ISSN: 0974-8571 Vol.13 No. 2 December, 2021

International Journal of Computational Intelligence in Control

YouTube Video Popularity Analysis & Prediction Using Deep Learning

Deshak Bhatnagar
Department of Computer Science,
Birla Institute of Technology and Science, Pilani,
Dubai International Academic City,
Dubai, United Arab Emirates.
deshak45@gmail.com

Siddhaling Urolagin
Department of Computer Science,
Birla Institute of Technology and Science, Pilani,
Dubai International Academic City,
Dubai, United Arab Emirates.
APP Centre for AI Research (APPCAIR),
BITS-Pilani, Goa Campus, India.
siddhaling@dubai.bits-pilani.ac.in

Date of Submission: 15th August 2021 Revised: 27th October 2021 Accepted: 18th December 2021

How to Cite:. **Deshak Bhatnagar & Siddhaling Urolagin (2021). YouTube Video Popularity Analysis& Prediction Using Deep Learning. International Journal of Computational Intelligence in Control 13(2)**

Abstract*-With the current situation of Covid-19 still developing all around the World & plus no or little relief on horizon yet, many people lost their jobs in past 2 years and some of them started working again on a already present self-employing system known as YouTube. It is still what we call a up & coming technology, and a product owned by the tech giant Google Inc. The YouTube is a platform to make money as well as gain popularity in a instant if your videos are right & interesting. A simple video can be made in a jiffy and put on or uploaded in matter of minutes or two. This led to a rapid increase in the number of viewers & views, likes on the videos as well as the subscribers on the channels in the past two years. But it is still a big mystery not only for the users but the creators as well that how videos are listed in terms of Popularity on the platform. So, we decided to take up a quite an unusual task of heading out the Research on Popularity Analysis & Predicting the YouTube video (YUTE) success using a Deep Learning (DL) framework. As mentioned this study only intends to process the Popularity Analysis of YouTube video (YUTE) and Predict their future success based on features-extracted from them. The secondary aim here about which it talks is a Dataset which is quite diverse & well versed in terms of different genre or categories of videos and the also in terms of number of views and likes already accumulated. The study further dictates the benefits of the system as well as its applications and its uses in the future, but it primarily focuses on explaining about a Deep Learning (DL) Framework and along with it a complete system which is efficient, robust & reliable.

Index Terms: **YouTube, Deep Learning (DL), YouTube video (YUTE), Video Analytics (VA), Neural Networks, Layers, Views, Likes, Scene Quality, Feature_Extraction, In-Video.**

INTRODUCTION

Video based content is one of the biggest stakeholder right now in the online World and currently controls as well as attracts a huge number of the internet traffic or the internet users. Plus, it is growing at such an exponential rate that its is expected to be the largest stakeholders not only in the online Entertainment domain but along with it in the Education as well. Few years ago, while the Video based content was just solidifying its roots in the market, a big platform was launched with aim of easing the way of accessing and watching videos. It is currently the biggest Free Video Streaming Platform and we call it as YouTube. Along with the spike of free streaming videos online, a demand for video editing tools also sore, which further led to the need of analyzing these videos to get better results of them and this gave birth to Video Analytics (VA). Video Analytics (VA) simply helps us in identifying that what is going on or is occurring in the video and can help us list some of these things, for example cars, emotions, facial detection, colour, location and etc. Whereas the concept of analyzing a YouTube video (YUTE) is not so common and their is little or no Research in this particular domain. The Video Analytics (VA) involves use of Artificial Intelligence as well as the concept of Deep Learning (DL) to accomplish the task with usage Python Programming. It will also involve Video Mining for feature extraction and creation of few Models for this task. As explained earlier I do not have a lot of previous work to back up this project but I do have references to quite a bit of related work done with Video Analytics (VA) in other domains like Live Video Feed.

The main primary Objective that we are trying achieve here is to successfully complete all the points mentioned down below with absolute accuracy and no error whatsoever. The secondary Objective is to not only simply achieve the below mentioned points but to create a Dataset with variety or different categories or genre of YouTube video (YUTE) along with the channel names, views, likes & duration.

The following are the primary Objectives of this study:

- To classify, differentiate & analyse different YouTube video (YUTE) at large in terms of Views & Likes
- Utilize Artificial Neural Networks and other Machine Learning techniques to extract video features
- A Dataset will be prepared of videos from popular YouTube video (YUTE)
- Perform the analysis for the Popularity of the YouTube video (YUTE)
- Perform the prediction of video to their Popularity

IMAGE 1
DEPICTS THE OVERALL REACH OF YOUTUBE VIDEO (YUTE) AROUND THE WORLD

The YouTube video (YUTE) Popularity Analysis & Prediction is just nothing but streamlining & getting to know about the most popular video among a fairly large amount of Dataset. Meanwhile, the Dataset has been solely procured as well as created in-house using our own understanding & knowledge of the YouTube system and how it works that is the, videos from the channels which are most popular and the one's that we like or watch or admire. The Dataset itself is quite a mixture of different genre or categories of videos. Plus, it also boosts some foreign languages apart from Hindi & English in it. The YouTube video (YUTE) Popularity Analysis & Prediction simply aims at finding out and expressing throughput the most famous video along with its further gain in popularity by predicting its future reach and the performance, further we also aim at creating a more robust as well as a dynamic system which can be used in other studies.

RELATED WORK

The YouTube video (YUTE) creation, watching and as well as referencing is one of the biggest markets nowadays. Further the focus on creating more & more self sustaining employment opportunities recently has led YouTube in becoming one of the biggest self employer for a lot of people. Creating a Video & putting or uploading it onto YouTube is just a matter of minutes, plus getting likes & views on it is isn't so difficult if the content is right & interesting. But despite being one of the most famed & newest product of Google it has seen a rise in number of new You-tubers which in-turn has led to influx in number of views. So, we decided to investigate & study this by developing our own model using Deep Learning (DL) & by the help of some features from & of the videos uploaded on YouTube. Not a lot of work has ever been done on this concept at any scale as far as our knowledge, hence it was really hard to find any related or similar work but the work done in the field of Video Analytics (VA) as well as popularity ratings of the systems and videos in general is what we tried to accumulate here. The concept of YouTube video (YUTE) Popularity Analysis & Prediction using Deep Learning (DL) might be new but the work in the field of Popularity Analysis as well as Predictions dates a long way back and is quite useful & helpful in what we are trying to achieve here. The [1], prescripts that every minute roughly 300 Hours of videos are uploaded onto YouTube with number of creators also increasing everyday. The rise of innovative smartphones is further leading to enhanced videos in terms of quality as well as speed, lighting, capturing of moving objects & faces plus the coverage demographic is also growing faster. In [2], the increase in number of videos as well as creators has led to creation of Video Analytics (VA) Systems (VA) to handle such big Datasets. These Video Analytics (VA) Systems provide users with the ability to perform several functions of videos like cropping, filtering, performance tracking as well as editing at large. Plus it also provides some new features like object & activity recognition, face recognition and so on. While the [3], tries to address a big issue of network bottleneck between cloud and edge computing framework. Plus it uses a Deep Learning (DL) model to perform Video Analytics (VA) tasks along with it includes object recognition which gives a 71% efficiency at last after going through the process. The [4], focuses on storage part of the Video Analytics (VA), as its important to the core of the system that the analytics or analysis of the data should be stored properly. It has the ability to store videos of different formats as well as it can retrieve & configure large videos and is specifically designed to manage them independently. In [5], a new storage system for videos is introduced with aim to achieve low cost, high efficiency and high accuracy. It further estimates a improvement of around 54% in video database management and 45% decrease in costs, it also lets it creators focus more on logic than the retrieval process. [6], discusses Deep Learning (DL) in depth along with it s computational graphs and its role in Machine Learning. While it deeps down into the World of Neural Networks explain use of regularization as well as normalization. To quote Deep Learning (DL) we can say it is a sub topic of Machine Learning which takes out high level features from unprocessed data. [7], simply describes the historical usage of Deep Learning (DL) in associations with YouTube. The method used is a split two way street in which a Deep Learning (DL) model is constructed and then a separate recommendation system is integrated for getting recommendations by using neural networks. It focuses on high level performance & improvements. The [8], sheds light on the improved scalability and heterogeneity of Deep Learning (DL) (DL). In [9], the analysis of the information supplied through multiple textual data & addition of Deep Learning (DL). Plus, also caters as a usher for the percipient in order to apply Deep Learning (DL) techniques.

In [10], the focus turns on Big Data & Deep Learning (DL) as they both go hand in hand, also our own dataset come under the category. Big Data actually collects and analyze huge volumes of data. It shows various Deep Learning (DL) techniques for implementation of Video Analytics (VA). As we know [11], portrays Deep Learning (DL) as quite a new concept and is currently utilized as a template to make important conclusion in finance & medicine, so on. As it is in its early childhood, but is probable that it will succeed in future as it can benefit from the huge sum of computation quality & data. [12], discusses a new framework named as AutoML as it maximizes the analytics and minimizes frame rates to constraints by configuring wireless networks & services jointly. It uses safe exploration in order to get near safe optimal rate of configuration. Also, use of Multiple Access-Edge Computing helps in calibrating a optimal solution to the issues. The [13], lays that Video Analytics (VA) itself has many applications starting from the video surveillance to the video content handling system. Plus this context leads to various scholarly & industrial solutions. The proposed system is called Distributed Video Analytics (VA) Framework for Intelligent Video Surveillance (SIAT), actually is not only able to process batch Video Analytics (VA) & real time videos at large. In [14], a distributed framework is enchanted in order to create perfect harmony between data and Video Analytics (VA) using cloud and edge computing. Feasibility of Deep Learning (DL) framework modules is determined by running edge components on IoT devices and use of AWS approach is done. In [15], Deep Neural Networks apart of Deep Learning (DL) can accomplish a lot high accuracy on computer vision work than classical Machine Learning algorithms. Edge Computing is a framework for Video Analytics (VA). It also furnish a upper-level, task oriented API for the developers.

In [16], Distributed Intelligent Video Surveillance (DIVS) system using Deep Learning (DL) algorithms is used & deployed in an Edge Computing Environment. The DIVS scheme can transmigrate computing work to network edges in order to reduce vast network communication & give low-latency. The [17], proposes Filter-Forward, a new Edge-Cloud System that alter data-center applications to operation content from 1000 of cameras through installation of lightweight Edge filters that focus on the related video frames. Filter-Forward acquaints faster & communicative petition small classifiers that portion computation to at the same time detect lots of outcome on computing strained Edge Node. It only transfers similar events to the cloud and Filter Forward do decrease the bandwidth in order to increases the accuracy of detection and efficiency. The [18], rebukes Video Analytics (VA) Arrangement by quoting that it executes self-acting events, movements & activity recognition. This leads to the outcome of a huge number of video data that is required to be vulcanized, optimized for the performance of Video Analytics (VA) systems. In [19], the focus shifts onto tuning hyper parameters tied in with Deep Learning (DL) models. A proposal on self-acting Video Object Classification way to for validation of system is made. The [20], further digs deep into the Cloud based Deep Learning (DL) model to classify objects, tuning of hyper-parameter with rate of learning and momentum. The deeper experimentation outcome including system validation of 100 GB on 8 node clouds with over 88,000 frames gives accuracy rate 0.97 & 0.96 for extensive execution of over 6 hours.

The [21], discusses the viewing patterns in videos with a focus on finding common grounds to extract certain features for the Video Analytics (VA). It concludes the implication of this study is regular monitoring which pain-taking and quite hard but is worth as the outcome is a set of correlated feature set. In [22], the Deep Learning (DL) investigates the video compressive sensing in the ambit of snap compressive imaging called SCI. In the video form of SCI, aggregated high speed frames are inflected by contrasting coding structures & further low speed sensor acquires the combine of the inflected frames. While the Quantitative Analysis on the above data is performed using comparison simulation on the synthetic dataset. In [23], a focus on temporal feature extraction is expressed for the Video Analytics (VA) in the offing to aid video segmentation process. Plus, it further leads to the creation of the biggest dataset of over 4,000 videos as the older video dataset of 90 short clips was too small and limited to do any segmentation scaling of videos. The [24], tries to exhibit the quality of the data freely extracted through YouTube Analytics & how it further exploits videos. It do detail the value of working on a profound analysis of the extracted data. The [25], explains rise plus success of video content in recent years and its scope in future. While it argues about and unveils tool for extracting over 100 variables from the YouTube by using Video Analytics (VA).

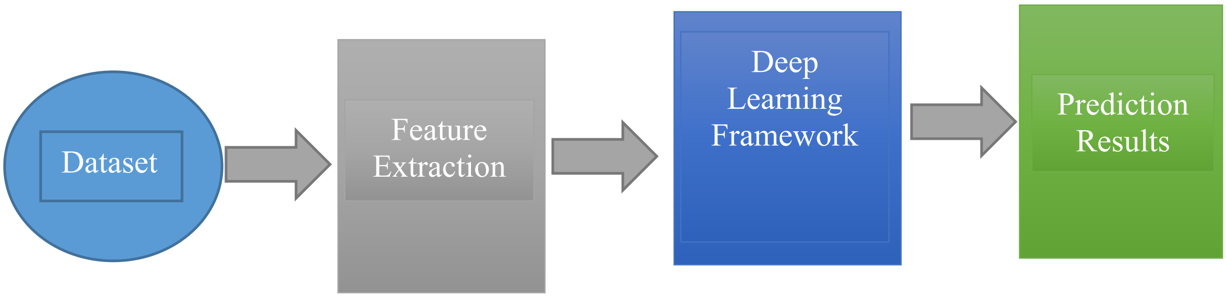

FIGURE 1
YOUTUBE VIDEO (YUTE) POPULARITY ANALYSIS & PREDICTION MODEL

METHODOLOGY

The problem which we are dealing here is essentially a Classification Problem. It is a type of problem in which the data is grouped all-together based on predetermined or already decided upon criteria or characteristics. The Classification Problem actually comes under the category of Supervised Learning which is the most used and spoken about type of Machine Learning. While the process of Prediction is quite easy when we speak but eventually it is a lot of work to do or undertake. The Architecture Model shown above depicts the complete process or steps through or by which YouTube video (YUTE) Popularity Analysis and Prediction using Deep Learning (DL) work. The whole process is divided into 3 stages, starting with Dataset creation moving onto feature_extraction and then the Prediction part which involves use of Deep Learning (DL) Framework, specifically using the Sequence Classifier. While the actually process simply starts with Feasibility Analysis of the topic and Requirement Specifications. Next, the YouTube Video URL Dataset is a combination of 6 carefully chosen features or attributes including the actual url's of the videos alongside Views & Likes, and there are total of 200 Videos carefully selected with duration of as low as 3 seconds to as high as over 30 minutes. The Dataset also provides with the information of the creator such as the name of the video and the name of the channel. The Feature Extraction is the next stage of this architecture and is a quite important part of the whole process of Prediction.

I. *FEATURE EXTRACTION*

The Complete Feature Extraction Process is done on the Google Colab Platform, the actual process starts with reading the YouTube Url Dataset.csv file and then extracting two important features from it i.e., why Views & Like, further save them into a csv file called as yt.csv and download it. The features are read using the Pandas library then extracted using the simple technique of selecting one or more columns or features or attributes and then saving them into simple csv file. This all is done using simple Machine Learning Technique. Next, we move onto the extraction of a in-video feature, the actual process of extracting another feature starts when we target extracting feature from the YouTube Url Dataset.csv file which is the Url's of the videos into a simple file and these urls are extracted using the same technique mentioned above with the exception of not downloading the file. The features are directly extracted from the YouTube Url Dataset.csv file using the simple Machine Learning Technique employed using the Python Language, its simply works by selecting the required columns or features or attributes and then saving them into another files and in the end displaying them instead of downloading. To quote these urls are very important they are further needed or will be used for the Feature Extraction which involves in-video feature extraction.The in-video Feature Extraction starts with the creation of a url Dataset which is done by taking the Url's extracted earlier or above and further selecting the feature i.e., Frame Level Quality per Scene or simple called as Scene Quality. Eventually, only the Mean of the Scene Quality of every Scene in a Video is used. The process starts by simply importing some important libraries with first of them being the pillow, which helps in image manipulation as well as display. Second is tqdm, it is used for making progress bars. Third is shutil, offers high level operations to perform on files such as copy, create, etc. Here, we are using a GPU environment which is provided by Google Colab because we require video graphics extraction, Graphics Processing Unit is called as GPU. It one of the most important Computing-Technology for both Personal and Business Computing. It further leads to creation of a directory where all the downloaded videos are saved. Finally, the most important step in this whole process comes into effect and it is the usage of feature_main.py which finds the Scene Quality or just Quality by reading a video using OpenCV then going through each and every frame of the video or scene of the video & converting each scene or frame or image into gray scale or image or frame, then gives RGB value to each frame and further it calculates those values for every scene or frame then provides with a Mean value for the complete video and saves it in a .csv file for future use in Prediction Phase. The above mentioned file i.e., feature_main.py is imported and run then the values are collected and saved in a separate Qual.csv file. Then the above mentioned Qual.csv file is concatenated with the yt.csv file which has Views, Likes as its columns and they are saved into a new csv file called as ya.csv file for further usage in the Prediction Phase. The Scene Quality is measured in ps which is called as Progressive Scan. The Progressive Scan is a format which stores, displays moving images in a sequence of lines in a frame. The following is the part of the calculation for the above process the feature_main.py goes through.

```
if cap.get(cv2.CAP_PROP_POS_FRAMES) %
skip_frames == 0:
        #convert image to grayscale
        frame = cv2.cvtColor(frame,
cv2.COLOR_BGR2GRAY)
        frame_count =
cap.get(cv2.CAP_PROP_POS_FRAMES)
        qual_stats =
          qual_stats.append(pd.DataFrame([[frame_cou
          nt, cv2.Laplacian(frame,
            cv2.CV_64F).var()]],
columns=['frame_num', 'quality']),ignore_index=True,
sort=False)
```

II. Predictions

Now going further on we start the process by first importing the ya.csv file which has all the features which we extracted earlier in Feature_Extraction Phase. The features we have in ya.csv are Views, Likes & Quality or Scene Quality. Next step is train_test_split, here we divide our Dataset that is the ya.csv file into Training Data & Testing Data respectively. The Training Data & Testing Data is divided in a ration 70:30 which means the Training Data has a shape of (130,3) while the Testing Data has a shape of (70,3). The train_test_split is done by the help of a library called as sklearn, the sklearn splits a dataset into two subsets of arrays. The parameters used for this process can be as random as you want or specified to a extent as shown below.

$$X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size = 0.35)$$

Before we start with building up the Deep Learning (DL) Model, we need to make sure our dataset is good enough to perform efficient computations by properly scaling it. The scaling of dataset is done after train_test_split part discussed at the start. The scaling part involves usage of standard scaler a library which confirms that the dataset has zero as unit variable & mean. The fit_transform is the actual function you use to perform the process. Next we move onto the part of creating the Deep Learning (DL) Model which is specifically based or uses the Sequential Classifier and the process of Sequence Classification. The concept of classification in itself is quite interesting & highly used. Classification is a process that involves grouping data together by a particular criteria. While the Sequence Classification is a predictive modeling problem where you have some kind of sequence of inputs over space or time and the task is to predict a category of the sequence and in our case we use it to predict True or False whether the video will succeed or not by setting it up to check for a specific limit value or threshold value. For starting the actual Model we need to import keras library and tensorflow library & then import sequential in order to initialize Artificial Neural Network or simply ANN. Further you will also import dense so that we can create & add different layers to Deep Learning (DL) Model. But before moving forward we discuss the two libraries mentioned above; first the keras, it is a Neural Network components library which is open source and can run Tensorflow. While the Tensorflow is a end to end platform for creating Machine Learning based applications and is also open source which implies it is free to use. Now initialize the classifier with sequential and then start adding the layers to the network. Generally there are 3 layers in a network starting with,

- Initial layer or Input layer: It is a layer where you pass dataset features. There is no computations performed at this layer
- Hidden layer or Closed layer: This is where the computation occurs and results are passed to the output or final layer. It is the layer between initial and final. There can be multiple hidden layers in a neural network
- Final layer or Output layer: It is the layer which gives the results and is sometimes referred as neural network result layer

But in our case we are using only two layers because are computation requires that only. Also, we are using two layers for faster and better computation in terms of steady results i.e., predictions and simply to not make the neural network more complex then it is already is. This is how usually the input layer of a model looks like.

$$\mathrm{classifier.add(Dense(66, kernel\_initializer = "uniform", activation = "relu", input\_dim = 3))}$$

A network layer typically is a combination of few elements starting with activation function; it is where learning of the model occurs, they are usually linear or non-linear. Next is input_dim which is the number of features in a dataset. The element right at the start is the number of nodes that form the actual neural network, these are usually chosen at random. Also, sometimes the average of total number of dataset values is also considered for selecting the number of nodes for the network i.e., what we exactly did. The last one is the kernel_initializer, it is used to fit our Deep Learning (DL) model by initializing the weights almost to zero & not zero. To accomplish this we use the uniform distribution. Further we use Gradient Descent to minimize the error to the least. Actually, the Gradient Descent is the way by which we adjust the at random allotted weights in the network by reducing the function cost. We are using the 'adam' here as the optimization strategy and we use loss network reducing the function cost. We are using the 'adam' here as the optimization strategy and we use loss function as binary classification with accuracy the metric to be calculated. The example is mentioned below.

$$\text{classifier.compile}(\text{optimizer} = \text{"adam"}, \text{loss} = \text{"binary\_crossentropy"}, \text{metrics} = [\text{"accuracy"}])$$

Next we fit the classifier using keras and run the epochs, the epochs denotes the number of times the dataset will be passed by the network of layers or the neural networks. More epochs means more time taken for the results. This part usually has number of epochs as mentioned above, batch size of the dataset which will go through the neural network every time with training set which has the features & column set on which prediction will be done. Now we run predictions on the column set mentioned earlier by setting a threshold value and then creating a confusion matrix in order to predict the number of possible errors that can be made. Also, it provides the number of true positives, false negatives, true negatives & false positives of a classifier. All this done using the scikit-learn which provides the sklearn.metrics library. Going a bit further back to network layers in our case we are using the 'relu' activation function for the input layer as it simply generalizes the data well enough, plus the Linear functions are usually not considered beneficial for making predictions because they sort a straight line. And for the output or final neural network layer we use activation function called as 'sigmoid' to get the prediction or probability of whether a video will succeed or not. In case of more than 2 labels we use activation function which is a variant of 'sigmoid' called as 'softmax'. In the next step, we start making Predictions with our model, this is done by probability prediction whether the video will succeed over a certain threshold value and if it does then answer is True otherwise answer is False. Focusing onto the threshold value for the prediction making we have made the assumption using the fact which was outlined on Quora by verityfit.com & on a blog oberlo that in theory a person or individual only watches 40-60% of a video on average. Taking the fact mentioned earlier we decided to increase this watch percentage up-to the mark of over 65% and in our case exactly 69% to get better results, also considering that our dataset is full of videos which come under certain categories that are most liked around the World. Further, we took a 5-minute and calculated the 69% watch time on it which came to 3:45-minute to be exact and this justifies are claim mentioned above and hence using the 0.69 or 69% as our threshold value for Predictions. The example of how we actually predict is mentioned below.

$$\text{new\_pred} = \text{classifier.predict}$$
$$(\text{sc.transform}(\text{np.array}([[0,2,2]])))$$
$$a = (\text{new\_pred} > 0.69)$$

Now for this whole process of prediction making or predicting or predictions we use the video features extracted before i.e., views, likes & quality or scene quality. We simply scale the features and convert or transform them into numpy array as we did at the start. Following the similar process we make a total of 200 predictions which is the number of videos in our dataset and save it in prd.csv file and further we concatenate that prd.csv file with the YouTube Url Dataset.csv file which has Video Name, Channel Name, Views, Likes as its columns and they are saved into a new .csv file called as pr.csv file. Thus we represent the results acquired along with columns mentioned previously.

RESULTS

The results for the Predictions as well as the Model are accumulated using the following manner. The Dataset is first of all self-created using 2 major things or the criteria; The first one is the Creator Power, the second & the final one is the views and likes difference. Creator Power here is simply where we judged & selected videos with value or stardom of the Creator or video owner. The other is difference between views & likes which is a set bar of maximum consideration number of view & likes that actually in this case is Million for upper level and as low as in Hundreds. The YouTube video (YUTE) URL Dataset is a combination of total of 6 carefully chosen features or attributes including the actual url's of the videos alongside Views & Likes, and there are total of 200 Videos carefully selected with duration of as low as 3 seconds to as high as over 30 minutes. The videos on YouTube can be as long as you want but longer the videos the less popular they will that's why nobody puts longer video than 30 minutes except in case of live-streams. Also, the live-streams are not considered for the fact that they are highly un-predictable due to connection loss as well as time-limit is a issue as they are impossible to download & exploit for features. Further, the Dataset is taken and divided into Training set & Testing set. Using train_test_split the dataset is divided with a test_size of 0.35 or simply 35%. The Google Colab or known as Colaboratory by Google is specifically used for feature_extraction part, as this part involves usage of YouTube library & GPU (Graphical Processing Unit) for Scene Quality or Quality feature extraction. While Jupyter Notebook is used for Deep Learning (DL) part as we require use of Keras & Tensorflow along with these it has queries that require no-limit usage, plus environment gives better performance with Deep Learning (DL) Frameworks. The Libraries used from start are numpy, pandas, shutil, tqdm, os, youtube-dl, sklearn, keras & tensorflow along with matplotlib for visualization purposes especially graphs. We exploit results in form of Predictions for each video as well as some visualization in form of graphs.

	Video Name	Channel Name	Views	Prediction
0	This Video Is 3 Seconds	MrBeast Shorts	9.0000	[[True]]
1	Zipline Crash1	WAO RYU!ONLY in JAPAN	0.0180	[[False]]
2	Nice	MrBeast Shorts	3.5000	[[True]]
3	Priorities #shorts Credits everythingf1	F1 Shorts	0.0180	[[False]]
4	Threading the needle!!! Which circuit #shorts ...	F1 Shorts	0.0017	[[False]]
5	The Funniest Joke You Will Ever Hear In Your Life	MrBeast Shorts	6.7000	[[True]]
6	50 Foot Lego Tower Falls Over	MrBeast Shorts	4.7000	[[True]]
7	I Threw My Recording Phone Into A Ceiling Fan	MrBeast Shorts	9.2000	[[True]]
8	We Started A Band!	MrBeast Shorts	3.6000	[[True]]
9	When Ocon took Verstappen out in the lead!!! S...	F1 Shorts	0.0780	[[False]]

FIGURE 2
YOUTUBE VIDEO (YUTE) PREDICTION

We start with finding out the Predictions made for the total number of videos i.e., either True or False with respect to the threshold value, we have made the assumption using the fact which was outlined on Quora by verityfit.com & on a blog oberlo that in theory a person or individual only watches 40-60% of a video on average. Taking the fact mentioned earlier we decided to increase this watch percentage up-to the mark of over 65% and in our case exactly 69% to get better results, also considering that our dataset is full of videos which come under certain categories that are most liked around the World. The above shown results are for the 10 videos out of 200 videos with 6 of them showing True & only 4 showing False in the Prediction column which means that 6 videos will become popular from the first 10 our of 200 videos, this is based on the classifier calculation which we have carried out.

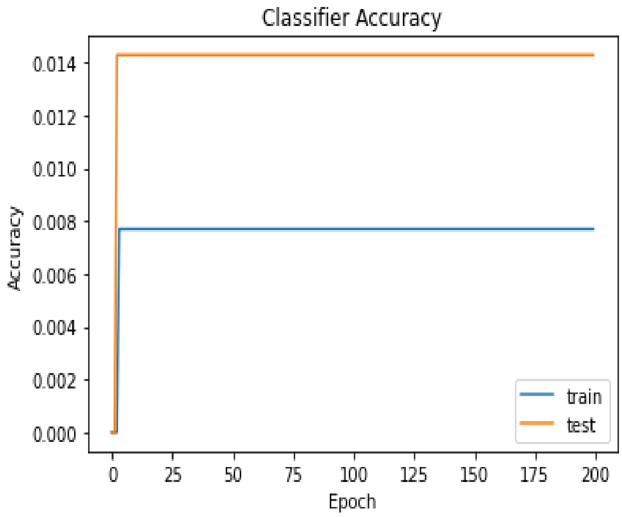

FIGURE 3
CLASSIFIER ACCURACY

Now, we go forward and a plot a graph which shows the Classifier Accuracy i.e., the plot of accuracy on the training and testing datasets over training epochs. The plot is carried out between the Accuracy & the Epoch. Simply, accuracy is the measure of being correct or free from error while epoch means carrying out of a single cycle through the whole dataset. The total number of epochs used are 200 which is same as the number of total videos in the dataset. Further, the training & testing datasets are obtained from already present X & Y values which are just initialized before the train_test_split & are used for carrying out the latter. Then the testing_size is determined & before all of this, the layers of neural network are created as well as sequential classifier is initialized along with this matplotlib is imported & used for the visualization purposes. In the results shown above, the plot simply depict the classifier accuracy which can further be improved a bit by training the classifier more as the accuracy for both train & test steadily increases at the start but then stabilizes. In our opinion it can be exploited a bit more. Plus, it also does not show any signs of over-fitting here.

The plot shows the Classifier Loss i.e., the plot of loss function or loss on the training and testing datasets over training epochs. The plot is carried out between the Loss Function & the Epoch. Simply, loss function or loss is the measure of quantifying that how well the classifier is performing based on the set target & the classifier output while epoch means carrying out of a single cycle through the whole dataset. The total number of epochs used are 200 which is same as the number of total videos in the dataset. Further, the training & testing datasets are obtained from already present X & Y values which are just initialized before the train_test_split & are used for carrying out the latter. Then the testing_size is determined & before all of this, the layers of neural network are created as well as sequential classifier is initialized along with this matplotlib is imported & used for the visualization purposes. In the results shown below, the plot simply depict the classifier loss which has comparable performance & shows drastic visualizations with a large dip in the train dataset in comparison to the test dataset which goes off the charts. In our opinion the epoch are trained well specifically in this case.

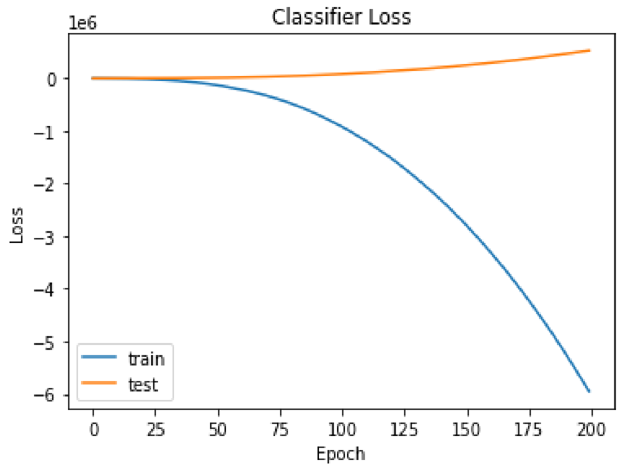

FIGURE 4
CLASSIFIER LOSS

CONCLUSION & FUTURE SCOPE

The condescending idea of not analyzing videos, specifically YouTube video (YUTE) for popularity & growth prediction which are all the rage in the World for quite a bit of time arguably for few years now. And is still growing constantly at a unbelievable rate or pace which even streaming services are not being able to match. The YouTube video (YUTE) Popularity Analysis & Prediction using Deep Learning (DL) not only works on the theory but is actually proven to give results further it adds some profile to it by exploring the YouTube video (YUTE) feature_extraction part in depth right from creation of dataset to the collection of all the progressive scan values for quality, along with this it also explains the usage of Deep Learning (DL) model & it working in depth. The whole model is to true to itself & hence gives the most accurate results that it can through the rigorous process, the Prediction values that is in this case the True & False value for each YouTube video (YUTE) as per the set threshold or minimum percentage or minimum succeed value. Finally, it provides us with two plots both over Training Epochs, the first plot is of the Accuracy on the Training & Testing datasets and other plot of the Loss on Training & Testing datasets. The methods as well as tools used in the development of this project are available to use only if you have the right or we can say the proper knowledge to apply them efficiently as well as accurately in order to get the best results. The use of sequential classifier as well as proper feature_extraction and determining the features to be used at the very start led to the perfect results which were envisioned throughout. Whereas quoting about the Future we can graciously conclude that it is capricious, and there is no real indicant what will unravel but YouTube Video Popularity Analysis & Predictions has a wide scope with aiming at improving as well as knowing further more about how the videos actually end-up on the Trending block or the page as well as it can be used to counter the issues many young creators face which are lack of motivation to put more content in order to elevate the popularity in totality. On a better-note, we forestall it to be used into more industrialized ways than personal usage and further we intend to explore it as basis for even more avenues in this particular domain with handy, high-performance & even more efficient system.

REFERENCES

[1] Verdhan, V. (2021). Video Analytics (VA) Using Deep Learning (DL). In Computer Vision Using Deep Learning (DL) (pp. 221-255). Apress, Berkeley, CA.

[2] Bastani, F., Moll, O., & Madden, S. (2020). Vaas: Video Analytics (VA) at scale. Proceedings of the VLDB Endowment, 13(12), 2877-2880.

[3] Ali, M., Anjum, A., Yaseen, M. U., Zamani, A. R., Balouek-Thomert, D., Rana, O., & Parashar, M. (2018, May). Edge enhanced Deep Learning (DL) system for large-scale video stream analytics. In 2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC) (pp. 1-10). IEEE.

[4] Xu, T., Botelho, L. M., & Lin, F. X. (2019, March). Vstore: A data store for analytics on large videos. In Proceedings of the Fourteenth EuroSys Conference 2019 (pp. 1-17).

[5] Haynes, B., Daum, M., He, D., Mazumdar, A., Balazinska, M., Cheung, A., & Ceze, L. (2021). VSS: A Storage System for Video Analytics (VA) [Technical Report]. arXiv preprint arXiv:2103.16604.

[6] Nikolenko, S., Kadurin, A., & Arkhangelskaya, E. (2018). Deep Learning (DL). SPb.: Peter.

[7] Covington, P., Adams, J., & Sargin, E. (2016, September). Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM conference on recommender systems (pp. 191-198).

[8] Hayat, M. K., Daud, A., Alshdadi, A. A., Banjar, A., Abbasi, R. A., Bao, Y., & Dawood, H. (2019). Towards Deep Learning (DL) prospects: insights for social media analytics. IEEE Access, 7, 36958-36979.

[9] Sun, T., & Vasarhelyi, M. A. (2018). Embracing textual data analytics in auditing with Deep Learning (DL).

[10] Jan, B., Farman, H., Khan, M., Imran, M., Islam, I. U., Ahmad, A., ... & Jeon, G. (2019). Deep Learning (DL) in big data analytics: a comparative study. Computers & Electrical Engineering, 75, 275-287.

[11] Arif Wani, M., Kantardzic, M., & Sayed-Mouchaweh, M. (2020). Trends in Deep Learning (DL) Applications. In Deep Learning (DL) Applications (pp. 1-7). Springer, Singapore.

[12] Galanopoulos, A., Ayala-Romero, J. A., Leith, D. J., & Iosifidis, G. (2021, May). AutoML for Video Analytics (VA) with Edge Computing. In IEEE INFOCOM 2021-IEEE Conference on Computer Communications (pp. 1-10). IEEE.

[13] Uddin, M. A., Alam, A., Tu, N. A., Islam, M. S., & Lee, Y. K. (2019). SIAT: A distributed Video Analytics (VA) framework for intelligent video surveillance. Symmetry, 11(7), 911.

[14] Pathak, T., Patel, V., Kanani, S., Arya, S., Patel, P., Ali, M. I., & Breslin, J. (2020). A Distributed Framework to Orchestrate Video Analytics (VA) Applications. arXiv preprint arXiv:2009.09065.

[15] Liu, P., Qi, B., & Banerjee, S. (2018, June). Edgeeye: An edge service framework for real-time intelligent Video Analytics (VA). In Proceedings of the 1st international workshop on edge systems, analytics and networking (pp. 1-6).

[16] Chen, J., Li, K., Deng, Q., Li, K., & Philip, S. Y. (2019). Distributed Deep Learning (DL) model for intelligent video surveillance systems with edge computing. IEEE Transactions on Industrial Informatics.

[17] Doleck, T., Lemay, D. J., Basnet, R. B., & Bazelais, P. (2020). Predictive analytics in education: A comparison of Deep Learning (DL) frameworks. Education and Information Technologies, 25(3), 1951-1963.

[18] Ibrahim, N., Maurya, P., Jafari, O., & Nagarkar, P. (2021). A Survey of Performance Optimization in Neural Network-Based Video Analytics (VA) Systems. arXiv preprint arXiv:2105.14195.

[19] Yaseen, M. U., Anjum, A., Rana, O., & Antonopoulos, N. (2018). Deep Learning (DL) hyper-parameter optimization for Video Analytics (VA) in clouds. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(1), 253-264.

[20] Yaseen, M. U., Anjum, A., & Antonopoulos, N. (2017, December). Modeling and analysis of a Deep Learning (DL) pipeline for cloud based Video Analytics (VA). In Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (pp. 121-130).

[21] Walsh, J. N., O'Brien, M. P., & Slattery, D. M. (2019). Video viewing patterns using different teaching treatments: a case study using YouTube analytics.

[22] Qiao, M., Meng, Z., Ma, J., & Yuan, X. (2020). Deep Learning (DL) for video compressive sensing. APL Photonics, 5(3), 030801.

[23] Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., & Huang, T. (2018). Youtube-vos: A large-scale video object segmentation benchmark. arXiv preprint arXiv:1809.03327..

[24] O'Brien, M., Slattery, D., & Walsh, J. (2019, November). Using YouTube Analytics to Investigate Instructional Video Viewing Patterns. In Proceedings of the 18 th Europea n conference on E-learning, Copenhagen, Denmark (pp. 428-436).

[25] Schwenzow, J., Hartmann, J., Schikowsky, A., & Heitmann, M. (2021). Understanding videos at scale: How to extract insights for business research. Journal of Business Research, 123, 367-379.

AUTHOR INFORMATION

Deshak Bhatnagar, M.E. Student, Department of Computer Science, Birla Institute of Technology and Science-Pilani, Dubai Campus, United Arab Emirates.

Siddhaling Urolagin, Assistant Professor, Department of Computer Science, APP Centre for AI Research (APPCAIR) BITS-Pilani, Goa Campus, India; Birla Institute of Technology and Science-Pilani, Dubai Campus, United Arab Emirates.