FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Muhammad Sheraz Khan

EVALUATING STATE-OF-THE-ART VISION-LANGUAGE MODELS FOR VIDEO RECOGNITION ON REAL WORLD DATASET

Master's Thesis
Degree Programme in Computer Science and Engineering
June 2023

Sheraz Khan M. (2023) Evaluating State-Of-The-Art Vision-Language Models for Video Recognition on Real World Dataset. University of Oulu, Degree Programme in Computer Science and Engineering, 55 p.

ABSTRACT

One of the main challenges in Computer Vision is the training of custom models from scratch. This process is highly computer-intensive, time-consuming, and requires vast amount of labeled datasets to achieve reasonable results. Recently, various foundation models trained using self-supervised learning techniques have been proposed, claiming to achieve good results for downstream tasks after fine-tuning. This document aims to discuss the results obtained by three multi-class video recognition methods based on such vision-language foundation models using a dataset that closely corresponds to real-world. The primary objective of this work was to investigate the number of instances required by these models to provide competitive results.

Three models, namely VideoMAE [1], X-CLIP [2], and Text4Vis [3], are chosen for the evaluation in this study. Their performance is assessed using YT8M [4] dataset which include YouTube videos captured in uncontrolled environments, closely resembling real-world settings. Notably, Text4Vis [3] stood out by achieving an impressive weighted F1-score of 0.87 after fine-tuning with just 1142 videos. The results of X-CLIP [2] are also competitive with Text4Vis [3], while VideoMAE [1] exhibits comparatively lower performance.

Keywords: Computer Vision, Vision-Language Foundation models, video recognition

TABLE OF CONTENTS

FOREWORD

This thesis work has been conducted at Valossa Labs Oy in Oulu, Finland. I would like to express my sincere gratitude to my company supervisors, Pirkko Mustamo and Mika Rautiainen, for their invaluable guidance and support throughout the thesis process. Their expertise and assistance were instrumental in defining the objectives and scope of the study, as well as in establishing the experimental procedures. I would also like to extend my appreciation to Guoying Zhao and Hanlin Mo for their valuable guidance and timely feedback, which have greatly contributed to steering this research in the right direction.

Oulu, June 1st, 2023

Muhammad Sheraz Khan

LIST OF ABBREVIATIONS AND SYMBOLS

AI	Artificial Intelligence
CV	Computer Vision
SOTA	State of the Art
VL	Vision Language
NLP	Natural Language Processing
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
VideoMAE	Video Masked Autoencoder
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
LLM	Large Language Model
VQA	Visual Question Answering
BERT	Bidirectional Encoder Representations from Transformers
GPT	Generative Pre-trained Transformers
CL	Contrastive Learning
SSL	Self-Supervised Learning
MLM	Masked Language Modeling
ITM	Image-Text Matching
VQA	Visual Question Answering
YT8M	Youtube Eight million
CCT	Cross-frame Communication Transformer
MIT	Multi-Frame Integration Transformer
GRU	Gated Recurrent Unit
ViT	Vision Transformer
CLIP	Contrastive Language-Image Pre-training
MIM	Masked Image Modeling
MVM	Masked Video Modeling
LRCN	Long-term Recurrent Convolutional Networks
FPS	Frames Per Second
SSVP	Self-Supervised Video Pre-training
VSP	Video-Specific Prmomting
CFA	Cross-frame Fusion Attention
IFA	Intra-Frame Diffusion Attention
LDA	Linear Discriminant Analysis
MAP	Mean Average Precision

1. INTRODUCTION

In recent years, Artificial Intelligence (AI) has gained significant momentum and has found applications in numerous fields like finance, healthcare, transportation and entertainment. For instance, AI is being used to recommend articles on the web, detect fraudulent activities, and even assist in driving cars. A rapid surge in its adoption can be mainly attributed to the availability of improved computational capacities at cheaper prices and the abundance of data being generated everyday. This allows the development of models that contain millions of learnable parameters, usually referred to as large models, trained on massive datasets. As a result, these models are more robust and closer to mimicking the workings of human brains. These large models have a greater capacity to learn underlying features of the data and make informed decisions.

Video recognition, a crucial research area in Artificial Intelligence, involves the detection and recognition of scenes, locations, activities, or objects within a video. This task is highly challenging due to the scarcity of annotated datasets and the high computational requirements involved. This thesis discusses the evaluation of recent State-of-the-Art (SOTA) Vison-Language (VL) models for video recognition on real world dataset to assess their usability for practical scenarios. Recently, some VL foundational models, trained on the large datasets were proposed with an aim to utilize them for various downstream tasks like image classification, image-text retrieval, video classification and video-text retrieval after zero/few-shot training or the fine-tuning. This evaluation will be carried out for the video recognition tasks, which is a crucial component of video understanding. This study will help to determine the effectiveness of these methodologies on data which corresponds closely to actual settings.

1.1. Background and Motivation

AI has undergone tremendous advancements in recent years, and its impact on industries is significant. In the early days of AI, the technology was limited to solving simple problems because of the lack of available data and limited computational power. Machine learning algorithms such as Linear Regression [5], Logistic Regression [6], Support Vector Machines [7], Decision Trees [8], K-means [9], K-nearest Neighbors [10], Random Forest [11] and AdaBoost [12], were used in these early days to harness the predictive power of the data available. However, with the advent of more powerful computing devices, it is now possible to train deep learning networks. These networks have a greater ability to recognize patterns and understand the underlying relationships within the data. These networks have an ability to process data from various modalities like audio, video, images and text. Different variations of these networks have been proposed, each with its own specific objectives like Convolutional Neural Network (CNN) [13], General Adversarial Networks (GANs) [14], Recurrent Neural Networks (RNN) [15], Long Short-Term Memory (LSTM) [16], and Transformers [17, 18]. These networks have enabled AI to move beyond simple problem-solving and into more complex applications, such as Natural Language Processing (NLP), video understanding, robotics and game playing.

Although, AI has made a significant progress in the field of NLP, its potential in vision systems has not been fully realized. In recent times, several Large Language Models (LLMs) have been developed using enormous language corpora, such as Bidirectional Encoder Representations from Transformers (BERT) [19] and Generative Pre-trained Transformers (GPT) [20]. These models have a general understanding of the language and can be fine-tuned for various downstream tasks, thus eliminating the need for extensive training and large labelled datasets. They have revolutionized the role of AI for many NLP tasks such as sentiment analysis [21], spam detection [22] and language translation [23]. Researchers are working towards adopting the success of LLMs to vision systems as well. The major aim for these vision models is to have the basic understanding of the scene with an ability to recognize objects, their contexts and relationships in images and videos. But so far, AI models lack the ability to interpret the vision data in the same way as humans do, due to the complexities of visual features such as intra-class variability, scale variability, occlusion, viewpoint variations, illumination changes, and more. These factors can make it challenging for models to accurately identify objects and understand their relationships within a scene.

Training large image encoders that can effectively generate representations for images is a hot research topic in the field of Computer Vision (CV). Similar to LLMs, Self-Supervised Learning (SSL) is becoming increasingly popular to train these image encoders, as it doesn't require any annotated datasets unlike traditional supervised learning methods. Self-supervised learning is referred to a method where model learns from the data points without relying on the labelled dataset. Although an enormous amount of vision data is available, the lack of annotations has forced the CV scientists to devise approaches independent of these annotations. Therefore, self-supervised learning is a way to go, where a model is trained to predict some part of an input based on other parts of the same input. Several techniques are used for this purpose such as pixelRNN [24] which is based on generative self-supervised method where the idea is to generate some hidden pixels of an image based on other pixels around them. Another method of SSL is Contrastive Learning (CL), where different tasks are assigned to the model to learn general features of the data. For example, predicting the relative positions of two image patches [25], predicting the color of grayscale images and orientation of the rotated images. Another self-supervised approach is Masked Language Modeling (MLM) [26, 27], where a masked word needs to be recovered based on image information and surrounding words. Image-Text Matching (ITM) [27] is also often employed, which measures the visual-semantic similarity between image and text. On the other hand, the Damaged Jigsaw puzzle [28] approach combines generative and contrastive techniques by dividing an image into several parts prior to colorization. The task involves colorizing the image, generating a missing part, and solving the puzzle. These approaches are typically employed to build background knowledge for the models and fine-tuning is required to make them usable for downstream tasks, such as object detection, classification and segmentation.

In recent years, several SOTA VL foundational models, inspired by the successful new architectures applied to large language models, have been proposed that aim to establish a connection between image's features and a natural language. These models, comprising millions of parameters, are trained on very large datasets using advanced techniques like SSL and attention mechanisms. These are usually capable of performing a variety of tasks like image/video classification, Visual Question

Answering (VQA), image/video captioning and image/video text retrievals. This not only reduces the need for labeled datasets but also speeds up the training procedure. These models are highly useful in situations where models need to be frequently retrained due to changing requirements or the addition of new classes. They offer a more efficient alternative to starting the training process from scratch. Although these models have demonstrated impressive results on a range of benchmark datasets, it is important to understand that these datasets may not accurately reflect real-world scenarios. One issue is that these datasets may have a narrow set of concepts, which may not be representative of the diverse and complex images and language used in real-world situations. This limitation may prevent the models from achieving optimal performance when applied to real-world data. Additionally, researchers tend to report the results of these models only for those datasets where the models perform exceptionally well. This approach can create a bias in the evaluation of these models as their performance on other datasets may not be impressive. Hence, it is challenging to determine the effectiveness of these models based only on a limited set of selective datasets.

The extension of pre-trained VL models to the video domain for understanding its content along the temporal axis has emerged as a significant area of research in both academic and industrial settings. With an enormous amount of video data being produced everyday, there is a need to develop automated techniques that can analyze and understand the content of these videos. Video classification forms the fundamental part of this process, involving the identification of activity, objects or events within the video sequence. Video classification has a range of applications such as surveillance, entertainment, sports analysis, and healthcare. Video classification is a challenging task due to the high dimensionality and variability of video data. The length of videos can vary greatly, and videos may contain multiple objects or scenes, each with different visual features. Moreover, usually videos are shot from different angles under varying lightening conditions and frame rates. Additionally, videos may have complex temporal dynamics that require sophisticated analysis techniques. So, these complexities make it very difficult to build generalized models that can accommodate all these variations. One the other hand, video datasets are often very large and complex, requiring significant computational resources and expertise to process and analyze. Despite these challenges, video classification has made significant progress in recent years, thanks to advances in deep learning and the availability of large-scale video datasets. The use of deep learning techniques like CNN [29], RNNs [15] and transformers [30, 31] have shown promising results, effectively capturing the temporal information.

The general pipeline of the video classification process is illustrated in Figure 1. Initially, frames are sampled from the video. These individual frames often undergo pre-processing steps to enhance their quality or extract relevant information. Next, video-level features are extracted from the frames. This can be done by either extracting features individually for each frame using an image classifier and concatenating them, or by employing a dedicated video extractor that directly captures video-level features. These features are then used to classify the video.

For this study, the evaluation has been conducted on three methods that have achieved SOTA results for video classification tasks using pre-trained foundational VL models. The selection of the three models for this study has been influenced by

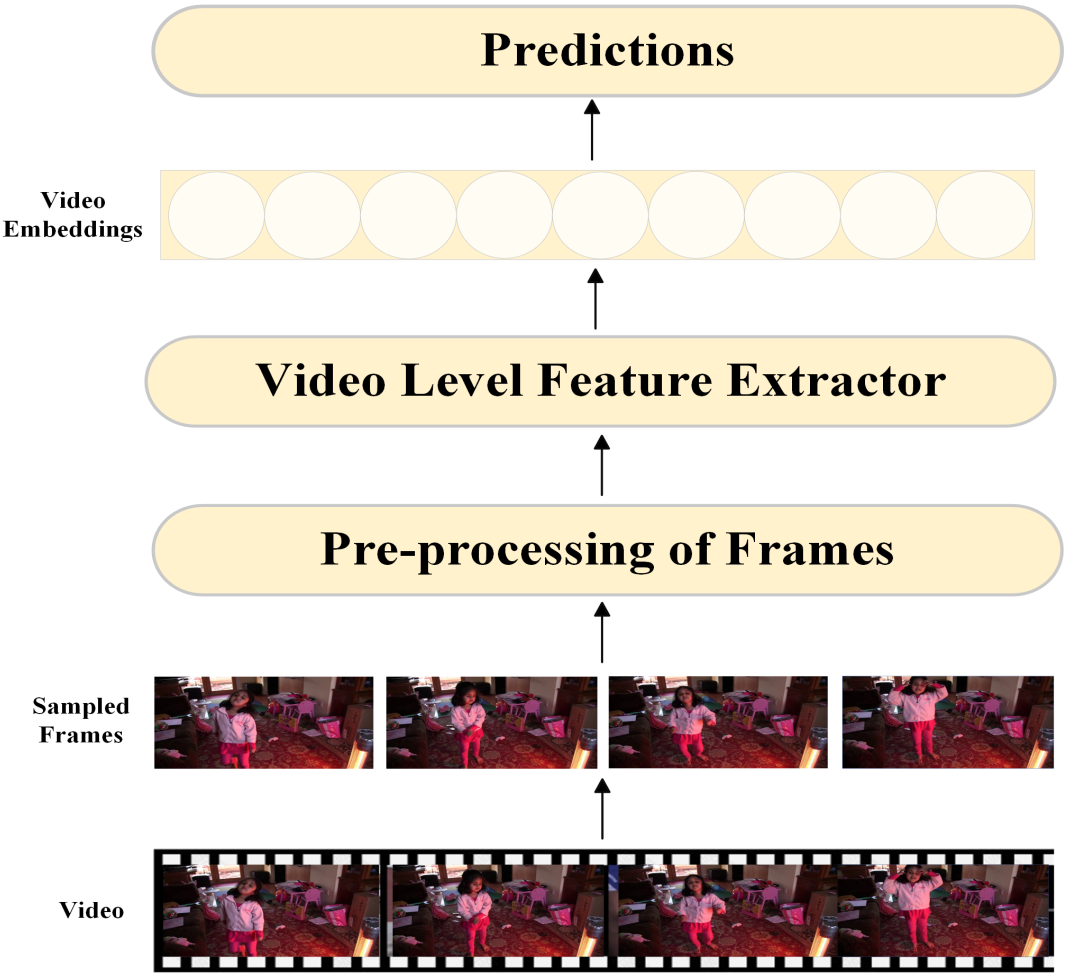

Figure 1. Pipeline of a video classification process

several factors, including their performance specifically in video recognition tasks, the availability of pre-trained models, and implementation details. To evaluate the performance of the VL models, YouTube 8 Million (YT8M) [4] dataset has been utilized. This dataset closely resembles real-world settings as it comprises 3872 categories, encompassing a diverse set of concepts. The videos included in the dataset are unprocessed and have been uploaded by users, which means they may not have been captured using professional cameras. Additionally, the videos within this dataset have longer durations, with an average of approximately 230 seconds. This extended duration provides a more realistic representation of the challenges that may arise in video classification tasks. The first of the shortlisted models is X-CLIP [2] which follows a strategy to directly adopt a pre-trained image model for video tasks by leveraging proposed Cross-frame Communication Transformer (CCT) and a Multi-Frame Integration Transformer (MIT) to adjust the temporal information. The second shortlisted model is Video Masked Autoencoders (VideoMAE) [1] which tries to learn the high level features by masking out the images at a very high ratio of 90-95% which allows it to be less computationally intensive and data hungry. The third of the short-listed model is Text4Vis [3] which tries to revisit the role of a classifier to transfer the knowledge of pre-trained VL models for video recognition

using classifiers' correlations. The goal of this study is to compare the performance of the aforementioned models and analyze how they perform on YT8M [4].

1.2. Objectives

The main objective of this study is to evaluate the performance of above mentioned models using YT8M dataset. In addition to assessing the classification performance of these models, this study also aims to investigate the number of training samples and training time required to achieve the competitive results. This assessment is very crucial as these models are designed to achieve good results with minimal training data and time, making them attractive for applications where resources such as training time and data may be limited. In some cases, acquiring and labeling data can be expensive and time-consuming. Additionally, firms may be hesitant to share sensitive data due to privacy and security concerns and limiting the amount of data that is available for training the models. On the other hand, many applications require frequent retraining of the model due to changes in business requirements or the addition of new categories. If the model requires too much time to train, it becomes impractical and can limit its usefulness. Therefore, it is crucial to consider training time and the amount of required data when assessing the effectiveness of models for practical situations. The findings of this study contribute to the understanding of the trade-offs between performance and resource requirements for these models, and provide valuable guidance for their adoption in real-world scenarios.

To broaden the scope of this study, it also aims to compare the performance of these fine-tuned models with those trained from scratch specifically for the YT8M dataset. These models are specifically designed to achieve excellent results for a given dataset, but to achieve this, they often require extensive training with large amounts of data. Therefore, comparing the performance of these fine-tuned models with those trained from scratch can help us understand the trade-offs between the two. To achieve this objective, classification results of the fine-tuned models will be compared with those of a Teacher-Student Network [32]. This approach is based on the Hierarchical-Recurrent Neural Networks [15], NetVLAD [33] and NeXtVLAD [34] models and is specifically designed to produce good results for the YT8M dataset. Overall, this objective will help us gain a better understanding of the effectiveness of using pre-trained VL models in comparison to training models from scratch.

1.3. Structure of the Thesis

This thesis is organized into five chapters, each serving a specific purpose. The first chapter serves as an introduction, providing background information on the topic and outlining the objectives of the study. The second chapter delves into a comprehensive discussion of related work conducted in the field before. It reviews existing literature that has explored similar or related topics. Moving on to the third chapter, it focuses on the methodology employed in the study. It provides a detailed explanation of the three foundational models used and describes the techniques, or frameworks utilized in their implementation. Additionally, this chapter delves into an extensive exploration of the

dataset used in the study. The fourth chapter of the thesis presents the experimentation process and the results obtained from the conducted experiments. The fifth chapter outlines the future work that can extend and build upon the findings of this study and concludes the whole thesis.

2. LITERATURE SURVEY

In the early days of CV, the developed techniques followed a closed-setting workflow, where models were trained for preset of categories. However, this methodology is almost impractical for situations where categories are not known in advance. Additionally, training large models to achieve good results requires extensive computational resources which may not be affordable for many firms. As a result, Contrastive Learning approaches have gained popularity for training foundational Visual-Language models, which can be used for many downstream tasks after fine-tuning or zero/few shot training. Zero shot learning refers to the utilization of a pre-trained model in a new domain or dataset without any additional training. In this approach, the model is applied directly to the new task without being specifically trained on the target data. On the other hand, few-shot learning involves training a new model using only a small number of examples per class. This approach allows the model to learn from a limited set of labeled examples and generalize its knowledge to make predictions on unseen instances. Much research has already been conducted in this area which is discussed in this chapter under the sections traditional vision models, Visual-Language pre-training and video learning.

2.1. Traditional Vision Models

Traditionally, CNNs have been the preferred building blocks for developing vision models due to their ability to effectively process spatial information in images. Typically, these models consist of series of convolutional and pooling layers to learn increasingly abstract and complex representations of an input image. In Figure 2, CNNs are depicted. Images undergo convolutional and pooling layers to reduce their size before being flattened and passed through fully connected layers for prediction. CNNs excel at capturing intricate patterns and structures in images, enabling accurate classifications. Over the years, many CNN-based SOTA models have been proposed to perform diverse set of CV tasks like VGG [35], Resnet [36], AlexNet [37], and Inception [38] for image classification, R-CNN [39], YOLO [40] for object detection and SegNet [41] for Image Segmentation. Although, some researchers have attempted to modify the general methodology used for CNNs by using self-attention to process the output of CNN [42], replacing spatial convolutions with a standalone self attention layer [43] and augmenting feature maps for image classification [44] among other techniques.

In contrast, neural networks based on the sequential mechanisms, such as RNN [15], LSTM [16] and Gated Recurrent Unit (GRU) [45] have not been as successful in learning the image features as CNNs. This is because they are designed to process sequential data and capture long-range dependencies, whereas in images, spatial information, local contexts and local relationships are more important. These properties make them well-suited for tasks involving natural language processing and time-series analysis. Nonetheless, some researchers have proposed combining RNN with CNN for various CV tasks such as, captioning images [46], predicting crop yield [47] and diagnosing COVID-19 [48] through Chest X-ray and CT images.

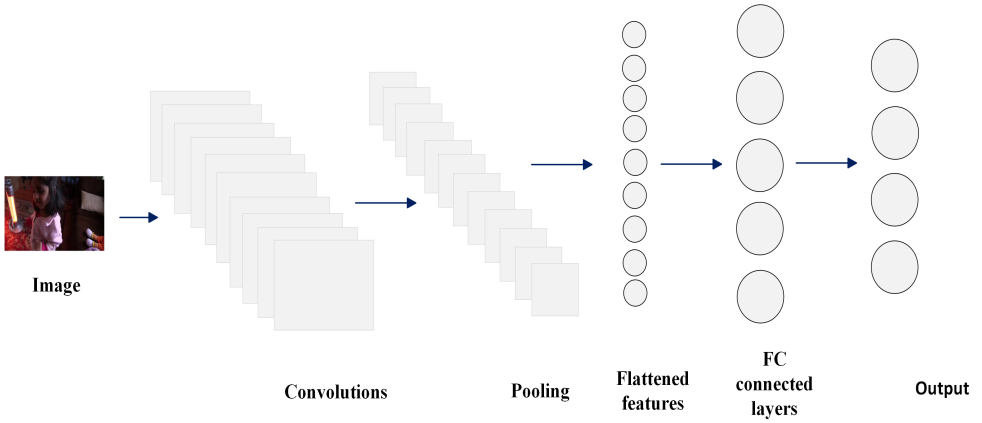

Figure 2. General working of CNN architecture

The race to develop better CNN-based models for vision modeling focused on building deeper models and training them with large datasets to improve their performance until the introduction of Vision Transformer (ViT) [18]. The success of transformers [17] for NLP prompted researchers to explore the use of transformers in vision models, revolutionizing the field with their ability to use an attention mechanism to selectively focus on important regions of an image and faster training. The ViT architecture breaks down an input image into a grid of non-overlapping patches, with each patch being 16x16 pixels in size. These patches are then flattened and fed into a series of transformer encoder layers, which process the patches and learn to capture the dependencies between them. The overview of proposed ViT [18] model is shown in Figure 3.

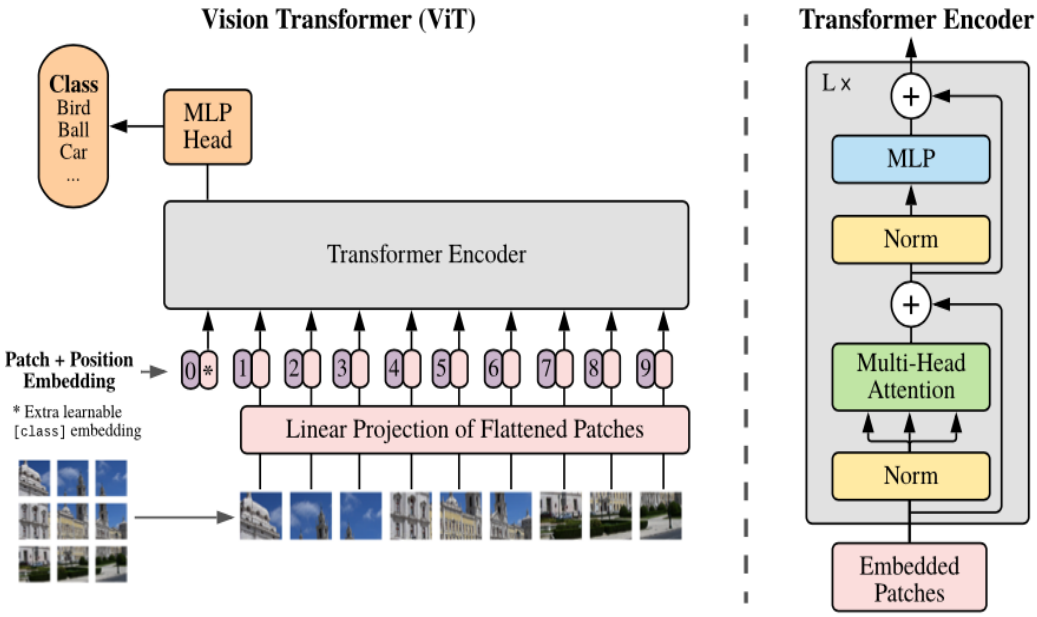

Figure 3. Vision Transformer [18] overview. Reprinted with permission

Since its inception, ViT has generated significant interests among the research community, leading to numerous derivative models. For instance, DeiT [49] has demonstrated that transformers can achieve good results even with medium-sized datasets through a distillation approach that uses a CNN-based teacher model to train the transformers. In contrast, standard ViT performs better only after pre-training with large JFT-300M [50] dataset. Another variant is Transformer in Transformer [51], which performs attention computation at two levels: patch-level, similar to conventional ViT, and local level by dividing the original 16x16 patch into 4x4 sub-patches. Over the past years, many variants of ViT have been proposed and have been successful in improving many traditional CV tasks. For example, CrossViT [52] learns multi-scale feature representations through cross-attention for image classification and the DETR [53] model, which eliminates the need for non-maximum suppression for object detection, unlike YOLO [40]. Furthermore, DPT [54] is a model designed for image segmentation tasks that employs ViT as the backbone. In this approach, tokens from various stages of ViT are converted into representations that resemble images, and then a CNN decoder combines these representations to make final predictions. Although these variations have demonstrated success in utilizing language transformers for vision tasks, the higher resolution of images compared to language may potentially limit their effectiveness. Therefore, Swin Transformer [55] aims to address this issue by forming hierarchical representations for images. It starts by breaking images into smaller patches and then merging neighboring patches in deeper layers. Overall, the use of ViT and its derivative models has opened up new avenues for research in vision modeling and improved the SOTA in many computer vision tasks.

2.2. Visual-Language Pre-Training

Visual-language pre-training using large general datasets in a self/semi supervised manner has emerged as a powerful technique for enhancing the ability of models to understand the connection between image and natural language. This pre-training stage enables model to get the general idea about the relationship between images and text. The learned representations of these models are then transferred to downstream tasks with either no training (zero-shot) or a light fine-tuning (few-shot) to adjust to new domains. In earlier methods, object detectors such as [39, 40] were used as image encoders to extract information about objects, which were then aligned with text using image-text pairs [56, 57, 58]. However, this approach has several drawbacks, including its inability to capture relationship between objects and its difficulty in listing all object categories that need to be detected.

One of the most representative example of VL pre-training is CLIP (Contrastive Language-Image Pre-training) [59] which employs image and text encoders to encode images and text, respectively. CLIP was trained using natural language supervision on 400 million image-text pairs extracted from the internet. The aim was to effectively link visual and language concepts, which cannot be achieved with SSL as it fails to create a linkage between the two modalities. During the training phase, the model maximizes the similarity of the encoded representations of each image-text pair, while minimizing the similarity between the representations of different image-text pairs. Since then, numerous follow-up works have been published, including

CLIP-Adapter[60] and Tip-Adapter [61] to improve few shot transfer, StyleCLIP [62] and StyleGAN-nada [63] for generating StyleGAN [64] images, CLIPCap [65] for image captioning, CLIP4CLIP [66] for video clip retrieval, CLIPasso [67] for object sketching, CLIP-Mesh [68] and ClipMatrix [69] for generating 2D and 3D textured meshes respectively from text, CLIP-Art [70] for fine-grained art classification, CLIP-Forge [71] for text-shape generation, CLIPort [72] for robotic path manipulation, Wav2CLIP [73] for learning audio representation from CLIP and many more. CLIP's success can be attributed to its ability to link up text and images in a joint embedding space, enabling it to generalize well across a wide range of tasks. Moreover, training CLIP on a diverse and unconstrained dataset has resulted in a highly versatile model with a vast vocabulary and a broad range of conceptual knowledge. In addition, pre-training on a diverse dataset also helps mitigate biases and constraints that may be present in standard public datasets.

In addition to CLIP, there have been other notable advancements in the field of visual-language pre-training. One such example is AliGn [74], which is based on a dual encoder architecture and is trained on a massive dataset of 1.8 billion image-alt-text pairs from the internet, utilizing a contrastive loss. For image and text encoding, AliGn uses BERTlarge [19] and EfficientNet [75], respectively, which are trained from scratch. The main difference between AliGn and CLIP lies in the dataset, as AliGn employs raw internet-sourced images and their associated alt-text pairs with minimal filtering. Florence [76] is another example of a dual-encoder model, utilizing transformer-based encoders for both images and text. It was trained on 900 million image-text pairs and has a unique approach in assuming that multiple images can be associated with a single text, which differs from CLIP's unique association approach. LiT [77] proposes a novel approach to VL pre-training by utilizing a pre-trained SOTA image encoder and training the text encoder using the dataset introduced in [74]. During training, the text encoder aligns with the pre-trained image representations. Unlike other methods that train both encoders from scratch, this approach offers a quicker training process while still benefiting from both data sources.

Although dual-encoder architectures [59, 74, 76] have shown promising results in VL tasks, they have a major limitation. They lack the ability to generate natural language text from images, making them unsuitable for tasks such as VQA and image captions. To address this limitation, SimVLM [78] uses an encoder-decoder architecture where the encoder encodes the image and a truncated version of the text, and the decoder decodes these encoding to generate text. SimVLM trained on a weakly labeled dataset of image-text pairs introduced in [74]. CoCa [79] takes a different approach by attempting to unify all three VL pre-training methods: encoder-only, dual-encoder, and encoder-decoder. The text decoder in CoCa [79] consists of two parts: the layers of first half uses a contrastive approach to encode the uni-modal text representations while the remaining layers also cross-attends to the image encoder's output to learn multi modal (image-text) representations using a generative loss. Instead of treating contrastive and generative approaches as a single pre-training phase, GIT [80] divides them into two separate sequential tasks. Firstly, it uses an image encoder from [76], which was pre-trained using a contrastive task. Afterwards, it pre-trains both image encoder and text decoder using a generative task on 0.8 billion image-text pairs.

Pre-trained VL models have been a significant leap towards building models with a strong global visual semantic understanding, enabling good results on many downstream tasks with zero/few-shot transfer. These tasks include image classification [59, 74, 76, 79, 80], text-image retrieval [59, 74, 76, 79], text-video retrieval [76, 79, 81], image captioning [79, 78, 80], VQA [78, 80, 81], and video captioning [79, 80]. The effectiveness of these models lies in their ability to learn from huge weakly labeled datasets, which provide rich semantics and diverse concepts. With these models, it is possible to capture the complex and subtle relationships between images and natural language, which enables them to achieve high performance on many downstream tasks without or with the minimal need for task-specific training data. As a result, these models are very powerful and offer a promising direction for future research in computer vision and natural language processing.

2.3. Video Recognition

Unlike image classification, which deals with static images, video recognition requires capturing both temporal and spatial information to detect different objects, activities, and events in a video sequence. Videos are essentially a collection of frames in a sequence, so image classification techniques can also be extended to classify videos. In the early years, researchers proposed methodologies to incorporate temporal information into CNNs for video classification [82, 83, 84] to capture the relationships between the sequence of frames. Another approach is to use LSTM in combination with CNNs as in LRCNs (Long-term Recurrent Convolutional Networks) [85]. In LRCNs, CNNs generate a fixed-length vector for each frame, which is then fed to the LSTM.

However, due to the success of transformers in decoding images, transformer-based approaches [31, 86, 87, 30] are also proposed for video tasks. These have shown relative success in capturing long-range dependencies. VTN [86] uses a temporal transformer encoder on the top of the frame level feature extractor network to encode the distant information in the video. Similarly, ViViT[31] has proposed several ViT [18] based models that process the tokens in spatiotemporal space. In contrast, [87] tries to process the videos' spatiotemporal space locally. Moreover, [88] proposed a two-stream network for video action recognition based on skeleton modeling, which captures the relationships between the different joints and their movements in the sequence of frames. Since these methods rely on supervised learning, they require annotated video datasets which are even more challenging to acquire than image datasets. Additionally, the annotations in video datasets are often limited in scope, which can result in models lacking a comprehensive understanding of the visual content. Furthermore, Videos are typically large in size, and training models on them requires high computational power, making it challenging to design video classification models that can be frequently retrained.

SSL methods offer a promising solution for leveraging vast amount of unlabelled video data to develop models with good video representations by exploiting the spatiotemporal information. There are various techniques used for learning these representations in an SSL manner. For instance, [89] learns spatiotemporal information by predicting whether the sequence of frames is in the correct temporal order while

[90] predicts the whole sequence of frames. In addition, [91] learns representations by predicting the clip order. [92] first performs Masked Image Modeling (MIM) to train the image encoder, and then simultaneously uses Masked Video Modeling (MVM) to train the video encoder. MIM and MVM are designed to capture spatial and temporal information, respectively. VideoBERT [93] employs automatic speech recognition systems to convert speech to text, which serves as the corresponding text pair. Pre-trained video classification model is then utilized to extract visual features from the video, followed by BERT [19] to learn joint distributions from these visual and language tokens. Alternatively, VideoMAE [1] extracts tokens using cube embeddings, which are then masked with a high ratio of approximately 90-95%. Based on an encoder-decoder architecture, VideoMAE [1] learns video representations by generating the masked out video tokens.

However, training video-based models is computationally expensive, balancing the effectiveness of learned video representations with computational requirements is crucial. Several methodologies have been proposed to address this, including using pre-trained general-purpose image models for video recognition. For instance, CoCa [79] calculates the mean over all frames for zero-shot transfer or fine-tunes only the attention pooler on top of the image encoder for better video representations for classification. In contrast, VideoCoCa [81] concatenates the frames embeddings obtained from the pre-trained CoCa[79] image encoder before feeding them to the attention poolers. On the other hand, [76] replaces the 2D CNN on top of image encoder with 3D CNN and duplicates the weights of pre-trained 2D CNN across temporal dimensions. These weights are then divided by the temporal kernel size to maintain the mean and variance. Alternatively, X-CLIP [2] utilizes pre-trained CLIP to encode frames and pre-defined labels. However, it employs CCT to exchange information between frames and the MIT model to generate video-level representations, by taking all the frames' encoding as input. Furthermore, it includes a video-specific prompter to enhance the text representations produced by the pre-trained CLIP encoder with the context of the video content. Text4Vis [3] takes a different approach by using a pre-trained textual encoder to extract the embeddings of class labels, which are used to initialize a projection matrix for the classifier head. During fine-tuning, only the pre-trained image encoder is retrained, while the weights of the projection matrix are kept fixed. To capture temporal information, Text4Vis [3] uses a temporal transformer to process the embeddings of frames. On the other hand, [94] utilizes bidirectional cross-modal knowledge for video recognition by first identifying the most relevant attributes for a video from a pre-defined lexicon (video-to-text) using a pre-trained CLIP textual encoder. Then, it calculates the frame-level saliency by measuring the similarity between the frames and class embeddings using pre-trained CLIP, which is used for producing video-level representations (text-to-video). Although, there has been a steady progress towards building effective video classification solutions, but it still hasn't been able to replicate the success of image classification

Table 1 displays the zero-shot classification outcomes obtained by different models for the popular kinetics-600 dataset [95] and Table 2 displays the models performance on the kinetics-400 dataset [96] after finetuning.

Kinetics-600 [95] dataset is extensively utilized in the research community for video recognition tasks as it includes 600 action classes, each containing at least 600 video

Table 1. Zero-shot transfer result obtained by different video recognition models on Kinetics-600

Model	Top 1 accuracy
VideoCoCa [81]	70.1
Text4Vis [3]	68.9
BIKE [94]	68.5
X-CLIP [2]	65.2

Table 2. Results obtained by different video recognition models on Kinetics-400 [96] dataset after finetuning

Model	Top 1 accuracy
CoCa [79]	88.9
BIKE [94]	88.7
Text4Vis [3]	87.8
X-CLIP [2]	87.7
VideoMAE [1]	87.4
Florence [76]	86.5
Swin-L [87]	84.9
BEVT [92]	81.1
ViViT [31]	80.0
VTN [86]	79.8

clips. VideoCoCa [81] has emerged as the leading model with an impressive accuracy of 70.1% for zero shot transfer result. This achievement is noteworthy as the model has not been trained on any domain-specific examples but instead generates classifications based on its general understanding of visual concepts and language. In a similar vein, the Kinetics-400 dataset, following the same data collection principles as Kinetics-600, is highly referenced. It comprises 400 action classes, each accompanied by a minimum of 400 video clips. It is evident from Table 1 that when tested on Kinetics-400 after fine-tuning on the same dataset, models pre-trained on large datasets such as COCA [79], Text4Vis [3], and Bike [94] demonstrate superior results compared to those trained from scratch, such as Swin-L [87], BEVT [92], and VTN [86], aligning with expectations.

3. METHODOLOGIES

In this study, three general-purpose VL methods that had achieved SOTA results for video recognition on various benchmark datasets have been evaluated. These models are tested on a dataset that is more diverse, large, and realistic than those commonly used in the literature, but surprisingly, its usage is relatively lower. Our study offers insight into the performance of three methods that uses pre-trained VL models for video recognition. This chapter will focus on providing background information on these methods and the dataset being used.

3.1. Dataset

The YT8M [4] dataset, published by Google, has been selected for this study and its latest version with classification annotations has been released in 2018. It remains the largest publicly available annotated video dataset with a total aggregate duration exceeding 0.35 million hours. This dataset consists of 6.1 million unique YouTube videos with each having at least 1000 views and a duration between 120 and 500 seconds. These videos are classified into 3862 classes which are arranged in the form of a knowledge graph with 24 top-level verticals. Although the annotations are machine-generated, they are highly reliable due to the integration of user engagement signals, content analysis, and video metadata. As the data of such scale would require an enormous amount of storage space, so frames were extracted with 1-FPS (Frame per Second) and vectorized using the Inception network [97], trained on Image-Net, to compress storage requirements.

The YT8M dataset has been chosen for its close correspondence to real-world settings. Unlike many public datasets that capture videos in controlled environments [98, 99] or perform video stabilization measures [100], the videos in YT8M are uploaded by random users on YouTube, making them diverse and representative of real-world scenarios. These videos are not necessarily produced by professionals using high-quality cameras or stage actors, which can result in noisy, low-quality videos with irregular camera movements. This diversity yields a wide range of video content, including different types of scenes, objects, actions, and backgrounds, as well as different camera angles, lighting conditions, and audio quality. Many recent datasets focused solely on actions [96, 95, 100, 101] while neglecting other themes and types of events that can occur in a video. However, YT8M contains not only actions but also events, objects, and scenes, making it more suitable for developing models that can detect all of these. Additionally, other datasets typically extract the clips that represent a specific class from the video, but in real-world scenarios, a video can also contain other themes that can affect the context of the action being performed. YT8M maintains the original content of the video, meaning that the developed methodology must learn to predict in the presence of other concepts. The mean duration of videos in YT8M is 230 seconds, whereas it is only 10 seconds for the kinetics dataset [96, 95] and 7.21 seconds for the UCF101 dataset [101]. Although the YouCook [102] and Sports-1m [103] datasets have longer mean duration of 315 and 336 seconds, respectively, these datasets are domain-specific, containing only sports

and cooking videos, respectively. Table 3 shows the detailed comparisons between public video datasets.

Table 3. Comparisons of public video datasets

Dataset	**environment**	**Number of Categories**	**category types**	**Average Duration (seconds)**
Kinetics-400 [96]	real	400	actions	10
Kinetics-600 [95]	real	600	actions	10
UCF101 [101]	real	101	actions	7.21
YouCook [102]	real	captioning	cooking recipes, cooking styles	315
Sports-1m [103]	real	487	sports type	336
KTH [98]	controlled	6	actions	4
Weizman [99]	controlled	9	actions	-
HMDB 51 [100]	controlled (post-processed)	51	actions	-
Youtube8M [4]	real	3862	actions, venues, objects, events	230

Due to the computational and time limitations of this study, a subset of the YT8M dataset was selected for experimentation. A total of 25 categories have been selected for this study, namely concert, dance, driving, drawing, gardening, soldier, sewing, restaurant, knitting, wedding dress, police officer, walking, injury, smoking (cooking), human swimming, ice skating, festival, stadium, desert, beach, snow, video game, cooking, cycling, and racing. The selection of classes for this study is diverse, encompassing not only actions and objects but also locations, venues, and events. This approach allows for a more comprehensive evaluation of the model's ability to accurately predict multiple types of information simultaneously. In addition to category selection, storage and duration constraints were also applied to limit the size of the dataset. Specifically, only videos with a storage size of less than 30 MB and a duration of less than 4 minutes were included in the subset. Initially, the YT8M dataset provided frame-level features instead of complete videos. Therefore, the videos were downloaded using the unique YouTube ID associated with each video. It is worth noting that the original videos were multi-labeled, with an average of three labels per video. However, for the purposes of this study, only single-label classification was of interest. Therefore, only those videos containing labels only from one of these 25 selected categories were included in the final subset. In other words, videos containing labels from more than one of these 25 selected categories are ignored.

To conduct this study, it was necessary to limit the number of videos for each category to ensure a manageable storage size while still ensuring that there were sufficient samples for analysis. Therefore, upper limit for the number of videos for each category was set to 701, and these videos were deemed sufficient for this study as the primary objective is to evaluate the samples required to produce good

results. This was a crucial consideration, given that the total storage size exceeded 180 GB, excluding the test set. The size increases significantly when videos are stored as individual frames, a common practice for faster training, which further adds to the storage requirements. Figure 4 displays the number of videos for each category, indicating that the data is imbalanced with categories such as "injury", "smoking (cooking)", and "walking" having relatively fewer instances (50, 68, and 105, respectively). These categories with few samples were deliberately included in the study to examine their impact on model performance as it can provide insights into the generalizability and robustness of the model to handle imbalanced data. Nonetheless, other classes have sufficient instances for analysis.

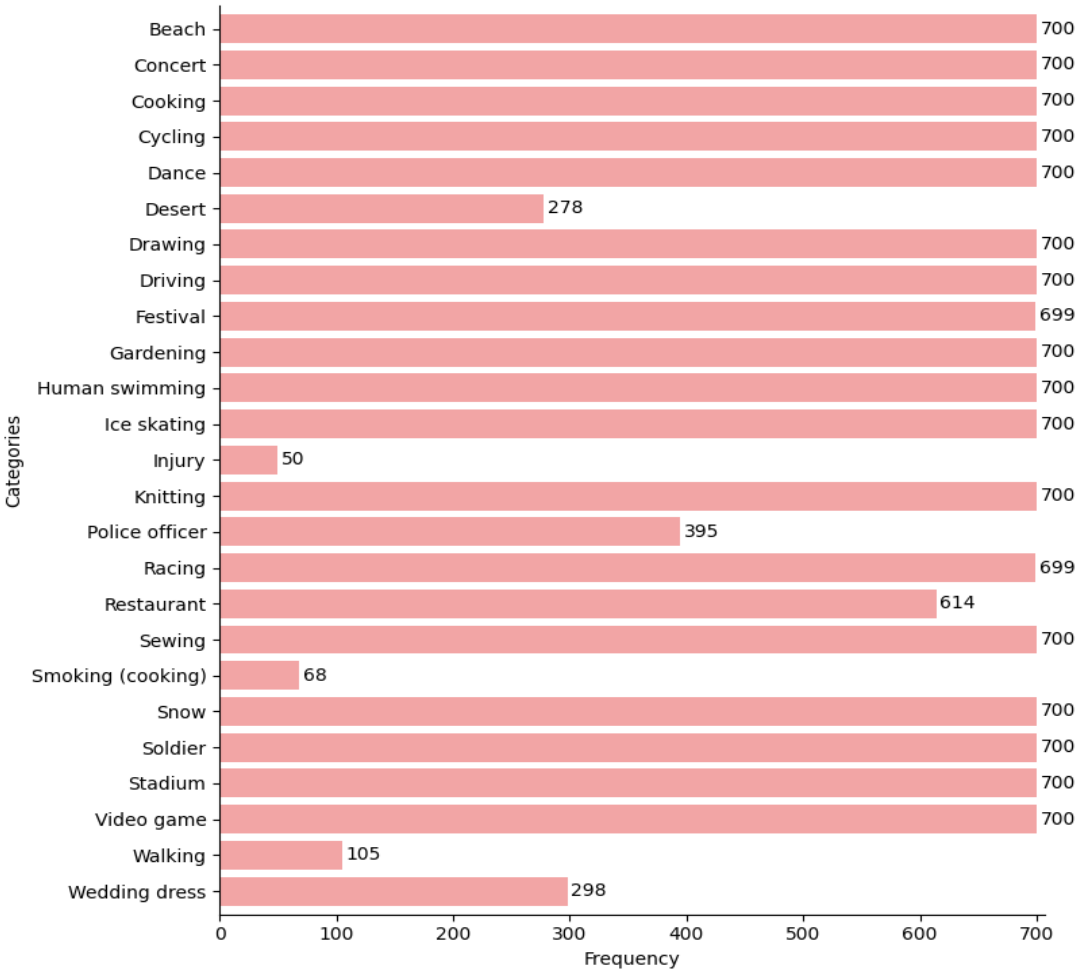

Figure 4. Number of videos for each category in train+validation sets

The YT8M dataset includes three sets of data, namely the train, validation, and test sets. However, the test set labels are not available publicly, as they are reserved for evaluating models submitted to a competition. Therefore, in this study, the validation set was used for testing, while the training set was further divided into two subsets with ratio 0.8:0.2. From now on, the terms training, validation and test sets refer to these newly divided datasets. The distribution of the number of videos for each category in the testing set is depicted in Figure 5, with an upper limit of 1000 videos per category. It is important to highlight that the same filters used to select videos

based on duration and size in the training set were also applied to the testing set. This approach helps ensure that the evaluation of the models' performance on the testing set is conducted under comparable conditions and provides a fair assessment of their generalization capabilities. However, the limit for the testing set was set higher to account for the fact that evaluating the model during testing requires less computational resources than training. Additionally, a large testing set is more likely to cover a broader range of variations that typically occur in real-world videos, such as changes in lighting conditions, camera angles, occlusion, and background clutter. Therefore, a large testing set can help to assess the generalizability of the model in handling these variations.

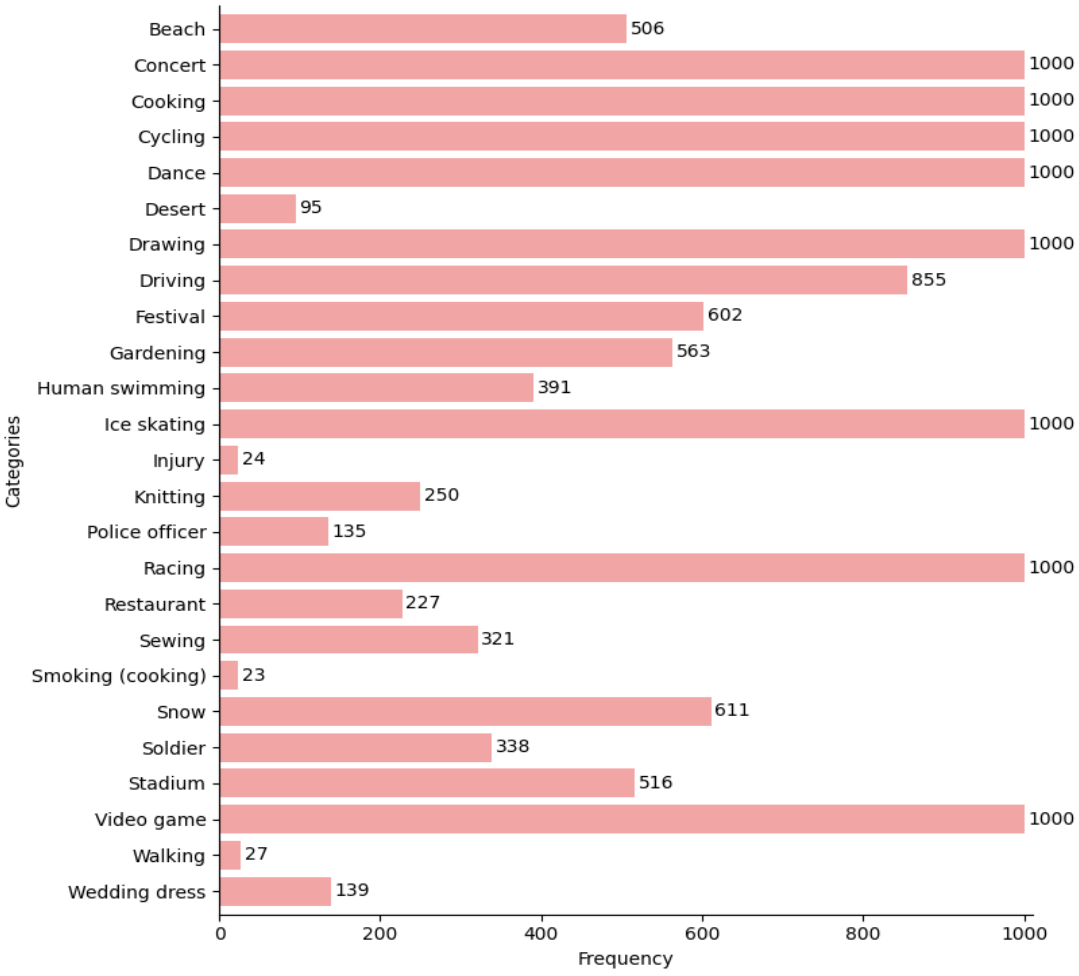

Figure 5. Number of videos for each category in test set

Table 4 table displays the key statistics of the combined training and validation dataset. The training and validation set contains 14,406 videos, with a total duration of 684 hours, and an average duration of 171 seconds per video. Similarly, the test set contains 13,623 videos, with a total duration of 652 hours, and an average duration of 172 seconds per video. These durations are adequate for the research purposes and allow models to capture the patterns present in the data. Furthermore, Figure 6 and Figure 7 depict the distribution of durations for the training and test sets, respectively. Both graphs show an almost even distribution of video durations. The

Figure 8 and Figure 9 provide a visual representation of the category-wise average durations for the training and testing sets. The data presented in these figures reveal that the average durations of videos for all categories are almost similar. This implies that the duration difference between different categories is not substantial enough to potentially impact the results of the study. Overall, the longer average duration of videos in this dataset is advantageous, as it provides more contextual information. This enables models to capture long-term dependencies and learn complex temporal relationships between actions and objects in a video. On the other hand, the total views for the training and testing sets are 721 million and 731 million, respectively. More views indicate that these videos have been watched and shared by a significant number of viewers, representing a broader range of user preferences, content types, and video characteristics. In general, these views do not affect the modeling process or the results.

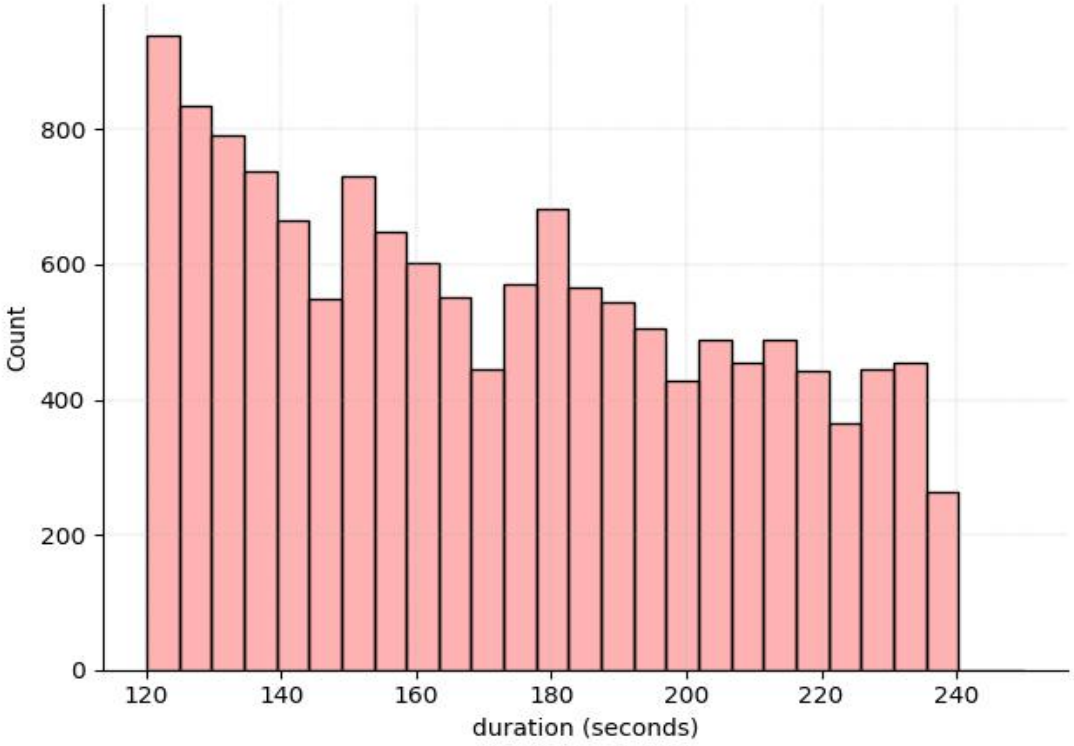

Figure 6. Duration distribution in train+validation sets

Table 4. General statistics of the dataset

	Train+val	**test set**
Total number of videos	14,406	13, 623
Total duration (hours)	684	652
Average duration (seconds)	171	172
Storage size (GB)	180	172
Total Views (million)	721	731
Average number of views per video	50149	53834

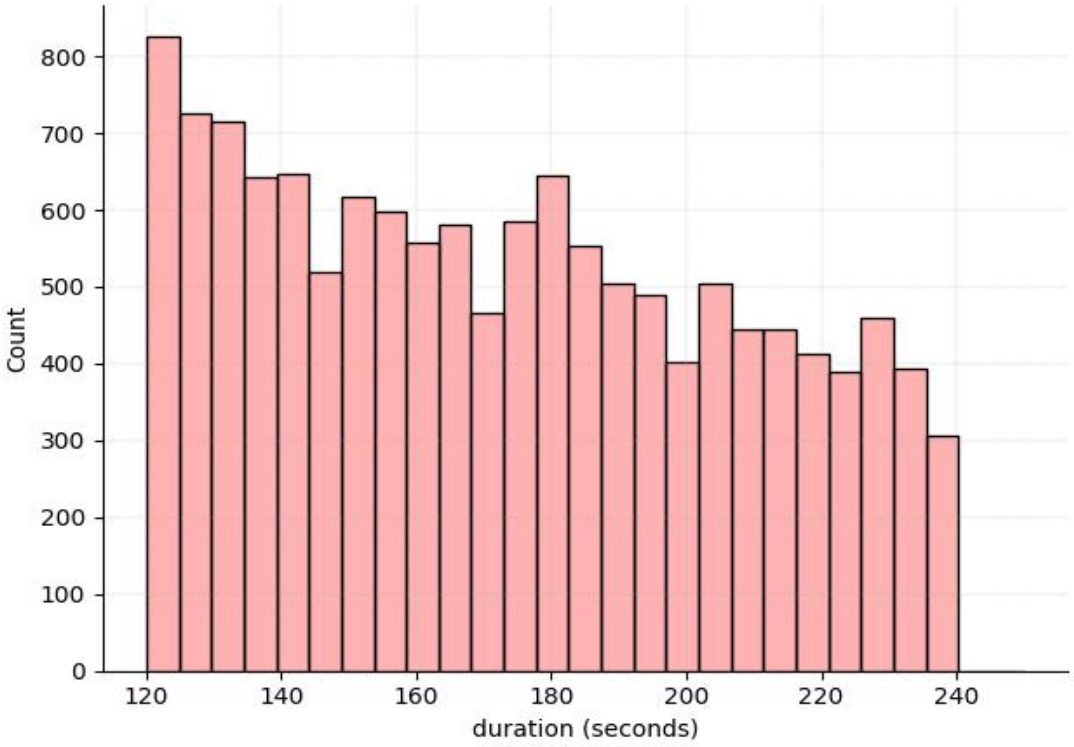

Figure 7. Duration distribution in test set

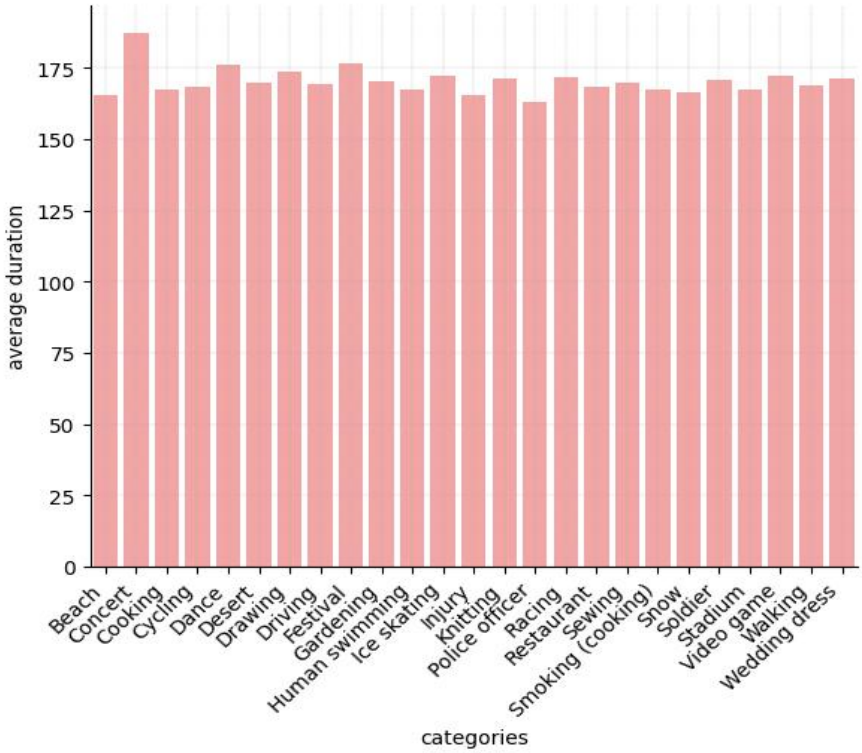

Figure 8. Category-wise distribution of average duration in the train+validation sets.

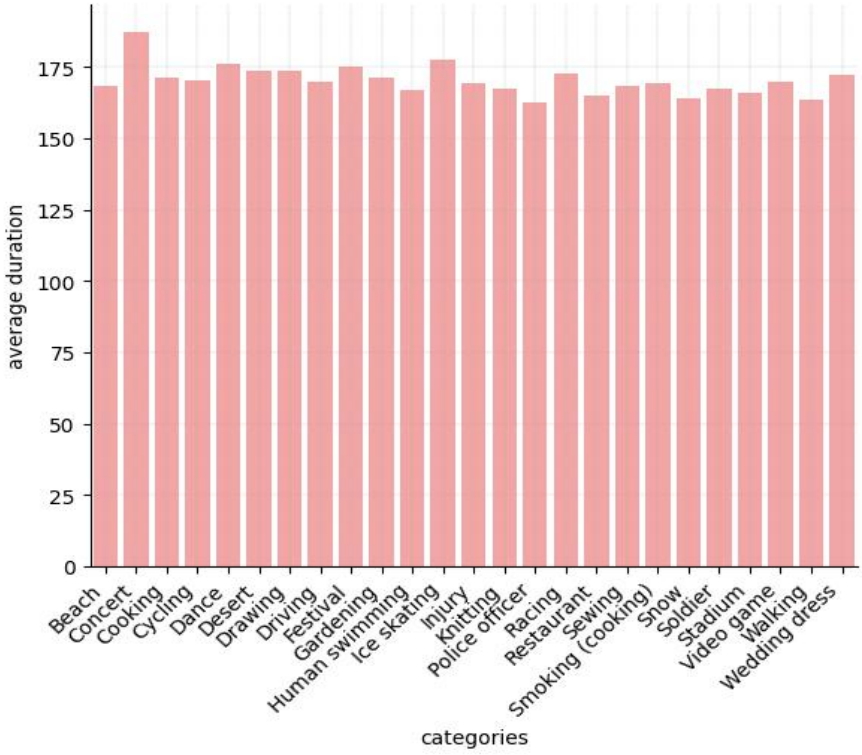

Figure 9. Category-wise distribution of average duration in the testing set.

To investigate the impact of the amount of training data on the performance of the models in this study, four subsets have been created from the original training data with fractions of 0.1, 0.3, 0.7 and 1.0. Then each fraction of the data is divided into training and validation sets using a ratio of 0.8:0.2. Each model have been trained separately on each of them. The instances for each subset were randomly selected by taking the given fraction of data from each category. However, this random selection could result in very few samples for under-sampled categories, such as "Injury" which only has four instances for training in 10% of the data. Nonetheless, it will be interesting to observe how the model performs when instances for some categories are very low and how it affects the overall performance. This is particularly important since collecting data for some categories may be more challenging than others. By studying the impact of varying amounts of training data on the models' performance, this study will help determine the optimal amount of data required to achieve good results.

Overall, the research study has been designed with a well-organized dataset that closely resembles real-world scenarios. The dataset contains a sufficient amount of data with evenly distributed durations and includes selected categories belonging to different events, actions, objects, and scenes. This makes it suitable for evaluating the model's ability to recognize various group of classes. The use of different fractions allows for analyzing the trade-off between data size and model performance, which is crucial in practical applications where data collection and labeling can be costly and time-consuming. Overall, the dataset represents a challenging task, given its large size and imbalanced nature. Therefore, the study's results will provide valuable insights into the models' performance

3.2. Models

To conduct this study, three models were selected based on their SOTA performance in video recognition tasks across various public datasets. In addition to their performance, the availability of pre-trained models and training code for few-shot transfer was also considered. This ensures that the models can be easily trained on the dataset used in this study without requiring extensive time and expertise. The three models chosen for this study are VideoMAE [1], X-CLIP [2] and Text4Vis [3]. These models serve as foundational VL models that can be fine-tuned for numerous downstream tasks. In the following section, a brief overview of each model's working methodology is provided.

3.2.1. VideoMAE

Inspired by ImageMae [104], VideoMAE [1] uses video masked autoencoder for Self-Supervised Video Pre-training (SSVP). The aim of VideoMAE is to learn spatiotemporal visual features by employing a method that involves a high masking ratio for videos. This ratio is set at 90-95%, surpassing the 75% used in ImageMAE [104], as videos exhibit temporal redundancy and correlation. By employing a high masking ratio, VideoMAE achieves faster training times while also demanding fewer computational resources. However, challenges arise from the slow propagation of semantic information across frames [105], which risks easy recovery of missing regions through neighboring spatiotemporal features, potentially leading to sub-optimal learned representations. Moreover, without a specific strategy, a masked object in one frame may be unintentionally unmasked in subsequent frames, potentially causing information leakage during the reconstruction process. VideoMAE generates these masked cubes by utilizing the encoder-decoder architecture.

The framework begins by down-sampling video frames to eliminate repetitive information, which can potentially improve pre-training. Since consecutive frames in a video often contain redundant information, preserving the original frame rate is not efficient. By down-sampling, the framework aims to retain important visual features while reducing computational complexity. Next, cubes are extracted along the spatiotemporal axis, with each cube measuring $2 \times 16 \times 16$ in size. These cubes are then passed through a cube embedding layer, which maps them to a D-dimensional space. To promote effective learning of semantic information over purely spatiotemporal features, the cubes are masked using a strategy where all frames share the same masking map. This approach helps prevent information leakage for objects with minimal or no motion. By forcing the model to reason based on semantics, the masking strategy encourages the utilization of high-level spatiotemporal features. Regarding the encoder-decoder architecture, the high masking ratio results in very few input tokens available for the reconstruction phase. To address this, ViT [18] has been employed as a backbone. ViT leverages spatiotemporal attention [31, 87] to capture meaningful interactions between tokens. The use of multi-head attention enables comprehensive token interactions, facilitating the reconstruction process.

3.2.2. X-CLIP

X-CLIP [2] proposes a methodology that leverages pre-trained image models to tackle video-related tasks by effectively incorporating temporal information. Pre-training image-based VL models can be accomplished more efficiently and effortlessly thanks to the availability of extensive public image datasets, and the inclusion of web-sourced images further expands the potential applications. Utilizing pre-trained image models offers a significant advantage by circumventing the high costs associated with training models from scratch using video datasets. The process of collecting and annotating large-scale video-text data is both challenging and time-consuming, and training models on such data requires substantial computational resources. In contrast, leveraging pre-trained image models provides a more practical alternative. These models have already been pre-trained on vast image datasets, enabling them to effectively capture VL features and their relationships. Consequently, adapting these pre-trained models to handle videos becomes a less complex and resource-intensive task. With the ability to leverage the existing knowledge and representations learned from images, the proposed methodology provides a practical and efficient approach of extending pre-trained models for video recognition.

The design choices of X-CLIP are driven by two key challenges in adapting image models to the video domain: how to incorporate temporal cues? and how to transform text representations for videos? The proposed methodology addresses the first challenge through the utilization of two components: CCT and MIT, both of which rely on a multi-head self attention mechanism. The CCT module facilitates the integration of temporal information by enabling information exchange between frames through the use of message tokens. Within CCT, two attention mechanisms are employed: cross-frame fusion attention (CFA) and intra-frame diffusion attention (IFA). CFA leverages all message tokens, derived from the linear transformation of a learnable class token, to obtain global visual representations. On the other hand, IFA takes frame tokens and associated message tokens as input to learn visual features. Notably, the associated token is removed prior to passing the data to the Feed-Forward Network. CFA is initialized randomly, whereas IFA utilizes pre-trained weights to enhance its performance. The MIT module takes the frame-level representations as input and generates video-level representations. By considering information from multiple frames, the MIT enables a more comprehensive understanding of the visual content, enhancing the model's ability to analyze videos effectively.

To effectively represent text in the context of videos, the X-CLIP framework employs the video-specific prompting (VSP) module. This module utilizes the video-level representations to generate video-specific text prompts, enabling the model to better understand and interpret textual content within the visual context provided by the video frames. In the pipeline, the text is first encoded using a text encoder pre-trained with an image-based VL model. The resulting text representations are then passed through the VSP module to obtain embeddings specifically tailored for videos. To determine the most probable outcome, the similarity between the category features and the video-level representations is calculated.

An overview of the X-CLIP pipeline is depicted in Figure 10, illustrating the sequence of operations. Initially, the frame patches of the video frames are inputted into the CCT, followed by the MIT. Meanwhile, the text labels associated with the

video are passed through a text encoder, and the resulting representations are refined using the VSP module to obtain instance-level label representations customized for the video. It is worth noting that the VSP and MIT modules are initialized randomly, while the CCT module benefits partially from pre-trained initialization. This method achieves the best results when utilizing CLIP [59] as the pre-trained model.

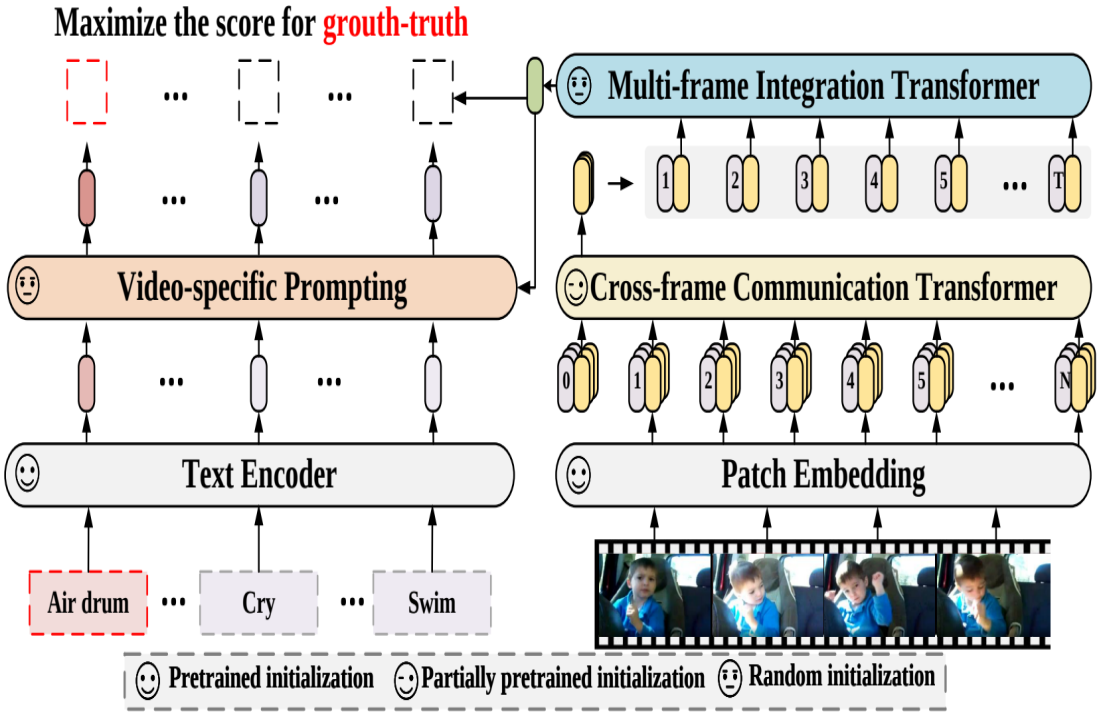

Figure 10. Overview of the X-CLIP [2]. Reprinted with permission

3.2.3. Text4Vis

The Text4Vis[3] approach aims to redefine the role of a classifier in transferring the pre-trained visual and textual models for video recognition tasks. Instead of training a classifier head from scratch, the approach initializes the classifiers' weights using the textual and visual knowledge from pre-trained models and keeps them frozen during the fine-tuning process. The classifier head is represented by a projection matrix of size $d \times c$, where d represents the dimensionality of feature vectors and c represents the number of classes. The projection matrix is used to compute the logits for the feature vectors. During the training process, only the visual encoder is fine-tuned, with the objective of adjusting its weights to align with the weights of the projection matrix. This is illustrated in Figure 11, which compares the proposed Text4Vis paradigm with existing paradigms. In Figure 11(a), the standard vision model is shown, where both the encoder and the classifier head are trained. In Figure 11(b), the dual encoder architecture paradigm is depicted, where both the visual and textual encoders are fine-tuned to map to a similar space.

To explore various initialization methods for the projection matrix, several approaches were examined. The first method involved random initialization of

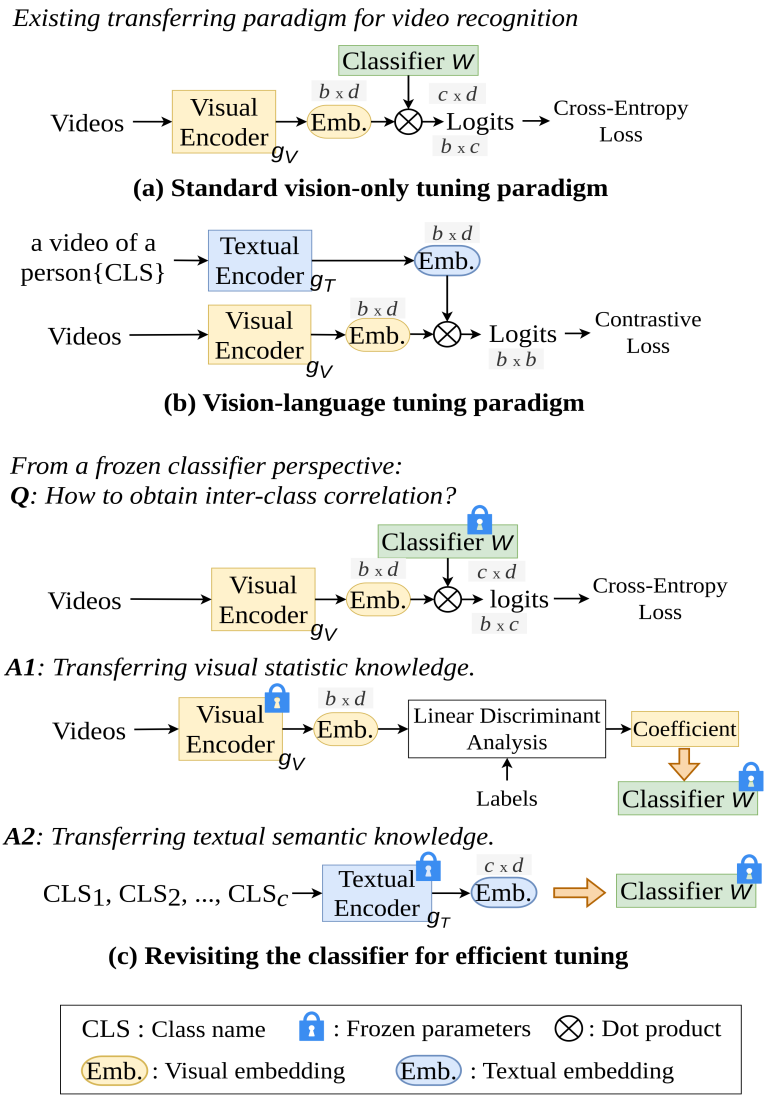

Figure 11. Comparing the methodology of Text4Vis [3] with other methods. Reprinted with permission

the matrix, resulting in trivial correlation. In the second method, orthogonal row vectors were obtained by removing the correlation from a randomly initialized matrix, ensuring no correlation among the rows. The third method utilized the visual knowledge of the pre-trained encoder through linear discriminant analysis (LDA). This approach entailed extracting encoded features from all training samples using a pre-trained encoder and then employing LDA to learn coefficients that optimize inter-class covariance while minimizing intra-class covariance. This method, known as maximal correlation, can be visually understood through Figure 11(c:A1). The final method capitalized on the textual knowledge of a pre-trained textual encoder to initialize the projection matrix, utilizing textual embeddings of the labels. During experimentation, textual embeddings were extracted using both the pre-trained encoders of CLIP [59] and DistilBERT [106]. This process is depicted in Figure 11(c:A2). To incorporate temporal information, a temporal transformer was employed.

Experimental results demonstrate that the best performance is achieved when the projection matrix is initialized using the pre-trained textual encoder. For the first two methods, the classification accuracy achieved on the kinetics-400 [96] dataset was 59.3% and 59.4%, respectively. Initializing the weights arbitrarily did not yield satisfactory results. However, by using the visual encoder, the accuracy improved significantly to 80.8%. As for the textual encoder, both DistilBERT [106] and CLIP achieved almost similar accuracies of 81.4% and 81.5%, respectively. This similarity arises from the substantial textual knowledge embedded in both models, as they have been trained on extensive datasets. Overall, Text4Vis introduces a novel approach to train VL models for video recognition tasks. Instead of focusing on fine-tuning the classifier head, Text4Vis [3] fine-tunes the visual models themselves.

4. EXPERIMENTATION

This chapter provides a comprehensive overview of the experiments conducted using three selected VL models: VideoMAE [1], X-CLIP [2], and Text4Vis [3]. The experiments were carried out on four distinct subsets of the original data, representing different proportions of the entire dataset while testing data reamians the same. Specifically, the subsets consisted of 10%, 30%, 70%, and 100% of randomly sampled instances from the selected training data (for more information, refer to Section 3.1). The rationale behind using multiple subsets was to investigate the performance and behavior of the VL models across varying degrees of data availability. During the experiments, various metrics have been collected to evaluate the performance of the VL models. These included accuracy, precision, recall, and F1-score, among others, which has allowed to assess the models' classification prediction capabilities. The F1-score will act as a primary metric because it combines precision and recall, providing a balanced evaluation of a model's performance. The implementation details and the obtained results for each model are discussed in their respective subsections.

4.1. Environmental Setup

This project was developed using Python as the programming language, with Ubuntu chosen as the operating system, and an Nvidia Titan XP GPU utilized to expedite the training process. Python was chosen for its widespread popularity in the AI community and its robust online support, which makes it easier to find solutions to problems. Additionally, Python has a broad range of built-in functionality, making it an excellent choice for deep and machine learning tasks.

To take advantage of Python's extensive functionality, a range of Python packages were employed in this project. Tensorflow, Pytorch, and Transformers are used as deep learning frameworks since they provide a rich set of tools and APIs for building and training neural networks. Matplotlib and Seaborn were employed for creating visualizations and plots, while Pandas and Numpy were used for data manipulation purposes. Finally, Decord is a package that was used for reading videos, a crucial requirement for this project.

4.2. Results

In this section, a comprehensive exploration of the experiments conducted using the VideoMAE [1], X-CLIP [2], and Text4Vis [3] models is provided. Each model is examined individually, with dedicated subsections that encompass the implementation details and the corresponding results obtained from the experiments.

4.2.1. VideoMAE

In implementing this methodology, a pre-trained model trained on kinetics-400 [96] was utilized and subsequently fine-tuned using the YT8M dataset. As a crucial pre-

processing step, the frames extracted from each video were stored locally, with one frame captured every 2 seconds. Considering the average video duration of 171 seconds, capturing frames at the original frame rate would have resulted in a large number of frames, potentially leading to redundant information and increased storage requirements. The frame extraction process was performed beforehand, as conducting it during runtime would have imposed a higher computational overhead, especially considering the need to conduct multiple experiments.

For the fine-tuning stage, a uniform sampling strategy was employed, selecting 16 frames from each video. Following the frame selection, the chosen frames underwent a series of pre-processing steps. Firstly, normalization was applied to ensure consistent and reliable data representation by standardizing the pixel values across the frames. Additionally, random crop and horizontal flip techniques were employed to prevent overfitting and enhance the model's robustness. These techniques introduced diversity in the training data, allowing the model to generalize better to different visual scenarios.

The detailed results for a model that has been fine-tuned using only 10% of the data for 50 epochs are presented in Figure 12.

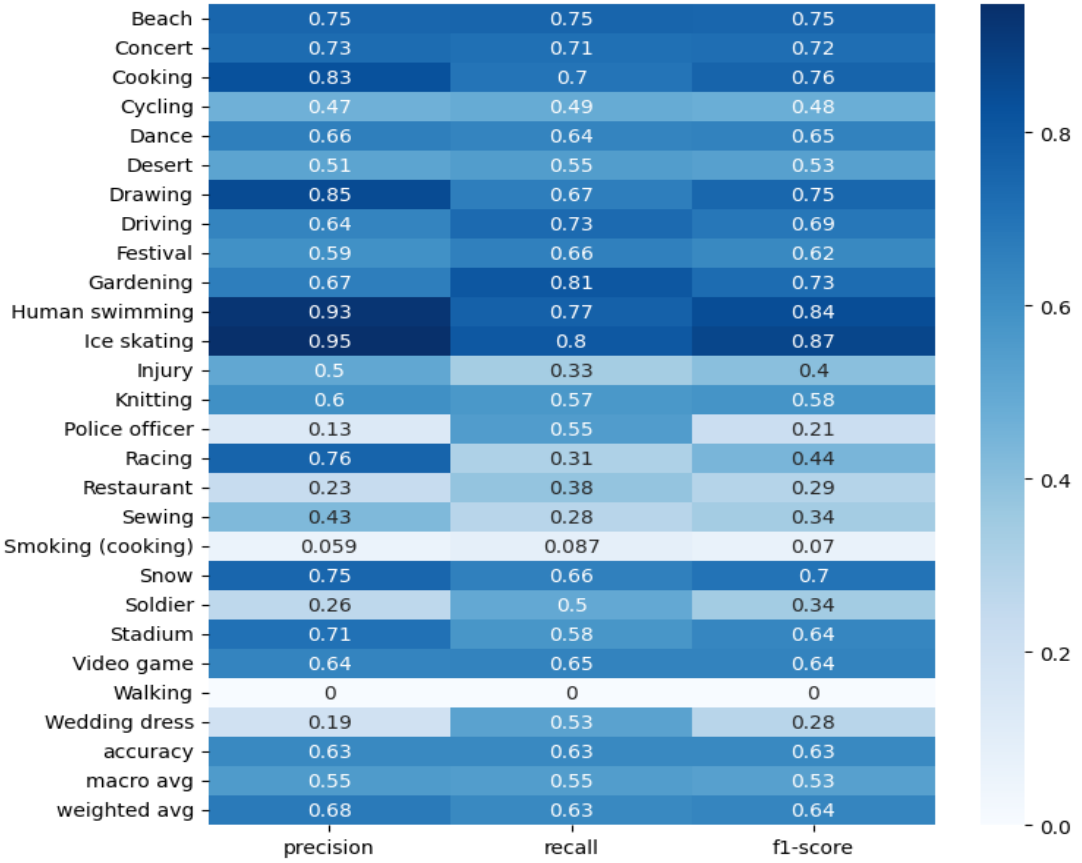

Figure 12. Results of VideoMAE on test data when fine-tuned on 10% of the data

This subset of data comprises an average of 46 instances per class. However, it should be noted that certain classes, such as "injury", "walking", and "smoking (cooking)", have significantly fewer instances, with 4, 4, and 7 instances respectively.

The results reveal some challenges, particularly for the minority classes. The weighted average F1-score of 0.64 indicates an overall performance that is relatively low, with "walking" not being predicted at all. Additionally, other minority classes of "injury" and "smoking (cooking)" have also obtained low F1-scores of 0.4 and 0.007 respectively. These results suggest that the model faces challenges in accurately classifying instances from these specific classes due to their limited representation in the dataset. But some other classes like Soldier and Wedding dress also performs poorly.

The results obtained from fine-tuning the model for 50 epochs using 30% of the available data are depicted in Figure 13. In this subset, the average number of instances per class are 138. Increasing the amount of data has led to a slight improvement in the weighted F1-score, which now stands at 0.69. However, considering that the data provided has tripled, the improvement remains relatively low. Similar to the previous experiment, classes such as "injury", "walking", "smoking (cooking)", "wedding dress", "police officer and "soldier" continue to exhibit poor performance. These classes struggled to be accurately classified despite the increase in data. On the other hand, classes like "beach", "human swimming", and "ice skating" demonstrate promising results, indicating that the model is able to effectively distinguish instances belonging to these categories.

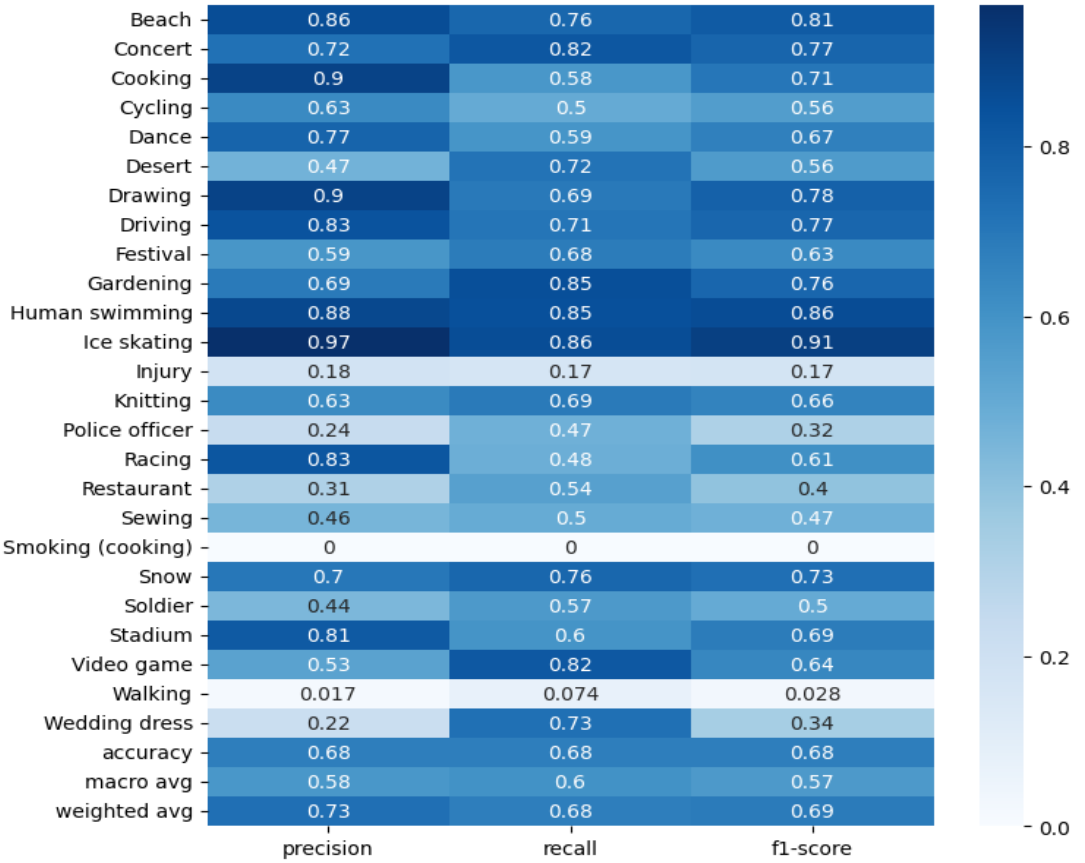

Figure 13. Results of VideoMAE on test data when fine-tuned on 30% of the data

The results achieved after fine-tuning the model using 70% of the available data are visualized in Figure 14. In this subset, each class is represented by an average of 322 instances. Notably, there is an improvement in the weighted F1-score, which now stands at 0.75. Although the performance has increased compared to the previous experiments, it is still considered unsatisfactory. Unfortunately, the classes that exhibited poor performance in the earlier experiments continue to demonstrate relatively low results in this scenario as well. Despite the larger amount of data provided, these classes struggle to be accurately classified by the model.

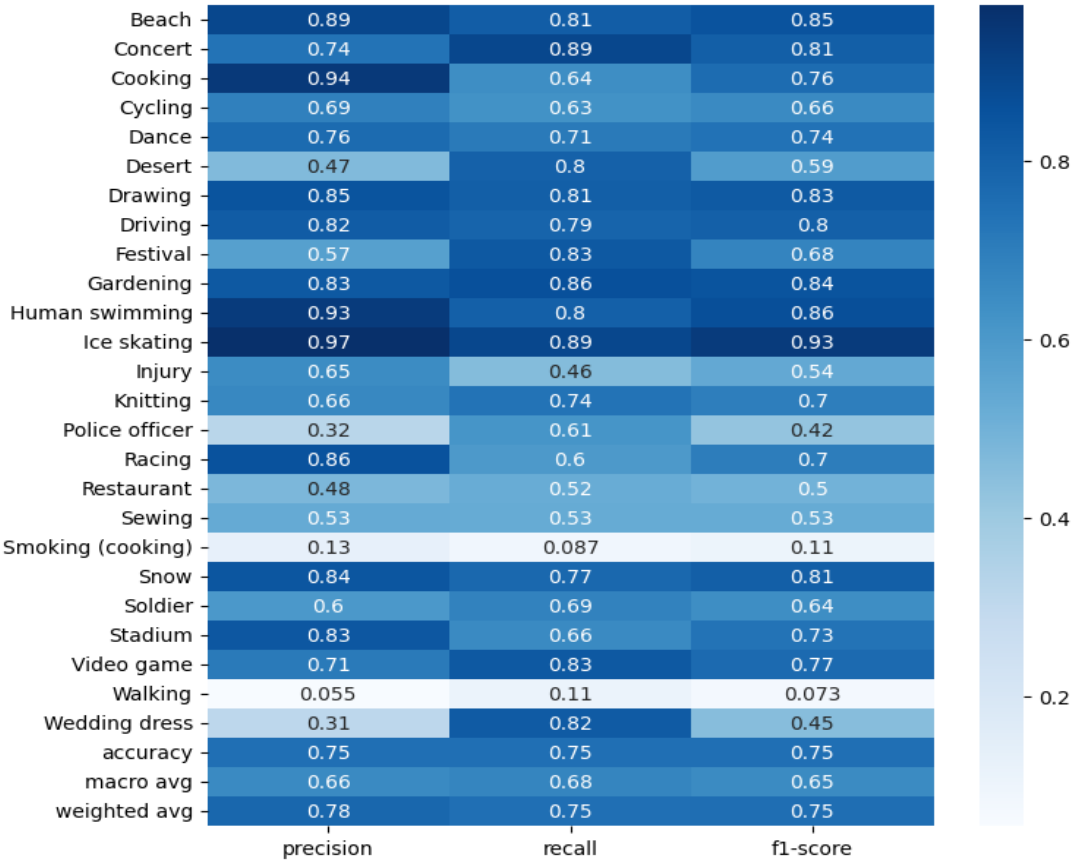

Figure 14. Results of VideoMAE on test data when fine-tuned on 70% of the data

Figure 15 shows the detailed results obtained when the model was trained with all of the selected dataset. The weighted average F1-score improves to 0.76 when the average number of instances is increased to 460. However, this improvement is very small, only 0.01, which suggests that increasing the data any further will not have a significant impact on the results. The distribution of results across the categories is similar to the previous experiments. However, unlike the first two subsets, the model is now able to predict all of the categories.

The summary of results obtained using VideoMAE is presented in the Table 5. The data reveals a gradual increase in the F1-score as the training data size increases. Notably, when the training data was augmented from 8,043 videos to 11,511 videos, there was a marginal improvement in the results, with the weighted F1-score rising from 0.75 to 0.76. This indicates that the model may have reached its learning

capacity, suggesting that further increases in data may not significantly enhance the results. Upon examining the category-wise results, it becomes apparent that the model encounters difficulties with classes that have limited data availability, as well as those representing specific objects such as a "wedding officer" or characters like a "soldier" or "police officer". However, the model performs relatively better when it comes to activities and locations. Overall, while considering the substantial training time of 131 hours, the results are not particularly impressive.

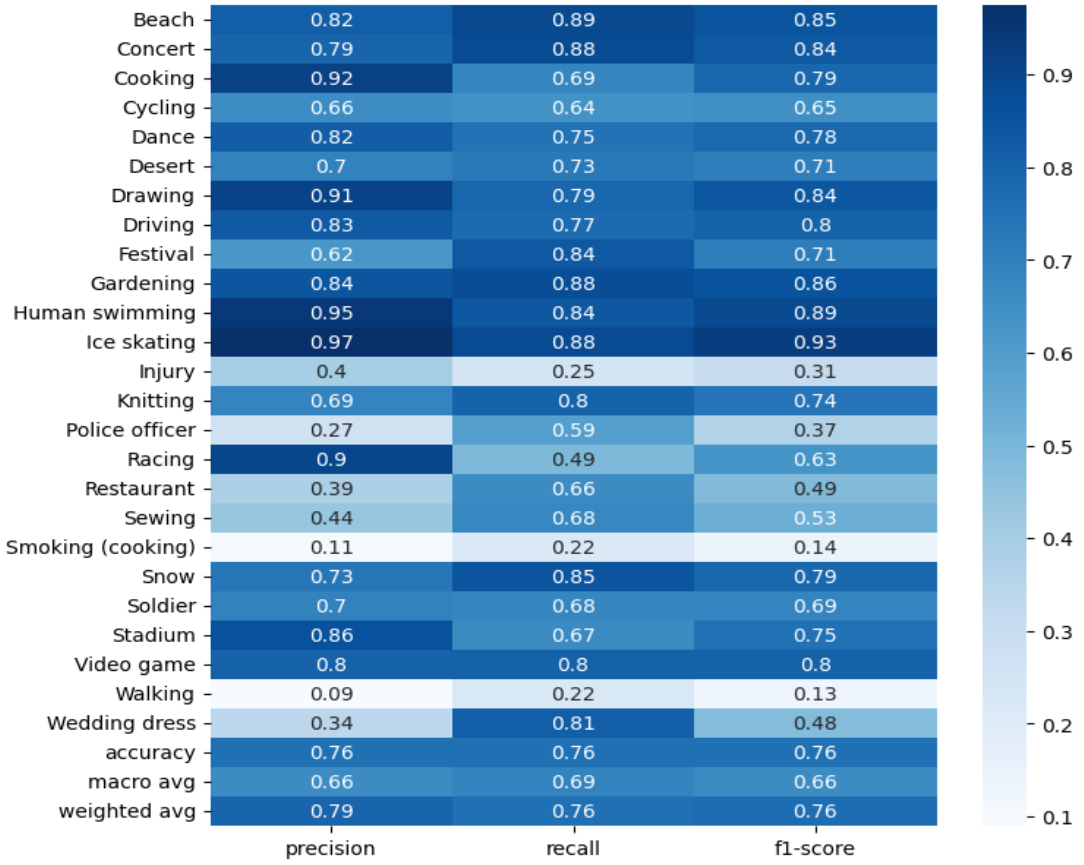

Figure 15. Results of VideoMAE on test data when fine-tuned on 100% of the data

Table 5. The summary of VideoMAE [1] results

Subset	Total videos (Training)	Weighted F1-score	Training time (hours)
10%	1142	0.64	13
30%	3446	0.69	34
70%	8043	0.75	92
100%	11511	0.76	131

4.2.2. X-CLIP

To implement this methodology, official GitHub repository of the selected model has been utilized. In this case, CLIP [59] served as the pre-trained model. As part of

the pre-processing steps, 32 frames were sampled from each video. These frames underwent various pre-processing transformations, including normalization, flipping, and random adjustments to color, saturation, and hue. These steps were taken to ensure optimal data representation and enhance the model's ability to learn from the video frames.

The performance of the model fine-tuned using only 10% of the dataset, is illustrated in Figure 16. The model achieved promising results, correctly classifying 1142 videos with a weighted F1-score of 0.92. Notably, the classes that posed a challenge for the VideoMAE model showed significant improvement in the fine-tuned model. It is worth noting that the "smoking (cooking)" class had a precision score of 1.00, indicating that all videos predicted as "smoking (cooking)" were indeed relevant to that class. However, the recall score of 0.13 suggests that the model still struggles to identify all the relevant videos for that class, with many being mislabeled as other classes. Overall, performance remains good but model still struggles with minority classes.

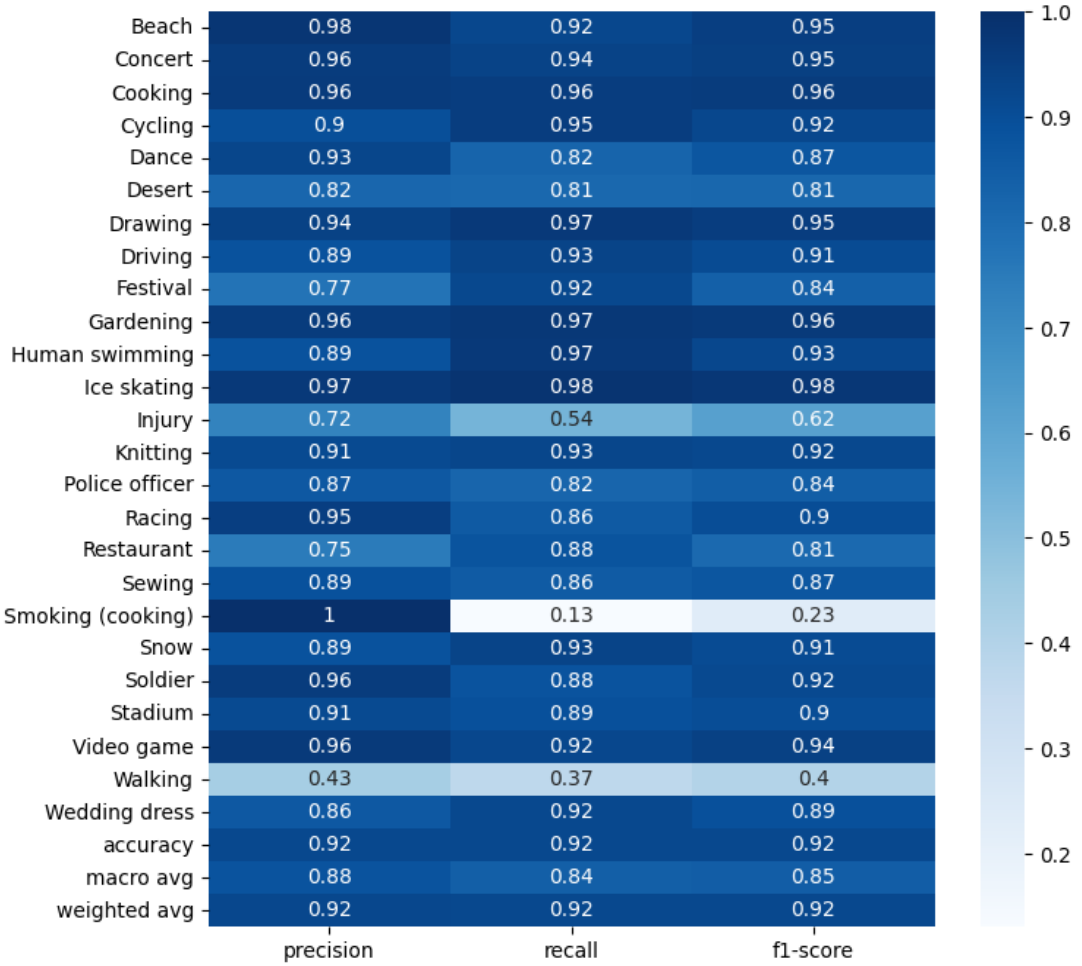

Figure 16. Results of X-CLIP [2] on test data when fine-tuned on 10% of the data

The performance of the model, after fine-tuning it on 30% of the available data, is presented in Figure 17. The weighted average F1-score shows a noticeable improvement, reaching 0.94. However, it's important to note that the macro average F1-score is slightly lower at 0.87. The lower macro average score can be attributed

to the limited number of instances available for certain classes. When there are fewer instances of a particular class in the dataset, the model's performance on that class tends to be lower. As a result, the average score is brought down. It is encouraging to observe that the performance of these classes has also improved as their instance count increased. Aside from the classes with limited instances, the results achieved across the remaining classes exhibit a consistent distribution and show good performance. This observation suggests that the models are able to effectively generalize and make accurate predictions for the majority of classes in the dataset.

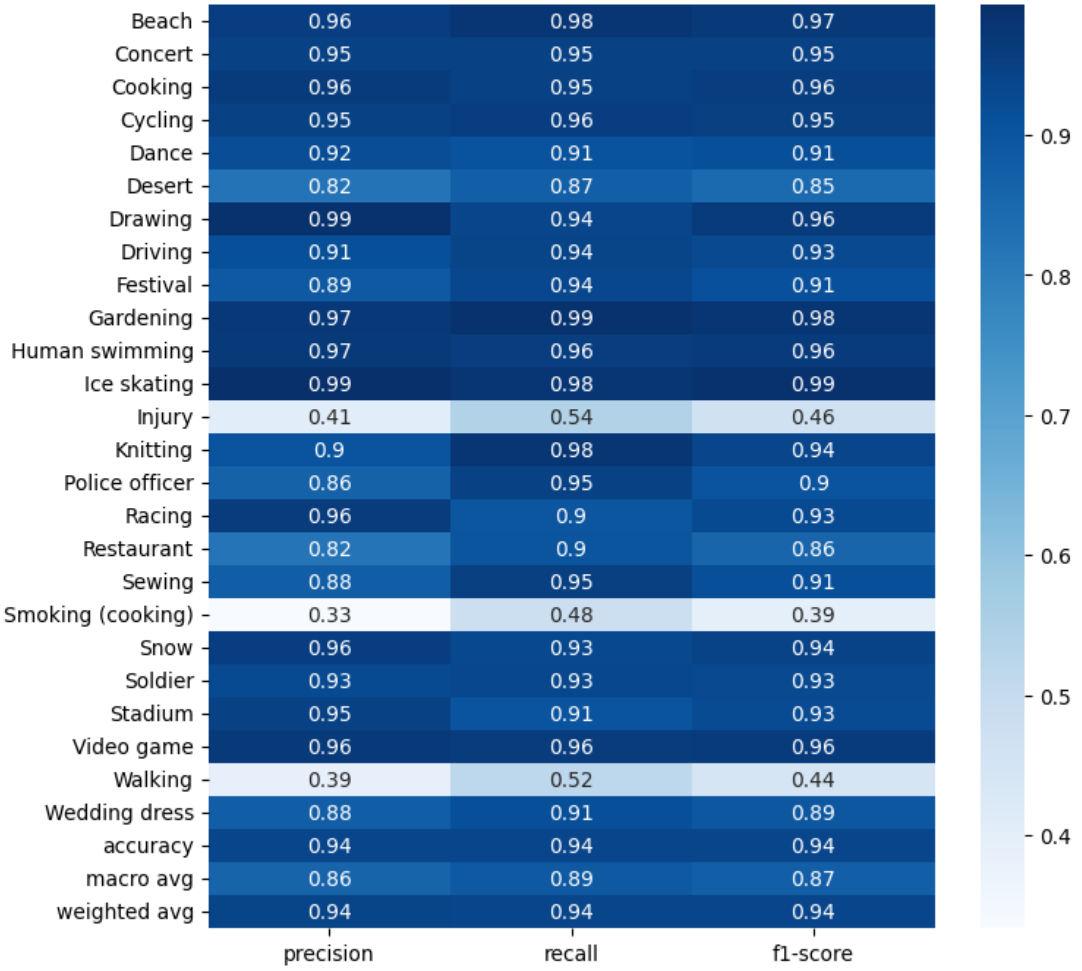

Figure 17. Results of X-CLIP [2] on test data when fine-tuned on 30% of the data

The results of fine-tuning the model on 70% of the available data are shown in Figure 18. In this experiment, the weighted average score remains at 0.94, while the macro average score increases to 0.89. This indicates a more balanced distribution of scores among the different classes. Compared to the previous experiments, there is a noticeable improvement in the performance of minority classes. This suggests that the model may have already reached its learning capacity with classes that have sufficient instances in the 30% subset, and including more classes does not have any positive effect on their performance. However, there is still room for improvement in the performance of minority classes, which is reflected after adding more instances.

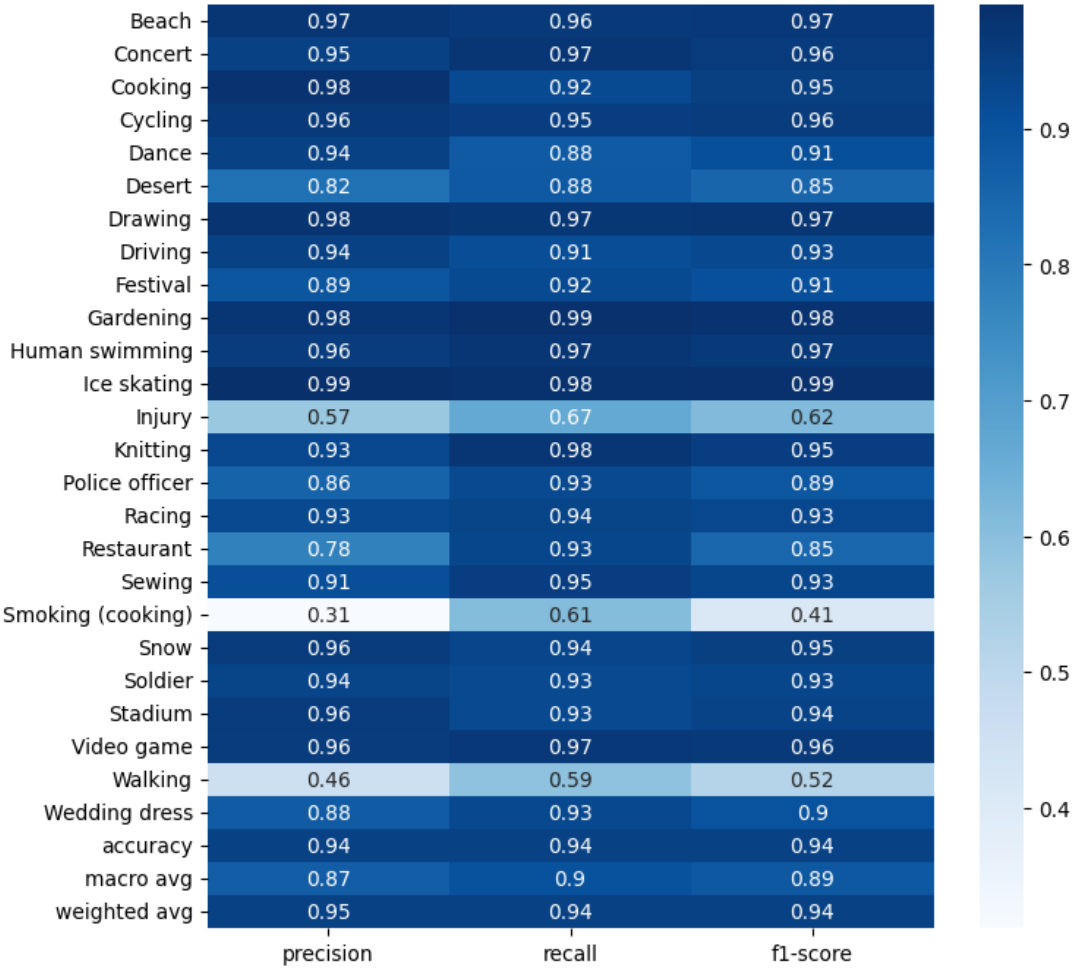

Figure 18. Results of X-CLIP [2] on test data when fine-tuned on 70% of the data

The results of the model fine-tuned on the entire dataset are depicted in Figure 19. It is apparent from the results that there is a slight improvement in performance, with the weighted F1-score increasing to 0.95 and the macro average reaching 0.90. While the performance of the minority classes has improved with the addition of more instances, their scores still hover around 0.6. Overall, the improvement in the results is very marginal considering almost 3500 new videos were added. The marginal improvement implies that the model has already captured a significant portion of the underlying patterns and features present in the dataset. Further additions to the dataset may not significantly enhance the model's performance

The X-CLIP model has demonstrated significant performance, achieving a weighted F1-score of approximately 0.92 using just 10% of the available data. The summary of the results can be seen in Table 6, which also reveals that fine-tuning the model with a 10% proportion of the data required only 4 hours. However, the X-CLIP model encounters challenges when learning patterns from classes with a relatively lower number of instances. In experiments where the input data has been increased, the improvement in results is not very significant, with only a marginal increase of 0.03 in weighted F1-score is observed after a ten-fold increase in training data size. By examining Table 6, it can be deduced that the best results are obtained with 30% of

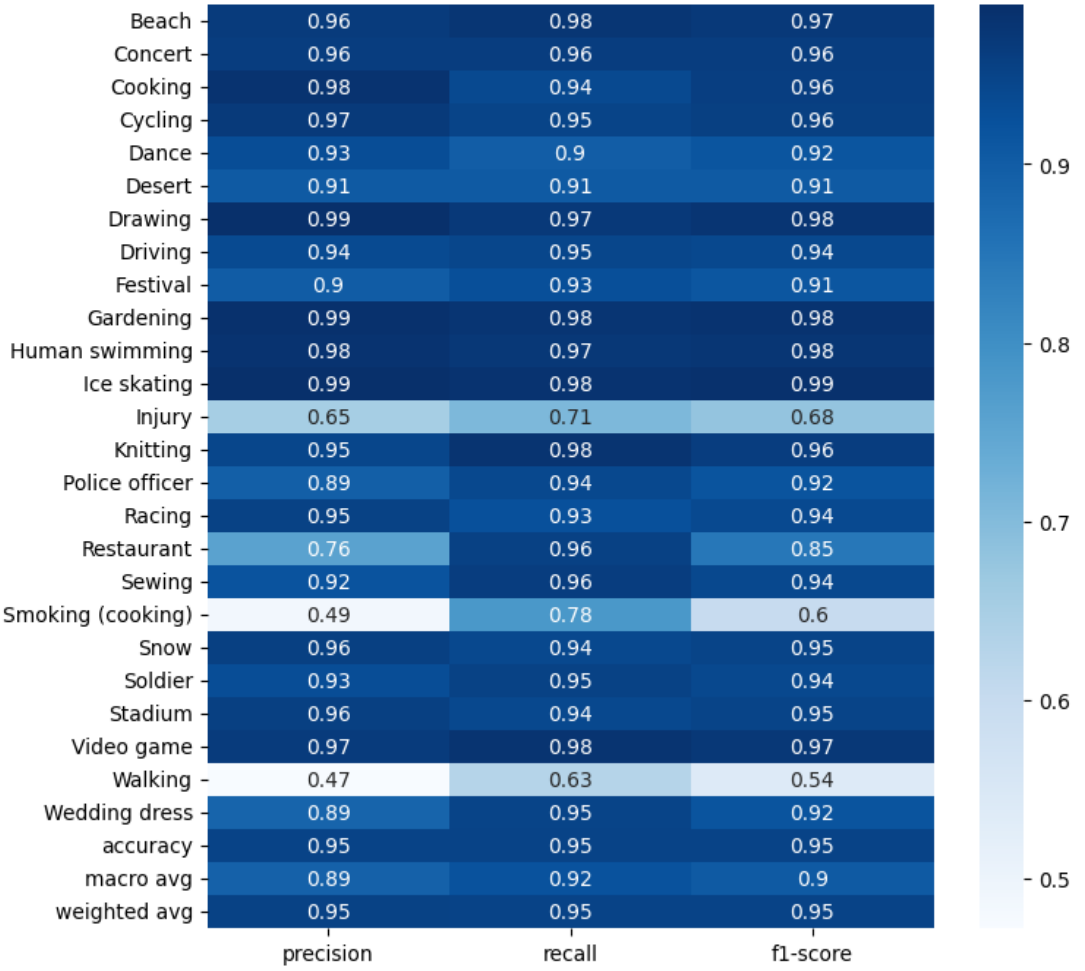

Figure 19. Results of X-CLIP [3] on test data when fine-tuned on 100% of the data

the data, considering the trade-offs between performance, data usage, and training time. Given these findings, the X-CLIP model presents a potentially favorable option in scenarios where the model needs to be retrained frequently.

Table 6. The summary of X-CLIP [2] results

Subset	Total videos (Training)	Weighted F1-score	Training time (hours)
10%	1142	0.92	4
30%	3446	0.94	8
70%	8043	0.94	17
100%	11511	0.95	22

4.2.3. Text4Vis

The implementation of Text4Vis [3] has been carried out using their official Github repository, which is actively maintained by the authors. In this study, 8 frames were randomly sampled from each video, and each frame underwent pre-processing steps

such as normalization, cropping, scaling, and flipping. The pre-trained VL model used for this implementation is CLIP [59].

The performance of the Text4Vis model was evaluated after fine-tuning on 10% of the available data. The results, depicted in Figure 20, reveal that the average weighted

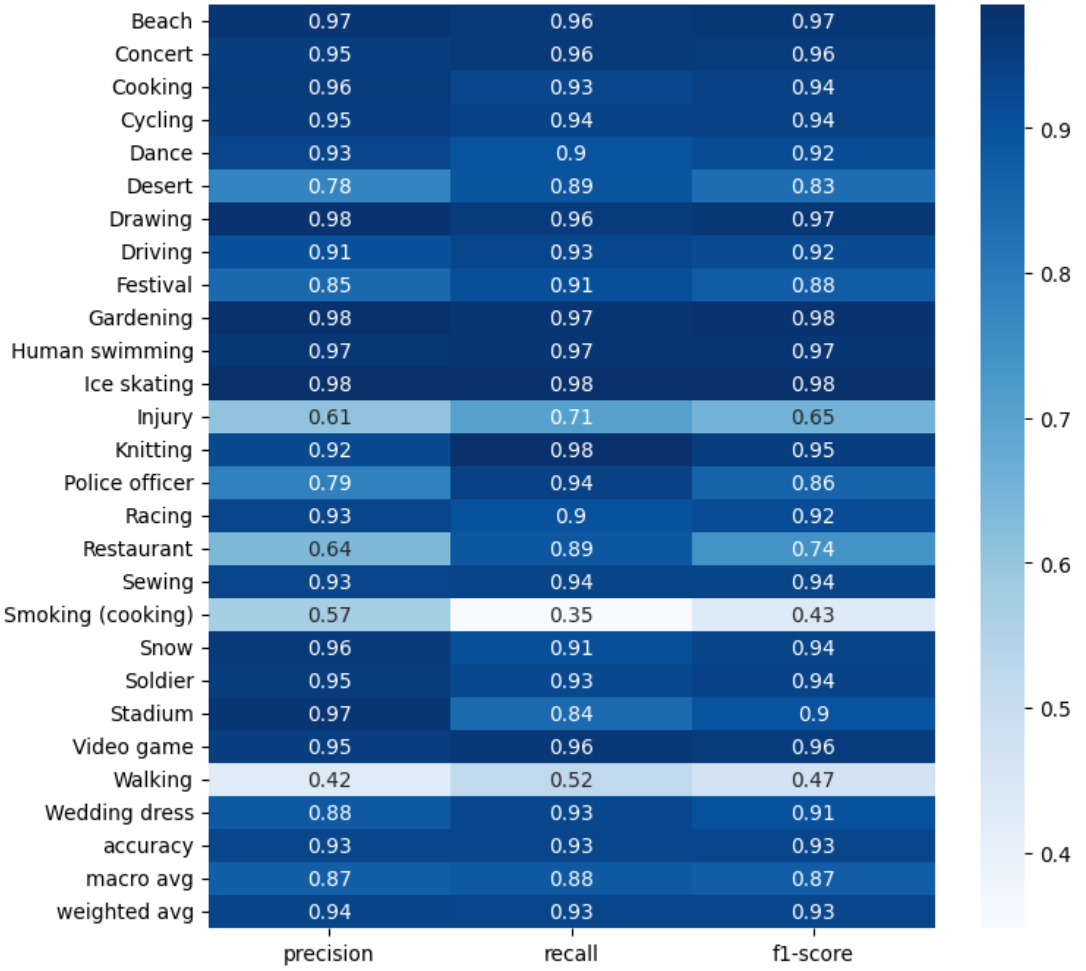

Figure 20. Results of Text4Vis [3] on test data when fine-tuned on 10% of the data

F1-score achieved by the model is 0.93, while the macro average score is 0.87. These scores indicate a high level of performance, surpassing that of the VideoMAE [1], and X-CLIP. These results are very similar to that of X-CLIP [2]. However, it is important to note that Text4Vis still faces challenges in accurately classifying the minority classes but results are a lot better than the previous two models. On the other hand, the classes "desert" and "restaurant" demonstrate relatively lower performance compared to other classes. It is important to highlight the impressive performance of the model in accurately classifying categories representing various activities. Notably, for the category "ice skating", the model achieved an exceptionally high F1-score of 0.98, showcasing its effectiveness in accurately recognizing and classifying this specific activity.

The results of fine-tuning the model on 30% of the available data are displayed in the Figure 21. Unsurprisingly, as the amount of data increased from 10% to 30%, the performance of the model also improves. Overall there was a little improvement in

the score as weighted and macro F1-scores increased to 0.89 and 0.94 respectively. However, certain categories like "desert", "police officer", and "restaurant" still exhibited relatively poor performance.

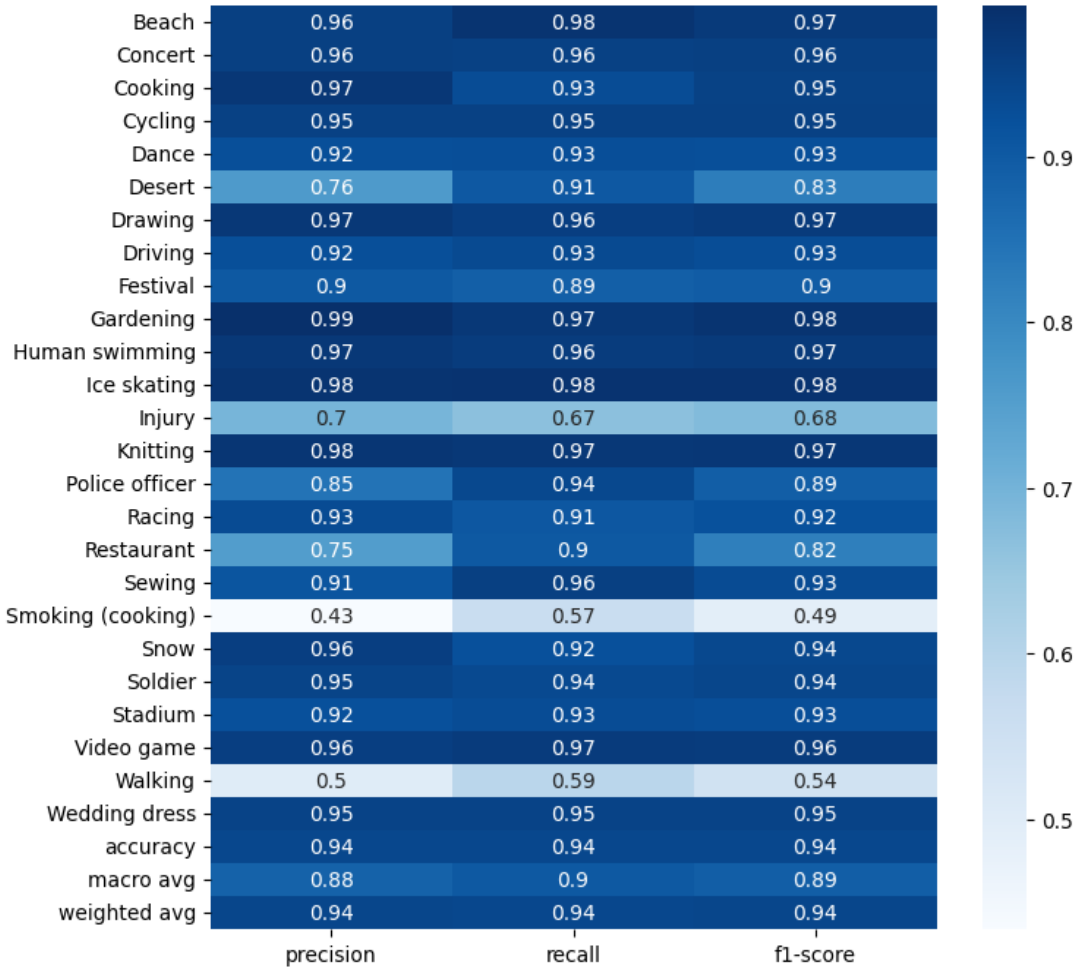

Figure 21. Results of Text4Vis [3] on test data when fine-tuned on 30% of the data

Figure 22 illustrates the results obtained after fine-tuning the model using 70% of the available data. Remarkably, there was a slight improvement in performance as the amount of data increased from 30% to 70%. Both the weighted average and macro scores increased to 0.89 and 0.95, respectively. While the overall improvement is encouraging, it is important to note that minority classes are still underperforming in comparison to the majority classes. The addition of more instances to the training data didn't have a significant improvement on their performance.

Figure 23 presents the results obtained after fine-tuning the model using 100% of the available data. Surprisingly, there was not a significant improvement in the overall performance compared to the model fine-tuned on 70% of the data. While the macro score did improve slightly to 0.90, the weighted score remained the same. Considering the additional time and resources required to train the model on the complete dataset, the observed improvement may not be substantial enough to justify the effort. Despite the increase in data, the minority classes continued to underperform, albeit with some improvement as more instances were added.

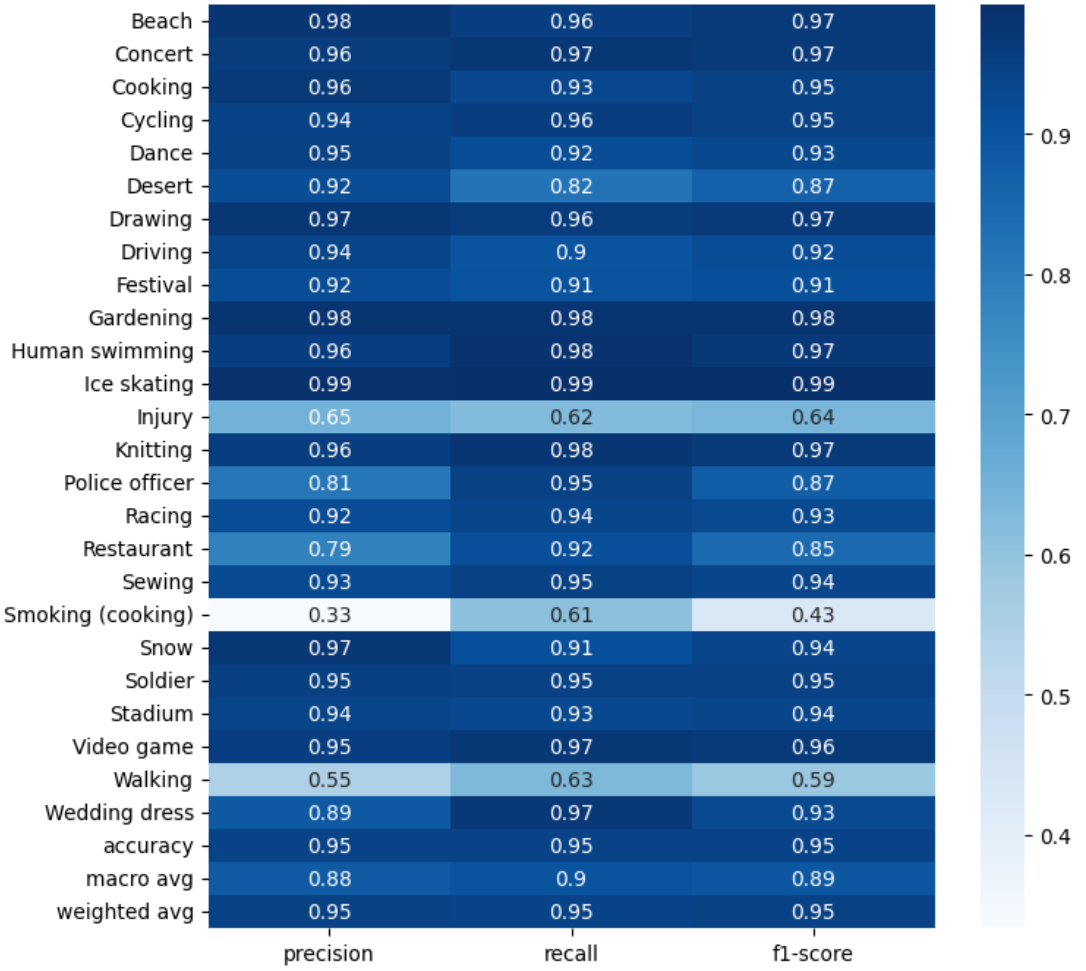

Figure 22. Results of Text4Vis [2] on test data when fine-tuned on 70% of the data

The results summary of fine-tuning the Text4Vis [3] model using various proportions of the selected dataset can be found in Table 7. The obtained results are found to be comparable to those achieved by the X-CLIP [2] model. With the addition of data, the performance improved. Considering the training time and increased computational requirements, it can be suggested that the best results were achieved with only 10% of the data. Adding more data beyond this point did not enhance the performance significantly, making it not worth the additional effort. The improvement in weighted F1-score was only 0.02 after increasing the data by 10 times.

Table 7. The summary of Text4Vis [3] results

Subset	**Total videos (Training)**	**Weighted F1-score**	**Training time (hours)**
10%	1142	0.93	7
30%	3446	0.94	23
70%	8043	0.95	49
100%	11511	0.95	73

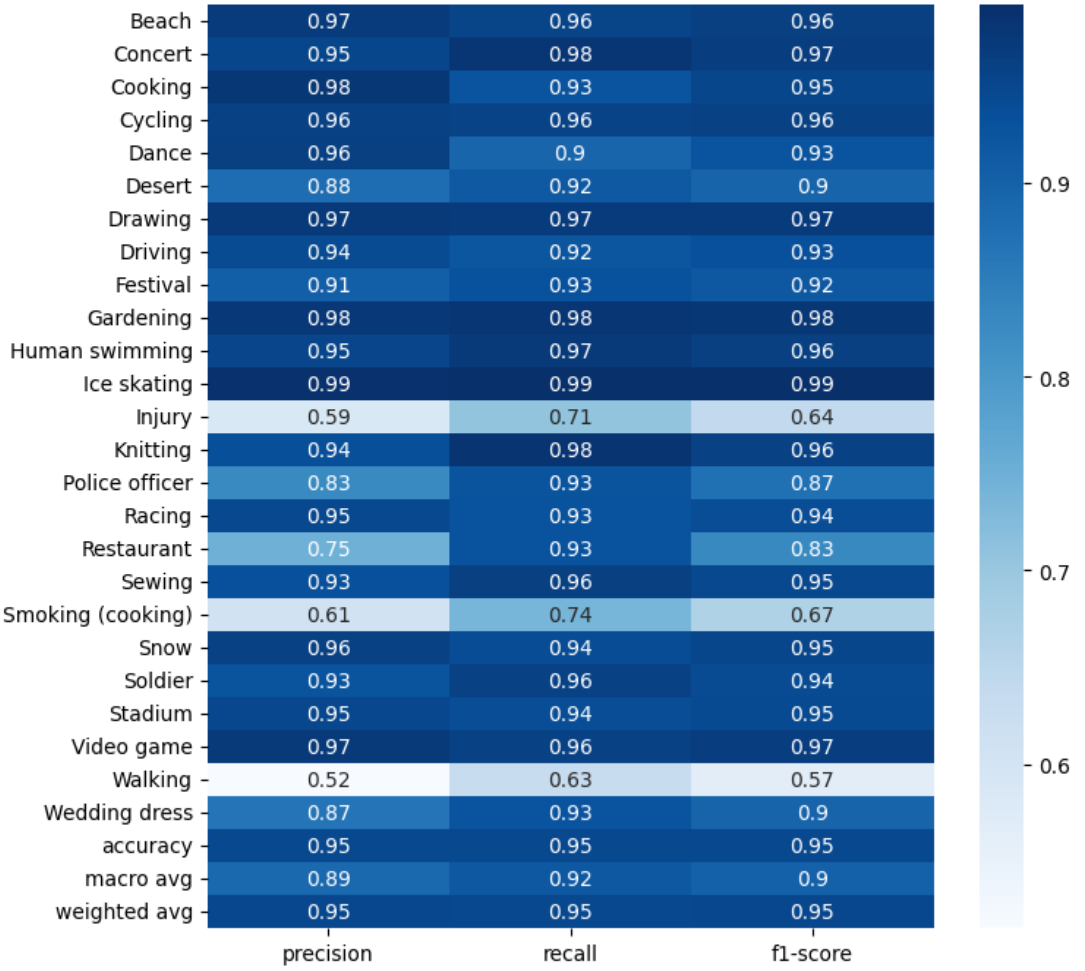

Figure 23. Results of Text4Vis [3] on test data when fine-tuned on 100% of the data

4.3. Comparison of Models Performance

This subsection provides a comparison of the results obtained from the evaluated methods in this study, along with the Teacher-Student [32] network trained on the YT8M [4] dataset. It is important to note that the results reported by Teacher-Student [32] are based on the entire testing dataset, which introduces a bias in the comparison. As the videos are chosen randomly for this study, that helps to reduce the bias to some extent. However, due to the unavailability of their trained model and the computational expense of reproducing their approach from scratch for all selected categories, it was not feasible to include their implementation in this project. Furthermore, it is important to consider that the original YT8M dataset is designed for multi-label classification, which adds another layer of complexity to the comparison. Directly comparing the results with the evaluated models might not provide a fair assessment. However, it can still provide a rough indication of the capabilities of the different approaches. The authors of the Teacher-Student network have reported results in terms of mean average precision (MAP) that is 0.41. In addition to the Teacher-Student network, [107] have also conducted evaluations using the YT8M dataset. However, their trained models were not publicly available, and they reported their results using a different evaluation

metric, Global Average Precision. Due to these differences, their approach and results are excluded from the comparison in this study.

Table 8 provides a comparison of results among evaluated models, highlighting the performance of the Text4Vis [3] method. It is noteworthy that Text4Vis outperforms the other models, achieving a MAP score of 0.87. Remarkably, this impressive result is obtained using only 10% of the data, which in total corresponds to a mere 1,142 videos. Furthermore, the training time required for Text4Vis is only 7 hours. Although the addition of more data does yield slight improvements in the performance of Text4Vis, these improvements are negligible. Additionally, Text4Vis demonstrates better handling of imbalanced data compared to the other models, further highlighting its effectiveness in dealing with such challenges.

Table 8. Comparison of best models for each approach

Model type	**Subset**	**MAP**	**weighted F1-score Training time (hours)**	
VideoMAE [1]	70%	0.66	0.75	92
X-CLIP [2]	30%	0.86	0.94	8
Text4Vis [3]	10%	0.87	0.93	7

Figure 24 illustrates the performance of the models in relation to the number of training videos. It is evident that both X-CLIP and Text4Vis exhibit a similar trend, with marginal improvement in results as the training data increases. Despite the increase, the performance gains are minimal for these models. In contrast, VideoMAE lags behind the other models in terms of performance, but it does show improvement with the addition of more training data. This observation suggests that the VideoMAE may have reached a saturation point in terms of their ability to extract meaningful information from the training data with 8000 videos.

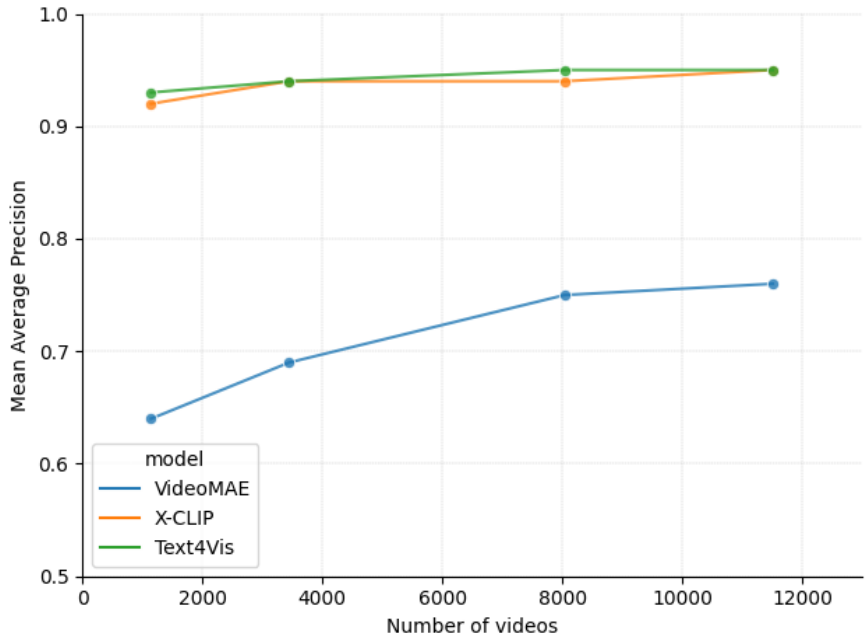

Figure 24. Performance of models with respect to number of training videos

After examining the results for each category, it becomes evident that all three models face challenges in achieving high scores for minority classes. This can be attributed to the overfitting of the models during fine-tuning, leading to a decrease in their ability to generalize well. However, it is worth noting that the models demonstrate a relatively better understanding of actions compared to other aspects such as objects, characters, or locations within the videos. Moreover, X-CLIP [2] and Text4Vis [3] excels in providing better results for classes that represent objects or characters where the VideoMAE model struggles. In conclusion, the findings suggest that pre-trained VL models can be effectively adopted for downstream video recognition tasks, eliminating the need for training the models from scratch. This not only saves computational resources but also leverages the learned representations from the pre-trained models, providing a strong foundation for video understanding tasks.

5. CONCLUSION AND FUTURE WORK

In this study, the focus was on evaluating three methods that utilize pre-trained VL models for video recognition tasks. The evaluation was conducted on the YT8M dataset, and the models were fine-tuned using varying proportions of the dataset. The obtained results are of significant interest and provided valuable insights. The earlier chapters of this thesis have extensively discussed the existing research work in the field, including the employed methods and their implementations. Moving forward, this chapter focuses on outlining the future directions for research in this field and the conclusion.

5.1. Future Work

This study serves as a foundational step towards conducting more extensive evaluations of pre-trained VL models to assess their usability for real-world applications. Due to time and computational limitations, the evaluation focused on 25 selected categories, providing a preliminary understanding of the models' generalizability. However, it is important to expand the evaluation by including more classes, as the original dataset consists of 3862 classes. This expanded evaluation would help to identify potential biases or limitations of the models, as certain classes may be more challenging to recognize or require specialized adaptations. This would provide a more comprehensive assessment of the models' capabilities.

Additionally, it is crucial to evaluate these models for multi-label classification. In many real-world scenarios, systems are required to predict multiple classes present in a video, which presents additional challenges and complexities. By evaluating the VL models in this context, would help to assess their ability to capture the relationships between different labels within a video. Moreover, these models can be evaluated even with a dataset size of less than 10% as both X-CLIP and Text4Vis are producing good scores on the 10% subset. Therefore, it would be intriguing to observe their performance on an even smaller amount of data.

Furthermore, to fully gauge the potential of VL models as foundational models, it is essential to test them in other tasks such as video-text retrieval or video captioning. These tasks provide a broader perspective on the models' capabilities and their suitability for various downstream applications. Solely focusing on video recognition might not provide a complete understanding of the models' potential. By examining their performance in these tasks, a deeper understanding of their language understanding capabilities and their potential for multimodal applications can be gained.

5.2. Conclusion

Nowadays, Computer Vision has become a hot research topic, with scientists striving to develop systems capable of better understanding visual data. The remarkable success of foundational models in NLP has attracted the interest of vision researchers, leading them to explore and construct vision-based foundational models. Consequently, a lot

of pre-trained models have recently been introduced, trained on extensive datasets, particularly those sourced from the web. These datasets come with short descriptions, which ensure a stronger integration between vision and language, surpassing the limitations of class names alone. In this study, three video recognition methods that rely on pre-trained VL models have been evaluated. The evaluation has been conducted using the YT8M [4] dataset, which closely resembles real-world settings and has remained relatively unexplored by the research community.

After evaluating the performance of the models on 25 categories of the YT8M dataset [4], several key findings have emerged. Firstly, Text4Vis [3] outperformed the other models, even when trained on only 10% of the available data. Notably, X-CLIP [2] also demonstrated competitive results, while VideoMAE [1] exhibited poor performance. All the evaluated models faced challenges in effectively handling minority classes, indicating room for further improvement in this aspect. The effect could be reduced by applying data balancing techniques such as undersampling and oversampling. One interesting aspect of Text4Vis is its unique approach, whereby the weights of the classifier head are initialized using label embeddings obtained from a pre-trained encoder. These weights are kept frozen during the fine-tuning process of the vision encoder. These models showcase a promising approach to address the complexities of video understanding as the weighted F1-score of 0.87 achieved by fine-tuning on only 1142 videos for 7 hours. It signifies the advancement in the capabilities of pre-trained VL models for tackling video recognition challenges.

6. REFERENCES

[1] Tong Z., Song Y., Wang J. & Wang L. (2022) VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. Neural Information Processing Systems .

[2] Ni B., Peng H., Chen M., Zhang S., Meng G., Fu J., Xiang S. & Ling H. (2022) Expanding language-image pretrained models for general video recognition. arXiv preprint arXiv:2208.02816 .

[3] Wu W., Sun Z., Ouyang W. & Inc. B. (2022) Revisiting classifier: Transferring vision-language models for video recognition. arXiv preprint arXiv:2207.01297 .

[4] Abu-El-Haija S., Kothari N., Lee J., Natsev P., Toderici G. & Varadarajan B. (2016) Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:1609:08675 .

[5] Hoerl A. & Kennard R. (1970) Technometrics. Ridge Regression: Biased Estimation for Nonorthogonal Problems .

[6] Cramer J. (2003) The origins and development of the logit model, Cambridge University Press. p. 149–157.

[7] Cortes C. & Vapnik V. (1995) Support-vector networks. Machine Learning 20 .

[8] Quinlan J. (1986) Induction of decision trees. Machine Learning 1 .

[9] J. M. (1967) Kmeans and analysis of multivariate observations. In 5th Berkeley Symposium on Mathematical Statistics and Probability .

[10] Fix E. & Hodges L. (1951) Discriminatory analysis, nonparametric discrimination: Consistency properties. USAF School of Aviation Medicine, Randolph Field .

[11] Breiman L. (2001) Random forests. Machine Learning 45 .

[12] Freund Y. & Schapire R. (1999) Adaptive game playing using multiplicative weights. Games and Economic Behavior .

[13] LeCun Y., Bottou L., Bengio Y. & Haffner P. (1998) Gradient based learning applied to document recognition. proceedings of IEEE .

[14] Goodfellow I., Pouget-Abadie j., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A. & Bengio Y. (2014) Generative adversarial nets. Neural Information Processing Systems .

[15] Zhao B., Li X. & Lu X. (2019) Hierarchical recurrent neural network for video. arXiv preprint arXiv:1904.12251 .

[16] S. H. & Schmidhuber J. (1997) Long short-term memory. Neural Computation .

[17] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A., Kaiser L., & Polosukhin I. (2017) Attention is all you need. in advances in neural. Information Processing Systems .

[18] Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., Gelly S., Uszkoreit J. & Houlsby N. (2021) An image is worth 16x6 words: Transformers for image recognition at scale. International Conference on Learning Representaions. .

[19] Devlin J., Chang M., Lee K. & Toutanova K. (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 .

[20] Brown T., B. M., Ryder N., Subbiah M., Kaplan J., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A., Agarwal S., Herbert-Voss A., Krueger G., Henighan T., Child R., Ramesh A., Ziegler D., Wu J., WInter C., Hesse C., Chen M., Sigler E., Litwin M., Gray S., Chess B., Clark J., Burner C., McCandish S., Radford A., Sutskever I. & Amodei D. (2020) Language models are few-shot learners. arXiv preprint arXiv:2005.14165 .

[21] Xu H., Liu B., Shu L. & Yu P. (2019) BERT post-training for review reading comprehension and aspect-based sentiment analysis. arXiv preprint arXiv:1904.02232 .

[22] Sahmoud T. & Mikki M. (2022) Spam detection using BERT. arXiv preprint arXiv:2206.02443 .

[23] Zhu J., Xia Y., Wu L., He D., Qin T., Zhou W., Li H. & Liu T. (2020) Incorporating BERT into neural machine translation. International Conference on Learning Representations .

[24] A. O., Kalchbrenner N. & Kavukcuoglu (2016) Pixel recuurent neural networks. arXiv preprint arXiv:1601.06759 .

[25] Doersch C., Gupta A. & Efros A. (2015) Unsupervised visual representation learning by context prediction. ICVV .

[26] Li X., Yin X., Li C., Zhang P., Hu X., Zhang L., Wang L., Hu H., Dong L., Wei F., Choi Y. & Gao J. (2020) Oscar: Object-semantics aligned pre-training for vision-language tasks. arXiv preprint arXiv:2004.06165 .

[27] Kim W., Son B. & Kim I. (2021) Vilt: Vision-and-language transformer without convolution or region supervision. International Conference on Machine Learning .

[28] Kim D., Cho D., Yoo D. & Kweon I. (2018) Learning image representation by completing damaged jigsaw puzzles. arXiv preprint arXiv:1802.01880 .

[29] Ji S., Xu W., Yang M. & Yu K. (2013) 3d convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, pp. 221–231.

[30] Fan H., Xiong B., Mangalam K., Li Y., Yan Z., Malik J. & Feichtenhofer C. (2021) Multiscale vision transformers. In Conference on Computer Vision and Pattern Recognition .

[31] Arnab A., Dehghani M., Heigold G., Sun C., Lučić M. & Schmid C. (2021) ViViT: A video vision transformer. In IEEE/CVF International Conference on Computer Vision .

[32] Bhardwaj S., Srinivasan M. & Khapra M.M. (2019) Efficient video classification using fewer frames. Computer Vision and Pattern Recognition Conference .

[33] Arandjelović R., Gronat P., Torii A., Pajdla T. & Sivic J. (2015) Netvlad: CNN architecture for weakly supervised place recognition. arXiv preprint arXiv:1511:07247. .

[34] Lin R., Xiao J., & Fan J. (2018) Nextvlad: An efficient neural network to aggregate frame-level features for large-scale video classification. arXiv preprint arXiv:1811.05014. .

[35] Simonyan K. & Zisserman A. (2015) Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations .

[36] He K., Zhang X., Ren S. & Sun J. (2015) Deep residual learning for image recognition. Conference on Computer Vision and Pattern Recognition .

[37] Krizhevsky A., Sutskever I. & Hinton G. (2012) Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems .

[38] Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V. & A. R. (2015) Going deeper with convolutions. IEEE Conference on Computer Vision and Pattern Recognition .

[39] Girshick R., Donahue J., Darrell T. & Malik J. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. Conference on Computer Vision and Pattern Recognition .

[40] Redmon J., Divvala S., Girshick R. & Farhadi A. (2016) You only look once: Unified, real-time object detection. IEEE Conference on Computer Vision and Pattern Recognition .

[41] Badrinarayanan V., Kendall A. & Cipolla R. (2015) Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence 39.

[42] Wu B., Xu C., Dai X., Wan A., Zhang P., Yan Z., Tomizuka M., Gonzalez J., Keutzer K. & Vajda P. (2020) Visual transformers: Token-based image representation and processing for computer vision. arXiv preprint arXiv:2006.03677 .

[43] Ramachandran P., Parmar N., Vaswani A., Bello I., Levskaya A. & Shlens J. (2019) Stand-alone self-attention in vision models. Neural Information Processing Systems .

[44] Bello I., Zoph B., Le Q., Vaswani A. & Shlens J. (2019) Attention augmented convolutional networks. In International Conference for Computer Vision .

[45] Cho K., Merrienboer B., Gulcehre C., Bougares F., Schwenk H. & Bengio Y. (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. In proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) .

[46] Mao J., Xu W., Yang Y., Wang J., Huang Z. & Yuille A. (2015) Deep captioning with multimodal recurrent neural networks (M-RNN). International Conference on Learning Representations .

[47] Khaki S., Wang L. & Archontoulis S. (2020) A CNN-RNN framework for crop yield prediction. Frontiers in Plant Science .

[48] Kanjanasurat I., Tenghongsakul K., Purahong B. & Lasakul A. (2023) CNN–RNN network integration for the diagnosis of covid-19 using chest x-ray and ct images. AI for Biomedical Sensing and Imaging .

[49] Touvron H., Cord M., Douze M., Massa F., ablayrollesm A. & Jégou H. (2020) Training data-efficient image transformers & distillation through attention. arXiv preprint arXiv:2012.12877 .

[50] Sun C., Shrivastava A., Singh S. & Gupta A. (2017) Revisiting unreasonable effectiveness of data in deep learning era. arXiv preprint arXiv:1707.02968 .

[51] Han K., Xiao A., Wu E., Guo J., Xu C. & Wang Y. (2021) Transformer in transformer. arXiv preprint arXiv:2103.00112 .

[52] Chen C., Fan Q. & Panda R. (2021) CrossViT: Cross-attention multi-scale vision transformer for image classification. Internation Conference on Computer Vision .

[53] Carion N., Massa F., Synnaeve G., Usunier N., Kirillov A. & Zagoruyko S. (2020) End-to-end object detection with transformers. arXiv preprint arXiv:2005.12872. .

[54] Ranftl R., Bochkovskiy A. & Koltun V. (2021) Vision transformers for dense prediction. arXiv preprint arXiv:2103.13413 .

[55] Liu Z., Lin Y., Cao Y., Hu H., Wei Y., Zhang Z., Lin S. & Guo B. (2021) Swin transformer: Hierarchical vision transformer using shifted windows. In International Conference on Computer Vision .

[56] Li G., Duan N., Fang Y., Gong M. & Jiang D. (2020) Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In International Conference on Artificial Intelligence, .

[57] Chen Y., Li L., Yu L., Kholy A., Ahmed F., Gan Z., Cheng Y. & Liu J. (2020) Uniter: Universal image-text representation learning. In European conference on computer vision .

[58] Li L., Yatskar M., Yin D., Hsieh C. & Kai-Wei Chang K. (2019) VisualBert:a simple and performant baseline for vision and language. arXiv preprint arXiv:1908.03557 .

[59] Radford A., Kim J., Hallacy C., Ramesh A., Goh G., Agarwal S., Sastry G., Askell A., Mishkin P., Clark J., Krueger G. & Sutskever I. (2021) Learning transferable visual models from natural language supervision. International Conference on Machine Learning .

[60] Gao P., Geng S., Zhang R., Ma T., Fang R., Zhang Y., Li H. & Qiao Y. (2021) CLIP-Adapter: Better vision-language models with feature adapters. arXiv preprint arXiv:2110.04544 .

[61] Zhang R., Fang R., Gao P., Zhang W., Li K., Dai J., Qiao Y. & Li H. (2021) Tipadapter: Training-free CLIP-adapter for better vision-language modeling. arXiv preprint arXiv:2111.03930 .

[62] Patashnik O., Wu Z., Shechtman E., Cohen-Or D. & Lischinski D. (2021) StyleCLIP: Text-driven manipulation of StyleGAN imagery. arXiv preprint arXiv:2103.17249 .

[63] Gal R., Patashnik O., Maron H., Chechik G. & Cohen-Or D. (2021) StyleGAN-NADA: CLIP-guided domain adaptation of image generators. arXiv preprint arXiv:2108.00946 .

[64] Karras T., Laine S. & Aila T. (2019) A style-based generator architecture for generative adversarial networks. In proceedings of Computer Vision and Pattern Recognition .

[65] Mokady R., Hertz A. & Bermano A. (2021) ClipCap: CLIP prefix for image captioning. arXiv preprint arXiv:2111.09734 .

[66] Luo H., Ji L., Zhong M., Chen Y., Lei W., Duan N. & Li T. (2021) CLIP4CLIP: An empirical study of CLIP for end to end video clip retrieval. arXiv preprint arXiv:2104.08860 .

[67] Vinker Y., Pajouheshgar E., Bo J., Bachmann R., Bermano A., Cohen-Or D., Zamir A. & Shamir A. (2022) CLIPasso: Semantically-aware object sketching. arXiv preprint arXiv:2202.05822 .

[68] Khalid M., Xie T., Belilovsky E. & Popa T. (2022) CLIP-Mesh: Generating textured meshes from text using pretrained image-text models. arXiv preprint arXiv:2203.13333 .

[69] Jetchev N. (2022) ClipMatrix: Text-controlled creation of 3d textured meshes. arXiv preprint arXiv:2109.12922 .

[70] Conde M. & Turgutlu K. (2021) CLIP-Art: Contrastive pre-training for fine-grained art classification. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops .

[71] Sanghi A., Chu H., Lambourne J., Wang Y., Cheng C., Fumero M. & Malekshan K. (2022) CLIP-Forge: Towards zero-shot text-to-shape generation. Computer Vision and Pattern Recognition .

[72] Shridhar M., Manuelli L. & Fox D. (2021) CLIPort: What and where pathways for robotic manipulation. arXiv preprint arXiv:2109.12098 .

[73] Wu H., Seetharaman P., Kumar K. & Bello J. (2021) Wav2CLIP: Learning robust audio representations from CLIP. IEEE International Conference on Acoustics, Speech and Signal Processing , pp. 4563–4567.

[74] Jia X., Yang Y., Xia Y., Chen Y., Parekh Z., Pham H., Le Q., Sung Y., Li Z. & Duerig T. (2021) Scaling up visual and vision-language representation learning with noisy text supervision. arXiv preprint arXiv:2102.05918 .

[75] Tan M. & Le Q. (2019) EfficientNet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning .

[76] Yuan L., Chen D., Chen Y., Codella N., Dai X., Gao J., Hu H., Huang X., Li B., Li C., Liu C., Liu M., Liu Z., Lu Y., Shi Y., Wang L., Wang J., Xiao B., Xiao Z., Yang J., Zeng M., Zhou L. & Zhang P. (2021) Florence: A new foundation model for computer vision. arXiv preprint arXiv:2111.11432 .

[77] Zhai X., Wang X., Mustafa B., Steiner A., Keysers D., Kolesnikov A. & Lucas Beyer L. (2022) LiT: Zero-shot transfer with locked-image text tuning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition .

[78] Wang Z., Yu J., Yu A., Dai Z., Tsvetkov Y. & Cao Y. (2022) SimVLM: Simple visual language model pre-training with weak supervision. In International Conference on Learning Representations .

[79] Yu J. & Wang Z. (2022) CoCa: Contrastive captioners are image-text foundation models. arXiv preprint arXiv:2205.01917 .

[80] Wang J., Yang Z., Hu X., Li L., Lin K., Gan Z., Liu Z., Liu C. & Wang L. (2022) GIT: A generative image-to-text transformer for vision and language. arXiv preprint arXiv:2205.14100 .

[81] Yan S., Zhu T., Wang Z., Cao Y., Zhang M., Ghosh S., Wu Y. & Yu J. (2022) VideoCoCa: Video-text modeling with zero-shot transfer from contrastive captioners. arXiv preprint arXiv:2212.04979 .

[82] Simonyan K. & Zisserman A. (2014) Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems .

[83] Feichtenhofer C., Fan H., Malik J. & He K. (2019) Slowfat networks for video recognition. In Proceedings of the IEEE international conference on computer vision .

[84] Wang X., Girshick R., Gupta A. & He K. (2018) Non-local neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition .

[85] Donahue J., Hendricks L., Rohrbach M., Venugopalan S., Guadarrama S., Saenko K. & Darrell T. (2016) Long-term recurrent convolutional networks for visual recognition and description. In Conference on Computer Vision and Pattern Recognition .

[86] Neimark D., Bar O., Zohar M. & Asselmann D. (2021) Video transformer network. arXiv preprint arXiv:2102.00719 .

[87] Liu Z., Ning J., Cao Y., Wei Y., Zhang Z., Lin S. & Hu H. (2022) Video swin transformer. In IEEE/CVF Conference on Computer Vision and Pattern Recognition .

[88] Plizzari C., Cannici M. & Matteucci M. (2020) Spatial temporal transformer network for skeleton-based action recognition. arXiv preprint arXiv:2012.06399 .

[89] Misra I., Zitnick C. & Hebert M. (2016) Shuffle and learn: Unsupervised learning using temporal order verification. In Proceedings of the European Conference on Computer Vision , p. 527–544.

[90] Lee H., Huang J., Singh M. & Yang M. (2017) Unsupervised representation learning by sorting sequences. In Proceedings of the IEEE International Conference on Computer Vision , p. 667–676.

[91] Xu D., Xiao J., Zhao Z., Shao J., Xie D. & Zhuang Y. (2019) Self-supervised spatiotemporal learning via video clip order prediction. In IEEE/CVF Conference on Computer Vision and Pattern Recognition .

[92] Wang R., Chen D., Wu Z., Chen Y., Dai X., Liu M., Jiang Y., Zhou L. & Lu Yuan L. (2022) BEVT: BERT pretraining of video transformers. In IEEE/CVF Conference on Computer Vision and Pattern Recognition .

[93] Sun C., Myers A., Vondrick C., Murphy K. & Schmid C. (2019) VideoBERT: A joint model for video and language representation learning. International Conference on Computer Vision .

[94] Wu W., Wang X., Luo H., Wang J., Yang Y. & Ouyang W. (2022) Bidirectional cross-modal knowledge exploration for video recognition with pre-trained vision-language models. In Computer Vision and Pattern Recognition .

[95] Carreira J., Noland E., Banki-Horvath A., Hillier C. & Andrew A. (2018) A short note about kinetics-600. arXiv preprint arXiv:1808.01340 .

[96] Kay W., Carreira J., Simonyan K., Zhang B., Hillier C., Vijayanarasimhan S., Viola F., Green T., Back T., Natsev P., Suleyman M. & Zisserman A. (2017) The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 .

[97] Szegedy C., Vanhoucke V., Ioffe S., Shlens J. & Wojna Z. (2016) Rethinking the inception architecture for computer vision. In proceedings of IEEE Conference on Computer Vision and Pattern Recognition , pp. 2818–2826.

[98] Schuldt C., Laptev I. & Caputo B. (2004) Recognizing human actions: A local svm approach. International Conference on Pattern Recognition .

[99] Blank M., Gorelick L., Shechtman E., Irani M. & Basri R. (2005) Actions as space-time shapes. International Conference on Computer Vision .

[100] Kuehne H., Jhuang H., Garrote E., Poggio T. & Serre T. (2011) Hmdb: A large video database for human motion recognition. International Conference on Computer Vision .

[101] Soomro K., Zamir A. & Shah M. (2012) UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 .

[102] Zhou.L, Xu C. & Corso J. (2017) Towards automatic learning of procedures from web instructional videos. arXiv preprint arXiv:1703.09788 .

[103] Karpathy1 A., Toderici G., Shetty S., Leung T., Sukthankar R. & Fei-Fei L. (2014) Large-scale video classification with convolutional neural networks. . In IEEE Conference on Computer Vision and Pattern Recognition .

[104] He K., Chen X., Xie S., Li Y., Dollár P. & Girshick R. (2022) Masked autoencoders are scalable vision learners. In IEEE/CVF Conference on Computer Vision and Pattern Recognition .

[105] Zhang Z. & Tao D. (2012) Slow feature analysis for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence .

[106] Sanh V., Debut L., Chaumond J. & Wolf T. (2019) DistilBERT, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 .

[107] Mao F., Wu X., Xue H. & Zhang R. (2018) Hierarchical video frame sequence representation with deep convolutional graph network. In proceedings of European Conference on Computer Vision .