Improving self-supervision in video representation learning

Liu, Hualin

2021

Liu, H. (2021). Improving self-supervision in video representation learning. Master's thesis, Nanyang Technological University, Singapore. <https://hdl.handle.net/10356/152209>

<https://hdl.handle.net/10356/152209>

<https://doi.org/10.32657/10356/152209>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 19:12:04 SGT

IMPROVING SELF-SUPERVISION IN VIDEO REPRESENTATION LEARNING

Hualin Liu

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Master of Engineering (M.Eng)

July 8, 2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

08/07/2021

Date

[LIU HUALIN]

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

08/07/2021

Date

[Prof. Zhang Hanwang]

Authorship Attribution Statement

This thesis **does not** contain any materials from papers published in peer-reviewed journals or from papers accepted at conferences in which I am listed as an author.

08/07/2021

.....

Date

NTU NTU NTU NTU NTU NTU NTU NTU

NTU NTU NTU NTU NTU NTU NTU NTU

NTU NTU NTU NTU NTU NTU NTU NTU

NTU NTU NTU NTU NTU NTU NTU NTU

.....

[LIU HUALIN]

Acknowledgments

I would like to express my sincere appreciation to my supervisor, Asst Prof Hanwang Zhang, and my academic mentors at Salesforce Research Asia, Dr Steven Hoi and Dr Junnan Li, for their academic advice during my Master's study. In addition, I would like to thank members of MReal Lab and all colleagues at Salesforce who offer me help. I would also like to extend my gratitude to Salesforce Singapore Pte Ltd and the Economic Development Board for their generous funding of my Master program.

Lastly, special thanks to my parents and Miss Gong Zhen for all the support during my study.

Abstract

With the rapid advancement of deep learning techniques in computer vision, researchers have achieved high performance in video related downstream tasks such as action classification and action detection. However, a pressing issue in this field is the scarcity of labeled data. A video contains hundreds of frames and hence it would take a daunting effort to manually collect and label a large video dataset for researchers. There are two promising directions to tackle this problem. One is self-supervised learning and the other is semi-supervised learning. In our research, we focus on improving self-supervised video representation learning methods. Current methods based on instance discrimination tasks suffer from a major limitation: semantically-similar samples are treated as negatives and their representations are enforced to be different. To address this limitation, we propose smooth contrastive learning with a weak teacher, where we employ a teacher model to mine additional supervisory signals. Specifically, the teacher model computes a similarity distribution over weakly-augmented negative samples and uses it as an artificial label to smooth the one-hot label. The student is trained on strongly-augmented samples using the smoothed label. We evaluate the learned representation on action recognition and video retrieval tasks. The proposed Weak Teacher outperforms the baseline methods under the same dataset and computation budget.

Contents

Acknowledgments	iv
Abstract	v
List of Figures	ix
List of Tables	x
List of Abbreviations	xii
1 Introduction	1
1.1 Motivation and Potential Impacts	3
1.2 Project Roadmap	3
1.3 Software and Hardware Environment	4
1.4 Organization of the Thesis	4
2 Literature Review	6
2.1 Video recognition	6
2.1.1 Handcrafted video descriptors	6
2.1.2 Spatiotemporal networks	7
2.1.2.1 Two stream networks	8
2.1.2.2 3D ConvNets	8
2.2 Self-supervised video representation learning	9
2.2.1 Temporal-related pre-training tasks	9
2.2.2 Multi-modal Learning	10
2.3 Instance-wise contrastive learning	11
2.4 Semi-supervised learning	12
2.4.1 Consistency Regularization	12
2.4.2 Entropy Minimization	14

2.4.3	Proxy-Label Methods	14
2.4.3.1	Self-training	14
2.4.3.2	Multi-view Training	14
2.4.4	Holistic Methods	15
3	Smooth Contrastive Learning with Weak Teacher	16
3.1	Preliminaries	16
3.1.1	S3D Network	16
3.1.1.1	Top-heavy Model	17
3.1.1.2	Separable Convolution	17
3.1.2	Hard Contrastive Learning	18
3.1.2.1	Contrastive Loss	18
3.1.2.2	Momentum Contrast	19
3.1.2.3	Usage of one-hot labels	20
3.2	Learning from Weak Teacher	21
3.2.1	Smooth Label	23
3.2.2	Multi Modalities	24
3.2.3	Linear Ramping	24
3.3	Weak and Strong Spatiotemporal Augmentation	25
4	Experiments	28
4.1	Dataset	28
4.1.1	UCF101	28
4.1.2	HMDB51	29
4.1.3	Usage of Datasets	30
4.2	Implementation Details	30
4.3	Action Classification	31
4.3.1	Linear Probe	31
4.3.2	Finetune	33
4.4	Video Retrieval	33
4.5	Visualization of Weak Teacher Prediction	35

5	Ablation Experiments and Results	37
5.1	Linear Ramping	37
5.2	Augmentation and Smoothing Strategy	38
5.3	Mix Ratio	39
6	Conclusions and Future Work	41
	References	43

List of Figures

1.1	Illustration of Weak Teacher Algorithm.	2
2.1	Illustration of Various Spatiotemporal networks [1].	7
2.2	Toy examples of Various Consistency Regularization Methods. Image source: [2].	12
3.1	Overview of S3D backbone [3].	17
3.2	Overview of Different Kinds of Inception Blocks [3].	18
3.3	Illustration of Momentum Contrast [4].	19
3.4	Overview of Weak Teacher framework.	21
3.5	Illustration of Temporal Augmentation.	27
4.1	Examples of Frames in UCF101 Videos	29
4.2	Visualization of Weak Prediction given Query Clips. We visualize the top 3 “negative clips” from UCF101 dataset with the highest probabilities in weak prediction. The weak prediction is calculated over a memory queue with $ K = 2048$. Notice that weak negatives with highest probability is put at the left side. We also mark the action label on the top right of frames.	36

List of Tables

3.1	A curated list of strong and weak augmentations	25
4.1	Comparison of Representation learned from Visual Features on action classification on UCF101 and HMDB51 datasets. Experiment results from both linear probe and finetuning results are shown in this table. In left columns, we list the pre-training settings in terms of dataset, backbone architecture, number of parameters and input modality. In right columns, top-1 classification accuracy is reported for both datasets. Notice that in the Dataset column, total video dataset duration is specified by number of days ('d') in brackets. In the Modality column, ' RGB ' represents RGB frames and ' O ' represents the optical flow. * stands for CoCLR results obtained using the code released by the authors, which are lower than the numbers reported in the paper [5].	32
4.2	Comparison of Representation learned by using Visual Features on Video Retrieval task on UCF101 and HMDB51 datasets. Notice CoCLR uses both RGB frames and optical flows. We report the top-K recall (R@k) on both UCF101 and HMDB51 datasets. ' RGB ' represents RGB frames and ' O ' represents the optical flow in the Modality column. * stands for CoCLR results [5] obtained by running released code, which is lower than reported.	34
5.1	Ablation Studies on Linear Ramping of Weak Teacher Algorithm. Comparison of learned representation under the Linear Probe setting on UCF101 RGB frames. Left column refers to experiment settings. Notice that learning rate is decayed to 1/10 when reaching 300 and 400 steps. We fix the mix ratio to be 0.8 in all settings.	37

5.2	Ablation Studies on Augmentation and Smoothing strategy. Comparison of learned representation under the Linear Probe setting on UCF101 RGB frames. Left column refers to experiment settings. The hyperparameters follow the ones in Table 5.1.	38
5.3	Ablation Studies on Mix Ratio of Weak Teacher Algorithm. We compare representation under Linear Probe setting on UCF101 RGB Frames. Ramping step is fixed at 250 for all experiment settings. Other hyperparameters follow the ones in Table 5.1	39

List of Abbreviations

ASR	Automatic Speech Recognition
BT	Back Translation
C3D	3D ConvNet
CE	Cross Entropy
CNN	Convolutional Neural Network
DPC	Deep Predictive Coding
FIFO	First-In-First-Out
HOF	Histogram of Optical Flow
HOG	Histogram of Gradients
DT	Dense Trajectories
I3D	Two-stream Inflated 3D ConvNet
iDT	improved Dense Trajectories
k-NN	K Nearest Neighbor
LSTM	Long-Short Term Memory
MIL	Multi-Instance Learning
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
MoCo	Momentum Contrast
MVT	Multi-View Training
NCE	Noise Contrastive Estimation
NLP	Natural Language Processing
R(2+1)D	(2+1)D ResNet
R3D	3D ResNet
RNN	Recurrent Neural Network
SIFT	Scale-Invariant Feature Transform
UDA	Unsupervised Data Augmentation

Chapter 1

Introduction

Self-supervised learning has recently become one of the most popular topics in video representation learning as it is prohibitively expensive to annotate each frame manually. Researchers have developed various self-supervised video learning techniques. Most works focus on multi-modal learning between video and audio [6,7] or between video and aligned instructions [8,9]. Despite having achieved impressive results, these works require training videos to have high-quality soundtrack and subtitles. Some focus on visual-only learning and design a series of novel pretext tasks [10–12]. Unfortunately, these hand-engineered pretext tasks heavily rely on human intuition and have no theoretical proof about why these tasks work well.

Inspired by recent advances in self-supervised image representation learning [4,13–15], there are works applying contrastive learning with visual-only features and achieved state-of-the-art performance. MemDPC [16] adopted context auto-encoding approach [17] by predicting future frames using a dense predictive coding framework with a memory. On the other hand, some works took instance discrimination approach [14]. CVRL [18] adapted contrastive learning framework to video field with temporal-consistent augmentations. However, the main problems of instance discrimination were not addressed. A well-known problem in instance discrimination was that there might be semantically-similar instances contrasted by the algorithm. Recently, CoCLR [5] addressed this problem by mining top k positive examples from alternative views (e.g. Optical Flow) using k -NN methods. However, some issues still persist: (1) The training procedure is unnecessarily complex. It involves training separate encoders for both modalities and using one encoder as teacher to train the other in iterative cycles. (2) Its main k -NN method

Figure 1.1: Illustration of Weak Teacher Algorithm.

manually specifies the number of neighbors and therefore model performance is sensitive to the value of k and potential noise in the data.

To address the above limitations, we introduce a new contrastive learning paradigm called Weak Teacher to further improve video representation learning (Idea illustrated in Figure 1.1). Using teacher predictions from weakly-augmented video clips as the learning target, the model can mine potential positives in negative examples and keep them closer to the query clip. **Green** box stands for positive and **red** box indicates negative instances.

We formulate our algorithm as a type of smoothed contrastive representation learning with weak augmentation as teacher. Our algorithm produces predictions of potential positive instances in negative examples by computing pairwise dot similarity. We then conduct a weighted sum of the teacher’s prediction and one-hot contrastive learning label. The combined prediction will be used as the target for contrastive learning. The details of the methodology will be given in later chapters. We partially draw our inspirations from the semi-supervised learning field, where predictions of weakly-augmented images are used as the target to supervise the model. Different from semi-supervised learning approaches, our method is for general representation learning and it is task-unaware. Our work is most similar to [5]. Compared to their work, we use weakly-augmented clips as the other view to mine possible positives and our training process is simpler without

multi-stage training. We conducted an extensive evaluation on two downstream tasks: action classification and video retrieval on UCF101 and HMDB51 datasets. The detailed results are shown in later chapters.

1.1 Motivation and Potential Impacts

With the advancement of social media video platforms such as Youtube and TikTok, there are millions of videos uploaded everyday. Although these videos could be great resources for training machine learning models, it would take a huge effort to manually annotate these videos. Inspired by that, our research project aims to study the pre-training algorithm of learning effective video representation from unlabeled data. We hope the proposed self-supervised learning framework is able to benefit industries such as advanced robotics, advertisement, e-commerce and surveillance.

1.2 Project Roadmap

During the course of the project, there are several goals we set at the start in order to successfully conduct our research:

1. To study current solutions and identify their limitations.
2. To survey trending methods in other vision research communities and study whether it can be integrated into video representation learning.
3. To come up with a novel self-supervised learning framework that resolves identified shortcomings in current solutions.
4. To draft an extensive experiment plan and carry it out.
5. To investigate the conclusion drawn from experiments and conduct further ablation studies.

1.3 Software and Hardware Environment

All of our experiments were carried out on a Google Cloud Platform pod with 8 Nvidia V100 GPUs sponsored by Salesforce Singapore Ltd. Codes were written in Python 3.8 ¹ with Conda ² and Docker ³ from Nvidia for development environment setup. For data engineering, we used OpenCV ⁴ for frame processing and store the frame using LMDB ⁵ for faster frame loading. As for deep learning packages, we adopted Pytorch ⁶ to write the training script as well as our proposed model. We made use of torchvision ⁷ for visual augmentation to video frames. In addition, we utilized tensorboard ⁸ (visualize model training), Seaborn ⁹ (visualize attention mechanism) and matplotlib ¹⁰ (drawing graphs) for visualization.

1.4 Organization of the Thesis

The overall organization of the thesis is as follows:

1. In Chapter 2, we will give a brief literature review on key works in video recognition, self-supervised learning and semi-supervised learning.
2. In Chapter 3, we will explain the conventional hard contrastive learning and our improved weak teacher algorithm in details.
3. In Chapter 4, we will first present the results of our experiments on two downstream tasks, action recognition and video retrieval. Then, we will compare our results with other state-of-the-art frameworks to demonstrate the effectiveness of our learned video features. Finally, we will discuss experiment results.

¹<https://www.python.org/>

²<https://anaconda.org/anaconda/conda>

³<https://www.docker.com/>

⁴<https://opencv.org/>

⁵<https://lmdb.readthedocs.io/en/release/>

⁶<https://pytorch.org/>

⁷<https://pytorch.org/vision/stable/index.html>

⁸<https://www.tensorflow.org/tensorboard>

⁹<https://seaborn.pydata.org/>

¹⁰<https://matplotlib.org/>

4. In Chapter 5, we will introduce several ablation experiments to study the effects of warm-up stage, different augmentation strategies and varying softness of label to better understand the inner mechanism of our proposed framework.
5. In Chapter 6, we will conclude our work and talk about the future direction for our project.

Chapter 2

Literature Review

In this chapter, we present a literature review on related fields. We start with an introduction of recent video recognition models, these models are able to extract spatiotemporal features from videos and will be used as visual backbones in our framework. Then we present a survey of current methodologies in self-supervised video representation learning and identify the shortcomings of these methods. Next we discuss about key works in two related fields. The first one is the instance-wise contrastive learning which is popular in self-supervised learning field. The second is semi-supervised learning. While reviewing, we will point out the inspirations our method takes from these works.

2.1 Video recognition

There has been a long line of research in video recognition. We will go through works from three main categories: handcrafted video descriptors, spatiotemporal networks and two-stream networks.

2.1.1 Handcrafted video descriptors

In the pre-deeplearning era, researchers have devised several video descriptors based on heuristics. To capture motions of objects and humans in the video, researchers propose the optical flow [19]. It is calculated by measuring displacement vectors of pixel points through consecutive frames. There are a few algorithms to compute it such as TVL¹ [20] and Deepflow [21]. From optical flow comes the emergence of local descriptors, one of the popular descriptors is HOG (Histogram of Gradients) + HOF (Histogram of Optical

Flow). As the name suggests, these are histogram descriptors computed on different modalities. Histogram is computed for each interest point with a specified neighborhood in the video. Besides local descriptors, trajectory-based descriptors are also prevalent. Dense Trajectory [22] tracks the trajectory of moving points by median filtering in the dense optical flow. Later, Wang et al. [23] comes up with improved dense trajectories (iDT). It refines dense trajectories by removing estimated effects of camera movement and inconsistent point matches. Nowadays, we have mostly relied on training video recognition backbone to extract video features. However, above mentioned handcrafted descriptors are still used as baselines or extra model inputs to boost model performance.

2.1.2 Spatiotemporal networks

As deep learning gains its traction and models like Convolutional Neural Network (CNN) start to shine in Computer Vision field, people naturally wonder if the success of CNN can be replicated in the video recognition field. Since video can be seen as a series of frames, the key research question is to effectively model temporal relationships of video frames. Over the years, researchers have designed different strategies of temporal modeling. To summarize, these include CNN with auto-regressive models, 3D convolution nets, two stream networks and hybrid models which combines 3D and two-stream networks together. An overview of model structures is given in the Figure 2.1.

Figure 2.1: Illustration of Various Spatiotemporal networks [1].

At early stages, researchers work on applying successful 2D CNNs to videos. One intuitive solution is to leverage CNN for extracting frame features and then fuse them.

Karpathy et al. [24] experiment with several fusion techniques using a single CNN network. Since the fusion of frame features introduced increasing number of parameters for 2D CNN, Karpathy et al. propose multi-resolution network which takes in two stream of inputs with different resolutions. Donahue et al. [25] consider to combine CNN with Recurrent Neural Network (RNN). They use CNN to extract frame features and leverage RNN for sequence modeling.

2.1.2.1 Two stream networks

Another direction to model temporal relationship is two stream networks. Simonyan et al. [26] propose the two stream network which separates spatial and temporal reasoning by taking frame and optical flow inputs as two streams and process them with different convolution networks. Feichtenhofer et al. [27] later study the effect of fusing spatial and temporal features and find out fusing features before pooling layer help the model to distinguish similar motions in different spatial locations on the image.

2.1.2.2 3D ConvNets

Since 2D CNN is successful in images, why not extend them into 3D for videos? Tran et al. [28] argue 3D convolution is better to model the temporal relationship. Thus, they expand 2D CNN into 3D CNN (C3D) and demonstrate 3D convolution learned good video features that captures both spatial appearances and temporal motions. With the adoption of effective ResNet [29] structure in the community, R3D is proposed later which outperforms C3D structure. To combine merits of 3D network and two stream networks, Joao et al. [1] propose Two Stream Inflated 3D ConvNet (I3D), it inflates the weights of filter and pooling layer of 2D ConvNet into 3D ConvNet and it takes both images and optical flow as inputs. This enables better transfer learning as model can leverage pretrained 2D weights. Though it has very good performance, the number of parameters is huge. To reduce the computation cost, Tran et al. [30] proposed R(2+1)D model which replaces 3D convolution operation with 2D spatial convolution followed by 1D temporal convolution. Xie et al. [3] adopt similar idea and devised Separable 3D CNN (S3D) model which incorporates top-heavy design to further improve model performance. In this work, we take S3D model as our model backbone since it is lightweight and has better performance.

2.2 Self-supervised video representation learning

Though effective video features can be obtained by training with labeled dataset, the manual effort of annotating large scale video datasets is huge. Thus, there is a need to study the un/self-supervised learning algorithm for video representation learning. Different from images, videos have much richer contents. It has a time dimension as it is formed as arrays of frames. In addition, video is multi-modal since it may contain soundtracks and subtitles. Based on these, most existing works focus on designing novel pre-training tasks and loss functions. We categorize these works into two directions, temporal-related pre-training tasks and multi-modal learning.

2.2.1 Temporal-related pre-training tasks

How can we make use of temporal dimension to devise novel pre-training tasks and losses? We can begin by coming up with temporal augmentations to the video. Considering the control panel of an ordinary video player, we can think of some basic temporal augmentations like playing a video forward or backward and playing the video at different speed. If the model can recognize the augmentation applied, we would know for sure that the model understands some properties about videos.

A lot of works follow this “augment and learn” strategy to predict the temporal augmentation applied. Pickup et al. [10] address the direction of playing the video as the “arrow of time” and consider predicting arrow of time as a special case of causal inference. They extract hand-crafted, SIFT-like features to determine in which direction the video is playing. Wei et al. [31] follow the work and evaluate it using modern CNN features and large-scale datasets. Misra et al. [32], Lee et al. [33], Xu et al. [34] further build on this strategy to predict the correct frame order from a shuffled frame sequence. Similarly, Kim et al. [35] slice the video into space-time cube and shuffle it like a space-time puzzle. The model is then trained to predict the correct space-time index given a video segment. There are also works focusing other forms of augmentations. Some examples include predicting speed of video after altering video frame rates [11] and colorizing gray-scaled videos [36].

Another line of works focuses on predicting the future based on current and past frames. This resemble the sequence prediction task in Natural Language Processing

(NLP) field. Srivastava et al. [37] build a Long Short-term Memory (LSTM) model to model the sequence and utilized it to predict subsequent frames by optimizing Mean-squared Error (MSE) loss. Mathieu et al. [38] improve the blurry predictions from MSE by introducing a combined loss with three parts: multi-scale image loss, adversarial training loss and an image gradient loss. Later, Lotter et al. [39] propose Deep Predictive Coding networks to replace the LSTM. Instead of passing context vector to the next layer, they forwarded the deviation of the motion and obtained better results.

There are several works focusing on the temporal correspondence in the videos, specifically a type of correspondence called cycle consistency that stems from the classical forward-backward consistency in tracking. Basically, cycle consistency assumes an object tracking model tracking forward should be able to return to the starting point when retracing. Wang et al. [12] propose cycle consistency loss as the euclidean distance between visual features of initial patch and end-of-cycle patch. Dwidebi et al. [40] extend cycle consistency loss to frame alignment where the cycle is defined by two aligned frames in two different videos. The tracker tracking the aligned frame in another video should be able to cycle back to itself. Recently, [41] improve the method by tracking multiple cycles at once by random walk and contrastive learning.

2.2.2 Multi-modal Learning

Besides frames, as a type of multimedia, video contains other modalities which has extra contextual information. Generally, there are four types of modalities in video: frame, optical flow, sound, subtitles. Researchers may choose any subsets of these modalities to conduct multi-modal training.

Some work studied multi-modal representation between frames and audio. Owens et al. [42] encode frames and leverage them to predict future audio features. Arandjelovic et al. [7, 43] encode frames and sound separately and devise a correspondence pretext task. Besides predicting correspondence, Alwassel et al. [6] propose cross-modal deep clustering which leverages clustering results from one modality as the supervisor signal for the other to train both frame and audio encoder.

Additionally, there are efforts to learn a joint embedding for frames and aligned instructions. Sun et al. [8] extract aligned instructions using Automatic Speech Recognition

(ASR) model for training and train the multi-modal transformer using contrastive loss. However, this one-to-one matching is not practical since the one visual frame could correspond to several subtitles. To deal with potential mismatch, Miech et al. propose a large-scale narrated video dataset HowTo100M [44] and then design a method called MIL-NCE [9] to incorporate multiple positives, which combines multi-instance learning with Noise Contrastive learning.

In several recent works, researchers explore learning between visual signals itself. Han et al. [5, 16] and Qian et al. [18] propose methods based on contrastive learning to learn an effective embedding from frames and optical flow or frames and visual-augmented frames. They obtain a similar performance when finetuning the pretrained model for classification tasks compared to model trained with audio and text. This significantly lowers the requirements for the videos as they no longer need high-quality soundtrack or subtitles. These methods are the most similar to our work and we will use a similar setting in our project.

2.3 Instance-wise contrastive learning

Recent advances in self-supervised image representation learning [4, 13–15, 45–48] have obtained remarkable results that are even comparable to supervised learning in several downstream tasks [4, 15]. Their methods are usually based on instance discrimination task [49] with Noise Contrastive Estimation [14]. Specifically, given an image, the model first produces two views of that image as positive pairs by applying strong spatial augmentations [15] and sample instances from different images to form negative pairs. Then, the model learns an embedding space by pulling positive pairs together and pushing negative pairs apart. These methods differ from each other in the way of generating negative pairs. [13] use the memory bank mechanism to store all encoded features in previous steps for sampling negative examples. Some [14, 49, 50] adopt an end-to-end mechanism where they used samples from the current mini-batch as negative samples and back-propagate gradients to the key encoder. Recently, there is a new momentum encoder approach by [4] where keys are encoded on-the-fly by a momentarily updated key encoder.

Although there has been a lot of progress in the field, there are several limitations with this approach: There may be semantically-similar positives in negative examples

Figure 2.2: Toy examples of Various Consistency Regularization Methods. Image source: [2].

which affects the performance of contrastive learning. Our technique addresses it by employing weakly-augmented clips to mine potential positive examples. By using the prediction from weakly-augmented clips as the learning target instead of one-hot label, we keep positives to be closer and in turn facilitate learning.

2.4 Semi-supervised learning

Different from self-supervised learning, semi-supervised learning aims to learn a good representation by training with a small portion of labelled data and a large amount of unlabelled data. It is well fitted to the scenario where the amount of unlabelled data is huge and labelled data is hard to acquire. In this section, we mainly introduce four key ideas in this field: Consistency Regularization, Entropy Minimization, Proxy-Label Methods and Holistic Methods. Our method draws inspirations from above mentioned ideas and can be considered as a task-unaware consistency training method with artificial labels generated from weakly-augmented inputs.

2.4.1 Consistency Regularization

Consistency Regularization proposes that models should preserve output distribution when inputs are perturbed. This idea comes from the cluster assumption which assumes that two points under the same category should be in the same cluster. This assumption requires that the decision boundary of the classifier under this assumption has to be in the low-density regions.

Initial implementation of Consistency Regularization is to minimize the distance between features of original and perturbed samples. The distance terms are treated as losses during training and common ways computing distances include Mean Squared Error (MSE) and Kullback-Leibler Divergence (KL distance). The idea was brought out initially by [51]. Over the years, researchers have gradually improved the techniques. Laine et al. [52] propose π -model which takes in two augmented version of the same input and included two loss terms. The first one is a cross-entropy loss for classification and the second is a MSE between features of two augmented inputs. The first loss term is only computed if the point has its label. Since for the most of time the first loss term is useless, an alternative way is to leverage current model's prediction as the teacher for unsupervised learning. A shortcoming of this approach is that the labels are constantly changing during training which leads to instable learning. To solve this, Laine et al. [52] propose the second version of π -model called Temporal Ensembling. It replaces current model prediction with an aggregate of past predictions. In this way, it creates a more stable teacher and hence leads to better training. The aggregation is implemented by a momentum update. Later, Tarvainen et al. [53] improve the Temporal Ensembling and propose Mean Teacher in which two models are used as teacher and student. This enables faster incorporation of new knowledge learned in training. However, Mean Teacher also carries over the instability and fluctuations of early iterations to the student model. To solve this, Ke et al. [54] designs the Dual Students framework. In this setup, two students will be trained at the same time and the role of teacher model is dynamically decided. The teacher model should be the one with the most stable prediction. The stability is measured by the following criteria: given the original and perturbed inputs, the model must (1) output the same prediction for both inputs and (2) the prediction must be confident. The confidence is measured by maximum probability of the prediction. Miyato et al. [55] propose virtual adversarial training which replaces the mean squared error with cross entropy loss and applies adversarial transformation. By adopting adversarial perturbations, it is observed that the representation is more robust against all kinds of random augmentations to inputs. Besides adversarial transformation, Xie et al. [56] studies the effect of Unsupervised Data Augmentation (UDA) for consistency training. Authors find applying RandAugment [57] best for image classification and Back Translation (BT) best for text classification.

2.4.2 Entropy Minimization

To satisfy the cluster assumption, there is another way called Entropy Minimization [58]. It makes sure predictions from unlabelled data should have high confidence and low entropy. It is usually implemented by adding an entropy term in the computation of loss. By doing so, points from the same class are close together and the decision boundary is situated in low-density regions. Researchers often incorporate this method with other ideas to achieve state-of-the-art results.

2.4.3 Proxy-Label Methods

Pseudo-labelling [59] is a technique that generates synthetic labels for unlabelled data using predictive models. These labels are created from model's prediction by picking the class with the largest probability, often realized by confidence thresholding [60]. This technique provides additional training information. The methodologies could mainly be divided into two groups: self-training and multi-view training.

2.4.3.1 Self-training

In self-training, a predictive model is firstly trained using available labelled data. Then the model is used to assign synthetic label to unlabelled data. Self-training has similar goal as the entropy minimization which encourages the model to output confident predictions. Yalniz et al. [61] leverage self-training to improve accuracy of ResNet model. The model is initially trained with synthetic labels before finetuning on labeled data. Xie et al. [62] incorporate self-training with a teacher-student model. The teacher model is trained on labelled data and it generates the pseudo labels for unlabelled data. Then the student model is trained on both labelled and unlabelled data with perturbed inputs. In the next iteration, the students become teacher and a bigger model is used for student to increase the capacity of the model. This process would be repeated throughout training.

2.4.3.2 Multi-view Training

Multi-view Training (MVT) leverages multi-modalities in the input data, which is common in real-life scenarios. Notice that the modality does not have to be another view of input data, it can also be a limited view of original data. MVT trains distinctive

predictive model for each view of data and optimize all models together. In this way, model trained with one view may benefit or complement another which could improve their performances.

One of the popular techniques under MVT is called co-training [63]. Co-training first trains two conditionally-independent models for two views of the same input point. After training, model trained with one view would be taken to create pseudo labels for model trained with the other. In this way, the information of one view is able to combine with the other view through pseudo labels.

Since co-training needs sufficient data for training two good individual models, to deal with possible scarcity of data in multiple views, an improved technique called tri-training [64] is proposed. As the name suggests, tri-training would train three individual models and optimize them together. However, it does not necessarily take three views to train three models. The three training sets can be subsets from the same labelled dataset. In the pseudo labelling process, the pseudo label for one model is only accepted if it is supported by other two models' predictions. The training process would be stopped when there is no labels added to other model's training sets.

2.4.4 Holistic Methods

Recently, holistic approaches [65–67] have combined these above mentioned strategies and have greatly improved the performance. Berthlot et al. [65] propose MixMatch which combines consistency regularization, pseudo labelling and sharpening. In the training, author propose to mix labelled data with pseudo labelled data to create two augmented dataset for consistence training. To improve MixMatch, Berthlot et al. [67] later propose ReMixMatch with two tricks called distribution alignment and augmentation anchoring. The former enforces the marginal distribution of the model prediction to be close to the marginal distribution of ground-truth label. The latter creates multiple strongly-augmented images and use prediction of weakly-augmented image as soft target for consistency training. Recently, a simpler and more powerful learning paradigm named FixMatch [66] is proposed. Instead of applying various tricks, it adopts consistency training between prediction of strongly-augmented image and pseudo labels generated from weakly-augmented images.

Chapter 3

Smooth Contrastive Learning with Weak Teacher

In this section, we will start by introducing some preliminaries. We begin with the visual backbone used in the framework. Then we briefly go through the existing contrastive learning method in the form of instance discrimination. Since only a single positive sample is used as the target, we refer to it as hard contrastive learning. Next, we will introduce the proposed Weak Teacher algorithm. Our method employs weakly-augmented images to generate additional supervisory signals in the form of a distribution over negative samples. Other modalities such as optical flow can also be incorporated under this framework and distributions from different modalities can be combined using the ensemble method. We use this probability distribution to smooth the original one-hot label as the new contrastive learning target. By doing so, we mine potential positives from the pool of negative videos, which facilitate learning of semantically-meaningful representations.

3.1 Preliminaries

3.1.1 S3D Network

In our framework, we need a video recognition model as the visual backbone to extract video feature. We chose S3D [3] since it has less parameters and has high performance. The overview of S3D model is given in Figure 3.1

Figure 3.1: Overview of S3D backbone [3].

3.1.1.1 Top-heavy Model

As I3D model was extremely heavy, the author of S3D first explored replacing partial 3D convolution operations with 2D convolutions. By experiments, authors found the model performed best when designed to be "top-heavy" which meant the model only applied computationally-heavy 3D convolution at higher layers of the model while applying 2D convolution at lower layers. The "top-heavy" design was comparably better than the reversed "bottom-heavy" design. Interestingly, when comparing it with the full I3D model, top-heavy design outperformed the original I3D. From these experiments, the author concluded 3D convolution captured the temporal mechanics better in higher layers. Moreover, it was considered to have much fewer parameters since 3D convolution was applied on a much smaller feature map.

3.1.1.2 Separable Convolution

Besides replacing part of 3D with 2D convolutions, author also incorporated separable convolution to replace 3D convolution in the "top-heavy" design. As shown in Figure 3.1, the separable convolution block was defined as a spatial convolution ($1 \times 3 \times 3$) followed by a temporal convolution ($3 \times 1 \times 1$). The diagram showing difference of regular and separable inception block is shown in Figure 3.2. (c) is the separable inception block.

Figure 3.2: Overview of Different Kinds of Inception Blocks [3].

3.1.2 Hard Contrastive Learning

Contrastive learning is a form of unsupervised learning that learns representation by contrasting positive pairs against negative pairs. The details are described as follows.

3.1.2.1 Contrastive Loss

Consider an encoded query instance q and a pool of K encoded key instances: $\{k_1, k_2, \dots, k_i\}$. Assuming there is only one positive pair, it includes one query and one positive key k^+ which matches the query. A contrastive loss function would have a low value when query q is similar to positive key k^+ and different from all other keys (negative keys) in the pool. In this paper, a loss function called InfoNCE is used. With matching score defined as the dot product between feature embeddings, minimizing this loss amounts to increasing the matching probability of positive pair. The formula is given by:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(q \cdot k^+ / \tau)}{\sum_{i=0}^K \exp(q \cdot k_i / \tau)} \quad (3.1)$$

Where τ stands for the temperature hyperparameter which scales the prediction logits.

Figure 3.3: Illustration of Momentum Contrast [4].

3.1.2.2 Momentum Contrast

In this paper, we adopt Momentum Contrast (MoCo) [4] as our contrastive learning framework. Overall idea is shown in Figure 3.3. Input clips are firstly sampled from videos. Then a strong spatial augmentation is applied to the clips. After that, augmented clips are fed through a CNN encoder. The query is encoded by a regular visual encoder represented as $f_s(\cdot)$ while the key is encoded by a momentarily-updated encoder $f_t(\cdot)$ called Teacher Model in the context of this paper. The structures of two encoders are the same. During training, the parameters of the teacher model θ_t are updated at each step as the exponential-moving-average of the student's parameters θ_s . The weight update formula for the teacher model is given below.

$$\theta_t^* = \lambda \theta_t + (1 - \lambda) \theta_s^* \quad (3.2)$$

Where λ represents the momentum. * represents the new weight.

The query representation is given by $q = f_s(x^q)$ where x^q represents the query instance. Similarly, key representation is obtained by $k = f_t(x^k)$. x^k represents the key instance. For negative keys, we use a memory queue in practice to store the negative examples for sampling.

Following that, the feature embedding will go through a Multi-Layer Perceptron (MLP). Then, feature embeddings are l2-normalized before the InfoNCE loss is computed. Please refer to Algorithm 1 for the details of Momentum Contrast.

Algorithm 1: Momentum Contrast

Input: Query clip x , key clip x_k , memory queue mq , Temperature T

Output: One Hot Label, Logit

initialization

$q \leftarrow \text{QueryEncoder}(x)$

With no gradient calculated:

$\text{UpdateKeyEncoderWeights}()$

$k \leftarrow \text{KeyEncoder}(x_k)$

$\text{NormalizeFeat}(q, k)$

$\text{pos_logit} \leftarrow \text{MatrixMul}(q, k)$

$\text{neg_logit} \leftarrow \text{MatrixMul}(q, mq)$

$\text{ScaleByTemperature}(\text{pos_logit}, \text{neg_logit}, T)$

$\text{logits} \leftarrow \text{Concat}(\text{pos_logit}, \text{neg_logit}, \text{dim}=1)$

$\text{UpdateQueue}(mq)$

$\text{label} \leftarrow \text{Concat}(1, \text{zeros_like}(k.\text{size}(1)))$

return label, logits

3.1.2.3 Usage of one-hot labels

In practise, InfoNCE is implemented by a Cross Entropy (CE) between predictive logits p_s (Eq. 3.3) and one-hot label r (Eq. 3.4) which has a value of 1 for positive pair and 0 for negative pairs. Hence, we call it hard contrastive learning in this paper.

$$p_s(k_i; q, \theta_s) = \frac{\exp(q \cdot k_i / \tau)}{\sum_{j=0}^K \exp(q \cdot k_j / \tau)} \quad (3.3)$$

$$r(k_i) = \begin{cases} 1 & \text{if } k_i = k^+ \\ 0 & \text{otherwise} \end{cases} \quad (3.4)$$

To show it clearly, we rewrite the formula of InfoNCE loss in the cross-entropy form.

Figure 3.4: Overview of Weak Teacher framework.

$$\mathcal{L}_{\text{InfoNCE}} = - \sum_{k_i=0}^K r(k_i) \cdot \log(p_s(k_i)) \quad (3.5)$$

One of the major limitations in hard contrastive learning is the usage of one-hot labels. It encourages the model to push away all negative pairs. Existing contrastive learning methods [4, 13] require a large number of negative samples, thus the memory queue inevitably contains samples that are semantically-similar to the query.

3.2 Learning from Weak Teacher

To address the above limitation, we propose a smoothed contrastive learning framework as shown in Figure 3.4. In addition to the contrastive learning structure [4] (displayed

in **Orange**), we incorporate a Weak Teacher (displayed in **Blue**) to facilitate learning. Weak videos are firstly encoded by a momentarily-updated teacher model. Then, a weak prediction is obtained by (1) computing negative similarity between the encoded weak query and weakly-augmented negatives and (2) combining negative similarity with one-hot positive similarity using a mix ratio of α . Finally, the weak prediction is used as the label for contrastive learning. Weak Teacher utilizes informative weakly-augmented videos to prevent pushing away potential positive videos. A pseudo code written in is given to illustrate the training process (See Algorithm 2).

Algorithm 2: Weak Teacher

Input: Query clip x , Strongly-augmented key clip x_s , Strong-augmented memory queue mq_s , Weakly-augmented key clip x_w , Weakly-augmented memory queue mq_w , Temperature T

Output: Weak Label, Logit

initialization

$q \leftarrow \text{QueryEncoder}(x)$

With no gradient calculated:

$\text{UpdateKeyEncoderWeights}()$

$k \leftarrow \text{KeyEncoder}(x_s)$

$w \leftarrow \text{WeakKeyEncoder}(x_w)$

$\text{NormalizeFeat}(q, k, w)$

$\text{pos_logit} \leftarrow \text{MatrixMul}(q, k)$

$\text{neg_logit} \leftarrow \text{MatrixMul}(q, mq_s)$

$\text{weak_neg_logit} \leftarrow \text{MatrixMul}(w, mq_w)$

$\text{ScaleByTemperature}(\text{pos_logit}, \text{neg_logit}, \text{weak_neg_logit}, T)$

$\text{logits} \leftarrow \text{Concat}(\text{pos_logit}, \text{neg_logit}, \text{dim}=1)$

$\text{weak_neg_prob} \leftarrow \text{Softmax}(\text{weak_neg_logit})$

$\text{weak_label} \leftarrow \text{Concat}(\text{mix_ratio} * 1, (1 - \text{mix_ratio}) * \text{weak_neg_prob})$

$\text{UpdateQueue}(mq_s, mq_w)$

return weak_label, logits

With the extension of a Weak Teacher framework, we introduce some new notations. We use w as the representation of weak query clip (Blue clip in Figure 3.4) and k^w to

stand for representations of weakly-augmented key clips. In addition to that, we keep a separate memory queue to store all weakly-augmented clips.

3.2.1 Smooth Label

We replace the original hard label with the smooth artificial label generated from the teacher’s prediction of weakly-augmented images. By doing so, our model is able to mine potential positives and keep them closer to the query instance.

The matching probability from the teacher’s prediction on weak videos p_w is similar to the student’s probability p_s with two subtle differences. First, a weakly-augmented clip w is used as weak query. Second, there is no positive key defined. Therefore, p_w can only specify the probability for negative pairs:

$$p_w(k_i^w; w, \theta_t) = \frac{\exp(w \cdot k_i^w / \tau)}{\sum_{j=0}^{K^w} \exp(w \cdot k_j^w / \tau)} \quad (3.6)$$

Where K^w stands for the number of weakly-augmented keys. Notice that negative instances are sampled from memory queues in a First-In-First-Out (FIFO) manner, hence the order of negative samples in two memory queues is the same. Once p_w is acquired, the artificial label s can then be generated by a weighted combination of a one-hot positive probability r and negative probability with mix ratio α .

$$s = \alpha r + (1 - \alpha)p_w \quad (3.7)$$

Finally, the new loss function is defined as a standard cross-entropy loss between smooth label s and student’s prediction p_s .

$$\mathcal{L}_{CE} = - \sum_{k_i=0}^K s(k_i) \log(p_s(k_i)) \quad (3.8)$$

3.2.2 Multi Modalities

In light of the multi-modal nature of videos, we assume a multi-modal teacher gives better prediction of potential positives. Therefore, we design our framework to be able to incorporate other modalities besides frames. During our experiment, we follow [5] and introduce a frozen pre-trained optical flow encoder as our flow teacher model. To generate teacher’s prediction for optical flow, we create a new memory queue to hold flow clips for positive mining. The formula for flow prediction p_f is similar to p_w :

$$p_f(k_i^f; f, \theta_{ft}) = \frac{\exp(f \cdot k_i^f / \tau)}{\sum_{j=0}^{K^f} \exp(f \cdot k_j^f / \tau)} \quad (3.9)$$

Where k^f, K^f denotes the representation and the total number of flow key clips respectively and θ_{ft} represents the parameters of flow encoder. We then combine the flow prediction and the weakly-augmented clip prediction by simple average ensemble.

3.2.3 Linear Ramping

In our assumption, weak probability is not accurate and may affect performance at the beginning of training because the encoder is not sufficiently trained. To avoid this, we adopt a linear ramping strategy to dynamically adjust the mix ratio α . The mix ratio is firstly set to 1 and gradually reduced to target value within fixed ramping steps. Before approaching the target value, the mix ratio is updated every epoch by:

$$\alpha_n = \max(\alpha_{\min}, 1 - \frac{n(1 - \alpha_{\min})}{N}) \quad (3.10)$$

Where the target mix ratio is denoted as $\alpha_{\min}$. The current epoch is represented by n and the number of total ramping steps is represented as N . By doing so, at the initial stage of training, we stick to the original InfoNCE when the weak prediction is inaccurate. As the weak prediction gains its accuracy, we gradually switch to the smooth artificial label. The optimal linear ramping ratio will be selected based on manual parameter tuning.

Augmentation	Explanation	Probability
<i>Strong Augmentation</i>		
Random Crop and Resize	Crop the frame at random scale and Resize	Uniform Distribution
Random Horizontal Flip	Randomly flip the image horizontally	0.5
Color Jittering	Randomly tweak the color	0.8
Random Gray	Randomly make the image gray	0.2
Gaussian Blur	Randomly blur the image	0.5
<i>Weak Augmentation</i>		
Scaling	Scale the clip to a fixed scale	1
Center Crop	Make a 224x224 crop in the center	1

Table 3.1: A curated list of strong and weak augmentations

3.3 Weak and Strong Spatiotemporal Augmentation

We apply two types of augmentations on input clips, namely the weak and strong augmentations. Our strategy for strong augmentations follows [15] by applying a series of frame-wise operations: random scale, random horizontal flip, color jitter, random gray and Gaussian Blur. For weak augmentations, we hope to preserve more information for accurate positive mining. Therefore we only scale the clip and then take a center crop. All augmentations are temporally-consistent (e.g. the same augmentation is applied to every frame in the clip). The explanation and parameter of these augmentation can be found in Table 3.1.

Besides spatial augmentations, temporal augmentation is applied between the query and strongly-augmented key clip [18]. It is implemented by taking clips from random starting points within the video as illustrated in Figure 3.5. The assumption here is that clips from one video should have similar high-level action semantics. By taking different starting points, the temporal augmentation keeps the same action semantics but introduce change of pose, change of lighting, and sometimes object occlusion, etc.

It is considered as a very strong augmentation. We conducted ablation experiments to compare different strategies for the augmentation. The results are shown in Chapter 5.

Figure 3.5: Illustration of Temporal Augmentation.

Chapter 4

Experiments

In this section, we introduce datasets we used, present experiments conducted to evaluate the learned video representation from our proposed algorithm and provide implementation details of our model. We extensively test the feature’s transferability on two downstream tasks: action recognition and video retrieval tasks.

4.1 Dataset

There are two action recognition datasets used in our experiments, namely UCF101 and HMDB51 dataset. We will first give a brief introduction of these two datasets. Then we will explain how we train on these two datasets in our experimental setting.

4.1.1 UCF101

UCF101 dataset [68] has 13,320 realistic YouTube videos from 101 human action classes. These videos contain large variation of pose, lighting and camera movements. These human action classes are diverse and are divided into five main types:

- Human-Object Interaction (Apply Lipstick)
- Body-Motion Only (Punch)
- Human-Human Interaction (Sumo Wrestling)
- Playing Musical Instruments (Playing Guitar, Playing Piano)

- Sports (Skiing)

Some examples of the videos in the dataset are listed in the Figure 4.1.

Figure 4.1: Examples of Frames in UCF101 Videos

4.1.2 HMDB51

HMDB51 [69] is another famous dataset with 6,766 videos from 51 distinct action classes. The videos are from an array of sources including movie clips, the Prelinger archive and

YouTube videos. The author classify action categories into 5 groups:

- General facial actions (Smile, Laugh)
- Facial actions with object manipulation (Smoke, Eat)
- General body movements (Climb, Dive)
- Body movements with object manipulation (Draw sword, Golf)
- Body movements for human interaction (Fencing, Hug)

Compared to UCF101, the lighting and camera pose are too professional to represent real-life videos since most of them are from movies. We evaluate on this dataset to demonstrate the learned embeddings’ transferability to another dataset.

4.1.3 Usage of Datasets

During our experiments, we pre-train the visual encoder on the UCF101 dataset. We follow previous work [5] that all pre-training and evaluation are conducted on split 1 of the datasets. It comes to our notice that a lot of baselines are using large-scale pre-training datasets such as Kinetics-400 [1] and Kinetics-600 [70]. We didn’t conduct experiments on these datasets due to hardware limits. Surely, we expect our method would gain a huge performance boost when pre-training on those.

4.2 Implementation Details

We implement the S3D model [71] as our visual encoder. For each video instance, we take clips of 16 frames extracted at 30 fps as input clips with a temporal down-sample rate of 2. In this way, one clip would cover around 1 second. The input video has 224×224 resolution. The optical flow is computed using the un-supervised TV-L1 algorithm [20]. Following [5], the optical flow is incorporated by adding a frozen pre-trained flow encoder as the other teacher model. During self-supervised training, we use Adam as our optimizer with an initial learning rate of $5e-4$ and a weight decay of $1e-5$. We train the model with a batch size of 64 for 500 epochs. The learning rate is reduced by multiplying 0.1 at

epoch 300 and 400. All experiment results are obtained using this self-supervised setting unless otherwise notified. Generally, it takes 25 hours to pre-train the encoder on the UCF101 dataset.

For the implementation of Weak Teacher, we choose the temperature parameter τ to be 0.07 and the mix ratio to be 0.8 to combine one-hot label and weak label. Besides, we linearly ramp down the mix ratio from 1 to target value of 0.8 over 150 epochs. In ablation studies, we will analyze the effect of different hyper-parameters.

4.3 Action Classification

We tested our pretrained video features on action classification downstream task. Given a video $V = \{f_1, f_2, f_3, \dots, f_n\}$, the objective of the task is to process the frames and summarize the clip into a feature vector and then build a softmax classifier to categorize the content into one of the pre-defined actions. There are two common evaluation protocols used in our experiments, namely Linear Probe and Finetune. We evaluate the learned representation on both settings. Notice that many self-supervised learning frameworks have been proposed in these years. Here, we list out the most recent ones trained with **visual-only features** for comparison. There are two types of visual modalities (RGB frames and optical flow) used in previous works.

4.3.1 Linear Probe

For this setting, after pre-training, a frozen visual encoder is used as the backbone and a separate linear classifier is added on top. Then, the model is trained in a supervised-learning manner with real labels. During training, we set the Batch Normalization layer in the backbone to be evaluation mode to prevent any changes to the pre-trained encoder. We follow most of the augmentation strategies in self-supervised training without Gaussian Blur. In the inference stage, we follow [5] to densely sample clips from video with a sliding window of half of the video sequence length. Then, we evaluate using the ten-crop protocol: we firstly make two videos by horizontal flipping and in each video we take five crops (center crop + four corners) and use the ensemble of ten crops as the final prediction.

					UCF101	HMDB51
Method	Dataset (Duration)	Arch	Params(M)	Modality	Acc@1	Acc@1
<i>Linear Probe Setting:</i>						
CoCLR [5]	UCF101 (1d)	S3D	7.9	RGB+O	49.0*	26.0*
Weak Teacher	UCF101 (1d)	S3D	7.9	RGB	51.2	28.0
Weak Teacher	UCF101 (1d)	S3D	7.9	RGB+O	52.3	29.3
CBT [8]	K600 (273d)	S3D	8.8	RGB	54.0	29.5
MemDPC [16]	K400 (28d)	R-2D3D	32.6	RGB+O	54.1	30.5
CoCLR [5]	K400 (28d)	S3D	7.9	RGB+O	77.8	52.4
<i>Finetuning Setting:</i>						
Jigsaw [72]	UCF101 (1d)	CaffeNet	2.3	RGB	51.5	22.5
Buchler [72]	UCF101 (1d)	CaffeNet	2.3	RGB	58.6	25.0
OPN [73]	UCF101 (1d)	VGG	5.8	RGB	59.8	23.8
VCP [74]	UCF101 (1d)	C3D	17.5	RGB	68.5	32.5
VCOP [75]	UCF101 (1d)	R(2+1)D	33.1	RGB	72.4	30.9
CoCLR [5]	UCF101 (1d)	S3D	7.9	RGB+O	76.8*	41.5*
Weak Teacher	UCF101 (1d)	S3D	7.9	RGB+O	78.8	42.0
3D-RotNet [76]	K400 (28d)	R3D	33.6	RGB	62.9	33.7
ST-Puzzle [35]	K400 (28d)	R3D	33.6	RGB	63.9	33.7
CBT [8]	K600 (273d)	S3D	8.8	RGB	79.5	44.6
SpeedNet [11]	K400 (28d)	S3D-G	11.6	RGB	81.1	48.8
MemDPC [16]	K400 (28d)	R-2D3D	32.6	RGB+O	86.1	54.5
DynamoNet [77]	Youtube8M (58d)	STC-Net	-	RGB	88.1	59.9
CoCLR [5]	K400 (28d)	S3D	7.9	RGB+O	90.6	62.9
CVRL [18]	K400 (28d)	R3D	36.1	RGB	92.1	65.4
Supervised [71]	K400 (28d)	S3D	8.8	RGB	96.8	75.9

Table 4.1: **Comparison of Representation learned from Visual Features on action classification on UCF101 and HMDB51 datasets.** Experiment results from both linear probe and finetuning results are shown in this table. In left columns, we list the pre-training settings in terms of dataset, backbone architecture, number of parameters and input modality. In right columns, top-1 classification accuracy is reported for both datasets. Notice that in the Dataset column, total video dataset duration is specified by number of days (‘d’) in brackets. In the Modality column, ‘**RGB**’ represents RGB frames and ‘**O**’ represents the optical flow. * stands for CoCLR results obtained using the code released by the authors, which are lower than the numbers reported in the paper [5]. 32

The experiment results are shown in Table 4.1. There are several points from the results which are worth mentioning. First, we can observe that Weak Teacher can outperform current SOTA method CoCLR (53.3% v.s. 49.0%). Even without the flow input, our method still outperforms the CoCLR by 2.2 percent. Second, with a tiny amount of data (1/28 of K400 dataset and 1/273 of K600 dataset) and much shorter training time (25h on UCF101 instead of 1 week or more for Kinetics training), our method can achieve comparable or even better results compared to recent self-supervised learning baselines like CBT and MemDPC.

4.3.2 Finetune

In this setting, we finetune the whole visual encoder end-to-end to compare with other state-of-the-art methods. To prevent overfitting on UCF101, a strong dropout rate of 0.9 and large weight decay $1e-3$ is used. We apply the ten-crop evaluation protocol during inference as described previously.

The results are shown in the Table 4.1. It can be observed that Weak Teacher outperforms all baseline methods trained with UCF101 dataset and it surpasses several baselines with K400 pre-training as well.

4.4 Video Retrieval

To test the feature’s transferability, we evaluate our learned representation in another downstream task called video retrieval. Video retrieval tests the whether pretrained video features of the same action category can be easily grouped together. We extract video features with a pre-trained video encoder and directly use it for nearest-neighbor retrieval in the dataset. No training or fine-tuning is needed in this process. Since clips may be from the same video with similar visual appearances, we follow the practice of [5] and query training set using testing set videos (training set and test set videos are different).

The results are shown in Table 4.2, we record top-k recall for both UCF101 and HMDB51 datasets. This evaluation metric would count a retrieval as successful whenever the correct video is found in top-k nearest neighbors. From the table, it can be observed that our method outperforms all current self-supervised learning baselines.

Method	Dataset	Modality	UCF101				HMDB51			
			R@1	R@5	R@10	R@20	R@1	R@5	R@10	R@20
Jigsaw [78]	UCF101	RGB	19.7	28.5	33.5	40.0	-	-	-	-
OPN [73]	UCF101	RGB	19.9	28.7	34.0	40.6	-	-	-	-
Buchler [72]	UCF101	RGB	25.7	36.2	42.2	49.2	-	-	-	-
VCOP [75]	UCF101	RGB	14.1	30.3	40.4	51.1	7.6	22.9	34.4	48.8
VCP [74]	UCF101	RGB	18.6	33.6	42.5	53.5	7.6	24.4	36.3	53.6
MemDPC [16]	UCF101	RGB	20.2	40.4	52.4	64.7	7.7	25.7	40.6	57.7
SpeedNet [11]	K400	RGB	13.0	28.1	37.5	49.5	-	-	-	-
CoCLR [5]	UCF101	RGB+O	37.4*	53.6*	62.6*	72.4*	13.4*	35.6*	47.5*	64.4*
Weak Teacher	UCF101	RGB+O	39.2	56.7	66.1	75.6	17.4	39.1	52.2	66.1

Table 4.2: **Comparison of Representation learned by using Visual Features on Video Retrieval task on UCF101 and HMDB51 datasets.** Notice CoCLR uses both RGB frames and optical flows. We report the top-K recall (R@k) on both UCF101 and HMDB51 datasets. ‘**RGB**’ represents RGB frames and ‘**O**’ represents the optical flow in the Modality column. * stands for CoCLR results [5] obtained by running released code, which is lower than reported.

4.5 Visualization of Weak Teacher Prediction

To better understand how Weak Teacher mines the potential positives in the negative memory queue, we visualize the top 3 weak negatives with the highest probabilities in the teacher model’s prediction. We display the frames and corresponding labels for the clips. As shown in Figure 4.2, top weak predictions of Weak Teacher are from the same or semantically-similar action categories. It empirically validates our assumption that weak prediction implicitly mines the potential positives. Despite a few mistakes, we consider the positive mining by weak teacher effective since we are training the encoder without any labels.

Figure 4.2: **Visualization of Weak Prediction given Query Clips.** We visualize the top 3 “negative clips” from UCF101 dataset with the highest probabilities in weak prediction. The weak prediction is calculated over a memory queue with $|K| = 2048$. Notice that weak negatives with highest probability is put at the left side. We also mark the action label on the top right of frames.

Chapter 5

Ablation Experiments and Results

To understand why Weak Teacher improves the performance of contrastive learning, we conduct several ablation experiments to observe the effect of different hyperparameters used, including mix ratio, linear ramping steps and different augmentation and smoothing strategies. The experiment setting is as follows: visual encoder is firstly pre-trained for 500 epochs and the learned representation is then tested with linear probe setting. Both pre-training and testing are conducted on UCF101 split 1. For convenience, we only train on the RGB frames.

5.1 Linear Ramping

Method	UCF101 Acc@1	Retrieval R@1
InfoNCE [5]	46.8	33.1
Weak Teacher _{N=0}	49.4	37.2
Weak Teacher _{N=50}	50.1	37.1
Weak Teacher _{N=150}	51.2	38.5
Weak Teacher _{N=250}	49.6	37.8

Table 5.1: **Ablation Studies on Linear Ramping of Weak Teacher Algorithm.** Comparison of learned representation under the Linear Probe setting on UCF101 RGB frames. Left column refers to experiment settings. Notice that learning rate is decayed to 1/10 when reaching 300 and 400 steps. We fix the mix ratio to be 0.8 in all settings.

First, we would like to see if the linear ramping strategy is necessary to achieve good performance in the weak teacher algorithm. We compare Weak Teacher with InfoNCE baseline and report results with different linear ramping steps in Table 5.1. Since this hyperparameter is related to training epochs, it is important to note that the learning rate is decayed to 1/10 at 300 and 400 steps during training.

As seen from the table, it appears that even without any linear ramping, the weak teacher method is able to improve the InfoNCE baseline (46.8% $\rightarrow$ 49.4%). Despite the improvement, other rows with more ramping steps outperform the no linear ramping result and the performance reaches its peak when $N = 150$. This result supports our initial motivation to adopt the linear ramping strategy.

5.2 Augmentation and Smoothing Strategy

Method	UCF101	Retrieval
	Acc@1	R@1
InfoNCE [5]	46.8	33.1
Weak Teacher	51.2	38.5
Weak Teacher _{weak aug + color jitter}	49.3	37.8
Weak Teacher _{uniform-teacher}	50.1	36.3
Weak Teacher _{top-10-teacher}	48.7	38.0
Weak Teacher _{top-20-teacher}	50.2	36.8

Table 5.2: **Ablation Studies on Augmentation and Smoothing strategy.** Comparison of learned representation under the Linear Probe setting on UCF101 RGB frames. Left column refers to experiment settings. The hyperparameters follow the ones in Table 5.1.

We test our framework with different augmentation and smoothing settings. These experiments would validate our assumption and help us understand if our augmentation and smoothing strategies are reasonable. We devise several experiments settings. The tested settings and corresponding goals are as follows:

- Stronger "Weak Teacher" - Adopt a stronger augmentation for weak teacher, discover the necessity of keeping the most semantics by weak augmentation in weak teacher's prediction.
- Uniform Teacher - A baseline to beat, represents a teacher model that knows nothing
- Top K Teacher - Only retain top K predictions for weak labels, discover whether weak predictions are noisy and part of predictive distribution are enough. We choose K to be 10 and 20 to test the impact.

Results are shown in the Table 5.2. From the table, several conclusions could be drawn. In the setting of Stronger "Weak Teacher", we observe a performance drop from (51.2% $\rightarrow$ 49.3%) which indicates a stronger augmentation like color jittering may damage the semantics of image and results in poor weak prediction and therefore inferior performance. For label smoothing, we test two strategies: uniform teacher and top K teacher. Our weak teacher strategy beats the uniform teacher baseline (51.2% v.s. 50.1%). And in Top K Teacher scenarios, it seems that including all classes instead of top k classes gives the best performance.

5.3 Mix Ratio

Method	UCF101	Retrieval
	Acc@1	R@1
InfoNCE [5]	46.8	33.1
Weak Teacher $_{\alpha=0.9}$	48.7	38.4
Weak Teacher $_{\alpha=0.8}$	49.6	37.8
Weak Teacher $_{\alpha=0.7}$	49.1	39.7

Table 5.3: **Ablation Studies on Mix Ratio of Weak Teacher Algorithm.** We compare representation under Linear Probe setting on UCF101 RGB Frames. Ramping step is fixed at 250 for all experiment settings. Other hyperparameters follow the ones in Table 5.1

We follow previous settings and conduct experiments to study the effect of the mix ratio. The larger the α is, the stronger the effect of weak prediction will be. The complete results are shown in Table 5.3. From results in table, it seems that the mix ratio does not affect much as the ramping step does. We find out model performs reasonably well when $\alpha = 0.8$ or 0.7 .

Chapter 6

Conclusions and Future Work

In this thesis, we propose a new self-supervised video representation learning algorithm. By using weakly-augmented videos as the teacher, model can find potential positive instances in sampled negatives and keep positive embeddings closer. This framework can be easily integrated to the current contrastive learning framework. We train the visual encoder on the UCF101 dataset and conduct extensive evaluation experiments on action recognition and video retrieval tasks. From main results in action classification and video retrieval, we observe the best performance compared with current state-of-the-art models when training on UCF101 dataset and the result is comparable to some old baselines trained with much larger datasets. We also conduct ablation studies to determine the effects of some designs in our framework like linear ramping, augmentation and smoothing strategies and softness of label. From those experiments, we discover the following:

1. Soft labels produced by weak augmentation effectively help the contrastive learning.
2. Mixing ratio of soft and hard label greatly impacts the model performance
3. A linear ramping warm-up process to transit from hard to soft labels helps the model learn better
4. Consistency regularization between strong and weak augmented videos is effective in video representation learning

As for future work, we would like to discover more ways of ramping strategy such as including a learnable training scheduler for curriculum learning. We hope we can enable self-paced learning for contrastive learning and we expect the learning process can be more efficient.

References

- [1] J. Carreira and A. Zisserman, “Quo vadis, action recognition? A new model and the kinetics dataset,” in *CVPR*, 2017.
- [2] Y. Ouali, C. Hudelot, and M. Tami, “An overview of deep semi-supervised learning,” *CoRR*, vol. abs/2006.05278, 2020.
- [3] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, “Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification,” in *ECCV*, 2018.
- [4] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *CVPR*, 2020.
- [5] T. Han, W. Xie, and A. Zisserman, “Self-supervised co-training for video representation learning,” in *NeurIPS*, 2020.
- [6] H. Alwassel, D. Mahajan, L. Torresani, B. Ghanem, and D. Tran, “Self-supervised learning by cross-modal audio-video clustering,” *arXiv:1911.12667*, 2019.
- [7] R. Arandjelovic and A. Zisserman, “Look, listen and learn,” in *ICCV*, 2017.
- [8] C. Sun, F. Baradel, K. Murphy, and C. Schmid, “Contrastive bidirectional transformer for temporal representation learning,” *arXiv:1906.05743*, 2019.
- [9] A. Miech, J. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, “End-to-end learning of visual representations from uncurated instructional videos,” in *CVPR*, 2020.
- [10] L. C. Pickup, Z. Pan, D. Wei, Y. Shih, C. Zhang, A. Zisserman, B. Schölkopf, and W. T. Freeman, “Seeing the arrow of time,” in *CVPR*, 2014.

REFERENCES

- [11] S. Benaim, A. Ephrat, O. Lang, I. Mosseri, W. T. Freeman, M. Rubinstein, M. Irani, and T. Dekel, “Speednet: Learning the speediness in videos,” in *CVPR*, 2020.
- [12] X. Wang, A. Jabri, and A. A. Efros, “Learning correspondence from the cycle-consistency of time,” in *CVPR*, 2019.
- [13] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, “Unsupervised feature learning via non-parametric instance discrimination,” in *CVPR*, 2018.
- [14] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv:1807.03748*, 2018.
- [15] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, “A simple framework for contrastive learning of visual representations,” in *ICML*, 2020.
- [16] T. Han, W. Xie, and A. Zisserman, “Memory-augmented dense predictive coding for video representation learning,” in *ECCV*, 2020.
- [17] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *CVPR*, 2016.
- [18] R. Qian, T. Meng, B. Gong, M. Yang, H. Wang, S. J. Belongie, and Y. Cui, “Spatiotemporal contrastive video representation learning,” *arXiv:2008.03800*, 2020.
- [19] B. K. P. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, 1981.
- [20] C. Zach, T. Pock, and H. Bischof, “A duality based approach for realtime tv- L^1 optical flow,” in *Pattern Recognition*, 2007.
- [21] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, “Deepflow: Large displacement optical flow with deep matching,” in *ICCV*, 2013.
- [22] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, “Action Recognition by Dense Trajectories,” in *CVPR*, 2011.
- [23] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *ICCV*, 2013.

REFERENCES

- [24] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. Li, “Large-scale video classification with convolutional neural networks,” in *CVPR*, 2014.
- [25] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, “Long-term recurrent convolutional networks for visual recognition and description,” in *CVPR*, 2015.
- [26] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *NeurIPS*, 2014.
- [27] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutional two-stream network fusion for video action recognition,” in *CVPR*, 2016.
- [28] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *ICCV*, 2015.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.
- [30] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” in *CVPR*, 2018.
- [31] D. Wei, J. J. Lim, A. Zisserman, and W. T. Freeman, “Learning and using the arrow of time,” in *CVPR*, 2018.
- [32] I. Misra, C. L. Zitnick, and M. Hebert, “Shuffle and learn: Unsupervised learning using temporal order verification,” in *ECCV*, 2016.
- [33] H. Lee, J. Huang, M. Singh, and M. Yang, “Unsupervised representation learning by sorting sequences,” in *ICCV*, 2017.
- [34] D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang, “Self-supervised spatiotemporal learning via video clip order prediction,” in *CVPR*, 2019.
- [35] D. Kim, D. Cho, and I. S. Kweon, “Self-supervised video representation learning with space-time cubic puzzles,” in *AAAI*, 2019.

REFERENCES

- [36] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy, “Tracking emerges by colorizing videos,” *CoRR*, vol. abs/1806.09594, 2018.
- [37] N. Srivastava, E. Mansimov, and R. Salakhutdinov, “Unsupervised learning of video representations using lstms,” in *ICML*, 2015.
- [38] M. Mathieu, C. Couprie, and Y. LeCun, “Deep multi-scale video prediction beyond mean square error,” in *ICLR*, 2016.
- [39] W. Lotter, G. Kreiman, and D. D. Cox, “Deep predictive coding networks for video prediction and unsupervised learning,” in *ICLR*, 2017.
- [40] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, “Temporal cycle-consistency learning,” in *CVPR*, 2019.
- [41] A. Jabri, A. Owens, and A. A. Efros, “Space-time correspondence as a contrastive random walk,” in *NeurIPS*, 2020.
- [42] A. Owens, P. Isola, J. H. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman, “Visually indicated sounds,” in *CVPR*, 2016.
- [43] R. Arandjelovic and A. Zisserman, “Objects that sound,” in *ECCV*, 2018.
- [44] A. Miech, D. Zhukov, J. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, “Howto100m: Learning a text-video embedding by watching hundred million narrated video clips,” in *ICCV*, 2019.
- [45] Y. Tian, D. Krishnan, and P. Isola, “Contrastive multiview coding,” *arXiv:1906.05849*, 2019.
- [46] J. Li, P. Zhou, C. Xiong, R. Socher, and S. C. H. Hoi, “Prototypical contrastive learning of unsupervised representations,” *arXiv:2005.04966*, 2020.
- [47] J. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Á. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, “Bootstrap your own latent: A new approach to self-supervised learning,” *arXiv:2006.07733*, 2020.

REFERENCES

- [48] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” in *NeurIPS*, 2020.
- [49] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in *CVPR*, 2006.
- [50] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, “Learning deep representations by mutual information estimation and maximization,” in *ICLR*, 2019.
- [51] P. Bachman, O. Alsharif, and D. Precup, “Learning with pseudo-ensembles,” in *NeurIPS*, 2014.
- [52] S. Laine and T. Aila, “Temporal ensembling for semi-supervised learning,” in *ICLR*, 2017.
- [53] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *NeurIPS*, 2017.
- [54] Z. Ke, D. Wang, Q. Yan, J. S. J. Ren, and R. W. H. Lau, “Dual student: Breaking the limits of the teacher in semi-supervised learning,” in *ICCV*, 2019.
- [55] T. Miyato, S. Maeda, M. Koyama, and S. Ishii, “Virtual adversarial training: A regularization method for supervised and semi-supervised learning,” *TPAMI*, 2019.
- [56] Q. Xie, Z. Dai, E. H. Hovy, M. Luong, and Q. V. Le, “Unsupervised data augmentation,” in *NeurIPS*, 2020.
- [57] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “Randaugment: Practical data augmentation with no separate search,” *CoRR*, vol. abs/1909.13719, 2019.
- [58] Y. Grandvalet and Y. Bengio, “Semi-supervised learning by entropy minimization,” in *NeurIPS*, 2004.
- [59] D.-H. Lee, “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in *ICCV Workshop*, 2013.

REFERENCES

- [60] C. Rosenberg, M. Hebert, and H. Schneiderman, “Semi-supervised self-training of object detection models,” in *WACV*, 2005.
- [61] I. Z. Yalniz, H. Jégou, K. Chen, M. Paluri, and D. Mahajan, “Billion-scale semi-supervised learning for image classification,” *CoRR*, vol. abs/1905.00546, 2019.
- [62] Q. Xie, M. Luong, E. H. Hovy, and Q. V. Le, “Self-training with noisy student improves imagenet classification,” in *CVPR*, 2020.
- [63] A. Blum and T. M. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, 1998.
- [64] Z. Zhou and M. Li, “Tri-training: Exploiting unlabeled data using three classifiers,” *IEEE Transactions on Knowledge of Data Engineering*, 2005.
- [65] D. Berthelot, N. Carlini, I. J. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, “Mixmatch: A holistic approach to semi-supervised learning,” in *NeurIPS*, 2019.
- [66] K. Sohn, D. Berthelot, C. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” in *NeurIPS*, 2020.
- [67] 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,” in *ICLR*, 2020.
- [68] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild,” *arXiv:1212.0402*, 2012.
- [69] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “HMDB: a large video database for human motion recognition,” in *ICCV*, 2011.
- [70] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, “A short note about kinetics-600,” *CoRR*, vol. abs/1808.01340, 2018.

REFERENCES

- [71] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, “Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification,” in *ECCV*, 2018.
- [72] U. Büchler, B. Brattoli, and B. Ommer, “Improving spatiotemporal self-supervision by deep reinforcement learning,” in *ECCV*, 2018.
- [73] H. Lee, J. Huang, M. Singh, and M. Yang, “Unsupervised representation learning by sorting sequences,” in *ICCV*, 2017.
- [74] D. Luo, C. Liu, Y. Zhou, D. Yang, C. Ma, Q. Ye, and W. Wang, “Video cloze procedure for self-supervised spatio-temporal learning,” in *AAAI*, 2020.
- [75] D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang, “Self-supervised spatiotemporal learning via video clip order prediction,” in *CVPR*, 2019.
- [76] L. Jing, X. Yang, J. Liu, and Y. Tian, “Self-supervised spatiotemporal feature learning by video geometric transformations,” in *arXiv: 1811.11387*, 2018.
- [77] A. Diba, V. Sharma, L. V. Gool, and R. Stiefelhagen, “Dynamonet: Dynamic action and motion network,” in *ICCV*, 2019.
- [78] M. Noroozi and P. Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in *ECCV*, 2016.