Self-supervised representation learning for long-complex activities using multiple modalities

Arushi Rai[1]
arai4@hawk.iit.edu
Madeline Schiappa[2]
mschiappa@ucf.edu
Dr. Yogesh Rawat[2]
yogesh@crcv.ucf.edu
Dr. Mubarak Shah[2]
shah@crcv.ucf.edu

[1] Illinois Institute of Technology
Chicago, IL

[2] University of Central Florida
Center for Research in Computer Vision
Orlando, FL

Abstract

Annotation for large video datasets is expensive and is difficult to ensure quality and exclusivity of labels. This is why learning representation on a large, diverse dataset through self-supervised tasks is highly valuable. As humans we tend to ground our visual perception with external modalities such as sound, touch, or even subtitles when watching movies. Will this generalize to representation learning such that text supervision produces better action representation? Our method involves the deep clustering of embedded video and text features to generate psuedolabels for training in combination with a cross-modal contrastive loss between video and the aligned text segments. The weights of trained encoder are frozen and then this frozen representation is used to evaluate on down-stream tasks to classify long-complex actions in video. Both multi-modal approaches outperform the single modal approach in our experiments.

1 Introduction

Since 2005, one hundred hours of content are uploaded on YouTube every minute. Despite having this rich content from video sharing platforms, we curate expensive labeled datasets to do both learning and evaluation due to the reliance on supervised learning. These curated datasets contain select tasks that are not representative of the wide distribution of actions. Furthermore, assigning labels to actions are a subjective task and perhaps by training for classification, we lose out on other forms of association like hierarchical relationships or dependencies.

The issue with short action recognition datasets such as UCF-101 is that the activities are quite short and samples from the same class contain the same background, which is why it is prone to scene bias. This is why learning from larger and longer uncurated video datasets leads to more meaningful representations [1]. The HowTo100m dataset contains 100 million pairs of videos and narrations, which are generated through automatic speech recognition (ASR). By utilizing this dataset we can take advantage of multiple modalities to aid in supervision.

Despite the noisiness of ASR outputs, our experiments find that it's embeddings produce higher accuracy on task classification than video embeddings. Our goal is to design a model that can utilize these narrations to supervise the learning of activities in videos.

Deep Cluster [2] is an unsupervised representation learning framework for images. We adapt this approach to video and implement three multi-modal approaches that incorporate ASR outputs to improve feature embeddings from video. In [2], Deep Cluster uses pseudolabels produced by clustering assignments of image embeddings to train the model. We're adapting DeepCluster by incorporating multiple modalities such as video and text. In addition, clusters prematurely converge in our empirical experiments observing the clusters formed by the single-modal and cross-modal embeddings. We hypothesize that using a combined loss with cross entropy and triplet loss will produce well-separated clusters and paired video-text embeddings closer and utlimately, produce a better representation.

2 Related Work

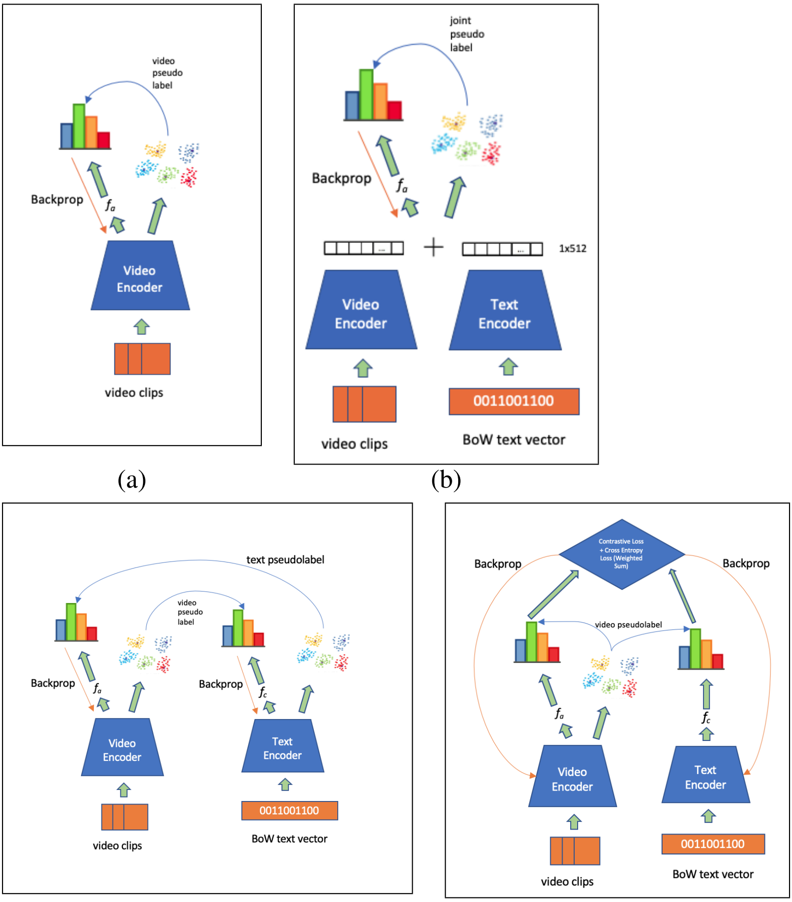
2.1 Unsupervised representation learning

Representation learning produces a high-level data encoding which can be applied across different tasks (classification, detection, etc) in the same modality as the input. Many works use an encoder to extract a high level representation and a decoder to reconstruct the input. [4, 8] trained with a reconstruction loss between the original input and decoder output. Other methods [5] use modifiers to alter the input and predict the modification. In fact, a common approach to train models in natural language processing (NLP) is to randomly mask text inputs and then predict the masked tokens [6].

Discriminative approaches in representation learning [1, 2, 3] often involve clustering. DeepCluster [2] created an end-to-end framework for unsupervised learning with any clustering method. This paper also included retrieval tasks to further evaluate the representation's ability to capture instance-level information. In the DeepCluster framework, feature embeddings are extracted from the output of final Convolutional Layer before the fully connected layers. These feature embeddings are clustered and then their respective clustering assignments are used as pseudolabels to train the model.

2.2 Multi-modal Deep Learning

Many techniques employ signals from additional modalities to improve the performance of their model. For example, [3] fuses embeddings from both RGB and Dense Trajectory and improved their performance by 5% on the UCF-101 dataset. [1] uses the signal of one modality to supervise the other modality. Cross-Modal Deep Clustering (XDC) [1] adapts the DeepCluster framework to utilize the pseudolabels produced by clustering audio embeddings to train the video encoder and vice versa. Also training on HowTo100M, the MIL-NCE approach [7] learns a joint embedding between video and text and finds that NCE is well-suited to handling the misalignment in narrations and the action present in HowTo100M and any sort of instructional video.



(c) (d)

Figure 1: The different architectures: (a) single modal deep clustering (SMD); (b) joint deep clustering (JDC); (c) cross modal deep clustering (XDC); (d) cross modal contrastive deep clustering (XCC)

3 Method

In this section, we discuss the methods proposed in Figure 1. This two part framework, first generates pseudolabels which is the same across the clusters the embeddings and then uses these clustering assignments as psuedolabels to classify and compute the loss. To train the mutli-modal models, the video encoder requires a set of frames and the narrations for the entire video vectorized by Bag of Words.

3.1 Generating Pseudolabels

In order to generate pseudolabels, we must first extract features with an encoder and cluster the extracted features. We then use the clustering assignment label for each video as its pseudolabel for training. In the following sections we will be explaining the specifics of video and text feature extraction.

3.1.1 Video Feature Extraction

In this paragraph I will describe the video encoder model for our preliminary experiments. To extract motion features, we use the Inflated 3d Inception architecture (I3D network), and initialize it with weights pretrained on Kinetics with the classification layer removed. In order to test across multiple clip lengths, we added an adaptive average pooling layer before classification. We also reduced the size of the input frames to 112x112 from 224x224 to reduce the memory load, so then we changed the size of the average pooling from 2x7x7 to 2x4x4.

When we pursue our actual experiments, we will use I3D on multiple clips and pool their results to get the features for the long-complex video. We will discuss other approaches more suitable for long-complex activities in the extended discussion section.

3.1.2 Text Feature Extraction

We tried two separate text encoder methods, roBERTa and Bag of Words (BoW) to encode the text before it went through the fully-connected layer. Our first approach was to use BPE to encode all narrations for the video, and took the first 512 tokens due to the token limit on roBERTa. Then we used roBERTa to extract an encoding which was of size (number of tokens)x1024 and did adaptive pooling on this tensor to produce a vector of shape 1x1024 that represented the entire video.

Our second approach was to use BoW to represent the narrations as one vector. This approach requires two steps, building a vocabulary of known words (1) and representing the presence of these words for each sample (2). The first part required a preprocessing step that would that capture the 300 most frequent words in our train narrations corpus and then each text narration would be represented as a binary vector (1x300). One of the downsides of this approach is that this preprocessing would need to be done prior to experimentation. Although temporal information is lost in this approach, it was lost in the previous approach by doing adaptive pooling as well. Because we are simply trying to perform text classification rather than predictive tasks, order may not be important.

3.1.3 Clustering

After the encoder layers, the embedding is passed to a K-Means clustering algorithm which produced cluster assignments for each sample.

3.2 Classifier and Loss

Clustering assignment labels change every epoch since the value of the label is assigned arbitrarily to each cluster. This is why the classifier head would need to be re-initialized every epoch. The classifier F_v and F_t are 1-2 FC layers, dependent on the architecture. We mainly use a cross-entropy classification loss for all architectures except for the cross

3.2.1 Single-modal deep clustering

We save the pseudolabels, described in the previous section, for each train sample, and use this and the classifier output as inputs to the cross-entropy loss.

3.2.2 Multi-modal deep clustering

Cross-Modal Deep Clustering. This approach clusters text and video embeddings separately and uses the pseudolabel learned from one modality to train the other modality's model serving as a supervisory signal. The same logic applies on the other modality, so at the end both modality's are being supervised by each others clustering assignments. Let p_v and p_t represent video and text pseudolabels respectively. For further clarification the inputs to the cross entropy loss is:

$$L_{crossentropy_v}(p_t, F_v(x_v)) \tag{1}$$

$$L_{crossentropy_t}(p_v, F_t(x_t)) \tag{2}$$

Joint Deep Clustering. This approach tacks on an addition FC layer on the encoder to reduce the feature vector size from 1x1024 to 1x512. Then the out of the video and text encoder are concatenated on axis 0 to produce a 1x1024 vector which is then clustered. Then this joint pseudolabel is used to compute a cross-entropy loss to update the weights of both encoders.

Cross-Modal Contrastive Deep Clustering. In this approach, we cluster both the video embedding and the text embedding but only use the video cluster assignments to compute the cross entropy loss. In this approach we're using a combined loss which is a weighted sum between contrastive loss and cross entropy loss.

$$L_{contrastive}(A,P,N,m) = \max\{|F_v(A) - F_t(P)|^2 - |F_v(A) - F_t(N)|^2 + m, 0\} \tag{3}$$

$$L = \alpha * L_{crossentropy}(A, pseudolabel_A) + (1-\alpha) * L_{contrastive}(A,P,N,m) \tag{4}$$

For the contrastive loss, we use the video as the anchor A and it's respective narrations as a positive example P. We select the negative text example N by choosing a random text sample from a different cluster than the postive text example. So even though the text pseudolabels are not used to compute the cross entropy loss, it is still used to select negative examples in real-time rather than mined beforehand. However, an issue with this approach is that ideally we want to select a moderate negative rather than something that is already far apart in terms of embedding.

4 Experiments

In this section, we will detail our experiments done on UCF101 and COIN with different single modal and multi modal deep cluster setups. We have a Future Experiments section which will detail our downstream tasks.

4.1 Dataset

Preliminary experiments datasets. We adapt the DeepCluster framework to video and evaluate the effectiveness of the resulting Single-Modal Deep Clustering by using the UCF-101 and a subset of the COIN dataset. We also use this subset, explained later, to pretrain and compare multi-modal and single-modal approaches since the subset also contains ASR text data not found in the original COIN dataset. UCF-101 is short activity dataset with 13K

Method	Dataset	MM	Model	Frozen	Accuracy
ClipOrder	UCF101	None	R(2+1)D	No	72.4
Hou *et al*. 2018	UCF101	Flow	K-Means	No	85.5
CBT	K600	None	S3D	Yes	54.0
Alwassel *et al*. 2019	Kinetics	None	SDC	No	61.8
Alwassel *et al*. 2019	Kinetics	Audio	XDC	No	74.2
Alwassel *et al*. 2019	IG-Kinetics	Audio	XDC	No	95.5
MIL-NCE	HTM	Text	I3D	Yes	83.4
Fully Supervised	Kinetics	None	I3D	Yes	42.0
Ours	UCF101	None	SDC	Yes	77.0

Table 1: Self-Supervised methods on UCF101 Results. Second best performance on frozen features. The performance of XDC pretrained on IG-Kinetics is not a fair comparison because IG-Kinetics contains 65M samples versus our roughly 10k sample and they are using more frames (32 vs 16). The same is true for MIL-NCE training. However, this may suggest better performance when we use the HowTo100M dataset to pretrain.

Evaluation Dataset	MM	Method	Pretraining	Accuracy
COIN - Text	None	SMC [roBERTa]	COIN	12.7
COIN - Text	None	SMC [BoW]	COIN	82.7
COIN – Text	Video	XDC	COIN	80.4

Table 2: Even though roBERTa was SoTA for text-related tasks it doesn't generalize to ASR vocabulary. As shown above, a simple BoW approach boosts the performance. Adding video signal seems to downgrade the performance of BoW by 2 points.

samples from 101 different action classes with each clip lasting about 7-8 seconds on average. On the contrary, COIN is long-complex activity dataset containing 11K samples with rich hierarchical annotations at multiple levels like domain, task, and step. For our preliminary experiments we took the intersection of videos between COIN and HowTo100M, 1101 videos, to get the HowTo100M metadata and narrations and the COIN dataset's detailed annotations so we could monitor details like cluster quality and to use the same dataset to simplify, yet gain meaningful information from our preliminary experiments.

Pretraining dataset. We will use the uncurated, large, diverse HowTo100M dataset to pretrain the models. HowTo100M is a complex dataset with over 136M samples from over 23K domains complete with narrations downloaded as captions from YouTube. The captions are either transcribed by the original creator but mainly generated with ASR, making for very noisy data. Another detail is most pretrained NLP models are on written language either on web or literature rather than ASR outputs.

Downstream datasets. To demonstrate the generalizability of the representation learned during pretraining, we will use 4 datasets across 2 tasks. For the action recognition task we will be evaluating on *UCF-101, HMDB-51,* and *Kinetics-700* and comparing on the many existing benchmarks. For the action segmentation task we will be evaluating on the full COIN dataset described earlier in this section.

Evaluation Dataset	Clip length	MM	Method	Pretraining	Accuracy
COIN - Video	16	None	SMC	COIN	22.8
COIN - Video	64	None	SMC	COIN	44.9
COIN - Video	64	None	Fully Supervised	None	59.1
COIN – Video	64	Text	XDC	COIN	50.4
COIN - Video	64	Text	XCC	COIN	49.6

Table 3: With the fully supervised approach as an upper bound, both XDC and XCC outperform our single modal implementation.

4.2 Set Up

1. For experiments using the UCF-101 extract frames from input video 16 frames, with a random starting point and a skip rate of 2 and for other datasets, extract 8 segments of 8 frames continuously with a skip rate of two
2. Resize frames to 224x224 and then crop the frames to get 112x112
3. Normalize the RGB values for the frames

We pretrain on UCF-101 or COIN using the various architectures and then freeze the encoder weights and evaluate on either UCF-101 or COIN for our experiments.
Optimization. Stochastic gradient descent optimizer with an initial learning rate of 10e-2, momentum -0.9, and weight decay of 10e-5.

5 Conclusion

We find that single modal deep clustering provides a better action representation than an I3D model with weights pretrained on Kinetics. Expectedly, learning representation from the UCF-101 dataset does not generalize to the COIN dataset. We also find a simple bag of words approach outperforms roBERTa by a large margin. This is because roBERTa is pretrained on a web text corpus, and our narration data does not follow the same distribution due to it's noisiness from ASR and the difference between spoken language and written language. Lastly, both multi-modal approaches outperform the single modal approach on COIN. They both come within 10% of the fully supervised training performance, and if we pretrain on a larger dataset, we could potentially come close to this upper bound or surpass it.

References

[1] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering, 2019.

[2] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features, 2018.

[3] Jingyi Hou, Xinxiao Wu, Jin Chen, Jiebo Luo, and Yunde Jia. Unsupervised deep learning of mid-level video representation for action recognition. In *AAAI*, 2018.

[4] Yunji Kim, Seonghyeon Nam, In Cho, and Seon Joo Kim. Unsupervised keypoint learning for guiding class-conditional video prediction, 2019.

[5] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning, 2019.

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[7] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos, 2019.

[8] Fitsum A. Reda, Deqing Sun, Aysegul Dundar, Mohammad Shoeybi, Guilin Liu, Kevin J. Shih, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Unsupervised video interpolation using cycle consistency, 2019.