Efficient Video Classification Model with Pretrained Embeddings

Changho Choi
Korea University
changho98@korea.ac.kr

Daniel Jader Pellattiero
Ca' Foscari University of Venice
danieljaderpellattiero@outlook.com

Hwanseok Sim
Korea University
hwanseoksim@gmail.com

Abstract

This paper presents a fast and scalable video classification framework that leverages pre-trained multimodal embedding and shallow neural networks. The integration of visual, audio, and text metadata embeddings within a supervised pipeline has been demonstrated to yield substantial enhancement in accuracy compared to zero shot video embedding classifiers and fine-tuned TimeSformer model, while exhibiting a reduction in computational demand of up to an order of magnitude. Our ablation studies demonstrate the relative contributions of each modality and the robustness of embedding-space augmentation. The findings of this study demonstrate that pre-trained multimodal features yield semantically rich representations, thus facilitating efficient and flexible video classification.

1. Introduction

Video classification is a key task in computer vision, applicable to content recommendation and moderation. Processing raw videos typically requires extensive preprocessing (e.g., frame extraction, optical flow) and large models demanding significant GPU memory and training time. Recent embedding APIs, trained on large datasets, provide multimodal representations without custom feature extractors, enabling lightweight yet accurate video applications.

Although models like TimeSformer [4] achieve state-of-the-art performance, their computational demands are impractical for many scenarios. Using embedding APIs reduces preprocessing overhead, allowing the use of compact neural networks to integrate these embeddings. This motivated our efficient embedding-based approach.

The scope of the project evolved from short-form to broader video classification tasks using a curated subset of YouTube-8M [1].

The principal contributions of this study are as follows.

- A lightweight model integrating multimodal embeddings (video, audio, metadata) using TwelveLabs's API [14].
- Ablation studies evaluating embedding combinations, augmentation strategies, and architectures.
- A model significantly outperforming standard baselines in accuracy and training speed (approx. one minute versus ten hours for TimeSformer).
- Discussion on metadata embeddings and a novel augmentation strategy to enhance generalization and reduce overfitting.

The complete codebase is available in the GitHub repository. [1]

2. Related Work

The advancements in video classification have been significantly driven by large-scale datasets such as Kinetics [7] and YouTube-8M [1], encouraging the development of sophisticated models.

Transformer-based architectures, successful in natural language processing, have notably influenced video understanding. TimeSformer [4] adapted self-attention mechanisms by factorizing space-time attention. VideoMAE [13] introduced masked autoencoder pre-training for efficient self-supervised video learning, and ViViT [2] explored pure Transformer and hybrid video processing approaches. TikGuard [3] created TikHarm, a dataset for classifying harmful TikTok videos, fine-tuning TimeSformer, VideoMAE, and ViViT.

In recent video understanding research, TwelveLabs has released a suite of pre-trained models, namely Marengo [6], Pegasus [6], and their Embed API [8, 14], which streamline video and audio representation learning. Marengo serves as a multimodal video understanding backbone, enabling tasks such as classification and captioning directly from raw video streams, while Pegasus builds on Marengo to improve text-to-video alignment and generation. The Embed API, powered by these underlying models, provides 1,024dimensional multimodal embeddings.

[1] https : / / github . com / usingcolor / 20251R0136COSE47400

3. Methodology

3.1. Dataset and Preprocessing

The present study utilizes a subset of the YouTube-8M dataset (May 14$^{\text{th}}$, 2018 version) [1], consisting of approximately 1,800 videos spanning 10 classes: *Arts & Entertainment*, *Autos & Vehicles*, *Business & Industrial*, *Computers & Electronics*, *Food & Drink*, *Games*, *Law & Government*, *Pets & Animals*, *Reference*, and *Sports*. The duration of each video is approximately 20 seconds on average. The data was then divided into training, validation and test sets in an 8:1:1 ratio.

The preprocessing procedure is outlined as follows:

1. YouTube-8M dataset video shards are downloaded.
2. The working subset of 1,800 videos, balanced across the 10 classes, is sampled.
3. For each video, the following embeddings are extracted:
   - **VideoText and Audio Embeddings**: The TwelveLabs embedding API [8, 14] facilitates the extraction of 1,024-dimensional vectors.
   - **Metadata Embeddings**: A total of 3,072-dimensional vectors have been computed by applying OpenAI's `text-embedding-3-large` model [12] to textual metadata, such as video tags.
4. All embeddings are serialized into individual HDF5 files, thus ensuring efficient loading during the training and evaluation phases.

Videos are additionally processed to meet the resolution and aspect ratio requirements defined by the TwelveLabs Embed API. Specifically, a Python module leveraging the MoviePy library assesses each video's original resolution and aspect ratio, resizing and padding the videos to align with the nearest allowed dimensions (e.g., 1:1, 4:3, 16:9). This ensures all videos consistently meet API specifications, streamlining the embedding extraction process.

To verify that the pretrained embeddings exhibit class-wise structure, the Unsupervised Manifold Alignment Projection (UMAP) [11] algorithm is applied to reduce the dimensionality of the video-text embeddings. As demonstrated in Figure 1, the resulting 2D projection reveals distinct clusters corresponding to different classes.

3.2. Baseline Models

Two baselines are implemented for the purpose of comparison:

- **Zero-Shot Classification**: The precomputed TwelveLabs video-text and text embeddings are directly fed into a nearest-neighbor classification scheme, without the necessity for any task-specific training. This approach has been demonstrated to achieve an accuracy of 55.6%.
- **TimeSformer Fine-Tuning**: The TimeSformer model, which has been pre-trained on Kinetics-400, is fine-tuned using a subset of 1,800 videos. Following a training period of approximately 10 hours, the model attains an accuracy of 64.0%.

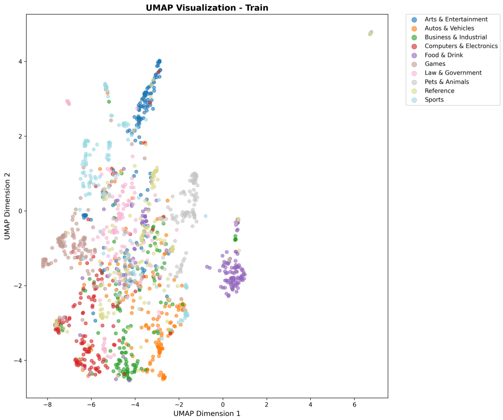

Figure 1. UMAP projection of video–text embeddings from the training set, illustrating distinct clusters for each class.

3.3. Proposed Method

The proposed model integrates the multimodal embeddings for each video, subsequently inputting them into a lightweight neural network comprising the following components:

- The input layer is responsible for accepting the concatenated vector.
- The Multi-Layer Perceptron (MLP) layer, with four hidden layers, each of which is followed by Swish activations, batch normalization, and dropout.
- The final linear classification head, which generates logits for the 10 classes.

The training process is conducted over a total of 20 epochs, with a batch size of 8. The AdamW optimizer and a CosineAnnealingLR scheduler [9] are employed to ensure the efficient convergence of the model parameters. The total duration of training on our hardware configuration is estimated to be approximately one minute.

Embedding-Space Augmentation To enhance the generalization capabilities and mitigate the risk of overfitting, a novel augmentation technique is employed directly within the embedding space. For each original embedding, denoted by $\mathbf{e}$, a perturbed version is generated according to the following method.

$$\mathbf{n} = \frac{\boldsymbol{\epsilon}}{\|\boldsymbol{\epsilon}\|_2}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

$$\mathbf{n}_{\text{scaled}} = \alpha\, \mathbf{n}, \quad \widetilde{\mathbf{e}} = \frac{\mathbf{e} + \mathbf{n}_{\text{scaled}}}{\|\mathbf{e} + \mathbf{n}_{\text{scaled}}\|_2},$$

where α is a user-defined *strength* parameter. This procedure ensures that the cosine distance between $\widetilde{\mathbf{e}}$ and $\mathbf{e}$ is at most

$$1 - \sqrt{1 - \alpha^2}.$$

The application of Gaussian noise in this normalized manner preserves the magnitude of the embedding in its entirety whilst encouraging robustness to minor perturbations in cosine space. As demonstrated in Figure 2, the augmentation process is illustrated.

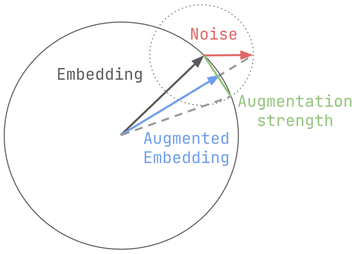

Figure 2. Schematic of the embedding-space augmentation: Gaussian noise is normalized, scaled by α, added to the original embedding, and then re-normalized.

4. Experiments

Extensive ablation studies were conducted to evaluate the contribution of each component of the proposed method. The emphasis placed on validation loss for model selection is due to the relatively small size of the validation set (180 videos), which can render accuracy metrics unreliable.

4.1. Implementation Details

For the proposed model, an MLP architecture was utilized. The input was formed by concatenating pre-extracted video, audio (from TwelveLabs models), and string metadata embeddings (from OpenAI models). Considering the findings from the ablation experiments, it can be concluded that the most effective architecture is a 4-layer MLP with Swish [5] activation functions, a dropout layer, batch normalization layers within the hidden layers and a final softmax layer, for the classification across the 10 distinct categories. The model was trained over a period of 20 epochs, with a batch size of 8. The AdamW [10] optimizer and a Cosine Annealing learning rate scheduler with a peak learning rate of 1×10^{-4} and a linear warm-up period of 1 epoch were employed. The standard cross-entropy loss function was utilized for the training process. The augmentation of data, with a strength of 0.5 as determined by our ablation studies, was applied to the input embeddings during the training process.

4.2. Ablation Studies

4.2.1. Input Features

This study has revealed the clear benefits of incorporating multimodal information alongside basic video embeddings. While the incorporation of audio embeddings into the video embeddings resulted in a modest enhancement to performance, the integration of metadata embeddings yielded the most substantial enhancement in validation accuracy. It is interesting to note that combining all three embedding types (video, audio, and metadata) did not result in the optimal level of accuracy and exhibited signs of potential overfitting, as indicated by its validation loss trend. Based on a comprehensive evaluation encompassing both the substantial enhancement in accuracy and the consistent stability of validation loss profiles, the integration of video and metadata embeddings was identified as the most effective approach. Consequently, this combination was selected for further experiments.

Table 1. Ablation Study on Input Features. Validation accuracy is reported.

Input Features Combination	Valid. Accuracy (%)
Video Embedding Only	72.8
+ Audio Embedding	73.9
+ Metadata Embedding	**81.1**
+ All Embeddings	79.4

4.2.2. Data Augmentation Strength

The study investigated the optimal augmentation strength for both video and metadata embeddings. In the context of video embeddings, a strength of 0.5 has been shown to consistently yield optimal performance across both validation loss and accuracy. This finding indicates a clear optimal point for introducing noise to the video features.

In the case of metadata embeddings, the results presented a slightly more nuanced scenario. While an augmentation strength of 1.0 led to a marginally superior peak validation accuracy, an augmentation strength of 0.5 demonstrated superior stability in its validation loss profile. Given the limited size of the validation set and the possibility of inaccurate accuracy readings, the more stable generalization indicated by the validation loss was given priority. Consequently, an augmentation strength of 0.5 was selected for metadata embeddings as well, with the objective of achieving a robust and reliable improvement.

4.2.3. Model Architecture

To determine the optimal model structure, experiments were conducted with MLP of varying depths and an attention-based model, whilst maintaining the data augmentation strength at 0.5. The findings, which are outlined in

Table 2. Ablation Study on Video Embedding Augmentation Strength. Validation accuracy is reported.

Augmentation Strength	Valid. Accuracy (%)
Video Embedding	72.8
+ 0.01 Aug. Strength	72.2
+ 0.05 Aug. Strength	72.8
+ 0.1 Aug. Strength	73.3
+ 0.5 Aug. Strength	**75.0**
+ 1.0 Aug. Strength	70.0

Table 3. Ablation Study on Metadata Embedding Augmentation Strength. The baseline is Video + Metadata embeddings without metadata augmentation. Validation accuracy is reported.

Augmentation Strength (on Metadata)	Valid. Accuracy (%)
Video emb. & Metadata emb	81.1
+ 0.1 Aug. Strength	80.0
+ 0.5 Aug. Strength	80.0
+ 1.0 Aug. Strength	**82.2**

Table 4, indicate that a 4-layer MLP achieved the maximum validation accuracy. Increasing the depth to a 5-layer MLP led to a decline in performance, suggesting that deeper models were prone to overfitting with the size of the dataset and the complexity of the features. The attention-based model, incorporating a 1D self-attention mechanism, demonstrated competitive accuracy and notably exhibited good generalization, as evidenced by its stable validation loss curve. However, this configuration did not exceed the peak accuracy of the 4-layer MLP. Consequently, the latter one was selected as the optimal architecture for the proposed method due to its superior accuracy.

Table 4. Ablation Study on Model Architecture. Validation accuracy is reported.

Model Architecture	Valid. Accuracy (%)
3 layers (MLP)	80.6
4 layers (MLP)	**81.1**
5 layers (MLP)	78.3
Attention	79.4

4.3. Overall Performance

The final model, incorporating video and metadata embeddings, 0.5 augmentation strength, and a 4-layer MLP, demonstrates significant improvements over the baselines. The test set confusion matrix for this model is shown in Figure 3.

The proposed methodology successfully achieves a compelling balance of high accuracy and significantly reduced training time, in comparison to conventional complex models such as TimeSformer fine-tuning, which can required up to ten hours.

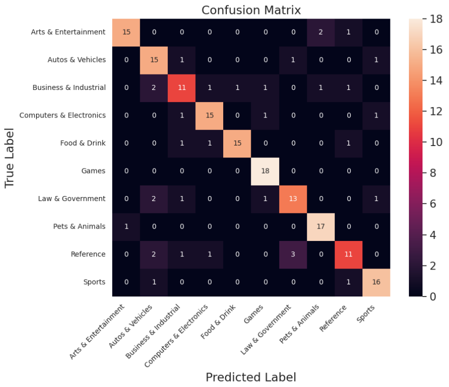

Figure 3. Confusion matrix of the 4 layers(MLP) model on the test set.

5. Conclusion

The paper sets out an efficient and effective approach for general video classification.

The following key findings are highlighted:

- The proposed method demonstrates a substantial improvement in accuracy, accompanied by a significant reduction in training time.
- The integration of multimodal embeddings, notably metadata embeddings derived from textual information, has resulted in a substantial enhancement of model performance.
- The novel data augmentation strategy applied in the embedding space has been proved to be effective in mitigating overfitting and enhancing model generalization capabilities.
- The 4-layer MLP architecture was determined to provide the optimal trade-off between performance and complexity for the task at hand.

This work demonstrates the potential of using carefully chosen pre-trained embeddings and systematic ablation studies to design a lightweight, yet powerful, video classification models.

While the proposed methodology outperformed all baseline approaches on the evaluated tasks, it should be noted that validation was undertaken on a small subset of YouTube-8M. For a more comprehensive assessment and future work, it should be applied to larger-scale datasets such as Kinetics or the original YouTube-8M dataset.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 1, 2

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 1

[3] Mazen Balat, Mahmoud Gabr, Hend Bakr, and Ahmed B Zaky. Tikguard: A deep learning transformer-based solution for detecting unsuitable tiktok content for kids. In *2024 6th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pages 337–340. IEEE, 2024. 1

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, page 4, 2021. 1

[5] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 3

[6] Raehyuk Jung, Hyojun Go, Jaehyuk Yi, Jiho Jang, Daniel Kim, Jay Suh, Aiden Lee, Cooper Han, Jae Lee, Jeff Kim, et al. Pegasus-v1 technical report. *arXiv preprint arXiv:2404.14687*, 2024. 1

[7] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1

[8] Hyeongmin Lee, Jin-Young Kim, Kyungjune Baek, Jihwan Kim, Hyojun Go, Seongsu Ha, Seokjin Han, Jiho Jang, Raehyuk Jung, Daewoo Kim, et al. Twlv-i: Analysis and insights from holistic evaluation on video foundation models. *arXiv preprint arXiv:2408.11318*, 2024. 1, 2

[9] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 2

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[11] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 2

[12] OpenAI. text-embedding-3-large. https://platform.openai.com/docs/models/text-embedding-3-large, 2023. 2

[13] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022. 1

[14] Twelve Labs. Create embeddings — twelve labs embed api. https://docs.twelvelabs.io/docs/create-embeddings, 2024. 1, 2