POLITECNICO DI TORINO

DEPARTMENT OF CONTROL AND COMPUTER ENGINEERING

Master Degree In Computer Engineering

Master Thesis

A Machine-learning Approach for Video Streaming Provisioning at the Network Edge

Relatore

prof. Chiasserini Carla Fabiana

Correlatori:

Somreeta Pramanik

Puligheddu Corradoi

Candidate

Tung Pak Man

matricola: S245432

Academic Year 2019 – 2020

To my family

Abstract

5G is the 5th generation mobile network, the planned successor to the 4G networks which provide connectivity to most current cellular devices .5G enables a new kind of network that is designed to connect virtually everyone and everything together, including machines, objects, and devices. The new mobile network provides greater bandwidth, giving higher download speeds, up to 10 gigabits per second. Owing to higher bandwidth than before, it is expected that the new networks will not just serve cellphones like before, but also serve for laptops and desktop computers. The workload of the network data center increases significantly, especially in the edge data center in which the resource is limited. This thesis aims to create an AI model for optimal resource allocation, which controls the workload to avoid overload in the edge data center.

There are different types of services, e.g AR/VR (Augmented Reality and Virtual Reality), V2X (Vehicle-toEverything), Live Video Streaming, manufacturing and health. Failure in safety service will cause damage to human lives, while failure in entertainment service will affect the user experience. Live Video Streaming is the primary concern of this thesis. Live streaming is when the streamed video is sent over the Internet in real-time, without first being recorded and stored. There are a lot of steps which take place behind the scenes in a live stream and only the encoding part will be investigated. The final objective of this study seeks to create a machine-learning algorithm to control the encoding process in Live Streaming to avoid latency due to overload.

Acknowledgements

Foremost, I wish to express my deepest gratitude to Professor CHIASSERINI CARLA FABIANA as well as my research supervisor, for her patient guidance, enthusiastic encouragement and useful critiques of this thesis, without whom I would not have been able to complete this thesis. She trusts me, gives me the opportunities to work on this thesis and helps me with it all the time from the beginning till the end.

Secondly, I would like to thank Somreeta Pramanik and Corrado Puligheddu. I appreciated their availability in answering and communicating even during the holiday, steering me in the right direction.

Last but not the least, a special thanks to everyone who has supported and encouraged me on this journey.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The goal of this thesis is to construct a machine learning (ML) model for the evaluation of video streaming. The computing resources in the edge data center is limited, improper encoding decision will lead to a miserable QoE for the user. In other words, the ML model will learn the optimize encoding parameters in a containerized environment that holds the video streaming application. The ML model takes video metadata and available CPU resources as input, it outputs the optimized parameters for encoding in order to attain a sufficient QoE for the user. The analysis is done only internally(edge data center), the external factors which will affect the QoE are ignored, such as transmitting time in the network and computing power in the client-side. This thesis work will include both the data generation and ML model construction.

Content of the thesis

- Chapter 2 gives an overview of the video streaming and edge data center. It discusses how those two technologies cooperate to provide a new kind of service to users.
- Chapter 3 introduces the adopted methodology of the machine learning algorithm and the video quality assessment method.
- Chapter 4 describes the tools used to build the experimental environment.

- Chapter 5 analyses the data collection method and discusses the impact of different encoding parameters in encoding.
- Chapter 6 analyses the performance of the Random Forest classifier and the result.
- Chapter 7 concludes the work done in this thesis and possible future exploring.

Chapter 2

Background Knowledge

The fifth-generation (5G) mobile network is identical to the early second generation (2G), third generation (3G), and fourth generation (4G) mobile networks. It is a digital cellular network in which the service area covered by the provider is divided into many small physical areas called cells. In a mobile phone, analogue signals representing sound and images are digitised, converted by an analogue-to-digital converter, and transmitted as a bitstream. All 5G wireless devices in a cell communicate with the local antenna array and low-power automatic transceivers (transmitters and receivers) in the cell through radio waves. The transceiver allocates frequency channels from a common frequency pool, which can be reused . The local antenna is connected to the telephone network and the internet through a higher bandwidth optical fibre or a wireless backhaul connection. As with mobile phones, when users migrate from one cell to another, their mobile device automatically 'switches' to the antenna in the new cell. The main advantage of 5G is that the data transmission rate is considerably higher than the 4G cellular network (by up to 10 Gbit/s), higher than the current wired internet, and higher than the 4G long-term-evolution cellular network (by 100 times). Another advantage is a lower network latency (faster response time; <1 ms) than the 4G network (30–70 ms). Owing to faster data transmission, 5G networks will not only provide services for mobile phones but also for serving general home and office network , competing with wired network providers. While previous-generation cellular networks provided low-data-rate internet access that was suitable for mobile phones, a cell phone tower could not economically provide sufficient bandwidth as a general

internet provider for home computers. Figure shows the advantages of 5G networks over 4G networks.

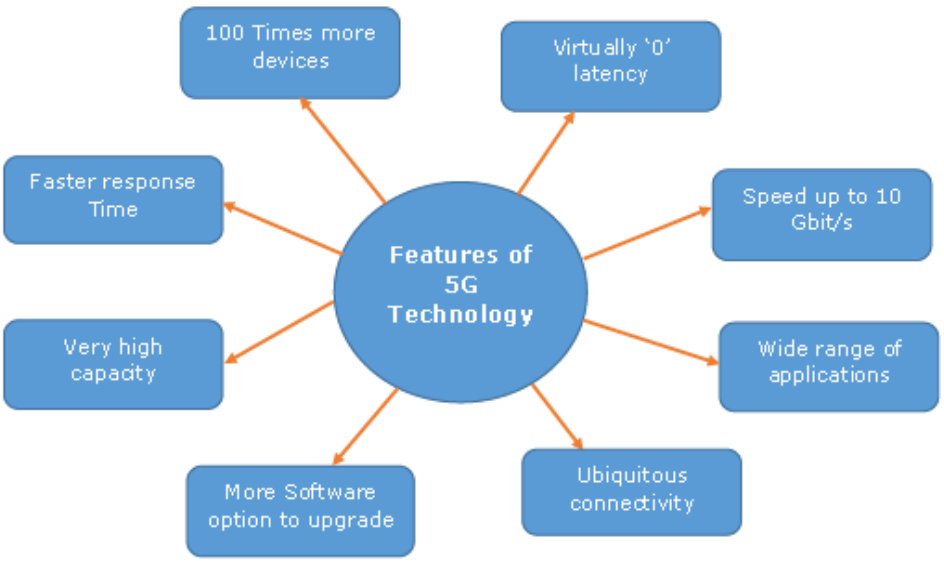


Figure 2.1: .
Advantages of 5G networks compared with 4G networks

2.1 Edge Computing

The 5G network is the next-generation cellular network, and it is characterised by ultra-high throughput, density, and extremely fast internet. Nevertheless, it is capable of much more. It can improve life-critical systems and Internet of Things (IoT) applications far beyond what we have seen with 4G networks. It can be expected that billions of devices that directly impact our lives in terms of communication, transportation, health, and entertainment would be required. Some services in the 5G network will require a latency as low as 1 ms. The 5G network will enable new applications requiring low latency and high reliability, thus providing high quality-of-experience services to billions of users globally.

Edge computing is an emerging technology in which the distance between end users (or user equipment, UE) and computing resources is reduced, in order to overcome the problems of the traditional cloud, such as high latency and poor security. It offers extraordinary benefits for data processing, service provisioning and resource optimisation. This section briefly introduces edge computing and discusses its importance and advantages. It also presents a few examples across industries.

2.1.1 A Changing Computing World

Prior to a discussion of the need for edge computing, an assessment of the performance of its predecessor, 'cloud computing', for IoT devices globally would be instructive. From household appliances to live support equipment and from smartphones to smartwatches, internet-connected devices are ubiquitous. Recent researches have predicted that the global

IoT market will exceed 1.1 trillion dollars by 2026. Consequently, cloud computing has become an increasingly dominant trend.

Cloud computing allows companies to store and process data outside their own physical hardware, in remote servers referred to as 'cloud'. For example, people may choose to back up their smartphone date on Apple's iCloud instead of their computer. They can then access their smartphone data via another internet-enabled device since the data are not confined to the internal hard drive on their smartphone or desktop. Another example is running full-scale applications that can be accessed

through a browser.

However, centralised cloud computing is not suitable for all types of applications. In the future, billions of devices will connect to the internet, and faster and reliable data processing will become important. The emergence of these devices will place heavy demands on networking bandwidth.

Data can be divided into three categories according to their temporal characteristics:

1. Time-critical data: It has a predefined latency. Examples are live video streaming, gaming and healthcare applications.

2. Semi-time-critical data: It has a predefined latency too, but it can tolerate some delay

3. Non-Time-critical data: It is not time sensitive and can tolerate latency.

Edge computing was developed for applications and services with time-critical requirements, and it involves the use of edge data centers physically close to UE. It facilitates data processing close to the data source, eliminating the need to transfer data to and from the cloud. For example, if a car has sensors that can provide the current status of its engine, sensor data need not be transmitted to the cloud to detect problems in the engine. Splitting the data processing and storage to local servers places less strain on computing networks. Reduced data in the network translates to faster data processing between the cloud and IoT devices. Edge computing is necessary as the traditional cloud computing model is not suitable for highly interactive that is computationally intensive and that has high throughput. This is because the cloud server may be far from the UE, which increases latency. In other words, cloud servers are usually located at the core while edge servers are usually located in some minicloud located at the edges [1]. Figure 2.2 depicts the cloud computing and edge computing models.

The reason for which edge computing is required is clarified by the following situation. In vehicle-to-vehicle communications, a real-time packet requires an end-to-end delay of less than 'X' ms. If the end-to-end delay via the cloud exceeds 'X + 100' ms, it is unacceptable.

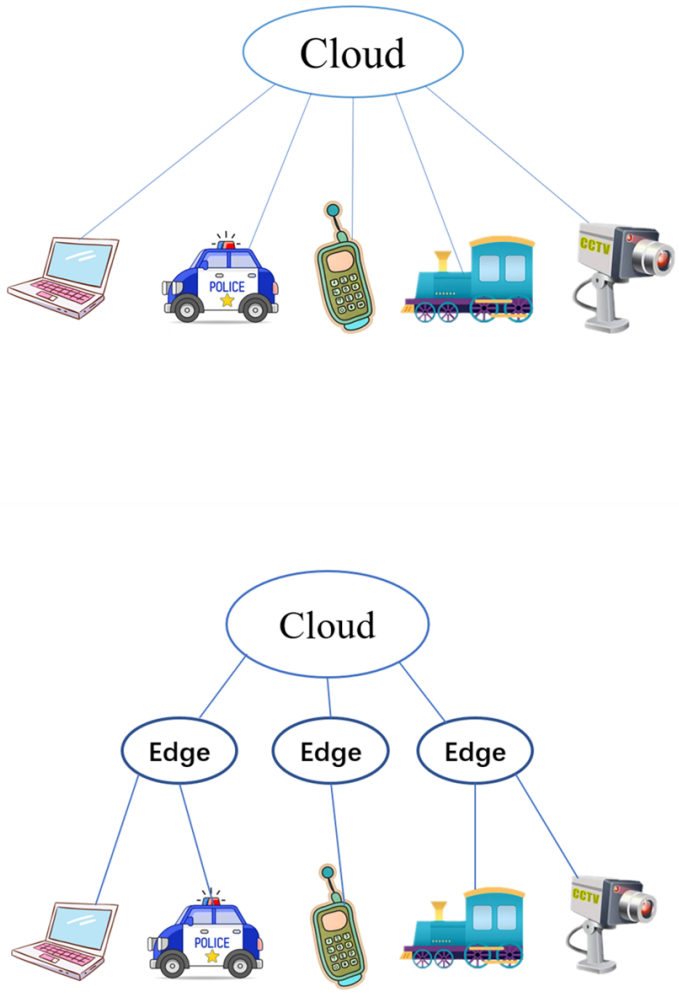


Figure 2.2: .
Cloud model (above) and edge model (below). In the cloud model, communication is directly with the cloud, while in the edge model, communication is with the edge server first. A shorter physical distance corresponds to a faster processing time.

2.1.2 Use Case and Benefits

Although edge computing is not mature, it has already provided many benefits[2] :

1. Faster and even real-time data processing: Data is processed closer to the source and not in an external data center or cloud, which reduces the lag time.

2. Lower cost: A company incurs lower cost for data management for local devices (in edge) compared with the cost of cloud and data center networks.

3. Less data traffic in the core network: The number of IoT devices is increasing with time, resulting in a significant increase in the data traffic. Consequently, the network bandwidth becomes more limited, which increases the traffic load in the core and leading to a greater bottleneck of data.

4. Improve application efficiency: With a shorter response delay than cloud computing, applications can run at higher speeds

Localisation of the data process can reduce the potential for a single point of failure. If a company uses a centralised cloud to store its data and the cloud server goes down, the data becomes inaccessible; this could have serious repercussions.

Currently, many 5G applications[1][2]require real-time data processing, local analysis, a high data rate, and high data security.

1. Healthcare Service. Remote diagnostics is pervasive in our world. Some health monitoring devices can monitor patients' vital signs without connecting to the cloud. From the data collected by these devices, doctors can evaluate patients and provide on-demand feedback about their health. Fast data processing near the source has the potential to be life-saving; an example is the prompt detection of a heart disease.

2. Multimedia Applications and Online Entertainment. These applications require a fast response time. For example, both live video streaming and on-demand video services require a small delay between frames. If the delay is high, these applications are intolerable.

3. Transportation. Self-driving vehicles are equipped with many types of sensors, ranging from camera-based systems and radar-based systems to lidar-based

systems, for their smooth operation. The vehicles should be capable of communicating with other objects on the road via the V2X solution. Autonomous vehicles could use edge computing to process the sensor data much closer to the vehicles, saving precious milliseconds that can help avoid an accident.

4. Internet of Things. Smart devices, ranging from the road sensor to the oven in the kitchen will generate massive amounts of data that will be sent on the internet. Edge Computing reduces the potential of a data bottleneck occurring.

5. Manufacturing in factories. The decreased latency from edge computing provides a faster, more responsive changes in manufacturing, which would be able to apply insight and action in real-time. This would help enhance safety and productivity. An example is the shutting down of a machine before it overheats.

2.2 Live Video Streaming

Video streaming refers to the continuous delivery of data from a video file to a remote user by a provider via a network. It allows a video to be played online without being downloaded on a local device. The use of 5G technology will lead to improvements in the media streaming industry. The first improvement will be in the quality of media streaming. The high speed of 5G networks will facilitate 4K or even 8K video streaming on devices. Another improvement will be in live streaming. The quality of media will increase while latency will decrease. With this improvement, live streaming of sports and other events will become more popular.

2.2.1 How it works?

When a video is streamed over a network in real time, without being recorded and stored, it is referred to as live streaming. Today, live streaming of TV broadcasts, social media, sports events, etc. is possible. The difference between regular video streaming and live video streaming can be considered to be similar to the difference between an actor acting in a movie and essaying a role in a drama. In the case of the movie, the content is created, stored, and then played for the audience in a

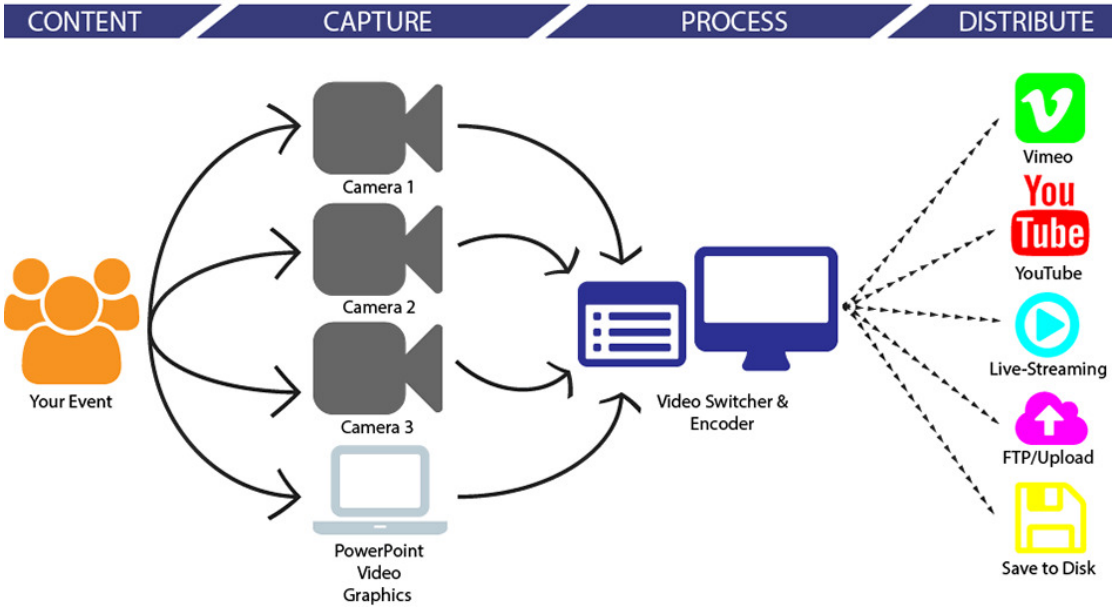


Figure 2.3: .
Workflow in live streaming [3]

cinema. In the case of the drama, the audience witnesses the actor live, similar to the case of live streaming [5].

Live streaming usually refers to the broadcast of a live stream to multiple users simultaneously through one-to-many connections. Figure 2.3 depicts the processes involved in live streaming. The main steps are as follows:

1. Segmentation
2. Compression
3. Encoding
4. Transit in network
5. Decoding
6. Video playback

Live video streaming starts with the capture of raw video data by an input device (e.g., camera). The video data is stored as digital data in a computing device to

which the camera is connected. The digital data then undergoes segmentation, namely, it is divided into small segments, each with a length of a few seconds. The reason for segmentation is that it is not appropriate to send all the video data together over the internet since there may be a size limit for packages in the network. The segmented video data are compressed and encoded. Redundant frames can be removed during compression to minimise the file size while maintaining its quality. For example, if the first frame of the video displays only a black background, the black background need not be rendered again for any subsequent frames that have the same background. The encoding process converts the data into a new format. Common video encoding standards are

1. H.264/H.265
2. VP8/VP9
3. AV1
4. HEVC

2.2.2 Latency

Figure 2.4 graphically depicts the concept of latency. Latency is the time between a user's action and the response received for the action. There are usually three types of latency: server-side latency, network latency, and client-side latency. Network latency refers to delays occurring within a network, or on the internet. One of its main causes is the distance between client devices making requests and the servers responding to those requests. If a server is hosted at a data center in Asia, it will respond fairly quickly to requests from users in Asia, most likely within 10–15 ms. Users in Europe, on the other hand, will face longer delays. The server- and client-side latency occurs in local computing devices, usually because of the lack of computing resources, which may be avoidable.

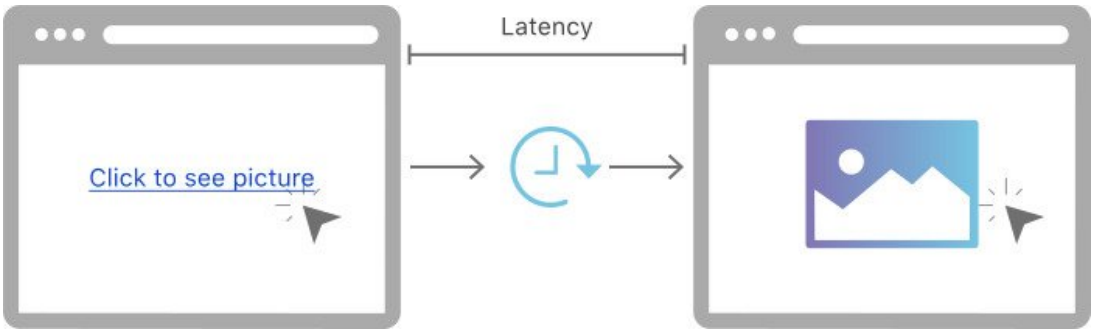


Figure 2.4: .
Graphic depiction of latency, which is the time interval between a user initiating an event and receiving a response.[4]

Chapter 3

Adopted Methodology

The data center in the cloud located in the core network usually comprises servers with high computing power and large storage capacity, while the majority of edge data centers consist of servers with limited computing resources. Existing cloud-based protocols were developed to distribute multiple applications on a single server in the cloud, while edge-based protocols were designed to distribute a single application to different servers in the edge because of resource constraints [6]. As mentioned in the previous chapter, a delay in the response time may lead to serious consequences, and therefore, resource control is an important aspect in edge computing. The objective of this study was to construct an artificial intelligence (AI) model to utilise the available resources giving to an application that is processing in the server. The AI model was intended to ensure that the available resources were used during live streaming to output the best quality. In other words, given an input video, the AI model generates a set of encoding parameters. While the encoding process can be slowed down to output the highest quality, it should be noted that in live streaming, the continuance of the video is the most important aspect. Therefore, it was necessary for the AI model to generate a set of encoding parameters that provided the highest quality while maintaining the continuance of the video with the available resources[1].

Remark. The objective of this study was as follows: Construct an AI model such that given a video and the available CPU power, it outputs a set of encoding

[1]In this study, only the CPU resources were considered

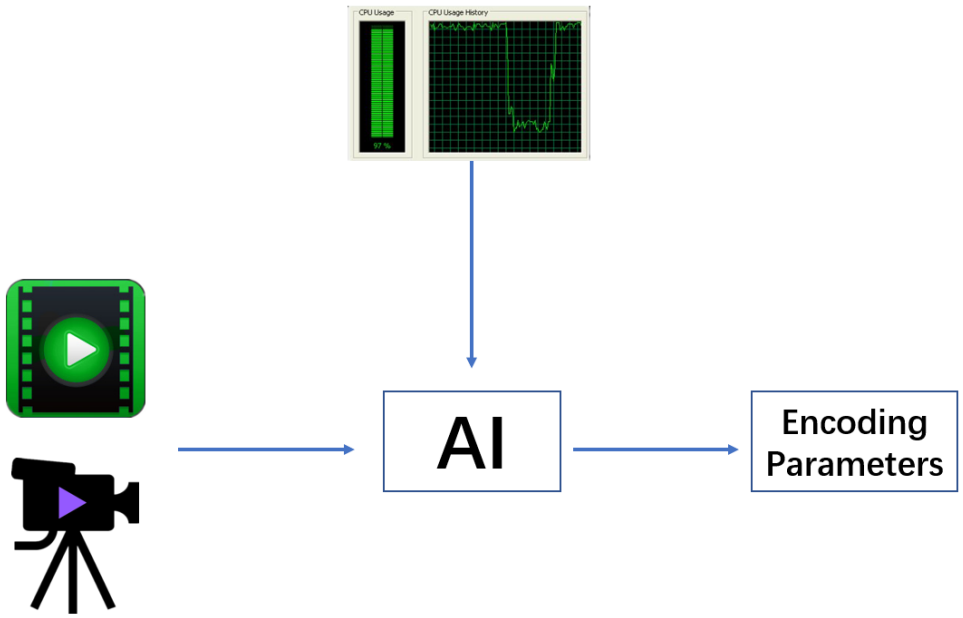


Figure 3.1: .
Basic principle of the AI model.

parameters that can generate the best quality video. Figure 3.1 shows the basic principle of the AI model.

This chapter analyses each sub-problem out of the target problem. Furthermore, the difficulties faced during the present work are discussed. The first task was to identify the algorithms available for constructing the AI model, and the next task was to define the 'best quality' for the encoded video.

3.1 Machine Learning Algorithm

If machine learning or AI is to be used to solve a problem, it is important to know to which type the problem belongs. In other words, the problem should be categorised. This will help us understand which tools or algorithm should be used to solve the problem. Usually, there are two main types of machine learning problems: supervised and unsupervised. In the former, predictions are made on the basis of a set of examples, while in the latter, the data do not have a set of defined labels, but AI helps us organise the data.

Remark. Supervised problems: All data are labelled. The algorithms learn from these labelled data and make predictions.

Remark. Unsupervised problems: All data are unlabeled. The algorithms learn to detect hidden structures, patterns, or relationships in the data.

Apparently, the present problem is a supervised learning problem. The label of the data used was the encoding parameter, and the input features were the attributes of the video and the available CPU resources. The second step was to choose a suitable machine learning algorithm.

3.1.1 Support Vector Machine

A support vector machine provides significant accuracy with less computation power, and therefore, it is highly preferred by many researchers. While it can be used for both regression and classification problems, it is usually used in classification problems. The objective of an SVM algorithm is to find a hyperplane or line in an N-dimensional space (N is the number of features) that can classify the data points (Figure 3.2) [7].

There are many possible hyperplanes that separate two classes of data points. The objective of the SVM is to find a plane that has the maximum margin for two separate classes of data points. Maximising the margin is desirable because points near the decision surface represent highly uncertain classification decisions. There is a high chance of the classifier deciding either way. A classifier with a large margin makes better classification decisions. In particular, it provides a classification safety margin. In other words, a slight error in measurement will not cause a misclassification, which indicates high accuracy. The points closest to the separating hyperplane are support vectors (Figure 3.3). The margin of the classifier can be maximised by using the support vectors. These points were considered for constructing the SVM.

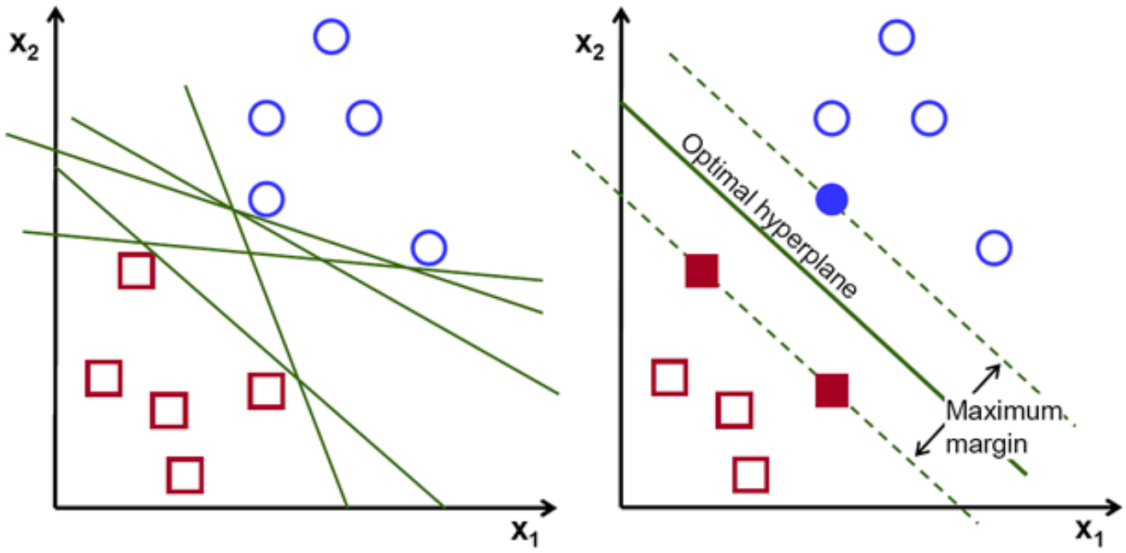


Figure 3.2: .
A SVM defines the criterion for identifying a decision surface that is maximally far away from any data point. The distance from the decision surface to the closest data point determines the margin of the classifier[7].

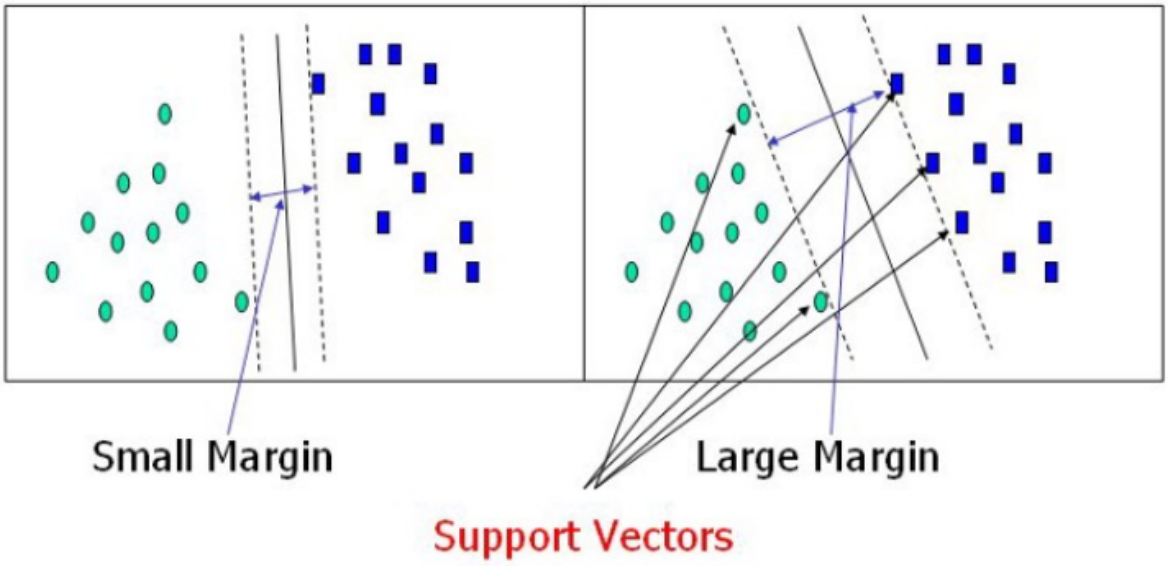


Figure 3.3: .
SVM with larger margin are more robustness[7].

Hyperplanes or lines are decision boundaries that classify different classes of data points. Data points belonging to different sides of the hyperplane are classified into different classes. The dimensions of the hyperplane depend on the number of features. If the number of input features is two, then the hyperplane is a line. If the number of input features is three, then the hyperplane becomes a two-dimensional plane. In the present case, it is difficult to image the hyperplane.

To conclude [8], the following observations were made.

Remark. The advantages of an SVM are as follows:

1. An SVM works better when there is a clear margin of separation between different classes.

2. It is more effective in high-dimensional spaces.

Remark. The disadvantages of an SVM are as follows:

1. An SVM algorithm is not suitable for large data sets.

2. An SVM does not perform well when the target classes overlap.

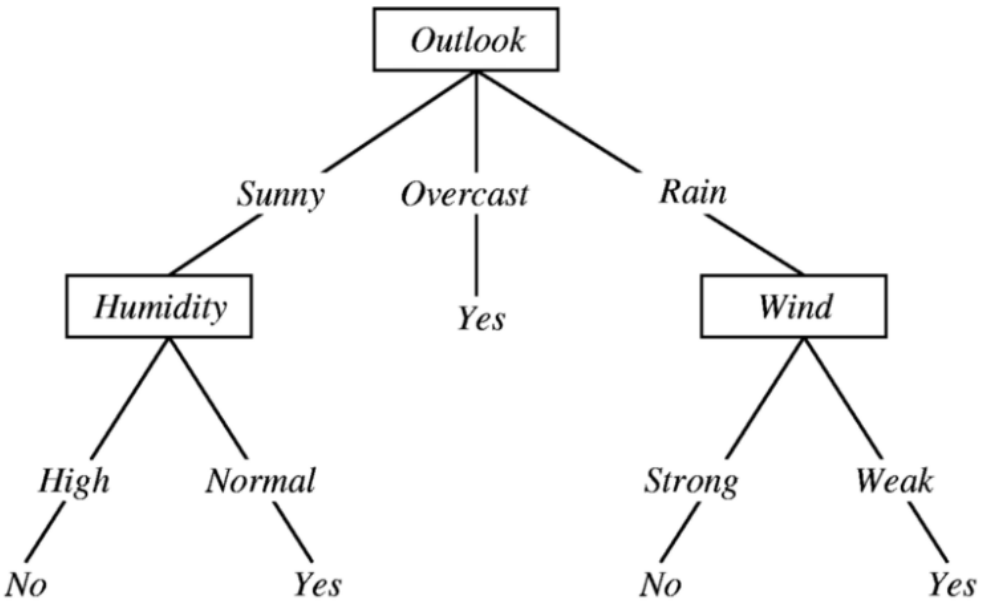


Figure 3.4: .
We start from the root of the decision tree to predict labels . We compare the value of each node's attribute with that of the record's attribute. On the basis of the comparison, we follow the branch corresponding to that value and move to the second node until we reach the leaf.[9]

3. An SVM works by identifying data points above and below the classifying hyperplane, and there is no probabilistic explanation for the classification.

3.1.2 Decision Tree

The decision tree algorithm is a supervised learning algorithm. It can be used for solving both regression and classification problems. A decision tree is a flow-chart-like structure, and each node represents the comparison of a feature, such as with hair or without hair. Each leaf node represents a class label, such as the type of animal, and the branches represent conjunctions of features and lead to those class labels. The paths from the root to the leaf represent the classification rules [9].

There are two types of trees: classification trees and regression trees. A classification tree has a discrete target variable, while a regression tree has a continuous target variable. Decision trees adopt a top-down approach to data. They try to group and label observations that are similar, and seek the best rules that split the

observations that are different from each other until they reach a certain degree of similarity. The splitting can be binary (into two nodes) or multiway (into more than two nodes).

To conclude [10], the following observations were made.

Remark. The advantages of decision trees are as follows:

1. Decision trees require less effort for data preparation during preprocessing. Normalisation and scaling of data are not required.

2. Missing values in the data do NOT affect the process of building a decision tree

Remark. The disadvantages of decision trees are as follows:

1. Small changes in the data may cause a large change in the structure of the tree.

2. The calculation sometimes can be more complex compared with other algorithms

3.1.3 Random Forest

Decision trees are simple but a wrong choice of the next node can lead to a completely different result. There is a high risk of overfitting, which implies that the deeper the tree and the more specific it is to the data set, the higher is the risk of overfitting. Random forests are constructed by using many decision trees, and there is no correlation between different decision trees. Therefore, the use of random forests reduces the risk of overfitting and offers better accuracy. Each decision tree in a random forest provides a prediction, and the class with the most votes is the final prediction (Figure 3.5). A random forest performs better than a single decision tree because even if some trees are wrong, many other trees will be correct [11]. Therefore, the decision trees that work as a group provide the correct result (Figure 3.5).

To conclude [13], the following observations were made.

Remark. The advantages of a random forest are as follows:

1. It is not easy to overfit, and therefore, a random forest was helpful in the present study. However, insufficient data may easily lead to overfitting.

2. It has methods for balancing errors in data sets where classes are imbalanced. This was helpful in the present study, where there were nearly 150 classes in total and it is imbalanced.

3. A random forest involves sampling of the input data with replacement, called bootstrap sampling.

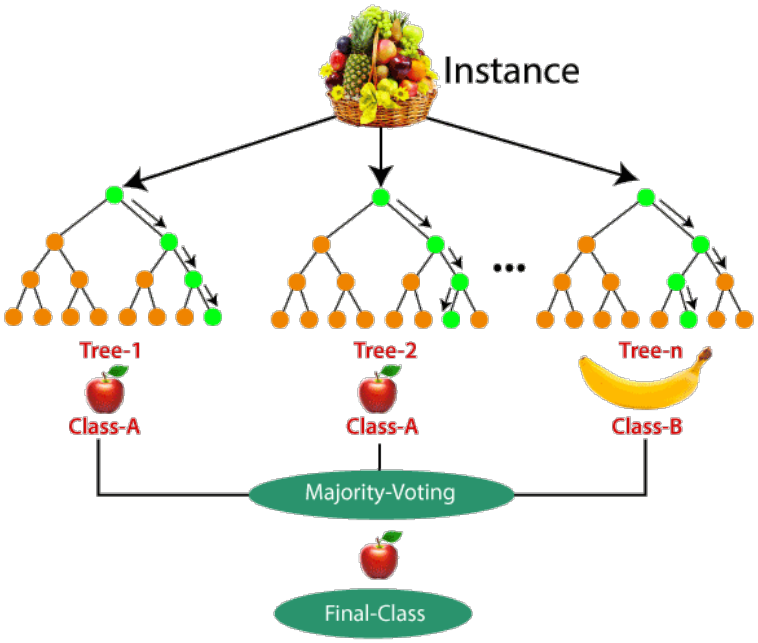


Figure 3.5: .
The class with the highest vote is the final class.[12]

Remark. The disadvantages of a random forest are as follows:

1. The use of a random forest is like a black box approach as we have very little control on what the model does. The user can only use different parameters and random seeds to see what will happen.

In this work, a random forest was used as the machine learning algorithm.

3.2 Video quality assessment

Video/image quality assessment refers to objective and subjective ways of measuring the difference between two images or videos with the same main content [14]. The evaluation includes the fidelity and intelligibility of the video or image. The former refers to the degree of deviation of the evaluated image from a standard image, and the latter refers to the ability of the image to provide information to humans or computers. Evaluation methods can be divided into subjective evaluation methods (3.6) and objective evaluation methods. The subjective score is

Figure 3.6: .
An observer is comparing the original video with an encoded video.[16]

generally expressed by the mean opinion score (MOS) or the difference mean opinion score (DMOS). The MOS is used to determine the image quality by normalising the observer's score,and the DMOS is to evaluate the image quality by normalising the difference between the observer's evaluation scores of the original image and distorted image. The surjective way was not feasible for the present study since it was difficult to ask someone to watch the videos and provide a score. Therefore, the objective way was used in this work.

In the objective evaluation method, a machine determined the quality index of the video/image through an algorithm, instead of a human. Although the objective evaluation method enabled the computer to predict the score of a specific video from the view of a human to a considerable extent, there was still a big gap in the degree of conformity between different objective evaluation methods and subjective feelings. The quality of an evaluation method can be measured from the accuracy, consistency, stability, and monotonicity of its prediction. The accuracy is the similarity between subjective and objective evaluations, consistency means that the method should perform well for all types of videos/images, stability means that the values obtained in different evaluations of the same video/image should be equal or

the difference should be small, and monotonicity means that the evaluation score should follow the MOS if the MOS it increases or decreases.

Furthermore, on the basis of their dependence on the reference video/image, evaluation methods can be divided into three types: full reference, reduced reference, and no reference methods. The full reference method requires a distortion-free original video. The method compares the video with the encoded video to obtain the evaluation result of the distorted video. The full reference method was adopted in the present study.

Below, a few famous methods are described and the selection of the best one from them is discussed.

3.2.1 Mean square error, MSE

The original reference frame and the distorted frame are directly summed by the square of the difference. There are similar indicators such as absolute error, root mean square error, and standard deviation.

$$MSE = \frac{1}{H * W} \sum_{i=1}^{W} \sum_{j=1}^{H} X(i,j) - Y(i,j)^2$$

Advantage: Low computational complexity

Disadvantage: There is a big gap between subjective evaluations

3.2.2 Peak signal noise ratio, PSNR

The logarithm of the ratio of the energy of the peak signal to the average energy of noise is usually expressed in decibels. Since the mean square error (MSE) is the energy mean of the difference between the real image and the noisy image, and the difference between the two is the noise, the peak signal noise ratio (PSNR) is the ratio of the peak value signal energy to the MSE. The PSNR is the most common and widely used objective measurement method for evaluating picture or video quality. Although it is not completely consistent with the visual quality perceived by the human eye, it is still used as a baseline for comparison with other indicators.

$$PNSR = \log_{10} \frac{MaxValue^2}{MSE}$$

Advantage: Low computational complexity

Disadvantage: There is a gap between subjective evaluation and the colour change, which results in a lower score

3.2.3 Structure similarity Index, SSIM

The human eye's perception sensitivity has three major characteristics:

1. It is more sensitive to contrast differences with lower spatial frequencies. This characteristic was helpful in this study, since insufficient data may easily lead to overfitting

2. The human eye is more sensitive to brightness contrast differences than chromaticity.

3. The human eye's perception of an area is affected by the surrounding neighbouring areas.

Therefore, the colour space should be converted to the HVS space for evaluation. The structure similarity (SSIM) index evaluates the similarity between two images from their brightness, contrast and structure. A block diagram is shown in Figure 3.7.

The brightness is represented by the mean value, the contrast is represented by the variance normalised by the mean value, and the structure is represented by the correlation coefficient, which is the ratio of the product of covariance and variance in a statistical sense. The SSIM index can be used locally to resist sudden changes in distortion. In fact, the SSIM index of various partial windows is averaged, and the statistical value of each partial window is weighted with a Gaussian weighting function to prevent blockiness.

First determine the mean, variance, and covariance, and then obtain the brightness factor, contrast factor, and structural similarity factor, and finally multiply the three factors together.

The formula is as follows:

$$u_X = \frac{x}{H * W} \sum_{i=1}^{H} \sum_{j=1}^{W} X(i,j)$$

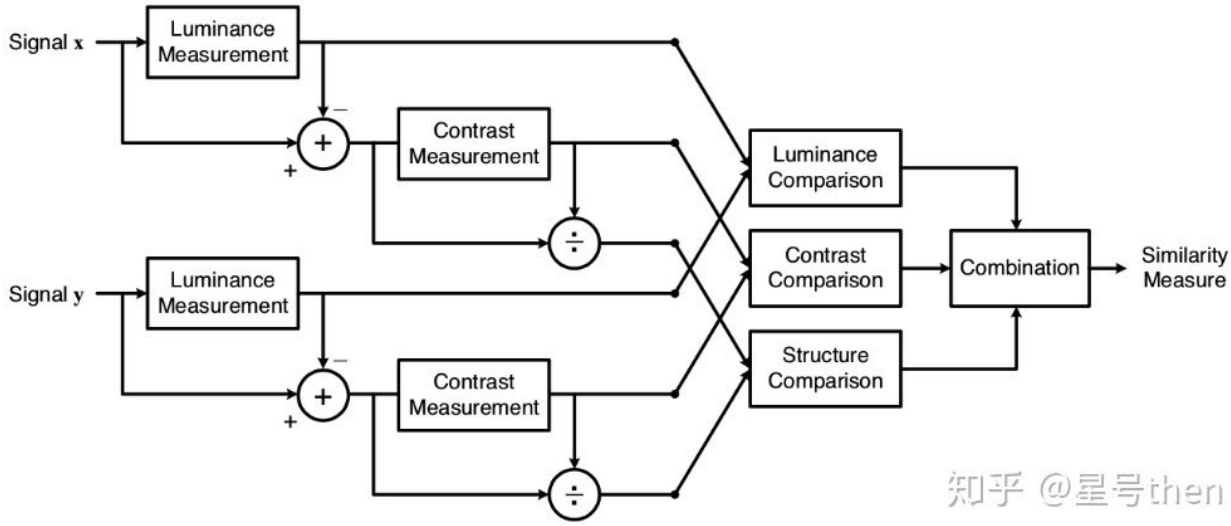


Figure 3.7: .
The architecture of SSIM.[14]

$$\sigma_X = (\frac{1}{H*W-1}\sum_{i=1}^{W}\sum_{j=1}^{H}X(i,j) - {u_X}^2)^{\frac{1}{2}}$$

$$\sigma_{XY} = \frac{1}{H*W-1}\sum_{i=1}^{W}\sum_{j=1}^{H}(X(i,j) - u_X)(Y(i,j) - u_Y)$$

$$l(X,Y) = \frac{2u_Xu_Y + C_1}{u_X^2 + u_Y^2 + C_1}$$

$$c(X,Y) = \frac{2\sigma_X\sigma_Y + C_2}{\sigma_X^2 + \sigma_Y^2 + C_2}$$

$$s(X,Y) = \frac{\sigma_{XY} + C_3}{\sigma_X\sigma_Y + C_3}$$

$$SSIM(X,Y) = l(X,Y) * c(X,Y) * s(X,Y)$$

Advantage: Structural information can be used to evaluate distortion closer to the human eye.

Disadvantage: The human visual system is highly non-linear. Comparing the similarity of the two signal structures is not sufficient. There is still considerable scope to explore.

3.2.4 VMAF

This indicator is a set of objective evaluation indicators developed by Netflix to overcome the incapability of traditional indicators to reflect video scenes with multiple

scenes and multiple characteristics [15]. This indicator was first disclosed on Netflix's technical blog, and it is based on machine learning methods (mainly an SVM) for classification prediction. The results of basic indicators with different weights are merged, the machine learning model is trained and tested using supervised learning, and finally an objective evaluation model is generated for multiple source content features, underlying features, and distortion types. This indicator is currently the most famous objective video evaluation indicator for online videos. It is suitable for measuring the perception of streaming video quality in different environments, and it is open source. It has a variety of source content features, including animation, camera panning, indoor/outdoor, face zooming, people, water surface, prominent objects and multiple objects. Multiple underlying feature types include different brightness levels, contrast, material, activity, colour changes, colour richness and sharpness. Distortion types include blocking, ringing, film-grain noise and mosquito noise.

The basic indicators integrated include visual information fidelity, detail loss metric and time domain motion index/mean co-located pixel difference. Compared with the current mainstream PSNR-HVS and VQM-VFD indicators, it is more clustered from the scatter chart. From the three indicators SRCC, PCC and RMSE that measure objective evaluation indicators, VMAF achieved higher scores (Figure3.8).

-	SRCC	PCC	RMSE
PSNR	0.416	0.394	16.934
SSIM	0.658	0.618	12.340
FastSSIM	0.566	0.561	13.691
PSNR-HVS	0.589	0.595	13.213
VQM-VFD	0.763	0.767	9.897
VMAF 0.3.1	0.690	0.655	12.180

-	SRCC	PCC	RMSE
PSNR	0.772	0.759	0.738
SSIM	0.856	0.834	0.621
FastSSIM	0.910	0.922	0.415
PSNR-HVS	0.858	0.850	0.580
VQM-VFD	0.925	0.924	0.420
VMAF 0.3.1	0.929	0.939	0.372

-	SRCC	PCC	RMSE
PSNR	0.632	0.643	0.850
SSIM	0.664	0.682	0.831
FastSSIM	0.747	0.745	0.718
PSNR-HVS	0.703	0.726	0.722
VQM-VFD	0.770	0.795	0.639
VMAF 0.3.1	0.872	0.905	0.401

Figure 3.8: .
Generalisation ability of VMAF is strong, and its score was nearly the highest for the three test data sets.[17]

Advantage: Combining multiple objective indicators based on machine learning methods can facilitate the merging of different good objective indicators into one's own system. It can also help improve accuracy through continuous training with

continuous expansion of data sets (subjective quality scores are required). Furthermore, various machine learning methods can be replaced (e.g., an SVM can be replaced with random forest regression factors) to reduce regression errors.

In this study, VMAF was chosen as the tool to define video quality.

Chapter 4

Experimental Tools

This chapter describes the tools used in this study and the reason for their use As mentioned in Chapter 2, the experiment environment will built by using the tools discussed here. They are Docker, FFmpeg and Google Cloud Platform. Docker can create a pre-warmed container which holds the FFmpeg application to run in the edge data center. To avoid all possible confliction due to other running programs, all the experiments and tests will be performed in the virtual machine on the Google Cloud Platform.

4.1 Docker

Docker is an open-source and container-based technology for developing, deploying, and running applications. It can be used to create a container that includes the application and its dependent libraries, and any other tools the application requires. Isolating applications from their underlying infrastructure makes delivery faster.

'A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Container images become containers at runtime and in the case of Docker containers - images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerised software will always

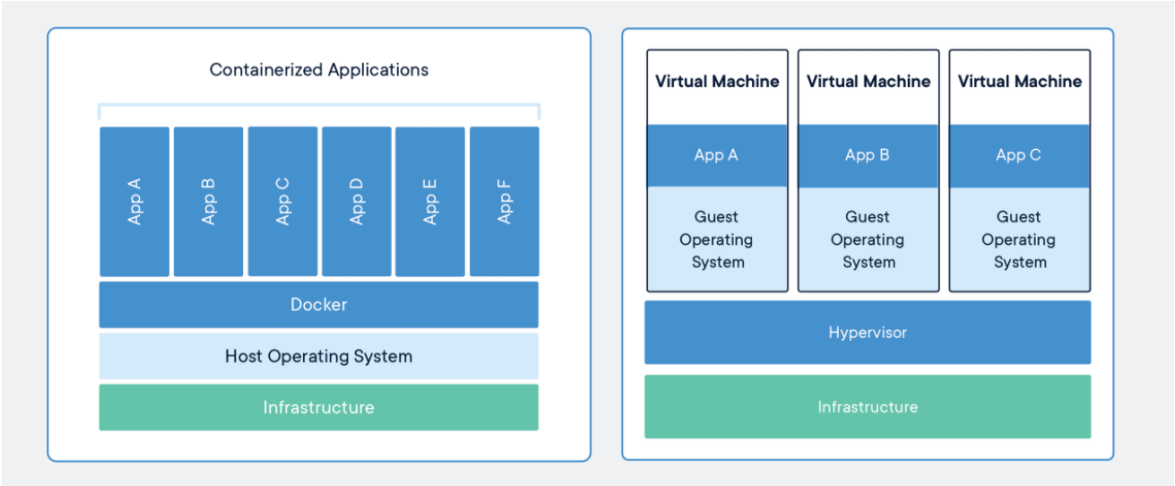


Figure 4.1: .
The virtual machine has a hypervisor layer while Docker has a Docker engine layer. [18]

run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging'[18]

As the Docker official page states, a docker container image is an all-in-one lightweight package that contains everything to run an application. Furthermore, it allows applications to be isolated from each other, for ensuring that they are working correctly. The container appears to be similar to virtual machines. Another reason why Docker is famous is that it is superior to virtual machines. The difference between them is shown in Figure 4.1.

The memory usage is high in a virtual machine, but low in the docker environment. In addition, when multiple virtual machines are run on a server, their performance is poor, while in Docker, their performance is maintained steady. A virtual machine is not designed for portability while Docker is. A sudden change in a host machine may lead to failure of some software in the virtual machine because some dependencies may not be inherited correctly. By contrast, Docker is guaranteed to work as it has been designed.[19]

The reasons for Docker suiting the requirements of this study are as follows:

1. The boot-up time of a virtual machine is longer than that of Docker, which boots up almost instantaneously. In the 5G world, the server usually maintains

a pool of prewarm containers. Once they are required for an action, they become warm containers in milliseconds

2. We cannot reallocate any computing resources in a virtual machine. In Docker, if we have unused memory, we can reallocate and reuse it across other containers. This suited the present study because the study involved an edge server, and the server was required to allocate computing resources dynamically according to the situation (Figure 4.2).

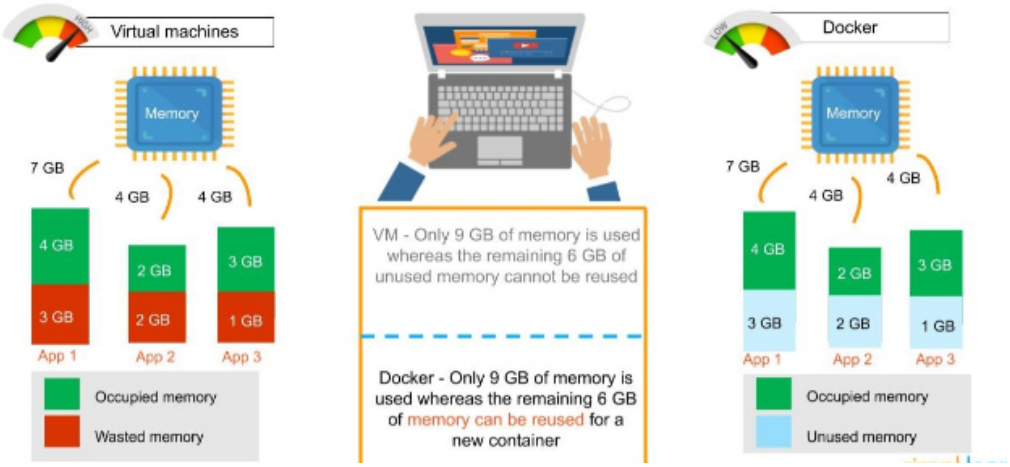


Figure 4.2: .
To conclude, Docker is better than virtual machines in terms of memory usage, performance, portability and boot-up time.[19]

4.2 FFmpeg

FFmpeg is a powerful open-source tool that can encode or decode almost all types of videos. It supports many video codecs such as H264, H265, VP8 and VP9. It also supports multiple file formats (e.g., mp4, flv, webm and mkv). Moreover, it can be used as a standalone tool in custom software. Below some basic functionalities of FFmpeg are discussed. FFmpeg basically contains three tools: FFmpeg tool, FFplay tool and FFprobe tool.

1. FFmpeg tool is a command-line tool for encoding and decoding a video from one file format to another with a chosen codec. This tool was used to perform all the video processing processes in this study.

2. FFplay is a very simple media player using the FFmpeg libraries. In this study, it was used to test live video streaming.

3. FFprobe tool is used for analysing video streams. In this study, it was used to obtain the metadata of the input video.

The basic workflow of FFmpeg can be divided as follows (Figure 4.3):

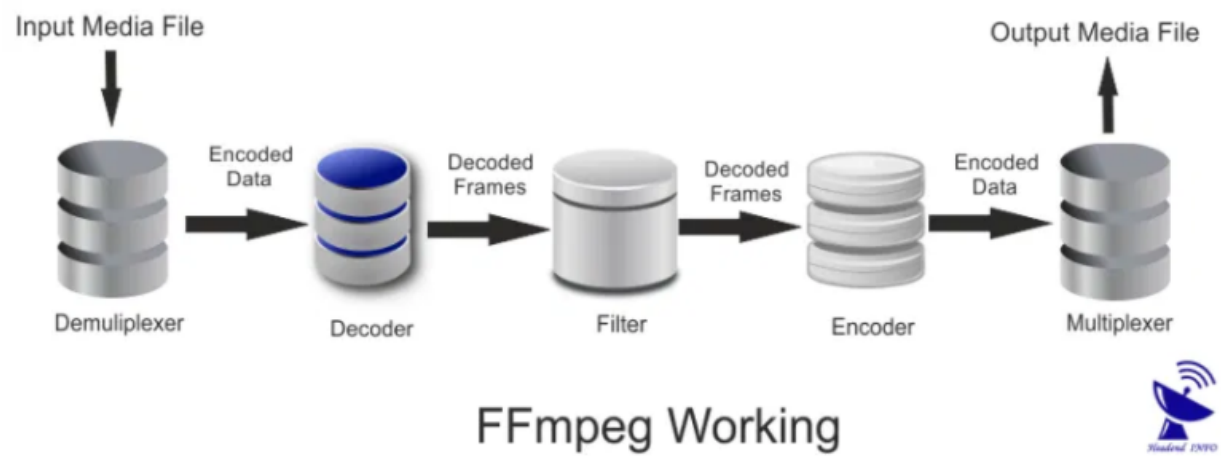


Figure 4.3: .
In the demultiplexer phase, the libavcidec library was used to process the input file, demultiplex the input file and obtain packets containing encoded data from them. All encoded data packets pass through the decoder, which decodes the data and generates decoded data frames that are uncompressed. Subsequently, the filtration of the uncompressed data is performed by the libavfilter library.The data pass through a chain of filters to produce a graph. Finally, the data are encoded and multiplexed to provide the output file.[20]

4.2.1 FFserver

Another advantage of FFmpeg is that it can be used for broadcasting a live stream video. It can stream in two ways: to a server that re-streams the content for to one or more clients or directly to a single receiver. FFserver can pick up the stream from FFmpeg and restream it to the client. The internal system of FFserver works as shown in Figure 4.4.

External applications, such as FFmpeg, send audio or video streams to FFserver and bind themselves with one or more feeds if these feeds are free. If the input video source produces only one stream, the video source is useless if it connects to two or more feeds. The feed element will link to one or more streams if one input source is to be streamed by using several different output formats. The stream element is a connection point for a client wishing to receive a specific stream, and it can handle multiple client connections simultaneously.

If streaming of a local file is attempted in FFserver, it is fast. However, if the input file is obtained from FFmpeg, the streaming will have a lag because of the

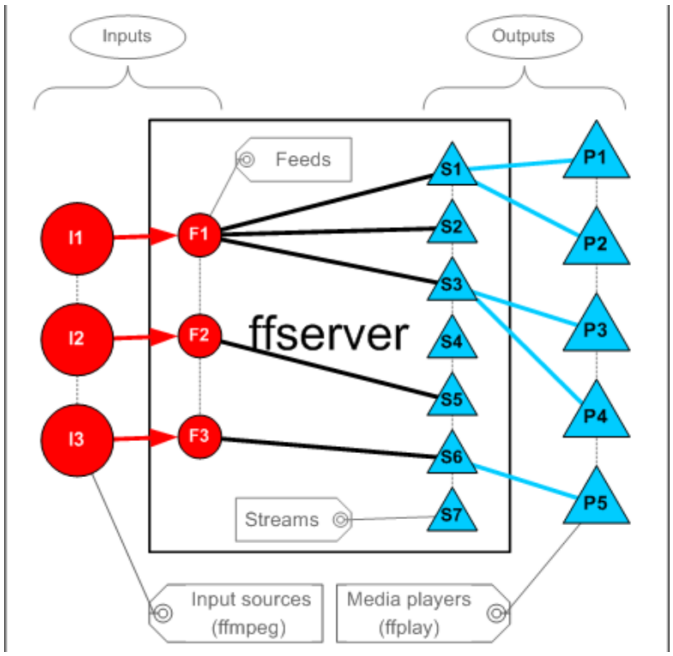


Figure 4.4: .
There are four main elements that contribute to FFserver.

encoding process in FFmpeg. The encoding process in FFmpeg or FFserver that causes the lag is discussed in detail in Chapter 5.

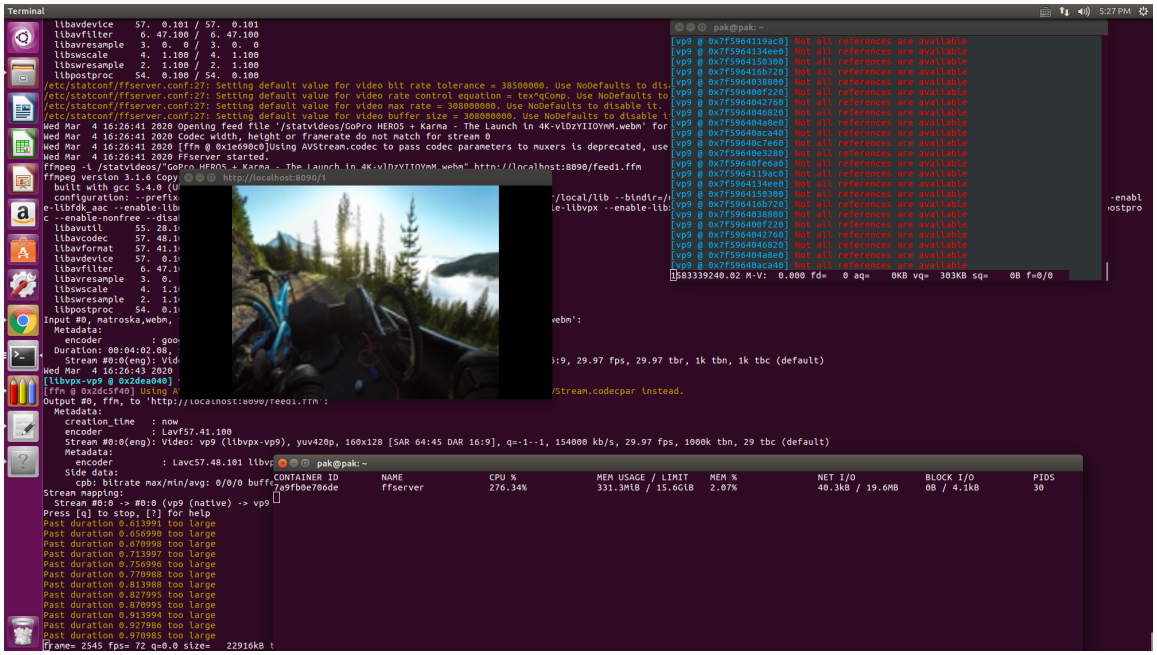


Figure 4.5: .
Insufficient computing resources slow down the encoding process, which interrupts the streaming.

4.3 Google Cloud Platform

To avoid the influence of CPU conflict because of another running program, I decided to run the generation on the Google Cloud Platform. I created 10 virtual machine instances, each with 8–32 CPUs and 32 GB memory. In each VM instance, only the encoding process that was to be analysed was in progress. In this way, I could ensure that there was no CPU or memory conflict.

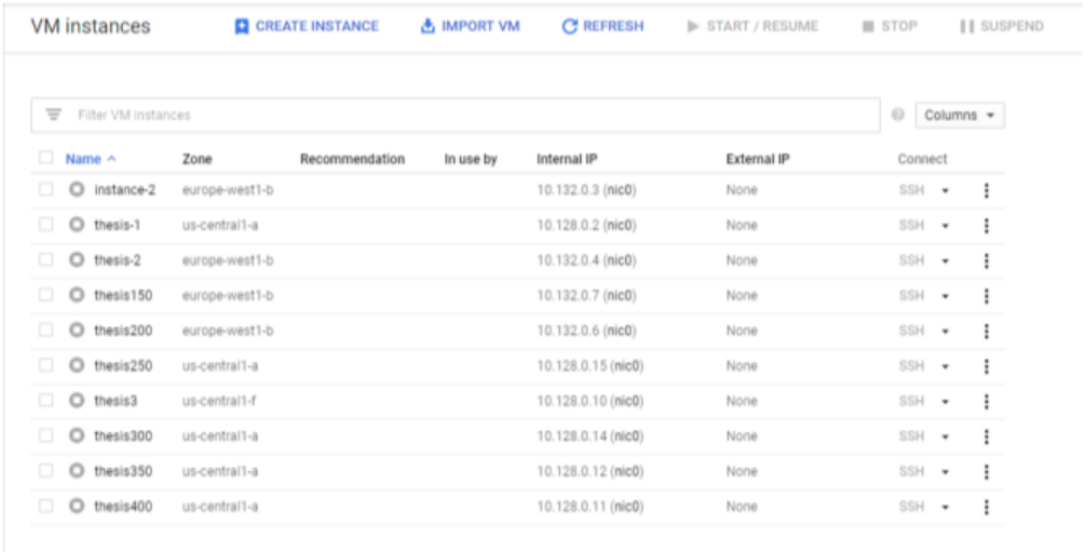


Figure 4.6: .
I have created 10 VMs for generating the data set

Chapter 5

Data Collection and Analysis

The first step is the data set generation. To be able to create an AI, we need to provide it the labeled data set for training. In other words, we are teaching the Ai what should the output look like when we have different input data. To repeat the aim of this thesis work, we are creating an AI that will give us the best encoding parameter for encoding to get the best quality streaming according to the computing resource(CPU). As mentioned in the previous chapter, there is a lot of video quality metric for video assessment. However, those metrics are designed for offline video, they measure the difference (e.g distortion) only between two videos. The quality of live streaming should not consider the difference between frames in two videos only, they consider also the continuity of the streaming when the client is watching.

How do we define quality for live streaming?

There is no off-the-shelf application to measure the quality of a live steaming directly and those video assessment methods described in the previous chapter can not be used because the lag or discontinuance events during streaming will cause an unexpectable error or result.

In this situation, the problem is divided into smaller sub-problems until these

sub-problems become simple enough to be solved. Measuring the quality of streaming can be divided into

1. The continuity of the streaming. This is the most important aspect in Live streaming, the client is willing to sacrifice sharpness to trade for full continuity during the streaming.

2. The sharpness of the video in streaming. In this way, the encoded video can compare with the original video by using the video assessment methods mentioned in the previous chapter .

In some famous live streaming platforms, such as Youtube and Twitch. They will always limit the output format of the live streaming that available to the client , also the input source you send to the server. For example, the resolution class the client can choose on Youtube is usually 1080P,720P,480P and 360P. To simplify the problem, classification instead of regressionis is chosen to be the problem type in this study . In other words, the combinations of the output encoding parameters are limited, a detailed discussion will be in the following subsections.

5.1 Data Structure

Two types of data were collected , they are the encoding parameter and the measure of the encoding process. These data are used for data analysis and training(some unnecessary features will be removed).

1. Parameters:

   (a) *CPU limit*: These are the available CPU resources allocated for the container. The output encoding parameter given by the AI should change according to this value. Bear in mind that, available CPU limit in perceentage is machine-dependent which means that the AI train in one computer should not be applied to another computer with different hardware. We can try to collect the total number of instructions which is machine-independent, but also measuring the number of instructions generated by a given task is problematic since this kind of profiling adds overhead to resource consumption. Basically you would have to run the experiment

twice: the first time to collect the performance metrics (that profiling would affect), the second time to collect the number of instructions generated using profiling. The other problem is that we cannot set a limit in terms of instructions execution speed, the limit we can set is in terms of CPU time. To solve the portability problem, in order to avoid the need to recreate the dataset in the machine in which you want to deploy the AI model, one solution could be to derive a multiplicator that expresses CPU performance in relation to the CPU performance of your machine. In this way, one single test is enough to translate the entire dataset to the new machine.The multiplicator may be derived in this way: select a combination of encoding parameters from your dataset, with a certain CPU limit. Run the same test on the new machine, adjusting the applied CPU limit until you can run the test achieving the same performance (execution time, encoder speed, it's the same). NewCpuLimit / OldCpuLimit is the multiplier that can be applied to the entire dataset to port it to the new machine.

(b) *Input codec*: All source video is pristine. There is no input codec.

(c) *Output codec* : H264 codec is used in this study. It is old but tested and optimized. VP9 or H265 are new and more efficient. However, it is not easy to collect data by a script (Linux) if use VP9/H265 because their output format is not stable.

(d) *Input resolution* : To simplify the problem, only four resolution classes were considered. They are 4K(3180x2160), 1080P(1980x1080), 720P(1280x720), 480P(854x480).

(e) *Output resolution* : Same with the input resolution.only four resolution classes were considered. They are 4K,1080P,480P,360P. The output resolution will change according to the CPU limit in order to give the best quality for encoded video.

(f) *Input framerate* : The input FPS of the video. The average encoding speed or FPS was compared to the Input framerate, the combination of encoding parameter that with the highest sharpness and its average encoding speed is higher or equal to the input framerate will be selected.

(g) *Output framerate* :The input FPS of the video. FPS was kept the same

value is because lower the output FPS will not actually help to reduce the CPU usage while it decreases the sharpness of a video significantly.

(h) *Input bitrate*

(i) *Target bitrate* : CRF value was used instead of bitrate. CRF value is from 0-51,0 is the best. The range is exponential, so increasing the CRF value +6 results in roughly half the bit-rate / file size, while -6 leads to roughly twice the bitrate. In this case, we don't need to consider the different bit-rate intervals for different levels of video (high bit-rate vs low bit-rate). The AI can output the CRF value to represent the bit-rate to use in encoding[21].

(j) *Encoder preset* : Basically whatever setting is used to adjust the quality/encoding-speed. A preset is a set of options that will provide a certain encoding speed to the compression ratio. For example, a slower preset will provide better compression (compression is quality per filesize). This means that, if you target a certain file size or bit rate, slower preset gives a better quality.

2. Measures:

(a) *CPU consumption* : This value is collected for data analysis.

(b) *Memory usage* : This value is collected for data analysis.

(c) *Encoding FPS/encoding speed* : The encoding FPS collected was the average encoding FPS. The encoding FPS during the encoding process is not stable, it is fast at the start and decreasing until the end. Therefore, the average value was chosen, if the average encoding FPS is larger or equal than the input FPS, it means that there is no latency created in the encoding process. Please note that, if the available CPU power is not enough to encode any video that the average encoding FPS is larger than the input FPS, This study will select the highest average encoding FPS available.

(d) *Output bitrate* : The output bitrate usually is different from the target bitrate. This value is collected for data analysis.

(e) *VMAF score*: This score represents the sharpness of encoded video when compared to its original video.The highest score will be selected.

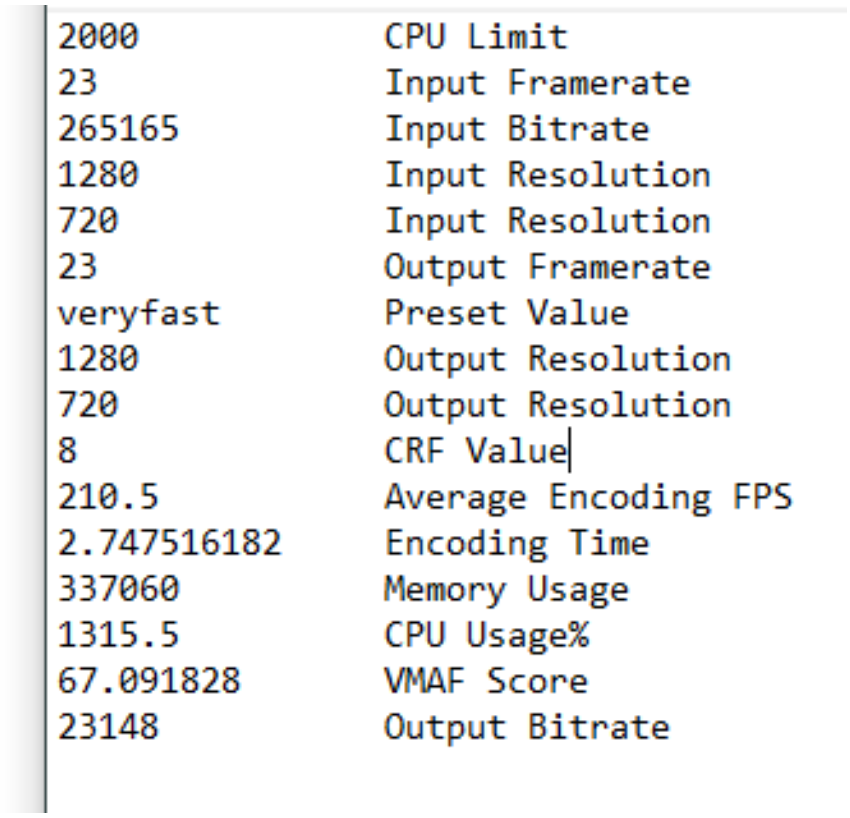


Figure 5.1: .
An example of the data set

5.2 Data Collection and pre-processing

The source videos are taken from User Generated Content (UGC) Dataset. There is around 1500 pristine video(.mkv)clips with a duration of 20 seconds each. Their format is all the same. Most of the videos in the same resolution in that dataset are with the same bitrate, framerate and file-size. Therefore, it is useless if we take into account. After filtering these videos, only around 50 videos left.In order to speed up the data generation process. Some unnecessary range in parameters was removed, these parameters will never be selected.

1. *CPU limit* : From 100%-500% , the step is 50%: From 500%- 1500% . the step is 100%; From 1600% - 3200% , the step is 400%.

2. *CRF value* : For the CRF value , this study considered only from 8-32 except for 4K. For 4K , it considered from 24-32 to avoid huge bitrate. Usually, the

VMAF score of 720P with CRF value 8 is better than 1080P with a CRF value 32.

3. *Preset value* :For preset value from slow to veryfast except for 4K. For 4K, the range from 'medium' to 'veryfast' was considered.

4. To be able to adjust the range of the parameter , a Linux script per resolution class per CPU limit was chosen to generate the data.

```
1. Assign different resolution(width) value to $width
2  Assign different resolution(hight) value to $height
3. Assign different preset value to $preset
4. Assign differrent CRF value to $CRF
5. Assign 0 to $total_number_record
6. For each video (0.mkv-?.mkv) in a resolution class
    7.Save the Input framrate
    8.For each resoltion class $width + $height
        9.For each $CRF
            10.For each preset value $preset
                11. Save the available CPU power
                12. Save the Input resolution
                13. Save the Output frameRate
                14. Save the Output resolution
                15. Save the CRF value
                16. Save the average encoding FPS
                17. Save the encoding time
                18. Save the memory usage
                19. Save the CPU usage
                20. Save the VMAF score
                21. Save the Output bitrate

            end
        end
    end
end
```

Figure 5.2: .
The pseudocode of the linux script for data generation.

5.2.1 Data pre-processing

In order to apply the data to the machine learning algorithm, data cleaning was needed.

1. Removed all unnecessary features. Keep only the CPU limit, Input FPS, Input Bitrate, Input Resolution, Preset Value , Output Resolution and CRF.

2. Merged the width and height both for Input Resolution and Output Resolution was needed, after that, the resolution was changed to a numeric value. For example, 0 = 480P , 1=720P, 2=1080P and 3=4K.

3. The Preset value was changed to a numeric value.For example ,Superfast = 0 , veryfast=1 ,fast =2 , medium =3 , slow =4 and slower =5

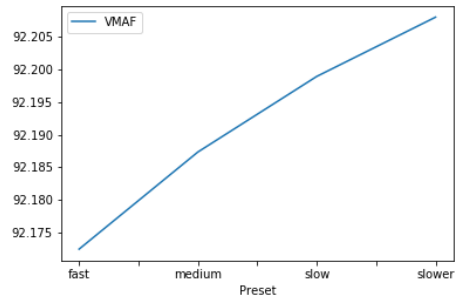
	A	B	C	D	E	F	G
1	CPU_Limit	In_FrameRate	In-Bitrate	In_Resoultion	Presest	Out_Resolution	CRF
2	1000	50	2488341	2	1	2	12
3	1000	23	1193210	2	3	2	8
4	1000	50	1244181	2	1	2	8
5	1000	30	746509	2	1	2	8
6	1000	12	298603	2	4	2	8
7	1000	59	1491526	2	1	2	12
8	1000	30	746509	2	2	2	8
9	1000	60	1493017	2	2	2	12
10	1000	29	745763	2	2	2	8

Figure 5.3: .

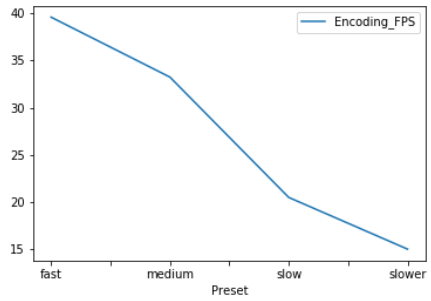
The 4 columns on the left are the features and the 3 columns on the right are the labels. In total, we have 216 different labels, but actually only 75 are active. It is because for some labels we will never select it. For example, encoding a 4K video, we will never choose the output resolution 4K, CFU value 48, preset medium. It is because the output resolution 1080P with CFU value 12, preset value 'medium' is always having higher VMAF score than it. In case both of them have Encoding FPS higher than Input FPS.

5.3 Data Analysis

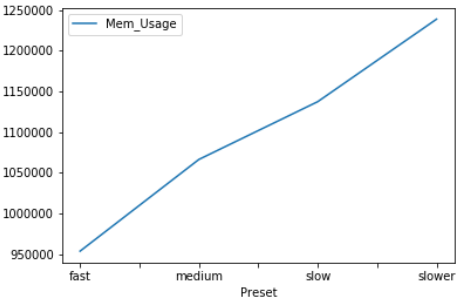
In this section, the correlation between different encoding parameters and their impact on the video will be analyzed. The following three parameters will be considered: preset , CRF and output resolution. Such understanding will avail the design of the ML model.



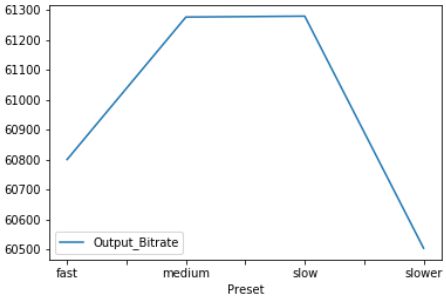
(a) Preset - Vmaf



(b) Preset - Encoding FPS



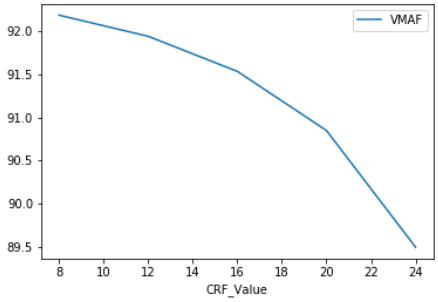
(c) Preset - Mem Usage



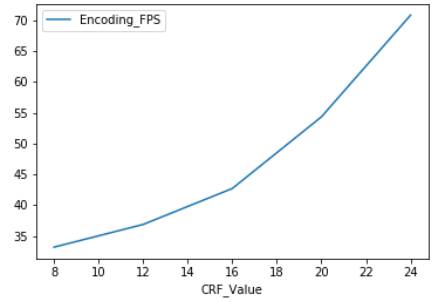
(d) Preset - Output bitrate

Figure 5.4: Preset Value

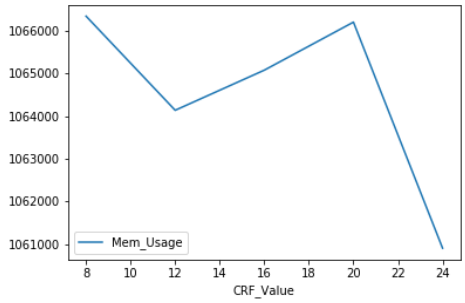
The variation in preset value induces only a neglectable impact on VMAF score. Increase the preset value can increase encoding speed and reduce memory usage significantly. If the network condition is ignored, the preset value can always set to'veryfast' in order to speed up the encoding process. This parameter should be adjusted prior to other parameters cause this will not affect the VMAF score.



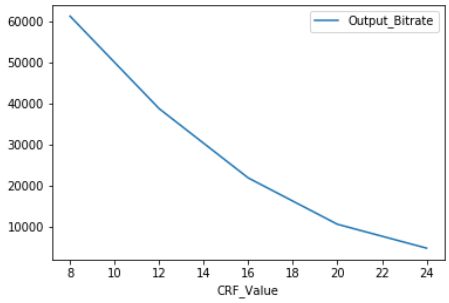
(a) Crf - Vmaf



(b) Crf - Encoding Fps



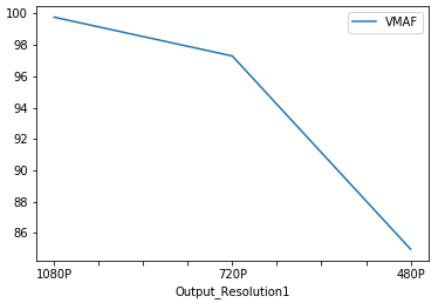
(c) Crf -Mem Usage



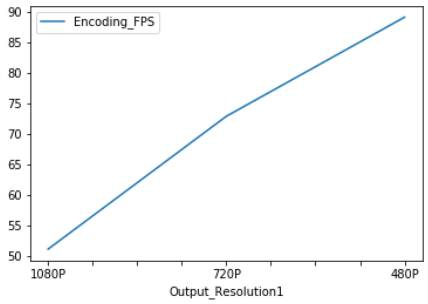
(d) Crf - Output bitrate

Figure 5.5: CRF value

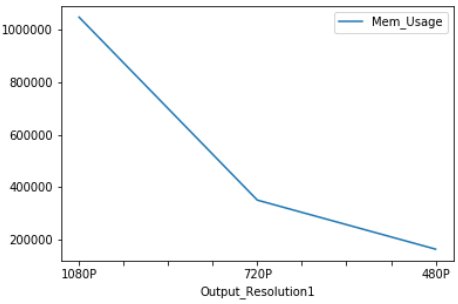
The variation in CRF value induces a notable impact on VMAF score and encoding speed. Increase the CRF value lower the bitrate and VMAF score. Borne in mind that CRF value should work with output resolution to obtain the best result. For instance , the VMAF score and encoding speed for output resolution 1080P with CRF 38 is lower than the 720P with CRF 16.



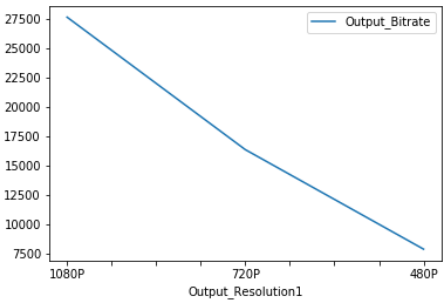
(a) Output Resolution - Vmaf



(b) Output Resolution - Encoding Fps



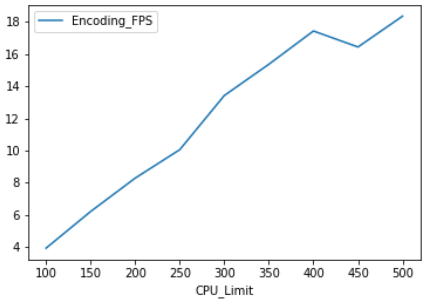
(c) Output Resolution - Mem Usage



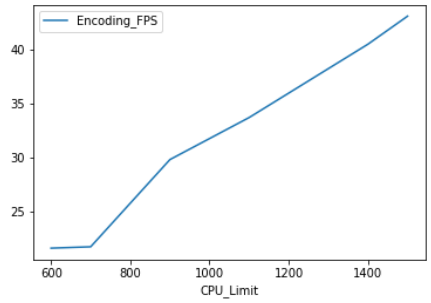
(d) Output Resolution - Output bitrate

Figure 5.6: Output Resolution

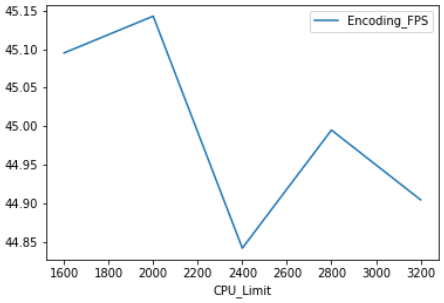
The impact on VAMF and encoding speed for variation in output resolution are similar to CRF value. The only difference is lower the output resolution can decrease the memory usage.



(a) CPU - Encoding FPS 1



(b) CPU - Encoding FPS 2



(c) CPU - Encoding FPS 3

Figure 5.7: CPU limit
The encoding speed remains steady after the available CPU resource gets over 1600% . Thanks to this, unnecessary data collection may be avoided.

Chapter 6

Experiment and Results

6.1 Classification Performance

The following metrics are used to measure the quality of predictions.The symbols used below are :tp is the number of true positives, fp the number of false positives and fn the number of false negatives.

1. Precision
   The ratio of tp / (tp+fp) is precision.
   Precision is intuitively the ability of a classifier not to label as positive a sample that should be negative. Value 1 is the best and 0 is the worst.

2. Recall
   The ratio of tp / (tp + fn) is recall.
   The recall is intuitively the ability of a classifier to find all the positive samples.Value 1 is the best and 0 is the worst.

3. Balanced Accuracy It is the average of recall obtained in each class.Value 1 is the best and 0 is the worst.

4. F-measure
   The F1 score is a weighted average of the precision and recall.
   The formula is :

$$F1 = 2 * (precision * recall)/(precision + recall)$$

6.2 Label Transformation

Recall from Section 4.2.1, there are four features and 4 outputs in each label per sample. In theory, there exist 216 labels but only 75 are active in this work. In order to use the off-the-shelf library for metrics and classifiers, multioutput-labels will transform to numeric values as shown in the figure **??**.

	CPU_Limit	In_FrameRate	In-Bitrate	In_Resoultion	Presest	Out_Resolution	CRF
0	1000	50	2488341	2	1	2	12
1	1000	23	1193210	2	3	2	8
2	1000	50	1244181	2	1	2	8
3	1000	30	746509	2	1	2	8
4	1000	12	298603	2	4	2	8

(a) multioutput multiclass labels

	CPU_Limit	In_FrameRate	In_Bitrate	In_Resolution	Label
0	1000	50	2488341	2	55
1	1000	23	1193210	2	126
2	1000	50	1244181	2	54
3	1000	30	746509	2	54
4	1000	12	298603	2	162

(b) multiclass labels

Figure 6.1: Label Transformation

6.3 Result

Optimizing performance for machine learning models is a key step in making accurate predictions. Hyperparameters define characteristics of the model that can impact prediction accuracy and computational efficiency. They are typically set prior to fit the model to the data. In contrast, parameters are values estimated during the training process that allows the model to fit the data. Hyperparameter tuning is necessary for any ML algorithm. Grid Search will use in the tuning process which looks through each combination of hyperparameters. This means that every combination of specified hyperparameter values will be tried. The following hyperparameter in random forest will be tuned:

1. Default Value

Default Value	Value
oob score	0.6049
precision score	0.7
recall score	0.7
balanced accuracy score	0.625
f1 score	0.7

Table 6.1: Default Value

2. n_estimators
   It is the number of decision trees in the forest. Small n_estimators value will lead to underfitting easily while large n_estimators value is computationally expensive. After n_estimators reaches a certain value, the model improvement obtained by increasing n_estimators will be small.The n_estimators parameter is tuned, and the other parameters are the default values. The range of n_estimators parameter is 1 - 201, and the step size is 10, with 3-fold cross-validation. The best n_estimators found is *81* . Its mean cross-validated score is *0.599*

3. max_features
   Max_features determines the number of features to resample. Larger max_feature values can result in improved model performance because trees have more selection of features from which choose the best split but can also cause trees to become less diverse and lead to the potential for overfitting. Generally, we can use the default "auto". In tunning , other parameters are set to constant, and n_estimators is set to 81. The range of max_features parameter is 1-4, the step size is 1 with 3-fold cross-validation. The best max_features found is *3* . Its mean cross-validated score is *0.6078*

4. max_depth
   Larger numbers of splits allowed in each tree enable the trees to explain more variation in the data, however, trees with many splits may lead to overfitting. The decision tree will not limit the depth of the subtree if this value is ignored. Generally speaking, this value can be ignored when there are few data

or features. If the model has a large sample size and many features, it is recommended to limit this maximum depth. The specific value depends on the data distribution. Commonly used values can be between 10-100. *This value is ignored.*

5. min_samples_split
   Min_samples_split is the minimum number of samples required to split each node. A large value may lead to under-fitting, as the trees won't be able to split enough times to achieve node purity. This hyperparameter should be based on the number of records in the training dataset. The default value is 2. If the sample size is not large, the default value is recommended. If the sample size is very large, it is recommended to increase this value. *This value is ignored.*

6. min_samples_leaf
   Min_samples_leaf is the minimum number of samples required for each leaf. If the value is too large, the trees may not be able to split enough to create sufficient variation in the data. Optimal values for this hyperparameter are dependent on the size of the training set. The default value is 1. If the sample size is not large, the default value is recommended. If the sample size is very large, it is recommended to increase this value. *This value is ignored.*

7. min_weight_fraction_leaf
   This value limits the minimum value of the weighted sum of all samples of the same leaf node. If it is less than this value, the node will be removed. The default value is 0 means that the weight issue is not considered. Generally speaking, if we have more samples with missing values, or if the distribution of the category of the classification tree samples has a large deviation, this value should be tuned. Otherwise, the default value is recommended. *This value is ignored.*

8. max_leaf_nodes
   By limiting the maximum number of leaf nodes to prevent overfitting. The default is "None", that is, the maximum number of leaf nodes is not limited. If there are not many features, it is recommended to use the default value. *This value is ignored.*

9. min_impurity_split
   This value limits the growth of each decision tree in RF. If the impurity of a node (based on Gini coefficient or mean square error) is less than this threshold, the node no longer generates child nodes. It is generally recommended to use the default value 1e-7.*This value is ignored.*

10. Tuned model
    The OOB score of the tuned model increases slightly.

Tuned model	Value
oob score	0.617647
precision score	0.7
recall score	0.7
balanced accuracy score	0.625
f1 score	0.7

Table 6.2: Tuned model

Remark. The above results are using 1020 samples out of the 1050 samples, the remaining 30 samples are kept for another evaluation below using the same hyper-parameters.

30 Samples	Value
n_estimators	81
oob_score	True
max_features	3
max_depth	default
min_samples_split	default
min_samples_leaf	default
min_weight_fraction_leaf	default
min_weight_fraction_leaf	default
max_leaf_nodes	default
min_impurity_splite	default
mean accuracy	**0.7**

Table 6.3: 30 Samples

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Edge computing holds the key to the success of lightning-fast 5G networks. In this thesis , we have presented a Machine Learning(ML) solution that learns the optimal encoding parameters. Given an available CPU resource, the ML model determines the best encoding parameters which give the best encoded video. Experimental results have highlighted many facts worth discussing. The Random Forest Classifier is strong enough even without hyperparameter tuning, it obtains a similar score on all metrics when compared to the model with hyperparameter tuning. RF classifier is a powerful classifier, the final accuracy is not bad as the training data samples are extremely insufficient, there is only one sample in some classes. The main ordeals is the data generation process. Neglect the need for a standalone environment to avoid resource conflict and the compatibility and stability of different tools, it is hard to seek a group of source videos. The format of these videos should be the same such as pristine. This is the main reason cause the data set to be insufficient. Another reason is the data generation process is very time-consuming. The unstable output from these tools (e,g FFmpeg) will lead to a restart in the generation process. I firmly believe the performance of the RF classifier will significantly improve when there are enough samples for training.

7.2 Future Work

I am deeply convinced that the final points this work has lead to are worth exploring. Additional features can be analyzed to further improve the accuracy in predictions. For instance, 'video type' can be added as a new feature. Frame variation is usually high in action move and low in sports video, this information can facilitate the classification process. Aside from that, the ML model can dynamically learn computing resource and network resources. If there are sufficient network resources, we can adopt a faster preset value to speed up the encoding process by scarfing the file size which will increase the transmit time in the network.

Appendix A

Experiment Video

UGC Dataset

	A	B	C	D	E
1	4K	Input Bitrate	Size	Input Framerate	Input Resolution
2	0.mkv-HDR-2160P-5275.mkv	5 956 Mbps	13.9 GiB	29.970 (30000/1001) fps	3840x2160
3	1.mkv-Sports-2160P-7a11.mkv	2 983 Mbps	6.94 GiB	29.970 fps	3840x2160
4	2.mkv-Gming-2160P-6cd8.mkv	4 972 Mbps	11.6 GiB	50.000 fps	3840x2160
5	3.mkv-Vlog_2160P-7324.mkv	2 488 Mbps	5.79 GiB	25.000 fps	3840x2160
6	4.mkv-Vlog-2160P-03a9.mkv	2 389 Mbps	5.55 GiB	24.000 fps	3840x2160
7	5.mkv-Gaming_2160P-2436.mkv	5 967 Mbps	13.9 GiB	60.000 fps	3840x2160
8	6.mkv-HDR_2160P-6c6e.mkv	6 364 Mbps	14.8 GiB	23.976 (24000/1001) fps	3840x2160
9	7.mkv-HDR_2160P-664d.mkv	4 977 Mbps	11.6 GiB	25.000 fps	3840x2160
10	8.mkv-HDR_2160P-4581.mkv	4 778 Mbps	11.1 GiB	24.000 fps	3840x2160
11	9.mkv-Vlog_2160P-310b.mkv	2 986 Mbps	6.95 GiB	30.000 fps	3840x2160
12	10.Vlog_2160P-77d8.mkv	2 386 Mbps	5.56 GiB	23.976 (24000/1001) fps	3840x2160
13	11.mkv-Gaming_2160P-3a25.mkv	5 867 Mbps	13.7 GiB	60.000 fps	3840x2160
14	12.mkv-Vlog_2160P-2953.mkv	2 386 Mbps	5.55 GiB	23.976 (24000/1001) fps	3840x2160
15	13.mkv-Animation_2160P-41dc.mkv	1 493 Mbps	3.46 GiB	15.000 fps	3840x2160
16	14.mkv-Gaming_2160P-2cb1.mkv	5 951 Mbps	13.9 GiB	59.940 fps	3840x2160
17					

Figure A.1: .
4K Video

	A	B	C	D	E
18	1080P	Input Bitrate	Size	Input Framerate	Input Resolution
19	0.mkv-HDR_1080P-46a4.mkv	2 488 Mbps	5.79 GiB	50.000 fps	1920x1080
20	1.mkv-HDR_1080P-49d6.mkv	1 193 Mbps	2.78 GiB	23.976 (24000/1001) fps	1920x1080
21	2.mkv-Gaming_1080P-51fc.mkv	1 493 Mbps	3.48 GiB	60.000 fps	1920x1080
22	3.mkv-TelevisionClip_1080P-5e68.mkv	746 Mbps	1.74 GiB	29.970 fps	1920x1080
23	4.mkv-NewsClip_1080P-67dc.mkv	622 Mbps	1.45 GiB	25.000 fps	1920x1080
24	5.mkv-Gaming_1080P-0a5b.mkv	1 490 Mbps	3.47 GiB	**59.859 fps**	1920x1080
25	6.mkv-Gaming_1080P-3a9d.mkv	597 Mbps	1.39 GiB	23.976 (24000/1001) fps	1920x1080
26	7.mkv-HDR_1080P-13eb.mkv	2 986 Mbps	6.95 GiB	60.000 fps	1920x1080
27	8.mkv-HDR_1080P-548b.mkv	1 659 Mbps	3.85 GiB	25.000 fps	1920x1080
28	9.mkv-HowTo_1080P-36a9.mkv	775 Mbps	1.72 GiB	31.139 fps	1920x1080
29	10.mkv-MusicVideo_1080P-7736.mkv	1 244 Mbps	2.90 GiB	50.000 fps	1920x1080
30	11.mkv-Vlog_1080P-35cd.mkv	747 Mbps	1.74 GiB	30.000 fps	1920x1080
31	12.mkv-Animation_1080P-3e01.mkv	299 Mbps	709 MiB	12.000 fps	1920x1080
32	13.mkv-HowTo_1080P-763c.mkv	1 492 Mbps	3.47 GiB	59.940 fps	1920x1080
33	14.mkv-Vlog_1080P-1e70.mkv	746509 kb/s	1.74 GB	30fps	1920x1080
34					

Figure A.2: .
1080P Video

	A	B	C	D	E
35	720P	Input Bitrate	Size	Input Framerate	Input Resolution
36	0.mkv-Animation_720P-0acc.mkv	265165 kb/s	633 MiB	23.976 (24000/10	1280x720
37	1.mkv-Gaming_720P-0fdb.mkv	332 Mbps	791 MiB	30.000 fps	1280x720
38	2.mkv-NewsClip_720P-72d2.mkv	331 Mbps	791 MiB	29.970 fps	1280x720
39	3.mkv-Sports_720P-5ea4.mkv	553 Mbps	1.29 GiB	50.000 fps	1280x720
40	4.mkv-Animation_720P-01b3.mkv	276 Mbps	659 MiB	25.000 fps	1280x720
41	5.mkv-CoverSong_720P-60d3.mkv	333 Mbps	795 MiB	30.120 fps	1280x720
42	6.mkv-CoverSong_720P-6626.mkv	166 Mbps	396 MiB	15.000 fps	1280x720
43	7.mkv-Gaming_720P-0fba.mkv	664 Mbps	1.55 GiB	60.000 fps	1280x720
44	8.mkv-Gaming_720P-4813.mkv	666 Mbps	1.54 GiB	60.180 fps	1280x720
45	9.mkv-Lecture_720P-003a.mkv	265 Mbps	632 MiB	24.000 fps	1280x720
46	10.mkv-Lecture_720P-094d.mkv	220 Mbps	526 MiB	19.933 fps	1280x720
47	11.mkv-Sports_720P-058f.mkv	1 326 Mbps	3.09 GiB	119.880 fps	1280x720
48	12.mkv-TelevisionClip_720P-3f4c.mkv	592 Mbps	559 MiB	53.569 fps	1280x720
49	13.mkv-VR_720P-748e.mkv	472 Mbps	788 MiB	42.715 fps	1280x720
50	14.mkv-Sports_720P-2c80.mkv	330 Mbps	784 MiB	29.795 fps	1280x720

Figure A.3: .
720P Video

	A	B	C	D	E
52	480P	Input Bitrate	Size	Input Framerate	Input Resolution
53	0.mkv-Animation_480P-7ef2.mkv	123 Mbps	293 MiB	25.000 fps	854x480
54	1.mkv-NewsClip_480P-31bd.mkv	148 Mbps	352 MiB	30.000 fps	854x480
55	2.mkv-TelevisionClip_480P-280f.mkv	147 Mbps	351 MiB	29.970 fps	854x480
56	3.mkv-Gaming_480P-4560.mkv	293 Mbps	650 MiB	59.545 fps	854x480
57	4.mkv-HowTo_480P-187c.mkv	147087 kb/s	348 MiB	29.899 fps	854x480

Figure A.4: .
480P Video

Bibliography

[1] Hassan, Najmu e Yau, Kok-Lim e Wu, Celimuge *Edge Computing in 5G: A Review*, IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2938534

[2] Cbinsights, *What Is Edge Computing?* , URL:https://www.cbinsights.com/research/what-is -edge-computing/,April 28, 2020

[3] *Live-Streaming*, https://viewpointdv.com/live-streaming/

[4] *What Is Latency?* ,https://www.cloudflare.com/learning/performance/glossary/what-is-latency/

[5] *What Is Live Streaming?* ,https://www.cloudflare.com/learning/video/what-is-live-streaming/

[6] Nitinder Mohan*Edge Computing Platforms and Protocols*,ISBN 978-951-51-5561-0 (PDF)

[7] *Support Vector Machine — Introduction to Machine Learning Algorithms* ,https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[8] Dhiraj K *Top 4 advantages and disadvantages of Support Vector Machine or SVM* ,https://medium.com/@dhiraj8899/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107

[9] Prince Yadav *Decision Tree in Machine Learning* ,https://towardsdatascience.com/decision-tree-in-machine-learning-e380942a4c96

[10] Dhiraj K*Top 5 advantages and disadvantages of Decision Tree Algorithm*,https://medium.com/@dhiraj8899/top-5-advantages-and-disadvantages-of-decision-tree-algorithm-428ebd199d9a

[11] Tony Yiu *Understanding Random Forest*,https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[12] *Random Forest Algorithm*,https://www.javatpoint.com/machine-learning-random-forest-algorithm

[13] *Quora*,https://www.quora.com/What-are-some-advantages-of-using-a-random-forest-over-a-decision-tree-given-that-a-decision-tree-is-simpler

[14] https://zhuanlan.zhihu.com/p/54539091

[15] https://zhuanlan.zhihu.com/p/54950132

[16] *3D Video Quality Assessment*,https://www.epfl.ch/labs/mmspg/downloads/3dvqa/

[17] Netflix Technology Blog*Toward A Practical Perceptual Video Quality Metric*,https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652

[18] *What is a Container?* https://www.docker.com/resources/what-container

[19] Simplilearn*What is Docker: Advantages and Components* https://www.simplilearn.com/tutorials/docker-tutorial/what-is-docker

[20] *FFmpeg?* https://www.headendinfo.com/what-is-ffmpeg/

[21] *FFmpeg*https://ffmpeg.org/