXiv preprint arXiv:2106.04732*, 2021.

[18] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18408–18419, 2021.

[19] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[21] B. Xiong, H. Fan, K. Grauman, and C. Feichtenhofer, "Multiview pseudo-labeling for semi-supervised learning from video," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7209–7219, 2021.

[22] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6824–6835, 2021.

[23] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

[24] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[25] C.-F. Chen, R. Panda, K. Ramakrishnan, R. Feris, J. Cohn, A. Oliva, and Q. Fan, "Deep analysis of cnn-based spatio-temporal representations for action recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[26] R. Goyal, S. E. Kahou, V. Michalski, J. Materzyńska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic, "The "something something" video database for learning and evaluating visual common sense," 2017.

A Supervised learning hyperparameters

This section provides additional details about the regularizers used with supervised learning.

Base learning rate

With the base learning rate 0.5 used in [3], the models were diverging, and the training loss started to increase rapidly after a few epochs. With a base learning rate of 0.4 the pre-trained model convergences. Without the pre-trained weights, the base learning rate was reduced to 0.1.

B Semi-supervised learning hyperparameters

This section provides additional details about the regularizers used with semi-supervised learning. Since the model confidence was low, the confidence interval τ was reduced to 0.25. The other hyperparameters are the ones suggested in [4]. To avoid overfitting the model, stochastic droplayer rate and label smoothing was used.

Table 5: Training hyper parameters used with the supervised learning, the hyperparameters listed in the table where used with both pre-trained model and the model initialized from scratch

	Supervised learning
Optimization	
Optimizer	Synchronous SGD
Momentum	0.9
Batch size	64
Learning rate schedule	cosine with linear warmup
Linear warmup epochs	2.5
Epochs	25
Data augmentation	
Random crop probability	1.0
Random flip probability	0.5
Scale jitter probability	1.0
Maximum scale	1.3
Minimum scale	0.95
Other regularisation	
Stochastic droplayer rate	0.3
Label smoothing μ	0.3

Table 6: Training hyper parameters used with the semi-supervised learning. A single batch consists of 64 labeled and 44 unlabeled videos.

	Semi-supervised learning
Optimizer	Synchronous SGD
Momentum	0.9
Nesterov	True
Weight decay	0.0005
Learning rate	0.03
τ	0.25
λ_u	1
μ	7
Batch size	64
Epochs	40
Other regularisation	
Stochastic droplayer rate	0.3
Label smoothing μ	0.3

B.1 RandAugment

We adapted pytorch FixMatch implementation to work with the video data, and used the RandAugment transformations suggested in [4].

https://github.com/kekmodel/FixMatch-pytorch