Object-Centric Video Feature Extraction toward eSports Video Captioning

eスポーツビデオキャプショニングに向けた物体中心動画特徴抽出

Jan, 23, 2023

Tsunehiko TANAKA

田中 恒彦

Student ID Number: 5121F059

Advisor: Edgar Simo-Serra

Department: Computer Science and Communications Engineering

Research Guidance: Computer Graphics

A Thesis Submitted to the Department of Computer Science and Communications Engineering, the Graduate School of Fundamental Science and Engineering of Waseda University in Partial Fulfillment of the Requirements for the Degree of Master of Engineering

Abstract

eSports is competitive gaming at a professional level in which players or teams compete against each other. eSports is growing in popularity around the world and has many business opportunities. eSports tournaments live streaming plays a central role in the growth by attracting the audience with heated battles between professional players. Viewers who watch the streaming less frequently are potential customers of eSports, but they struggle to understand the complicated game rules of eSports. We aim to generate captions automatically that clearly explain eSports videos to support them. Preliminary experiments showed that more object-centric video feature extraction is needed in the eSports domain. We extend the video understanding model based on Transformer with an object detection model and achieve about six times the performance of existing methods in evaluating tubelet action detection. We introduce object queries to address the problems of existing approaches: the different sizes of detected objects and the high cost of high-frequency object detection. We also use the recurrent structure to store object information in object queries, allowing us to capture extended temporal contexts without a heuristic linking algorithm. Our proposed model may be widely applicable not only to eSports videos but also to sports video analysis and equipment anomaly detection in factories. Further research is required to apply our proposed model to data with few labels and much noise.

指導教員：シモセラ エドガー 准教授　　田中 恒彦

論文要旨

eスポーツはプロレベルでプレイヤーが競い合うゲームである．eスポーツは世界中で人気を集めており，ビジネスチャンスを多く含んでいる．特に大会配信はプロ同士の激しい戦いで観客を興奮させ，eスポーツの成長の一躍を担ってきた．大会配信を見る頻度が少ない視聴者こそeスポーツの潜在的な顧客であるといえる．大会配信を楽しむ上で彼らの障壁はeスポーツの複雑なゲームルールの理解である．この問題を解決するため，本研究はeスポーツ動画をわかりやすく説明するキャプションの生成に取り組む．事前実験によって，eスポーツドメインではより物体に注目した動画特徴抽出手法が必要だとわかった．我々は物体検出モデルを用いて動画理解モデルを拡張し，Tubelet Action Detectionの実験において既存手法の約6倍の性能を達成した．従来手法の課題である，検出物体のサイズの違いと高頻度な物体検出のコストの高さをobject queriesによって解決した．また，再帰構造を導入し，object queriesに物体の情報を保持することで，ヒューリスティックな接続アルゴリズムなしに長い時間的な文脈を捉えることが可能になった．本研究の成果はeスポーツ動画のみならず，スポーツ動画の解析や工場における機器の異常検知などにも応用することが可能である．ラベルが少なくノイズが多いデータに対しても提案モデルを適用可能にするため，更なる研究が必要である．

Acknowledgments

First of all, I would like to express my deepest gratitude to my supervisor, Prof. Edgar Simo-Serra. Without his continuous kind advices, I would not have completed my master research. I have learned a lot from him such as how to conduct research projects and how to write persuasive papers.

I also thank the staff and members of the Simo-Serra laboratory. Discussions and chats with the younger students have made my monotonous days enjoyable, and I have deepened my knowledge and learned new things from them.

Finally, my special thanks to my mother who has been an unstinting source of support.

Contents

Listing of figures

Listing of tables

1 Introduction

1.1 Motivation

eSports is competitive gaming at a professional level in which players or teams compete against each other for championships or prize money [60]. eSports is growing in popularity across all ages, genders, and regions. The revenue is increasing year by year and is expected to reach 1.38 billion by the end of 2022 (+16.4% from 2021) [60]. eSports tournaments play a central role in the growth. The audience of professional eSports tournaments is also increasing annually, reaching 532 million in 2022. The occasional viewers, who watch professional eSports less than once a month, account for 271 million people [60]. Reaching out to them will lead to further development of eSports.

The audience of eSports tournaments live streaming gets excited about watching the advanced and skilled gameplays of professional players and heated battles between professional teams. Skilled players use their knowledge and experience to focus on the gameplay of the professional players, while beginner players need help understanding the complicated rules of eSports games. For beginner players, clearly showing the gameplay in the live streaming is thus important for the occasional viewers to enjoy the tournaments. The tournament's live streaming should be a chance to attract occasional viewers to the eSports game.

We believe that captions, which clearly explain the gameplay, will keep the interest of the occasional viewers in the eSports game. Currently, commentators are always present at large tournaments. Their explanations of the situation and highlights allow even those unfamiliar with the game rules to enjoy the live streaming. However, only some people understand the eSports titles well enough to explain the advanced gameplay in the tournaments and have developed their talking skills. For even them, keeping up with increasing eSports titles, fast-paced trends, and game updates is challenging. We aim

to support them and occasional viewers with agents that understand and can explain game videos via captions.

1.2 Problem Settings and Challenges

Video Captioning Model understands input videos and generates captions to explain events in the video. Recently, video captioning has been greatly advanced by deep learning technology. The deep learning model in video captioning can be categorized into two approaches. The first approach is to reveal the correspondence between objects in the video and words in the caption [11, 62, 82, 92, 95, 98, 99]. The second is mapping the video feature space to the text feature space through self-supervised learning [25, 77, 102]. These approaches rely on video feature extraction models [10, 74, 80] to convert the videos into vectors. This model is pre-trained on simple videos, which include a few objects to explain with large motion. However, eSports videos are more complicated than these videos in containing multiple objects in a frame, objects are much smaller than the frame, and the object's motion is tiny. The extent to which existing approaches can be applied to the eSports domain is thus unclear. In addition, since the eSports domain is relatively young, few well-developed datasets exist. This study therefore aims to construct video feature extraction models for the eSports domain with limited data.

1.3 Our Approach

In this thesis, we advance three approaches towards video captioning for the eSports domain, as shown in Figure 1.1.

First, we experimentally apply existing video captioning methods to the eSports domain to evaluate current video feature extraction models. This experiment shows that all captioning models struggle significantly with eSports videos more than human action videos, and improvements are needed in the video feature extraction model.

Second, we consider how to train a video feature extraction model on unlabeled data. Specifically, we train an object detection model for characters in eSports videos in unsupervised settings. The results demonstrate that the model capture not only characters but also objects that should be included in the background (e.g., bushes and walls). We realized that the model needs to recognize the temporal dynamics of the game rules and the characters to distinguish between background and foreground.

Finally, we extend the video understanding model with an object detection model to recognize object-specific temporal dynamics. The object information is transported from the detector to the video understanding model via object queries in DETR [8]. This allows adaptive supporting long temporal dynamics while reducing the frequency of using the detector. As a first step toward unsupervised learning, we evaluate our model in the tubelet action detection task using a sports video dataset similar to eSports in a supervised setting.

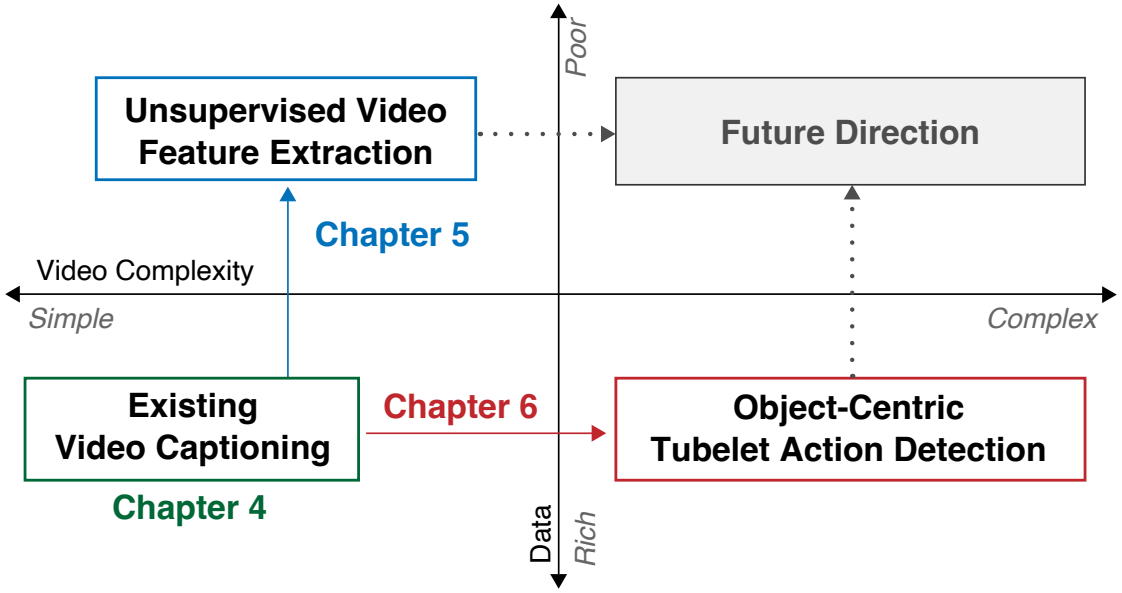


Figure 1.1: Overview of our contributions and future directions

1.4 Organization of the Thesis

This thesis begins with the background of our research in Chapter 2. It will then go on to related work in video captioning and video feature extraction in Chapter 3. Chapter 4 describes the experimentally applying existing video captioning methods to the eSports domain. Chapter 5 explains the experiment of video feature extraction in the unsupervised setting. Chapter 6 presents our approach using object queries for tubelet action detection. Finally, Chapter 7 summarizes our findings and limitations in this thesis. We also discuss future directions for this research.

2 Background

This chapter will briefly review the fundamental techniques associated with this study.

2.1 Machine Learning

Machine learning is one of the most common methods for finding rules or knowledge contained in data. Considering the example of object detection. Each object in an image can be represented by a vector comprising real numbers of colors of all pixels. The goal is to build a model that takes such a vector as input and predicts an object label and coordinates of a box surrounding the object, called bounding box, as output.

Studies of machine learning show the importance of training methods and model architectures. Training methods can be categorized into supervised learning, unsupervised learning, and reinforcement learning. Regarding model architecture, deep learning has been studied widely in recent years, and we will describe it in the next section. This section describes supervised learning and unsupervised learning used in our approach.

Supervised learning refers to problems in which training data consists of pairs of input data and corresponding ground truth data. It is divided into classification problems and regression problems. The goal of a classification problem is to assign an input vector to one of a finite number of discrete categories, while the goal of regression problems is to predict one or more continuous variables as outputs. In the example of object detection, predicting object labels corresponds to the classification problem, and predicting bounding boxes corresponds to the regression problem.

Unsupervised learning has been utilized to solve problems where the training data are only input vectors and no corresponding target labels. One of the most common methods of unsupervised learn-

ing is clustering, which is to discover groups of similar examples within the training data. Unsupervised learning can play a key role in reducing costly labeling in the construction of the training dataset and has received attention in feature representation extraction in recent years.

2.2 Neural Networks

Neural network is one of the basic types of machine learning, and deep learning, which we will discuss later, is based on it. Neural networks are used as multiple layers of logistic regression models with continuous nonlinearities, known as multilayer perceptron (MLP). For many applications, MLP can be significantly more compact and flexible than other machine learning models such as support vector machine. Instead, since the likelihood function is not a convex function of the model parameters, there is no guarantee that the training will converge to a globally optimal solution. However, in practical applications, it is often advantageous to feed new data to a compact and flexible model.

2.2.1 Feed-forward Network

As a neural network with a basic structure, we briefly discuss the mechanism of a feed-forward network (FFN). 'FFN' are often treated the same as 'MLP' in general, and this thesis follows this. The smallest element that makes up a neural network is a unit. A unit receives multiple inputs and multiplies each input by a different weight and adds a bias. The sum is then input to an activation function σ, and the resulting value $\mathbf{z}$ is the output of the unit. The calculation of the units is as in Eq. (2.2).

$$\mathbf{u} = \mathbf{W}\mathbf{x} + \mathbf{b}, \tag{2.1}$$

$$\mathbf{z} = \sigma(\mathbf{u}), \tag{2.2}$$

$\mathbf{W}$ denotes the weight, $\mathbf{b}$ denotes the bias, and σ denotes the activation function. There are various types of activation functions for different purposes. One of the most common activation functions is rectified linear unit (ReLU), as shown in Eq. (2.3). An advantage of ReLU is that it can reflect changes in the input to the output as long as the input is not negative.

$$\sigma(u) = max(u, 0), \tag{2.3}$$

The combination of several units is called a layer.

FFN is computed in order from the previous layer. An example of a simple two layers FFN is shown in Figure 2.1. The number of units and layers can be freely changed, enhancing the expressive power of FFN. Various networks with different layer shapes, numbers, and order of computation have been proposed, which are discussed in Section 2.3.

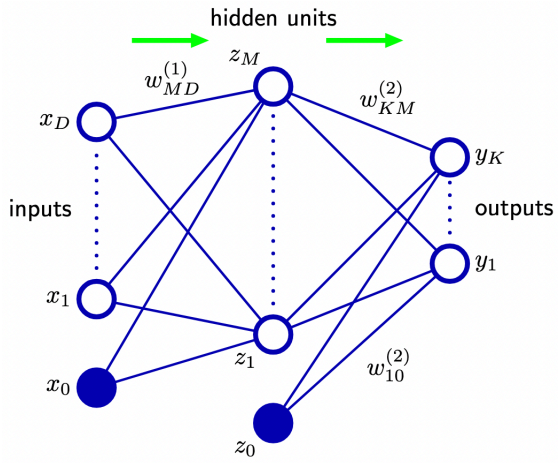


Figure 2.1: Network diagram for the two-layer neural network [7].

2.2.2 Network Training

The problem of determining the network parameters is solved by minimizing a function that represents the error between the predictions and the ground truths, which is called the objective (loss) function. In regression problems, the squared error shown in Eq. (2.4) is often used.

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} \|\mathbf{d}_n - y(x_n; \mathbf{w})\|\,, \tag{2.4}$$

where $y(x_n; \mathbf{w})$ denotes a network to be optimized, $\mathbf{w}$ denotes the network parameters, and $\mathbf{d}_n$ denotes a ground truth. Many objective functions are based on squared error, and the objective function needs to be customized depending on tasks. The differentiable objective function offers stochastic gradient descent (SGD) for minimizing the objective function. We will describe SGD in the next section.

2.2.3 Stochastic Gradient Descent

SGD is the most popular method for minimizing the objective function. The gradient is the first derivative of the objective function, as shown in Eq. (2.5).

$$\nabla E \equiv \frac{\partial E}{\partial \mathbf{w}}, \tag{2.5}$$

SGD searches for local optimal solutions by iteratively updating $\mathbf{w}$. A single update moves $\mathbf{w}$ a tiny distance in the negative gradient direction.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \epsilon \nabla E, \tag{2.6}$$

where $\mathbf{w}_t$ denotes the current weights, and $\mathbf{w}_{t+1}$ denotes the updated weights. ϵ is a constant that defines the magnitude of $\mathbf{w}$ updates, called the learning rate. Learning rate greatly affects training performance but is often determined empirically.

The cost of calculating the gradient increases as the number of layers increases. Backpropagation is often used as an efficient calculation method. Since backpropagation is a linear calculation, the gradient may diverge or disappear depending on the weight, called the vanishing gradient problem. There are some techniques for solving the problem, such as skip connection and residual connection.

2.3 Deep Learning Model

Deep learning is a machine learning technique that uses networks of many layers described in Section 2.2.1. The section below describes basic deep learning models and techniques.

2.3.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of FFN that can be applied to problems that use images as input, such as image classification. CNNs have convolution layers, which calculate as

$$u_{i,j} = \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{(i+p),(j+q)}\ h_{p,q}, \tag{2.7}$$

where $x_{i,j}$ denotes the pixel value of the pixel (i, j) in an input $W \times W$ image, and $h_{p,q}$ denotes the pixel value of the pixel (p, q) in an $H \times H$ image, which is called filter ($W > H$). The convolution operation is to extract color patterns from the input image that are similar to the color patterns of the filter. CNN has shown high performance in a wide range of fields, including image recognition.

2.3.2 Recurrent Neural Network

Recurrent neural network (RNN) is a type of neural network with cycles inside. By having cycles, RNNs can temporarily store information and change their behavior according to the stored data. It allows for capturing the context in a sequence of data, such as speech, language, or video. The simple RNN structure is shown in Figure 2.2, and the calculation in RNN is shown in Eq. (2.9).

$$\mathbf{Z}^t = \sigma_{\text{mid}}(\mathbf{W}_{\text{in}}\mathbf{X} + \mathbf{W}_{\text{r}}\mathbf{Z}^{t-1}), \tag{2.8}$$

$$\mathbf{Y}^t = \sigma_{\text{out}}(\mathbf{W}^{\text{out}}\mathbf{Z}^t), \tag{2.9}$$

The notation is described below:

- $\mathbf{X}$ denotes input.

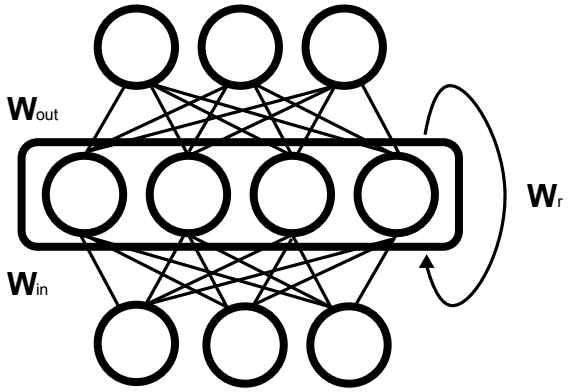


Figure 2.2: RNN structure.

- $\mathbf{Z}_t$ denotes the output of the middle layer at time t.
- $\mathbf{Y}$ represents the output of the model at time t.
- $\mathbf{W}_{\text{in}}$, $\mathbf{W}_{\text{r}}$, and $\mathbf{W}_{\text{out}}$ are the weights between layers.
- σ_{mid} and σ_{out} denote the activation functions.

By having a cycle W_{r}, the model can treat the output at time $t-1$ as the input at time t. Theoretically, RNN can deal with the entire input history in the past. In practice, however, it can only go back a limited history because repeating past outputs as inputs of the current layer is equivalent to increasing the number of layers and may cause the vanishing gradient problem described in Section 2.2.3. Long-Short Term Memory (LSTM) [32] was proposed to achieve long-term memory by using memory units.

2.3.3 Attention Mechanism

Attention mechanism weights the elements in a set by importance according to interest. For example, when recognizing the type of animal in a giraffe image, the attention mechanism gives high weight to just the giraffe part. We denote the content of interest as a vector $\mathbf{q}$ and each feature of the input sequence as $\mathbf{z}_i$ and r as a function that measures the relationship between $\mathbf{q}$ and $\mathbf{z}_i$.

$$r_i = r(\mathbf{z}_i, \mathbf{q}), \tag{2.10}$$

Various functions can be used for r; the most common is the inner product of $\mathbf{z}$ and $\mathbf{q}$.

$$r(\mathbf{z}, \mathbf{q}) = \frac{\mathbf{z}^\top \mathbf{q}}{\sqrt{D}} \tag{2.11}$$

To avoid polarizing the output to 0 and 1 in the next step, the inner product is divided by the square root of the dimensionality D of $\mathbf{z}$ and $\mathbf{q}$. We then normalize $r_1, \ldots, r_N$ of all data in the sequence

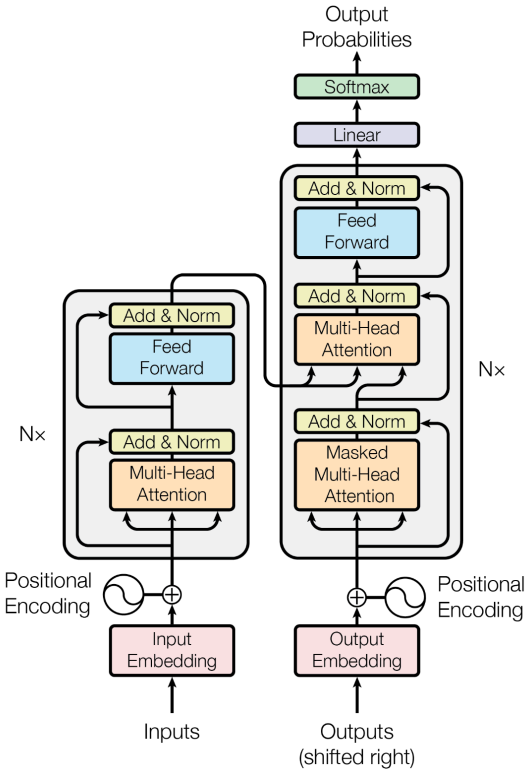


Figure 2.3: Transformer - model architecture [83].

with a softmax function.

$$a_i = \text{softmax}_i(r_1, ..., r_N) = \frac{exp(r_i)}{\sum_{l=1}^{N} exp(r_l)}, \tag{2.12}$$

Finally, we calculate the weighted average of the input sequence $\mathbf{z_i}$ with a_i as the weight.

$$\mathbf{z} = \sum_{i=1}^{N} a_i \mathbf{z}_i, \tag{2.13}$$

$\mathbf{z}$ can be regarded as the extracted part of the input sequence closely related to $\mathbf{q}$.

In the following section, we describe *transformer* [83], the most successful architecture that uses the attention mechanism.

2.3.4 Transformer

Transformer [83] is the most successful architecture that was proposed in the field of machine translation. The transformer has been developed as a model that takes sequence data as input and output,

such as sentences written in natural language. A significant advantage of the transformer is that it can process the entire information of sequence data at once. RNNs and LSTMs repeatedly use the output as input, which causes the vanishing gradient problem as described in Section 2.3.2, but the problem does not occur with the transformer.

The model architecture of the transformer is shown in Figure 2.3. The transformer has three components: multi-head attention, normalization, and feed-forward network. The core component is multi-head attention, which contains the attention mechanism described in the previous section.

Let $\mathbf{K}$, $\mathbf{V}$ be the inputs of multi-head attention and $\mathbf{Q}$ be the query. The attention is calculated as follows.

$$\mathcal{A}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D}}\right)\mathbf{V}, \tag{2.14}$$

In multi-head attention, H calculations of the attention are run in parallel. For each head, row vectors of $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ is linearly mapped to a D'-dimensinal space ($D' = D/H$). In other words, introduce $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V \in \mathbb{R}^{D\times D'}$ and transform as $\mathbf{Q} \to \mathbf{Q}\mathbf{W}_h^Q, \mathbf{K} \to \mathbf{K}\mathbf{W}_h^K, \mathbf{V} \to \mathbf{V}\mathbf{W}_h^V$, the head is calculated as follows.

$$\mathrm{head}_h = \mathcal{A}(\mathbf{Q}\mathbf{W}_h^Q, \mathbf{K}\mathbf{W}_h^K, \mathbf{V}\mathbf{W}_h^V), \tag{2.15}$$

where $\mathrm{head}_h \in \mathbb{R}^{M\times D}$. head_h are connected and linearly mapped by $\mathbf{W}^O \in \mathbb{R}^{D\times D}$.

$$\mathcal{A}^{\mathrm{M}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\mathrm{head}_1, ..., \mathrm{head}_H]\mathbf{W}^O, \tag{2.16}$$

$\mathcal{A}^{\mathrm{M}}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ is the output of the multi-head attention. $\mathbf{W}_h^Q$, $\mathbf{W}_h^K$, $\mathbf{W}_h^V$, and $\mathbf{W}^O$ are the learning parameters. If $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are the same, this component is called *self-attention* and is used to emphasize the important part in the sequence.

The order of elements in a sequence data is essential and must be reflected in the model's input. The transformer performs positional encoding, adding information indicating the sequence's position. Specifically, a vector $\mathbf{p}_i$ with the same length as $\mathbf{x}_i$ is created, representing the position of $\mathbf{x}_i$ in the sequence, and is added to or concatenated with $\mathbf{x}_i$. $\mathbf{p}_i$ is a vector of fixed values using a sine wave or a vector of learning parameters.

The disadvantage of the transformer is that the computational cost in Eq. (2.14) is enormous. In addition, since the transformer uses positional encoding, the input sequence is limited to a fixed length. When applying the transformer to video tasks, these shortcomings are significant constraints for designing model architecture.

The transformer has been applied to natural language processing, but now *vision transformer (ViT)* has been proposed, which can be applied to computer vision. As shown in Figure 2.4, the input of ViT

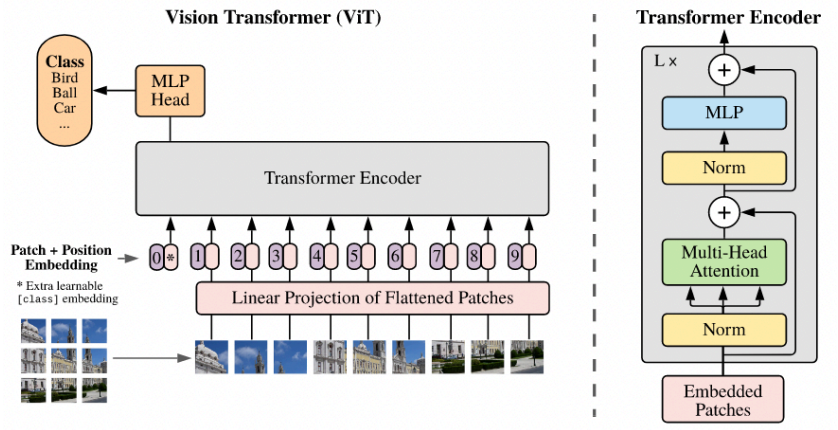


Figure 2.4: Overview of Vision Transformer [18].

is a sequence of patches into which the image is split. While CNN tends to extract local features of the image thanks to filters, ViT tends to extract global features of the image thanks to self-attention [69]. However, there is little difference in performance between CNN and ViT in various tasks. Since we believe the attention mechanism is effective for recognizing the motion in the video, we will use an extension of ViT to video in our approach.

3
Related Work

This chapter is divided into two sections. The first section will describe research on video captioning. The second section will review research on video feature extraction, which is one of the essential techniques in video captioning.

3.1 Video Captioning

Video captioning is a task that takes a video as input and generates captions describing events in the video as outputs. Video captioning models require recognizing video and grounding language abilities, it contributes significantly to the robot's visual and conversational skills. Recent trends in large-scale models [68, 70, 73] started in natural language processing have led to a proliferation of studies in vision and language. Although video captioning is related closely to the trend, the mechanism that can understand a video's temporal context and generate coherent sentences has yet to be established. In this section, we first review the study of Zhou et al. [101], mainly used as a baseline in video captioning, and describe the recent studies and problems with them.

Zhou et al. proposed a simple video captioning model [101] that combines a video understanding model, Temporal Segment Networks (TSN) [85], and a captioning model, Transformer [84]. The procedure consists of two steps: first, features are extracted from the input video using TSN, and then the features are transformed into sentences using the Transformer, as shown in Figure 3.1. Most research on video captioning has utilized the two-step procedure and has focused on how to ground the features of the video understanding model to the captioning model.

The basic approach to ground the video understanding model into the captioning model is explicitly mapping object regions to words in the texts. GVD [99] constructed ActivityNet-Entities, which

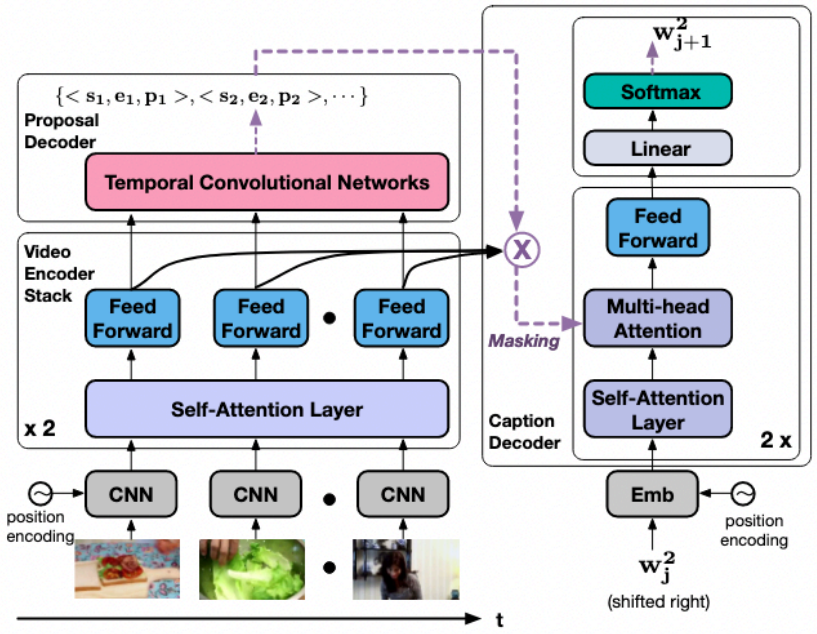


Figure 3.1: Overview of Masked Transformer [101].

includes entity-level bounding box annotations based on Activity Net Captions [42], and trained the captioning model with the annotations. Zhang and Peng proposed a model [94] for features representing temporal dynamics of salient objects extracted by an object detector such as Mask R-CNN [29]. This model can extract detailed dynamics for each object but has yet to extract dependencies between objects. Several studies [62, 95, 98] have proposed networks representing relationships between objects using GNNs and other methods to address this problem. HMN [92] splits captions into three hierarchical levels: entity, predicate, and sentence, and proposed dedicated loss functions for each level. At the entity level, words are mapped to object region features extracted by Faster R-CNN [71]. However, using pre-trained object detectors suffers from limited cover for different domain objects and training costs. Chen and Jiang, Vaidya et al. proposed models that use attention modules to extract spatial information without external object detectors [11, 82].

More recent attention has focused on self-supervised learning since BERT [16] appeared. BERT is a pretraining method for word embedding, where sentences are transformed into queries using word embedding and partially masked, and multi-Transformer blocks are trained to predict the token of the masked parts. Since a video can be treated as a sequence of frames, just like sentences, the BERT framework can be directly applied to video language tasks. VideoBERT [77] is the first study to apply BERT to video tasks and proposes to concatenate video tokens and word tokens for pretraining. Sun et al. discretized video features into video tokens via clustering, but detailed local information, e.g., interacting objects and human actions, could be lost during clustering [77]. ActBERT [102] adds

verbs from the texts and object regions extracted by the object detector to the tokens for this problem. Ging et al. focused on long-range temporal dependencies and proposed three hierarchy levels in video and language: frame/word, clip/sentence, and video/paragraph [25].

Much current literature on the captioning model has paid particular attention to the Transformer since Zhou et al. introduced it. MART [44] combines an encoder and a decoder and has a memory updater like LSTM to maintain a long-term context. PDVC [86] regards the Transformer as a converter of aggregate data and generates multiple captions in parallel. However, these models depend on offline-extracted video features by 2D/3D CNNs trained on video understanding tasks. Aafaq et al. argued that careful designing of visual features is important and proposed a method to apply Short Fourier Transform to CNN features of the video [1]. SwinBERT [52] trains the entire model, including the video feature extractor, by inputting the video directly into Video Swin Transformer [54] instead of CNNs.

Overall, these studies suggest that capturing relationships between objects in the video and self-supervised learning with large-scale datasets are useful for grounding the video understanding model to the captioning model. However, few studies have examined video feature extraction, such as object detection and understanding models. In the next section, we will review the existing studies on video feature extraction.

3.2 Video Feature Extraction

The section below reviews models and training methods for extracting latent space features from videos. The recent methods of video feature extraction can be classified in terms of the training procedure into contrastive learning and video understanding. We will first describe contrastive learning and then explain video understanding, especially tubelet action detection, which is closely related to this study.

3.2.1 Contrastive Learning

Contrastive Learning is self-supervised learning acquiring a feature representation space without ground truth labels. For a single query object $\mathbf{x}$, positive example $\mathbf{x}^+$ and K negative examples $\mathbf{x}_1^-, ..., \mathbf{x}_K^-$ are prepared. A network $\mathbf{z} = f(\mathbf{x}; \mathbf{w})$, which calculates feature $\mathbf{z}$ from $\mathbf{x}$, are trained to attract $\mathbf{x}$ and $\mathbf{x}^+$ and repel $\mathbf{x}_1^-, ..., \mathbf{x}_K^-$ from $\mathbf{x}$. The loss function for the network training is called InfoNCE [61] and is represented as

$$E(\mathbf{w}; \mathbf{x}, \{\mathbf{x}_i\}_{i=0,...,K}) = -\log \frac{\exp(\mathrm{sim}(\mathbf{z}, \mathbf{z}^+)/\tau)}{\sum_{i=0}^{K} \exp(\mathrm{sim}(\mathbf{z}, \mathbf{z}_i^-)/\tau)}, \tag{3.1}$$

where $\mathrm{sim}(\cdot, \cdot)$ denotes the similarity between feature vectors, the inner product is used for this.

Several studies of pre-training for image classification have demonstrated that contrastive learning is

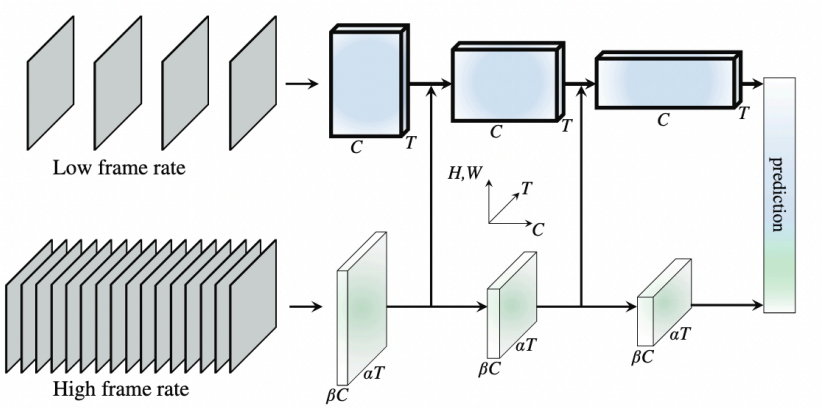


Figure 3.2: A SlowFast Network [23] with *Slow* pathway for low temporal resolution and *Fast* pathway for high temporal resolution.

effective for visual tasks, and then it has been applied to video tasks. SimCLR [13] uses a sample from a query image for a positive example and samples from the different images from the query image for negative samples. While SimCLR is a simple framework with only one encoder, it assumes ample memory space that can take a large K. MoCo [28] smoothes feature transitions using the second encoders with different weights, called momentum encoder, and reduces the required memory space by putting the negative samples in a queue. SwAV [9] avoids calculating contrastive loss for all negative samples by introducing clustering and comparing images at the cluster level. Furthermore, pre-training for object detection [5], which does not require bounding boxes and class labels, has been proposed as an extension of SwAV.

Most studies in contrastive learning for video tasks have focused on temporal dynamics and consistency. CVRL [67] extends SimCLR to video tasks, treating clips sampled temporally distant from the query sample as negative samples. Similarly, VideoMoCo [64] extends Moco to video tasks, sampling query clips to drop out several frames and adding temporal decay. Jenni and Jin used relative temporal transformations such as frame rate or playback direction as criteria for splitting positive and negative samples [37]. Some studies use additional information for temporal dynamics and consistency: graphs representing space-time correspondence [35], motion vectors from P-frames in mp4 [34], optical flow [40], foreground-background [17], and temporal gradient [89].

This section briefly summarizes the literature on contrastive learning in video feature extraction. Many published studies describe how to learn temporal dynamics and consistency. However, they suppose to use large memory, and how to reduce batch size for training in limited resources remains unclear.

3.2.2 Video Understanding

Video understanding, also called Action Recognition, is a task that takes a video as input and predicts a class of the video as output. After deep learning emerged, most researchers have used CNN and Transformer, successful methods in image classification. Simonyan and Zisserman was one of the first to apply CNN to video understanding and propose two-stream CNN trained on RGB images and optical flow [74]. Tran et al. expanded 2D-CNN to 3D-CNN for videos [80]. Carreira and Zisserman have provided a large-scale action classification dataset, Kinetics, and reports that pre-training on the large-scale dataset is effective for video understanding tasks [10]. After the success of two-stream networks, several studies have explored models that capture temporal dynamics with only RGB images because of the high computational costs of optical flow. Feichtenhofer et al. proposed a two-pathway SlowFast model, which has a path for capturing semantic information from sparse frames and a path for capturing rapidly changing motion by operating at high temporal resolution. They reduce the number of channels in the second path to support high temporal resolution without model oversizing. Feichtenhofer has shown that increasing the temporal and spatial resolution of the input is more effective than increasing the depth and parameters of CNN [22].

Recent studies [2, 6, 20, 47, 55, 57] have focused on applying Transformers to video understanding. Since Self-Attention is computationally expensive when applied directly to long sequences, improving computational efficiency is needed in video understanding. Motionformer [66] improves computational efficiency by approximating the inner product in Attention using a small size of prototype vector and aggregates information along implicitly determined motion paths. However, Kowal et al. found that most examined spatiotemporal models [6, 10, 20, 22] are biased toward spatial information except for certain two-stream architectures, such as SlowFast [23] in [41].

We can divide studies focusing on objects into Tubelet Action Detection, Object-Centric Action Recognition, and Group Action Recognition. Together with Multi-Object Tracking, we organize these groups according to the procedure for solving the tasks and create Figure 3.3. Tubelet Action Detection is to predict the set of bounding boxes (called tubelet) and the action label of each object from video features. Object-Centric Activity Recognition [31, 58] is to predict the class of the whole video using the information of objects obtained by external object detectors. Group Activity Recognition [24, 27, 46] targets team sports and predicts the individual action class of each player and the group action class of the team from the video features of players extracted by ground truth bounding boxes. Multi-Object Tracking [59, 87, 93] is to detect objects frame by frame and map objects between frames. Since eSports videos have multiple characters performing actions simultaneously and these actions are graphically decorated, extracting each character's actions in eSports videos is more complicated than in videos targeted in previous studies. Therefore, tubelet action detection is most relevant to our study in that it focuses on the motion of each object in a video.

Tubelet action detection can be divided into frame-level [12, 63] and tubelet-level detection, and

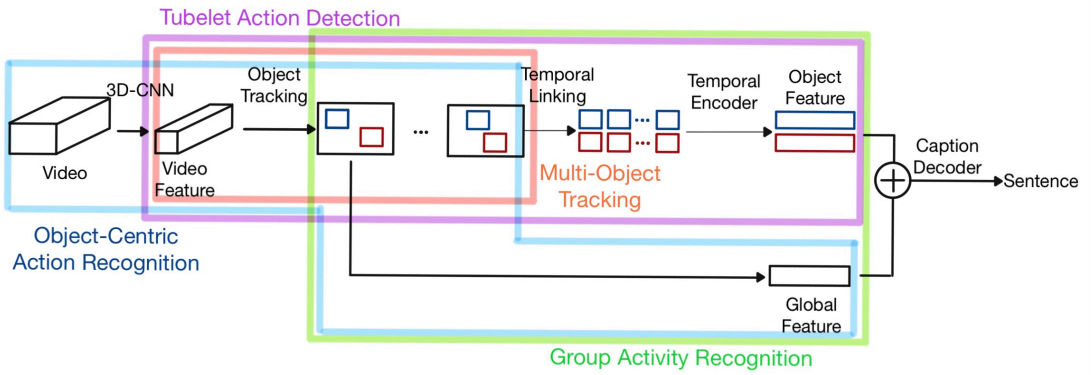


Figure 3.3: Four categories of object-centric video tasks.

tubelet-level detection is more related to our study than frame-level detection because our study focuses on each object's temporal dynamics. The following is a brief description of tubelet-level detection. Taking a tubelet as a representation unit was proposed by Jain et al. [36]. ACT-detector [39] uses an object detector frame-by-frame to extract ROI features and predict an action label from the stacked features of all frames. Similar to other video tasks, methods applying 3D-CNN [33] and LSTM [45] were proposed in tubelet action detection. STEP [91] proposed a method to sequentially improve tubelets based on the video features obtained from I3D [10]. Zhao and Snoek embeds RGB images and optical flow into a single two-in-one stream network with their proposed layers [96]. TACNet [75] defined the ambiguous frames as transitional states, not including any bounding boxes, and proposed a network to distinguish them. MOC [50] first predicts the coordinates of the center points in the bounding boxes in the center frame and then uses them to predict the trajectory in all frames and box sizes. CFAD [51] introduces the two-step procedures that first estimate coarse spatiotemporal action tubes from video streams and then refine the tube's location based on key timestamps. TubeR [97] is the first to apply the Transformer to this field and predicts simultaneously the positions and classes of tubelets and transitional states from video representations extracted by I3D [10]. HIT [21] leverages not only the RGB stream but also the hand and pose stream. In addition, since most studies support a limited length of videos, they process separately for each clip and apply the linking algorithm to the clips.

We leverage object queries in DETR [8] to improve the ability to capture long-term temporal context without the linking algorithm. We also use an object detector that allows us to capture even small objects that are difficult to capture with 3D-CNNs.

4

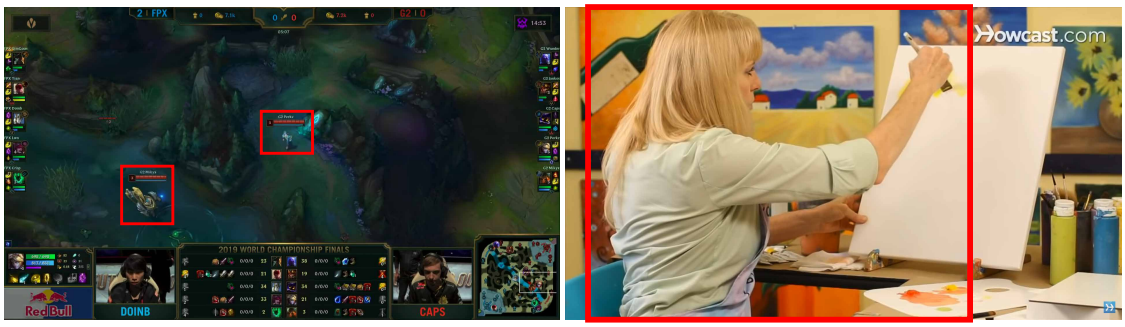
Video Captioning for eSports

We experimented with video captioning using state-of-the-art methods to see how well the existing video feature extraction performs for eSports videos. In this chapter, we will review our large-scale dataset for eSports video captioning and then discuss the state-of-the-art methods. The results demonstrate the need to improve video feature extraction performance for eSports videos and to collect well-organized caption data.

4.1 Dataset

We use LoL-V2T [79] and ActivityNet Captions [42]. LoL-V2T is a large-scale dataset for eSports video captioning, including 9,723 clips extracted from the competition footage of the popular eSports game *League of Legends* and 62,677 captions. The captions are sentences converted from the commentator's utterances into subtitles using automatic speech recognition (ASR) and segmented by sentence segmentation. ActivityNet Captions is an open-domain dataset in video captioning and is constructed by annotating captions by hand for videos from ActivityNet [19]. ActivityNet Captions is often used as a benchmark.

LoL-V2T differs from general video captioning datasets such as ActivityNet in two aspects. The first aspect is the object's size; the objects in LoL-V2T are much smaller than those in ActivityNet Captions. The second aspect is graphic effects; the videos in LoL-V2T include artificially created 3D characters with graphic effects, while the videos in ActivityNet captions include humans in natural images. We can see these differences in Figure 4.1.



(a) LoL-V2T

(b) ActivityNet Captions

Figure 4.1: Qualitative comparison of video captioning datasets.

4.2 Model

We use models with the two steps approach extracting video features and generating sentences from the features, as described in Section 3.1.

4.2.1 Video Feature Extraction Model

We use TSN [85] for video feature extraction, which many researchers [44, 99, 101] have utilized. TSN splits a video into small segments and extracts features from RGB images and optical flow for each segment using two CNNs. Xiong et al. proposed using TSN for video feature extraction in video captioning [90]. We could not apply their network to LoL-V2T because they published the extracted features of ActivityNet Captions, not the network. Therefore, we use TSN trained on Kinetics-400 [10] published in [14]. The model in [90] uses Inception V3 [78] for optical flow, while the model in [14] uses ResNet [30]. We use the features provided in [90] for ActivityNet Captions.

4.2.2 Captioning Model

We use Masked Transformer [101], MART [44], and PDVC [86] for the captioning model. Masked Transformer is a naïve application of Transformer to video captioning, as detailed in Section 3.1. This model also includes the temporal action proposal, which is a network that predicts the timestamp of actions in videos proposed in [100]. To simplify the problem in this study, we exclude this part and target only the pure captioning problem. MART is an extended Transformer by a memory module to keep the state of video segments and sentence history. With this module, MART has a structure similar to LSTM, which integrates the encoder and decoder of Transformer. PDVC applies Transformer as a converter of the query set proposed by DETR [8] to video captioning. CNNs extract the video features; the event queries are converted from the features by Transformer. PDVC predicts

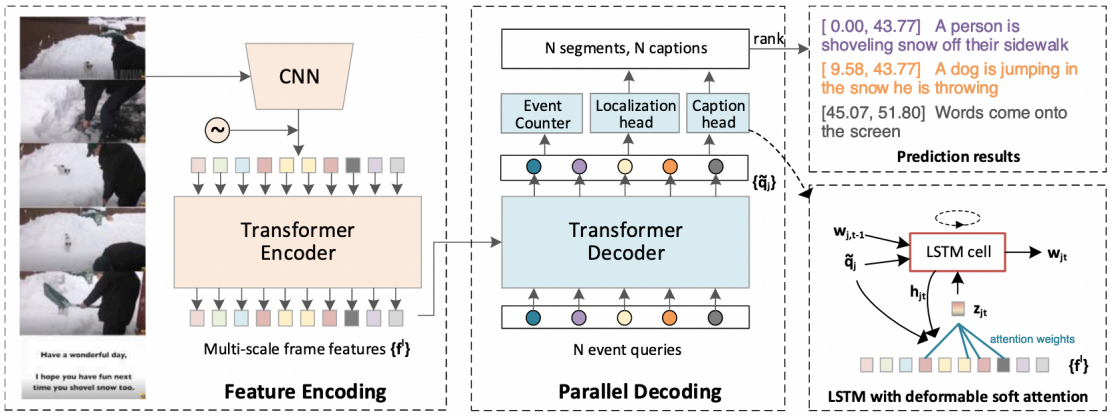


Figure 4.2: Decoding the outputs in parallel in PDVC [8].

Table 4.1: Performance of video captioning as measured by METROR [4]. For temporal action proposals, MART and PDVC do not predict, and Masked Transformer uses ground truth.

Methods	LoL-V2T	ActivityNet Captions
Masked Transformer [101]	8.58	11.20
MART [44]	11.50	15.68
PDVC [86]	1.64	15.80

the event number of the video, timestamps, and captions from the event queries in parallel, as shown in Figure 4.2. Predicting all outputs in parallel prevents the generated captions from relying on the performance of the timestamp prediction.

4.3 Evaluation

We measure the performance of video captioning with an automatic evaluation metric: METEOR [4], which indicates how similar the generated sentences and the ground truth are; the higher, the better. The quantitative results are shown in Table 4.1. What stands out in the table is that all methods perform lower in LoL-V2T than in ActivityNet Captions. PDVC especially resulted in the lowest value of LoL-V2T.

We also show the qualitative results in Figure 4.3. This figure shows that the generated sentences contain many similar patterns; the sentences generated by Masked Transformer and MART repeat the same expressions in a video (blue); Masked Transformer and PDVC use the same phrases in different videos (red). We also quote the results from PDVC in Figure 4.4 for comparison. By comparing Figure 4.3 and Figure 4.4, we can see that the generated sentences for LoL-V2T are more complicated than those for ActivityNet Captions.

Masked Transformer: i think that 's a lot of damage that you can see that the <unk> <unk> is going. i 'm gon na. i 'm gon na.
MART: the gold lead. I mean you can see that the gold lead is still in the mid lane for the side of the. a very nice. Lane and the.
PDVC: i think that is a very good job of the game. <unk> is gonna be able to get the kill.
Ground-Truth: lane continue to fight for experience so as knows as the <Champion> he is never solar carrying this lane so instead he uses his advantage to help out the other side of the map

Masked Transformer: i think that 's a lot of damage that you can see that. <unk> <unk> <unk>. they 're gon na get a lot of damage down here.
MART: I think that this is a really good play for vitality to be able to do it. you can see that the gold lead. I think that this is a really good play for vitality to be able to do it in the game.
PDVC: UNK is gonna be able to get the kill. i think that is a very good.
Ground-Truth: his old <Team> is actually going to die out of the time that was an oopsie other choice will die as well

Figure 4.3: Comparison of generated captions in LoL-V2T. Red indicates the same representations between videos, and blue indicates representations within a video.

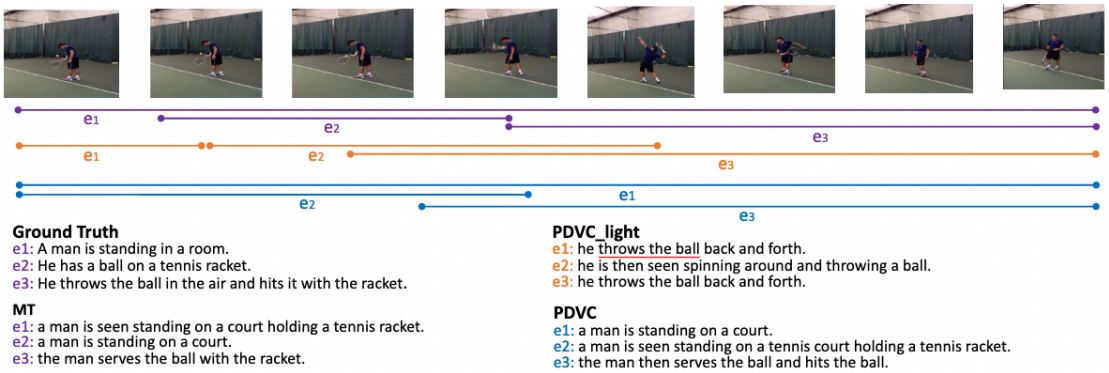


Figure 4.4: Comparison of generated captions in ActivityNet Captions from PDVC [86].

4.4 Discussion

The most obvious finding is that LoL-V2T is significantly more difficult than ActivityNet Captions. We can explain this by two factors. Firstly, the captions in LoL-V2T contain noise because ASR from the video subtitles automatically generated them. The noise harms the training of the captioning model. Secondly, objects are much smaller than the spatial resolution of the video frame, as shown in Figure 4.1. The objects may also be downsized by resizing or encoding to a size difficult to recognize.

Another interesting finding is that the generated sentences contain repeated expressions. A possible explanation for this could be that the ability to classify videos is limited when encoding video into features. Although TSN was pre-trained on Kinetics-400 in this experiment, it naturally needs to be trained on LoL-V2T. However, since LoL-V2T is not labeled for cations on videos, we could not train TSN in the same way as Kinetics-400.

Surprisingly, PDVC was found to perform poorly against LoL-V2T. Insufficient teacher signals may cause this. PDVC is designed to be trained with temporal action proposal, but it was omitted from the training for simplicity in this experiment.

Thus, we need well-organized caption data and improvement of the video feature extraction model. Since building such a caption dataset is costly, we seek to improve the video extraction model as a first step.

5

Unsupervised Frame Feature Extraction for eSports

In this chapter, we will describe our attempts to improve video feature extraction, the need for which was revealed in our preliminary experiments with the video captioning in Chapter 4. Since victory or defeat depends on the character's skills or positions in eSports, character behavior is the most important factor in classifying eSports videos. We thus apply an existing unsupervised object detection method to LoL-V2T to be able to extract characters from eSports videos. We also build a mechanism to obtain video frame by frame in the data processing. Qualitative demonstrates that the model can detect the characters even with unsupervised learning but cannot distinguish background and foreground objects.

5.1 Unsupervised Pretraining for Object Detection

When deep learning was first introduced to video understanding models, the mainstream approach was to aggregate the outputs of 2D-CNNs applied frame by frame. Following this trend, this study investigates how to train 2D-CNN models to detect characters from a frame as the first step in recognizing character behavior in eSports videos. Since the eSports video dataset LoL-V2T does not include bounding box labels, we focus on the unsupervised learning method DETReg [5].

DETReg is a method for pre-training the entire object detection, including object localization and embedding components. Many existing methods before DETReg were limited to learning embeddings by contrastive learning, as described in Section 3.2.1. UP-DETR [15] extends these methods to object detection but still needs to include additional pre-training for localization.

Specifically, supervised learning is performed on a single input image using three types of pseudo-ground truths: bounding box, class, and embedding. The pseudo ground truth bounding boxes are

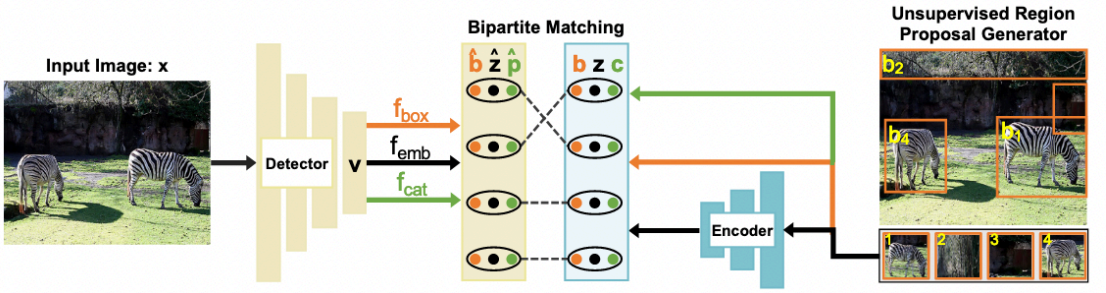


Figure 5.1: Overall framework of the pretraining method for object detection: DETReg [5].

obtained by Selective Search [81] as it requires no training data. SwAV [9] obtains the pseudo ground truth embeddings as one of the strongest performing methods for pretraining image classifiers. Since the ground truth class represents a foreground object or background, no additional labels are needed. A bipartite matching problem is defined between these pseudo-labels and the predictions, and the loss function is calculated between the matched pairs. Let us denote by y the pseudo ground truth set of objects, and $\hat{y} = \{\hat{y}_i\}_{i=1}^{N}$ the set of N predictions. A permutation of N elements $\delta \in \mathfrak{S}_N$ with the lowest cost in is searched.

$$\hat{\delta} = \underset{\delta \in \mathfrak{S}_N}{\operatorname{argmin}} \sum_{i}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\delta(i)}), \tag{5.1}$$

where $\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\delta(i)})$ is a pair-wise matching cost between pseudo ground truth y_i and a prediction with index $\delta(i)$ and is proposed in DETR [8]. Each element i of the pseudo ground truth set can be denoted as a $y_i = (c_i, \mathbf{b}_i)$ where c_i is the target class, and $\mathbf{b}_i \in [0, 1]^4$ is a vector that defines the pseudo ground truth box's center coordinates, height, and width relative to the input image size. $\mathcal{L}_{\text{match}}$ is defined as

$$\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\delta(i)}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\delta}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(\mathbf{b}_i, \hat{\mathbf{b}}_{\hat{\delta}(i)}) \right], \tag{5.2}$$

where $\hat{\delta}$ denotes the optimal assignment, and $\hat{p}_{\hat{\delta}(i)}(c_i)$ denotes the probability of class c_i. $\mathcal{L}_{\text{box}}(\mathbf{b}_i, \hat{\mathbf{b}}_{\hat{\delta}(i)})$ is defined as

$$\mathcal{L}_{\text{box}}(\mathbf{b}_i, \hat{\mathbf{b}}_{\hat{\delta}(i)}) = \lambda_{\text{iou}} \mathcal{L}_{\text{iou}}(\mathbf{b}_i, \hat{\mathbf{b}}_{\delta(i)}) + \lambda_{\text{L1}} \|\mathbf{b}_i - \hat{\mathbf{b}}_{\hat{\delta}(i)}\|, \tag{5.3}$$

where $\lambda_{\text{iou}}, \lambda_{\text{L1}} \in \mathbb{R}$ are hyperparameters. For $\mathcal{L}_{\text{iou}}$, the generalized IoU loss [72] is used. This optimal assignment is computed efficiently with the Hungarian algorithm [43]. The loss function contains the embeddings (the pseudo ground truth is denoted as $\mathbf{z}_i$, and a prediction is denoted as $\hat{\mathbf{z}}_{\hat{\delta}(i)}$) and

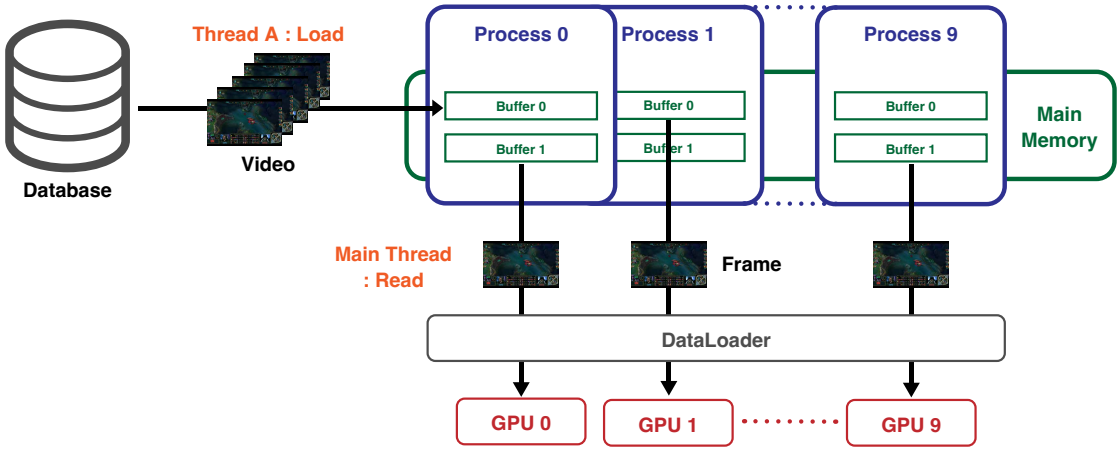


Figure 5.2: Overall of our data processing using buffers on main memory.

can be represented using the optimal δ as follows

$$\mathcal{L}(y, \hat{y}) = \sum_{i=1}^{N} \left[\lambda_f \mathcal{L}_{\text{class}}(c_i, \hat{p}_{\hat{\delta}(i)}) + \mathbb{1}_{\{c_i \neq \varnothing\}} (\lambda_b \mathcal{L}_{\text{box}}(\mathbf{b}_i, \hat{\mathbf{b}}_{\hat{\delta}(i)})) + \lambda_e \mathcal{L}_{\text{emb}}(\mathbf{z}_i, \hat{\mathbf{z}}_{\hat{\delta}(i)}) \right], \quad (5.4)$$

where $\mathcal{L}_{\text{class}}$ is the cross entropy loss, and $\mathcal{L}_{\text{emb}}$ is the L_1 loss.

5.2 Data Processing

We build data processing to allow videos to be saved and read as video files. Many researchers save videos as image files and read them as images. For example, UCF101[76] and JHMDB [38] save every frame as an image. The disadvantage of this method is that it requires enormous storage because it does not apply video compression techniques. This is not a problem for UCF and JHMDB because of their low spatial resolution, but it is expected to become more serious in the future when high-resolution videos become the focus of research (e.g., LoL-V2T [79] and MultiSports [48]). However, when processing frame by frame in random order, loading video frame by frame takes a long time to decode and uses ample space of the main memory. To solve this problem, we develop a data processing method that builds buffers on main memory and pools videos there.

The overall of our data processing is shown in Figure 5.2. First, several videos are loaded into a buffer in memory in a separate thread from the main thread. The main thread extracts a frame from the buffer that has already been loaded and loads a frame for training (e.g., PyTorch's DataLoader [65]). While reading a frame from one buffer, the other buffer is loaded. This procedure is executed in parallel by processes for the number of GPUs. Note that the video's allocation to the GPU and buffer size must be determined before training.

Figure 5.3: The output of Selective Search for a frame in LoL-V2T. The input frame is cropped inside the pink line box to prevent focusing on information UI outside the box.

The benefit of this approach is that the overall execution time can be significantly reduced by running the load in a different thread from the main thread. Performing preprocessing, such as resizing frames to a fixed size, in the different thread could be faster than loading from image files. Furthermore, this process can be performed on multi-GPUs to achieve higher speeds through multi-processing (e.g., PyTorch's DistributedDataParallel [65]). The disadvantage is that it requires a certain large size of main memory. We used a machine with 250 GB of main memory, and our experiment used about 100 GB of space for 30,000 frames.

5.3 Evaluation

5.3.1 Settings

We use LoL-V2T [79] for the training dataset. Since LoL-V2T is only labeled with captions and cannot be evaluated for object detection, we conduct a qualitative evaluation. We do not use SwAV embedding in this experiment because the improvement by SwAV is limited, as described in DETReg. The number of all frames in LoL-V2T is 197040, and the model is trained for five epochs as in DETReg. We resized the size of the frames input to Selective Search to 200x200. The input was also cropped to the game screen to prevent the attention of selective search from being drawn to the information UI displayed outside the game screen, as shown in Figure 5.3. We set the hyperparameters the same as in DETReg.

5.4 Results

The qualitative results are shown in Figure 5.4. Interestingly, we can observe that the model trained with DETReg can recognize the characters, even though selective search does not recognize them well. For example, the model recognizes the character on the right side of the third row of frames, but the selective search does not. In addition, while selective search surrounds the densely populated characters area with a single bounding box, the model attaches a bounding box to each character inside the area (in the center of the first or second row of the frames). However, the model also puts boxes on objects unrelated to the game rules, such as a bush (upper part of the third row of the frames) and a rock (lower left part of the fourth row of the frames). The result shows that the unsupervised learning method is effective for acquiring the ability to recognize game characters in eSports videos. DETReg can refine the noisy results of the selective search with a large amount of data, and the model is enough to recognize objects in the frame. However, it also recognizes objects that should be treated as background. This result suggests that the model with unsupervised learning from only images cannot learn game rules that depend on temporal context. Therefore, it is necessary to research learning methods to recognize characters' motion from the temporal context without teacher signals.

Figure 5.4: Qualitative results of unsupervised object detection for LoL-V2T. The left is the output of the model we trained, and the right is the output of the Selective Search we used for the supervised signal.

6

Object-Centric Tubelet Action Detection

The experiments up to this point have shown the necessity of a video understanding model that recognizes the characters in the eSports video and then understands their movements. In this chapter, we will describe our proposed model that detects objects in a video and recognizes the detailed temporal dynamics of the objects.

Several models have been proposed in video understanding with the supervised setting. In video understanding, the approach using the proposals provided by the object detection model to localize the features to objects has shown high performance. However, challenges remain in the method of applying the detector. Applying the detector to all frames [31] is costly, while applying it only to the key frame and replicating the proposals to the remaining frames [88] cannot support the intense motion. In tubelet action detection, bounding boxes and classes are predicted based on the coordinates of the center point of the box [50] generated from the features extracted by the 2D/3D CNN backbone or coarse proposals generated from inference to the previous clip [91]. However, since 2D/3D-CNN extracts temporally localized features, these approaches split the input video split into short segments for processing and use the linking algorithm for longer videos.

We leverage object queries as a medium to keep object information. The object queries were proposed by DETR [8], which showed that the Transformer decoder could transform a particular vector set into a vector set containing object coordinates and class information using image features. We extract the object's temporal dynamics from the video features using the attention mechanism with the object quries, i.e., the model uses object queries instead of features cropped from video features by bounding boxes and RoI Align. Since attention extracts the parts of the video feature that are similar to the object queries, we can avoid information loss due to duplication of the bounding box of the key frame and resizing to absorb differences in object size. In addition, we assume that the objects in the

neighboring frames do not change. If the object itself does not change, there is no need to update the object queries frequently, and it is sufficient to apply the detector only to key frames.

Inspired by Trackformer [59], we also propose a tracking algorithm for tubelet action detection. Our tracking algorithm includes object queries extracted from keyframes in the previous clip as input to the decoder in detecting the key frame in the following clip. The tubulet detected by object queries from the previous clip is linked to the corresponding tubulet in the previous frame.

We choose the tubelet action detection task, which evaluates the object's bounding box, rather than the video understanding task, which only predicts the class. We also evaluate our proposed model using a sports video dataset similar to eSports in a supervised setting as a first step toward unsupervised learning.

6.1 Approach

6.1.1 Framework Overview

The input video is first split into clips of fixed length T, and then a clip $I_i \in \mathbb{R}^{T\times 3\times H\times W}$ of resolution $H \times W$ is input to the model. Our model consists of extracting object queries from the center frame of a clip by the object detector and transforming them into tubelet queries using features from all frames of the clip. Each object query for an object o is represented by d-dimensional vector $\mathbf{z}_o^{\text{obj}} \in \mathbb{R}^d$, and each tubelet query for a tube tu is represented by $\mathbf{z}_{tu}^{\text{tube}} \in \mathbb{R}^{T\times d}$. According to Trackformer, we use Deformable DETR [103] for the object detection model in our model. Tubelet bounding boxes and class labels are predicted in parallel from the tubelet queries. The object queries in clip i are concatenated with the initial object queries and reused as the input of the detector in clip $i + 1$. This process is applied repeatedly to all clips to predict the tubelets for the entire video. An overview of the proposed approach can be seen in Figure 6.1.

6.1.2 Extracting Object Features

We assume that most objects in a clip are in the center frame and apply the object detector to the center frame to extract detailed spatial information about the objects. We follow Trackformer to use Deformable DETR, which consists of the ResNet50 [30] CNN backbone and Transformer encoder-decoder architecture. The image features $\mathbf{z}^{\text{img}}$ of the center frame are extracted by ResNet50 and then refined by the Transformer encoder. The Transformer decoder transforms N embeddings $\mathbf{z}^{\text{obj}}$ of size

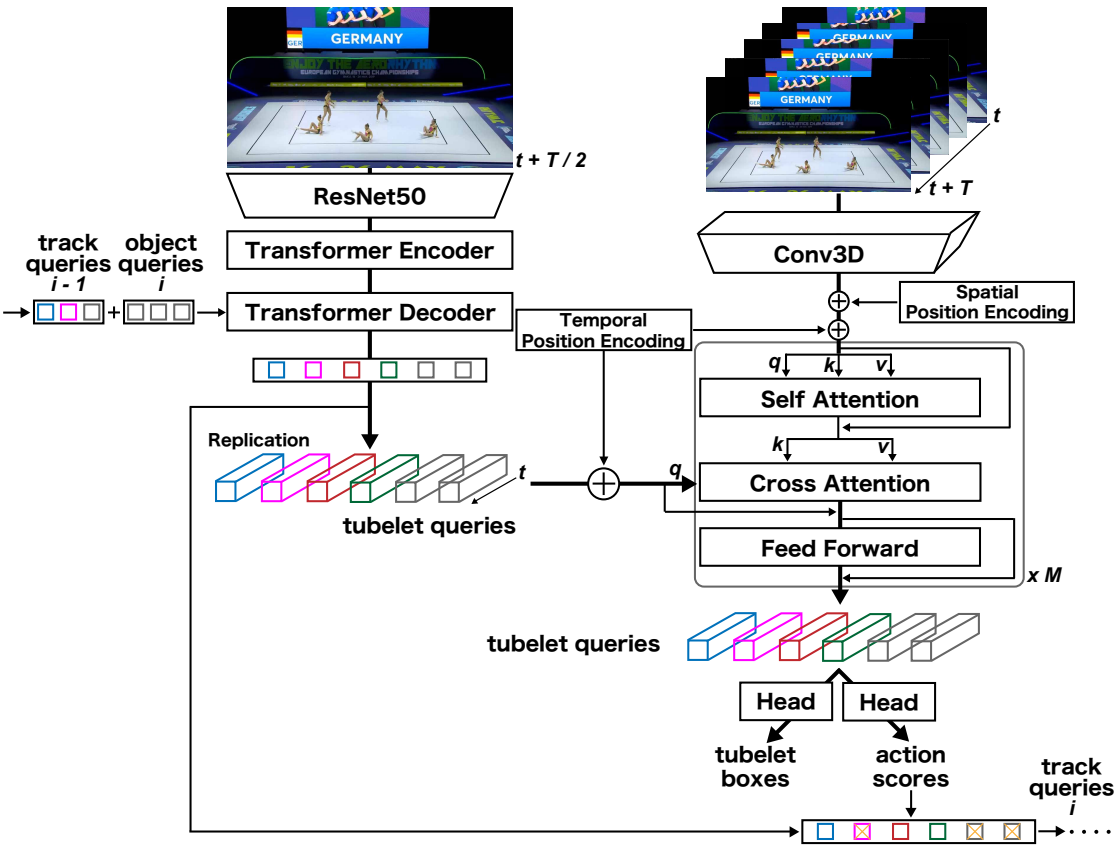


Figure 6.1: Overview of our proposed tubelet action detection model.

d into the embeddings containing object identification and position, which are called *object queries*.

$$\mathbf{z}^{\text{img}} = \text{Encoder}(\text{Backbone}(I_i^{(\frac{T}{2})})) \tag{6.1}$$

$$\mathbf{z}^{\text{obj}} = \text{Decoder}(\mathbf{z}^{\text{img}}, \mathbf{z}^{\text{obj}}) \tag{6.2}$$

$$\mathbf{y}^{\text{box}} = \text{Head}_{\text{box}}(\mathbf{z}^{\text{obj}}) \tag{6.3}$$

$$\mathbf{y}^{\text{cls}} = \text{Head}_{\text{cls}}(\mathbf{z}^{\text{obj}}), \tag{6.4}$$

where $I_i^{(\frac{T}{2})}$ denotes the center frame of the clip I_i, and $\mathbf{z}^{\text{obj}} = \{\mathbf{z}_0^{\text{obj}} ... \mathbf{z}_N^{\text{obj}}\}$ denotes a set of N object queries. Although the object detection model is trained to predict the bounding boxes and classes of the objects in a frame (Eq. (6.3) and Eq. (6.4)), we use the output queries of the Transformer Decoder in Deformable DETR directly as the object queries (Eq. (6.2)). Deformable DETR is pre-trained on the frames containing one or more bounding boxes in the target dataset, and the weights are frozen when training and inference in tubelet action detection. The output queries are applied non maximum suppression (NMS) to the predictions of the frozen bounding box head and class head to reduce redundant queries.

6.1.3 Object-Centric Tubelet Attention

In order to add temporal dynamics to the extracted object queries, we refine the object queries according to the clip features using attention. To begin this process, we convert the clip into a sequence of ST tokens $\mathbf{x}_{st} \in \mathbb{R}^d$, for a spatial resolution of S and a temporal resolution of T, following the existing video transformers [2, 6, 66]. We use a cuboid embedding [2, 66] for projecting the input volume to $\mathbb{R}^d$, equivalent to a 3D convolution with downsampling. The clip embeddings is added to a learnable position encoding $e \in \mathbb{R}^d$ for spatial and temporal dimensions separately, $\mathbf{z}_{st}^{\text{clip}} = \mathbf{x}_{st} + e_s^s + e_t^t$. The object queries are replicated by temporal resolution T, $\mathbf{z}_o^{\text{obj}} \in \mathbb{R}^{T \times d}$, and are added to the temporal position encoding e_t^t. For simplicity, we use the dimension of single-head attention as the same dimension as multi-head attention in the following.

The clip embeddings are updated to aggregate spatiotemporal features in the clip by a sequence of the Transformer layers consisting of layer normalization [3], multi-head attention (MHA) [83], residual connection [30], and a feed-forward network (MLP), as in the Transformer decoder [84]. Consider a set of query, key, and value vectors $\mathbf{q}, \mathbf{k}, \mathbf{v}$ for the input of the MHA. For the clip embeddings, the MHA is a self-attention and is represented as

$$\tilde{\mathbf{q}} = \text{MHA}(\text{LN}(\mathbf{q}), \text{LN}(\mathbf{k}), \text{LN}(\mathbf{v})) + \mathbf{q} \tag{6.5}$$

The inputs are calculated as linear projections:

$$\mathbf{q}_{st} = \mathbf{W}_q \mathbf{z}_{st}^{\text{clip}}, \mathbf{k}_{st} = \mathbf{W}_k \mathbf{z}_{st}^{\text{clip}}, \mathbf{v}_{st} = \mathbf{W}_v \mathbf{z}_{st}^{\text{clip}}, \tag{6.6}$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d \times d}$ are projection matrices. The output $\tilde{\mathbf{q}}$ is the updated clip embeddings $\tilde{\mathbf{z}}^{\text{clip}} \in \mathbb{R}^{ST \times d}$ and becomes the input $\mathbf{k}$ and $\mathbf{v}$ of the next layer.

The tubelet queries are obtained by the cross attention on the updated clip embeddings and the object queries. The inputs of the cross attention are calculated similarly to Eq. (6.6)

$$\mathbf{q}_{ot} = \mathbf{W}_q \mathbf{z}_{ot}^{\text{obj}}, \mathbf{k}_{st} = \mathbf{W}_k \tilde{\mathbf{z}}_{st}^{\text{clip}}, \mathbf{v}_{st} = \mathbf{W}_v \tilde{\mathbf{z}}_{st}^{\text{clip}}, \tag{6.7}$$

The cross attention operation is the same as in Eq. (6.5). The tubelet queries are further input to the MLP.

$$\tilde{\mathbf{q}} = \text{MLP}(\text{LN}(\tilde{\mathbf{q}})) + \tilde{\mathbf{q}}, \tag{6.8}$$

The tubelet queries are the outputs $\tilde{\mathbf{q}} \in \mathbb{R}^{T \times O \times d}$ of M iterations of the self-attention, cross attention, MLP, and layer normalization.

We use Trajectory attention [66] for the MHA, which divides the attention operation into two stages corresponding to space and time. It is superior to the joint space-time attention [2, 6] in its capture of temporal dynamics. In addition, introducing the approximating scheme to the attention operation reduces the computational cost and memory requirements.

The bounding boxes $\mathbf{b} \in [0, 1]^{N \times T \times 4}$ and class probabilities $\mathbf{p} \in [0, 1]^{N \times T \times N_{\text{class}}}$ are computed in parallel using the MLP heads from the tubelet queries. Note that the probability of classes per tube is calculated by the average of $\mathbf{p}$ in the temporal dimension.

6.1.4 Tubelet Tracking with Queries

In order to achieve tracking objects between clips, we introduce the concept of track queries into our proposed approach based on Trackformer [59]. Track queries trace the object's appearance through a video sequence while adapting to changes in shape and position as the object motions.

Track queries are initialized using the object queries detected in the previous clip $i - 1$. The valid object queries have a tubelet classification score above σ_{object} and do not predict the background class. The N_{track} valid object queries in clip $i - 1$ are combined as track queries with object queries for clip i and are input to the decoder in Eq. (6.9). The number of object queries in clip i is thus $N + N_{\text{track}}$. Note that no track queries are used for clip 0.

$$\mathbf{z}^{\text{obj}} = \text{Decoder}(\mathbf{z}^{\text{img}}, \text{Concat}(\mathbf{z}^{\text{obj}}, \mathbf{z}^{\text{track}})), \tag{6.9}$$

Once a query has started tracking, it is removed from the track queries when its tubelet classification score drops below σ_{tack}, or when non-maximum suppression (NMS) deletes it.

6.2 Training

We first find optimal bipartite matching δ between the predicted tubelets and ground truth tubelets, and then compute the objective function between the matched pairs. Let use denote by y the ground truth set of tubelets, and $\hat{y} = \{\hat{y}_i\}_{i=1}^{N+N_{\text{track}}}$ the set of $N + N_{\text{track}}$ predicted tubelets. We define the cost function as Eq. (6.10) and search the index $\delta(i)$ of the prediction for y_i at minimum cost by Hungarian algorithm [43].

Let use denote by $y_i = (c_i, \mathbf{b}_i)$ the each element of ground truth set, and $\hat{y}_j = (\hat{c}_j, \hat{\mathbf{b}}_j)$ one of $N + N_{\text{track}}$ predicted tubelets. c_i and $\hat{c}_j$ are the class labels. $\mathbf{b}_i \in [0,1]^{(T' \times 4)}$ is the set of center coodinates and its height and width of box in the ground truth tubelet; since an object may only appear in part of the clip, the number of box is denoted as $T' \leqq T$. On the other hand, since the prediction is generated for the entire clip, the number of the prediction $\hat{\mathbf{b}}_j \in [0,1]^{(T \times 4)}$ is T. We consider y as a set of size $N + N_{\text{track}}$ padded with $\varnothing$ (no tubelet), and we will denote $N + N_{\text{track}}$ as N in the following. The cost function $\mathcal{L}_{\text{match}}$ is defined as

$$\mathcal{L}_{\text{match}} = \lambda_{\text{cls}} \mathcal{L}_{\text{class}} + \mathcal{L}_{\text{box}} \tag{6.10}$$

$$\mathcal{L}_{\text{class}} = \sum_{i=1}^{N} -\log \hat{p}_{\delta(i)}(c_i) \tag{6.11}$$

$$\mathcal{L}_{\text{box}} = \sum_{i=1}^{N} \mathbb{1}_{\{c_i \neq \varnothing\}} \sum_{t \in \psi} \lambda_{\text{iou}} \mathcal{L}_{\text{iou}}(\mathbf{b}_i^{(t)}, \mathbf{b}_{\delta(i)}^{(t)}) + \lambda_{\text{L1}} \|\mathbf{b}_i^{(t)} - \mathbf{b}_{\delta(i)}^{(t)}\|, \tag{6.12}$$

where $\hat{p}_{\delta(i)}(c_i)$ is the probability of class c_i, and ψ is the product set of the frames in the ground truth tube and the prediction tube. We use the generalized IoU [72] loss for $\mathcal{L}_{\text{iou}}$. $\mathcal{L}_{\text{iou}}$ and L1 loss are normalized by the number of tubelets inside the batch. $\mathcal{L}_{\text{match}}$ is used not only as a cost function for the bipartite matching, but also as an objective function for the training.

6.3 Evaluation

This section describes the details of the evaluation for our approach.

6.3.1 Dataset

We perform experiments on MultiSports [49], containing 3200 sports video clips with 66 fine-grained classes in four sports. Compared to other similar datasets [38, 76], the spatial resolution of 1280x720

is higher, and multi-person boxes are labeled per frame. The temporal resolution is 25 FPS. It is similar to eSports videos in that the object size is significantly tiny relative to the frame size, multi-object in a frame, and object's motion is fast and subtle.

6.3.2 Baselines and Metrics

We use two baselines, MOC [50] and Motionformer [66]. MOC first predicts the coordinates of the center point of the box from 2D/3D CNN backbone features and then predicts bounding boxes and classes based on the coordinates. Motionformer [66] is a video understanding model using trajectory attention. We replace self-attention in Motionformer with cross attention as in Eq. (6.7) and use tubelet queries as $\mathbf{z}^{\text{obj}}$, which we call this *Motionformer+TQ*. Our method differs from Motionformer+TQ in that object queries from the detector are used for the tubelet queries.

We report the video-mAP [26] with different IoU thresholds to evaluate spatiotemporal action detection.

6.3.3 Implementation Details

We follow the hyperparameters in Deformable DETR [103] for our object detector. We initialize with the model weights from [103] pre-trained on COCO [53]. Our object detector operates $1 \times 480 \times 854$, while our video model operates $16 \times 224 \times 224$ videos with temporal stride 2. Our model uses patch-size $2 \times 16 \times 16$. The trajectory attention has 6 layers, 8 heads, and an embedding dimension of 256. We empirically set the number of tubelet queries to 32 and σ_{object} and σ_{track} to 0.4. We use the AdamW [56] optimizer with initial learning rate $2.0e-4$ for tubelet action detection. During inference, NMS with threshold 0.4 is applied to the predicted boxes in the center frame to reduce redundant tubelets. All models are trained for 50 epochs.

6.3.4 Results

In Table 6.1, we compare our method against the baselines. The MOC results are borrowed from [49]. We find that our method performs favorably against the baselines in both IoU thresholds. In particular, it outperforms MOC by a wide margin in video-mAP@0.5. Our method is also significantly higher than Motionformer + TQ, suggesting that object queries from the detector are the key to performance.

We show the qualitative results of our approach and Motionformer + TQ in Figure 6.2. We can observe that ours can detect multi-person individually while Motionformer + TQ can detect only a single person. We can also see that ours generates the correct box without confusion about changes in the scene. For Motionformer + TQ, the detected box only covers a little of the ground truth, even the player's face, which is large on the screen.

Figure 6.2: Qualitative results of tubelet action detection. Green indicates ground truth and red indicates prediction.

Table 6.1: General results for tubelet action detection with MultiSports [49].

Method	video-mAP@0.2	video-mAP@0.5
MOC [50]	12.13	0.77
Motionformer + TQ [66]	3.45	0.15
Ours	**14.89**	**4.89**

In summary, we show from Table 6.1 and Figure 6.2 that our proposed method is superior to the baselines.

7 Conclusions

7.1 Summary

In this thesis, we investigated eSports video captioning through three approaches.

First, we experimentally evaluate existing video feature extraction methods to the eSports domain through video captioning. We confirmed that the existing captioning models struggle significantly with eSports videos more than human action videos, and improvements are needed in the video feature extraction model.

Second, we experimented with unsupervised learning in object detection to recognize eSports characters with unlabeled data. We realized that the model needs to understand the temporal dynamics of the game rules and the characters.

Finally, we extend the video understanding model based on Transformer with an object detection model to recognize object-centric temporal dynamics. We introduce object queries to address the problem of existing methods: the different sizes of detected objects and the high cost of high-frequency object detection. We also use the recurrent structure to identify tubelet correspondence between clips without any heuristic linking algorithm. Our experimental results showed that our approach outperformed baselines and demonstrated the effectiveness of object queries from the detector.

7.2 Future Directions

A limitation of this study is that we left to train our object-centric action tubelet detection model with unlabeled eSports videos. We aim to generate captions using our object-centric video feature extraction model in the eSports domain where datasets are not well-developed. As a first step, we proposed a feature extraction model and evaluated it on a well-developed sports video dataset in the supervised

setting. Future work will extend our model to the unsupervised setting and bring the model to the feature extraction part of video captioning.

References

[1] Nayyer Aafaq, Naveed Akhtar, Wei Liu, Syed Zulqarnain Gilani, and Ajmal Mian. "Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12487–12496.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. "Vivit: A video vision transformer". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6836–6846.

[3] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer Normalization". In: *CoRR* abs/1607.06450 (2016). arXiv: 1607.06450.

[4] Satanjeev Banerjee and Alon Lavie. "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments". In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. June 2005, pp. 65–72. URL: https://aclanthology.org/W05-0909.

[5] Amir Bar, Xin Wang, Vadim Kantorov, Colorado J Reed, Roei Herzig, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. "Detreg: Unsupervised pretraining with region priors for object detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 14605–14615.

[6] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. "Is space-time attention all you need for video understanding?" In: *ICML*. Vol. 2. 3. 2021, p. 4.

[7] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. "End-to-end object detection with transformers". In: *European conference on computer vision*. Springer. 2020, pp. 213–229.

[9] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. "Unsupervised learning of visual features by contrasting cluster assignments". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9912–9924.

[10] Joao Carreira and Andrew Zisserman. “Quo vadis, action recognition? a new model and the kinetics dataset”. In: *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6299–6308.

[11] Shaoxiang Chen and Yu-Gang Jiang. “Motion guided region message passing for video captioning”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 1543–1552.

[12] Shoufa Chen, Peize Sun, Enze Xie, Chongjian Ge, Jiannan Wu, Lan Ma, Jiajun Shen, and Ping Luo. “Watch Only Once: An End-to-End Video Action Detection Framework”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 8178–8187.

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. “A simple framework for contrastive learning of visual representations”. In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[14] MMAction2 Contributors. *OpenMMLab’s Next Generation Video Understanding Toolbox and Benchmark*. https://github.com/open-mmlab/mmaction2. 2020.

[15] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. “Up-detr: Unsupervised pre-training for object detection with transformers”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 1601–1610.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[17] Shuangrui Ding, Maomao Li, Tianyu Yang, Rui Qian, Haohang Xu, Qingyi Chen, Jue Wang, and Hongkai Xiong. “Motion-Aware Contrastive Video Representation Learning via Foreground-Background Merging”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9716–9726.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *ICLR* (2021).

[19] Bernard Ghanem Fabian Caba Heilbron Victor Escorcia and Juan Carlos Niebles. “ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 961–970.

[20] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. "Multiscale vision transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6824–6835.

[21] Gueter Josmy Faure, Min-Hung Chen, and Shang-Hong Lai. "Holistic Interaction Transformer Network for Action Detection". In: *arXiv preprint arXiv:2210.12686* (2022).

[22] Christoph Feichtenhofer. "X3d: Expanding architectures for efficient video recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 203–213.

[23] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. "Slowfast networks for video recognition". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 6202–6211.

[24] Kirill Gavrilyuk, Ryan Sanford, Mehrsan Javan, and Cees GM Snoek. "Actor-transformers for group activity recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 839–848.

[25] Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. "Coot: Cooperative hierarchical transformer for video-text representation learning". In: *Advances in neural information processing systems* 33 (2020), pp. 22605–22618.

[26] Georgia Gkioxari and Jitendra Malik. "Finding action tubes". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 759–768.

[27] Mingfei Han, David Junhao Zhang, Yali Wang, Rui Yan, Lina Yao, Xiaojun Chang, and Yu Qiao. "Dual-AI: Dual-path Actor Interaction Learning for Group Activity Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2990–2999.

[28] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738.

[29] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[31] Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. "Object-region video transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3148–3159.

[32] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[33] Rui Hou, Chen Chen, and Mubarak Shah. “Tube convolutional neural network (T-CNN) for action detection in videos”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5822–5831.

[34] Lianghua Huang, Yu Liu, Bin Wang, Pan Pan, Yinghui Xu, and Rong Jin. “Self-supervised video representation learning by context and motion decoupling”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13886–13895.

[35] Allan Jabri, Andrew Owens, and Alexei Efros. “Space-Time Correspondence as a Contrastive Random Walk”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. 2020, pp. 19545–19560. URL: https://proceedings.neurips.cc/paper/2020/file/e2ef524fbf3d9fe611d5a8e90fefdc9c-Paper.pdf.

[36] Mihir Jain, Jan Van Gemert, Hervé Jégou, Patrick Bouthemy, and Cees G.M. Snoek. “Action Localization with Tubelets from Motion”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 740–747.

[37] Simon Jenni and Hailin Jin. “Time-equivariant contrastive video representation learning”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 9970–9980.

[38] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. “Towards understanding action recognition”. In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 3192–3199.

[39] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. “Action tubelet detector for spatio-temporal action localization”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4405–4413.

[40] Donghyun Kim, Yi-Hsuan Tsai, Bingbing Zhuang, Xiang Yu, Stan Sclaroff, Kate Saenko, and Manmohan Chandraker. “Learning cross-modal contrastive features for video domain adaptation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13618–13627.

[41] Matthew Kowal, Mennatullah Siam, Md Amirul Islam, Neil DB Bruce, Richard P Wildes, and Konstantinos G Derpanis. “A Deeper Dive Into What Deep Spatiotemporal Networks Encode: Quantifying Static vs. Dynamic Information”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 13999–14009.

[42] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. "Dense-Captioning Events in Videos". In: *International Conference on Computer Vision (ICCV)*. 2017.

[43] Harold W Kuhn. "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[44] Jie Lei, Liwei Wang, Yelong Shen, Dong Yu, Tamara L Berg, and Mohit Bansal. "MART: Memory-Augmented Recurrent Transformer for Coherent Video Paragraph Captioning". In: *ACL*. 2020.

[45] Dong Li, Zhaofan Qiu, Qi Dai, Ting Yao, and Tao Mei. "Recurrent Tubelet Proposal and Recognition Networks for Action Detection". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.

[46] Shuaicheng Li, Qianggang Cao, Lingbo Liu, Kunlin Yang, Shinan Liu, Jun Hou, and Shuai Yi. "Groupformer: Group activity recognition with clustered spatial-temporal transformer". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13668–13677.

[47] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. "MViTv2: Improved multiscale vision transformers for classification and detection". In: *CVPR*. 2022.

[48] Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. "MultiSports: A Multi-Person Video Dataset of Spatio-Temporally Localized Sports Actions". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 13536–13545.

[49] Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. "Multisports: A multi-person video dataset of spatio-temporally localized sports actions". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13536–13545.

[50] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. "Actions as moving points". In: *European Conference on Computer Vision*. Springer. 2020, pp. 68–84.

[51] Yuxi Li, Weiyao Lin, John See, Ning Xu, Shugong Xu, Ke Yan, and Cong Yang. "Cfad: Coarse-to-fine action detector for spatiotemporal action localization". In: *European Conference on Computer Vision*. Springer. 2020, pp. 510–527.

[52] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. "SwinBERT: End-to-end transformers with sparse attention for video captioning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 17949–17958.

[53] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[54] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. "Video Swin Transformer". In: *arXiv preprint arXiv:2106.13230* (2021).

[55] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. "Video swin transformer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3202–3211.

[56] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[57] Karttikeya Mangalam, Haoqi Fan, Yanghao Li, Chao-Yuan Wu, Bo Xiong, Christoph Feichtenhofer, and Jitendra Malik. "Reversible Vision Transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10830–10840.

[58] Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell. "Something-else: Compositional action recognition with spatial-temporal interaction networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 1049–1059.

[59] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. "Trackformer: Multi-object tracking with transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 8844–8854.

[60] Newzoo, ed. *Global Esports & Live Streaming Market Report Free Version*. 2022.

[61] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding". In: *arXiv preprint arXiv:1807.03748* (2018).

[62] Boxiao Pan, Haoye Cai, De-An Huang, Kuan-Hui Lee, Adrien Gaidon, Ehsan Adeli, and Juan Carlos Niebles. "Spatio-temporal graph for video captioning with knowledge distillation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10870–10879.

[63] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. "Actor-context-actor relation network for spatio-temporal action localization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 464–474.

[64] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. "Videomoco: Contrastive video representation learning with temporally adversarial examples". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11205–11214.

[65] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[66] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. "Keeping your eye on the ball: Trajectory attention in video transformers". In: vol. 34. 2021, pp. 12493–12506.

[67] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. "Spatiotemporal contrastive video representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 6964–6974.

[68] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. "Learning transferable visual models from natural language supervision". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763.

[69] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. "Do Vision Transformers See Like Convolutional Neural Networks?" In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. 2021. URL: https://openreview.net/forum?id=Gl8FHfMVTZu.

[70] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. "Hierarchical text-conditional image generation with clip latents". In: *arXiv preprint arXiv:2204.06125* (2022).

[71] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. 2015. URL: https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

[72] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. "Generalized intersection over union: A metric and a loss for bounding box regression". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 658–666.

[73] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding". In: *arXiv preprint arXiv:2205.11487* (2022).

[74] Karen Simonyan and Andrew Zisserman. “Two-stream convolutional networks for action recognition in videos”. In: *Advances in neural information processing systems* 27 (2014).

[75] Lin Song, Shiwei Zhang, Gang Yu, and Hongbin Sun. “Tacnet: Transition-aware context network for spatio-temporal action detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11987–11995.

[76] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. “UCF101: A dataset of 101 human actions classes from videos in the wild”. In: *arXiv preprint arXiv:1212.0402* (2012).

[77] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. “Videobert: A joint model for video and language representation learning”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 7464–7473.

[78] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. “Rethinking the inception architecture for computer vision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[79] Tsunehiko Tanaka and Edgar Simo-Serra. “LoL-V2T: Large-Scale Esports Video Description Dataset”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2021, pp. 4557–4566.

[80] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. “Learning spatiotemporal features with 3d convolutional networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4489–4497.

[81] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. “Selective search for object recognition”. In: *International journal of computer vision* 104.2 (2013), pp. 154–171.

[82] Jayesh Vaidya, Arulkumar Subramaniam, and Anurag Mittal. “Co-Segmentation Aided Two-Stream Architecture for Video Captioning”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 2774–2784.

[83] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[84] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention Is All You Need”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Dec. 4, 2017, pp. 6000–6010. ISBN: 978-1-5108-6096-4.

[85] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. "Temporal segment networks: Towards good practices for deep action recognition". In: *European conference on computer vision*. Springer. 2016, pp. 20–36.

[86] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. "End-to-End Dense Video Captioning with Parallel Decoding". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6847–6857.

[87] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. "Track to detect and segment: An online multi-object tracker". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 12352–12361.

[88] Jianchao Wu, Zhanghui Kuang, Limin Wang, Wayne Zhang, and Gangshan Wu. "Context-aware rcnn: A baseline for action detection in videos". In: *European Conference on Computer Vision*. Springer. 2020, pp. 440–456.

[89] Junfei Xiao, Longlong Jing, Lin Zhang, Ju He, Qi She, Zongwei Zhou, Alan Yuille, and Yingwei Li. "Learning from temporal gradient for semi-supervised action recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3252–3262.

[90] Yuanjun Xiong, Limin Wang, Zhe Wang, Bowen Zhang, Hang Song, Wei Li, Dahua Lin, Yu Qiao, Luc Van Gool, and Xiaoou Tang. "Cuhk & ethz & siat submission to activitynet challenge 2016". In: *arXiv preprint arXiv:1608.00797* (2016).

[91] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. "Step: Spatio-temporal progressive learning for video action detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 264–272.

[92] Hanhua Ye, Guorong Li, Yuankai Qi, Shuhui Wang, Qingming Huang, and Ming-Hsuan Yang. "Hierarchical Modular Network for Video Captioning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 17939–17948.

[93] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. "Motr: End-to-end multiple-object tracking with transformer". In: *European Conference on Computer Vision*. Springer. 2022, pp. 659–675.

[94] Junchao Zhang and Yuxin Peng. "Object-aware aggregation with bidirectional temporal graph for video captioning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8327–8336.

[95] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. "Object relational graph with teacher-recommended learning for video captioning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 13278–13288.

[96] Jiaojiao Zhao and Cees GM Snoek. "Dance with flow: Two-in-one stream action detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9935–9944.

[97] Jiaojiao Zhao, Yanyi Zhang, Xinyu Li, Hao Chen, Bing Shuai, Mingze Xu, Chunhui Liu, Kaustav Kundu, Yuanjun Xiong, Davide Modolo, et al. "TubeR: Tubelet transformer for video action detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 13598–13607.

[98] Qi Zheng, Chaoyue Wang, and Dacheng Tao. "Syntax-Aware Action Targeting for Video Captioning". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[99] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. "Grounded Video Description". In: *CVPR*. 2019.

[100] Luowei Zhou, Chenliang Xu, and Jason J Corso. "Towards automatic learning of procedures from web instructional videos". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[101] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. "End-to-end dense video captioning with masked transformer". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8739–8748.

[102] Linchao Zhu and Yi Yang. "Actbert: Learning global-local video-text representations". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8746–8755.

[103] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. "Deformable {DETR}: Deformable Transformers for End-to-End Object Detection". In: *International Conference on Learning Representations*. 2021.