Project Report - ECE 285

Zhaofang Qian
Department of ECE
A59019350

Yuyang Wu
Department of ECE
A15858158

Abstract

With the development of Transformers and LLaMa, large model using self-supervised learning approach shows promising generalizability on downstream tasks. Specifically, DINOv2 has clear improvements than OpenClip on video action recognition task. To fully optimized on this task, we combined DINOv2 with ST-Adapter to include temporal information. Our model is then tested on UCF101 and HMDB51 dataset and reaches 99.7% and 73.1% respectively. In the following report, We first introduce our project motivation and problem statement. Then, we discuss each previous work and dataset we used. Lastly, experiment and ablation studies were done specifically on HMDB51 dataset and results were shown.

1 Introduction

- The Motivation

  Video recognition is a long-standing problem in computer vision, which has generated significant attention from the artificial intelligence field due to its diverse applications in the real world. To accomplish the goal of video action recognition, the computer needs to understand the variety of elements, including humans, objects, and their actions within a video. It is one of the essential technologies to implement autonomous driving, video searching, or video surveillance etc.

  Unlike the traditional computer vision fields like image recognition which only deals with 2D representations, video recognition involves handling temporal information to understand spatial data. With high growth of video contents from social platforms such as Tiktok and YouTube, the data that could be used to train video recognition becomes easier to access. Also, the large platforms found having a reliable model for video recognition tasks is profitable. Our motivation for solving this problem is based on this highly growing demands for building an efficient video recognition systems.

- Problem Statement

  The existing techniques have meet a lot of limitations. The traditional supervised learning methods require tons of data labeling. This will lead to extensive labor cost as the systems requires more and more data for future training. Therefore, the self-supervised learning models are welcomed as they could learn useful features from unlabeled data to accurately identify objectives. However, the video recognition of objects in complex scenes are still difficult for these models to achieve a high accuracy. Thus, we select the DINOv2, which stands for "Self-Distillation with NO labels version 2"[4], to test the performance of newest self-supervised learning model for video recognition.

- Approaches

  To test the performance of our model, we will run it on two dataset that has not been tested for the model, which are UCF101 and HMDB51. These dataset are very commonly used for video action recognition models.

  As the DINOv2 does not include the temporal information in its training, we will then include this information as we include ST-Adapter in our model to check if adding temporal information would add more features for the model improve the performances.

Last, we would add key frame extractor to the model architecture. This extractor would help us pick key frames that represent most different frames.

- Results Summary

  After we run different models on both dataset. We would reach 99.7% at most on the UCF101 dataset and 73.1% on HMDB51 dataset, which shows competitive performance among other video recognition models from Papers With Code.

2 Related Work

2.1 DINOv2: Learning Robust Visual Features without Supervision

DINOv2 stands for "Self-Distillation with NO labels version 2". [4] The model is developed by MetaAI for self-supervised learning of video recognition. The model trained a ViT model[2] with 1B parameters and distill it into smaller parts to improve the understanding of video features.

The self-supervised learning[3]has showed excellent performance in Natural Language Processing by implementing Large Language Models. This method is different from the traditional Computer Vision models as it does not require massive labeling. Also, compared to other self-supervised learning methods that are based on reconstruction, DINOv2 does not require fine-tuning. This is because DINOv2 could generate features that can be used as input directly for linear classifiers and on different tasks.

2.2 ST-Adapter: Parameter-Efficient Image-to-Video Transfer Learning

The ST-Adapter stands for a new Spartio-Temporal Adapter. The ST-adapter solve the problem of efficiently adapting large pre-trained image models for video downstream tasks.[5]

As the pre-trained image models, like DINOv2, are not able to contain temporal structured information in its training process, Space-Time Adapter can extract and leverage features from image models to implement video understanding with a small parameter cost. We add ST-Adapter to DINOv2 model to check if adding temporal information could make it achieve a better performance.

3 Dataset

The UCF101 dataset consists 13,320 video clips from Youtube that are separated into 101 small categories.[6] These categories can be combined into 5 larger types, including body motion, human-human interactions, human-object interactions, and so on. The top models have 3-fold accuracy of above 98% on UCF101 dataset. We downloaded UCF101 video dataset and TrainTestSplits documents from its website. Then we use UCFDataset to load the UCF101 data, in this loader, we will separate data into train or test, crop the frames into 224*224 and make sure each clips has exact 'frames_per_clip' frames. The dataset will output output each video clip as a tensor of shape '(frames_per_clip, 3, 224, 224)'

The HMDB51 dataset is collected from movies and web videos with 6,766 video of 51 action categories.[1] The top models have 3-fold accuracy of between 80% to 90% on HMDB51 dataset. First, we defined a custom PyTorch 'HMDB51Dataset' to read video files from the directory. Then, in the dataloader, it crops the frames into 224*224, set the corresponding label for each video, and pick key frames with the highest differences for training. The loader splits the dataset into 70% -15% -15% ratio for train, vailation and test.

4 Method

We are using DINOv2 as our backbone to extract and assumable feature information from each input image. As DINOv2 shares a similar structure as standard ViT, we add ST-Adapter before each Multi-Head Self-Attention block to include the temporal information. After extracting feature vectors of each image, we take the average of these vectors and use a MLP to classify each video. A model architecture is shown in Figure 1

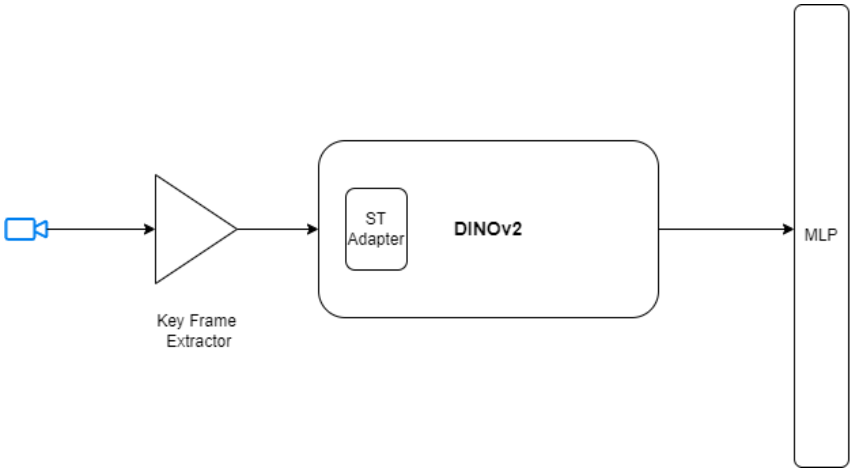

Figure 1: **Overview of our Model Architecture.** Input Video will first go over the Key Frame Extractor to sample only key frames to represent the whole video. Then each image is then processed by DINOv2 + ST-Adapter to get a feature vector representation. Each feature vector is concatenated or average with other vectors and finally using a MLP to classify the action.

4.1 Key Frame Extractor

Video is composed of frames. However, each frame in video do not have a huge spatial difference especially nowadays with videos using 60 or 120 fps. Then, it would be tedious and extremely increasing computation load when directly feed all frames in one video through a model. It may also dilute those important spatial information. We believe a more elegant and efficient way to process a video is by using key frames. Inspired by video compression, I-frames contains the main general information about a video and only takes up small space. Whereas P-frames and B-frames could be calculated or "linearly transformed" from I-frames. For our Key Frame Extractor, we firstly calculate each frame's difference between its previous frame in LUV color space and sort the differences in descending order. We then pick top K frames as our key frames to represent the whole video.

4.2 ST-Adapter

DINOv2 shows strong ability in encoding images. It is also important to include temporal information into the video. ST-Adapter is a lightweight and effective way to accomplish that goal. Specifically, we follow the standard way proposed in [5]. Before feeding images through each Transformers blocks, we add ST-Adapter before Multi-Head Self-Attention for each layer. A detailed placement is shown in Figure 2.

5 Experiments

We conduct experiments on UCF101 [6] and HMDB51[1] dataset and many ablation studies.

5.1 Experiments Setup

Dataset For UCF101, we use 85% training data (24434) as training set and 15% as validation set (4313). It has 11213 videos as testing set. The video length in UCF101 is mostly less than 10s. This dataset contains 101 different actions. For HMDB51, we use 70% training data (4736) as training set, 15% as validation set, and 15% as test set. This data set has 51 categories with video length mostly less than 15s.

Pre-trained Model We use DINOv2 ViT-S/14 distilled as our pretrained model and freeze all parameters in DINOv2. It takes image input directly and output feature vectors with 384 dimensions.

Implementation details We use 10 epochs with 8 uniformly sampled frames. The learning rate is set to 0.01 and using Adam optimizer with weight decay 0.95. For ST-Adapter, we use kernel size 3x1x1

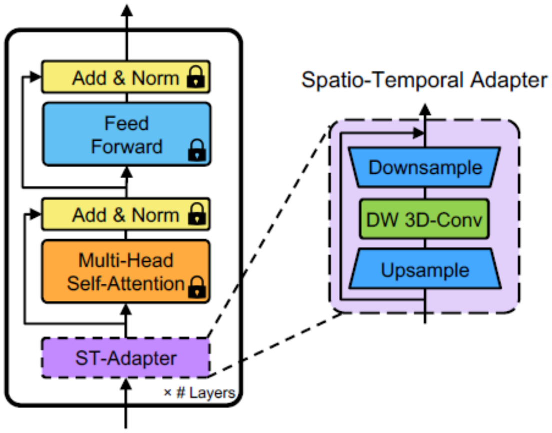

Figure 2: Spatial-Temporal Adapter. Figure from [5]

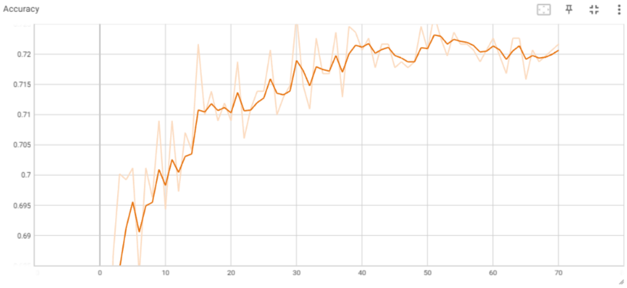

Figure 3: Accuracy curve on validation set for our best model.

with input, hidden, and output dimensions all equal to 384. The final MLP layer has 512 hidden dimensions and then classify to 51 classes.

5.2 Main Result and Analysis

In UCF101 dataset, our model could reach 87% with DINOv2 only. This is relatively low when we use ResNet152, which achieve 90.4%. We believe this is because the DINOv2 is trained using self-supervised learning, while ResNet152 is using supervised learning. After including ST-Adapter in the model and retrained for 5 epochs, we could reach 99.7% accuracy in the test set. This result is even better than the state-of the art. We believe each frame from UCF101 video has already have enough spatial information to help classify. The ST-Adapter is proven helpful to further utilize temporal information between frames. In HMDB51 dataset, we have done ablation studies to experiment with different hyperparameters and detailed results is shown in Section 4.3. Our best model use 8 frames and ST-Adapter without Key Frame Extractor and one extra MLP layer. It reaches 73.10% on test set, which is 15% of the total training video. The accuracy curve is shown in Figure 3

5.3 Ablations

In HMDB51 dataset, we conduct ablation studies on many settings. From UCF101, we realize the effectiveness of ST-Adapter. Thus, all of our experiment below will include ST-Adapter.

Number of Input Frames More input frames gives model more spatial information, but it may also dilute those important information by adding noise. We show ablation experiment on number of input frames and postprocess feature vectors in Table 1. From result, less input frames leads to a better output no matter using average or concatenation after having the output feature vectors. Also, because we are not using a strong MLP, concatenating vectors make it hard to classify.

Table 1: Result with different number of input frames and postprocess approach

Frames	PostProcess	Accuracy
8	Avg	68.08%
16	Avg	66.7%
32	Avg	83%
8	Concat	65.12%
16	Concat	60.79%
32	Concat	59.01%

MLP layers We then tried whether increasing number of mlp layers might lead a better result. From Table 2, applying one extra MLP layer is a good trade-off to further process the info from the encoder.

Table 2: Results with different MLP layers

MLP Layers	PostProcess	Accuracy
0	Avg	68.08%
1	Avg	68.97%
2	Avg	67.49%
0	Concat	65.12%
1	Concat	40.79%
2	Concat	61.67%

Key Frame Extractor We conduct several experiments on whether key frame extractor could help further improve our model's accuracy. We show the result in Table 3. When using average as the post processing approach, applying key frame might have a minor decrease on the accuracy. The influence is even larger with using concatenation as the post processing approach. We believe this is due to the noisy HMDB51 dataset. Some video might stop in the middle and actual key frames are not well distributed along the whole timeline. Also, each video in HMDB51 is relatively short. With such a short time window, using key frame would have a similar effect as uniformly sample frames. key frame extractor might help and show better result with longer videos.

Table 3: Results with Key Frame Extractor

Frames	PostProcess	Key Frame	Accuracy
8	Avg	✓	67.00%
8	Avg	×	68.08%
8	Concat	✓	55.37%
8	Concat	×	65.12%

ST-Adapter Kernel Size From [5], the author suggests using 3x1x1 kernel size. We further test whether a larger temporal window could have a better result, shown in Table 4. From the result, the smaller kernel size do have a better accuracy on the test set.

Table 4: Results with Different Kernel size in ST-Adapter

Frame	Kernel	Postprocess	Accuracy
8	3x1x1	Avg	68.08%
8	5x1x1	Avg	67.98%
16	3x1x1	Avg	66.70%
16	5x1x1	Avg	64.14%

6 Supplementary Material

A video presentation could be found in this link: `https://youtu.be/xx2drA-02pk`.

References

[1] Serre lab » HMDB: a large human motion database.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[3] Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M. Hospedales. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39(3):42–62, may 2022.

[4] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[5] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. ST-adapter: Parameter-efficient image-to-video transfer learning.

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