Classification of Live Video Stream from Pakistani News Channels (Urdu) using Deep Learning Latest Techniques

Author

MUHAMMAD AFZAL

Regn Number

318898

Supervisor

DR KARAM DAD KALLU

ROBOTICS & AI (R&AI)

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

MAY, 2023

Classification of Live Video Stream from Pakistani News Channels (Urdu) using Deep Learning Latest Techniques

Author

MUAHMMAD AFZAL

Regn Number

318898

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Robotics and Intelligent Machine Engineering

Thesis Supervisor:

DR KARAM DAD KALLU

Thesis Supervisor's Signature: ____

ROBOTICS & AI (R&AI)
SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,
ISLAMABAD
MAY, 2023

Declaration

I certify that this research work titled “*Classification of Live Video Stream from Pakistani News Channels (Urdu) using Deep Learning Latest Techniques”* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources has been properly acknowledged / referred.

Signature of Student
MUHAMMAD AFZAL
2019-NUST-MS-RIME-318898

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student
MUHAMMAD AFZAL
318898

Signature of Supervisor

Copyright Statement

-
-
-

Acknowledgements

I am thankful to my Creator Allah Subhana-Watala to have guided me throughout this work at every step and for every new thought which You setup in my mind to improve it. Indeed, I could have done nothing without Your priceless help and guidance. Whosoever helped me throughout the course of my thesis, whether my family or any other individual, was Your will, so indeed none be worthy of praise but You.

I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout every department of my life.

I would also like to express special thanks to my supervisor Dr. Karam Dad Kallu and Dr Hassan Sajid for their help throughout my thesis and also for Machine Learning and Deep Learning courses which he has taught me. I can safely say that I haven't learned any other engineering subject in such depth than the ones which he has taught.

I would also like to pay special thanks to Dr. Imran HoD Mechanical Dept College of EME for providing necessary computational resources. Without which I could not have completed my experimentation portion successfully and in time.

I would also like to thank Dr. Muhammad Jawad Khan for being on my thesis guidance and evaluation committee. I am also thankful for his support and cooperation.

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

Dedicated to my exceptional family whose tremendous support and cooperation led me to this wonderful accomplishment.

Abstract

In our contemporary era, information is of prime importance and its dominant use by social media and TV channels for making public opinion and cultural influence is quite evident. Videos form the major portion of media and contain more elaborate information than a single image. Today, videos are piling up in millions every day and their segregation, classification and analysis are upheaval tasks. Live TV video stream contains voice, metadata and image frames full of multiple information including written scripts etc. which can contribute to video classification. But utilization of each type of data we need to do a separate study. However, we have focused on classification of video stream using deep learning (DL) neural networks which are well established solutions for images and small videos classification and gesture recognition.

In our study, we have suggested a mechanism for classification of big or live video streams obtained from Pakistani TV News Channels into 5 classes (Advertisement, News, Talk Show, Sports & Entertainment Program) using supervised DL pretrained neural networks. Due to non-availability of authentic dataset on this subject, we have created a customized data of videos recorded (approximately 335 hours videos) from various sources like different TV channels' websites and YouTube. Videos were processed to extract image frames to prepare a trainable dataset. For our experimentation, we have mainly used pretrained ResNet variants (ResNet18, ResNet34, ResNet50, ResNet101 & ResNet152) on ImageNet dataset and few other models like AlexNet, ConvNeXt_Tiny, DenseNet121, SqueezeNet and VGG11 for comparison purposes. Then modified the last classification layer of the network as per number of target classes and finetuned all weights of neural network on the subject dataset. We carried out various experiments on these neural networks and achieved quite encouraging results having accuracies ranging from 95% to 99%. For testing of videos on trained models, dynamic averaging time domain window was applied to diminish the jitters in the output results. This can be useful in many other applications as well including social media & advertisements analysis, classification of small videos, industrial and business automation etc.

Key Words: Information, Video Stream Classification, Image Frame, Metadata, Dataset, Deep Learning, Neural Networks, ResNet, AlexNet, ConvNeXt, DenseNet, Squeeze Net, VGG11

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

In the present era, information plays an important role in different spheres of life. We can list hundreds of its roles/ contributions but some of the few important ones are like extracting relevant information for certain specific educational research, building masses/ public opinion through propaganda, for cultural influence, gathering knowledge, entertainment etc. Information can be found in many forms such as written scripts in hundreds of languages, arts, audios and for the last couple of decades videos have acquired the dominant part. Since the advent of internet and mobile phones, huge quantum of information is piling up. There are millions of websites like YouTube, Twitter, Facebook, TikTok, TV channels websites etc. on which hundreds of millions of videos are being added. Even if we talk about the statistics related to YouTube only, it is learnt that videos of approximately 300 hours length are being added every minute and over 5 billion videos are being watched daily on YouTube[1]. Moreso, there are hundreds of thousands of TV and Web channels[2][3] which are spreading their content 24 hours a day throughout the year through satellites dish receivers, cables and online. There are variety of channels[4] like entertainment, sports, games, informative, cultural, educational and especially news channels which are big source of current information, propaganda and public opinion making. Moreover, these TV channels are broadcasting their content almost in all languages of the world. It is also important to notice that individual videos being uploaded on websites are generally related to some particular topic but the most of TV channels especially News Channels are feeding their content as a continuous live stream of multiple programs such as talk shows, entertainment, sports programs, breaking news/ news etc. Moreover, TV channels broadcast these programs in segments which are intermittently filled with advertisements as well.

Today we know that videos, pictures and audios are the main source for spreading/ extracting some kind of information, communication of one’s opinion, advertisement, entertainment, news etc. Even one small well-prepared video can express a lot, which cannot be explained in dozens of pages. But at the same time, we know that extraction of useful and desired information from the huge video content uploaded or being uploaded online each minute individually or by TV channels is an upheaval task. This requirement can either be done manually, which is nearly impossible, or through some automatic means using Artificial Intelligence. At first stage of gathering useful information or analysis of videos, there is need to classify the video content using some efficient and accurate methods[4]. Classification of videos can range from very few classes to hundreds of classes as per the need and use. One of the better ways can be to classify first in a few broad classes and then further refine each class into sub classes using the same technique.

Classification of videos is intrinsically a difficult job[4] due to various reasons such as inherent diversity and variations in content leading to difficulty in labeling, having audios contrary to the integral images, requirement of huge datasets and need of heavy systems for computation etc. So far, a lot of work has been carried out for classification of images using different techniques of Machine Learning (ML) and Deep Learning (DL) but not much work is available on classification of videos. One of the main causes is the lack of availability of suitable generic dataset and requisite computational resources. As usually the datasets are related to some particular requirement or activity covering either audio features or images. Some of the popular video datasets relying on images are ActivityNet[3], Sports-1M[5] and UCF-101[6] which deal with only some certain subjects. Similarly some of the mostly used audio datasets having similar limitations are NOISET-92[7] and AENet[8]. As in large videos classification, big datasets with temporal dimensions and complex architectures are involved, therefore heavy systems with high computation power are needed for speedy computations during training of models for reducing long training time usually in weeks & months[5] and better accuracy. Moreover, labeling of videos is extremely difficult due to inherent noise owing to the presence of unnecessary, confusing or blank frames. One can observe that some of the frames get totally wrong vis a vis the label.

As News and Web Media i.e., TV channels is one of the biggest sources for making public opinion worldwide, therefore, its analysis is of prime importance. For better analysis we need to classify the live stream of news channels first in major classes and then using each class for further in-depth evaluation to extract the useful desired information. Moreover, lot of work has been done for image classification using mainly ML and more recently the DL techniques involving convolutional neural networks with fully connected (FC) layers[5]. In our study, we intend to classify live stream of video frames from Pakistani TV News Channels using DL convolutional neural networks with FC layers and showing results in temporal domain for a specific clocked clip of video. After going through different Pakistani News Channels, although one can identify various classes, but we have restricted the stream into five broader groups/ classes to keep the problem simple and manageable. Although effort has been made to keep each group/ class prominent from others by ensuring the clear difference but still there is possibility of having some similarities in some of the scenes. These classes are as under: -

- Advertisements. (‘**Advertisements’)**
- News including breaking news etc. (**'News'**)
- Talk shows; some may mix up with some sort of news of similar pattern. (**'Talkshow_News')**
- Sports or programs of which major portion include sport event. **('Sports')**
- Entertainment programs or some other programs not covered in the above groups. (**'Entertainment_Misc_Program’)**

One of the biggest challenges, we were confronted was due to non-availability of authentic datasets of news channels video streams especially of Pakistani News Channels. To resolve this problem first of all we had to gather dataset for each class/ group by collecting publicly available videos on different websites. Most of the desired videos for each class have been downloaded from YouTube, particular News Channel’s websites[9]–[17] and numerous other websites. We then processed the huge video dataset and prepared the annotated trainable data which included images in each class for training of different neural networks for evaluation/ experimentation purpose. To carry out experimentation on different neural networks, we first modified these networks for the desired number of classes, which are five in our case. Thereafter, we finetuned and trained some of the networks initially on small dataset for desired stream classification. After getting encouraging results, we then trained the number of neural networks by finetuning on completely prepared annotated dataset for stream classification.

By using this technique, we can classify the live or recorded news channel stream into desired classes by suitably modifying the program. The major advantage of this technique is an easy classification of subject video on single image frame basis. And then we averaged out the results of image frames by applying dynamic time domain window on subject video for getting the desired results. We performed a lot of experimentation on multiple networks using our prepared dataset and some of the trained networks models gave very promising results. Now we shall give a brief introduction to News Channel video stream classification, its importance and the potential approach being used in our study.

1.1 News Channel Video Stream Classification and Its Importance

Video/ video stream classification has been explained and defined in various ways in different papers and websites. At some places it is considered as a method of generating labels relevant to the frames in the video and also links the features and annotations of different frames/ images in the given video[18], or discovering what a video is showing[19]. In simple words we can say that it is a process of segregating a given video stream into desired or target labels using ML/ DL techniques for different purposes like substance archiving or video scene interpretation[20] etc. A lot of work has been carried out on image processing, image classification and object detection using many different techniques of ML and DL. With the advent of high-speed processors and graphics processing units (GPU), the processing of large convolution networks/ complex architectures in DL has become feasible. Because of the recent advancements in deep learning, a number of latest architectures have been developed which are producing encouraging results on classification of images and small videos. After going through the literature available, it has been observed that there is not much work done on classification of large videos and especially the News Channels live video stream due to mainly non availability of authentic usable dataset and the high computation resources. It has been realized that by creating a suitable dataset and developing an architecture using deep learning techniques, a reasonable solution can be made for classification of the live stream from TV channels. Therefore, a need has been ascertained to experiment available latest DL convolution networks/ architectures for classification of News Channels live video stream. Various Pakistani News Channels broadcast was analyzed in detail for deciding to divide the stream into different classes. It was found that complete live video stream being telecast on various Pakistani News Channels can be divided into five approximately distinguished classes which includes '**Advertisements', 'News', 'Talkshow_News', 'Entertainment_Misc_Program'** and **'Sports'**. The architecture so developed is conceived to be able to classify the live stream of Pakistani News Channels into desired five major classes. It was a huge task to gather the relevant videos of each class from the live stream or annotate the stream to get the trainable data. Moreover, lot of time was wasted to arrange suitable computation resources for experimentation and training of models thereafter.

However, reasonable resources were managed to train the complete data on bit reduced scale and results remained extremely encouraging and satisfactory.

Video classification in general and classification of live stream from News Channels like Pakistani News Channels in particular, has many usages in different fields like media, advertisement agencies, related industries using the developed models in particular scenario. Few usages can be briefly explained for establishing its importance. There may be a requirement of some industry or department like sensor board to scrutinize and keep an eye on the content being telecasted on entire media. So that wrong or banned content may be removed or stopped thereafter from social media. It has been learnt that most of the work is being done manually by these agencies. Similarly, advertisement agencies, almost every department and big industry is spending a huge amount on their advertisement on TV Channels. As we know there are hundreds of channels on which ads are being telecasted and it is very difficult to keep a continuous check on every channel round the clock to identify how many times and for which duration advertisements were broadcasted. Also, the models can be employed with suitable modification, in manufacturing industries to keep a check on various processes and identify malfunctioning by processing the live video stream which may be overlooked while doing it manually. Moreover, even the news channels' quality control sections/ departments may need to audit the live stream for different purposes.

At present, most of the media channels, departments and industry are using manual ways involving huge infrastructure and manpower. Such tasks are very difficult to ensure accuracy round the clock due to human factor. Therefore, many chances are there to get wrong results/ feedback due to lack of vigilance or overlooking the continuous video stream by the staff. The result is less accurate data even after spending a lot of money. To settle such issues, there is a need to develop such systems which carry out all these above-mentioned tasks automatically. Thanks to latest DL techniques/ developed architectures, which can be used to automate the actions being done manually. This will result into less requirement for human staff and better accuracy in outputs/ decisions. In the suggested technique, classification of live video stream is done by classifying each image/ frame of live stream by utilizing the trained DL architectures/ models on the collected data meeting our need for desired number of classes. Classified images/frames are then passed through a timed window for floating average to get the output, resulting in the continuous classification of video stream based on averaged probability. In this way, we can differentiate between various classes present in the live video stream of News Channel on the basis of spatial and temporal information of each class. Moreover, the output can be used for further sub classification or analysis purposes.

As discussed earlier, live stream contains different classes like ‘Advertisements’, 'News', 'Talkshow_News', 'Entertainment_Misc_Program’ and 'Sports', which are televised as per their timeline and in most of the cases each program is shown in segments. Clear demarcation of these classes is sometimes very difficult as many number of frames in different classes may be similar or even blank having very less information, sufficient to put it into some specific class. Also, there are different patterns on each News Channel to broadcast any specific program. News can be displayed either by a single news caster or multiple news casters, Talk Shows can be having one guest or multiple guests, entertainment programs are shown in the form of drama, movie or some other pattern as well and so on. Advertisements are also shown in multiple patterns. It is important to highlight that each program which is aired in segments is filled in between by the ads. Also, ads are shown before and after some specific program as well in different protocols. Moreover, these programs are telecasted as per time schedule but sometime due to important events, breaking news etc. programs are disconnected, delayed and even discontinued. A rough pattern of a News Channel live video stream broadcast can be shown in the figure below: -

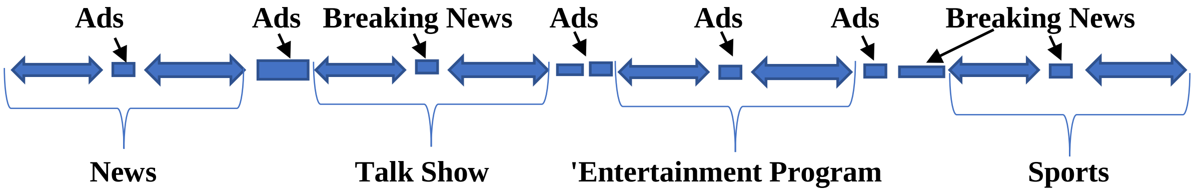


Figure 1.1.1: News Channel Live Video Stream Structure Example

Each class or group of videos contains images or frames relevant to that class and if analyzed each class images they comprise of a specific group of features. These features usually differ from each other based on spatial features. However, some features may look similar in different classes. These features are very important and are used for training ML and DL models. After selecting the model, it is important to find, select and prepare suitable features. In ML features are prepared manually and while in DL these are found and prepared by convolutional networks automatically themselves. Each set of features should be unique for the relevant class, features may be picked from frames on the basis of popular personality, some specific anchor person, brand, logo, combination of specific no of persons, politicians, instruments, writing words etc. Now we will see some samples from each class one by one. Few samples from Advertisements stream are as under: -

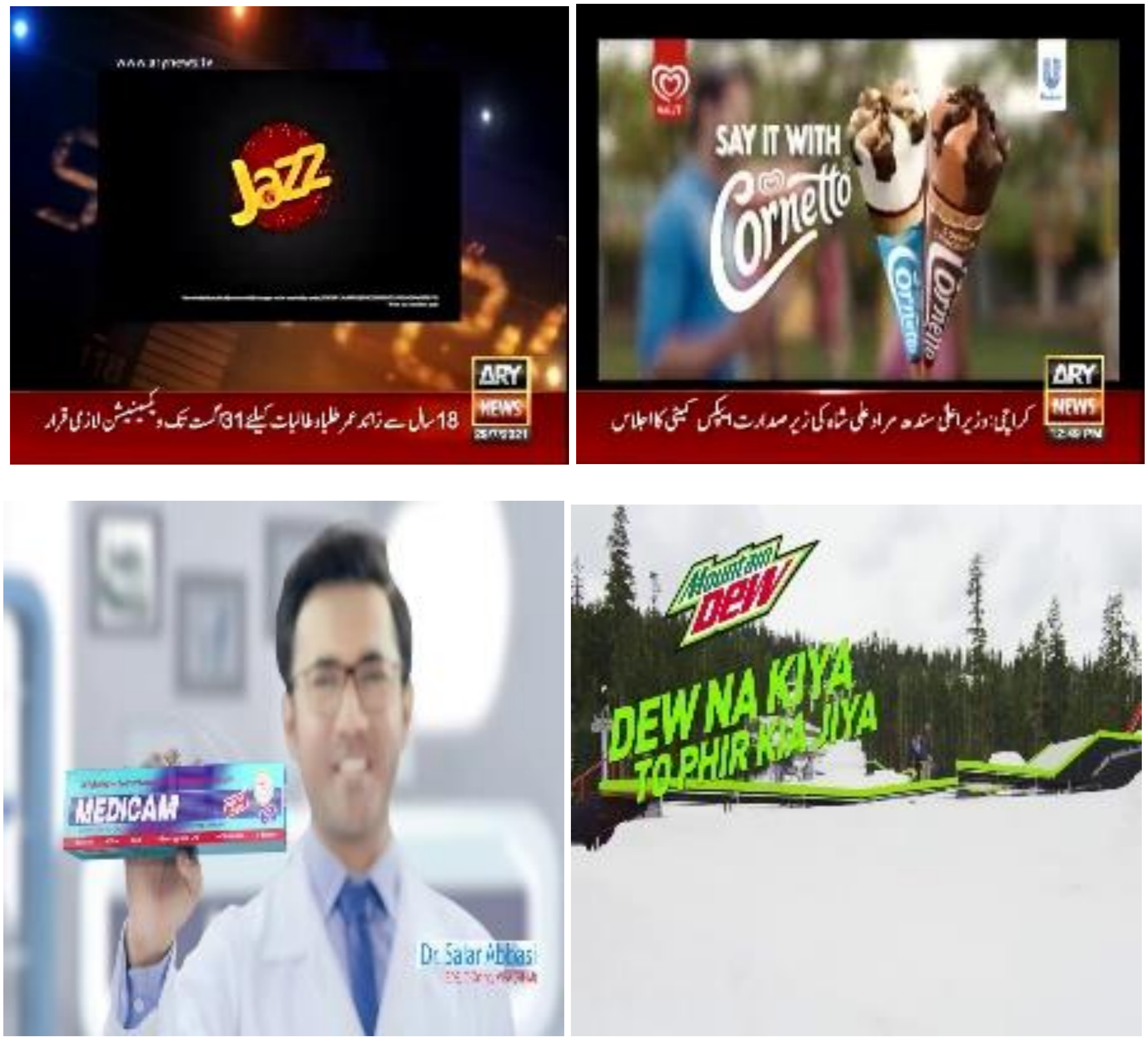

Jazz
ARY NEWS
18 سال سے زائد عمر طلباء وطالبات کیلئے 31 اگست تک ویکسینیشن لازمی قرار
SAY IT WITH Cornetto
کراچی: وزیراعلیٰ سندھ مراد علی شاہ کی زیر صدارت ایپکس کمیٹی کا اجلاس
ARY NEWS
12:49 PM
MEDICAM
Dr. Salar Abbasi
Mountain Dew
DEW NA KIYA TO PHIR KIA JIYA

Figure 1.1.2: Sample Frames from Advertisement Stream

These are only some samples from different advertisement videos. Different combinations of features can be identified, which may include different brands logo, scenes, style, color patterns etc. Usually advertisements are aired at start, end and in between different programs, news etc. and show comparatively different spatial appearances. Ads are essential for any channel to sustain their financial requirements. So is also important for different advertising agencies and industries for their popularity and earning confidence of the public to enhance their sales. So, advertisements class becomes an important class in classification of live video stream from any TV and Web channel and plays important role in evaluation/ audit of live stream by industries, advertisement agencies and TV channels.

Now we will see some of the samples from Entertainment program class as below: -

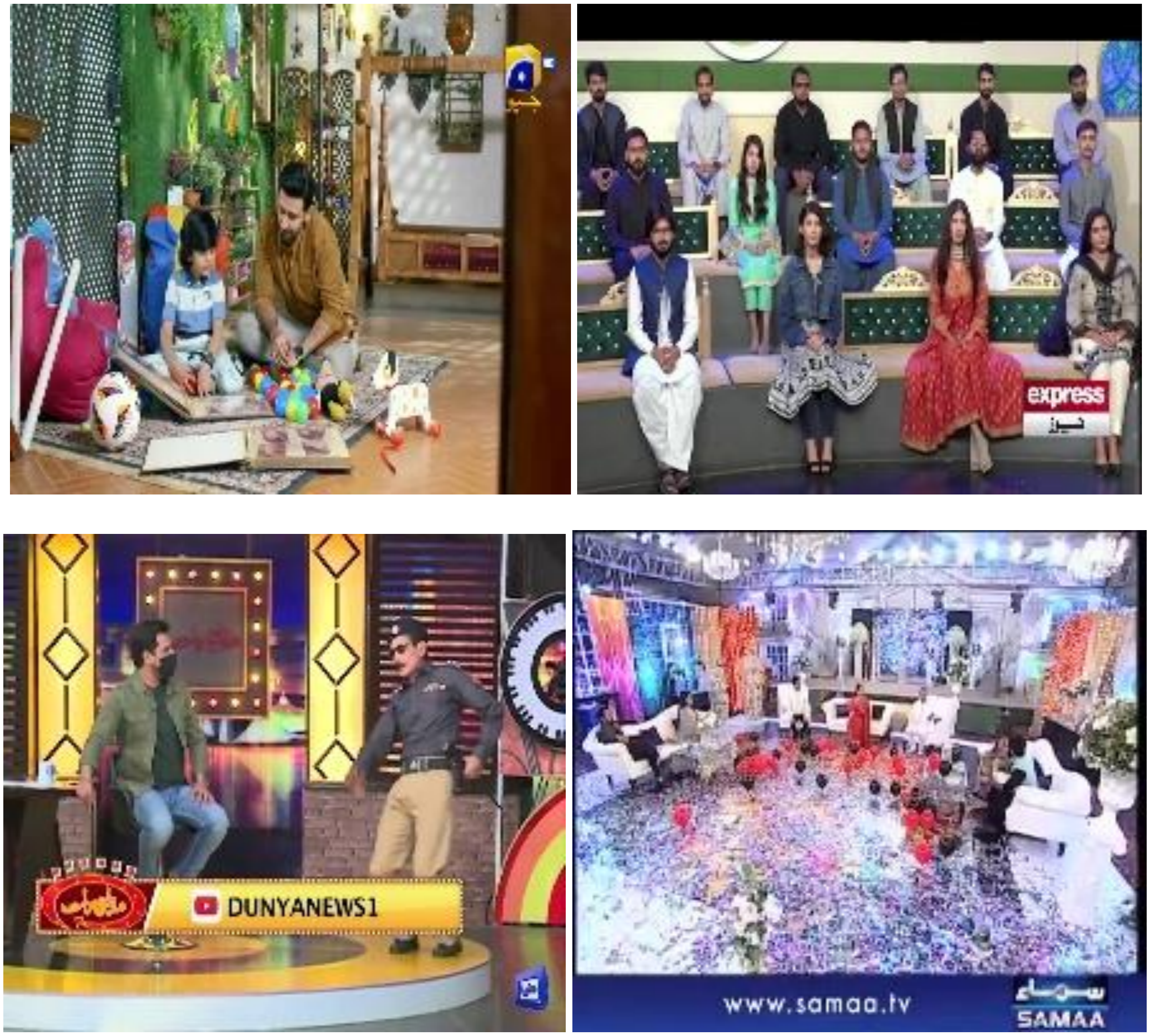
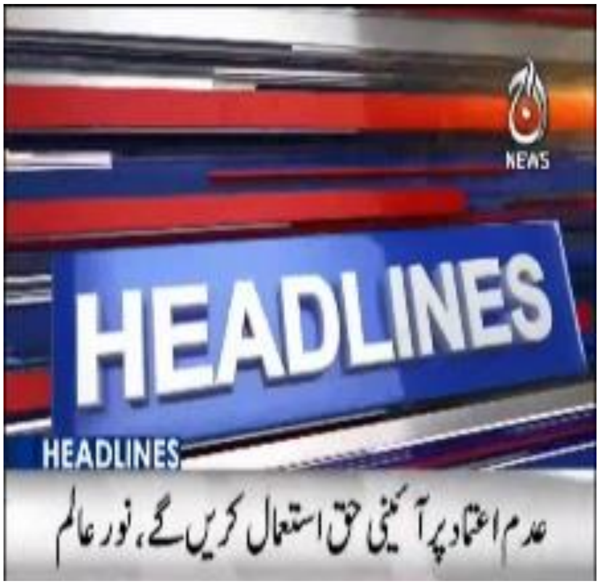
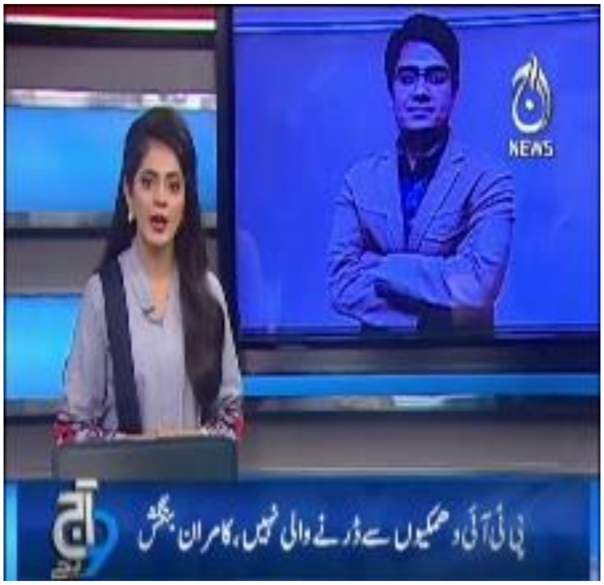


Figure 1.1.3: Sample Frames from Entertainment Programs’ Stream

If we compare these frames with the advertisement’s frames, one can immediately make out the differences and may label as entertainment program. Although some of the frames may be classified as other than the entertainment program, when a timed window is used to classify stream, it will be most likely be classified as pre the desired class. Frames can be identified based on their different color scheme, combinations, different names, actions, actors, musical instruments, appearance of sets etc.

Similarly, if we analyze News related frames considering their visual and spatial features, one can easily make out the big differences from the above mentioned two categories. Let’s see some of the following example samples from the News category frames for better understanding and creating difference from other classes: -





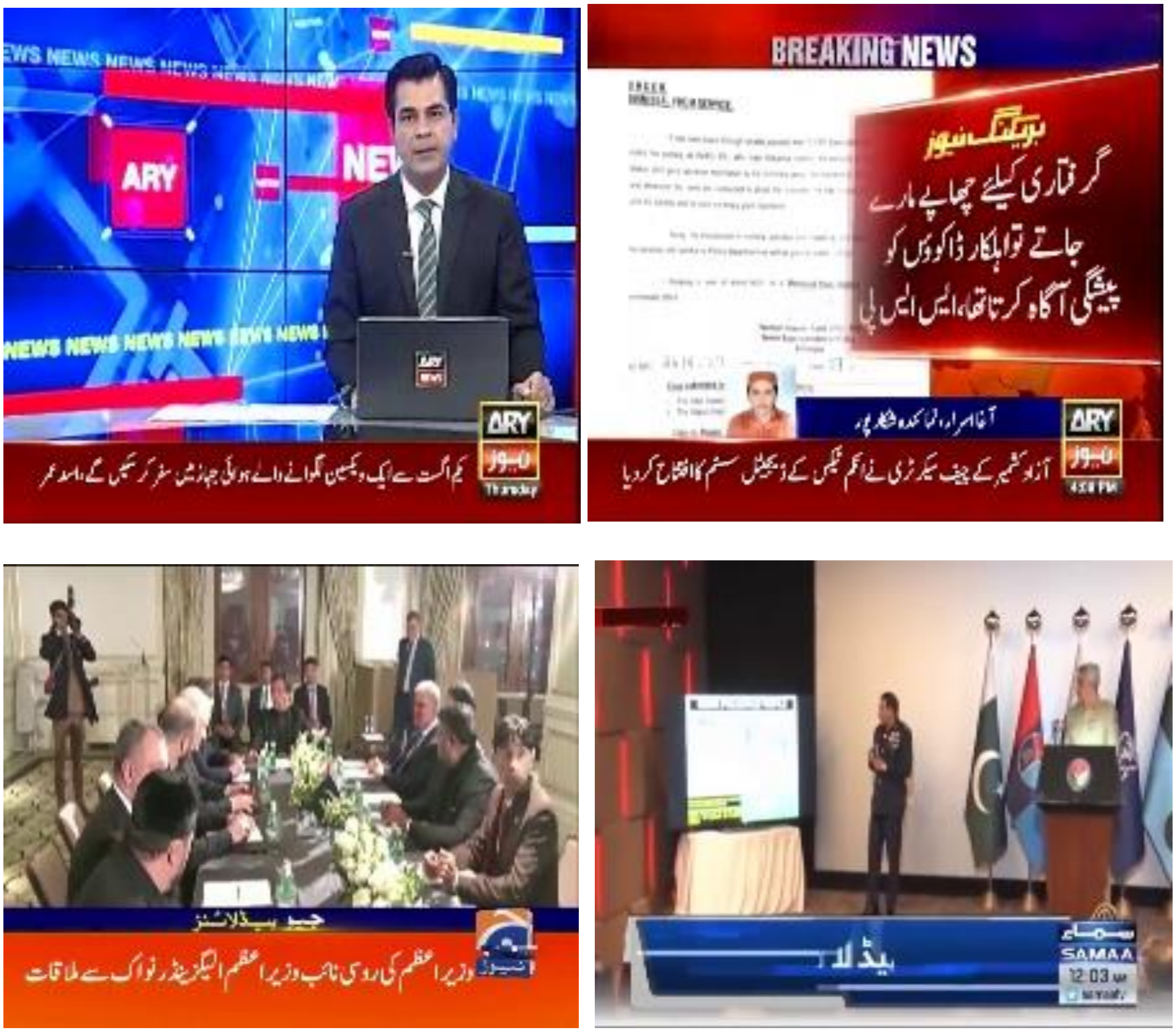

ARY
NEWS
یکم اگست سے ایک ویکسین لگوانے والے ہوائی جہاز میں سفر کر سکیں گے، اسد عمر
BREAKING NEWS
بریکنگ نیوز
گرفتاری کیلئے چھاپے مارے جاتے تو اہلکار ڈاکوؤں کو پیشگی آگاہ کرتا تھا، ایس ایس پی
آزاد کشمیر کے چیف سیکرٹری نے ایکم ٹیکس کے ڈیجیٹل سسٹم کا افتتاح کردیا
جیو ہیڈلائنز
وزیراعظم کی روسی نائب وزیراعظم الیگزینڈر نواک سے ملاقات
ہیڈلائنز
SAMAA
12:03

Figure 1.1.4: Sample Frames from News Stream

After seeing the above images from the stream, we can comparatively easily make out as News category due to many conspicuous features. We can see some anchors presenting news, selective words related to headline news, color scheme, special arrangements mostly related to newsroom, some known politician, other individuals appearing can be made out as a frame from News category. It is also possible that some of the news related frame may get resemblances with other classes due to many commonalities or some program may be shown during news due to some importance.

Now we will see some of the frames from Talk Shows video stream and here one can see many similarities with News class stream due to common features like common anchors coming both during news and talk shows, politicians, some news may be telecasted during program as reference. Yet there is quite a big difference between the frames of this class with the other classes. One can see that different logo, different written words/ names and difference in terms of contextual and circumstantial information than other class frames as under: -

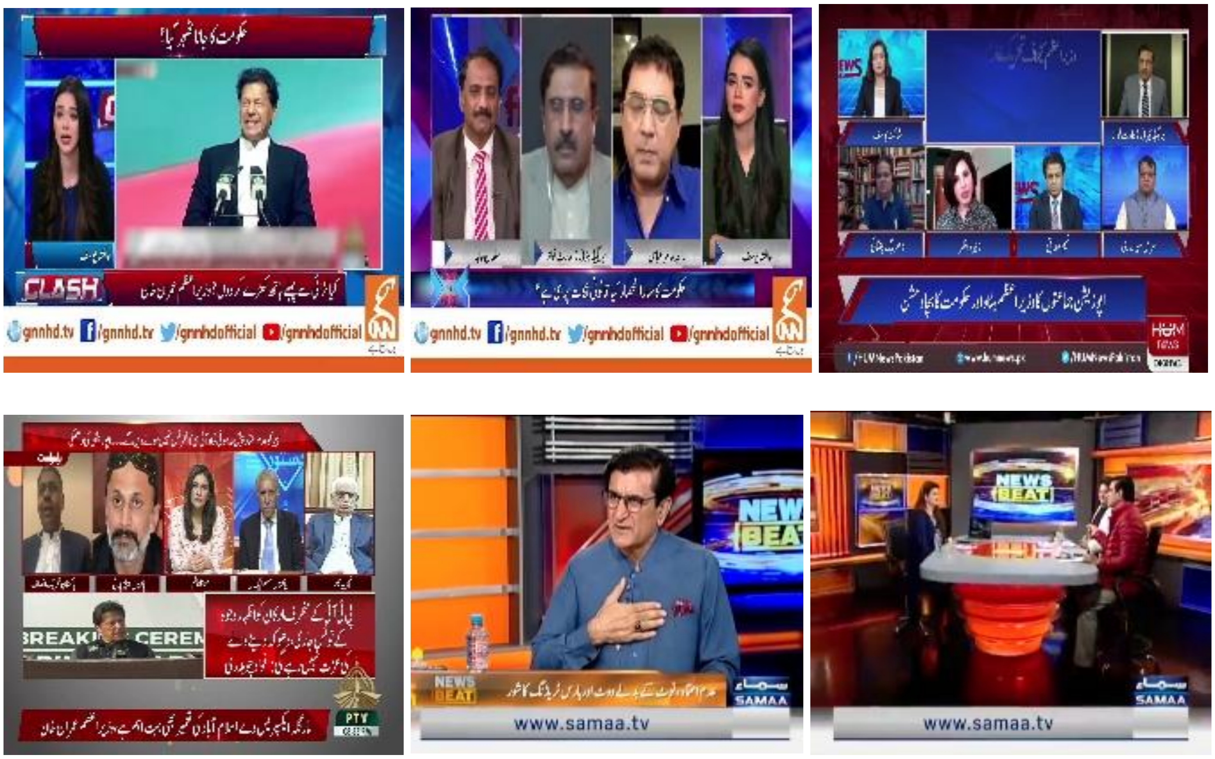

CLASH
gnnhd.tv /gnnhd.tv /gnnhdofficial /gnnhdofficial
gnnhd.tv /gnnhd.tv /gnnhdofficial /gnnhdofficial
HUM NEWS
PTV
NEWS BEAT
SAMAA
www.samaa.tv
NEWS BEAT
SAMAA
www.samaa.tv

Figure 1.1.5: Sample Frames from Talk Show Stream

The last class is related to the sports stream and relatively quite different than other classes. Though some similarities may be seen in the News, Talk Shows and even in entertainment programs as well. However, have many different features based on sports equipment, gear, grounds, specific words, phrases etc. as shown as under: -

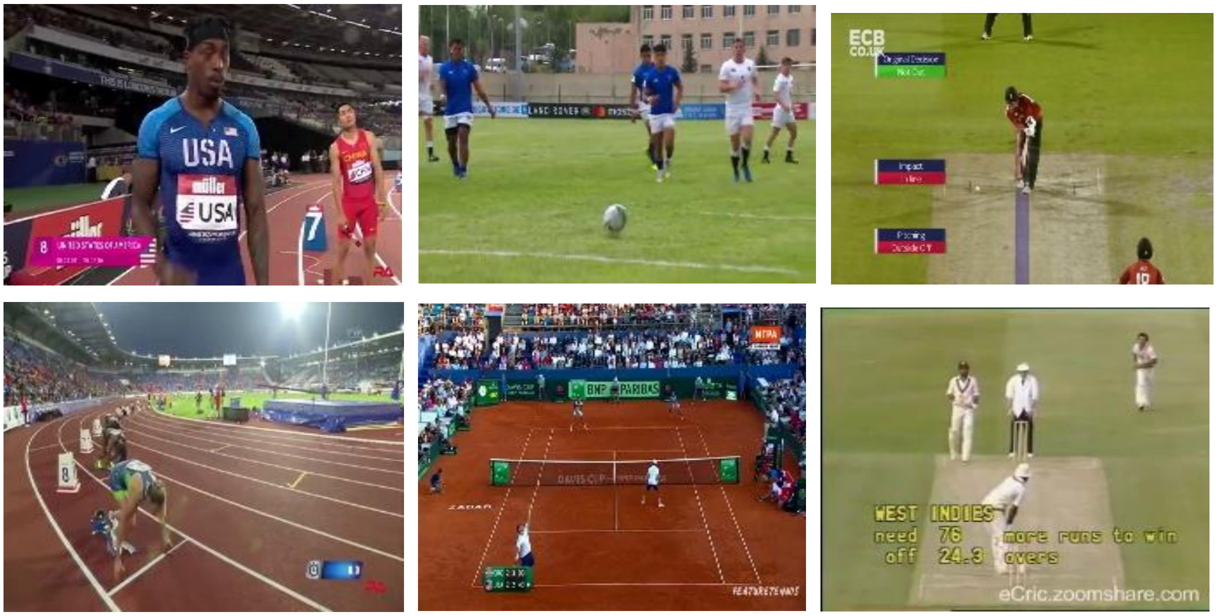


Figure 1.1.6: Sample Frames from Sports Stream

After going through all sample frames of all five classes, we can quite easily make out the differences between all categories including advertisements, news, talk shows, sports and entertainment programs on the basis of conspicuous spatial features mentioned above.

1.2 Potential Approach

We have already discussed that video stream classification is a complicated problem due to many reasons. Some of the major causes can be like, long videos of even a particular class may contain highly diverse types of features in each frame and even these features can get change drastically, non-availability of authentic and adequate quantum of data necessary for training of suitable network for desired classification and the non-availability of a well-established model for video stream classification. Although a lot of research has been carried out in image and small video clips classification and also in action recognition and segmentation. If action recognition algorithms and related datasets are analyzed, focus seems on specific actions within very short videos whereas, in our case the data in long videos is highly diverse, complicated and cannot be linked to any single action like the case in point of short video. Even in a single frame, one can see multiple actions, people, instruments and there may be abrupt changes after any scene. Videos length may be unpredictable in relation to any type of action or scene. This problem gets pronounced in case of live video streaming of News Channels, as many features, scene frames, characters, actions, visible movements of different classes may look alike and very similar. Many a times, several people may be sitting and making conversations, talking with each other like in case of a news programs, entertainment programs, talk shows and even in advertisements. This problem further gets complex when different channels are compared as each channel follows its own style making things quite difficult for creating a standard for classification. Moreover, some of the available models carry out classification of videos based on audio stream, some networks use the written scripts and words using OCR etc.

Video classification is very tricky, generally looking for some specific action, some specific items from which, model can take lead to specify the expected class. But due to the inherent problems in live stream, models may not find above mentioned features, actions etc. in number of frames for making some decision. Therefore, we need to utilize well established image classification techniques in such a way that a generalized model can be trained on the huge dataset for required classification of live stream videos from News Channels. So, we experimented with a number of latest DL convolutional networks with best results on image classification till to date, trained them on prepared dataset, which were able to classy approximately 95 to 99 % images correctly. This gave further confidence to apply these models on live video as video also consists of image frames. As models predicting most of the frames correctly, but a quite number of frames were being classified wrongly, which resulted into very unstable type of prediction like jitters/ flickers. Therefore, a specific dynamic time window was applied to average the continuous stream of video to predict each image quite stably after the initial half to one second delay at start of video. So, most of the jitters got resolved and reasonably a good stable result was achieved. Moreover, many other resources available with the video i.e., audio riding on the video, text, some information with the video in the form of metadata can also contribute to its classification. Although embedded audio on videos and text in most of the videos could also be used alongside the image classification networks, to avoid perceived complexity, in this study, we restricted ourselves to image classification networks only.

To make it possible, over 335 hours videos of desired classes were acquired from various official websites of Pakistani News Channels, YouTube and multiple other websites as well. This huge data was processed to get reasonable trainable dataset for training of selected models keeping in view the resources available, time required for training and sufficient quantity for making the model quite generalized. Different models like Alex Net, VGG, Resnet etc. were trained for a reasonable number of epochs, minimum ten and models requiring less time were trained for twenty epochs. While doing testing of video stream, prediction of each frame of video was carried out in sequence and these predictions then were passed through dynamic/ floating averaging which resulted in better and stable output.

We must acknowledge that we can resolve video stream classification problem to some degree, it can be further improved to create a generalized architecture by refining the dataset by choosing only the relevant frames having rich features to the class and creating a mechanism for dropping the irrelevant frames. So that, it can handle the diversity and complexity in the video streams data of different channels in time domain as well.

This study contains various terminologies and concepts which will be briefly explained and introduced in upcoming next two sections i.e., Background and Literature Review.

CHAPTER 2: BACKGROUND

In the first chapter of introduction, we have identified our objective of classifying the live stream from Pakistani TV News Channels into five classes i.e., Advertisements, News, Sports, Entertainment Program and Talk Show. We discussed potential approaches as well to get our objective. As we have come to know that video basically consists of a certain number of image frames arranged and presented with a certain speed called frames per second (FPS). Each video has its own characteristics[21] across whole video length and each image in the video follows these characteristics like resolution, size etc. We need to understand that the videos usually follow a sequence and some specific characteristics which help us to relate it with particular type like video related to cricket, hockey, tennis etc. These can be in the form of some chunks or small videos but in our case, as we are dealing with the continuous stream[2] of altogether different type or class of events being broad cast and no particular sequence can be focused upon. Moreover, if we are watching news, all the time different types of news on various topics are being shown which have quite confusing characteristics.

To classify live video stream, one may need to use various techniques either solely or in combination with to have better results. These techniques may focus on various properties of video i.e., image characteristics, image sequence, audio or sound riding on the video in relation to the images, written scripts, sentences or words and other conspicuous attributes. To get it done many methods can be used ranging from computer vision techniques to the artificial intelligence(AI)[22] modern methods[23] including ML, DL, OCR, NLP etc. All these modern techniques are driven by the use of AI. AI is considered to build machines which behave intelligently using especially prepared intelligent computer programs. Aim is to create a behavior of machine or a program similar to human intelligence specially having learning ability. AI is a diverse field mainly comprising machine learning, deep learning, reinforcement learning etc. Many industries are using these learning techniques for intelligent automation and specific requirements. Computer vision, natural language processing, natural language for communication, perception, robotics and many more use these AI techniques individually or in combination. The algorithms so prepared can be unsupervised or supervised or semi supervised, which are used to solve real world complicated problems[23] i.e., language translation, object detection, object recognition, speech recognition, text classification, information retrieval, information extraction, phrase structure grammar, semantic segmentation, image classification, image formation, robotic movements, driverless driving, medical robotics, brain computer interface and many more[23]–[26].

Image classification, object detection, semantic segmentation etc. can be carried out using different computer vision techniques especially the algorithms related to ML and DL perform better. Neural networks[24], [25] have outperformed in the field of ML and the most popular are convolutional neural networks have brought revolution in the field of supervised deep learning. One of the most difficult part in ML is extraction of useful features from the dataset which is done automatically in case of convolutional neural networks[5][27]–[30]. In case of image classification, there is no need of temporal information but in case of videos, time relation and sequence of images play an important role. As discussed earlier videos are structured in a specific sequence[31] resulting into clear demonstration of one type of video. So in video classification, feature extraction is needed to link with the time domain[32]. Therefore, we focus on feature extraction from video frames in relation to the time using convolutional neural networks(CNN) and then are classified by adding fully connected (FC) or recurrent neural networks i.e., LSTM etc[5], [19], [20],[33]. Now we will discuss some of the significant aspects related to AI like neural networks, feature extraction, activation and loss function, CNN etc. in ensuing sections. At the end we will also discuss important metrics to evaluate the performance of trained ANN.

2.1 Role of Neural Networks in Artificial Intelligence

In AI algorithms, neural networks play a significant role in getting the solution of any real-world problem. It is the revolutionary technique adopted in ML. Neural networks[34] have been inspired by the human brain. Human brains are jam packed with the neurons which have input and output wires called dendrites and axons respectively[23]. Neurons act like a computational unit which takes the input, carries out the computation task and sends out the output wires. In the same manner the neural network has been framed to work like the human brain. Human brain neuron and simple neural network is shown below: -

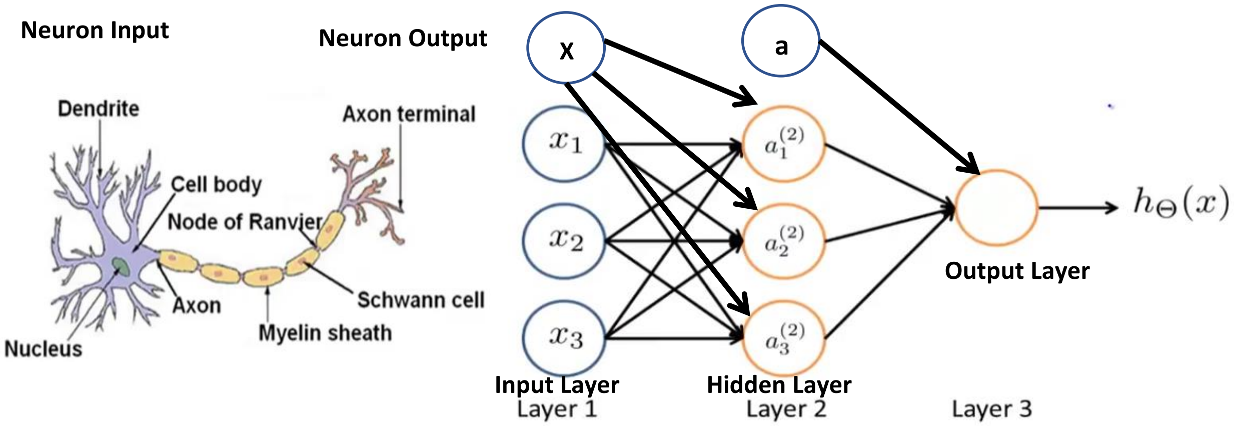


Figure 2.1.1: Human Brain Neuron Structure and Neural Network Representation

As in shown in the above figure, an artificial neural network (ANN) consists of neurons in the form of multiple number of layers called input, hidden and output layers[33], [35] and these neurons are connected with each other. A simple ANN will have information fragments similar to human brain cell. Each neuron gets one or more inputs, applies non-linearity to perform some computation[28] and gives an output. In the process some weights are multiplied with the input to have some output for application of non-linearity. The number of neurons in each cluster related to other clusters and number of layers are not fixed rather depending on the type of problem which are fixed either based on previous experience or worked out to use a suitable network with best output. The number of layers is considered as the depth of ANN. Input and output data is propagated forward and backward to get the desired output[36]. This network's structure and working is somehow like human brain cell and neuron structure. Input is sent to them as fragments of information similar to the dendrites which get information as input in human neuron. Thereafter, this information is processed in nucleus of neuron to get weighted result in the form of output, which is then sent to other neurons for further processing through Axon in brain cell[37]. This process continues in millions of neurons to get refined output.

The important part of any neural network is its peculiar ability to have a learning function called activation function which is used to train weights of each neuron of all layers[38]. It works out how the input information or data from each node is transmuted to give output contributing to the desired results. The method during training is adopted such that the input batch of samples or data is propagated forward, the output from each node of each layer is calculated by dot product of weights and the input. Result is passed through the activation function for application of nonlinearity[36], [39], or to limit the output which becomes input for the next layer. In this way output at last layer is got, which is then passed through loss function to establish a difference between the desired and achieved output. This loss function then is used to find the gradients of output with respect to the inputs. After that the network performs the backpropagation process of gradients to tune the neural weights of each node to achieve the output closer to the desired output or label of the example in case of supervised learning. We can say that the important part of this whole process is to update the all weights of whole ANN with the help of backpropagation of the calculated gradients[36], [40].

2.2 Importance of Features Extraction and Its Methods

Input to the neural networks is extremely important, as the quality of input in terms of its correctness and relevancy contributes much in determination of desired results. If the input is wrong, one cannot determine the correct output even if the model is designed very well. In ML the input is usually the major features extracted from the dataset on which basis model is trained and predicts the results. Therefore, features selection and extraction from the dataset gets prime importance. Either dataset is prepared manually or using various tools for good features extraction from complicated data[41]. These features are usually measurable quantities or are in the form of some information, properties or characteristics which show the data points in the available data. One can say that the extracted or given features are the basic building blocks of the available data, which form the trainable dataset. Moreover, these features may be dependent or independent of each other and some features may be combined as well to form new features[41].

Type of features solely depend on the problem's nature. As in our case the problem is related to video frames which are computer vision (CV) associated tasks. Therefore, the features would be required to be extracted from the available image dataset extracted from the videos[42]–[44]. The features extraction may include the traditional type of features in CV such as edge detection, scaling, normalization, filter application etc. From images information and measurable quantities are calculated which become the basis of prepared data. So the application of different filters can extract prominent edges, regions, color peculiarities and many more information[45], [46]. These features related to a single sample are then fed to ANN to resolve the confronted problem[47]. Speeded up robust feature (SURF)[48] a local feature detector and Scale invariant feature transform (SIFT) are also a method to extract features like key points from an image and matching. In case of natural language processing(NLP)[49]–[54] and speaker recognition[55]–[59] problems, features are related to the text and audio based data, which depend on lexical, vocabulary, grammatical morphology, word and sound structures. Here, lexical words are first of all tokenized (tokenized means to convert a word, sentence, character into a mathematical vector which can be used for subsequent computation) to get suitable features for ANN[60]. Tokenized process is done using usually the embeddings which are randomized vectors against the target data that can generated finely for preparing any data sample and can be propagated easily through neural networks[61]. Features processing and their normalization is very important and effective ways in computer vision and numerical jobs[62].

2.3 Features Extraction and Convolutional Neural Networks

Extraction of suitable features from the complicated data is a cumbersome task and it takes lot of time to have reasonable numbers of trainable features. Various methods are adopted for this purpose which include both manual and the use of automatic tools. But it is important that one may ignore important features and go for trivial types of features so compromising on the quality of output of ANN. These extracted features act as input data to the ANN and is propagated forward through the network for computation and application of activation function. These features are fetched from the raw sample data to extract some measurable points, quantities and values[63]. These values and quantities should be distinguished from others in the same dataset to have an effective role in the training of model. So, these can be considered as effective values. In the big dataset, identification of effective features is a major problem[28], [64]–[66]. Earlier techniques manual and even semiautomatic tools are quite complicated. Therefore, the solution lies in finding a fully automated features extraction method from the raw sample data.

The solution of this problem lies with the introduction of convolutional neural networks (CNN) which extract and refine features automatically[28], [66]. In CV terms, convolutions are sort of filter mapping, in simple words the dot product of filter kernel[67] and the corresponding image region. It may simply be a process of kernel or filter application on any data sample. It can be related to spatial operation in which each pixel in the output image is a function of all pixels in the surrounding region of that pixel[48]. Moreover, important linear spatial operators' correlation and convolution are closely related operations and all these can be mathematically and schematically as under: -

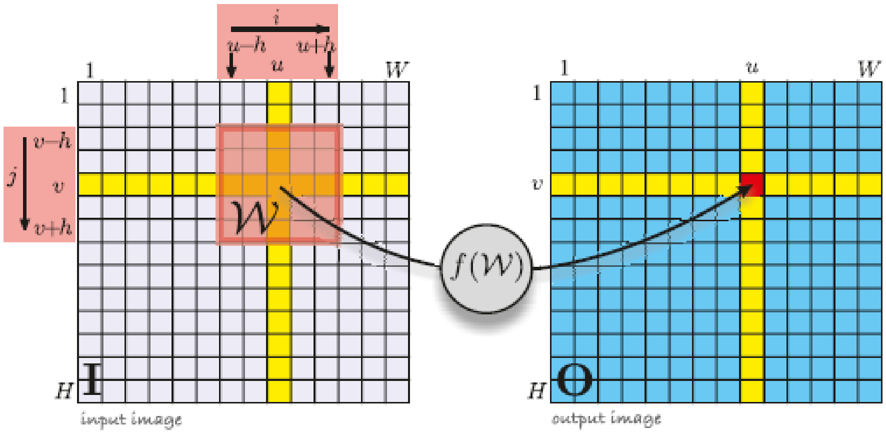


Figure 2.3.1: Spatial Image Processing Operations

$$O[u,v] = f\big(I[u+i, v+j]\big),\ \ \forall (i,j) \in \mathcal{W},\ \ \forall (u,v) \in I \qquad (2.3.1)$$

$$\mathbf{O} = \boldsymbol{K} \otimes \mathbf{I} \qquad (2.3.2)$$

$$\mathbf{O}[u,v] = \sum_{(i,j)\in \mathcal{W}} I[u+i, v+j]K[i,j],\ \forall (u,v) \in \mathbf{I} \qquad (2.3.3)$$

$$\mathbf{O}[u,v] = \sum_{(i,j)\in \mathcal{W}} I[u-i, v-j]K[i,j],\ \ \forall (u,v) \in \mathbf{I} \qquad (2.3.4)$$

Equation 2.3.1 shows the spatial operation as discussed above, equation 2.3.2 is a correlation operation considered as the weighted sum of pixels within the window and weights are depicted as K. Whereas in equation 2.3.3 ‘K’ kernel multiplied with the image corresponding pixels and equation 2.3.4 is convolution operation similar to correlation with small change.

In convolutional neural networks(CNN), convolutions analogous to the ANN are the spatial layers which are placed before neural networks[28], [36], [46], [64], [66], [67]. These convolutional layers play the role of features extractor tool from the images data. Specific filters called kernels[68] are defined depending on the construction of the network for a specific problem to achieve best results. These filters or kernels act as weight vectors and convolute over the input image frame to calculate the transformed output. Moreover, if the kernel size is such, which may reduce the dimension of output then padding is also resorted to in the form of some constant values. For sparse convolutional computations we can use stride of different steps to reduce the burden on resources[5], [27], [30]. So, get low dimension vector output. During the training of the network, input sample image data is propagated forward through the network and weighted output is calculated. Convolutional layers are combined with the pooling layers to form complete network to perform down sampling in spatial dimension of the input for reducing the number of parameters. Output from the convolutional layers is fed into neural networks which calculate the loss against the input data and their labels and the gradients are computed. Lastly these gradients are back propagated to fine tune the initial randomly selected weights or kernels at specified learning rate. In this way the network is trained to reduce the loss. [68]–[73].

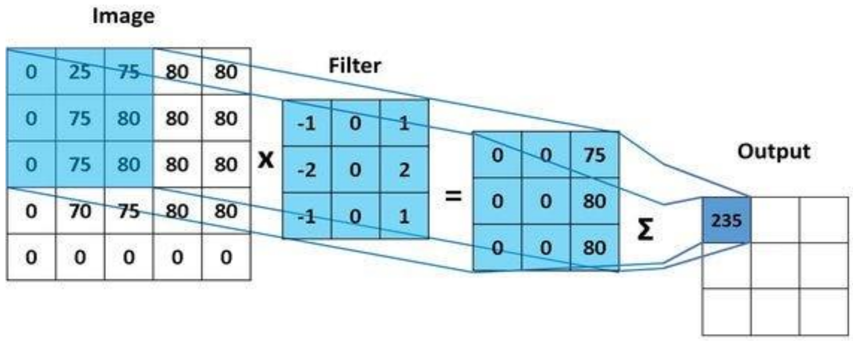


Figure 2.3.2: Convolutional Computation Process of Filter on An Image

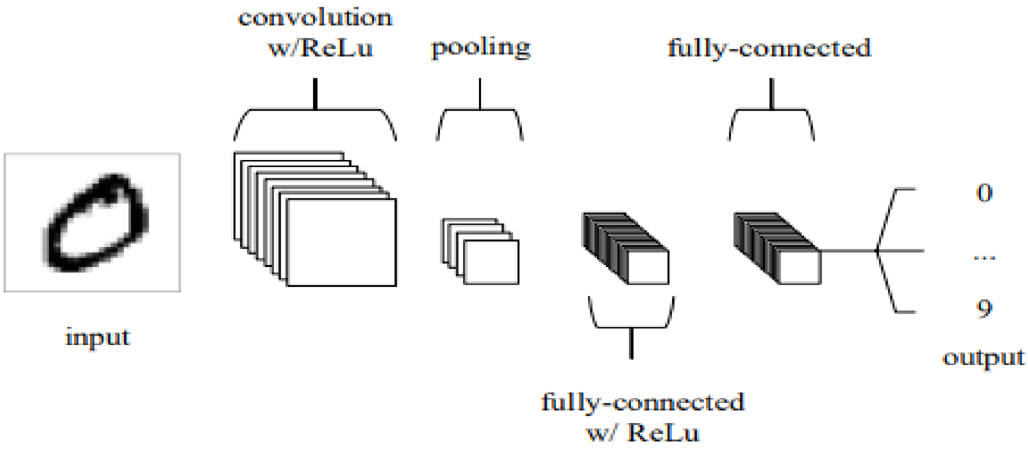
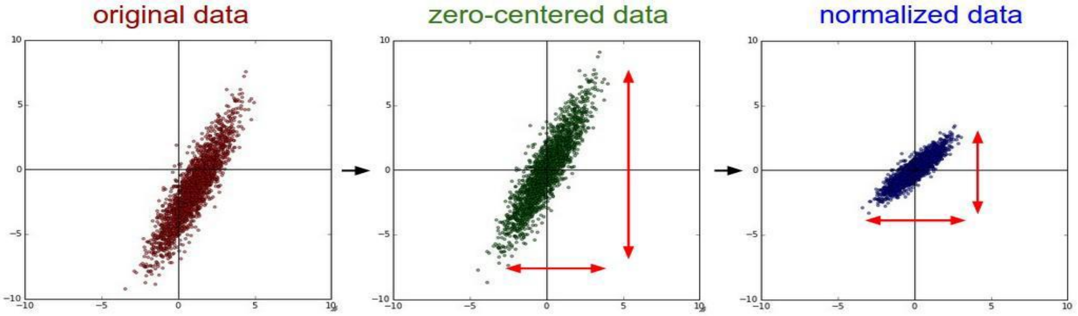


Figure 2.3.3: Simple CNN Architecture Comprising Five Layers

2.4 Activation Function

In the process of ANN and CNN heavy computations are involved and values may explode resulting in collapse of the whole training process of networks. During computation in ANN, input data and hidden layer weights or filters are multiplied through dot product to get the output[74] which sometimes may be exploded. Therefore, outputs are required to be regulated through the application of activation function. So, the activation function ensures the regulation and smooth flow of data through multiple CNN and ANN layers by limiting or transformations into numerical limits and application of non-linearity[75][76]. Therefore, their role in extracting and learning complicated patterns in neural networks has become very crucial. These also take care of other generic issues confronted to any neural network like vanishing gradients in large networks, no zero centered outputs, difficult differentiation and expensive computation to improve their performance. The role of zero centered data or output in neural networks is highly important as in data propagation decisions are based on zero centered outputs of weight layers[77]. Activation functions help to achieve desired results by limiting the output to a certain range. Also the input data processing for getting zero centered or normalized data contributes in reducing above narrated issues[78] and achieving good results.



of neural
netv mproved
resu [75] [79]
[81 :[84] and
Par

2.5 Loss Function

To resolve and find out output based on some input different equations or scientific methods are used to calculate the output. But there are many scenarios and problems which cannot be solved through established rules and methods. However, AI and its branches like ML, DL etc. provide a mechanism to predict quite satisfactory[85] and in some cases by using refined good networks, one can predict very close to the actual result. These ML and DL techniques work on a simple idea, in which one applies some model or hypothesis to find out the output and it is then compared with the actual results already calculated through experimentation. The difference between the two is calculated and thereafter methods of learning and optimization are used to reduce this gap which then produces results closer to the actual results in acceptable limits for good predictions[69][86]. This difference is brought in terms of gradient, which is backpropagated during training of the model, in this method network starts reducing the difference in predicted output and sample output. So, the weight of each neuron in the network is fine tuned to achieve desired results. This difference in outputs can also be called loss and the function named loss function devised to play its role in training of model provides the solution[86][87].

In simple terms we can say that loss function is a mathematical function which is formulated to find the relative change in predicted output and the ground truth or actual output against it. This loss may be termed as the cost linked with the output to improve the weights of ANN during training or learning process. So far, many loss functions have been developed being used in ML and DL depending on the confronted problem[88]. These functions help to calculate the loss or gradients. In the process of training this loss or differentiated values called gradients are backpropagated to compute the weight gradients for each neuron in the network. This rate of change or improvement in the weights learning can be preadjusted by some factor manually or some other mechanism and this rate is called learning rate[89][90]–[92].

There are many loss functions, but we will see few of the loss functions one by one. Log loss error also called cross entropy loss demonstrates a significant difference of probabilities against the actual result. It penalizes the wrong predicted probability and gives some reward to correct probabilities for quick training of the ANN[93]. Mean Square Error (MSE)[94] in which outliers produce large penalization of neural weights due to huge gradient computation[95]. Hinge loss penalizes the wrong probability predictions heavily, but it also penalizes the less confident predictions[96]. Mean Absolute Error (MAE)[97][98] which deals with the outliers data points by taking an average of samples. This loss has issue of data explosion resulting in higher average values than the other sample values[99]. Root Mean Square Error (RMSE) [98] squares the output got from difference of predicted output and actual results and then root is applied to reduce the outlier effect[100].

2.6 Important Metrics

Once we have done the training of an ANN, then there is a need to evaluate the model for checking its effectiveness. To evaluate the model, one need to find out the parameters on which model may be checked for its speed of training, accuracy, precision, error rate and how much it is prone to error in case data drift. In case of neural networks, evaluation is linked with the quality of training of model and thereafter the performance of model on unseen data or evaluation of training and test results on testing dataset[101]. The parameters used for evaluation of model are called metrics. There are numerous metrics, but we will discuss some of the important ones only. These metrics are employed to meet the desired requirements and type of problem. In case of discrete problems, accuracy can be used as metric, but if confronted with the continuous sort of problem, we will have to search for different metrics. Each metric may be suitable for multiple problems, and few may only be devised for some specific problems. Generally metrics are computed at the end of training of neural network and especially after the computation of loss function[102]. Prominent metrics include accuracy, precision, recall, F1 Score, Jaccard Score, Confusion Matrix, mean absolute error, mean squared error and area under cover etc.[102][103].

Accuracy is one of the main metrics which is usually used to identify the percentage of accurate predicted outputs after forward propagation in the neural network at every stage of training, validation and testing as compared to the labels of samples in the dataset[104]. Precision is another important metric used to find the quality of the trained model on the basis of how correctly the model has predicted the samples[105]. Meaning what is the fraction of true positives (TP) against total of true and false positives (FP). Whereas, recall is used to compute the quantity of a system or model, on the basis of applicable predicted samples[105],[106]. This means what fraction of correctly detected the TP against the total of TP and the false negative (FN). On the base of calculated precision and recall, we find the F1 Score[60], [105], [106], [107]. Which can be termed as the weighted average of precision and recall. In classification problem, F1 Score for each class label is calculated separately[105]. This is very important metric which takes good account of any unbalanced confronted data and provides very effective evaluation mechanism[105], [106]. Equations of precision, recall and F1 Score are given as under for better understanding: -

$$\text{Precision} = \frac{TP}{TP + FP} \qquad (2.6.1)$$

$$\text{Recall} = \frac{TP}{TP + FN} \qquad (2.6.2)$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad (2.6.3)$$

Confusion Matrix is one another important metric which shows the graph of the test results in which correct prediction of each class is shown along with the errors meaning by wrong predictions in each class like false negatives and false positives. It gives testing problems of classification in detail and shows the breakdown of the results generated by neural networks after testing of the complete dataset against each class[109]. Jaccard Score [109] is such a metric which determines in the dataset that how many sample sets are analogous. It is the measure of similarity of the samples in two sets and is shown usually in a range from 0 to 100%. If the value is closer to 100% it is considered to have better similarity in the two sets[110], [111].

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \qquad (2.6.4)$$

CHAPTER 3: LITERATURE REVIEW

Although handling of image classification problems using AI techniques has been researched by many researchers so far and lot of material is available. However, video stream classification and especially live video stream classification is considered relatively a new problem. Only a few papers are found on this subject. It is important to highlight that some work has been done on the classification of small videos on any one subject or identifying some specific actions i.e., sports events, gestures etc. In stream classification of live videos from TV channels consists of multiple actions and gestures of multiple individuals. Moreso, each frame of set of frames may contain multiple and dynamic situations which are challenging to link with some specific class. However, the available work on the different aspects of image and small video classifications including the gesture and action recognition can be helpful and set a start point for designing appropriate algorithms for the problem under consideration. Now we will discuss some of the relevant topics duly quoted from the notable papers in the following sections.

3.1 Image Classification

Identifying what an image shows may be considered as image classification[112]. Image classification is very important part in digital image analysis[113] and processing. If we simply imagine, the classification categorizes all the pixels of an image into any class like human, animals, insects, vehicles etc. Image classification is governed by various factors and formulation of a suitable image processing method is extremely important[114] for better accuracy. There are many methods used for classification of images including traditional techniques using manual, semi or fully automated features extraction methods[48], ML and DL supervised and unsupervised techniques. Some of the important traditional algorithms include K-Means, Maximum Likelihood, Minimum distance, Principle Components and Support Vector Machine (SVM)[115] etc. and various classification approaches i.e., knowledge per-pixel, subpixel, contextual and combination of multiple classifiers[114]. With the advent of artificial intelligence, ML and artificial neural networks (ANN), image classification has got revolutionized[114], [116] but the manual selection of useful features from the images remains quite a cumbersome job.

Whereas, over the last decade deep learning has solved the problem of manual selection of features with the help of convolutional neural networks (CNN)[117]. CNNs have been proved efficient for image classification. CNN are configured to have combinations of various number of CNN and pooling layers, which extract feauteres automatically and these features are refined in every subsequent layer, thereafter these features are fed into fully connnected layers for classification of images[114],[117]. CNN are not only useful for image classification but also play very important role in human activity recognition as well as the object detection, localization and semantic segmentation[119], [120]. However, these networks require a big high quality dataset for training. DL CNN has brought significant improvement in Image classification and is being used in many walks of life i.e., medical, satellite imagery, agricultural field, remote sensing etc. We know that the huge data being collected has placed a big requirement for its recognition and arrangement for efficent handling[118]. So to find out patterns from the image frames neural networks especially recurrent neural networks(RNN) have further improved the quality [122].

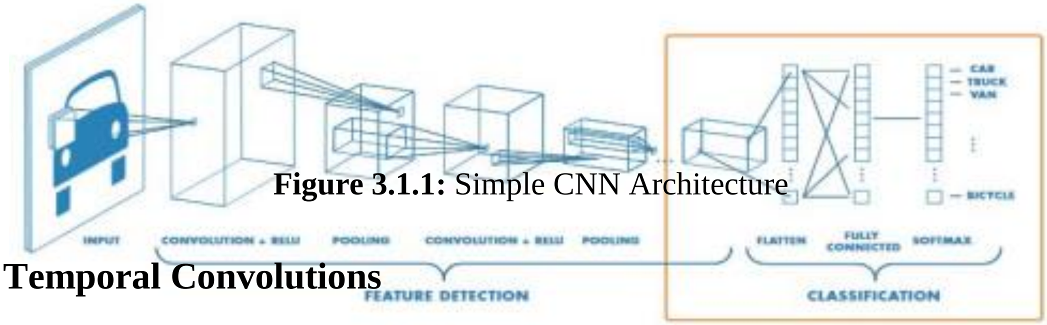


Figure 3.1.1: Simple CNN Architecture

3.2 Temporal Convolutions

As we have already discussed that CNNs are used to extract features from images and with the process of training, quality of features is enhanced for better prediction of images. CNNs are being employed extensively in a wide range of various applications related to computer vision tasks including image classification, recognition, segmentation and object detection etc. These networks are also giving good results in video classification tasks. It is important to highlight that simple CNNs do not perform well in each type of video classification tasks due to certain limitation related to videos. Videos as we know constitute of image frames arranged in a specific sequence and are kept in a certain number in one second called FPS[123]. If we further analyze, these frames are very much interconnected with each other, and we can say that these have variations on the bases of time[124]. So, if we run these frames at a certain speed against time (FPS), this gives a perception of continuous stream of video[5],[125], [124]. Therefore, reference of each frame in a video with respect to the time is very important and it has its own temporal features[126]. Such features are highly important while analyzing any action being performed[2],[8], [27]. To handle such issues, different techniques are being used worldwide and one of the popular is three dimensional CNN. These networks extract features from the video frames temporally as well as spatially[128]. It takes the input in the form of small batches of videos in five dimensions (batch, image number, channel, height width), each batch further contains batch of single frames just like a classification network for images and videos[129].

Action segmentation and recognition in the videos need two steps of computing low level features for every frame using CNN and then capturing high level temporal relationship using a classifier recurrent neural network (RNN)[130]. These steps are combined in one network called temporal convolutional neural network (TCN) which captures relationships at time scales. Also, in this method it first extracts spatiotemporal features locally from the video frames and secondly feeding them into [131]a temporal classifier that gains high level time-based patterns. This ability identifies human actions finely in the video which is very useful in robotics, surveillance, education and many more utilities[132].

Earlier Fisher vector representations and local motion features in specific motion boundary histogram were used[133] but for some years, different researchers have proposed various variants of such convolutional architectures[134], [135]. Some have used long term temporal convolutions for better accuracy in action recognition by employing basic representations like optical flow, video or image pixels and data construction[136]. Also suggested to capture temporal features from input video samples[137]. (Xiaoxia et al., 2019) [138] suggested to use 3 dimensional CNN for video stream classification and mentioned to replace earlier manual features extraction method like HOG[138] with CNN to include time domain features. Some recommended to use combination of three dimensional CNN with RNN and long short term memory networks (LSTM) for improving the results on action recognition tasks[139], [140].

3.3 Action Recognition in Videos

Action recognition is one of the very important computer vision tasks in which actions of one individual or group of people in videos or images are identified and understood[141]. It is very important to highlight that actions not only have the spatial correlation in two dimensional images but also have the attributes in temporal domain as well[142]. The major objective of this is to classify and do the categorization of the actions of human activity or even animal activities or some process may also be included in this category. Objective of action recognition is to identify action type and involves many activities including football, cricket, volleyball, basketball[143], badminton, tennis, running, walking, training, matches, speaking and hundreds of many other actions. Such mechanism will benefit to evaluate any player's performance, track and monitor his actions, movements and skills for preparing a statistical data. So, to make human action recognition (HAR) possible, we need to detect any activity in the video. The process involves detection of individual in the video, his location spatially as well as in time domain. For this HAR required to involve combination of various AI fields including computer vision, image processing, machine and deep learning[143].

As we have come to know that HAR in videos or images is very complex problem and many techniques have been evolved over the years by using various algorithms involving mainly CNN, LSTM and three dimensional CNN[144]. Some have used optical flow to verify the actions in the video accurately and suggested to feed the whole video as an object and applied a single frame as a spatial information with the help of CNNs[145]. For the purposes of extracting temporal and other features, multi frame optical flow and CNN can be used and lastly results are classified using class probabilities[69].

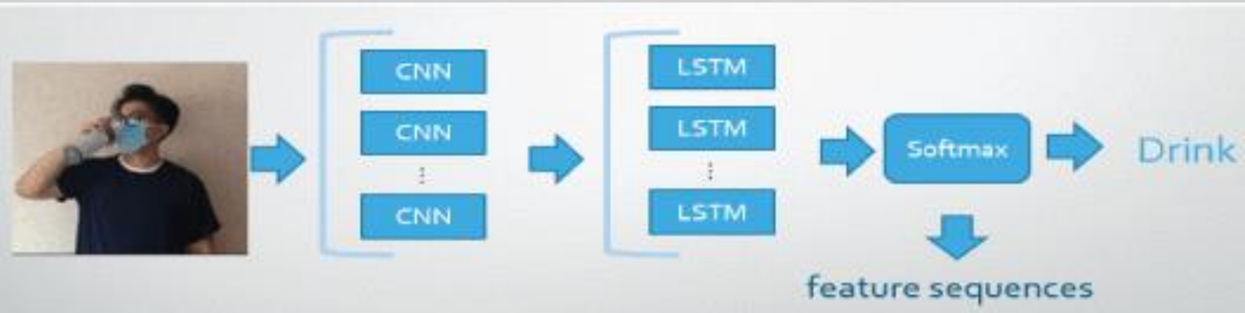


Figure 3.3.1: Simple Structure for HAR using CNN and LSTM

Many other techniques have been evolved for further improvement using deep ResNets for better gradient flow[146], [147] and some identified that the depth of neural networks involving greater numbers of parameters further makes training of the network difficult[148]. The video classification process is done in 3 stages i.e., 1: features extraction using CNN like ResNets for better results, 2: descriptor in which hidden layers of neural networks would have better comprehension for better description of the features for action recognition and finally 3: the classifier which predicts and makes clarity between different classes[149].

3.4 Video Segmentation

Video consists of images and images contain different objects. When important or desirable objects are found/ segregated for classification of the image and then the video is called image or video segmentation and is used for video stream classification. It facilitates in finding of useful information and features from the video. The process includes distributing video into segments or shots, camera angle, visual features or due to scene changes. Thereafter, segments are categorized based on content, attributes, duration and linkage with other parts for further analysis[150]. It further involves individual objects or events in specific scene and can be categorized into video object and semantic segmentation[151]. Video object segmentation's (VOS) main objective is to track specific objects in the video as required in case of surveillance, visual tracking and autonomous objects, vehicles or robots etc. Whereas video semantic segmentation (VSS) emphasizes [152]interpretation of the complete scene and is beneficial in augmented reality, creating virtual background in video conferencing and for summarization of videos. VSS is mostly employed for high precision environments like in case of robot sensing and autonomous driving by understanding everything happening in the surroundings[153]. For VSS and VOS different methods are used including automatic, semiautomatic or manual etc.[151].

In older techniques, manual feature extraction methods were used for each object and then these features were fed into specially formulated algorithms[154] according to the problem in hand i.e., motion boundaries, visual representations, graph models, visual attention mechanisms algorithms etc.[155], [156]. When objects are identified we can evaluate the theme or class or what is presented in the video[157]. When foreground and background of the video is segmented, it is treated as a binary classification problem which is further used for in detail analysis[158]. These problems are like in the case of images but when applied temporally, so similar methods can be used for videos as well. Now some deep learning convolutional models have been developed for end-to-end segmentation related problems which can handle multiple inputs including optical flow and automatic features extraction. But need extensive computational resources for training models. For this some used pretrained CNN models as well for training on the subject data[159], [160].

CHAPTER 4: DATASET PREPARATION

In ML and DL where formulation of a good architecture for developing a generalized model to be able to predict accurately is important, there the preparation of good dataset is of extremely significant as well for better performance of the trained model. We know that in artificial intelligence (AI), ML & DL, the main objective is to predict the outcome/ result on the basis of some example data using some hypothesis model as in our case are the convolutional architectures. We can illustrate it as under: -

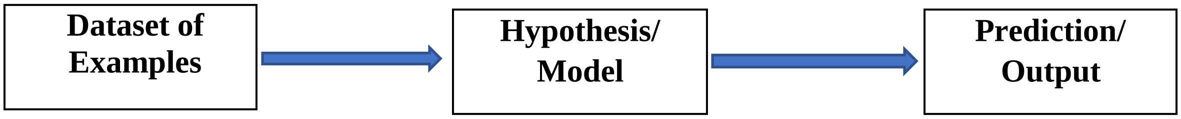


Figure 4.1: Machine Learning Prediction Model

Hence, we can say that the accuracy of predictions is directly dependent on the quality of dataset on which model is trained. Therefore, the dataset must be very accurate, precise, effective, and customized according to the problem and desired classes. Whereas, if the subject dataset has noise, drift and is not as per the problem articulation and relevant to the desired classes then not only the training of convolutional neural networks will be very difficult but also the test results will be quite noisy and unsatisfactory. In our study, as we are facing an unusual problem and the collection of relevant data remains a big challenge. Therefore, the accurate preparation of the dataset becomes very significant in articulation of video stream classification problems. In the next few sections, we will briefly discuss problem articulation, data collection, processing, challenges faced and encountered limitations.

4.1 Problem Articulation

The major focus of our study as we have already discussed above is to carry out classification of live video stream from various Pakistani News Channels. To achieve our objective, we need to do mainly two major tasks: one is to analyze the live video stream for preparation of good dataset, which we will be classifying and second one is either finding available good DL architecture or framing ourselves a suitable neural network. In this section, we will be focusing on analyzing the video stream and then processing it using the well-established techniques for preparing the trainable good dataset as per the requirements of our DL network/ architecture. We have analyzed live video streaming content being broadcasted by all major Pakistani News Channels and concluded that we can divide the content into five major categories. It is important to highlight that each category can be further sub categorized, which can be a sub problem and that can be undertaken with same technique being used in this study after preparing requisite dataset as per the number of sub classes and architecture.

As we know, most of the media, including TV channels are privately run and need financial support, which is mostly met through advertisements of various industries, companies etc. so each channel broadcasts advertisements. News channels’ prime objective as the name suggests is showing of news which is mostly covered in three ways normal news after some interval, breaking news which are usually similar in nature and in the form of talk shows which have some similarities with the news category but due to some distinct differences, we have kept it as a separate group. Some of the programs cover sports matches too. It was also seen that most of the Pakistani News Channels show various entertainment programs like stage shows, movies, dramas etc. other than the news as well. Therefore, we have grouped the Pakistani News Channels’ video stream into following five classes which we will be focusing on for preparation of dataset and thereafter for training of the model: -

- **Advertisements**
- **News** including breaking news etc. (**'News'**)
- **Talk shows**; some may mix up with some sort of **news** of similar pattern. (**'Talkshow_News')**
- Sports or programs which major portion include sport event. (**'Sports')**

- Entertainment programs or some other programs not covered in above groups. **('Entertainment_Misc_Program')**

After thorough analysis, we **tried to search for some authentic publicly available datasets** which can be used for training of our models. But we could not find any such dataset. Therefore, we decided to prepare our own dataset and use already collected a small data of advertisements video by our class **RIME 19** for DL class project. For this some of the news channels were selected keeping in view to inculcate diversity and generality in the collected data. It was further realized that downloading/ recording live stream and then manual annotation of the stream is extremely time taking task and expensive, therefore, we decided to download publicly available videos according to the decided groups/ classes. Thereafter, we shortlisted twelve news channels including **'Aaj News'**[17]**, 'Bol News'**[161]**, 'ARY News'**[11]**, 'Dunya News'**[16]**, 'Express News'**[15]**, 'Geo News'**[12]**, 'GNN'**[10]**, 'Hum News'**[162][163]**, 'PTV News'**[164][165]**, 'Samaa News'**[14]**, '92 News'**[166][167]**, '24 News TV'**[13] for downloading most of the publicly available videos on their websites and **YouTube**[9]**.** Sports related videos were recorded from some news channels and **YouTube**[9] and some other miscellaneous websites. After recording and downloading, we were able to gather approximately 335 hours of videos of different lengths.

Then these videos were processed for extraction of frames/ images of desired **size 3 x 224 x 224** and different qualities **40%**, **50%, 60%, 70%, 80%** and **100% (original)** for experimentation. Earlier more than 2.5 million frames with different qualities were extracted for each quality. But after initial experimentation it was realized that we need exceptionally heavy computational resources and lot of time for undertaking training of the models with high quality images than the low-quality images with very minute compromise on accuracy. Therefore, only images/ frames with **40%** and **100% (original)** quality were used in final training of models. Moreover, the training time required for high quality images was manifold. Therefore, two datasets of approximately 0.622 million frames with image quality of 40 percent and original quality were finalized. In this way we converted the problem into image classification. Thereafter, we selected ten different DL convolutional neural networks with best performance on image classification and trained our models. For testing of videos, we developed algorithm by applying the continuous sequential video frames for desired classification using dynamic adjustable time window. Results through our technique remained quite encouraging. Same has been depicted in the figure below: -

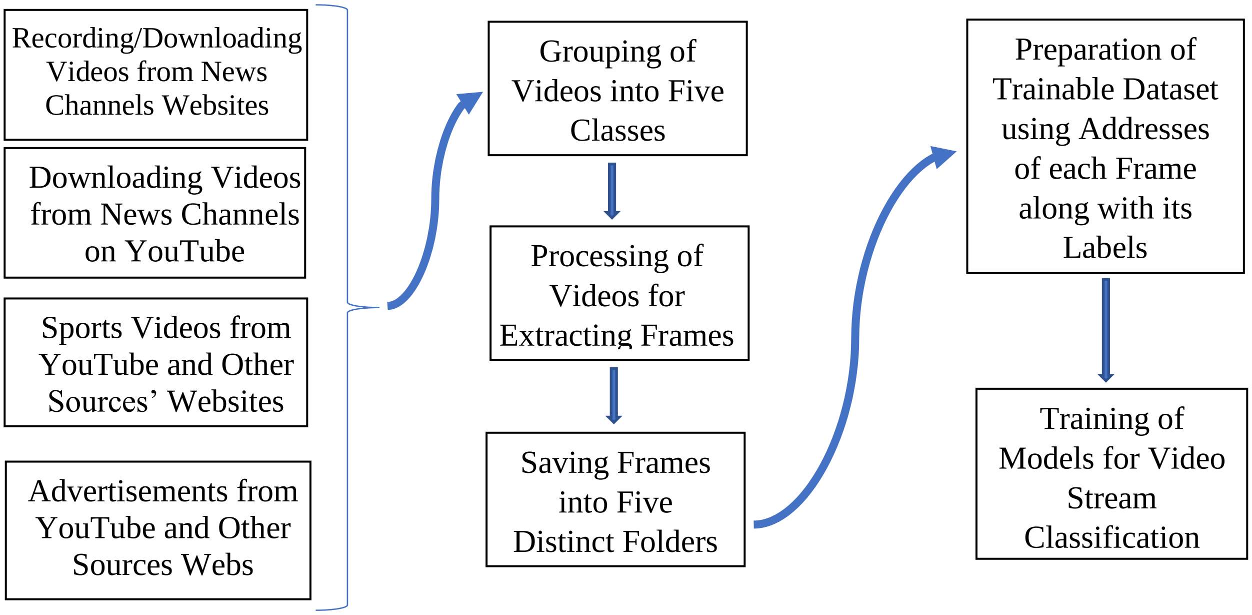

Recording/Downloading Videos from News Channels Websites
Downloading Videos from News Channels on YouTube
Sports Videos from YouTube and Other Sources' Websites
Advertisements from YouTube and Other Sources Webs
Grouping of Videos into Five Classes
Processing of Videos for Extracting Frames
Saving Frames into Five Distinct Folders
Preparation of Trainable Dataset using Addresses of each Frame along with its Labels
Training of Models for Video Stream Classification

Figure 4.1.1: Collection of Videos and Preparation of Dataset Pipeline for Stream Classification

4.2 Dataset Analysis and Properties

In the previous section, we discussed in detail the problem articulation keeping in view the evaluation of video streams of various Pakistani News TV Channels and grouped various programs being broadcast into five classes. To create a suitable dataset for training of neural networks, we needed to have sufficient quantity of videos related to each class. For this we downloaded/ recorded videos stream in three different methods. In the first case we recorded live stream from various news channels of different duration, having continuous stream of various classes i.e., news, talk shows, advertisements and entertainment programs. But this scenario needed manual identification of each portion related to a particular class and annotating it in the shape of separate Excel file. This proved very tedious work and resultantly resorted to other methods. In the other two methods videos particular to each class were downloaded from Pakistani TV News Channels Websites and various other websites like YouTube etc. These videos were saved in a specific folder as per the class relation. Annotation of these videos was carried out as per the name of folder. Now we will further analyze the dataset and processing separately in all three cases.

4.2.1 Continuous Live Stream Processing and Annotation

First effort we made was to record continuous live video stream from different Pakistani TV News Channels and then annotate the recorded videos. But this effort proved very expensive in terms of time and effort. Although later we switched our data collection technique, we utilized this collected data for which we had prepared the annotated files as well with the help of different groups of RIME-19 working on the related studies. The detail of recordings form six different channels is given below in the table: -

Serial	Channel	Video Clip Duration (Hours: Minutes: Seconds)	Resolution	Size on Disc (MBs)
1	ARY News	6: 15: 15	854 x 480	2120
2	Bol News	5: 29: 37	854 x 480	2353
3	Dunya News	5: 19: 02	854 x 480	2200
4	Express News	5: 29: 01	854 x 480	1831
5	Geo News	17: 27: 36	854 x 480	6032
6	92 News	4: 45: 47	854 x 480	2033
Total		**44: 46: 18**		**16,569**

Table 4.2.1-1: Continuous Video Streams Collected from Live News Channels

These continuous video streams mostly consist of advertisements, news, talk shows and a few entertainment programs as well. The problem being studied is related to supervised learning and as we are resolving it using image classification techniques through already trained good DL convolutional networks on ImageNet available in PyTorch Library. Therefore, we need to convert videos into a trainable dataset having images and their related labels. So, we will have to annotate the videos to get the related labels for each frame of video. It is important to note that each video stream recording contains sections of multiple advertisements, news, talk shows, sports programs and entertainment programs and the construction of video stream is given as under: -

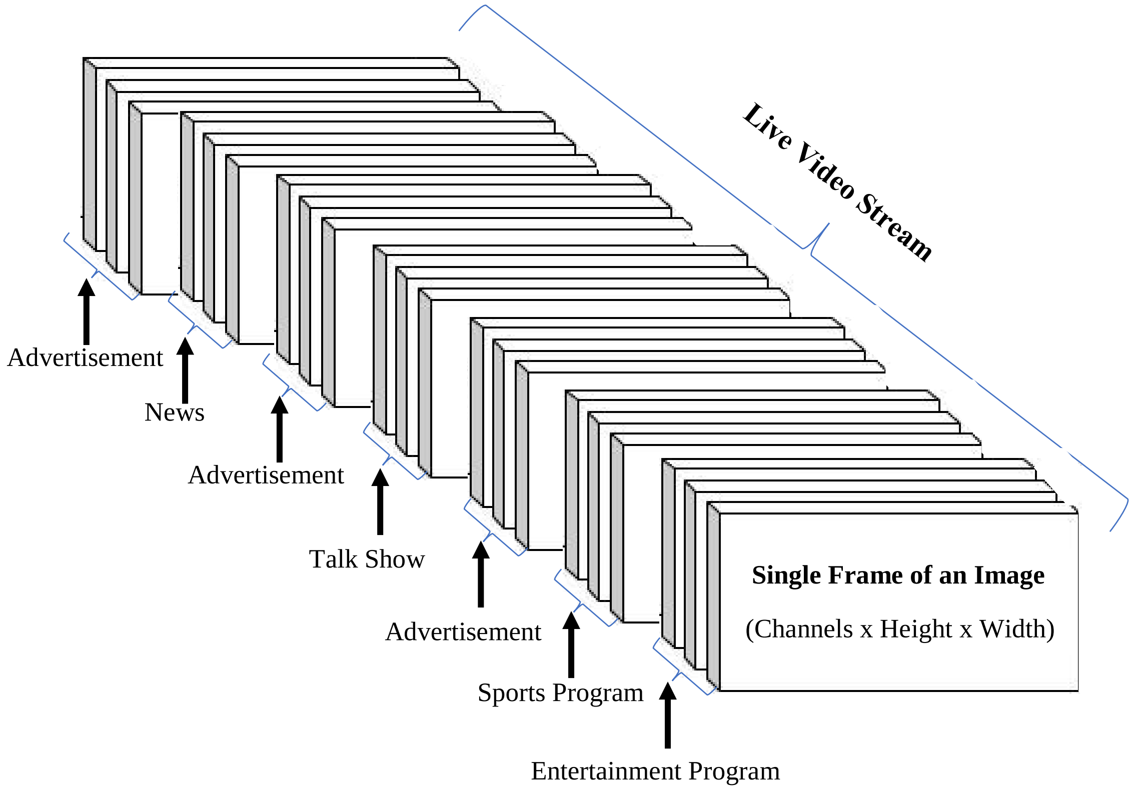


Figure 4.2.1.2: Construction of Video Stream

As discussed above, the video stream consists of different sections which keep repeating and changing after a specific time. We also know that video stream contains a continuous stream of frames, and each type of video has a specific number of frames per second (FPS). In our case most of the videos are of 30 FPS but may vary on a case-to-case basis i.e. 24 FPS, 26 FPS etc. It means that we can identify each specific frame in the video by simply counting the number of desired frames. So, we can address any scene in the video. Hence the position of frame can be related to the time domain of video. We can say that after specific time in hours, minutes, seconds and then the nth number of frames can be reached, identified and utilized for specific purpose. The purpose can also be to assign the relevant label. In this way we can recognize the start and end time of a particular program. By using this technique, we can quite easily extract frames class wise and save them in class related folder for creating trainable dataset for image classification. This process can be shown in following table: -

Sr #	Stream Type	Brand Name	Description	End Hr	End Min	End Sec	Ending Frame Cut
1	Talk Show	Express News	expresso	0	7	18	14
2	Advertisement	olx mall	app	0	7	33	10
3	Advertisement	jazz	jazzcash	0	7	48	8
4	News	express News	logo	1	9	51	17
5	Advertisement	olpers	milk	1	10	11	27
6	Sports Program	jazz	app	1	10	22	2
7	Advertisement	walls	cornetto	1	10	52	10
8	Entertainment	khabranak	Dunya	1	11	10	15
9	Talk Show	Express News	expresso	1	19	1	23
10	News	express News	News update	1	21	46	18
11	Advertisement	telenor	whatsapp	1	23	22	23
12	Advertisement	nestle milk	milk	1	23	42	20
13	Program	express News	expresso	2	34	44	11
14	Advertisement	tapal danedar	tea	2	36	6	16
15	Advertisement	national	rozana recipies	2	37	14	28
…….	…….	…….	…….	…….	…….	…….	…….
…….	…….	…….	…….	…….	…….	…….	…….

Table 4.2.1-3: Stream Classification Format for Frames Annotation

The table shows the complete picture of the recorded video clip from the live stream of news channels. Each section of class was identified by noting its start and ending frame. This data was stored in Excel file and was used for annotation of each frame of the recorded video. With the help of this file and video, a trainable dataset was prepared by saving images/ frames related to the class into relevant folder. For this we created a dataset by compromising on quality of images and their quantity, sufficient enough for training of a model to be able to give us a generalized result. The number of frames saved in the dataset related to each class folder from the recorded videos from live stream is given below: -

Serial #	Class	Frame Freq	Memory (MBs) 40% Quality	Memory (MBs) 100% Quality	Total No of Frames
1	Advertisement	1 FPS	455	1,140	54,062
2	News	1 FPS	756	1,910	72,283
3	Talk Shows	1 FPS	378	1,001	34,072
4	Sports	1 FPS	0	0	0
5	Entertainment	1 FPS	8	21	758
Total			**1,597**	**4,072**	**161,175**

Table 4.2.1-4: Detail of Dataset Prepared from Continuous Video Stream

4.2.2 Advertisement Video Clips Downloaded from Various Sources

Videos related to advertisements were gathered from various resources. Some of the dataset related to this class was segregated from the above explained continuous video stream and most of remaining advertisement clips were downloaded from YouTube and some from multiple websites. We downloaded approximately 1208 video clips of almost all different advertisements related to various brands, industries, educational institutions, electronics, clothings etc. All these videos are of different quality and resolution having an approximate size of 10.4 GB. All the videos were saved according to their related major field i.e., electronics, food, cars, etc. in separate folders and these folders were saved into master folder with the name advertisements. The Data Handler program was prepared to process these videos into trainable dataset as per required size of all frames i.e., 224 x 224. Frames were saved into folder with the name Advertisements and frame selection frequency was adjusted as per the compromised requirements. The detail of video clips and their extracted number of frames are given as under: -

Ser #	Advertisements' Class	No of Videos	Resolution	Video Clips Duration	Memory (MBs)	Total No of Frames
1	Academic Institutes	40	1280 x 720	00:56:52	707	68,809 1FPS
2	Appliances & Furniture	41	1280 x 720	00:53:07	430	
3	Automobile	40	1280 x 720	00:51:19	504	
4	Banks and Insurance	52	854 x 480	00:41:24	199	
5	Biscuits & Bakery	67	1280 x 720	00:53:27	464	
6	Cash Transfer	33	1280 x 720	00:24:14	203	
7	Clothing	59	1280 x 720	01:11:39	780	
8	Dairy	35	1280 x 720	00:21:57	224	
9	Dental Stuff	37	480 x 360	00:25:10	197	
10	Detergents & Softener	52	720 x 480	00:32:08	311	
11	Drinks	69	1280 x 720	00:46:49	458	

12	Edible Oil	44	320 x 240	00:57:15	485	
13	Fast Food	40	1280 x 720	00:22:46	224	
14	Food Seasoning	35	854 x 480	00:37:11	178	
15	Frozen Foods	56	1280 x 720	00:36:44	447	
16	Gadgets	45	1280 x 720	01:01:22	707	
17	Groceries	44	640 x 480	00:37:24	290	
18	Hair Care	56	480 x 360	00:37:35	210	
19	Household Items	48	1280 x 720	00:47:51	445	
20	Real Estate	40	1280 x 720	00:56:33	758	
21	Shoes	46	1280 x 720	00:44:13	364	
22	Snacks	23	1280 x 720	00:19:32	210	
23	Soap and Skincare	63	1280 x 720	00:49:12	487	
24	Telecommunication	46	1280 x 720	00:37:16	505	
25	Online Shopping	44	1280 x 720	00:39:29	264	
26	Misc	53	1280 x 720	01:00:08	609	
Total		**1208**		**19:06:47**	**10400**	**68,809**

Table 4.2.2-1: Detail of Advertisements Downloaded from Various Sources

4.2.3 Videos Downloaded from News Channels Websites and YouTube

It was realized after thorough analysis of various news tv channels websites that the videos related to advertisements have not been uploaded. Therefore, advertisement videos were downloaded from multiple sources like YouTube etc. In the case of the remaining four classes most of the videos were recorded/ downloaded from the news channels websites and YouTube. However, some of the sports videos were gathered from YouTube related to different sports to bring generality in the sports class. Resultantly we gathered 663 videos of approximate size of 108 GB. Videos are of different resolution, mostly with 1280 x 720 and quality. These videos were stored into each channel folder and these folders were then stored into their class specific folder. Thereafter, these videos were processed with a specially prepared program '**Data Preparation Video to Image**' program using Python language and some of the libraries. With this processing we prepared frames of desired size as per architecture input requirement i.e., 224 x 224 and with reduced quality of 40%. The detail of videos downloaded from various news channels and other sources has been shown in the following table: -

Serial	**Channels**	**News**		**Talk Shows**		**Entertainment**		**Sports**	
		Videos	**MBs**	**Videos**	**MBs**	**Videos**	**MBs**	**Videos**	**MBs**
1	Aaj News	16	1100	29	6500	9	2910	--	--
2	ARY News	17	1120	25	5210	25	6940	--	--
3	Dunya News	33	1970	24	4990	14	1810	--	--
4	Express News	31	1550	18	2180	13	2570	--	--

5	Geo News	57	2910	23	5290	21	4700	--	--
6	GNN	36	2150	27	5880	27	8130	--	--
7	Hum News	31	3280	24	7170	--	--	--	--
8	PTV	26	1100	18	5400	--	--	--	--
9	Samaa	34	1160	11	2420	10	1290	--	--
10	Misc/ YouTube	--	--	--	--	21	5400	43	2860
Total		**281**	**26300**	**199**	**45000**	**140**	**33700**	**43**	**2860**

Table 4.2.3-1: Detail of Videos Downloaded from News Channels

These videos have been further placed in as per their class in separate folders. The duration of videos class wise is given in the table below: -

Serial	**Videos Class**	**No of Videos**	**Duration (Hrs: Mins: Secs)**	**Resolution**	**Size on Disc (MBs)**
1	Advertisement	1208	19: 06: 50	1280 x 720	10400
2	News	281	57: 06: 45	1280 x 720	26300
3	Talk Shows	199	120: 35: 09	1280 x 720	45000
4	Sports	43	11: 23: 50	1280 x 720	2860
5	Entertainment	140	81: 58: 12	1280 x 720	33700
Total		**1871**	**290: 10: 46**	**--**	**118260**

Table 4.2.3-2: Duration of Downloaded Videos Class wise

These videos downloaded through different sources mentioned in above two sections were processed for extraction of frames of two qualities i.e., 40% and 100%. These frames were saved in relevant folder of class. As there was imbalance in quantity of videos and their length/ duration, therefore, we extracted frames from the videos in different ratio and frequency, so that we can have some sort of balanced dataset. In case of videos related to **Sports**, we extracted 2 frames from number of frames in one second, whereas in case of **News, Talk Shows and Entertainment**, one frame from number of frames in three seconds and one frame per second in case of Advertisements. By using this technique, we were able to prepare somewhat balanced dataset and detail of dataset class wise is shown as under: -

Serial #	**Class**	**Frame Freq**	**Memory (MBs) 40% Quality**	**Memory (MBs) 100% Quality**	**Total No of Frames**
1	Advertisement	1 FPS	479	1,290	68,809
2	News	1 FP3S	632	1,810	68,535
3	Talk Shows	1 FP3S	1,450	2,620	144,703
4	Sports	2 FPS	620	1,710	80,631
5	Entertainment	1 FP3S	910	2,622	98,364

Total		**4091**	**9,052**	**461042**

Table 4.2.3-3: Detail of Dataset Prepared from Videos Downloaded from Various Sources

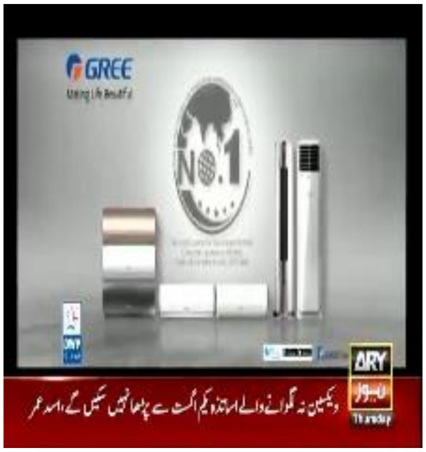
4.3 Videos Processing and Dataset Preparation

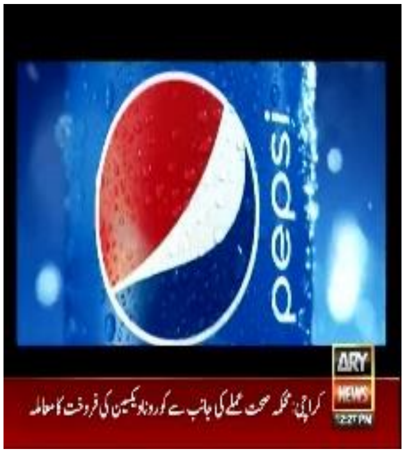
As no authentic dataset was publicly available, therefore we recorded live stream of news channels and downloaded videos related to each class from news channels websites, YouTube and other sources. To prepare trainable dataset from downloaded videos, we processed these videos using specially prepared program already mentioned above to extract frames/ images of desired size and resolution i.e., 224 x 224 pixels and 96 x 96 dpi respectively and reduced quality and quantity. So that a sufficient dataset is prepared to have generality in trained model and optimum utilization of meagre resources arranged after a huge struggle for a limited time. The detail of whole dataset prepared from all sources is given below: -

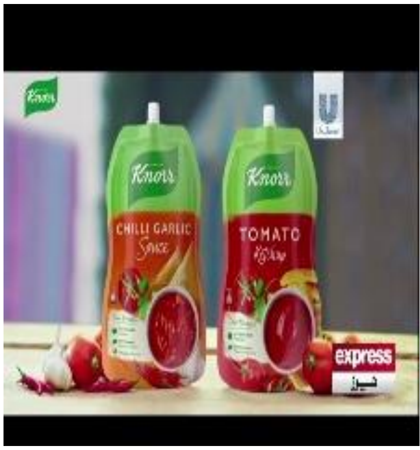
Serial #	**Class**	**Dimensions & Resolution**	**Memory (MBs) 40% Quality**	**Memory (MBs) 100% Quality**	**Total No of Frames**
1	Advertisement	224 x 224 Pixels & 96 x 96 dpi	934	2,440	122,871
2	News		1,388	3,720	140,818
3	Talk Shows		1,828	5,100	178,775
4	Sports		620	1,710	80,631
5	Entertainment		918	2,630	99,122
Total			**5,688**	**15,600**	**622,217**

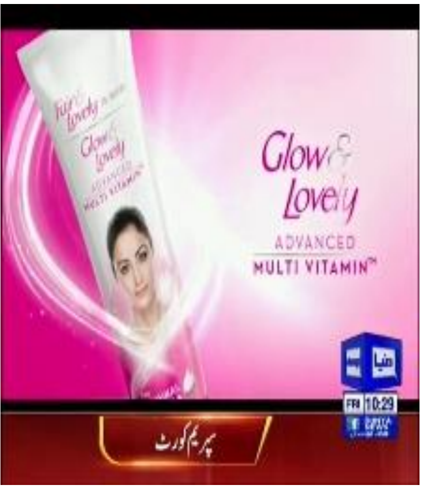
Table 4.3-1: Detail of Complete Trainable Dataset

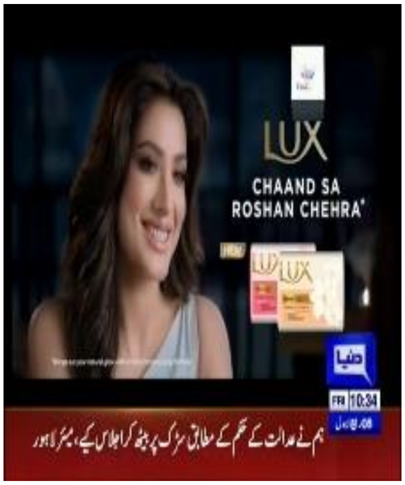
All of the above given classes' frames do have some peculiar characteristics which if extracted in the form of features can be utilized for classification of related images. In our case we are using DL convolutional networks which extract features automatically and refine them to classify into desired classes by means of neural networks layers. Now we will discuss each class with a few examples from each class one by one. In first case, here are some frames of advertisement class: -

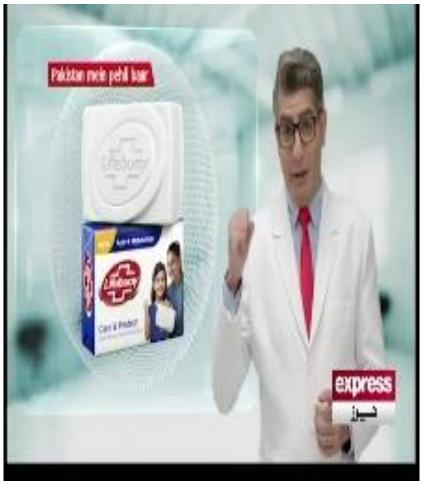

GREE
Making Life Beautiful
NO.1
ARY
نیوز
Thursday
ویکسین نہ لگوانے والے اساتذہ یکم اگست سے پڑھا نہیں سکیں گے، اسد عمر


pepsi
ARY
NEWS
12:27 PM
کراچی: محکمہ صحت عملے کی جانب سے کورونا ویکسین کی فروخت کا معاملہ


Knorr
CHILLI GARLIC Sauce
TOMATO Ketchup
Unilever
express
نیوز


Fair & Lovely
Glow & Lovely
ADVANCED
MULTI VITAMIN™
FRI 10:29
سپریم کورٹ


LUX
CHAAND SA
ROSHAN CHEHRA*
NEW
FRI 10:34
ہم نے عدالت کے حکم کے مطابق سڑک پر بیٹھ کر اجلاس کیے، میئر لاہور


Pakistan mein pehli baar
Lifebuoy
Care & Protect
express
نیوز

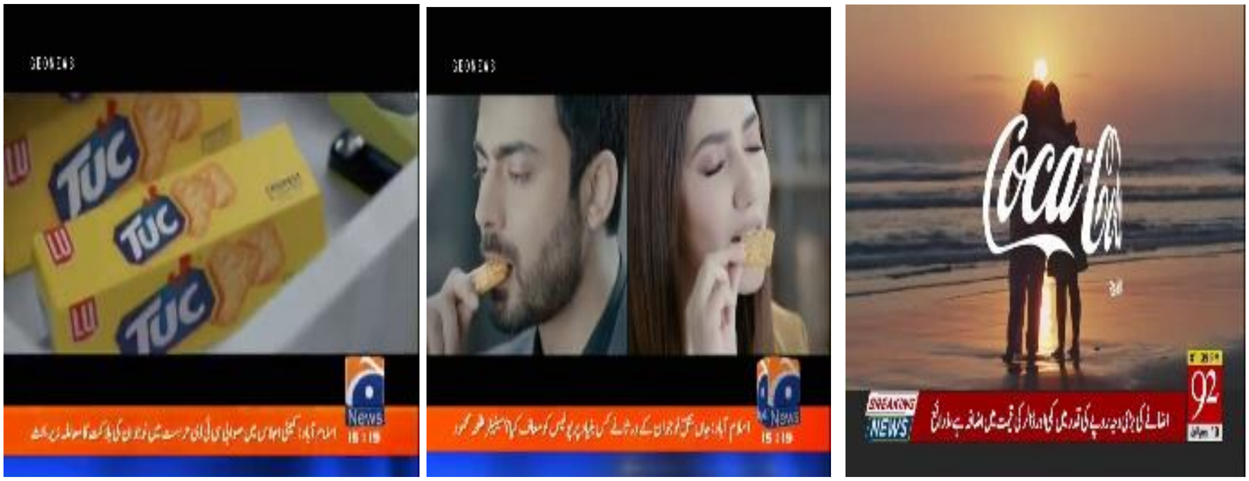


Figure 4.3.1: Advertisement Frames Extracted after Processing of Videos

If we analyze different advertisements, we can find out some specific features different than the other classes. Mostly ads consist of very small videos from few seconds to some minutes. These videos and their images contain distinctive logos, specific tickers, presentation, colors, animations, scenes etc. very typical as compared to other classes. Moreover, some of the frames related to the same ad may differ in such a way that both frames don’t have any relation. Also in news channels, different tickers also become part of a frame as shown in above images. There one can find some challenges while analyzing the ads like these may have inconsistent length and several presentation styles making identification very difficult. Sometimes, ads are shown in the corner of frame while broadcasting any program like news, talk shows and entertainment programs. Now some of the frames from news class are given below: -

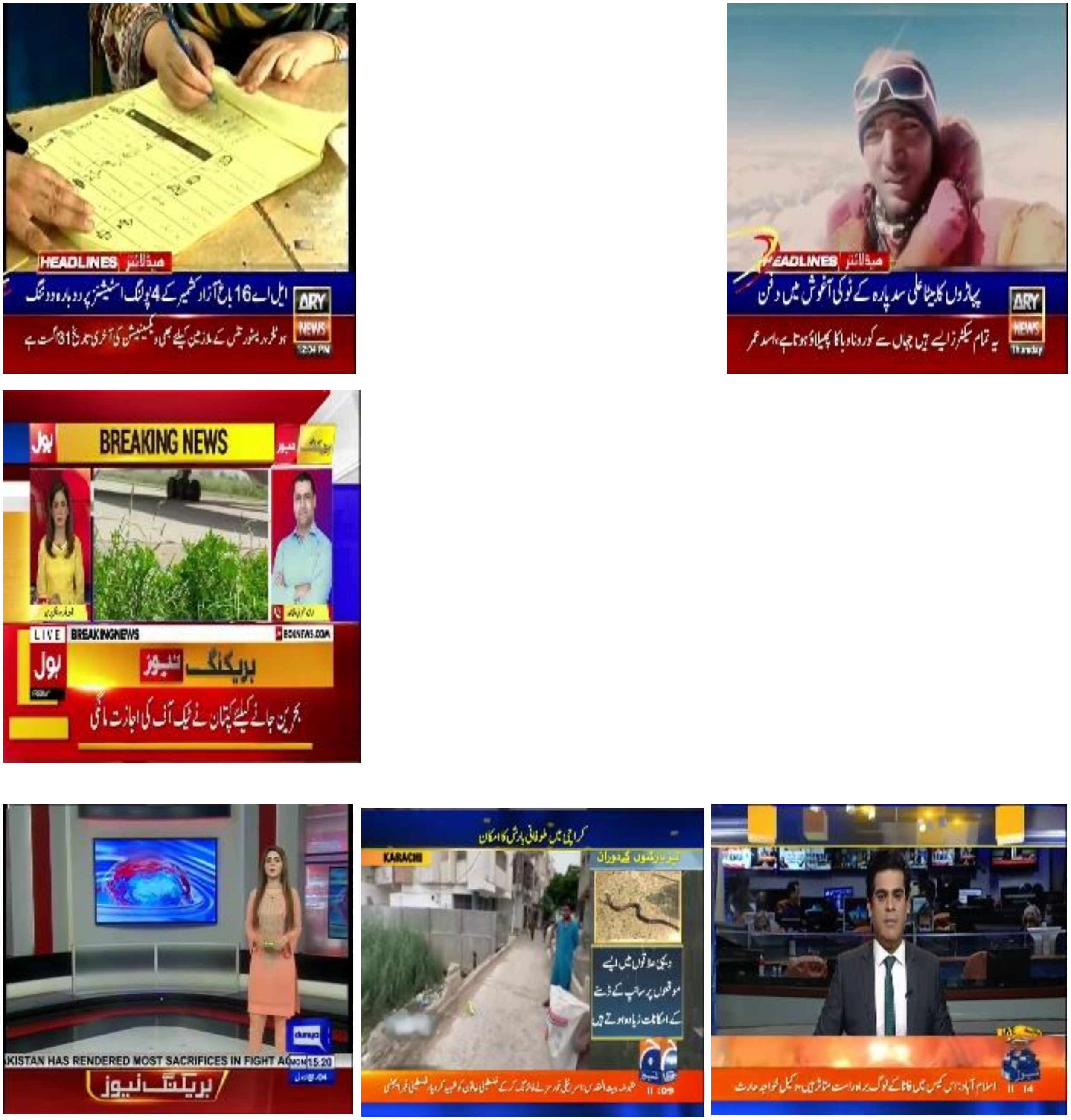


Figure 4.3.2: News Frames Extracted after Processing of Videos

It is important to highlight that news are flashed in variety of methods by each news channel and type of news varies dramatically that it is difficult to figure out the true class. However, there are many specific features which can be found in these frames to classify as news. One can find different writings, tickers, presentation styles, news anchors, their postures, specific words, color schemes, scene/ frame repetition etc. Moreover, different channels use specific style and theme different from other channels to broadcast even a same news as some

channels use one newscaster and some use two and in different postures i.e., sitting, standing, walking etc. Also, many things related to other classes may be broadcast as news or even advertisement which makes sometimes classification a very difficult task. Some of the samples from Talk shows are given as under: -

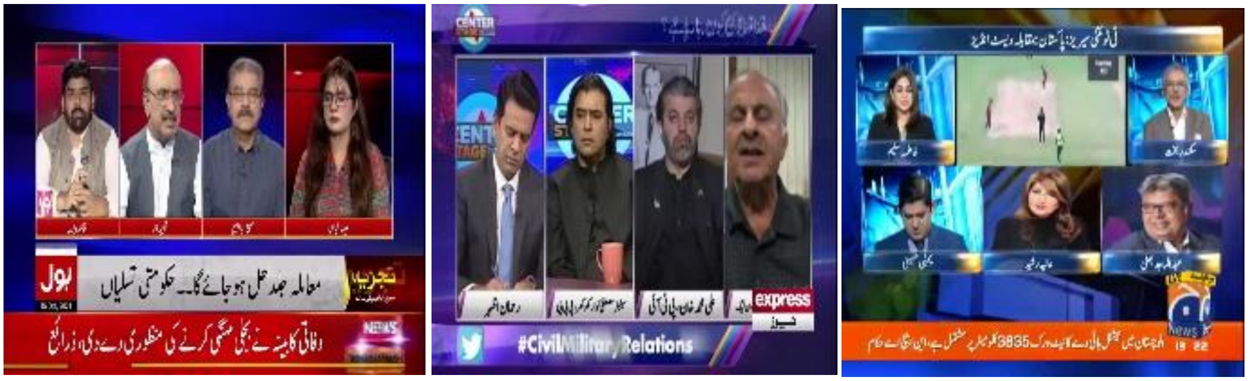


Figure 4.3.3: Talk Shows Frames Extracted after Processing of Videos

While analyzing talk shows images, there are number of frames and scenes which get confused with other classes specially, news, entertainment programs and somewhat with sports as well when programs are particularly related to sports due to relation of anchors, politician, sports, conferences etc. Talk shows are totally different in nature than other classes and some of the specific features and characteristics can be sorted out. These features may include particular anchors, their sitting styles, in some frame’s multiple anchors and guests like politician, different celebrities, usually stable frames repetitive in nature, some specific studio settings, tickers etc. Moreover, each channel adopts its own style and settings for talk shows making identification bit difficult. As one can see, some frames are very similar to news due to having only one anchor in some images just like new anchor. But still differentiation is possible with quite success. Although some of the talk shows are related to sports which may be classified as sports but due to the nature of the program, these have been places in the talk shows. However, specific scenes

and programs purely showing most of the content related to sports have been placed in sports class and its some of the examples are given below: -

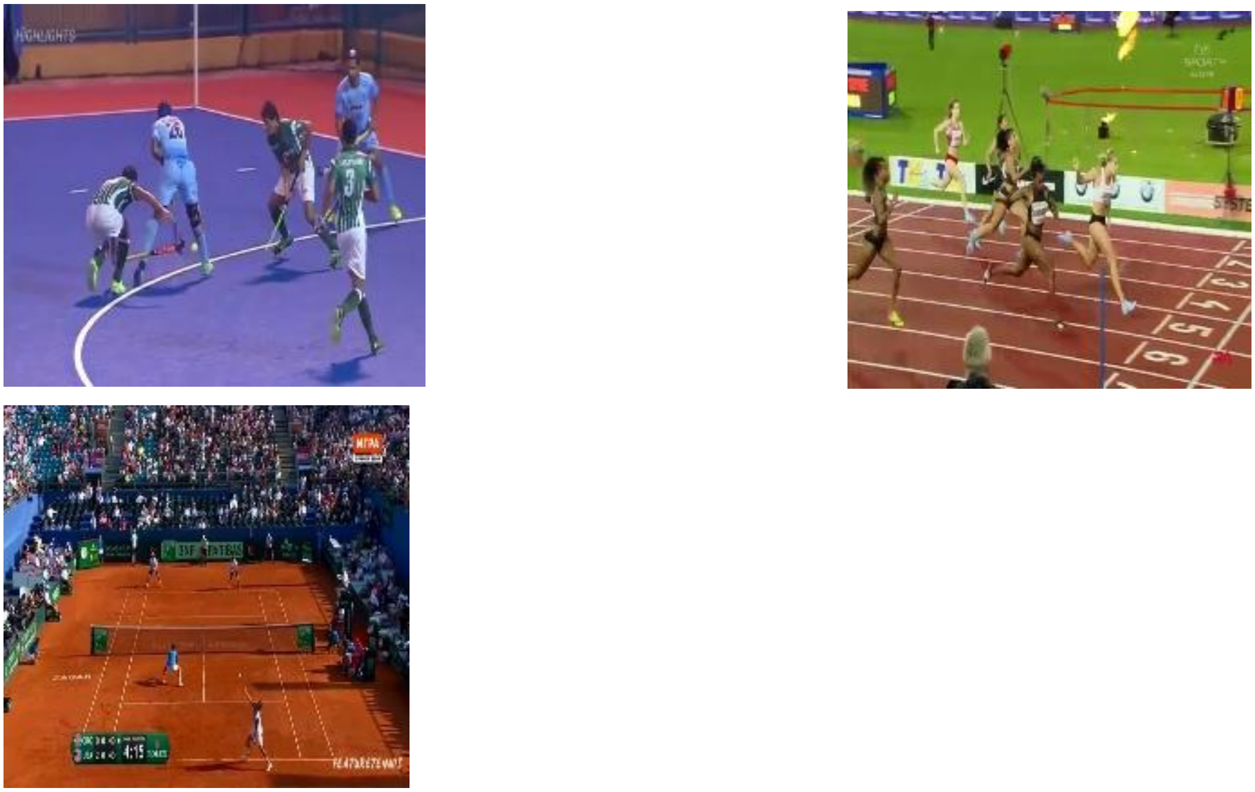


Figure 4.3.4: Sports Frames Extracted after Processing of Videos

Although sports events are not often broadcast, some of the channels do telecast sometimes the important sports matches, usually cricket, hockey, football etc. Therefore, this class was also introduced while classifying news channels content. To create generality in sports class, we downloaded sports related videos from different sources other than news channels as well. Sports frames have peculiar features due to their big differences from other classes due to having different equipment, tools, fast movements, specific conditions/ actions repetitive in nature etc. But on news channels some other peculiar writings, logos and even some ads on corners etc. are there, creating a bit of similarities with some of the other classes as well specially news. At the last we will give some of following examples from entertainment class: -

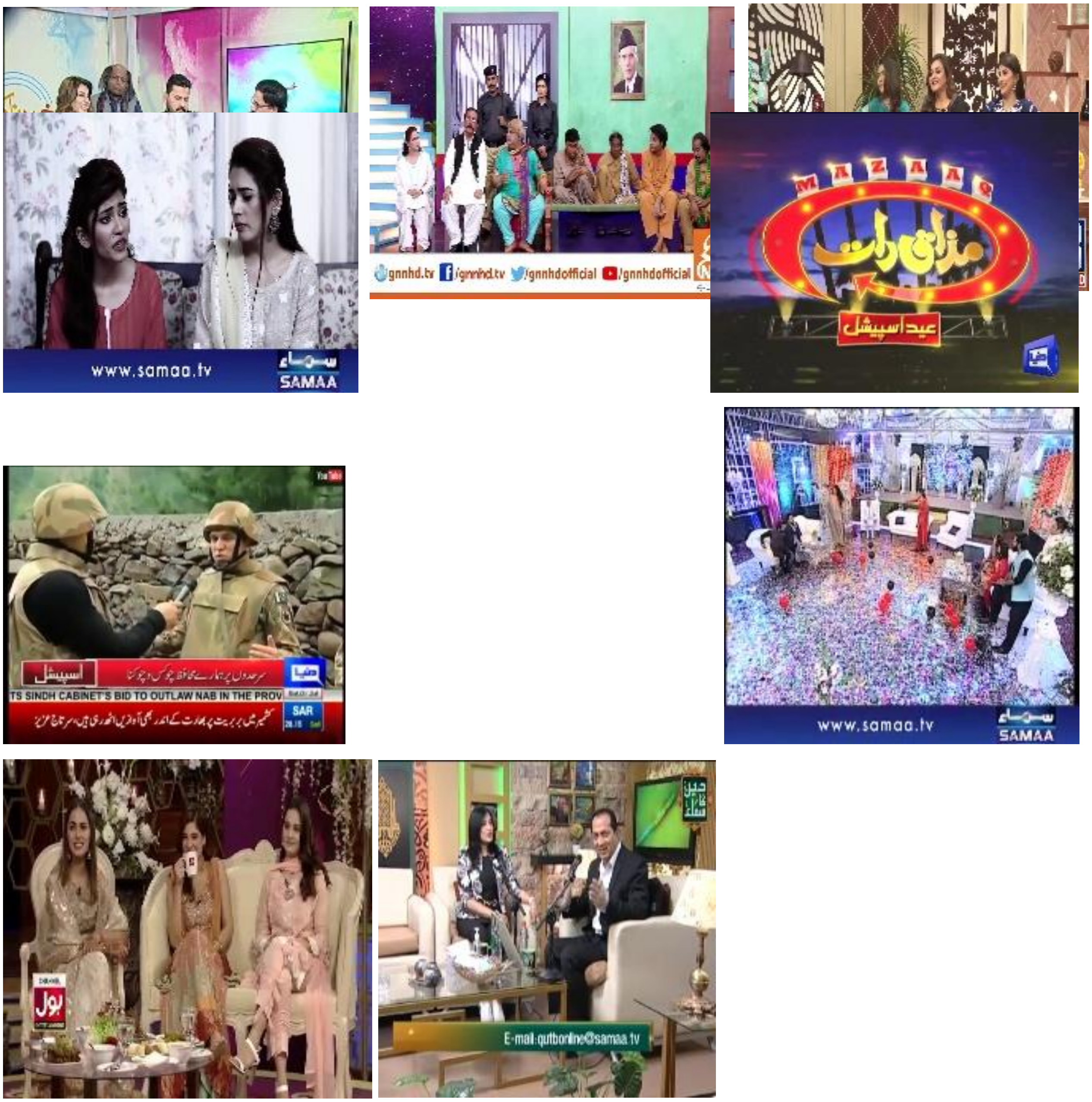

www.samaa.tv
SAMAA
gnnhd.tv /gnnhd.tv /gnnhdofficial /gnnhdofficial
MAZAAQ
مذاق رات
عید اسپیشل
اسپیشل
TS SINDH CABINET'S BID TO OUTLAW NAB IN THE PROV
SAR
www.samaa.tv
SAMAA
بول
E-mail:qutbonline@samaa.tv

Figure 4.3.5: Entertainment Frames Extracted after Processing of Videos

In Pakistan, news channels broadcast variety of programs other than news as well, like entertainment programs regularly and specially on different occasions. These programs are totally different than the other class programs but do have some similarities as well, which may conflict with talk shows and news. Even on Eid occasions, these channels telecast movies and different dramas as well. These frames also have features quite distinctive from the other classes like specific arrangements of sets, bright colors, dresses, actors, music instruments etc. Entertainment class has variety of programs and so variety of features need to be extracted from the videos, therefore, quite a big number of videos were downloaded to have maximum generality in dataset to have a good, trained model after training.

4.4 Data Challenges and Limitations

Live stream classification is a unique problem that has very less research work worldwide in general and particular in case of Pakistan, therefore, dataset preparation remained a hectic job. Video classification has very less specific literature for research. Moreover, non-availability of authentic related datasets enhanced the challenges for preparing the dataset ourselves. Although the problem is quite similar to the image classification, but the desired results are bigger and specific in nature as the stream may have very much similar scenes in different classes. So practically it proved a very difficult task for separating the videos as per classification, annotating the continuous stream and training of the neural networks. Videos of different classes do have frames quite similar in nature. Also, some channels show the same things but in quite a different style and moreover, these styles keep on changing with passage of time like altogether different sets of even same program.

Some challenges were quite unique in nature as annotating some frames, having multiple classes as having ads on programs frames etc. As channels also show advertisements logos, small videos on corners of news, entertainment, sports and even on talk shows programs as well. The data evaluation shows that all classes have some common confusing images which can be classified as multiple classes so affecting the results during testing. So, the accurate grouping can be considered as near to impossible if done at the level of frame selection. Image classification is quite easy as class image has peculiar features but when we are dealing with the videos, then getting classification purely based on images is quite difficult due to inconsistency, variety and diversity in presentation styles. The content remains dynamic in nature and keeps on drifting on the entire length of videos with passage of time. It is also important to note that the continuous evolution and changes in the presentation styles by all channels demands training of models on new data to deepen the validity of the current trained model.

CHAPTER 5: METHODOLOGY

The problem we have formulated is related to the classification of video stream of Pakistani News Channels and has been taken as a supervised learning problem. As we have gathered the video data related to each class using different sources discussed in detail in Chapter 4 of Dataset and these videos have been converted into frames and saved into separate folders as per each class. So, each frame has its own label once converted into trainable dataset using an especially prepared program. In fact we have transformed our video classification problem into an image classification problem using well established and renowned pretrained convolutional neural networks on ImageNet available on PyTorch website[168]. In the process, these models have been modified to suit our requirements as these models are trained on ImageNet to classify 1000 classes whereas we are classifying our video stream into 5 classes only. After suitable modification, we went for finetuning of these pretrained models on the especially prepared trainable dataset of extracted annotated frames from the downloaded videos from Pakistani TV Channels to achieve our purpose of video classification. Once the model is trained, we then use the trained model for video testing by developing a suitable algorithm to resolve our problem of video classification. For this, we have developed an algorithm by applying the continuous sequential video frames for desired classification using dynamic adjustable time window. This technique proved quite satisfactory. Although various other data like sound or metadata etc. may be used to enhance the quality of results. But it is not in the scope of this study, for which separate study may be carried out by students on each type of data and then combining them together for best results. Concept of method used has been shown diagrammatically as under for better understanding: -

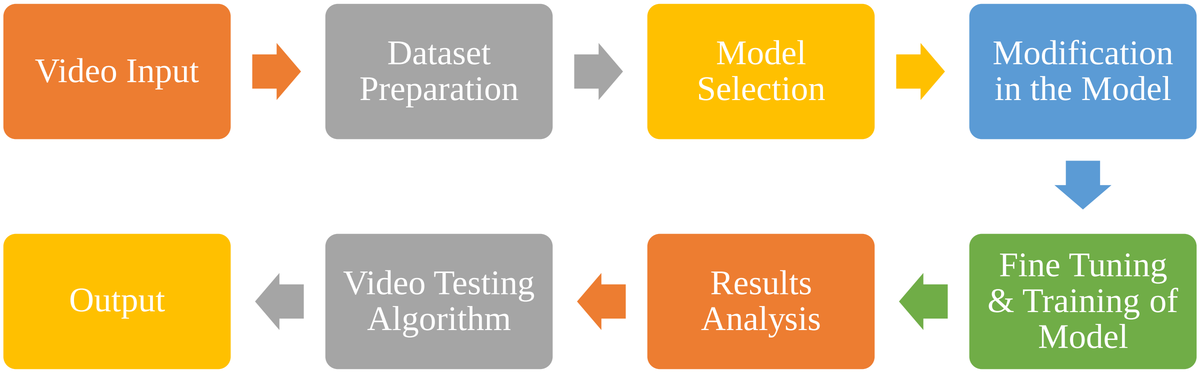


Figure 5.1: Methodology Diagrammatical Representation

We have gathered enough videos (approximately 145 GB) from various sources in the form of video clips of different lengths. These videos act as input to the methodology pipeline. These videos are then processed to prepare the dataset of desired quality. The detail of processing has already been explained in detail in previous chapter. So, the first step in our methodology is data preparation which has been done, keeping in view the available resources. Then we carried out model selection in which we selected renowned convolutional neural network models which have performed well on the ImageNet for out experimentation. These models are available on PyTorch Website both in trained and untrained form, however, we have chosen trained models after initial experimentation where trained models were further trained on our small dataset well than untrained models. Available models have been developed for image classification of 1000 classes. So, we had to do some modifications to suit our need of stream classification for five classes. Thereafter, the model has been finetuned on the prepared dataset. Results in the form of different metrics have been further analyzed to evaluate the experimented models. Finally, test videos are tested to get the desired video output duly classified using the algorithm.

5.1 Dataset Preparation

Data fetching and dataset preparation has been discussed in detail in the previous chapter. Here we will discuss more about the method we have followed for preparation of the dataset in the form of simple pseudo code. We first analyzed various Pakistani TV News Channels and came to conclusion that we can classify live video stream in five major distinct classes. Therefore, we gathered videos from live stream to suit our problem of classification into five classes. We used three methods to gather videos which include recording the live stream and then preparing annotation file for recorded stream and next two techniques are generally similar in which most of the advertisements from various websites and YouTube whereas other classes related video clips. from channels' websites and YouTube and saved into class folder.

We also could not find an authentic and easy to use method for directly classifying the videos. However, there are many established techniques of image classification which could be used to suit our problem by suitably developing a mechanism for video classification. The available DL neural networks need dataset in the form of frames or images. Therefore, we developed a program to suit our all three types of input videos and able to extract image frames at a directed interval i.e., 1/2/3 frame per 1/2/3 second and processing each frame as per desired size, resolution or quality of image to reduce the required memory. This not only produced desired size of images suitable for ANN but also reduced required space which not only made the training of the model convenient but also time conserving technique. Our data videos are mostly structured to have 24 to 30 frames per second. If we analyze the pattern of presentation of various programs, usually not much changes occur in the case of entertainment, talk shows and even in news during two to three seconds. Therefore, this technique not only reduced similar data but also generated quite reasonable data, generic enough to train the model suitable for our problem. We also tried to prepare a balanced data for each class. Lastly these extracted frames were saved into respective target class folder. We had total videos of approximately 335 hours length and prepared a dataset of almost 622,000 images of 224 x 224 x 3 dimensions. The pseudo code for above mentioned procedures for frame extraction from videos is given as under:-

```
INPUT1        Recorded video stream.
INPUT         Annotation file of recorded video stream
INPUT2        Advertisements and video clips of all other classes saved into class folder,
INPUT3        are similar in nature having folders and sub folders inside.
IF            INPUT1
    PROCESS       Annotation file of recorded video stream to identify the video sections.
    WHILE         Video annotation section from annotation file
        READ          Annotation section metadata from the file.
        READ          Video section.
        EXTRACT       One frame per second or as per the desired interval from video
        PROCESS       Each frame as per target dimensions and quality.
        SAVE          Each frame into respective target class folder.
        ENDWHILE
ELSE            (INPUT2&3)
        WHILE           Video clip from the class folder/ sub folder
                READ            Video clip.
                EXTRACT         1FPS/2 FPS/ 1 FP2S/ 1 FP3S or as per the desired interval
                PROCESS         Each frame as per target dimensions and quality.
                SAVE            Each frame into respective target class folder.
        ENDWHILE
```

In the case of INPUT1 in the pseudocode, it is the video clip recorded from the live stream and has multiple sections of different programs i.e., news, advertisements, talk show or entertainment program. These programs may vary in length which are exactly identified with the help of annotation file, segregated, processed for reading the video clip, extracted frames as per desired interval to reduce the space and processed for target dimensions and quality and finally saved into the target folder as per class label. From videos of INPUT1, we get frames related to four classes of News, Entertainment Program, Talk Show and Advertisements. Whereas in the case of INPUT2&3, there are video clips of different lengths already saved into folders of their related classes i.e., News, Entertainment Program, Talk Show, Sports and Advertisements. These videos from sub folder of class folder are picked one by one, program reads each clip, extract the frames as per interval set as 1FPS etc. for each class, process for desired target dimensions and quality to reduce required space and finally each processed image or frame is saved into relevant class folder. Hence, in the end we prepared a trainable dataset suitable for image classification problems. We may include various processing properties other than dimensions, size, quality like color variance, resolution, deformation etc. This way we get data suitable to feed into selected neural networks for training and testing of models. We can adjust number of desired classes in the models with minor changes. Some of the important properties of dataset are given below: -

Properties	**Values**
One Hot Encode (OHE) Labels	Advertisements: 0, News: 1, Talkshow_News: 2, Sports: 3, Entertainment_Misc_Program: 4
Total Samples	622,217
Data Split	Training: 70%, Validation: 15%, Testing: 15%
Data Distribution	Advertisements:122871, News:140818,Talkshow_News:178775, Sports:80594 , Entertainment_Misc_Program: 99122

Table 5.1.1: Important Properties of Dataset

5.2 Model Selection

Although our problem in hand is video classification however, we have converted our problem into image classification. Therefore, we can use different CNN as these are being used widely for various image classification and giving better results. We have discussed earlier that these networks have been designed into two parts like convolutional and fully connected layers. Convolutional layers are responsible for features extraction from the image dataset and fully connected layers use these features as input to compute the results in the form of desired classification of input image. Due to their ease in use and better results convolutional neural networks have become the top priority choice in computer vision related problems.

Many CNN have been developed to resolve image classification problems and over the years, various changes in terms of depth and structural innovations have been applied to improve the performance. Some of the popular networks are AlexNet, VGG, ResNet, GoogleNet, Inception-V3, SqueezeNet and DenseNet. With the passage of time, many researchers have further applied variations in their structures, innovations and depth by increasing or decreasing the number of layers for even better results. So, now we find many variations of these networks as well i.e., ResNet18, ResNet34, ResNet50, ResNet101 etc. Moreover, there are many worldwide challenges being organized for evaluation of various CNN and the popular one is ImageNet. This challenge consists of a huge dataset of images to produce 1000 types of classifications. During the challenge, models and their variations are put into evaluation by training these models on this huge dataset and results are compared to check the best performer. Different forums like PyTorch, TensorFlow etc. are using this dataset to train CNN and place in their libraries for further experimentation by other researchers. As these networks are already trained, researchers can fine tune these networks conveniently on their own dataset for desired results in less time as these networks learn quickly better features from given image dataset.

In this study, we have mainly used ResNet and their variations i.e., ResNet18, ResNet34, ResNet50, ResNet101 & ResNet152. However, we have tested other available CNN as well for better comparison and results which include AlexNet, ConvNeXt_Tiny, VGG11, DenseNet121 and SqueezeNet. ResNets are simple and use skip connections. Most consider these networks efficient and easy to handle as compared to other networks having large structures of stacked layers. Due to skip connections in ResNet, gradient flow has further improved than other networks. Networks with larger stacked layers are harder to train and take more time. We have used the above mentioned pretrained networks on the ImageNet-1K dataset which are able to classify images of 1000 classes. Therefore, extract better features than the un-trained same models and perform quite better. These nets are then finetuned on our data for stream classification by using specially prepared video testing algorithm.

5.3 Modification in the Pretrained Model

As we know, we have chosen different pretrained models on ImageNet and these are able to classify 1000 different image classes. But in our case, we have to classify the video stream into five classes. Therefore, first of all we need to do modify the network to be able to classify the dataset into five classes instead of 1000 classes. In most of the models last layer of fully connected part is used for classification of input data. Therefore, we can replace and modify last layer of model being used to suit our requirement of classification of data into five classes. So, in our modified model, the last layer will have only five neurons. Structural modification is shown as under: -

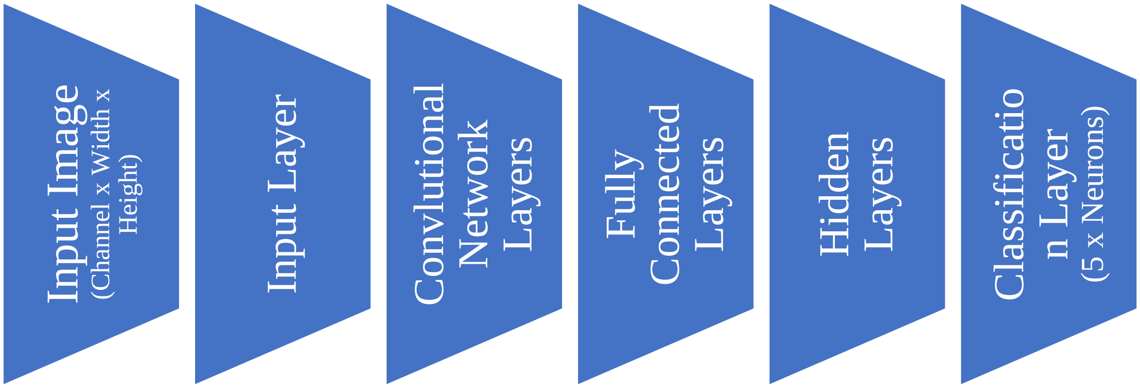


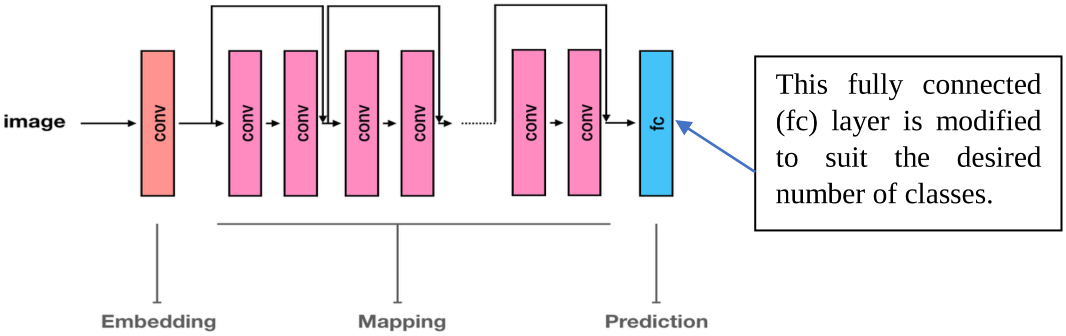
Figure 5.3.1: Modification of Neural Network’s Classification Layer

In the above generalized model, we have modified the classification layer to 5 neurons instead of 1000 neurons. So, due to this last and 2nd last layer architecture has also been changed as a result. Thereafter, the subject model (pretrained on ImageNet-1K dataset) after necessary modification is finetuned on our dataset to be able to predict extracted frames from live stream as per desired classification.

5.4 Fine Tuning and Training of Model

We carried out the structural changes in the classification layer of all pretrained models (ResNet18, ResNet34, ResNet50, ResNet101, ResNet152, AlexNet, ConvNeXt_Tiny, VGG11, DenseNet121 and SqueezeNet. ResNets) downloaded from the PyTorch Website and then adjusted their different parameters and hyperparameters for desired training of these models. After necessary modification of neural network architectures, we adjusted important parameters i.e., setting of requires_grads to true for finetuning of all weights, and then finetuned all weights of both parts (CNN & fully connected layers) on our datasets for stream classification. Earlier we did small experimentation to check the performance of the trained networks by keeping the setting of requires_grads to true and false. As per the result, the network with keeping requires_grads to true performed better than the other option. In the second option only the weights of last layers were trained on our data which gave lower accuracies. In machine learning, finedtuning means to get a pretrained neural network and then train it completely (both CNN & fully connected layers weights) on fresh dataset. While in case of transfer learning, we freeze all weights of the pretrained model except the last classification layer. Here we are finetuning the whole network on our dataset.

For better understanding we have taken the example of ResNet architecture, and all further discussions will revolve around this network. The simple schematic view of ResNet architecture[169] is shown below: -



This fully connected (fc) layer is modified to suit the desired number of classes.

Figure 5.4.1: Schematic view of ResNet architecture having three blocks: embedding, mapping and prediction.

For our problem formulation, we modify the fully connected layer of pretrained ResNet model on ImageNet-1K dataset to act as classification layer with five neurons. Then we set the parameters to finetune all weights of all layers on the prepared dataset for stream classification. The ResNet model finetuning pipeline is shown as under: -

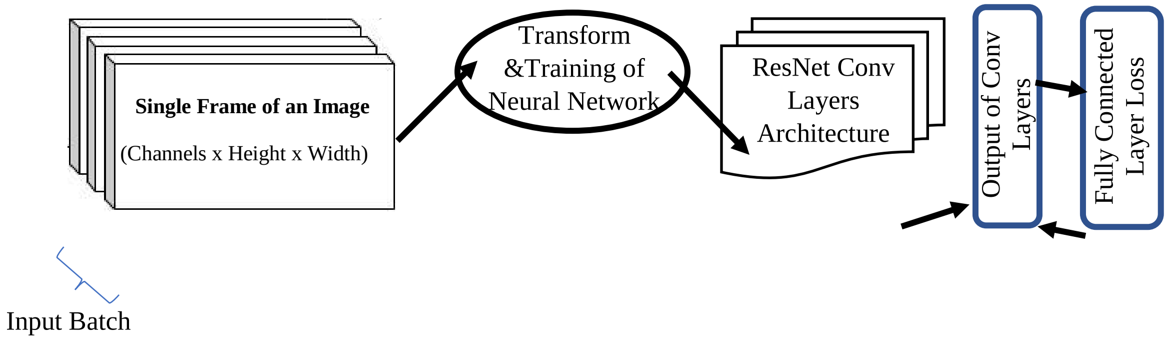

Single Frame of an Image
(Channels x Height x Width)
Input Batch
Transform &Training of Neural Network
ResNet Conv Layers Architecture
Output of Conv Layers
Fully Connected Layer Loss

Figure 5.4.2: ResNet Model Finetuning Pipeline

Input dataset batch is fed into the neural network architecture in the form of single frame or image. Which is then passed through the network transform to reshape, normalize and apply some other random transformations for creating a generalized data for training of the model to compensate variations in the data in real problem while performing testing of model. After applying transformations, the data is passed through the pretrained ResNet or subject pretrained model architecture having convolutional layers. Convolutional layers' existing weights are finetuned and updated on the current data. These convolutional layers extract the updated features, and the output is passed through the pooling layer before it is fed into the fully connected layer which acts as the classifier. On the basis of these new features the weights are updated and finetuned. As the weights of all layers are updated, output is computed and compared with the label of image using cross entropy or some other loss function to get the loss value. The gradient or derivative computed by the loss function is then backpropagated through all layers of network to update the weights of all layers i.e., both convolutional and fully connected layers. This process continues till the last image of the dataset for completing one epoch of the training of the model. Thereafter, model is trained for some epochs like 10 or 20 epochs for the video stream classification problem in hand.

5.5 Results Analysis using Different Metrics

The performance of any neural network model is judged through various metrics. We also need to compute the metrics to evaluate our trained model to compare it with other models trained on the same dataset and check its performance in terms of accuracy, precision etc. For stream classification, which has been converted into an image classification problem to utilize already established neural networks, the major metrics to estimate the performance of such models are given as under: -

- Accuracy.
- Precision.
- Recall.
- Jaccard Score.
- F1 Score.
- Confusion Matrix.

Each metric mentioned above has already been discussed in detail in earlier chapter. These metrics have been computed during the training of each model and their results have been placed in the Results Chapter. These have also been compared with each other for better understanding and decision-making.

5.6 Video Testing Algorithm and Output

Once we have finetuned and trained the pretrained model on the dataset, then we computed the necessary metrics mentioned above for its evaluation. This trained and finetuned model is then used to get the predictions for classification of live stream obtained from the Pakistani News Channels. The procedure for stream classification we have adopted works on the concept of use of dynamic time window. In which the average of output is taken to make out the resultant prediction. The output is usually a one-dimensional vector produced against the input image frame fed into the finetuned model architecture. This output is in the form of probabilities got from the classification layer. While doing stream classification, we feed the video to the specially prepared algorithm which then converts video into the frames and each frame is then fed into the trained model architecture to get the output probability against that frame which results in the output prediction for all frames of complete video without any break. By using the time-based dynamic window, we can handle the stream classification quite conveniently. It is important to note this may result in poor performance by having jitters in output due to the reason of dealing with number of such frames having weak or no features. As we are dealing with the continuous stream of video so, the frames extracted by the algorithm are also in continuity and sequence. Therefore, the result will be in the form of probabilities in sequence. To deal with our problem we have used probability averaging in which we take average of probabilities across the specified time window (window will have specific number of frames depending upon the video FPS). Due to this procedure, we will be able to remove jitters and confusion in the predictions due to weak frames in the video. Stream Classification Prediction Probability ($P_{out)}$ has been put in the form of mathematical representation as below: -

$$P_{out} = \sum^{n}{}_{i=1} [P_{advertisement} , P_{news}, P_{sports}, P_{\ entertainment,} P_{\ talkshow}] / n \qquad (5.6.1)$$

$P_{advertisement}$, P_{news}, P_{sports}, $P_{\ entertainment,}$ $P_{\ talkshow}$ are the probabilities of all classes, which are summed after the prediction against each input frame and divided by the total number of frames (n) put into the model as per the time window domain. This results in the prediction of each frame after initial time gap. Same can be shown in the pictorial representation as well for the prediction of output of stream classification as under: -

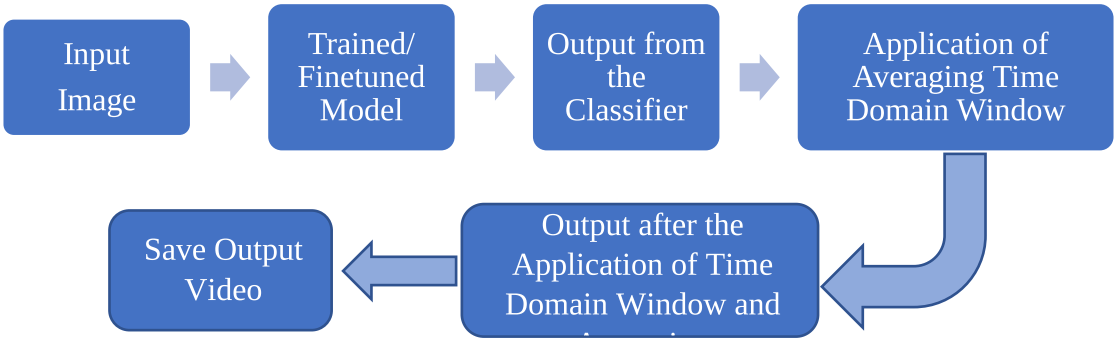

Input Image
Trained/ Finetuned Model
Output from the Classifier
Application of Averaging Time Domain Window
Output after the Application of Time Domain Window and
Save Output Video

Figure 5.6.1: Video Testing Pipeline

The above pipeline indicates the input in the form of sequenced frames are fed to the finetuned/ trained model one by one. Model applies inference and predicts the output class in the form of probability against each image frame. Then the time domain window is applied on the predicted output probabilities and averaged out for better prediction as final output which is then saved as video duly classified stream. The pseudo code of used algorithm for above mentioned procedures for frame extraction from testing videos and then testing of video stream by applying time domain window is given as under: -

INPUT1 Recorded or live video stream.

```
    WHILE
        READ        Video input.
        EXTRACT     Each frame one by one from video.
        PROCESS     Each frame as per target dimensions.
        PROCESS     Each frame fed into finetuned model.
        SAVE        Predictions or outputs against each frame into the list.
        SET         Time window length.
            IF      Time window length == Prediction list length.
                    PROCESS     Averaging the prediction outputs.
                    PROCESS     Putting Text on each output frame as per class.
        SAVE        Each output frame as video marked as per classification.
    ENDWHILE
OUTPUT      Video output marked as per classification of live video stream.
```

CHAPTER 6: EXPERIMENTATION

As we have discussed in the earlier chapters that although our problem in hand is to do the classification of the live stream obtained from the Pakistani News Channels. Moreover, we have come to the point that the problem can be resolved using the latest image classification techniques matured after thorough research throughout the world by the scientists so far by employing deep learning and machine learning neural networks. This has been further implemented with the use of pretrained neural networks on ImageNet datasets available in different libraries and in our case, we downloaded pretrained models from the PyTorch Library. Videos as we know, contain image frames arranged in a sequence with a specific number of frames per second. In a way, we can say that image classification methods can also be used with doing some innovations to resolve our problem of video stream classification. Moreover, one can find the reality that the contemporary techniques being used for image classification are still at a rudimentary stage. Because mostly cases the images in the current datasets are quite simple i.e., images of animals, aero planes, trains, ships etc. Whereas, when we look into detail at the live video stream, one can find multiple objects, individuals, animals etc. in a single image, each may depict quite a different scenario. Therefore, the classification of images obtained from video stream becomes altogether different and complicated problem than the simple image classification scenario.

There are many other techniques being used for image classification, we will just briefly talk about. Simplest and quite reliable techniques include the computer vision (CV) methods without involving ML and DL. These techniques are good to some extent but are not 100% reliable as these do not provide a generalized solution for complicated problems. But these are simple in use and don't require heavy computational resources like needed in case of DL. Some of the simple CV techniques include feature extraction and feature matching. If the dataset is dynamic, then these techniques may fail to extract useful and meaningful information for affective classification. As in such dataset, the color information, spatial construction, animation, resolution etc. may get variance leading to different feature extraction as compared to the earlier one, every time we feed any image for inference. These feature extraction methods of CV use Scale Invariant Features Transform (SIFT) to match similarity with other similar images. In complex problems these methods do not produce stable and consistent results for image classification. There are some machine learning techniques as well for image classification which include ANN, K nearest neighbors, decision tree, support vector machines (SVM), CNN etc. If we see their performance in computer vision tasks through results given in different papers and articles. We can easily make out the conclusion that with the use of ANN and CNN, the CV related tasks results have improved much[170]. Here after, we will focus on the actual experimentation carried during the process of this study involving DL techniques only. Initially we carried out preliminary experimentation for deciding use of pretrained models or not, finetuning of model or training of last classification layer only and effects of different data qualities. Thereafter, will discuss the experimentation on different models and various aspects for video stream classification.

6.1 Preliminary Experimentation

Initially we carried out some preliminary experimentation for deciding three important questions which include following: -

- Whether the performance of pretrained neural networks model is better or of raw model?
- Should we train only the last classification layer of the pretrained model or finetune the complete network on our dataset?

- What is the effect of changing the quality of dataset?

To get the answer to these questions, we took a small dataset of an earlier collection and performed small experiments using Google Collab due to non-availability of proper computer resources. We will not include the detailed results but only the conclusions we reached after the experimentation. We used ResNet18 neural network for this initial experiment. We took both pretrained and untrained ResNet18 model and trained on the available small dataset and compared results and found that the pretrained model showed quite better results than the untrained model. In 2nd case the performance of finetuned model was 3 to 7% better than the model trained for last classification layer only. Moreover, we prepared datasets of different qualities ranging from 30% to 100% (original image) to save the storage requirements and reduce the computational requirements. There was pleasant conclusion that the computational requirements and training time reduced much but there was nominal drop in quality of training and performance of model. We have included some of the results of using two datasets (dataset#1 with 40 % quality and dataset#2 with 100% quality images) in Chapter 7: Results. With this preliminary experimentation, we concluded to use pretrained models and applied finetuning of complete neural network on dataset#1 with 40 % quality except for few for comparison purpose in results.

6.2 Experimentation on Various Neural Networks

After the preliminary experimentation we shortlisted some of the established and popular neural networks keeping in view their performance etc. We mainly focused on use of different variants of ResNet due to their popularity and performance during worldwide challenges. However, we also carried out experimentation on other some old and few newer models as well for better comparison and creating flexibility for their use depending upon the problem severity. These models include AlexNet, ConvNeXt_Tiny, DenseNet121, SqueezeNet, VGG11 and variants of ResNet i.e., ResNet18, ResNet34, ResNet50, ResNet101 and ResNet152 all downloaded from PyTorch Torchvision Website[171]. All of the models used for experimentation are pretrained and we carried out finetuning of complete models including both parts i.e., convolutional and fully connected layers parts on mainly dataset#1 with 40% quality. Details have been covered in the previous chapter of methodology. Due to various limitations of computational resources, time constraints and electricity interruptions, we were able to do limited training of shortlisted models on datasets. Therefore, first we carried out training of all models for ten epochs on 1st dataset (dataset#1) only and then two models on 2nd dataset (dataset#2) for 20 epochs as well due to available cushion in time. We were able to generate quite encouraging results which have been given in detail in Chapter 7.

6.3 Other Aspects of Video Stream for Classification

Video stream obtained from the Pakistani TV News Channels carries various aspects in terms of information and type of data. So far, we have discussed much about the information available in each image frame of the video. But if we analyze in detail, these live videos also carry metadata embedded with the video which provides sufficient data to make out the type of content being broadcasted by the channel, secondly most of videos also carry embedded sound and lastly the many images in the video also have written scripts in the form of single word as well as short and long sentences. It is important to notice that the sound riding on the video has specific linkage with the images, variations in each frame. For example, if someone is talking, his lips movements have relation with the voice and if linkage is correctly linked one may classify the video part more accurately than just relying on the image features only. So, the sound features extracted and then creating a relation with each frame or group of frames may bring quite accurate results. As we have focused on the use of convolutional networks for extraction of features from the dataset, which definitely bring very useful features including written scripts as well. But these may not be well connected and have decisive impact as these features are being gathered randomly in the case of CNN. However, if we use the written script recognizing techniques like OCR software, we may bring better information from the image contributing well to its classification. Moreover, we can use all these four types of information collectively as well for more accurate image classification leading to even better stream classification of live video stream of news channels.

CHAPTER 7: RESULTS

As we didn’t have authentic solution for classification of video stream obtained from Pakistani News Channels, therefore, we tried different authentic DL architectures having good performance for image classification. After initial experimentation, we realized that already trained models on ImageNet give better results than the models with no training. Therefore, we tested ten different models including AlexNet, ConvNeXt_Tiny, DenseNet121, SqueezeNet, VGG11 and variants of ResNet i.e., ResNet18, ResNet34, ResNet50, ResNet101 and ResNet152 downloaded from PyTorch Torchvision[171]. We carried out fine tuning of all pretrained weights on our dataset and modifying the last output layer suitable for giving desired results on five classes. As we discussed earlier, we prepared two datasets i.e., 1st with 40% quality and 2nd with 100% quality in original. We could do limited training of these models on limited quantity of dataset (two qualities 40% & original) due to the reason that we could arrange limited resources for limited time. Therefore, first we carried out training of all models using ten epochs on 1st dataset **(dataset#1)** and two models on 2nd dataset **(dataset#2)** as well. Thereafter, we also training two models for 20 epochs as well. Moreover, due to electricity issues and having remote access, a number of times, we had to restart training which caused numerous problems.

However, sufficient training was completed to generate desired results, which we will discuss one by one in following sections.

7.1 AlexNet

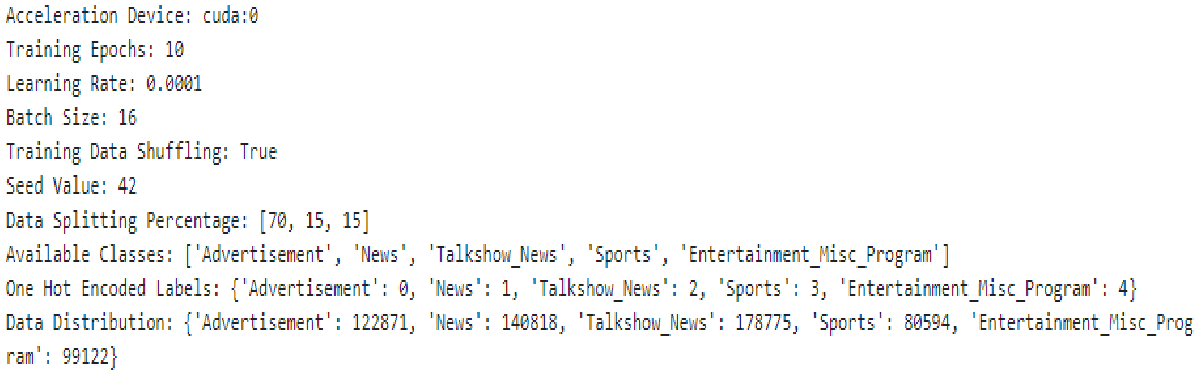
AlexNet[172][173] is one of the initial method to parallelize the training of convolutional neural networks with SGD (stochastic gradient descent) using ReLU first time. This network first introduced in 2012 and variants in 2014 which gave encouraging results and ways to speed up the training making use of convolutional neural networks possible with GPUs. Without much focus on details on network which one can find in number of papers, we will concentrate on the parameters used and results obtained through training of the model on our prepared dataset.

Following are the hyper parameters used while training of the model: -

Model Training Fine Tuning Hyper Parameters - AlexNet		
Serial #	**Parameters**	**Values**
1	Number of Epochs	10
2	Learning Rate	0.0001
3	Batch Size	16
4	Seed Value	42

Table 7.1-1: AlexNet Model Fine Tuning Hyper Parameters

This shows few parameters and dataset distribution for training, validation and testing along with the one hot encoding for each class and number of images in each class of dataset.



```
Acceleration Device: cuda:0
Training Epochs: 10
Learning Rate: 0.0001
Batch Size: 16
Training Data Shuffling: True
Seed Value: 42
Data Splitting Percentage: [70, 15, 15]
Available Classes: ['Advertisement', 'News', 'Talkshow_News', 'Sports', 'Entertainment_Misc_Program']
One Hot Encoded Labels: {'Advertisement': 0, 'News': 1, 'Talkshow_News': 2, 'Sports': 3, 'Entertainment_Misc_Program': 4}
Data Distribution: {'Advertisement': 122871, 'News': 140818, 'Talkshow_News': 178775, 'Sports': 80594, 'Entertainment_Misc_Prog
ram': 99122}
```

The overall **AlexNet Model** stream classification results on complete **dataset#1** after 10 epochs are as under: -

Stream Classification Overall Loss/ Accuracy – AlexNet Model		
Serial #	**Properties**	**Values**
1	Training Loss	0.1228771435420380
2	Training Accuracy	0.9591261021307850
3	Validation Loss	0.0575778938893738

4	Validation Accuracy	0.9810560270919060
5	Testing Accuracy	0.9820923353909465

Table 7.1-2: AlexNet Model Stream Classification Overall Training and Testing Results

These are the epoch wise training and validation loss and accuracies results of **AlexNet Model** on **dataset#1** for **10 epochs**: -

Training/ Validation Loss/ Accuracy Results Epoch Wise – AlexNet Model				
Epoch	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuracy**
1	0.2415262999956140	0.9168832659808960	0.0976678338338450	0.9665852194787380
2	0.1468965589677300	0.9510263592946360	0.0645233474320067	0.9786736968449930
3	0.1267421830167340	0.9581442872887580	0.0615917048350706	0.9803347908093270
4	0.1162046432869230	0.9617583578251280	0.0542398496918697	0.9821887860082300
5	0.1089330702850600	0.9640039493019830	0.0582323858973335	0.9812028463648830
6	0.1038722039342170	0.9658477222630410	0.0489148840024387	0.9830782750342930
7	0.1000554159049630	0.9669222997795730	0.0460758024729259	0.9857574588477360
8	0.0972940970275705	0.9682815944158700	0.0472040187741282	0.9855109739368990
9	0.0943958348004928	0.9691219691403380	0.0420896662065987	0.9857360253772290
10	0.0928511282010800	0.9692712160176340	0.0552394457475207	0.9814921982167350

Table 7.1-3: AlexNet Model Training Epoch Wise Results

Representation of above epoch wise training and validation loss results in the form of graph is given below: -

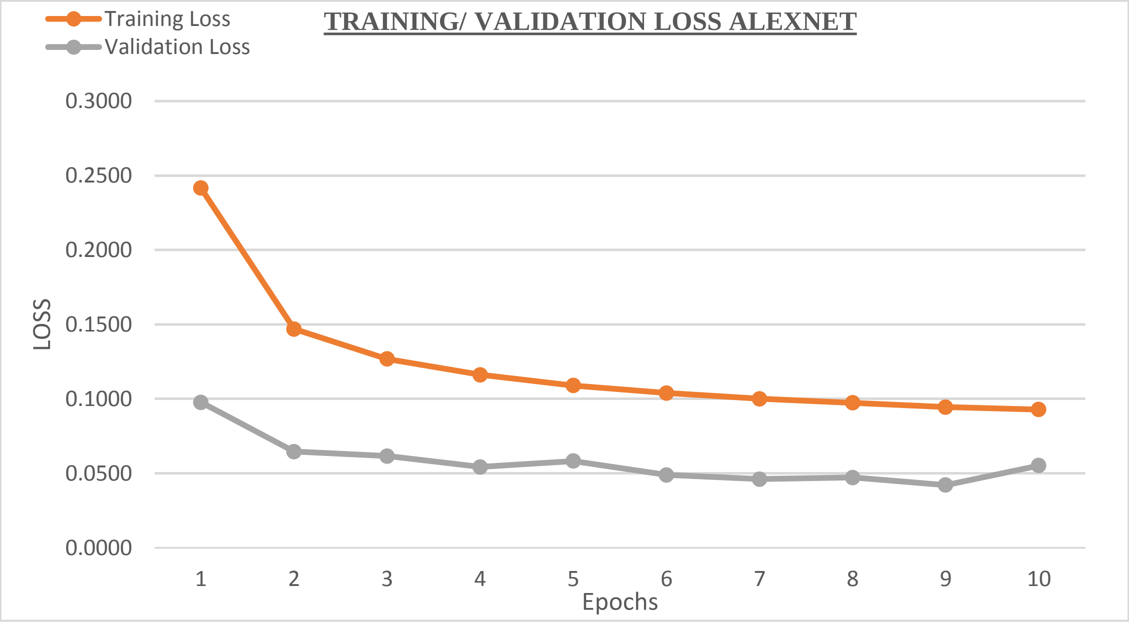


Figure 7.1.1: AlexNet Training and Validation Loss

Representation of above epoch wise training and validation Accuracies results in the form of graph is given below: -

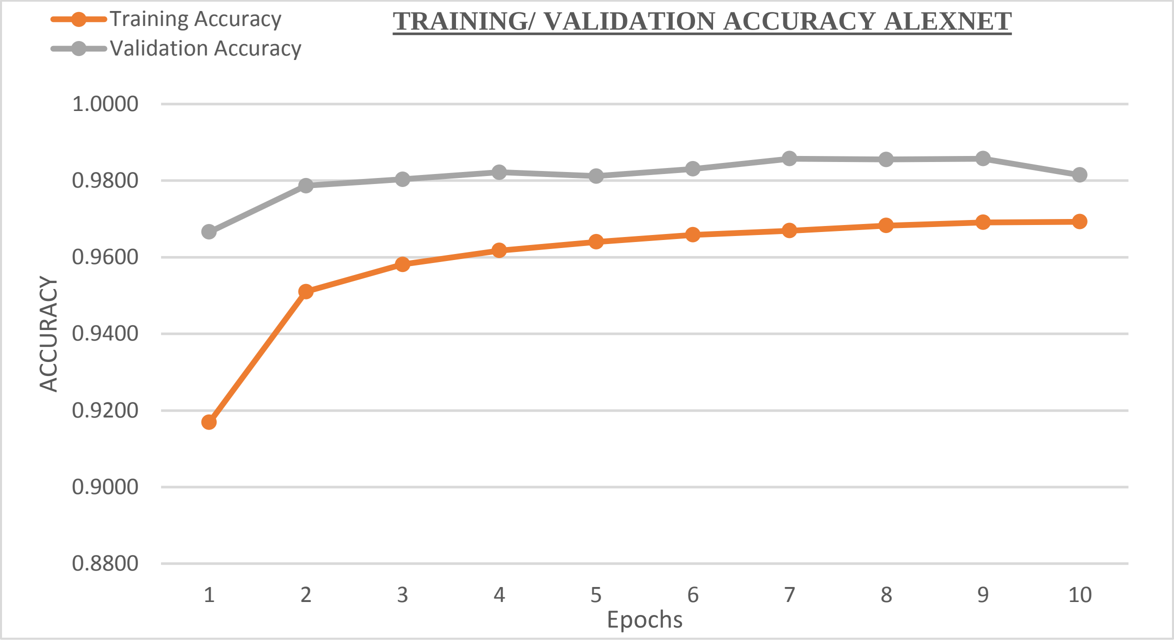


Figure 7.1.2: AlexNet Model Training and Validation Accuracies

Confusion matrix formed after training of **AlexNet Model** on **dataset#1** for 10 epochs, represents classwise wrong and correct classification results as below: -

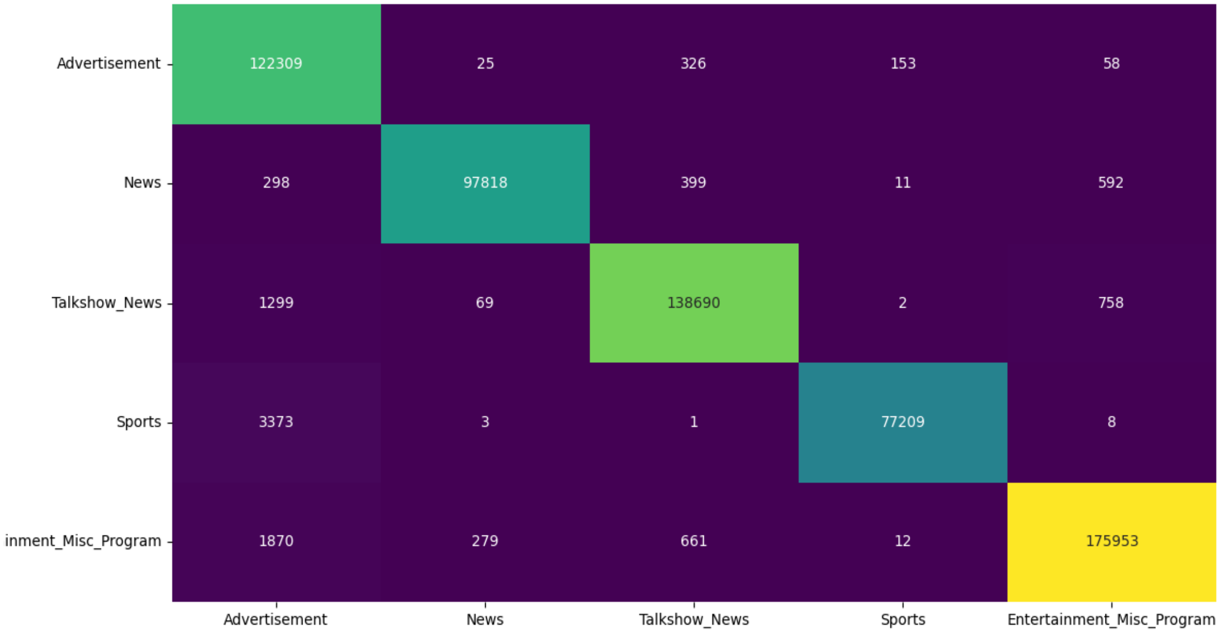


Figure 7.1.3: AlexNet Model Confusion Matrix

Four metrics scores calculated for analysis of trained **AlexNet Model** on **dataset#1** for 10 epochs are given as under: -

Stream Classification Metrics Scores – AlexNet Model							
Ser#	**Scores**	Advertisement	News	Talk Show	Sports	Entertainment	Average
1	Precision	0.947037917	0.996170845	0.990098303	0.997699872	0.992016643	0.984604716
2	Recall	0.995426097	0.98688432	0.984888296	0.957999355	0.984214795	0.981882572
3	F1 Score	0.970629315	0.991505838	0.987486427	0.977446655	0.988100319	0.983033711
4	Jaccard Score	0.942934678	0.983154763	0.975282163	0.955888179	0.976480512	0.966748059

Table 7.1-4: AlexNet Model Stream Classification Metrics Scores

7.2 SqueezeNet Model

SqueezeNet[174]–[176] is a smaller network using compression techniques and making it around 510 times smaller than AlexNet with equivalent accuracy. This network improved training requirements in terms of deployment, bandwidth and communication. This model has been trained on dataset#1 for ten epochs. Results have been shown in succeeding paragraphs.

Following are the hyper parameters used while training of the model: -

Model Training Fine Tuning Hyper Parameters - SqueezeNet Model		
Serial #	**Parameters**	**Values**

1	Number of Epochs	10
2	Learning Rate	0.0001
3	Batch Size	64
4	Seed Value	42

Table 7.2-1: SqueezeNet Model Fine Tuning Hyper Parameters

The overall **SqueezeNet Model** stream classification results on complete **dataset#1** after 10 epochs are as under: -

Stream Classification Overall Loss/ Accuracy – SqueezeNet Model		
Serial #	**Properties**	**Values**
1	Training Loss	0.1206047685492130
2	Training Accuracy	0.9590296656869940
3	Validation Loss	0.0580559055057703
4	Validation Accuracy	0.9811749828532230
5	Testing Accuracy	0.9880615569272977

Table 7.2-2: SqueezeNet Model Stream Classification Overall Training and Testing Results

These are the epoch wise training and validation loss and accuracies results of **SqueezeNet Model** on **dataset#1** for **10 epochs**: -

Training/ Validation Loss/ Accuracy Results Epoch Wise – SqueezeNet Model				
Epoch	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuracy**
1	0.306876987903423	0.892994581190301	0.109647283299579	0.965202760631001
2	0.158860057690686	0.946413482733284	0.075353278485299	0.975019290123456
3	0.125495842800470	0.957871050698016	0.061091964172647	0.980452674897119
4	0.109529984840594	0.963115356355620	0.053165372158629	0.983164008916323
5	0.098674704972355	0.966731722997795	0.051580608030888	0.982456704389574
6	0.089943550518916	0.969744213813372	0.046406141866632	0.984149948559670
7	0.084351889078385	0.971684423218221	0.048915420181238	0.983903463648834
8	0.080887831644811	0.972543166789125	0.045289224706841	0.985928926611797
9	0.076850016792048	0.974244581190301	0.050468810567964	0.984096364883401
10	0.074576819250439	0.974954077883908	0.038640951587986	0.987375685871056

Table 7.2-3: SqueezeNet Model Training Epoch Wise Results

Representation of above epoch wise training and validation loss results in the form of graph is given below: -

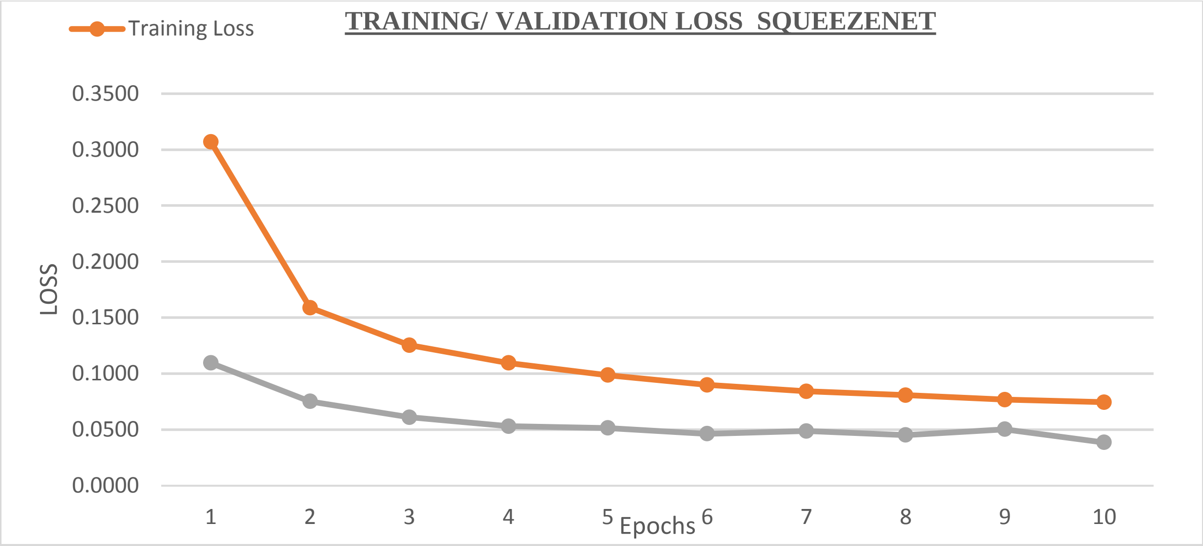


Figure 7.2.1: SqueezeNet Training and Validation Loss

Representation of above epoch wise training and validation Accuracies results in the form of graph is given below: -

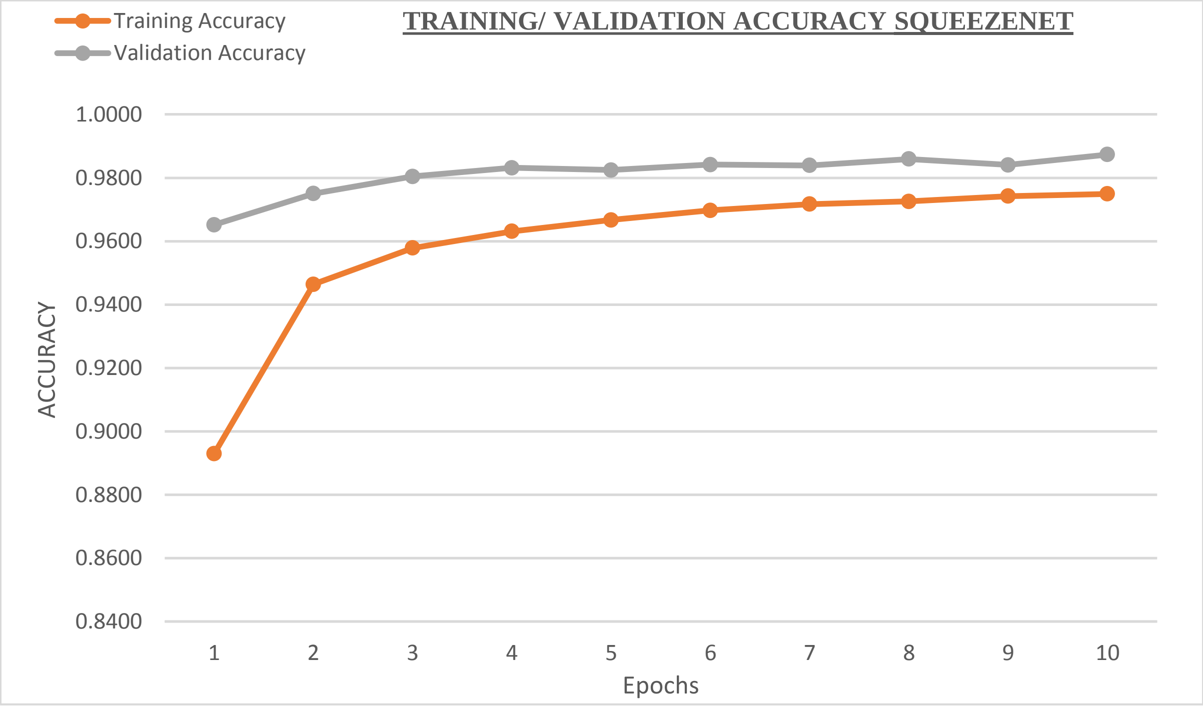


Figure 7.2.2: SqueezeNet Model Training and Validation Accuracies

Confusion matrix formed after training of **SqueezeNet Model** on **dataset#1** for 10 epochs, represents classwise wrong and correct classification results as below: -

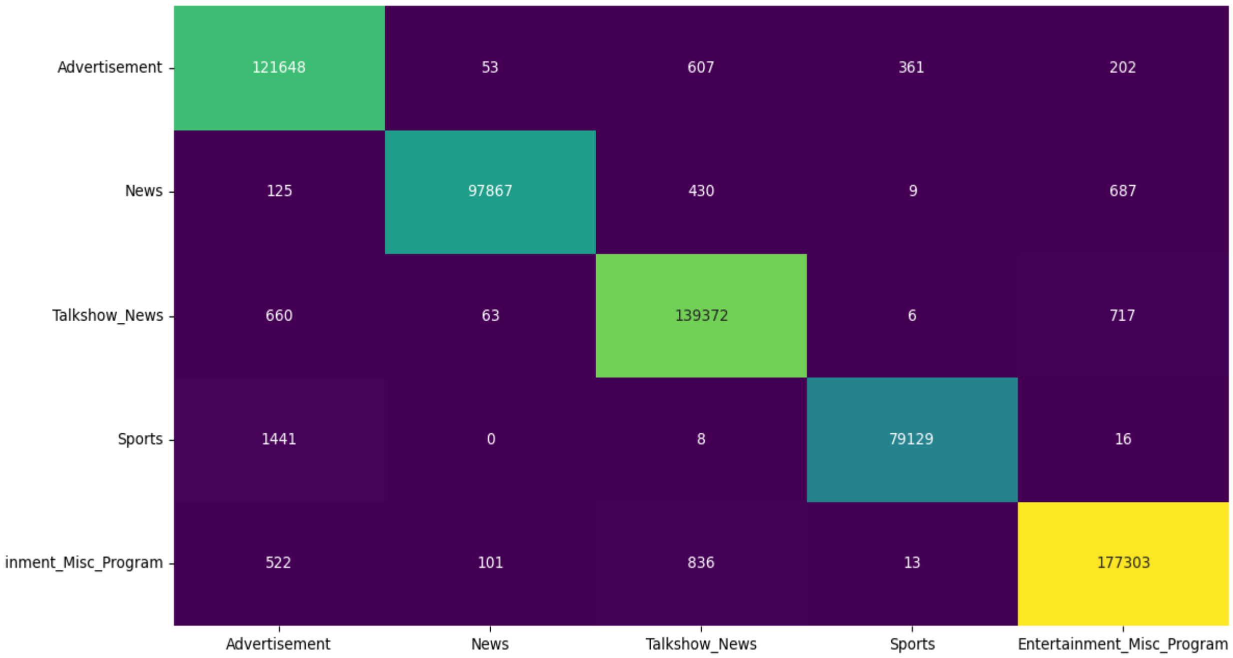


Figure 7.2.3: SqueezeNet Model Confusion Matrix

Four metrics scores calculated for analysis of trained **SqueezeNet Model** on D**ataset#1** for 10 epochs are given as under: -

Stream Classification Metrics Scores – SqueezeNet Model							
Ser#	**Scores**	**Advertisement**	**News**	**Talk Show**	**Sports**	**Entertainment**	**Average**
1	Precision	0.97790925753	0.99778761062	0.98668346867	0.99510802586	0.99093474920	0.98968462237
2	Recall	0.99004647150	0.98737867996	0.98973142638	0.98182246817	0.99176618655	0.98814904651
3	F1 Score	0.98394043686	0.99255585643	0.98820509730	0.98842060558	0.99135029354	0.98889445794
4	Jaccard Score	0.96838853995	0.98522172447	0.97668519051	0.97710630626	0.98284893873	0.97805013998

Table 7.2-4: SqueezeNet Model Stream Classification Metrics Scores

7.3 DenseNet121 Model

DenseNet[177]–[180] surfaced in 2017, is based on shorter connections between layers for accurate and efficient training, so each layer in this network is connected to every other layer in a feed forward pattern. In this way, each layer uses the feature maps of previous layers as inputs. Therefore, these networks reduced the number of parameters significantly and lessened the vanishing gradient problem in earlier networks. So, requires less computation to attain good performance. We trained this network on dataset#1 for 10 epochs and results have been given in the following paragraphs.

Following are the hyper parameters used while training of the model: -

Model Training Fine Tuning Hyper Parameters - DenseNet121 Model		
Serial #	**Parameters**	**Values**
1	Number of Epochs	10
2	Learning Rate	0.0001
3	Batch Size	12
4	Seed Value	42

Table 7.3-1: DenseNet121 Model Fine Tuning Hyper Parameters

The overall **DenseNet121 Model** stream classification results on complete **dataset#1** after 10 epochs are as under: -

Stream Classification Overall Loss/ Accuracy – DenseNet121 Model		
Serial #	**Properties**	**Values**
1	Training Loss	0.0725242403411777
2	Training Accuracy	0.9760125973559840
3	Validation Loss	0.0380437447030747
4	Validation Accuracy	0.9877930679104540
5	Testing Accuracy	0.9903883268383420

Table 7.3-2: DenseNet121 Model Stream Classification Overall Training and Testing Results

These are the epoch wise training and validation loss, and accuracies results of **DenseNet121 Model** on **dataset#1** for **10 epochs**: -

Training/ Validation Loss/ Accuracy Results Epoch Wise – DenseNet121 Model				
Epoch	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuracy**
1	0.1553201363846600	0.9484726265647630	0.0496075761790548	0.9837340914959370
2	0.0878170557188274	0.9712088714866370	0.0423776934993940	0.9863807841516540
3	0.0735112983642084	0.9757506094822240	0.0398009889908419	0.9873023046631110
4	0.0662436011063040	0.9779617791909230	0.0392527975694769	0.9876023346175720

5	0.0659169217492907	0.9781041390489920	0.0385151910447829	0.9879130798884060
6	0.0596878366393791	0.9803933770468770	0.0359524314701199	0.9889203232374260
7	0.0564668606465024	0.9812934585842520	0.0441968070591405	0.9855771325725010
8	0.0560024725172112	0.9814863331923650	0.0291161280598341	0.9901311583026630
9	0.0531661282999134	0.9823083464175620	0.0314572692315971	0.9903561807742560
10	0.0511100919854807	0.9831464325452460	0.0301605639265049	0.9900132894010150

Table 7.3-3: DenseNet121 Model Training Epoch Wise Results

Representation of above epoch wise training and validation loss results in the form of graph is given below: -

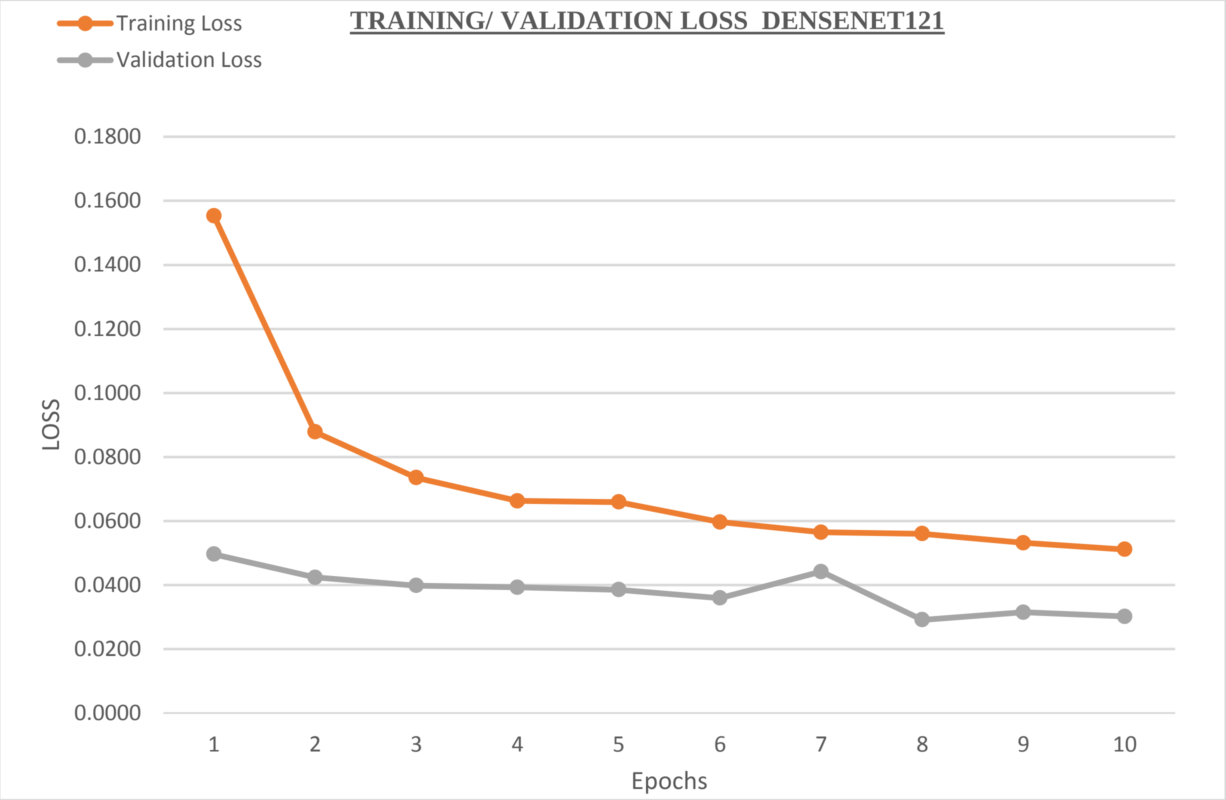


Figure 7.3.1: DenseNet121 Training and Validation Loss

Representation of above epoch wise training and validation Accuracies results in the form of graph is given below: -

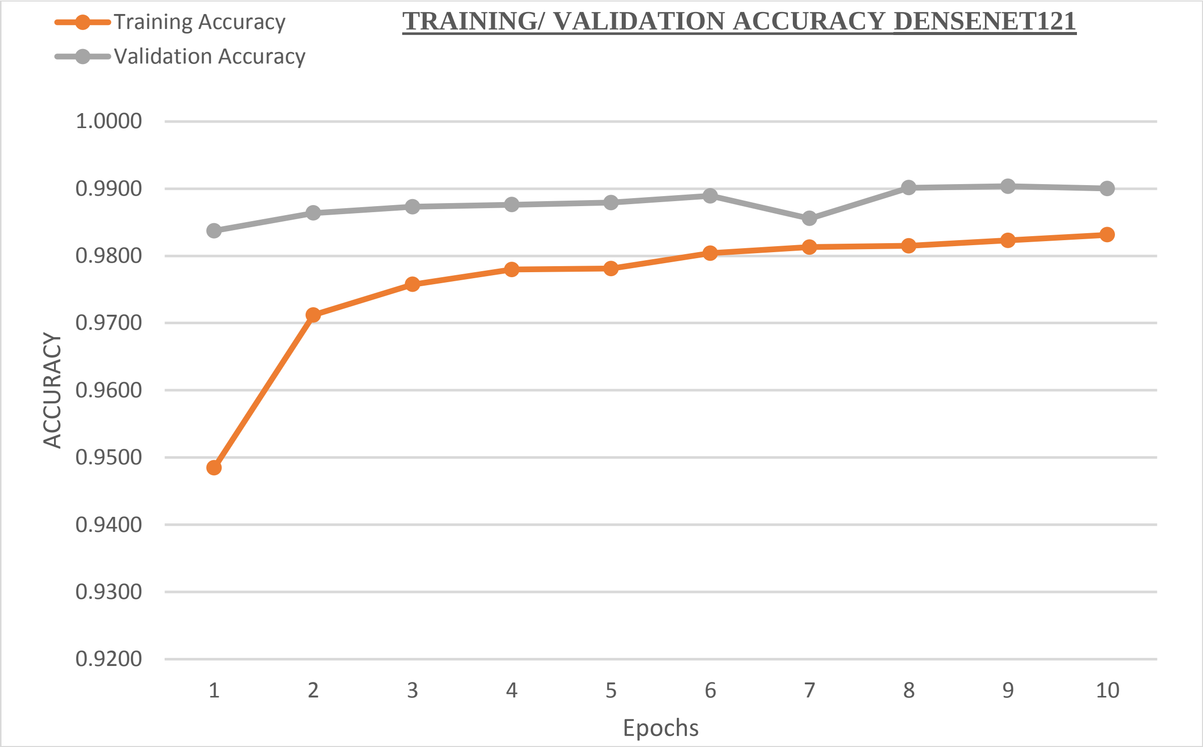
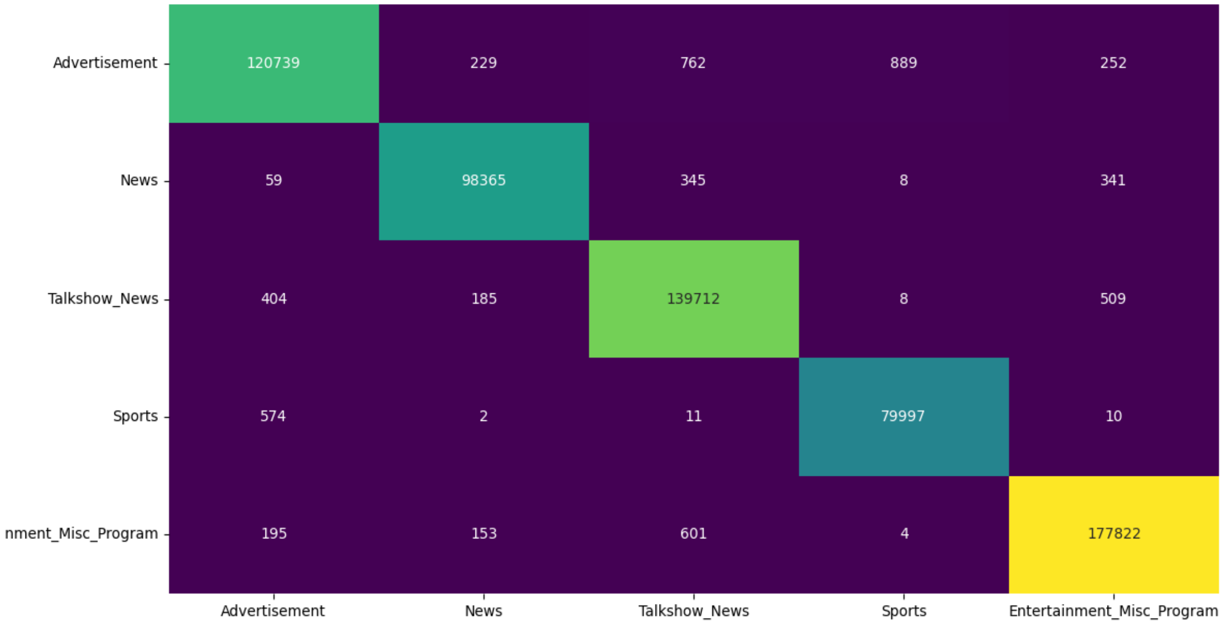


Figure 7.3.2: DenseNet121 Model Training and Validation Accuracies

Confusion matrix formed after training of **DenseNet121 Model** on **dataset#1** for 10 epochs, represents classwise wrong and correct classification results as below: -



	Advertisement	News	Talkshow_News	Sports	Entertainment_Misc_Program
Advertisement	120739	229	762	889	252
News	59	98365	345	8	341
Talkshow_News	404	185	139712	8	509
Sports	574	2	11	79997	10
nment_Misc_Program	195	153	601	4	177822

Figure 7.3.3: DenseNet121 Model Confusion Matrix

Four metrics scores calculated for analysis of trained **DenseNet121 Model** on **dataset#1** for 10 epochs are given as under: -

Stream Classification Metrics Scores – DenseNet121 Model							
Ser#	**Scores**	**Advertisement**	**News**	**Talk Show**	**Sports**	**Entertainment**	**Average**
1	Precision	0.98989923834	0.99424869105	0.98784566326	0.98876473933	0.98876473933	0.99090875001
2	Recall	0.98264846872	0.99240299441	0.99214589044	0.99259250068	0.99466927702	0.99089182626
3	F1 Score	0.98626052720	0.99332498536	0.98999110714	0.99067492260	0.99422715112	0.99089573868
4	Jaccard Score	0.97289348364	0.98673849148	0.98018058469	0.98152215256	0.98852057125	0.98197105672

Table 7.3-4: DenseNet121 Model Stream Classification Metrics Scores

7.4 VGG11 Model

VGG[35], [181], [182] uses convolutional networks by increasing depth of 16-19 layers using small filters 3x3 to achieve accuracy in large scale image recognition requirements. We have trained this model using dataset#1 for 10 epoch and results have been given accordingly.

Following are the hyper parameters used while training of the model: -

Model Training Fine Tuning Hyper Parameters - VGG11 Model		
Serial #	**Parameters**	**Values**
1	Number of Epochs	10
2	Learning Rate	0.0001
3	Batch Size	16
4	Seed Value	42

Table 7.4-1: VGG11 Model Fine Tuning Hyper Parameters

The overall **VGG11 Model** stream classification results on complete **dataset#1** after 10 epochs are as under: -

Stream Classification Overall Loss/ Accuracy – VGG11 Model		
Serial #	**Properties**	**Values**
1	Training Loss	0.0882741203715991
2	Training Accuracy	0.9712539033798670
3	Validation Loss	0.0474953885723813
4	Validation Accuracy	0.9848885555079290
5	Testing Accuracy	0.9890377196742391

Table 7.4-2: VGG11 Model Stream Classification Overall Training and Testing Results

These are the epoch wise training and validation loss, and accuracies results of **VGG11 Model** on **dataset#1** for **10 epochs**: -

Training/ Validation Loss/ Accuracy Results Epoch Wise – VGG11 Model				
Epoch	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuracy**
1	0.1594306770479480	0.9468910727406310	0.0794082141154717	0.9743677668238320
2	0.1024100133888390	0.9665067046289490	0.0553800988190695	0.9821581654522070
3	0.0912881805150115	0.9703756429096250	0.0553974154906332	0.9820081440205740
4	0.0852969675493041	0.9723479977957380	0.0479593908481970	0.9844406343763390
5	0.0796854141858276	0.9741573291697280	0.0427707624758085	0.9865302186026570
6	0.0765242247490365	0.9752273144746500	0.0388515081261575	0.9877196742391770
7	0.0747988841437383	0.9758954812637760	0.0437997883461588	0.9860158594084860
8	0.0724680478825616	0.9768001469507710	0.0403160355023743	0.9875160737248170
9	0.0709197422745205	0.9770733835415130	0.0361307849749470	0.9888662666095150
10	0.0699190519792042	0.9772639603232910	0.0349398870249957	0.9892627518216880

Table 7.4-3: VGG11 Model Training Epoch Wise Results

Representation of above epoch wise training and validation loss results in the form of graph is given below: -

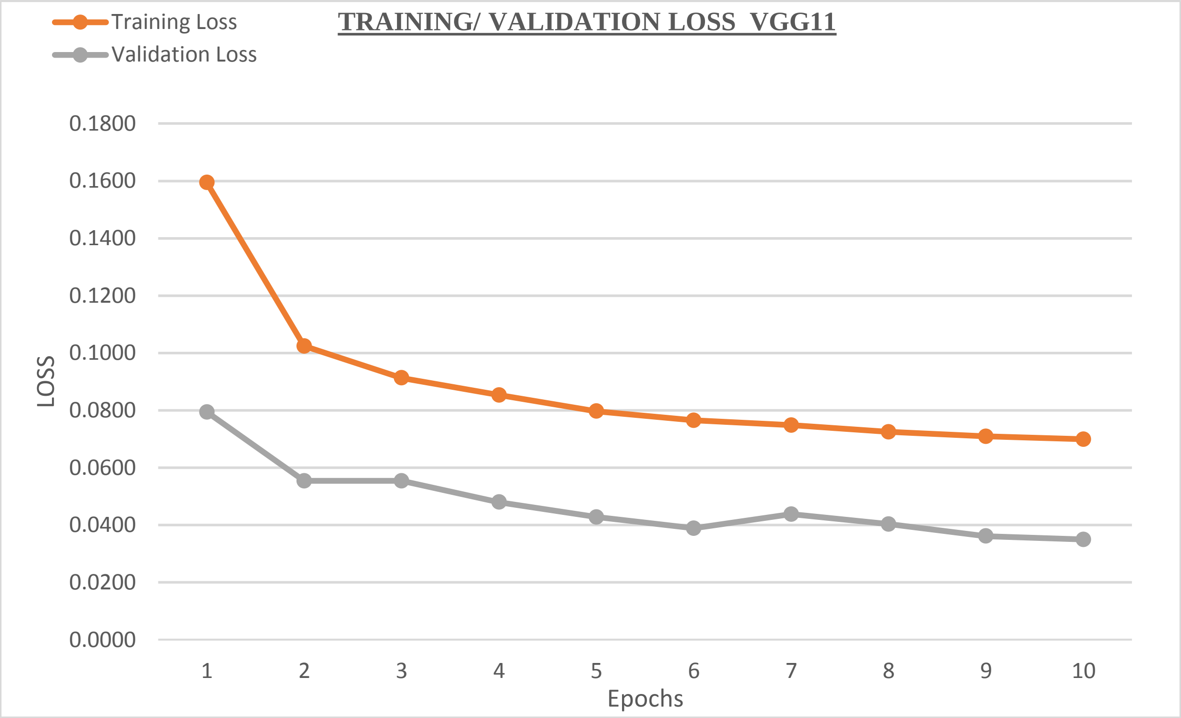


Figure 7.4.1: VGG11 Training and Validation Loss

Representation of above epoch wise training and validation Accuracies results in the form of graph is given below: -

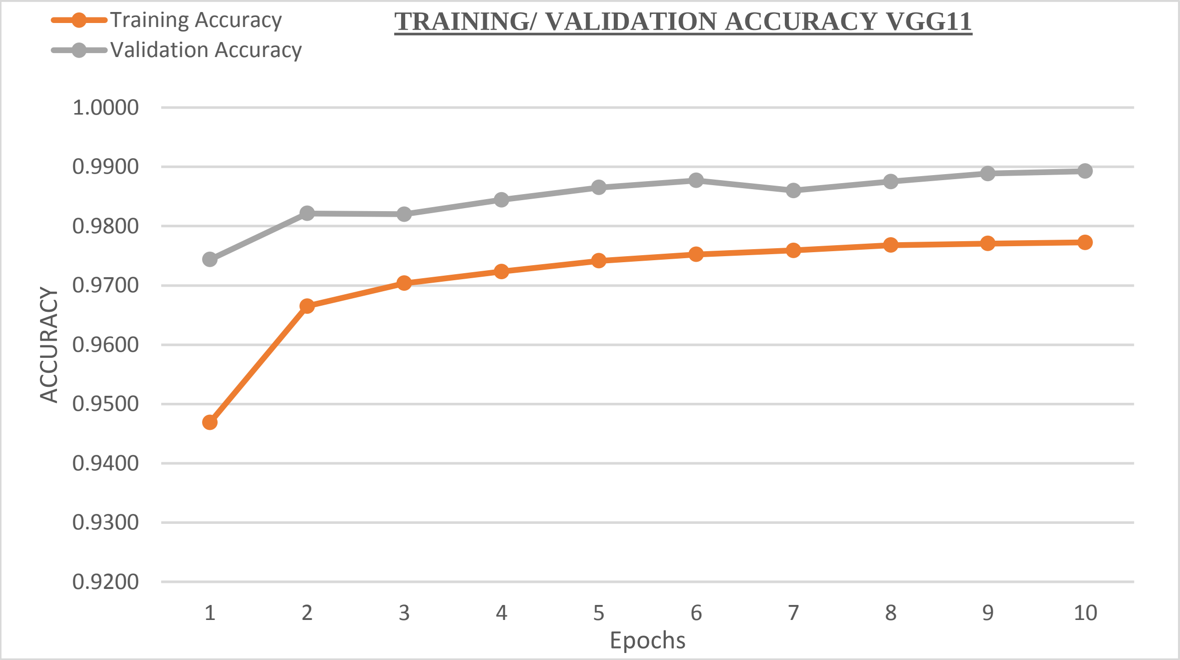


Figure 7.4.2: VGG11 Model Training and Validation Accuracies

Confusion matrix formed after training of **VGG11 Model** on **dataset#1** for 10 epochs, represents classwise wrong and correct classification results as below: -

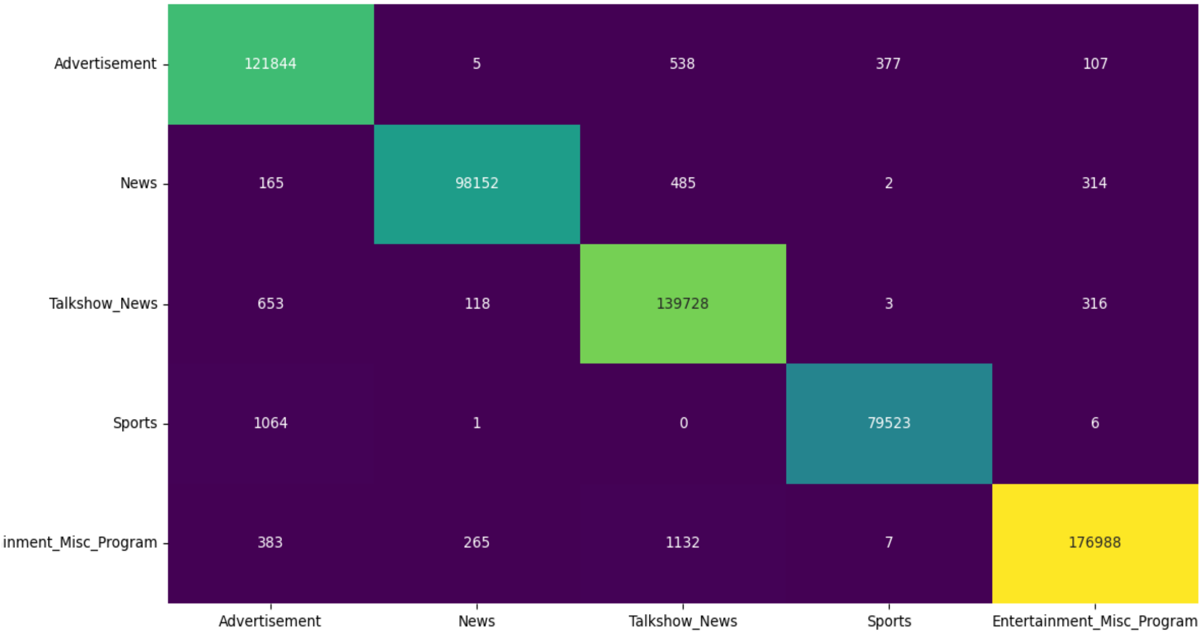


Figure 7.4.3: VGG11 Model Confusion Matrix

Four metrics scores calculated for analysis of trained **VGG11 Model** on **dataset#1** for 10 epochs are given as under: -

Stream Classification Metrics Scores – VGG11 Model							
Ser#	**Scores**	Advertisement	News	Talk Show	Sports	Entertainment	Average
1	Precision	0.98174991338	0.99605240458	0.98481142914	0.99513214536	0.99581952501	0.99071308349
2	Recall	0.99164164042	0.99025404064	0.99225951228	0.98671116957	0.99000419522	0.99017411162
3	F1 Score	0.98667098550	0.99314475941	0.98852144138	0.99090376684	0.99290334525	0.99042885968
4	Jaccard Score	0.97369262243	0.98638286754	0.97730340694	0.98197152489	0.98590670573	0.98105142550

Table 7.4-4: VGG11 Model Stream Classification Metrics Scores

7.5 ResNet Models

ResNets[86], [183] are skip connection models based on formulating layers as learning residual functions with reference to the layer inputs. This arrangement made training of model easier with better optimization and accuracy even at significant enhanced depth. ResNets have various variants based on number of layers, each having its own significance. However, ResNet-18, ResNet-34, ResNet-50, ResNet-101 & ResNet-152 are popular ones meeting various requirements like bigger datasets, complexity of problems etc. We have trained all these five variants and results have been given in subsequent sections.

7.5.1 ResNet18 Model

ResNet18[184] model has been trained on both datasets#1&2 for 10 and 20 epochs respectively and results have been shown ahead.

Following are the hyper parameters used while training of the model: -

Model Training Fine Tuning Hyper Parameters - ResNet18 Model		
Serial #	**Parameters**	**Values**
1	Number of Epochs	10 & 20
2	Learning Rate	0.0001
3	Batch Size	64
4	Seed Value	42

Table 7.5.1-1: ResNet18 Model Fine Tuning Hyper Parameters

The overall **ResNet18 Model** stream classification results on complete **dataset#1** after **10 epochs** are as under: -

Stream Classification Overall Loss/ Accuracy – ResNet18 Model		
Serial #	**Properties**	**Values**
1	Training Loss	0.0504263230578660
2	Training Accuracy	0.9831608192505510
3	Validation Loss	0.0261231881628216
4	Validation Accuracy	0.9912862225651570
5	Testing Accuracy	0.9926161694101509

Table 7.5.1-2: ResNet18 Model Stream Classification Overall Training and Testing Results

These are the epoch wise training and validation loss, and accuracies results of **ResNet18 Model** on **dataset#1** for **10 epochs**: -

Training/ Validation Loss/ Accuracy Results Epoch Wise – ResNet18 Model				
Epoch	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuracy**
1	0.1240719677142050	0.9583141991182954	0.0387544453030133	0.9872685185185180
2	0.0623485248975187	0.9792041697281410	0.0308884649793369	0.9893797153635110
3	0.0506197151348363	0.9831052534900800	0.0303056131287756	0.9895511831275720
4	0.0456316422087377	0.9848181484202790	0.0266135084774134	0.9914373285322360
5	0.0415536626946324	0.9863083210874350	0.0228567388807881	0.9924125514403290
6	0.0386655943232132	0.9870867009551800	0.0245723044200133	0.9921982167352530
7	0.0379190074765410	0.9872795738427620	0.0213503385184170	0.9932055898491080
8	0.0354340472584778	0.9882990448199850	0.0213503385184170	0.9926268861454040
9	0.0346219506027987	0.9883679279941220	0.0223641068051147	0.9923482510288060
10	0.0333971182676990	0.9888248530492280	0.0221760225969265	0.9924339849108360

Table 7.5.1-3: ResNet18 Model Training Epoch Wise Results

Representation of above epoch wise training and validation loss results in the form of graph is given below: -

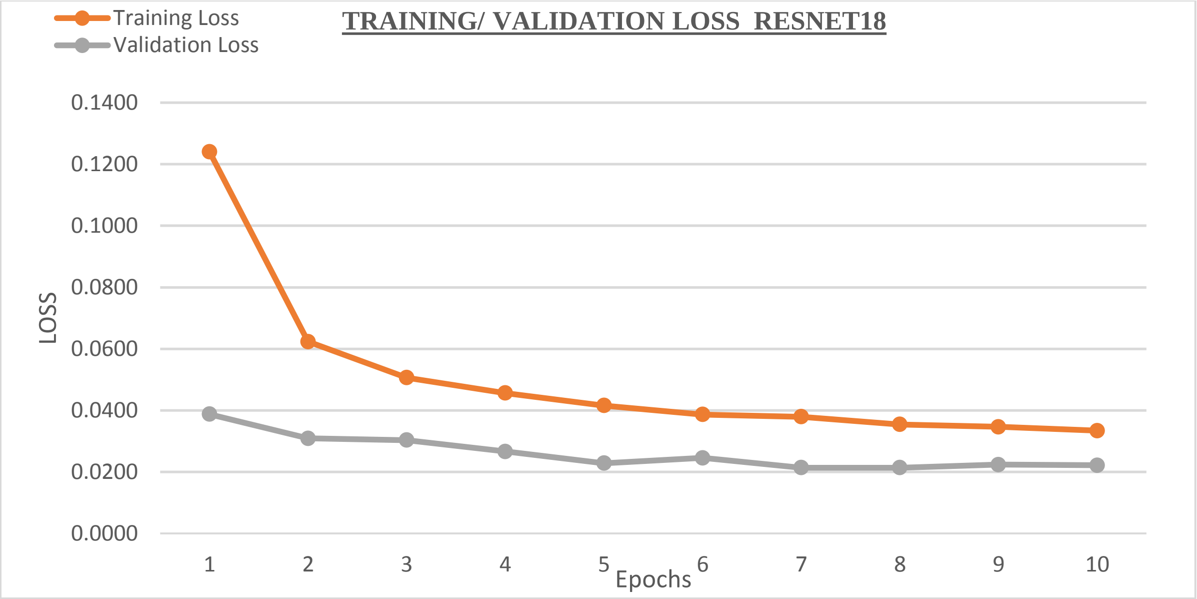


Figure 7.5.1.1: ResNet18 Training and Validation Loss

Representation of above epoch wise training and validation Accuracies results in the form of graph is given below: -

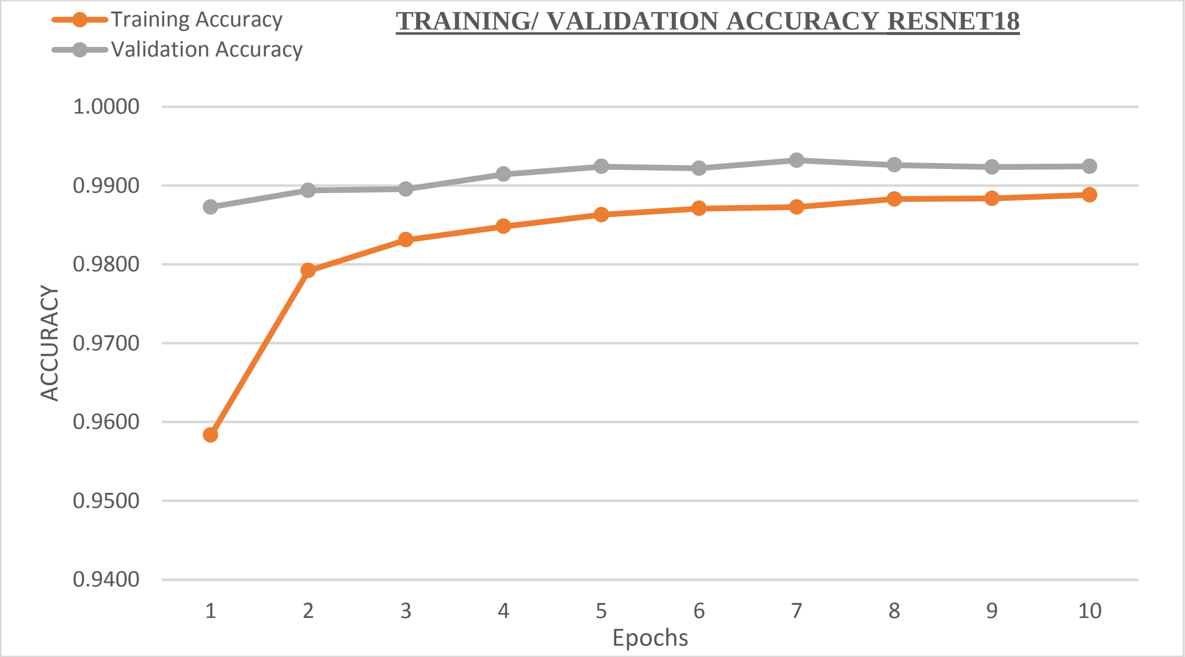


Figure 7.5.1.2: ResNet18 Model Training and Validation Accuracies

Confusion matrix formed after training of **ResNet18 Model** on **dataset#1** for 10 epochs, represents classwise wrong and correct classification results as below: -

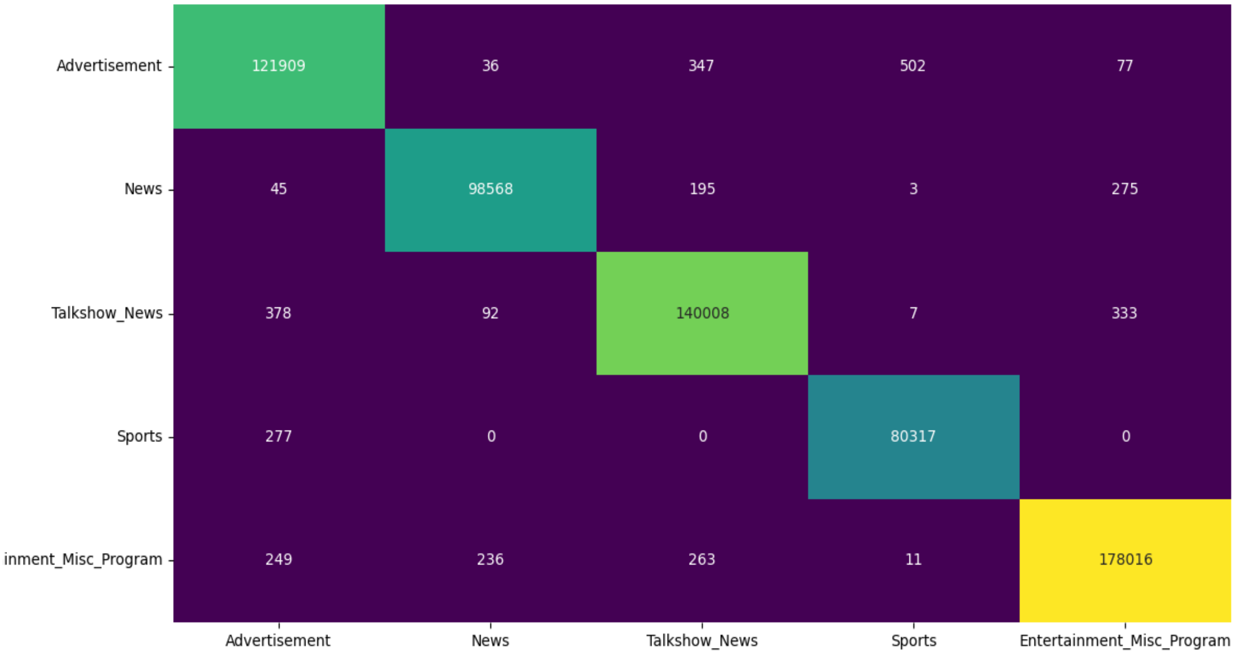


Figure 7.5.1.3: ResNet18 Model Confusion Matrix

Four metrics scores calculated for analysis of trained **ResNet18 Model** on **dataset#1** for 10 epochs are given as under: -

Stream Classification Metrics Scores – ResNet18 Model							
Ser#	**Scores**	Advertisement	News	Talk Show	Sports	Entertainment	Average
1	Precision	0.99227563529	0.99632070513	0.99428319828	0.99353043048	0.99616678138	0.99451535011
2	Recall	0.99217065052	0.99477221807	0.99424789445	0.99656301958	0.99575443994	0.99470164451
3	F1 Score	0.99222314013	0.99554585947	0.99426554605	0.99504441444	0.99596056798	0.99460790561
4	Jaccard Score	0.98456630593	0.99113122172	0.98859648503	0.99013770233	0.99195363869	0.98927707074

Table 7.5.1-4: ResNet18 Model Stream Classification Metrics Scores

The overall **ResNet18 Model** stream classification results on complete **dataset#2** after **10 epochs** are as under: -

Stream Classification Overall Loss/ Accuracy – ResNet18 Model_100		
Serial #	**Properties**	**Values**
1	Training Loss	0.0571434060007832
2	Training Accuracy	0.9809642384592760
3	Validation Loss	0.0285342789005563
4	Validation Accuracy	0.9904813464837050
5	Testing Accuracy	0.9931818181818182

Table 7.5.1-5: ResNet18 Model Stream Classification Overall Training and Testing Results

These are the epoch wise training and validation loss, and accuracies results of **ResNet18 Model** on **dataset#2** for **10 epochs**: -

Training/ Validation Loss/ Accuracy Results Epoch Wise – ResNet18 Model_100				
Epoch	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuracy**
1	0.129724112768584	0.956216921493678	0.041443350287625	0.986374356775300
2	0.070061059995760	0.976833559982358	0.035245190546550	0.988368353344768
3	0.058922236655579	0.980433144663334	0.029268613875302	0.990212264150943
4	0.052599259128401	0.982569464863275	0.033721893536479	0.988475557461406
5	0.048314430978928	0.983956924433990	0.025275582547904	0.991509433962264
6	0.046275626670706	0.984657545574831	0.027910517304990	0.990898370497427
7	0.044130380735823	0.985484508232872	0.024915868685602	0.991316466552315
8	0.042836104847573	0.985636118053513	0.024519974579000	0.991788164665523
9	0.039285424113239	0.986927098647456	0.021520898821056	0.992935248713550
10	0.039285424113239	0.986927098647456	0.021520898821056	0.992935248713550

Table 7.5.1-6: ResNet18 Model Training with Dataset 100% Quality Epoch Wise Results

Representation of above epoch wise training and validation loss results in the form of graph is given below: -

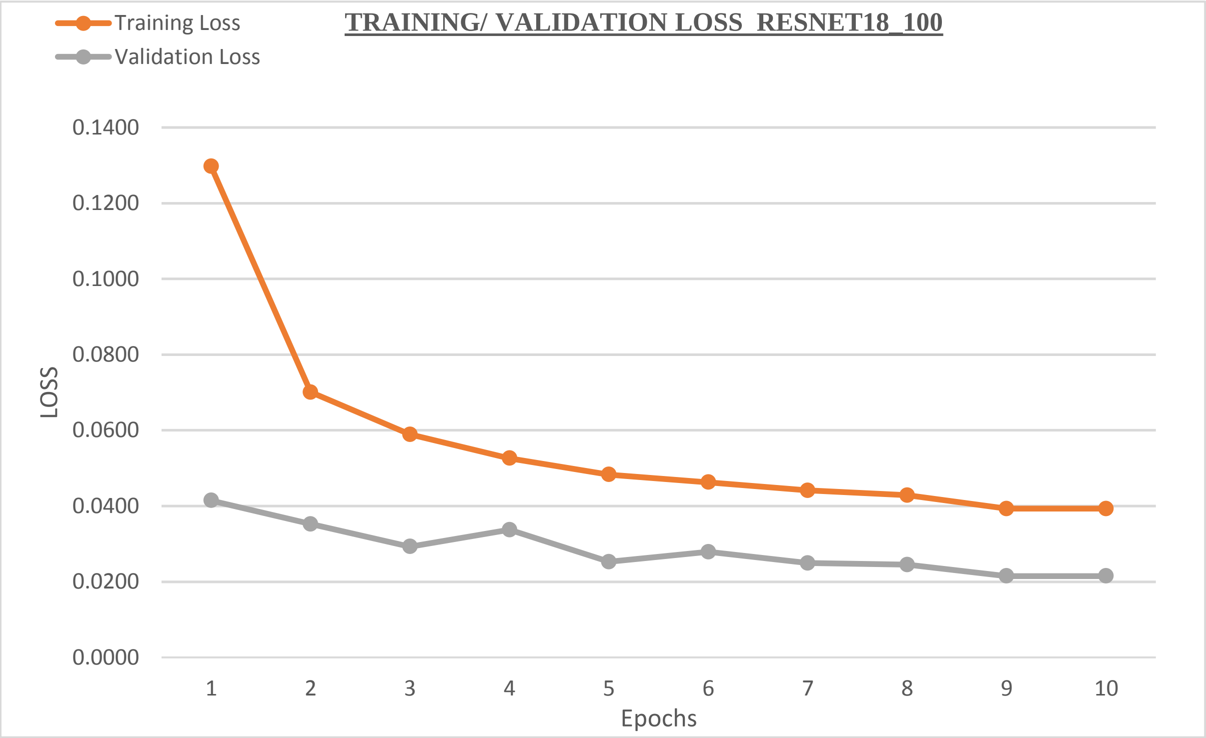


Figure 7.5.1.4: ResNet18 Model Training and Validation Loss

Representation of above epoch wise training and validation Accuracies results in the form of graph is given below: -

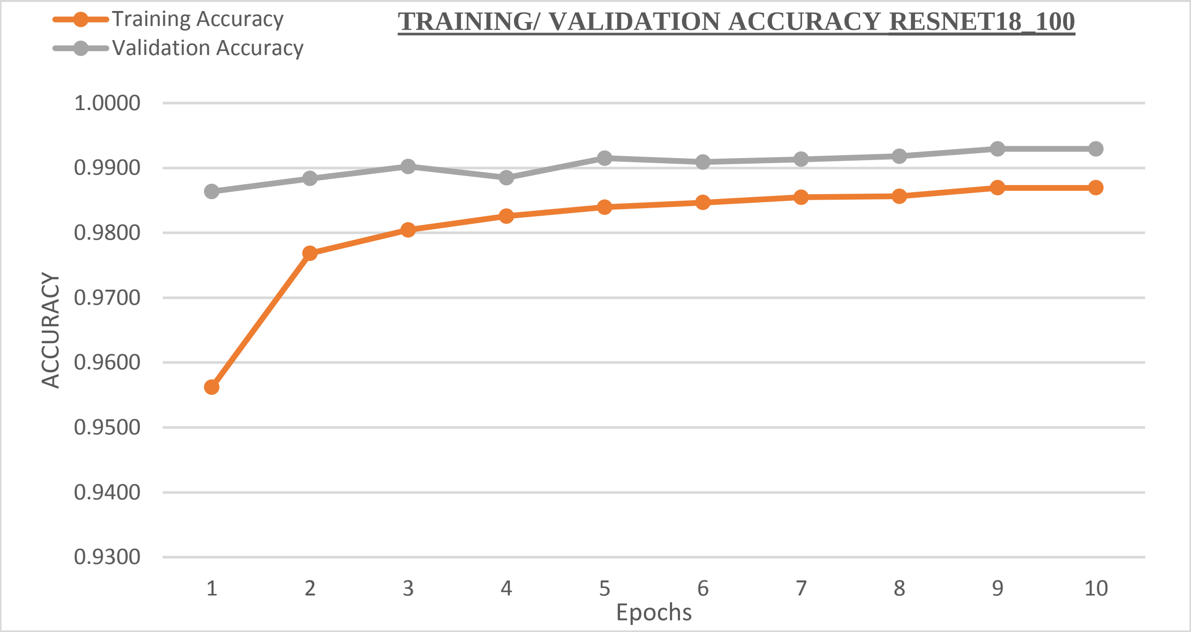


Figure 7.5.1.5: ResNet18 Model Training and Validation Accuracies

Confusion matrix formed after training of **ResNet18 Model** on **dataset#2** for 10 epochs, represents classwise wrong and correct classification results as below: -

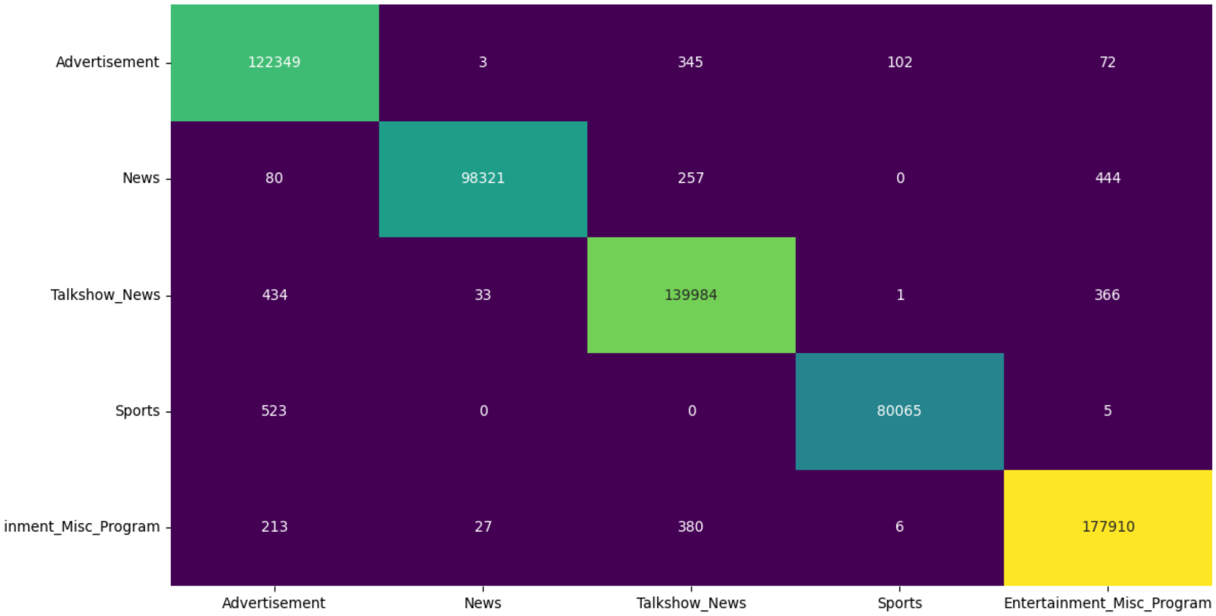


Figure 7.5.1.6: ResNet18 Model Confusion Matrix

Four metrics scores calculated for analysis of trained **ResNet18 Model** on **dataset#2** for 10 epochs are given as under: -

Stream Classification Metrics Scores – ResNet18 Model_100							
Ser#	**Scores**	Advertisement	News	Talk Show	Sports	Entertainment	Average
1	Precision	0.98988664957	0.99935965198	0.99303378120	0.998640457701	0.99503906665	0.99519192128
2	Recall	0.99575164197	0.99211923069	0.99407746169	0.993448562530	0.99649370435	0.99437812025
3	F1 Score	0.99281048403	0.99572627933	0.99355534736	0.996037744060	0.99576585426	0.99477914181
4	Jaccard Score	0.98572360841	0.99148893259	0.98719322990	0.992106763150	0.99156741332	0.98961598948

Table 7.5.1-7: ResNet18 Model Stream Classification Metrics Scores

The overall **ResNet18 Model** stream classification results on complete **dataset#2** after **20 epochs** are as under: -

Stream Classification Overall Loss/ Accuracy – ResNet18 Model_20Ep		
Serial #	**Properties**	**Values**
1	Training Loss	0.0467448729830174
2	Training Accuracy	0.9843925729564830
3	Validation Loss	0.0253611377966759
4	Validation Accuracy	0.9916177101200680
5	Testing Accuracy	0.9927208404802744

Table 7.5.1-8: ResNet18 Model Stream Classification Overall Training and Testing Results

These are the epoch wise training and validation loss, and accuracies results of **ResNet18 Model** on **dataset#2** for **20 epochs**: -

Training/ Validation Loss/ Accuracy Results Epoch Wise – ResNet18 Model_100_20Ep				
Epoch	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuracy**
1	0.129724112768584	0.956216921493678	0.041443350287625	0.986374356775300
2	0.070061059995760	0.976833559982358	0.035245190546550	0.988368353344768
3	0.058922236655579	0.980433144663334	0.029268613875302	0.990212264150943
4	0.052599259128401	0.982569464863275	0.033721893536479	0.988475557461406
5	0.048314430978928	0.983956924433990	0.025275582547904	0.991509433962264
6	0.046275626670706	0.984657545574831	0.027910517304990	0.990898370497427
7	0.044130380735823	0.985484508232872	0.024915868685602	0.991316466552315
8	0.042836104847573	0.985636118053513	0.024519974579000	0.991788164665523
9	0.040844129797573	0.986334442075860	0.023732105830554	0.992174099485420

10	0.040011409896805	0.986603204939723	0.025080976429916	0.991252144082332
11	0.038755787177084	0.987014389150250	0.021848817943806	0.992913807890223
12	0.038438486194999	0.987090194060570	0.022310961414347	0.992774442538593
13	0.037461888653951	0.987443950308732	0.021877156313850	0.992924528301886
14	0.036542518087168	0.987848243163775	0.022104891225295	0.992774442538593
15	0.035960705251074	0.987765546897971	0.021465981142265	0.992924528301886
16	0.035627871513501	0.988068766539253	0.022496569053971	0.992688679245283
17	0.035207822473901	0.988261724492796	0.020342560387513	0.993171097770154
18	0.034933706982409	0.988459276683328	0.020646111049491	0.993449828473413
19	0.034159565031159	0.988654531755366	0.020996528064141	0.993235420240137
20	0.034090356819370	0.988519001764187	0.022019105714919	0.993128216123499

Table 7.5.1-9: ResNet18 Model Training with Dataset 100 % Quality Epoch Wise Results

Representation of above epoch wise training and validation loss results in the form of graph is given below: -

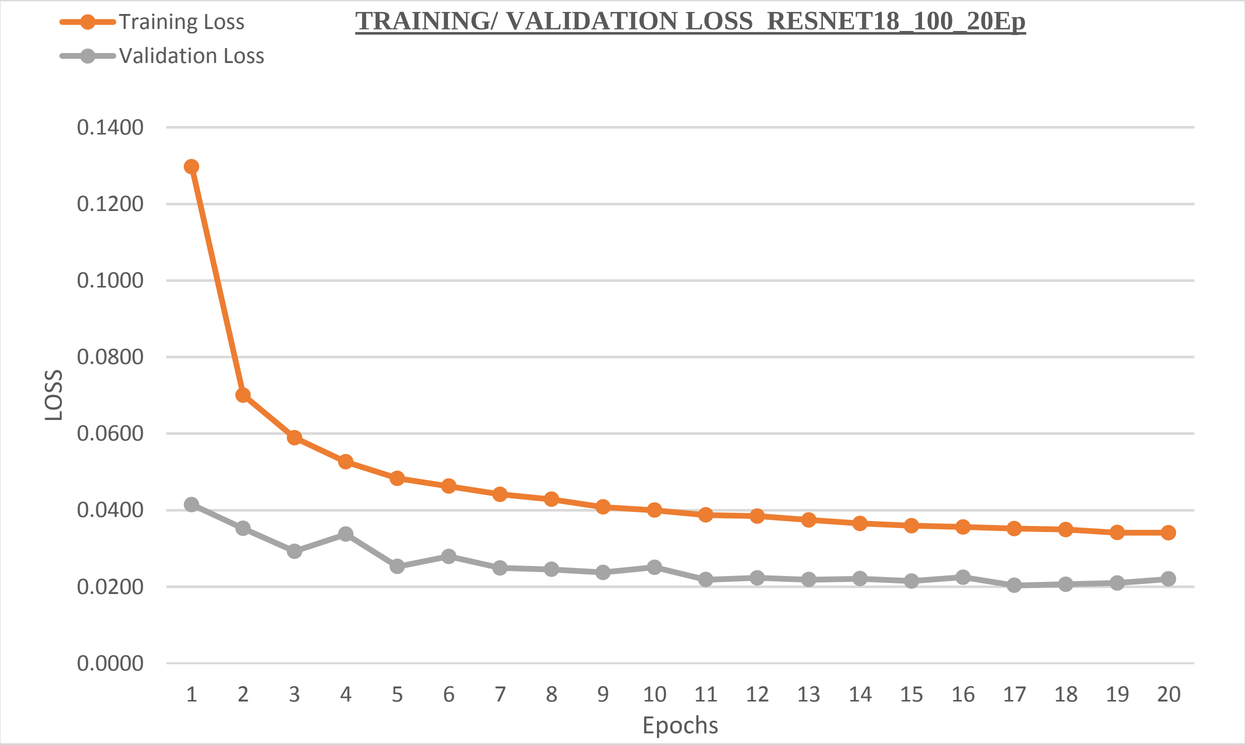


Figure 7.5.1.7: ResNet18 Model Training and Validation Loss

Representation of above epoch wise training and validation Accuracies results in the form of graph is given below: -

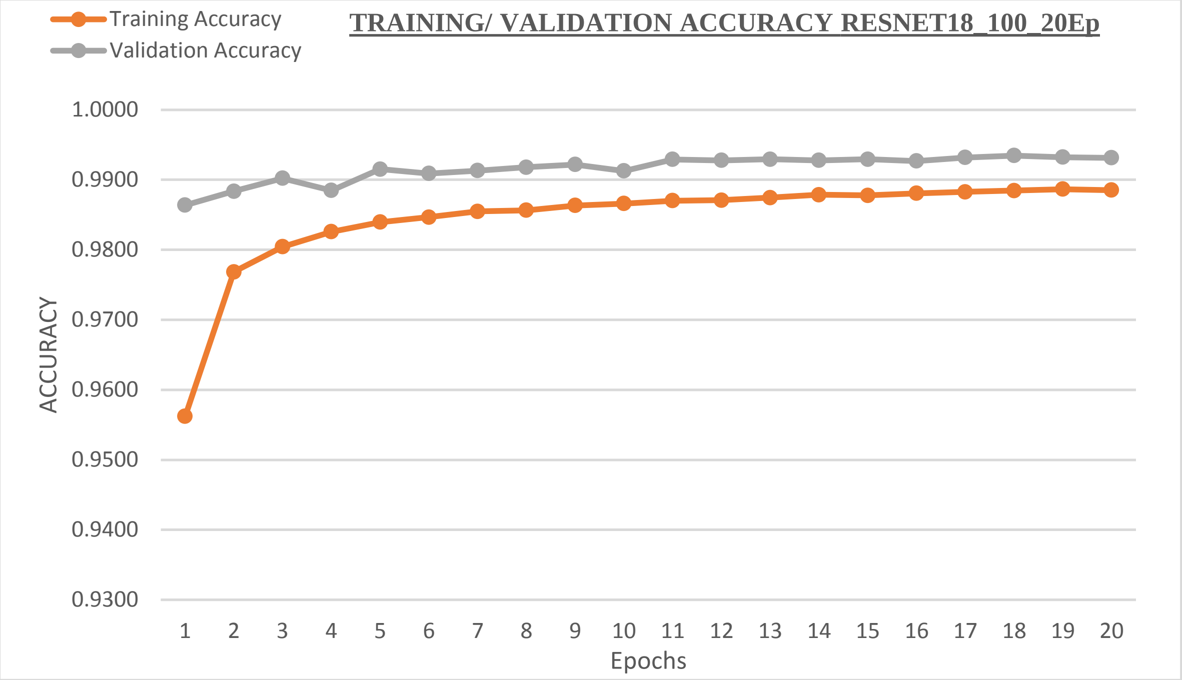
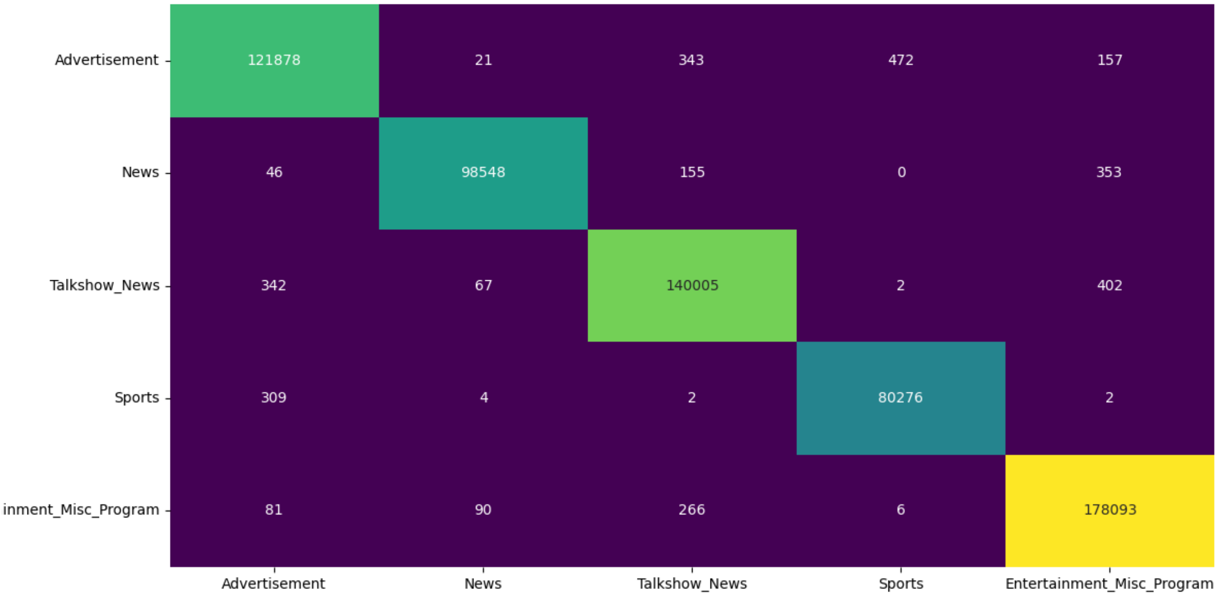


Figure 7.5.1.8: ResNet18 Model Training and Validation Accuracies of 20 Epochs on 2nd Dataset

Confusion matrix formed after training of **ResNet18 Model** on **dataset#2** for 20 epochs, represents classwise wrong and correct classification results as below: -



	Advertisement	News	Talkshow_News	Sports	Entertainment_Misc_Program
Advertisement	121878	21	343	472	157
News	46	98548	155	0	353
Talkshow_News	342	67	140005	2	402
Sports	309	4	2	80276	2
inment_Misc_Program	81	90	266	6	178093

Figure 7.5.1.9: ResNet18 Model Confusion Matrix

Four metrics scores calculated for analysis of trained **ResNet18 Model** on **dataset#2** for 20 epochs are given as under: -

Stream Classification Metrics Scores – ResNet18 Model_20Ep							
Ser#	**Scores**	Advertisement	News	Talk Show	Sports	Entertainment	Average
1	Precision	0.99365705714	0.99815658868	0.99455853834	0.99405616920	0.99489405442	0.99506448155
2	Recall	0.99191835340	0.99440980000	0.99422659035	0.99606665591	0.99751870771	0.99482802147
3	F1 Score	0.99278694400	0.99627967164	0.99439253664	0.99506039703	0.99620465231	0.99494484032
4	Jaccard Score	0.98567719917	0.99258692236	0.98884760990	0.99016935355	0.99243800502	0.98994381800

Table 7.5.1-10: ResNet18 Model Stream Classification Metrics Scores

7.5.2 ResNet34 Model

ResNet34[185] model has been trained on both datasets#1&2 for 10 and 20 epochs respectively and results have been shown ahead.

Following are the hyper parameters used while training of the model: -

Model Training Fine Tuning Hyper Parameters - ResNet34 Model		
Serial #	**Parameters**	**Values**
1	Number of Epochs	10
2	Learning Rate	0.0001
3	Batch Size	32
4	Seed Value	42

Table 7.5.2-1: ResNet34 Model Fine Tuning Hyper Parameters

The overall **ResNet34 Model** stream classification results on complete **dataset#1** after **11 epochs** are as under: -

Stream Classification Overall Loss/ Accuracy – ResNet34 Model		
Serial #	**Properties**	**Values**
1	Training Loss	0.0614142687210434
2	Training Accuracy	0.9794982800080150
3	Validation Loss	0.0322527945394546
4	Validation Accuracy	0.9893456166604310
5	Testing Accuracy	0.9899905692729767

Table 7.5.2-2: ResNet34 Model Stream Classification Overall Training and Testing Results

These are the epoch wise training and validation loss and accuracies results of **ResNet34 Model** on **dataset#1** for **11 epochs**: -

Training/ Validation Loss/ Accuracy Results Epoch Wise – ResNet34 Model				
Epoch	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuracy**
1	0.136395033805784	0.954231722997795	0.048347074715288	0.984021347736625
2	0.078109086560350	0.974136664217487	0.046510917115839	0.984664351851851
3	0.064998443206370	0.978494673034533	0.034533916646153	0.988468792866941
4	0.058416276866090	0.980708119030124	0.030912784521139	0.989604766803840
5	0.052926121637760	0.982384276267450	0.029388434995691	0.990719307270233
6	0.050252204704483	0.983357825128581	0.028209118774050	0.990247770919067
7	0.047324788784688	0.984081098457016	0.029971885327034	0.990483539094650
8	0.045346700941462	0.984675789860396	0.025444460520709	0.991640946502057
9	0.043506415765221	0.985461058045554	0.026501480387023	0.991233710562414
10	0.042206142949958	0.985830731080088	0.025344334715104	0.991619513031550
11	0.056075740709312	0.981119121969140	0.029616332215972	0.990097736625514

Table 7.5.2-3: ResNet34 Model Training Epoch Wise Results

Representation of above epoch wise training and validation loss results in the form of graph is given below: -

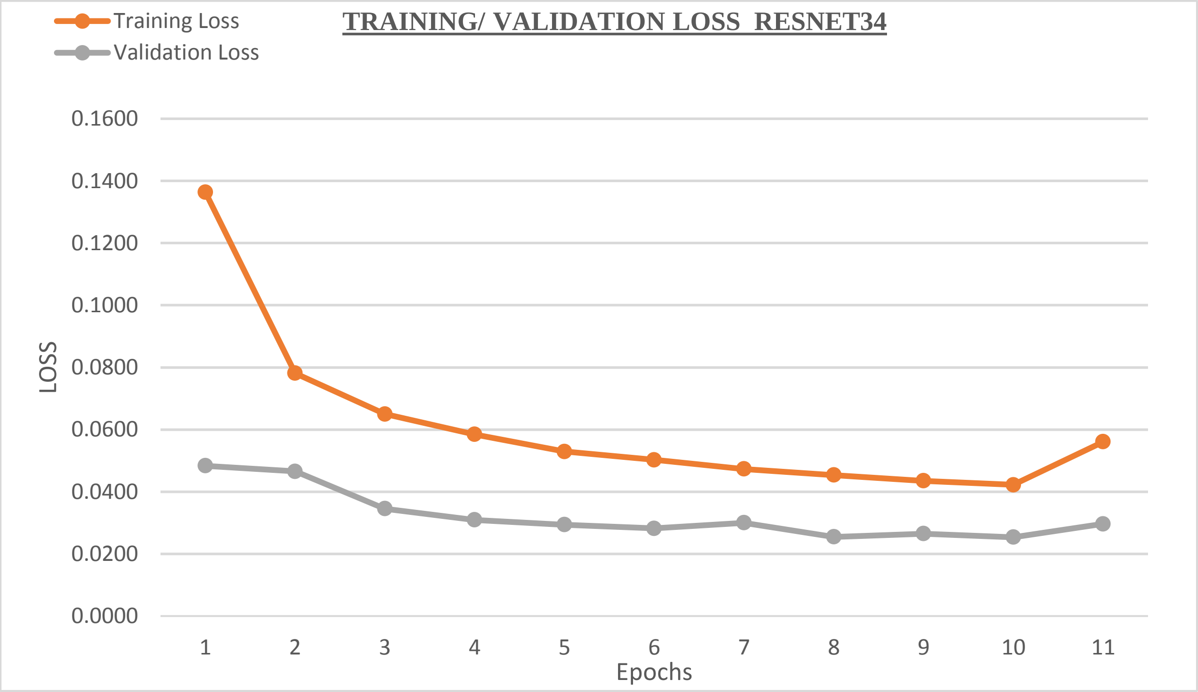


Figure 7.5.2.1: ResNet34 Model Training and Validation Loss

Representation of above epoch wise training and validation Accuracies results in the form of graph is given below: -

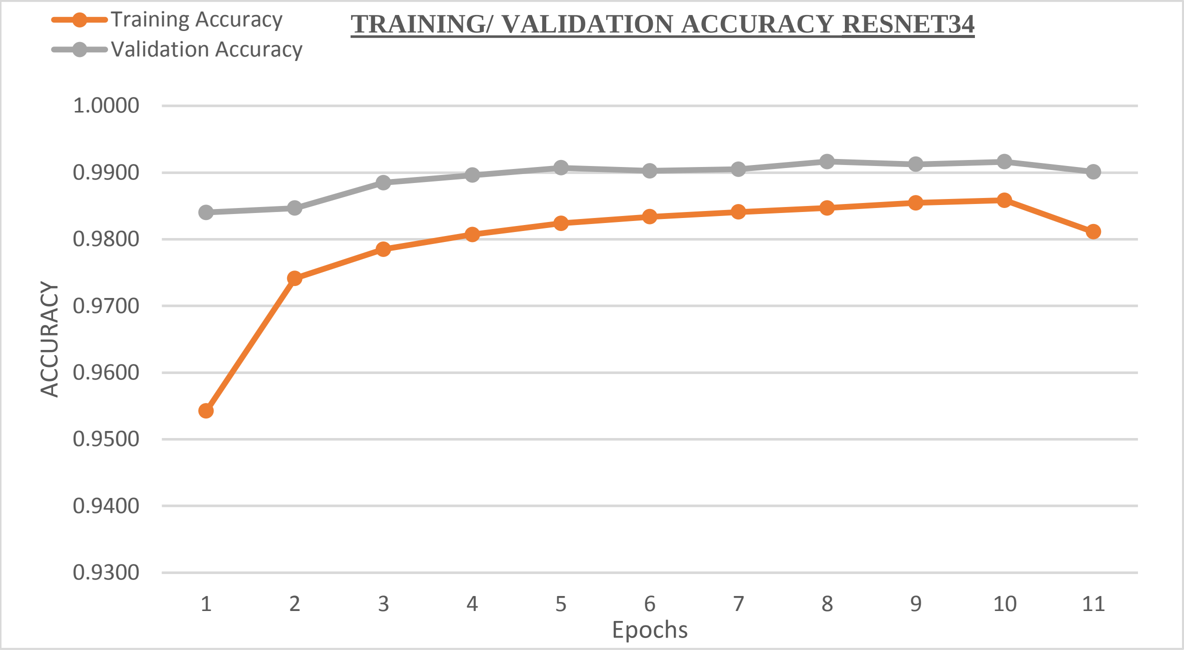


Figure 7.5.2.2: ResNet34 Model Training and Validation Accuracies

Confusion matrix formed after training of **ResNet34 Model** on **dataset#1** for 11 epochs, represents classwise wrong and correct classification results as below: -

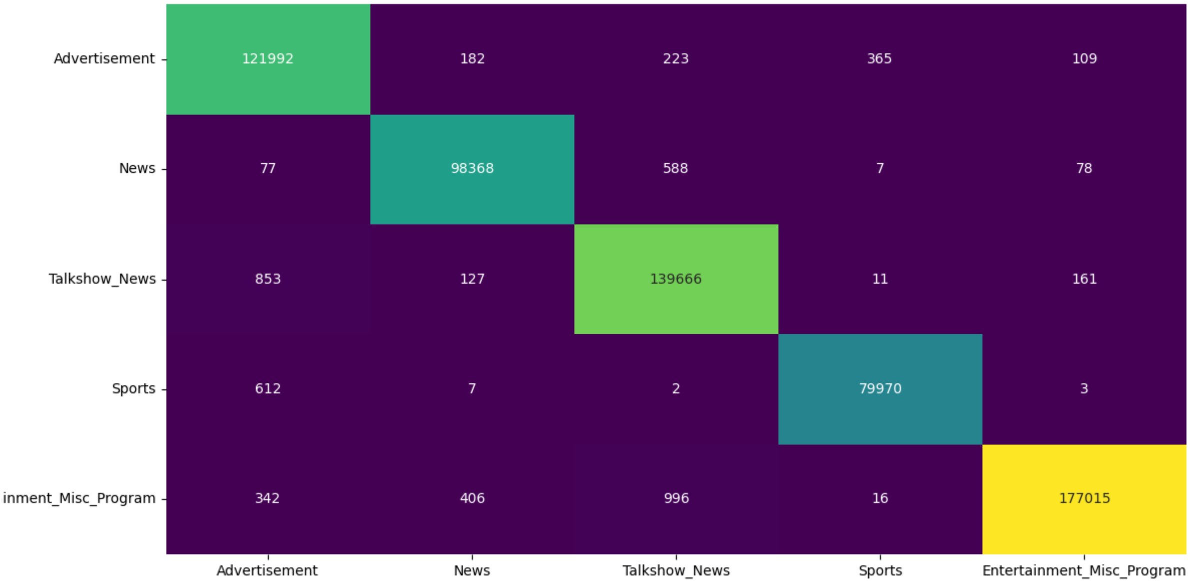


Figure 7.5.2.3: ResNet34 Model Confusion Matrix

Four metrics scores calculated for analysis of trained **ResNet34 Model** on **dataset#1** for 11 epochs are given as under: -

Stream Classification Metrics Scores – ResNet34 Model							
Ser#	**Scores**	Advertisement	News	Talk Show	Sports	Entertainment	Average
1	Precision	0.98479124286	0.99271369462	0.98721328857	0.99503539922	0.99802104124	0.99155493330
2	Recall	0.99284615572	0.99243326137	0.99181922766	0.99225748815	0.99015522305	0.99190227119
3	F1 Score	0.98880229547	0.99257345819	0.98951089825	0.99364450215	0.99407257238	0.99172074529
4	Jaccard Score	0.97785259108	0.98525641026	0.97923955492	0.98736927883	0.98821499950	0.98358656692

Table 7.5.2-4: ResNet34 Model Stream Classification Metrics Scores

The overall **ResNet34 Model** stream classification results on complete **dataset#2** after **20 epochs** are as under: -

Stream Classification Overall Loss/ Accuracy – ResNet34 Model_20Ep		
Serial #	**Properties**	**Values**
1	Training Loss	0.0485148779176120
2	Training Accuracy	0.9838053146133490
3	Validation Loss	0.0265700222288340
4	Validation Accuracy	0.9912419596912520
5	Testing Accuracy	0.9929245283018868

Table 7.5.2-5: ResNet34 Model Stream Classification Overall Training and Testing Results

These are the epoch wise training and validation loss and accuracies results of **ResNet34 Model** on **dataset#2** for **20 epochs**: -

Training/ Validation Loss/ Accuracy Results Epoch Wise – ResNet34 Model_100				
Epoch	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuracy**
1	0.1319778441280150	0.9554381983240220	0.0476191554970637	0.9850664665523150
2	0.0754174828490746	0.9750050536606880	0.0379079656841993	0.9876179245283010
3	0.0631804487792736	0.9791812150837980	0.0327619871560883	0.9891187821612340
4	0.0559029126617673	0.9814668479858860	0.0296854729986161	0.9901801029159510
5	0.0516824642099187	0.9829691634813290	0.0305223696441388	0.9896333619210970
6	0.0487346513588538	0.9838236915613050	0.0274659609730673	0.9910270154373920
7	0.0458500478254410	0.9848849603057920	0.0290992271946388	0.9906303602058320
8	0.0441883460687562	0.9852203396059980	0.0255586748724504	0.9917024013722120
9	0.0424412141383602	0.9859301492208170	0.0257281908877569	0.9914879931389360
10	0.0414931762288899	0.9861460783593060	0.0225210153929370	0.9924635506003430

11	0.0399338339265288	0.9867111695089680	0.0236374063916598	0.9920668953687820
12	0.0392486877951315	0.9869638525433690	0.0217501186610726	0.9926243567753000
13	0.0390673322488783	0.9868260254336960	0.0217494746931237	0.9927958833619210
14	0.0375159682023990	0.9873727396354010	0.0247784457208136	0.9916595197255570
15	0.0367039913241819	0.9876369082622750	0.0213010030039568	0.9929138078902230
16	0.0362298105332535	0.9877747353719490	0.0212422516700916	0.9926243567753000
17	0.0359634825376330	0.9879447221405460	0.0237387304498389	0.9923134648370490
18	0.0352461254204146	0.9882249705968830	0.0208181870685270	0.9933319039451110
19	0.0349013940192628	0.9881790282269920	0.0222625750056387	0.9928066037735840
20	0.0346183440962072	0.9884064429579530	0.0212522316110011	0.9927744425385930

Table 7.5.2-6: ResNet34 Model Training with Dataset 100% Quality Epoch Wise Results

Representation of above epoch wise training and validation loss results in the form of graph is given below: -

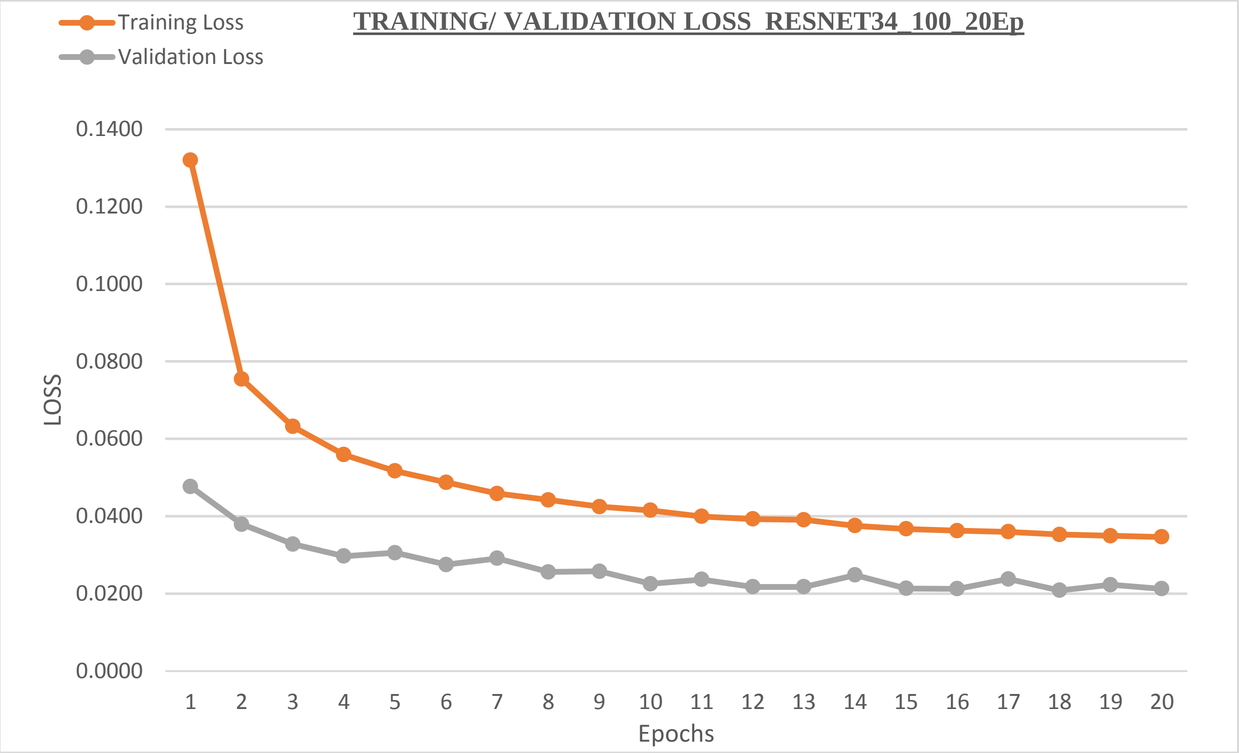


Figure *7.5.2.4*: ResNet34 Model Training and Validation Loss

Representation of above epoch wise training and validation Accuracies results in the form of graph is given below: -

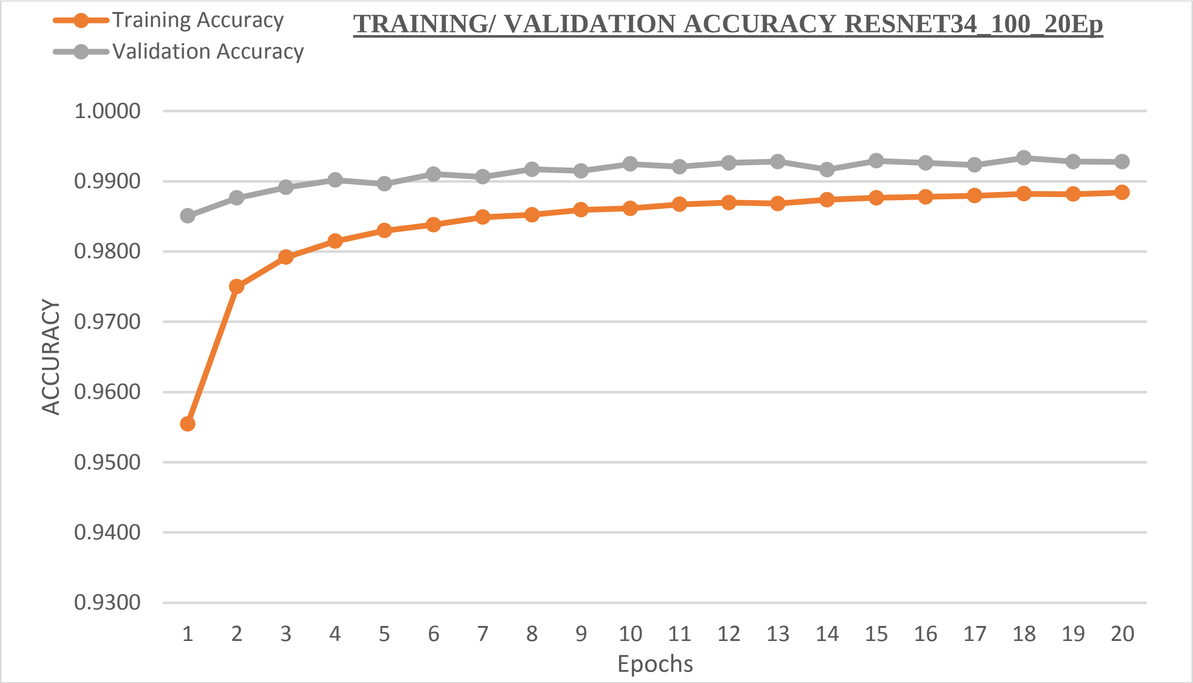
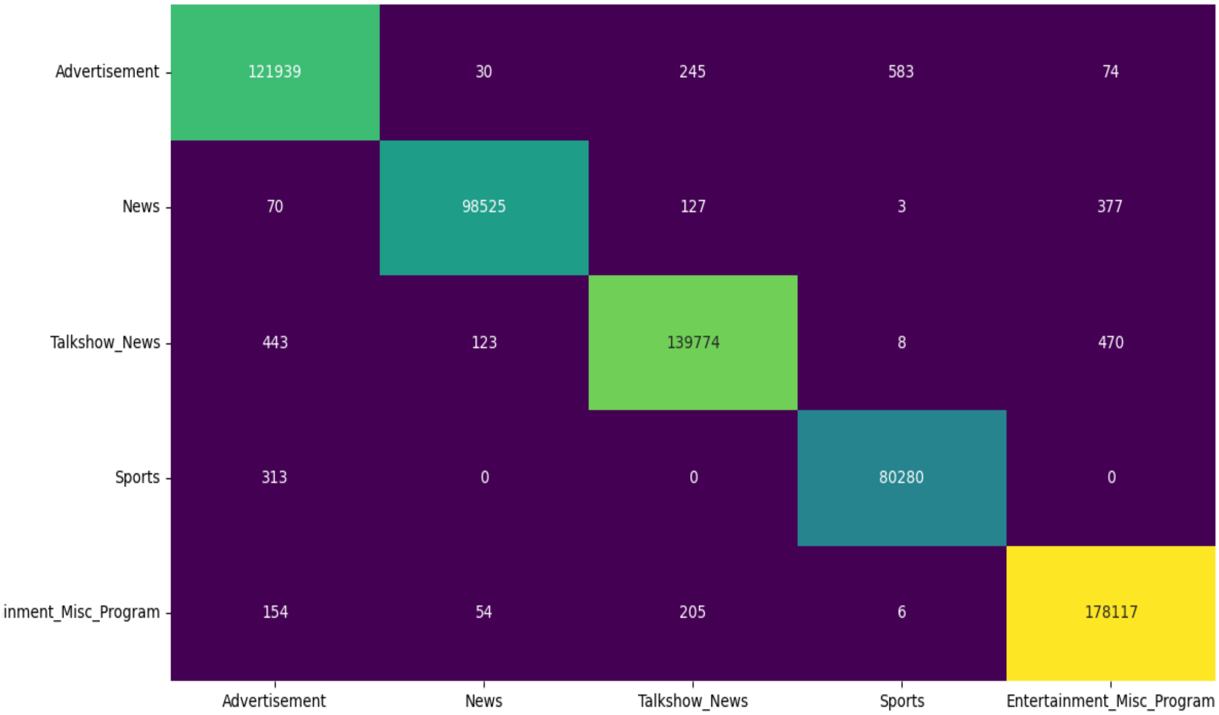


Figure 7.5.2.5: ResNet34 Model Training and Validation Accuracies of 20 Epochs on 2nd Dataset

Confusion matrix formed after training of **ResNet34 Model** on **dataset#2** for 20 epochs, represents classwise wrong and correct classification results as below: -



	Advertisement	News	Talkshow_News	Sports	Entertainment_Misc_Program
Advertisement	121939	30	245	583	74
News	70	98525	127	3	377
Talkshow_News	443	123	139774	8	470
Sports	313	0	0	80280	0
inment_Misc_Program	154	54	205	6	178117

Figure 7.5.2.6: ResNet34 Model Confusion Matrix

Four metrics scores calculated for analysis of trained **ResNet34 Model** on **dataset#2** for 20 epochs are given as under: -

Stream Classification Metrics Scores – ResNet34 Model							
Ser#	**Scores**	Advertisement	News	Talk Show	Sports	Entertainment	Average
1	Precision	0.9920272700	0.9979034153	0.9958888786	0.9925816024	0.9948558407	0.9946514014
2	Recall	0.9924148090	0.9941777159	0.9925861751	0.9961162880	0.9976531344	0.9945896245
3	F1 Score	0.9922210017	0.9960370816	0.9942347841	0.9943458039	0.9962525240	0.9946182390
4	Jaccard Score	0.9845620948	0.9921054487	0.9885356625	0.9887551883	0.9925330302	0.9892982849

Table 7.5.2-7: ResNet34 Model Stream Classification Metrics Scores

7.5.3 ResNet50 Model

ResNet50[186] model has been trained on both datasets#1 for 10 epochs respectively and results have been shown ahead.

Following are the hyper parameters used while training of the model: -

Model Training Fine Tuning Hyper Parameters - ResNet50 Model		
Serial #	**Parameters**	**Values**
1	Number of Epochs	10
2	Learning Rate	0.0001
3	Batch Size	16
4	Seed Value	42

Table 7.5.3-1: ResNet50 Model Fine Tuning Hyper Parameters

The overall **ResNet50 Model** stream classification results on complete **dataset#1** after **10 epochs** are as under: -

Stream Classification Overall Loss/ Accuracy – ResNet50 Model		
Serial #	**Properties**	**Values**
1	Training Loss	0.0725059776206874
2	Training Accuracy	0.9760031686260100
3	Validation Loss	0.0352849500083625
4	Validation Accuracy	0.9886874142661170
5	Testing Accuracy	0.9905156893004116

Table 7.5.3-2: ResNet50 Model Stream Classification Overall Training and Testing Results

These are the epoch wise training and validation loss and accuracies results of **ResNet50 Model** on **dataset#1** for **10 epochs**: -

Training/ Validation Loss/ Accuracy Results Epoch Wise – ResNet50 Model				
Epoch	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuracy**
1	0.161386537304394	0.946094324026451	0.056627624221960	0.983121141975308
2	0.089471626999124	0.970462894930198	0.042383174660942	0.986550497256515
3	0.081725540388279	0.973151634827332	0.035276622289100	0.988672410836762
4	0.068284267807255	0.977479794268920	0.034749166450110	0.988522376543209
5	0.062923523454952	0.979339639970609	0.030295424043899	0.990890775034293
6	0.057098395507355	0.981270664952241	0.031607828623487	0.989293981481481
7	0.054065544577382	0.982051340925789	0.028128476742953	0.990880058299039
8	0.051464285178385	0.982864162380602	0.029793176699021	0.989765517832647
9	0.050162896340834	0.983497887582659	0.034751485920272	0.988640260631001
10	0.048477158648914	0.983819342395297	0.029236520431881	0.990537122770919

Table 7.5.3-3: ResNet50 Model Training Epoch Wise Results

Representation of above epoch wise training and validation loss results in the form of graph is given below: -

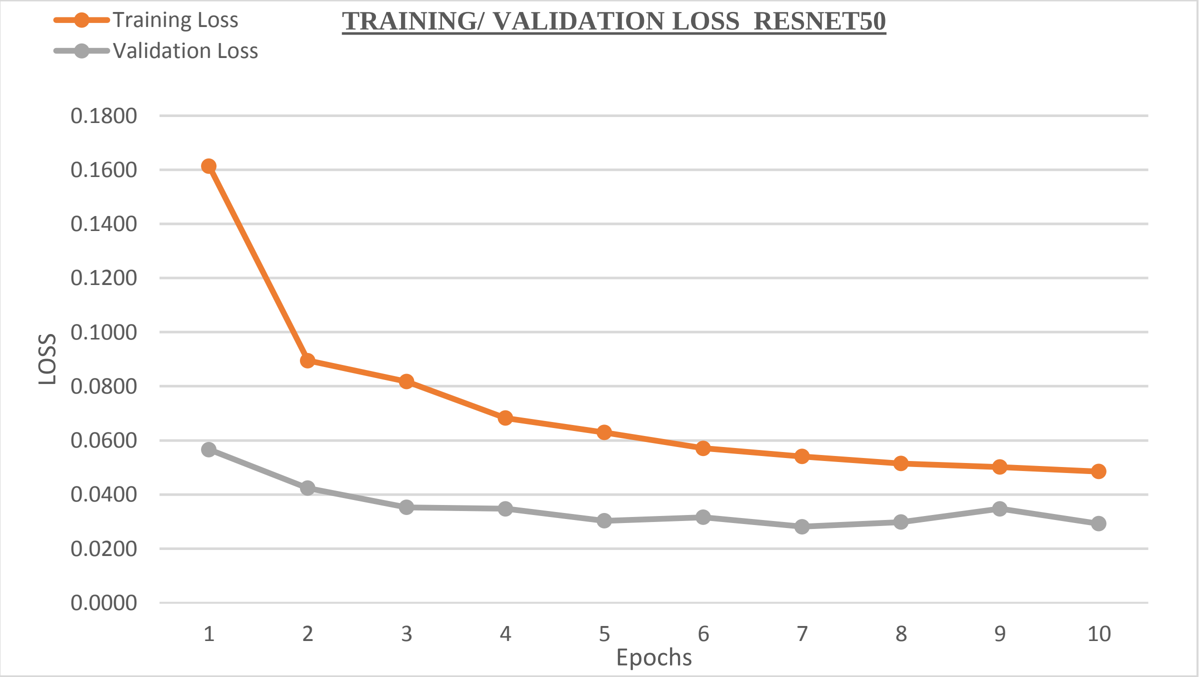


Figure 7.5.3.1: ResNet50 Model Training and Validation Loss

Representation of above epoch wise training and validation Accuracies results in the form of graph is given below: -

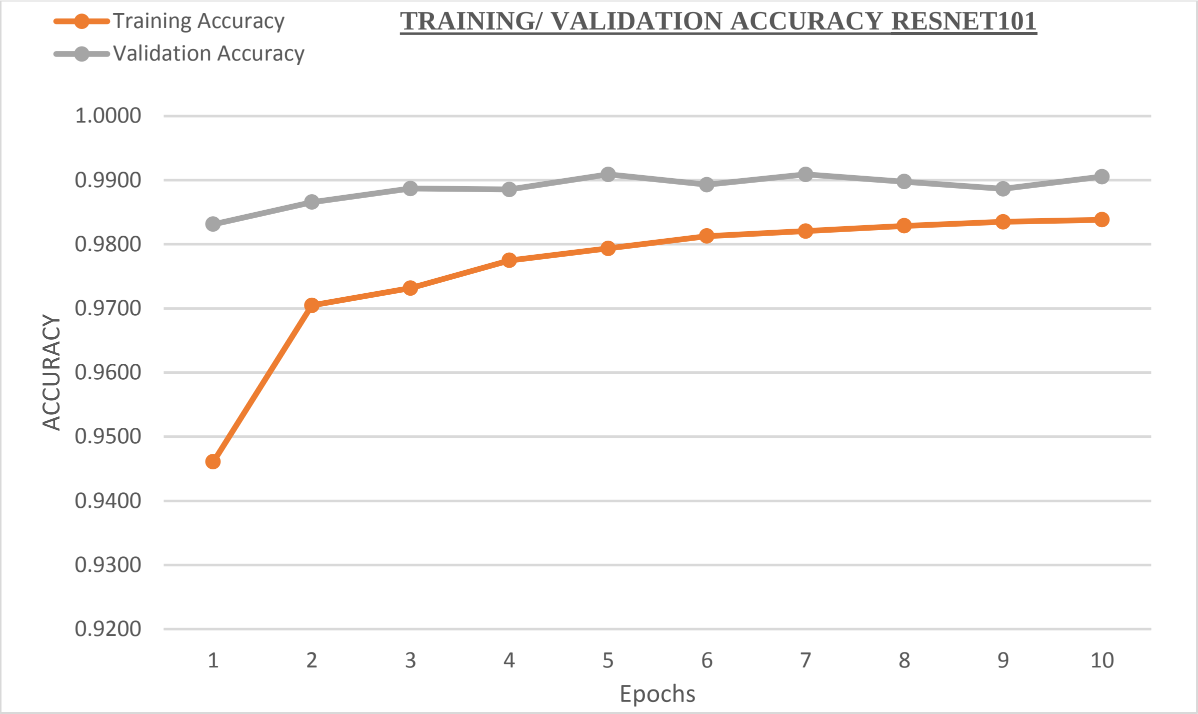
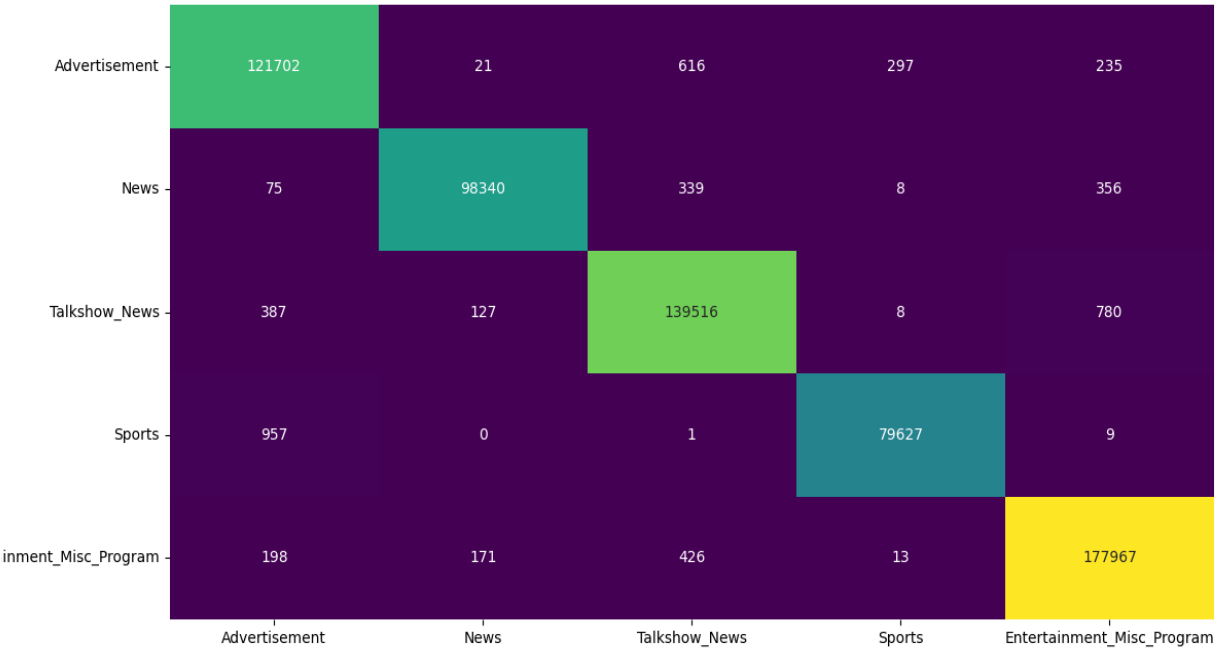


Figure 7.5.3.2: ResNet50 Model Training and Validation Accuracies

Confusion matrix formed after training of **ResNet50 Model** on **dataset#1** for 10 epochs, represents classwise wrong and correct classification results as below: -



	Advertisement	News	Talkshow_News	Sports	Entertainment_Misc_Program
Advertisement	121702	21	616	297	235
News	75	98340	339	8	356
Talkshow_News	387	127	139516	8	780
Sports	957	0	1	79627	9
inment_Misc_Program	198	171	426	13	177967

Figure 7.5.3.3: ResNet50 Model Confusion Matrix

Four metrics scores calculated for analysis of trained **ResNet50 Model** on **dataset#1** for 10 epochs are given as under: -

Stream Classification Metrics Scores – ResNet50 Model							
Ser#	**Scores**	Advertisement	News	Talk Show	Sports	Entertainment	Average
1	Precision	0.98688766532	0.99676664065	0.99019148604	0.99230541910	0.99241476313	0.99241476313
2	Recall	0.99048595682	0.99215076979	0.99075402292	0.98800158821	0.99548035240	0.99137453803
3	F1 Score	0.98868353711	0.99445334897	0.99047267461	0.99194628364	0.99389035022	0.99188923891
4	Jaccard Score	0.97762033288	0.98896788922	0.98112517581	0.98402125556	0.98785490272	0.98391791124

Table 7.5.3-4: ResNet50 Model Stream Classification Metrics Scores

7.5.4 ResNet101 Model

ResNet101[187] model has been trained on both dataset#1 for 10 epochs respectively and results have been shown ahead.

Following are the hyper parameters used while training of the model: -

Model Training Fine Tuning Hyper Parameters - ResNet101 Model		
Serial #	**Parameters**	**Values**
1	Number of Epochs	10
2	Learning Rate	0.0001
3	Batch Size	8
4	Seed Value	42

Table 7.5.4-1: ResNet101 Model Fine Tuning Hyper Parameters

The overall **ResNet101 Model** stream classification results on complete **dataset#1** after **10 epochs** are as under: -

Stream Classification Overall Loss/ Accuracy – ResNet101 Model		
Serial #	**Properties**	**Values**
1	Training Loss	0.0843395889217387
2	Training Accuracy	0.9719415870683320
3	Validation Loss	0.0444747009114265
4	Validation Accuracy	0.9863651950278610
5	Testing Accuracy	0.9873660522931847

Table 7.5.4-2: ResNet101 Model Stream Classification Overall Training and Testing Results

These are the epoch wise training and validation loss and accuracies results of **ResNet101 Model** on **dataset#1** for **10 epochs**: -

Training/ Validation Loss/ Accuracy Results Epoch Wise – ResNet101 Model				
Epoch	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuracy**
1	0.1974649361102970	0.9339157788390880	0.0665144474817934	0.9804329189884260
2	0.1082051726832440	0.9640613519470970	0.0541550983325404	0.9835726532361760
3	0.0862371783520074	0.9714180749448930	0.0468801765706502	0.9857908272610370
4	0.0742755563130884	0.9755051432770020	0.0454593164983728	0.9859301328761250
5	0.0831178544748303	0.9725707200587800	0.0410498508523659	0.9870124303471920
6	0.0697059485800401	0.9768139235855980	0.0377008000640448	0.9879768538362620
7	0.0612472577384365	0.9795279206465830	0.0384699772953495	0.9879982854693520
8	0.0575210834788998	0.9807792983100660	0.0364516009014320	0.9887162451778820
9	0.0540377202915464	0.9820054188096980	0.0378319297536463	0.9885662237462490
10	0.0515831811949966	0.9828182402645110	0.0402338113640699	0.9876553793399050

Table 7.5.4-3: ResNet101 Model Training Epoch Wise Results

Representation of above epoch wise training and validation loss results in the form of graph is given below: -

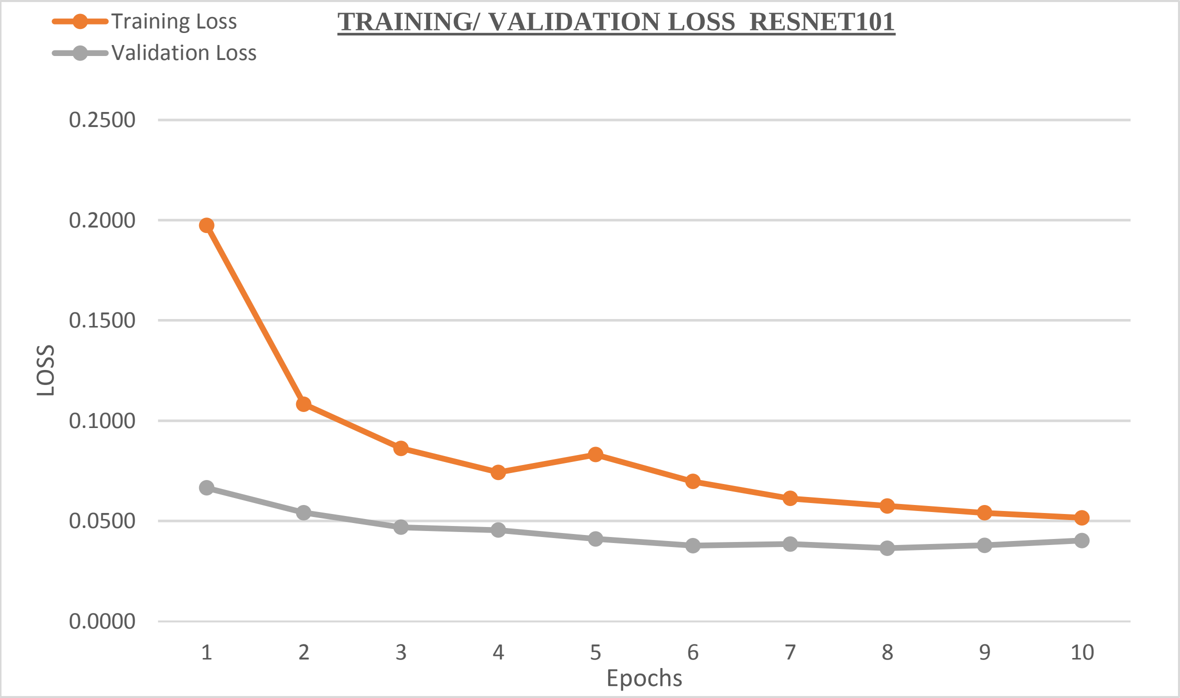


Figure 7.5.4.1: ResNet101 Model Training and Validation Loss

Representation of above epoch wise training and validation Accuracies results in the form of graph is given below: -

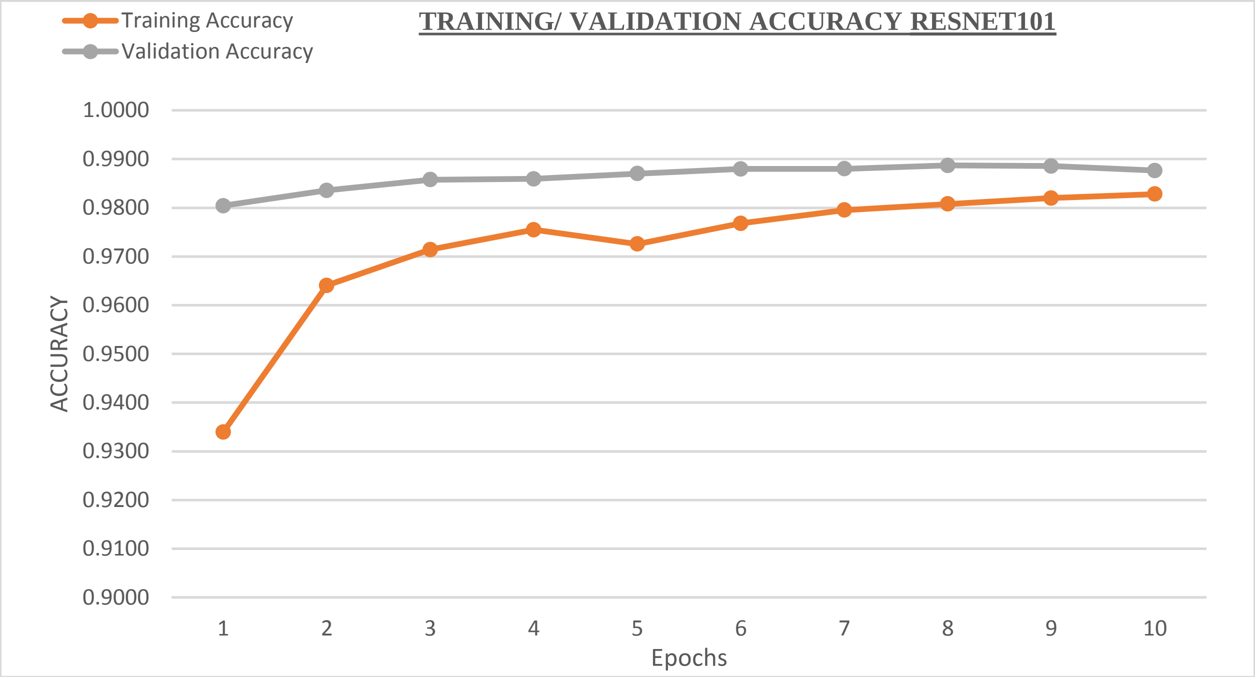
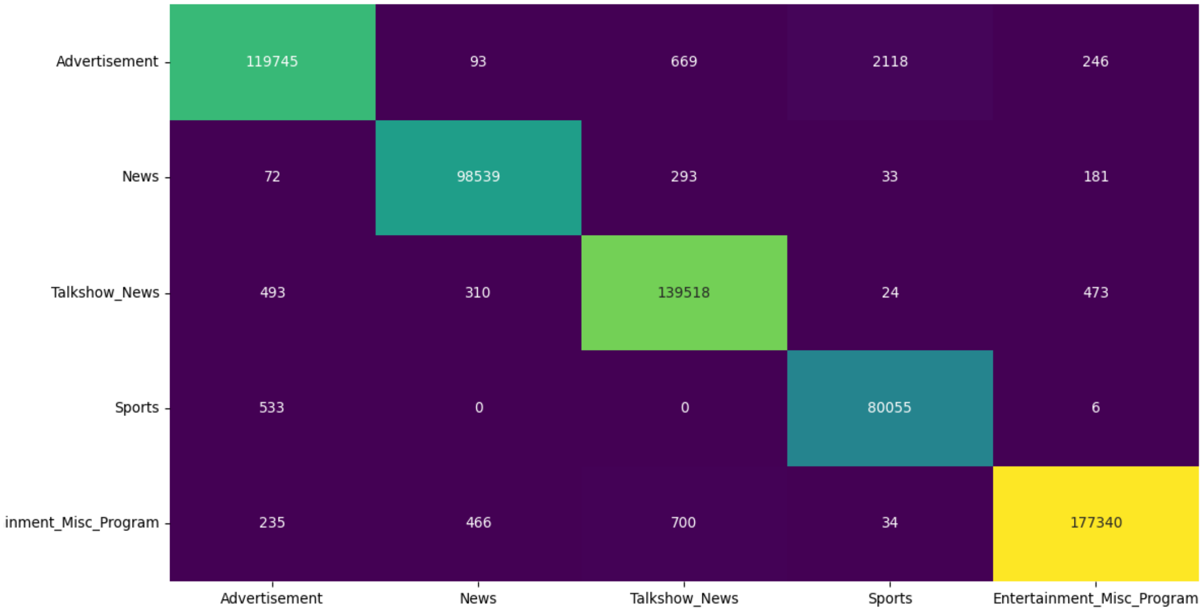


Figure 7.5.4.2: ResNet101 Model Training and Validation Accuracies

Confusion matrix formed after training of **ResNet101 Model** on **dataset#1** for 10 epochs, represents classwise wrong and correct classification results as below: -



	Advertisement	News	Talkshow_News	Sports	Entertainment_Misc_Program
Advertisement	119745	93	669	2118	246
News	72	98539	293	33	181
Talkshow_News	493	310	139518	24	473
Sports	533	0	0	80055	6
inment_Misc_Program	235	466	700	34	177340

Figure 7.5.4.3: ResNet101 Model Confusion Matrix

Four metrics scores calculated for analysis of trained **ResNet101 Model** on **dataset#1** for 10 epochs are given as under: -

Stream Classification Metrics Scores – ResNet101 Model							
Ser#	**Scores**	Advertisement	News	Talk Show	Sports	Entertainment	Average
1	Precision	0.98899056806	0.99125824883	0.98822779431	0.97314742779	0.99491713699	0.98730823520
2	Recall	0.97455868350	0.99415847777	0.99076822565	0.99331215723	0.99197315061	0.98895413895
3	F1 Score	0.98172158935	0.99270624503	0.98949637941	0.98312640460	0.99344296274	0.98809871622
4	Jaccard Score	0.96409938488	0.98551811736	0.97921111735	0.96681279664	0.98697135479	0.97652255420

Table 7.5.4-4: ResNet101 Model Stream Classification Metrics Scores

7.5.5 ResNet152 Model

ResNet152[188] model has been trained on both dataset#1 for 10 epochs respectively and results have been shown ahead.

Following are the hyper parameters used while training of the model: -

Model Training Fine Tuning Hyper Parameters - ResNet152 Model		
Serial #	**Parameters**	**Values**
1	Number of Epochs	10
2	Learning Rate	0.0001
3	Batch Size	6
4	Seed Value	42

Table 7.5.5-1: ResNet152 Model Fine Tuning Hyper Parameters

The overall **ResNet152 Model** stream classification results on complete **dataset#1** after **10 epochs** are as under: -

Stream Classification Overall Loss/ Accuracy – ResNet152 Model		
Serial #	**Properties**	**Values**
1	Training Loss	0.1008175415663910
2	Training Accuracy	0.9664416561537550
3	Validation Loss	0.0493528078198465
4	Validation Accuracy	0.9843898710450210
5	Testing Accuracy	0.9875487577061397

Table 7.5.5-2: ResNet152 Model Stream Classification Overall Training and Testing Results

These are the epoch wise training and validation loss and accuracies results of **ResNet152 Model** on **dataset#1** for **10 epochs**: -

Training/ Validation Loss/ Accuracy Results Epoch Wise – ResNet152 Model				
Epoch	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuracy**
1	0.2077304539547020	0.9297532734916870	0.0800812714480472	0.9744224477177550
2	0.1193165359372570	0.9601627553439010	0.0571338000886210	0.9816767431491900
3	0.1014527637939170	0.9665321224890700	0.0530078029980880	0.9841305595161950
4	0.0928514008645684	0.9690945643156100	0.0451190700983240	0.9858343009204070
5	0.0883680115078945	0.9706559088066490	0.0451147608778642	0.9855342709659470
6	0.0841286750790385	0.9719026883248850	0.0439997025475479	0.9860271773125860
7	0.0816632004691362	0.9730736966884430	0.0427740103248003	0.9862736304207600
8	0.0787543857709296	0.9741161237386310	0.0416579034553145	0.9867986828583100
9	0.0779015507310900	0.9742470011458930	0.0425340640474792	0.9866379525493770
10	0.0760084375553806	0.9748784271927800	0.0421056923123789	0.9865629450396850

Table 7.5.5-: ResNet152 Model Training Epoch Wise Results

Representation of above epoch wise training and validation loss results in the form of graph is given below: -

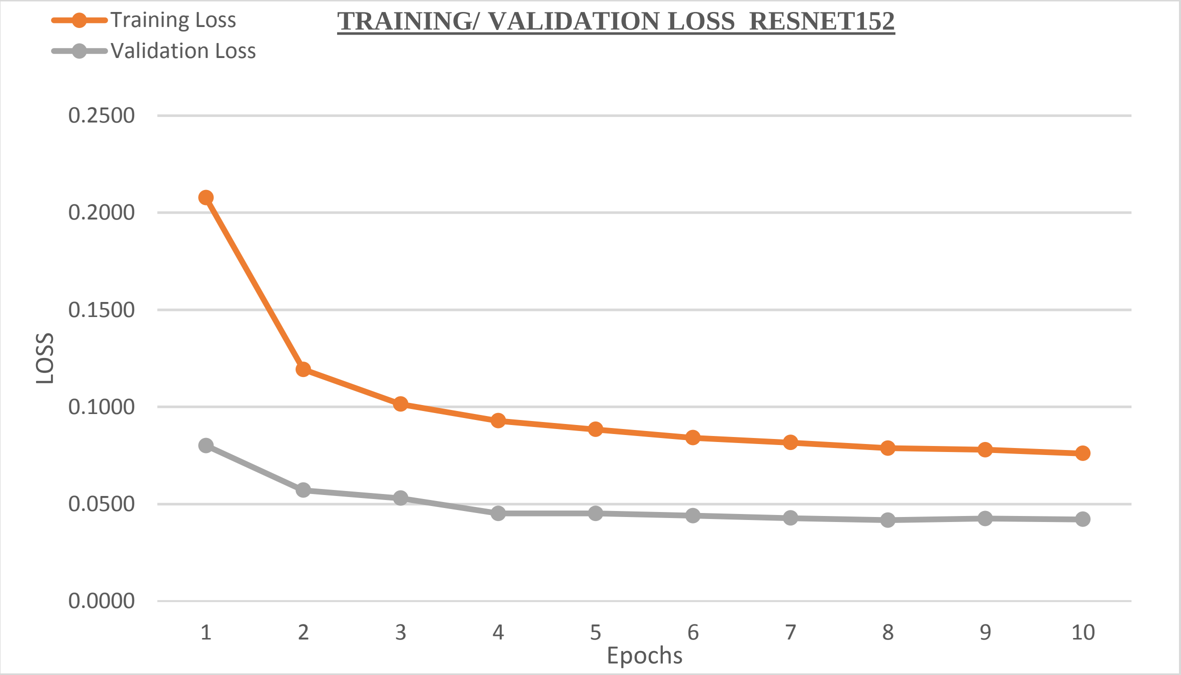


Figure 7.5.5.1: ResNet152 Model Training and Validation Loss

Representation of above epoch wise training and validation Accuracies results in the form of graph is given below: -

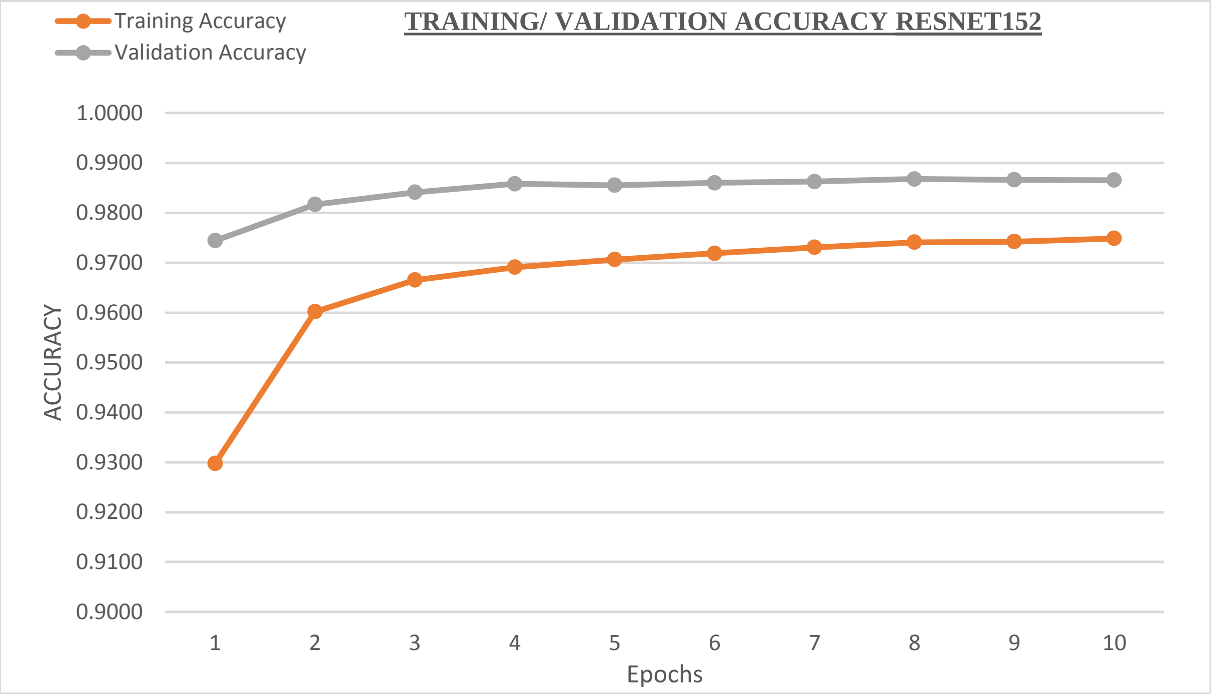


Figure 7.5.5.2: ResNet152 Model Training and Validation Accuracies

Confusion matrix formed after training of **ResNet152 Model** on **dataset#1** for 10 epochs, represents classwise wrong and correct classification results as below: -

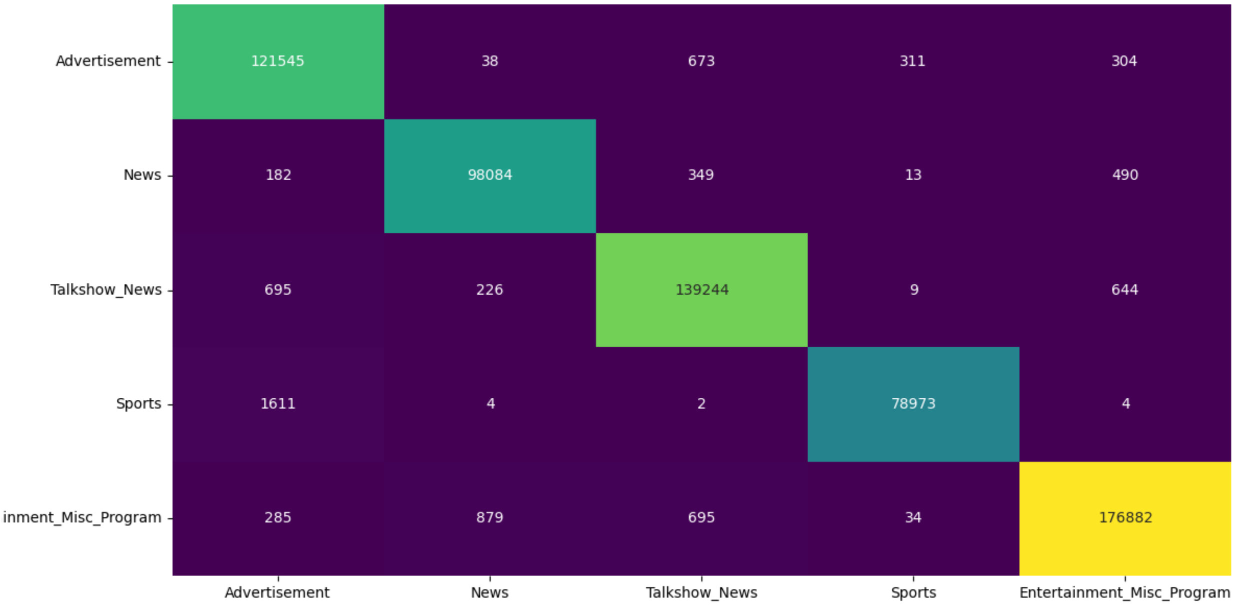


Figure 7.5.5.3: ResNet152 Model Confusion Matrix

Four metrics scores calculated for analysis of trained **ResNet152 Model** on **dataset#1** for 10 epochs are given as under: -

Stream Classification Metrics Scores – ResNet152 Model							
Ser#	**Scores**	Advertisement	News	Talk Show	Sports	Entertainment	Average
1	Precision	0.977694300	0.988441112	0.987805311	0.995374338	0.991913595	0.988245731
2	Recall	0.989208194	0.989567990	0.988822452	0.979886840	0.989411271	0.987379349
3	F1 Score	0.983417547	0.989004230	0.988313619	0.987569873	0.990660853	0.987793224
4	Jaccard Score	0.967376078	0.978247644	0.976897227	0.975444967	0.981494532	0.975892090

Table 7.5.5-4: ResNet152 Model Stream Classification Metrics Scores

7.6 ConvNext_Tiny Model

ConvNext[189]–[191] is a way to modernize the standard ResNet towards of a vision transformers, hence contributing to the performance difference and are constructed purely on convnets. It has variants, but we have only trained ConvNext_Tiny Model on dataset#1 for 10 epochs. The results have been shown in ensuing paragraphs.

Following are the hyper parameters used while training of the model: -

Model Training Fine Tuning Hyper Parameters - ConvNext_Tiny Model		
Serial #	**Parameters**	**Values**
1	Number of Epochs	10
2	Learning Rate	0.0001
3	Batch Size	8
4	Seed Value	42

Table 7.6-1: ConvNext_Tiny Model Fine Tuning Hyper Parameters

The overall **ConvNext_Tiny Model** stream classification results on complete **dataset#1** after 10 epochs are as under: -

Stream Classification Overall Loss/ Accuracy – ConvNext_Tiny Model		
Serial #	**Properties**	**Values**
1	Training Loss	0.0665330734658016
2	Training Accuracy	0.9772435249816310
3	Validation Loss	0.0475605164152739
4	Validation Accuracy	0.9840430775825110
5	Testing Accuracy	0.9833797685383620

Table 7.6-2: ConvNext_Tiny Model Stream Classification Overall Training and Testing Results

These are the epoch wise training and validation loss and accuracies results of **ConvNext_Tiny Model** on **dataset#1** for **10 epochs**: -

Training/ Validation Loss/ Accuracy Results Epoch Wise – ConvNext_Tiny Model				
Epoch	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuracy**
1	0.1274352937215750	0.9575518919911830	0.0560971817252931	0.9823939134162020
2	0.0886070553609191	0.9700679647318140	0.0446055300617828	0.9857265323617660
3	0.0745660835665133	0.9742882072005870	0.0521701250414893	0.9820081440205740
4	0.0660938831716752	0.9772524797942680	0.0453496684194670	0.9848906986712380
5	0.0598949391581120	0.9794682218956650	0.0498579035682650	0.9823296185169310
6	0.0559786583839866	0.9806621969140330	0.0383572396525715	0.9877196742391770
7	0.0520877508163307	0.9818699485672300	0.0546314202499069	0.9807222460351470
8	0.0490262838815354	0.9830455547391620	0.0464721258148624	0.9840870124303470
9	0.0468927974136242	0.9837711241734020	0.0391066471269556	0.9870338619802820
10	0.0447479891837451	0.9844576598089640	0.0489573224921452	0.9835190741534500

Table 7.6-3: ConvNext_Tiny Model Training Epoch Wise Results

Representation of above epoch wise training and validation loss results in the form of graph is given below: -

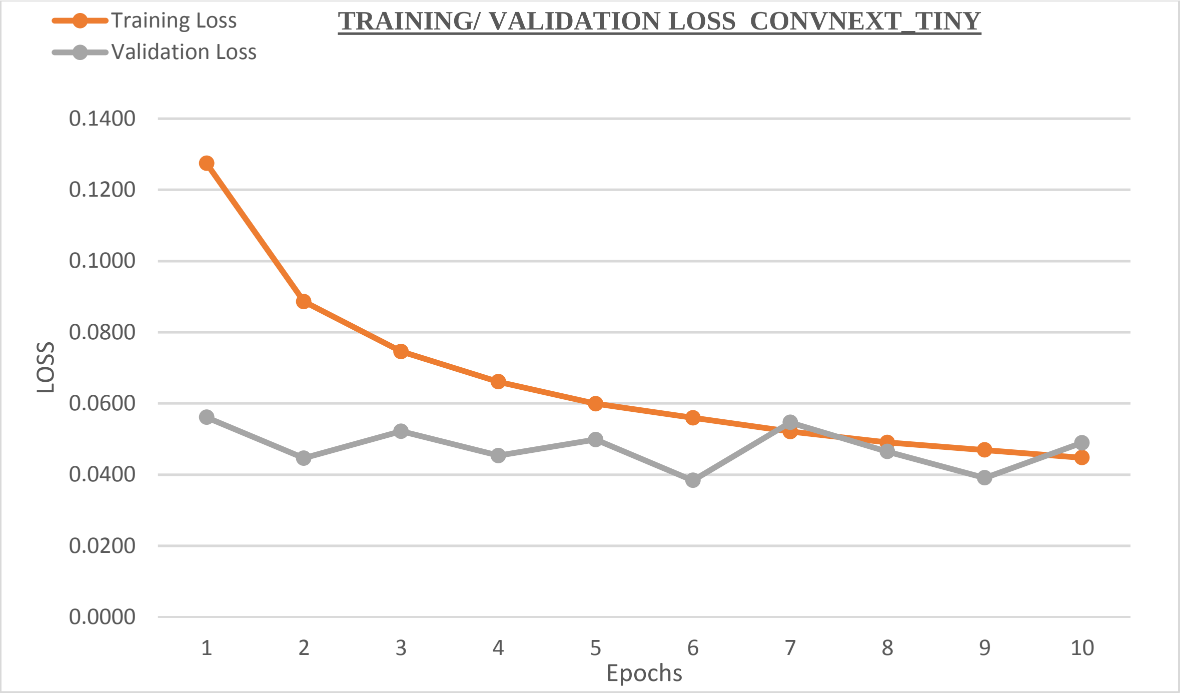


Figure 7.6.1: ConvNext_Tiny Training and Validation Loss

Representation of above epoch wise training and validation Accuracies results in the form of graph is given below: -

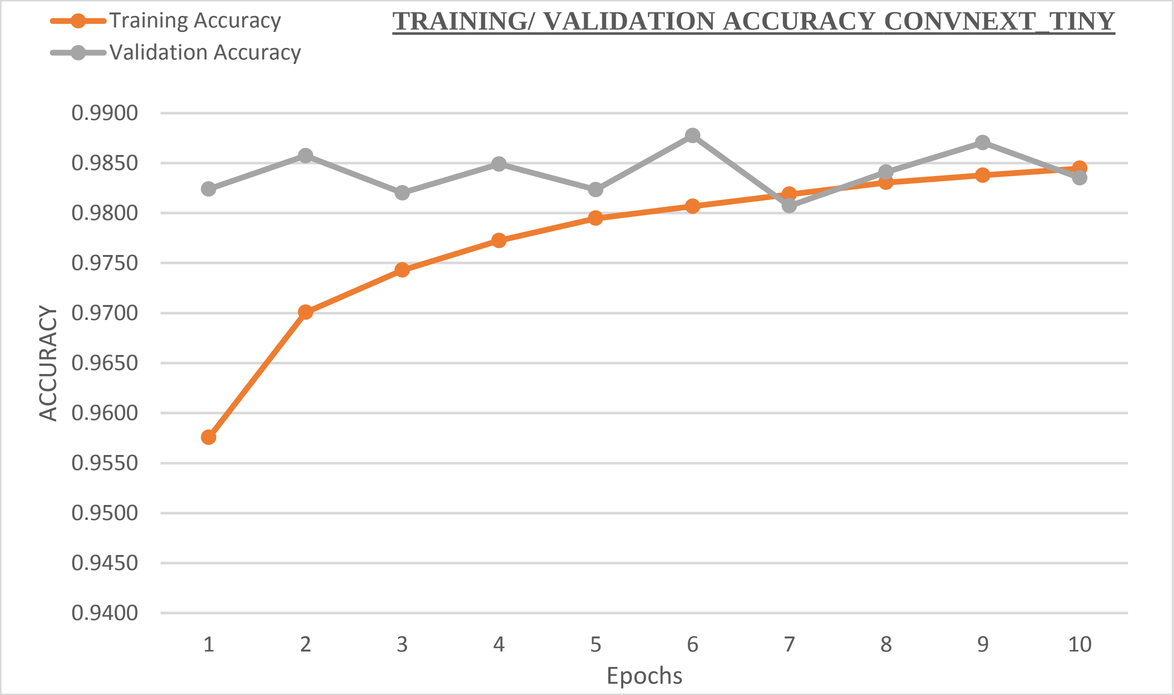


Figure 7.6.2: ConvNext_Tiny Model Training and Validation Accuracies

Confusion matrix formed after training of **ConvNext_Tiny Model** on **dataset#1** for 10 epochs, represents classwise wrong and correct classification results as below: -

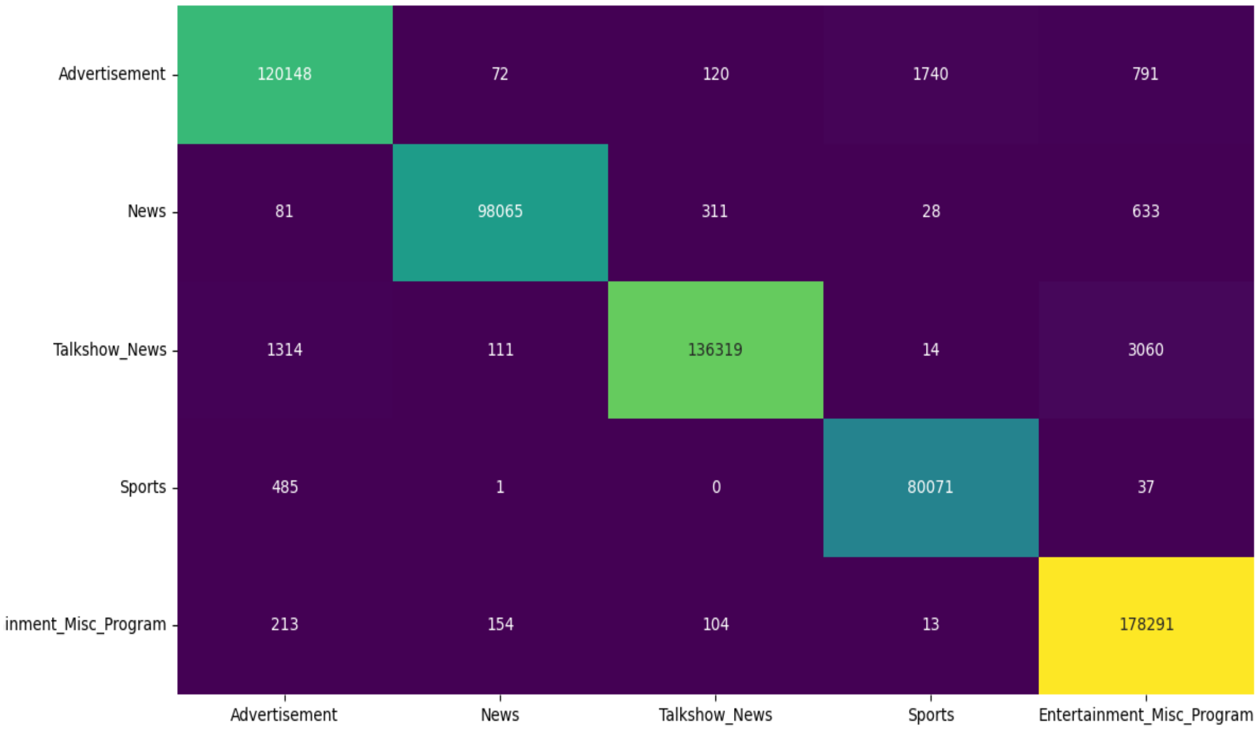


Figure 7.6.3: ConvNext_Tiny Model Confusion Matrix

Four metrics scores calculated for analysis of trained **ConvNext_Tiny Model** on **dataset#1** for 10 epochs are given as under: -

Stream Classification Metrics Scores – ConvNext_Tiny Model							
Ser#	**Scores**	**Advertisement**	**News**	**Talk Show**	**Sports**	**Entertainment**	**Average**
1	Precision	0.9828780851	0.9965651454	0.9960907244	0.9780739257	0.9752696760	0.9857755113
2	Recall	0.9778385461	0.9893762990	0.9680509594	0.9935106832	0.9972926863	0.9852138348
3	F1 Score	0.9803518392	0.9929577108	0.9818706964	0.9857318725	0.9861582413	0.9854140720
4	Jaccard Score	0.9614609007	0.9860139157	0.9643870310	0.9718651762	0.9726944396	0.9712842927

Table 7.6-4: ConvNext_Tiny Model Stream Classification Metrics Scores

7.7 Comparison of Overall Stream Classification Results of All Deep Neural Networks

Training and validation results (loss & accuracies) for all ten models finetuned for ten epochs on Dataset#1 are given as under: -

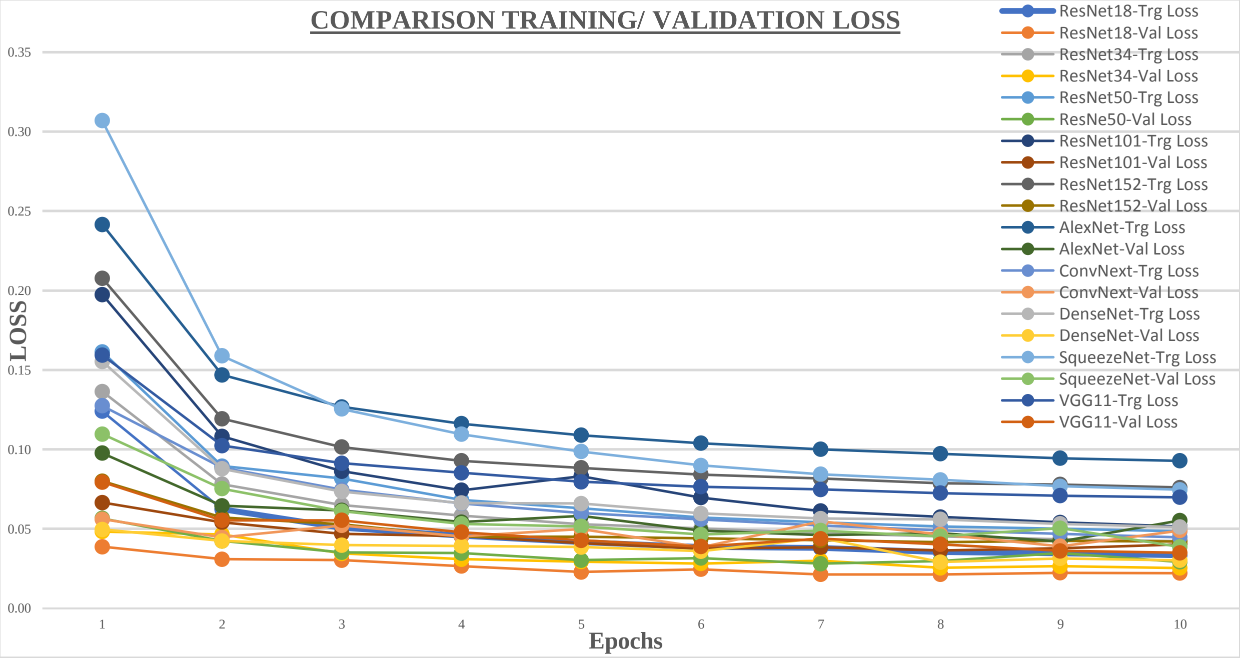


Figure 7.7.1: Comparison of Training and Validation Losses

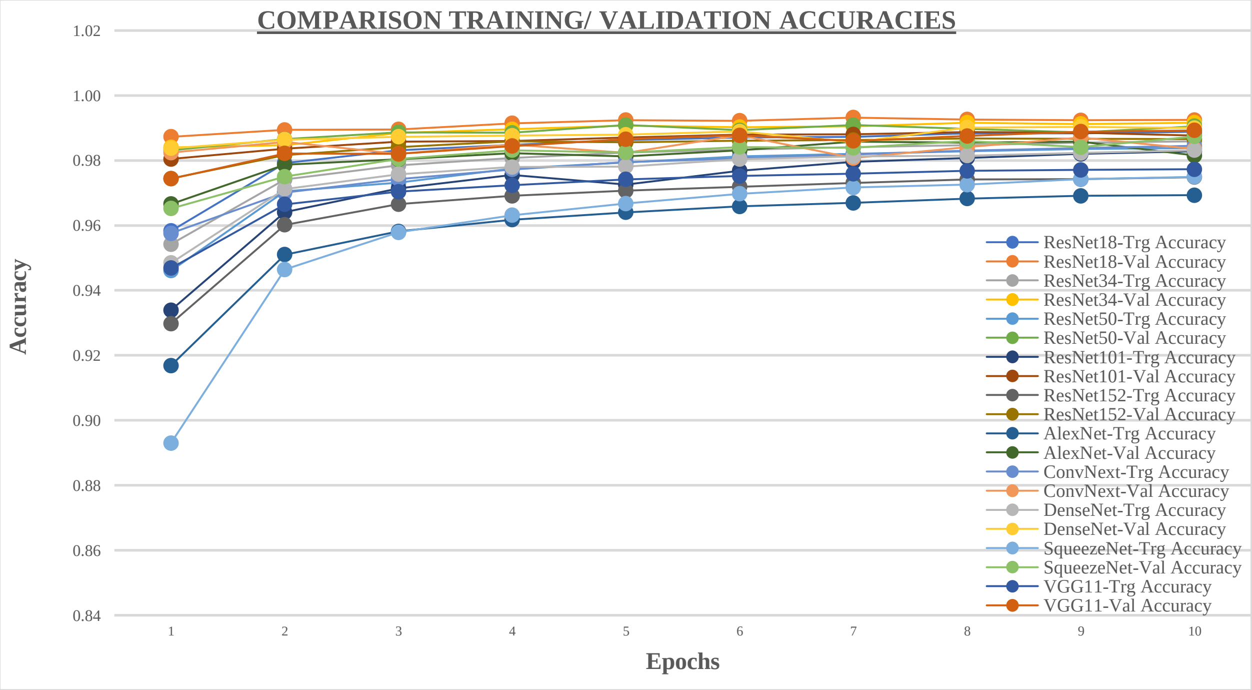


Figure 7.7.2: Comparison of Training and Validation Accuracies

In both figures one can see the performance of all models, in case of losses convergence is descending whereas ascending in case of accuracies. We achieved quite encouraging results which improved as the training goes on.

Overall training results of ResNet variants and other models are given below: -

Ser #	Properties	ResNet18	ResNet34	ResNet50	ResNet101	ResNet152
1	Trg Loss	0.050426	0.061414	0.072506	0.084340	0.100818
2	Trg Accuracy	0.983160	0.979498	0.976003	0.971942	0.966442
3	Val Loss	0.026123	0.032253	0.035285	0.044475	0.049353
4	Val Accuracy	0.991286	0.989346	0.988687	0.986365	0.984390
5	Test Accuracy	0.992616	0.989991	0.990516	0.987366	0.987549
6	Precision	0.9945154	0.9915549	0.9871190	0.9873082	0.9882457
7	Recall	0.9947016	0.9919023	0.9881935	0.9889541	0.9873793
8	F1 Score	0.9946079	0.9917207	0.9876484	0.9880987	0.9877932
9	Jaccard Score	0.9892771	0.9835866	0.9756358	0.9765226	0.9758921

Table 7.7-1: Overall Training Results of ResNet Variants

Ser #	Properties	AlexNet	ConvNext	DenseNet121	SqueezeNet	VGG11
1	Trg Loss	0.122877	0.066533	0.072524	0.120605	0.088274
2	Trg Accuracy	0.959126	0.977244	0.976013	0.959030	0.971254
3	Val Loss	0.057578	0.047561	0.038044	0.058056	0.047495
4	Val Accuracy	0.981056	0.984043	0.987793	0.981175	0.984889
5	Test Accuracy	0.982092	0.983380	0.990388	0.988062	0.989038
6	Precision	0.9846047	0.9857755	0.9909088	0.9896846	0.9924148
7	Recall	0.9818826	0.9852138	0.9908918	0.9881490	0.9913745
8	F1 Score	0.9830337	0.9854141	0.9908957	0.9888945	0.9918892
9	Jaccard Score	0.9667481	0.9712843	0.9819711	0.9780501	0.9839179

Table 7.7-2: Overall Training Results of 5 x Models

These results show the overall performance of each model in terms of different metrics which we have been set along with the training/ validation loss/ accuracies. Generally speaking, the performance of ResNet variants remained better than other five models.

In next two figures, we have given comparison of overall accuracies and losses achieved after training of each model on Dataset#1 for 10 epochs and ResNet18&34 on Dataset#2 for 20 epochs. Comparison clearly shows the overall best loss and accuracies against ResNet18 & ResNet34 and other models with next best results are ResNet50 and ConvNex.

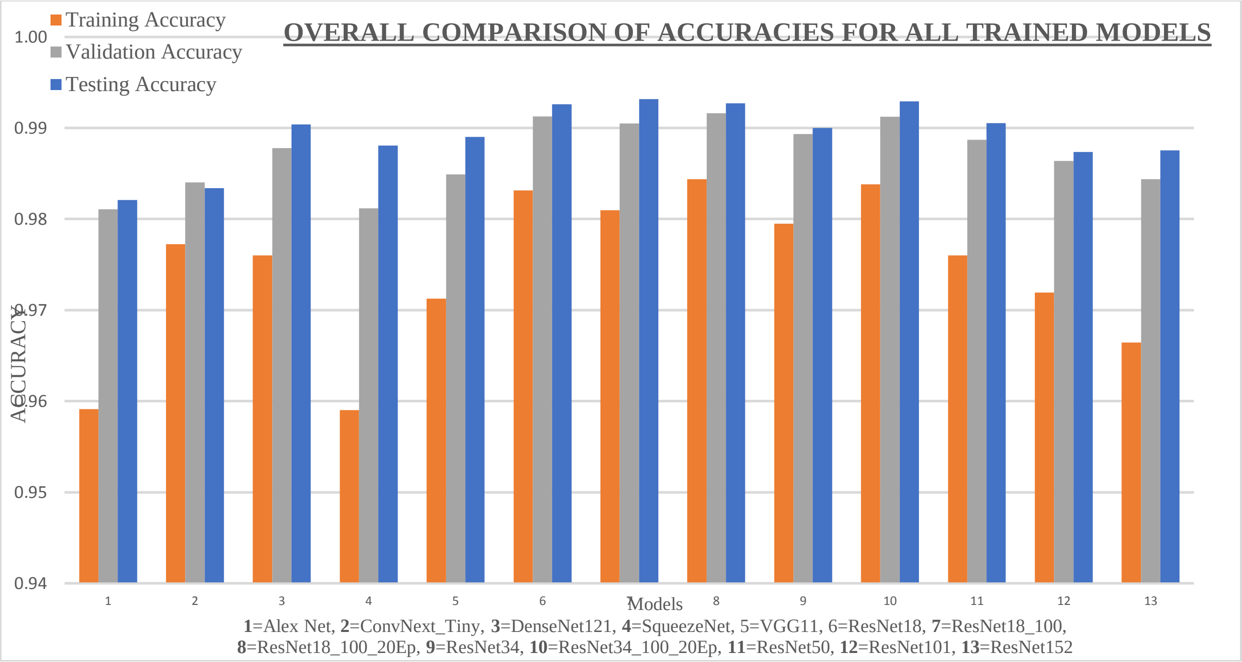


Figure 7.7.3: Comparison of Overall Stream Classification Accuracies

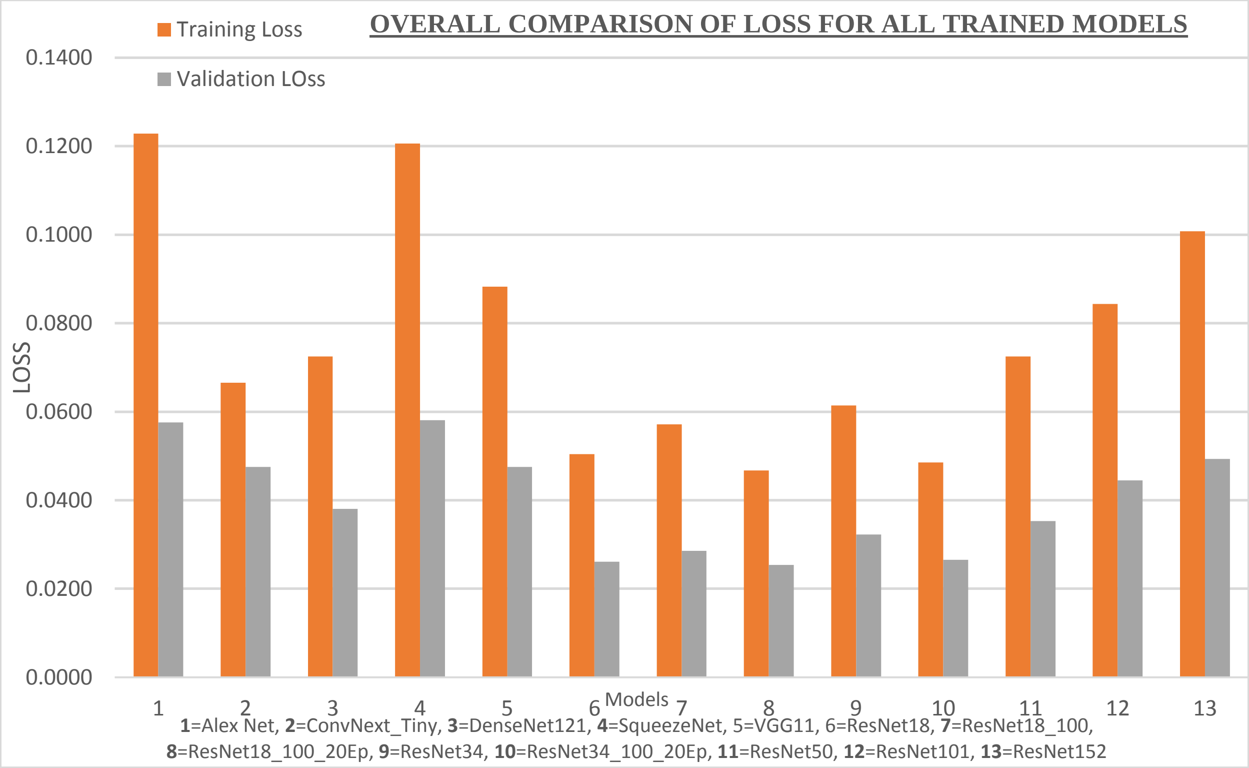


Figure 7.7.4: Comparison of Overall Stream Classification Loss

Graphical comparison of overall average values of metrics against each model experimented is given in Fig. 25 below: -

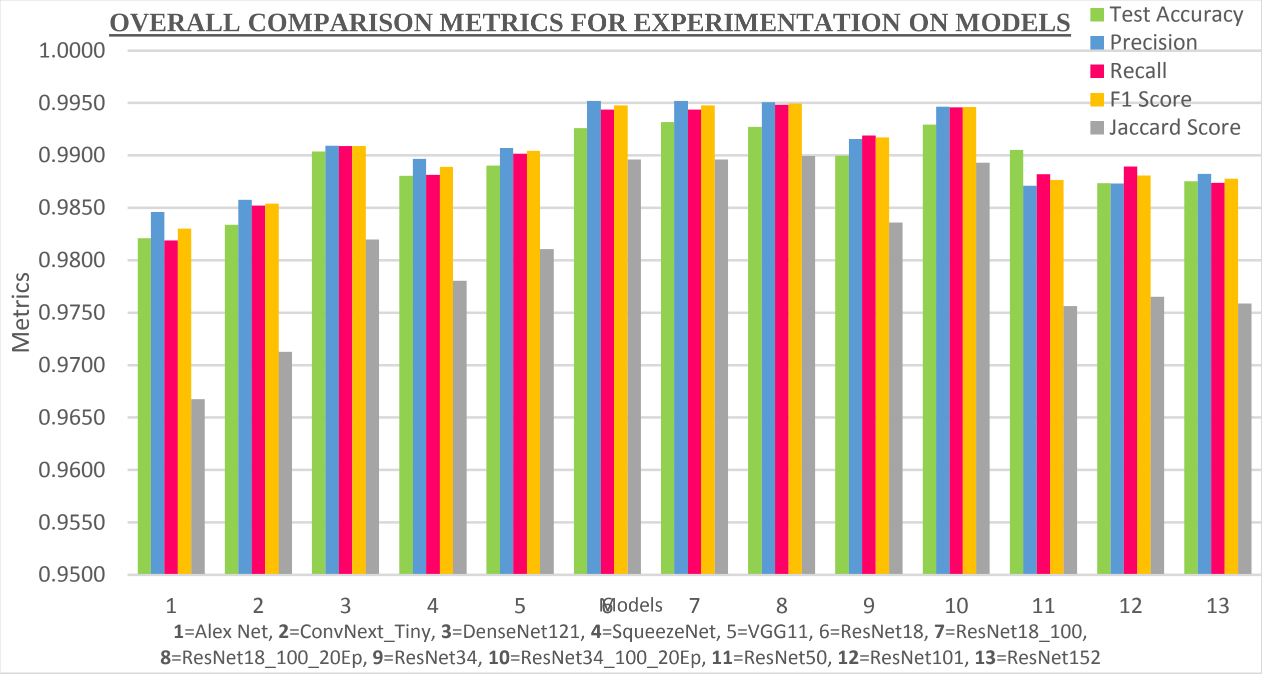


Figure 7.7.5: Comparison of Overall Stream Classification Metrics

We have drawn comparison of two models ResNet18&34 trained on both datasets#1&2. Which concluded that if the quality of dataset is reasonably enough for features extraction then there is not much difference on the results as shown in next two figures.

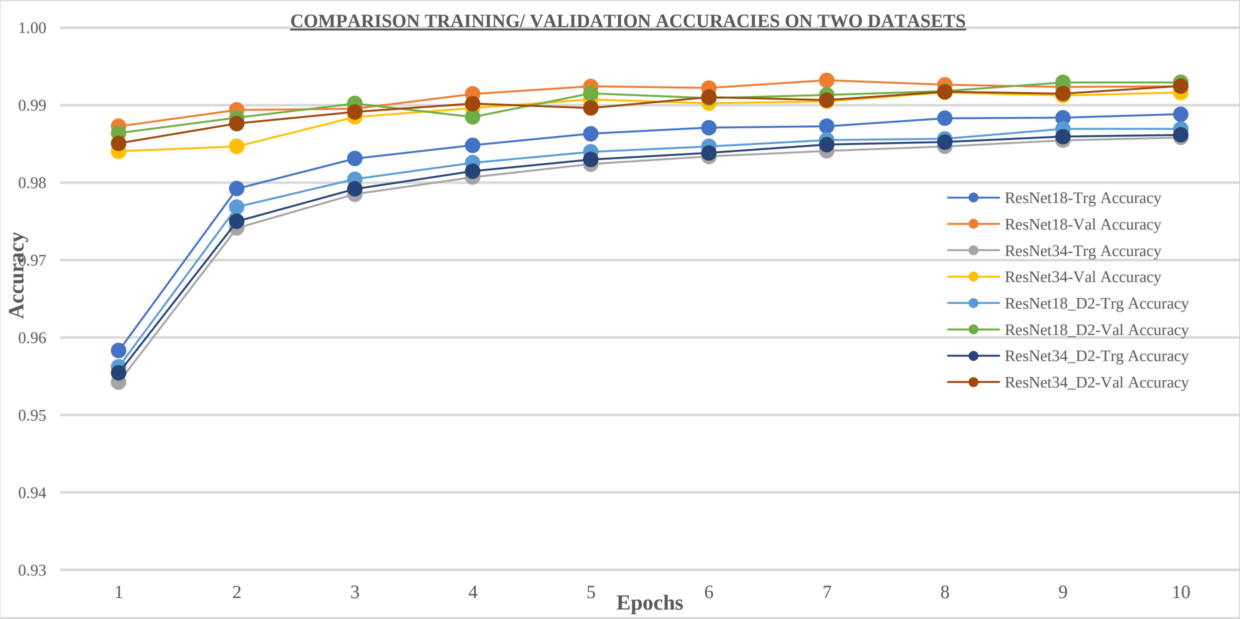


Figure 7.7.6: Comparison of Stream Classification Training/ Validation Accuracies of ResNet18&34 trained on Both Datasets#1&2

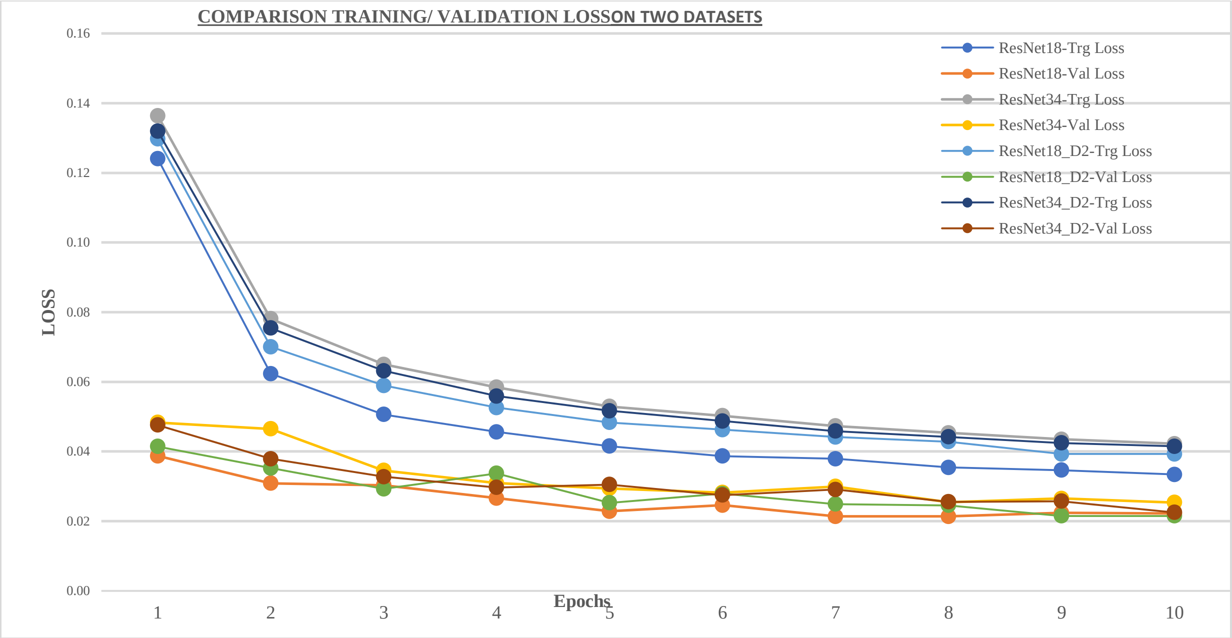


Figure 7.7.7: Comparison of Stream Classification Training/ Validation Loss of ResNet18&34 trained on Both Datasets#1&2

Models were tested on Pakistani Channels recorded videos and achieved quite reasonable results. Some of the image frames duly classified have been shown below as samples: -

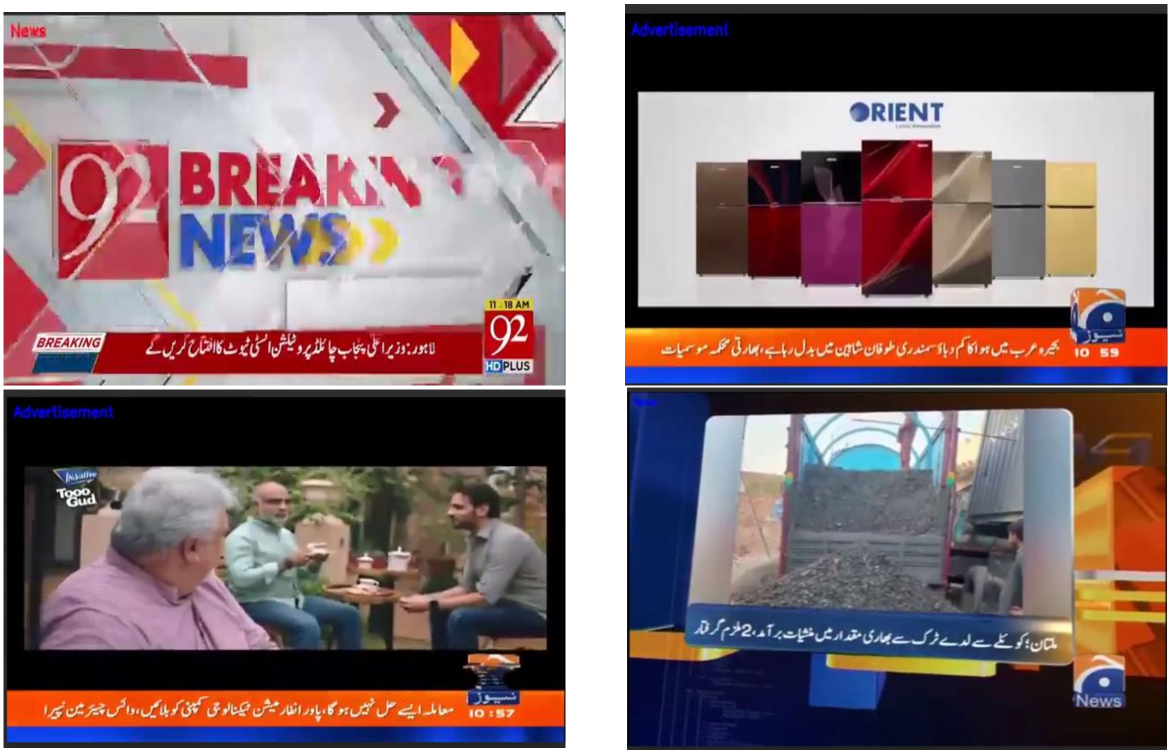


Figure 7.7.8: Test Results Samples duly Classified by Trained Models

CHAPTER 8: LIMITATIONS

In this study, it has been suggested to undertake the stream classification problem using image classification established deep learning (DL) techniques and then applying dynamic probability averaging time window for classifying the video understudy. We have discussed earlier that video stream not only consists of images, but it also contains other information like voice riding over the video, written scripts in the form of words and sentences in the image frames and the metadata. All this information may also contribute towards better decision making for accurate classification of live video stream. But each information will be required to be taken as a separate category for detailed study. However, in this study we are focusing only on classification of video stream using image classification DL techniques. As we know that the classification techniques using DL Neural Networks are highly dependent on the quality of dataset duly incorporated with the features of generality on which model are trained. However, no authentic dataset especially designed for stream classification of live video feed from Pakistani News Channels was available. So, we had to spend lot of time for preparation of a generalized and trainable dataset. Moreover, the content on Pakistani TV Channels keeps on varying continuously with the passage of time leading towards the huge drift necessitating update of the dataset with passage of time. We observed that programs presentation styles, patterns, anchors, hosts etc. keep on changing which may result into confused situation during classification. So, the dataset prepared at one time for subject purposes may lose its utilization over the time unless a separate mechanism is devised for continuous updating of the dataset and the already trained/ finetuned neural network.

Presentation styles of different channels are sometimes extremely diverse in nature making it difficult to correctly identify its class. Even sometimes the advertisements are displayed on any side, corner, up or down portion of the television screen, making difficult for the model to predict correctly and placing the frame into relevant program or advertisement class. Also, sometimes quite a number of frames of different classes seem very similar in nature resulting in false positives etc. However, effort has been made to prepare a big and generalized dataset to be able to remain valid for quite some years. It has also been taken care of shedding away similar frames by taking frames after a specific duration, which also reduced the quantity of trainable data. Thus, making it reasonable and matching our resources. But there are still some chances that it may reduce accuracy on different or a new channel data. So, we can say that the results may be compromised sometimes in certain cases.

Therefore, we need to remain within set boundaries to attain better and consistency in results. We may also need to focus more on preparation of huge, accurate and generalized dataset for further improvement in the stream classification problem.

While performing stream classification, we are confronted with the continuous video which contains many frames with extremely poor features and sometimes having no features at all resulting in drastic mistakes in classification. Therefore, we tried to handle this one of the problems by applying dynamic averaging time domain window which not only improved the results but also diminished the jitters and wrong predictions. But with this, one cannot claim the hundred percent accuracy in results. So, this aspect also needs to be improved further in future studies.

DL neural networks require huge computational resources during training of the models and to handle big datasets of videos and image frames. So, it is very important that sufficient and reliable computational resources are arranged before taking on such problems to avoid unnecessary delays. We also had to face this problem of non-availability of reliable and suitable computational resources in time. Due to this problem, study could not be completed in earlier timeframe although experimentation was completed a year ago. However, after putting lot of effort the reasonably required computational resources were arranged for limited time which were used to complete the requisite training of the models being experimented.

CHAPTER 9: CONCLUSION & FUTURE WORK

For the last few years, neural networks have become an essential part for resolving computer vision related problems especially for image and video classification tasks. CNNs have brought revolution by making the extraction of features from the images automatic. Neural networks work like human brains and find solutions which cannot be calculated with the mathematical applications. Availability of pretrained networks on ImageNet dataset has further enhanced the efficacy of these networks but require big dataset for training of generalized model and suitable computational resources.

In our study, we have tried to solve our problem of classification of live video stream from Pakistani TV News Channels into five classes. Videos consist of image frames arranged in sequence at a specific rate of FPS. We have proposed to resolve this issue using well established pretrained neural networks on ImageNet dataset for image classification and then finetuning the whole model on our especially prepared dataset. Thereafter, for testing of video on the trained model, we applied the averaging dynamic time domain window on the video output for removing jitters and getting the desired classification of the video. In the process we used mainly the variants of ResNet due to its popularity, but we have also carried out experimentation on other networks like AlexNet, DenseNet, ConvNext, VGG and SqueezeNet for comparison purposes. To make it possible, we prepared a new dataset due to non-availability of authentic dataset for this purpose. We gathered approximately 335 hours of videos of different desired classes and then prepared trainable dataset using specially articulated algorithm. Different TV channels broadcast mostly similar nature of programs but in altogether different styles, anchors, sets, context etc. and moreover, these programs keep evolving into different presentation styles or anchors get change etc. This brings drift in the dataset and diminishes viability with passage of time, so needs to have generic and evolving dataset. We were able to generate good results with 95% to 99 % accuracies.

Videos contain a lot of information in terms of image frames, overriding voice, metadata and written scripts in the form of words, sentences etc. In this study we have focused on video classification using image frames only, but other video data can also be used to get accurate video stream classification. For this, separate studies are recommended to be undertaken in future and finally fusing all these contributing factors for better results.

REFERENCES

[1] "Statistic Brain youtube company statistics - Statistic Brain." https://www.statisticbrain.com/?s=Statistic+Brain+youtube+company+statistics&post_type=post (accessed Feb. 07, 2023).

[2] M. Hmayda, R. Ejbali, and M. Zaied, "Classification program and story boundaries segmentation in TV news broadcast videos via deep convolutional neural network," *J. Comput. Sci.*, vol. 16, no. 5, pp. 601–619, 2020, doi: 10.3844/JCSSP.2020.601.619.

[3] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 961–970, 2015, doi: 10.1109/CVPR.2015.7298698.

[4] E. An, A. Ji, and E. Ng, "Large scale video classification using both visual and audio features on YouTube-8M dataset," 2019.

[5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale Video Classification with Convolutional Neural Networks", Accessed: Feb. 12, 2023. [Online]. Available: http://cs.stanford.edu/people/karpathy/deepvideo

[6] K. Soomro, A. Roshan Zamir, and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild," 2012, Accessed: Feb. 12, 2023. [Online]. Available: http://crcv.ucf.edu/data/UCF101.php

[7] X. Xu, W. Tu, and Y. Yang, "Selector-Enhancer: Learning Dynamic Selection of Local and Non-local Attention Operation for Speech Enhancement," 2022, [Online]. Available: http://arxiv.org/abs/2212.03408

[8] N. Takahashi, M. Gygli, and L. Van Gool, "AENet: Learning Deep Audio Features for Video Analysis," *IEEE Trans. Multimed.*, vol. 20, no. 3, pp. 513–524, 2018, doi: 10.1109/TMM.2017.2751969.

[9] "(89) YouTube." https://www.youtube.com/ (accessed Feb. 16, 2023).

[10] "(89) GNN - YouTube." https://www.youtube.com/channel/UC35KuZBNIj4S5Ls0yjY-UHQ (accessed Feb. 16, 2023).

[11] "ARY News - Latest Pakistan News, World News, Business and Sports." https://arynews.tv/ (accessed Feb. 16, 2023).

[12] "Geo News Live - Geo TV Live - WATCH Geo News Live Streaming - Pakistan News Channel." https://live.geo.tv/ (accessed Feb. 16, 2023).

[13] "(89) 24 News HD - YouTube." https://www.youtube.com/channel/UCcmpeVbSSQlZRvHfdC-CRwg (accessed Feb. 16, 2023).

[14] "SAMAA English - Watch all Latest Shows Here - SAMAA English TV Programs - SAMAA." https://www.samaaenglish.tv/tv-programs (accessed Feb. 16, 2023).

[15] "Express News Live - Watch Express News Channel Live." https://www.express.pk/live/ (accessed Feb. 16, 2023).

[16] "Dunya News Live - Dunya Tv Live - Live Streaming Dunya News TV - dunyanews.tv." https://dunyanews.tv/live/ (accessed Feb. 16, 2023).

[17] "Home - AAJ." https://www.aaj.tv/ (accessed Feb. 16, 2023).

[18] "Video classification | TensorFlow Lite." https://www.tensorflow.org/lite/examples/video_classification/overview#performance_benchmarks (accessed Feb. 17, 2023).

[19] "Video Classification | Papers With Code." https://paperswithcode.com/task/video-classification (accessed Feb. 17, 2023).

[20] Y. Liu, "Classification of Videos Based on Deep Learning," *J. Sensors*, vol. 2022, 2022, doi: 10.1155/2022/9876777.

[21] E. Rezazadeh Azar and B. McCabe, "Part based model and spatial-temporal reasoning to recognize hydraulic excavators in construction images and videos," *Autom. Constr.*, vol. 24, pp. 194–202, Jul. 2012, doi: 10.1016/J.AUTCON.2012.03.003.

[22] J. Mccarthy, "WHAT IS ARTIFICIAL INTELLIGENCE?," 2007, Accessed: Mar. 26, 2023. [Online]. Available: http://www-formal.stanford.edu/jmc/

[23] L. F. Huang, *Artificial intelligence,* vol. 4. 2010. doi: 10.1109/ICCAE.2010.5451578.

[24] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science (80-.).*, vol. 349, no. 6245, pp. 255–260, 2015, doi: 10.1126/science.aaa8415.

[25] A. V. Singh *et al.*, "Artificial intelligence and machine learning disciplines with the potential to improve the

nanotoxicology and nanomedicine fields: a comprehensive review," *Arch. Toxicol.*, vol. 97, pp. 963–979, 2023, doi: 10.1007/s00204-023-03471-x.

[26] T. M. Mitchell, "The Discipline of Machine Learning," 2006.

[27] A. Ajit, K. Acharya, and A. Samanta, "A Review of Convolutional Neural Networks; A Review of Convolutional Neural Networks," *2020 Int. Conf. Emerg. Trends Inf. Technol. Eng.*, 2020, doi: 10.1109/ic-ETITE47903.2020.049.

[28] K. O'shea and R. Nash, "An Introduction to Convolutional Neural Networks".

[29] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, 2022, doi: 10.1109/TNNLS.2021.3084827.

[30] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *Proc. 2017 Int. Conf. Eng. Technol. ICET 2017*, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.1109/ICEngTechnol.2017.8308186.

[31] X. Xiao *et al.*, "TRAJECTORIES - BASED MOTION NEIGHBORHOOD FEATURE FOR HUMAN ACTION RECOGNITION".

[32] A. Astorino, A. Fuduli, P. Veltri, and E. Vocaturo, "On a recent algorithm for multiple instance learning. Preliminary applications in image classification Computational Techniques for Biological Data Analysis View project Simpatico 3D View project On a recent algorithm for Multiple Instance Learning. Preliminary applications in image classification", doi: 10.1109/BIBM.2017.8217901.

[33] M. Asadi-Aghbolaghi *et al.*, "A Survey on Deep Learning Based Approaches for Action and Gesture Recognition in Image Sequences," pp. 476–483, 2017, doi: 10.1109/FG.2017.150ï.

[34] D. Silver, J. Schrittwieser, K. Simonyan, I. A.- Nature, and U. 2017, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2016.

[35] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, Sep. 2014, Accessed: Mar. 23, 2023. [Online]. Available: https://arxiv.org/abs/1409.1556v6

[36] C.-C. Yu and B.-D. Liu, "A BACKPROPAGATION ALGORITHM WITH ADAPTIVE LEARNING RATE AND MOMENTUM COEFFICIENT," 2002, doi: 10.1109/IJCNN.2002.1007668.

[37] S. Dong, P. Wang, and K. Abbas, "A survey on deep learning and its applications," *Comput. Sci. Rev.*, vol. 40, p. 100379, May 2021, doi: 10.1016/J.COSREV.2021.100379.

[38] C. Szegedy *et al.*, "Intriguing properties of neural networks," *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, pp. 1–10, 2014.

[39] S. E. Fahlman, "An Empirical Study of Learning Speed in Back-Propagation Networks," 1988.

[40] S. C. Ng, S. H. Leung, and A. Luk, "Fast Convergent Generalized Back-Propagation Algorithm with Constant Learning Rate," *Neural Process. Lett.*, vol. 9, no. 1, pp. 13–23, 1999, doi: 10.1023/A:1018611626332/METRICS.

[41] J. Tang, S. Alelyani, and H. Liu, "Feature Selection for Classification: A Review".

[42] J. Yang, W.-S. Zheng, Q. Yang, Y.-C. Chen, and Q. Tian, "Spatial-Temporal Graph Convolutional Network for Video-Based Person Re-Identification." pp. 3289–3299, 2020.

[43] J. Chang, Z. Lan, C. Cheng, and Y. Wei, "Data Uncertainty Learning in Face Recognition." pp. 5710–5719, 2020.

[44] D. Gehrig, M. Gehrig, J. Hidalgo-Carrio, and D. Scaramuzza, "Video to Events: Recycling Video Datasets for Event Cameras." pp. 3586–3595, 2020. Accessed: Aug. 04, 2021. [Online]. Available: https://github.com/uzh-rpg/rpg_vid2e.

[45] B. Tiwari, V. Tiwari, K. C. Das, D. K. Mishra, and J. C. Bansal, Eds., "Proceedings of International Conference on Recent Advancement on Computer and Communication," vol. 34, 2018, doi: 10.1007/978-981-10-8198-9.

[46] N. Ghosh, S. Agrawal, and M. Motwani, "A Survey of Feature Extraction for Content-Based Image Retrieval System," *Lect. Notes Networks Syst.*, vol. 34, pp. 305–313, 2018, doi: 10.1007/978-981-10-8198-9_32/COVER.

[47] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, Oct. 2007, doi: 10.1093/BIOINFORMATICS/BTM344.

[48] P. Corke, *Advanced visual servoing*, vol. 118. 2017. doi: 10.1007/978-3-319-54413-7_16.

[49] S. Wu *et al.*, "Deep learning in clinical natural language processing: A methodical review," *J. Am. Med. Informatics Assoc.*, vol. 27, no. 3, pp. 457–470, 2020, doi: 10.1093/jamia/ocz200.

[50] T. T. H. Nguyen, A. Jatowt, M. Coustaty, and A. Doucet, "Survey of Post-OCR Processing Approaches," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–37, Jul. 2021, doi: 10.1145/3453476.

[51] T. H. I. Tuyet, H. A. I. Nguyen, A. Jatowt, M. Coustaty, and A. Doucet, "Survey of Post-OCR Processing Approaches," vol. 54, no. 6, 2021.

[52] N. Subramani, M. Greaves, A. Matton, and A. Lam, "A Survey of Deep Learning Approaches for OCR and Document Understanding," no. Section 5, pp. 1–15, 2020.

[53] H. Shrestha, C. Dhasarathan, S. Munisamy, and A. Jayavel, “Natural Language Processing Based Sentimental Analysis of Hindi (SAH) Script an Optimization Approach,” *Int. J. Speech Technol.*, vol. 23, no. 4, pp. 757–766, 2020, doi: 10.1007/s10772-020-09730-x.

[54] R. G. Bhati, “A Survey on Sentiment Analysis Algorithms and Datasets,” *Rev. Comput. Eng. Res.*, vol. 6, no. 2, pp. 84–91, 2019, doi: 10.18488/journal.76.2019.62.84.91.

[55] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, “A Review of Speaker Diarization: Recent Advances with Deep Learning,” 2021, [Online]. Available: http://arxiv.org/abs/2101.09624

[56] R. Mohd Hanifa, K. Isa, and S. Mohamad, “A review on speaker recognition: Technology and challenges,” *Comput. Electr. Eng.*, vol. 90, no. January, p. 107005, 2021, doi: 10.1016/j.compeleceng.2021.107005.

[57] M. Farrús, “Voice disguise in automatic speaker recognition,” *ACM Comput. Surv.*, vol. 51, no. 4, 2018, doi: 10.1145/3195832.

[58] N. H. Tandel, H. B. Prajapati, and V. K. Dabhi, “Voice Recognition and Voice Comparison using Machine Learning Techniques: A Survey,” *2020 6th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2020*, pp. 459–465, 2020, doi: 10.1109/ICACCS48705.2020.9074184.

[59] A. Q. Ohi, M. F. Mridha, M. A. Hamid, and M. M. Monowar, “Deep Speaker Recognition: Process, Progress, and Challenges,” *IEEE Access*, vol. 9, pp. 89619–89643, 2021, doi: 10.1109/ACCESS.2021.3090109.

[60] D. Wang, J. Su, and H. Yu, “Feature extraction and analysis of natural language processing for deep learning english language,” *IEEE Access*, vol. 8, pp. 46335–46345, 2020, doi: 10.1109/ACCESS.2020.2974101.

[61] A. Srivastav and S. Singh, “Proposed Model for Context Topic Identification of English and Hindi News Article Through LDA Approach with NLP Technique,” *J. Inst. Eng. Ser. B*, 2021, doi: 10.1007/s40031-021-00655-w.

[62] T. K. Shih, C.-S. Wang, J. C. Hung, J.-Y. Huang, and C.-H. Kao, “An Intelligent Content-based Image Retrieval System Based on Color, Shape and Spatial Relations Study of Vision-based Indoor Positioning Technique for Mobile Augmented Reality View project Remote Healthcare Platform View project An Intelligent Content-based Image Retrieval System Based on Color, Shape and Spatial Relations,” *Proc. Natl. Sci. Counc. ROC(A)*, vol. 25, no. 4, pp. 232–243, 2001, Accessed: Mar. 29, 2023. [Online]. Available: https://www.researchgate.net/publication/250788298

[63] O. Abdel-Hamid, L. Deng, and D. Yu, “Exploring Convolutional Neural Network Structures and Optimization Techniques for Speech Recognition,” 2013.

[64] J. Wu, “Introduction to Convolutional Neural Networks,” *Introd. to Convolutional Neural Networks*, pp. 1–31, 2017, [Online]. Available: https://web.archive.org/web/20180928011532/https://cs.nju.edu.cn/wujx/teaching/15_CNN.pdf

[65] “An Introduction of CNN: Models and Training on Neural Network Models | IEEE Conference Publication | IEEE Xplore.” https://ieeexplore.ieee.org/document/9727030 (accessed Mar. 30, 2023).

[66] D. Dai, “An Introduction of CNN: Models and Training on Neural Network Models,” *Proc. - 2021 Int. Conf. Big Data, Artif. Intell. Risk Manag. ICBAR 2021*, pp. 135–138, 2021, doi: 10.1109/ICBAR55169.2021.00037.

[67] S. Albawi, … T. M.-… on engineering and, and undefined 2017, “Understanding of a convolutional neural network,” *ieeexplore.ieee.org*, Accessed: Mar. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8308186/

[68] “Getting Started With CNN | LaptrinhX.” https://laptrinhx.com/getting-started-with-cnn-153816276/ (accessed Mar. 31, 2023).

[69] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, Jun. 2017, doi: 10.1145/3065386.

[70] D. C. C. Cire¸san, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, “High-Performance Neural Networks for Visual Object Classification,” 2011.

[71] J. Fieres, J. Schemmel, and K. Meier, “Training convolutional networks of threshold neurons suited for low-power hardware implementation”.

[72] I. Arel, D. Rose, T. Karnowski, D. C. Rose, and T. P. Karnowski, “Deep Machine Learning-A New Frontier in Artificial Intelligence Research Frontier,” 2010, doi: 10.1109/MCI.2010.938364.

[73] S. Ruder, “An overview of gradient descent optimization algorithms *”, Accessed: Mar. 31, 2023. [Online]. Available: http://caffe.berkeleyvision.org/tutorial/solver.html

[74] X. Glorot, A. Bordes, and Y. Bengio, “Deep Sparse Rectifier Neural Networks,” 2011.

[75] K. He, X. Zhang, S. Ren, J. S.-P. of the IEEE, and undefined 2015, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *openaccess.thecvf.com*, Accessed: Apr. 02, 2023. [Online]. Available: http://openaccess.thecvf.com/content_iccv_2015/html/He_Delving_Deep_into_ICCV_2015_paper.html

[76] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier Nonlinearities Improve Neural Network Acoustic Models,” 2013.

[77] B. Karlik, A. O.-I. J. of A. I. and, and undefined 2011, “Performance analysis of various activation functions in generalized MLP architectures of neural networks,” *academia.edu*, Accessed: Apr. 02, 2023. [Online]. Available: https://www.academia.edu/download/53021242/Performance_Analysis_of_Various_Activati20170506-11879-tfq4su.pdf

[78] D. Mishkin and J. Matas, “All you need is a good init,” *4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc.*, 2016, Accessed: Apr. 02, 2023. [Online]. Available: https://github.com/ducha-aiki/LSUVinit

[79] L. Datta, “A Survey on Activation Functions and their relation with Xavier and He Normal Initialization”.

[80] I. Castelli and E. Trentin, “Combination of supervised and unsupervised learning for training the activation functions of neural networks,” *Pattern Recognit. Lett.*, vol. 37, no. 1, pp. 178–191, Feb. 2014, doi: 10.1016/J.PATREC.2013.06.013.

[81] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines”.

[82] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “FAST AND ACCURATE DEEP NETWORK LEARNING BY EXPONENTIAL LINEAR UNITS (ELUS)”.

[83] R. N.-A. intelligence and undefined 1992, “Connectionist learning of belief networks,” *Elsevier*, Accessed: Apr. 02, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0004370292900656

[84] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi, “LEARNING ACTIVATION FUNCTIONS TO IMPROVE DEEP NEURAL NETWORKS”.

[85] K. Hornik, M. Stinchcombe, and H. White, “‘Multilayer feedforward networks are universal approximators’”.

[86] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition”, Accessed: Mar. 23, 2023. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[87] R. Sharma, B. Kaushik, and N. Gondhi, “Character Recognition using Machine Learning and Deep Learning - A Survey,” *2020 Int. Conf. Emerg. Smart Comput. Informatics, ESCI 2020*, pp. 341–345, 2020, doi: 10.1109/ESCI48226.2020.9167649.

[88] J. Nin and V. Torra, “New approach to the re-identification problem using neural networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3885 LNAI, pp. 251–261, 2006, doi: 10.1007/11681960_25/COVER.

[89] L. Deng, G. Hinton, and B. Kingsbury, “NEW TYPES OF DEEP NEURAL NETWORK LEARNING FOR SPEECH RECOGNITION AND RELATED APPLICATIONS: AN OVERVIEW”.

[90] S. Hourri and J. Kharroubi, “A deep learning approach for speaker recognition,” *Int. J. Speech Technol.*, vol. 23, no. 1, pp. 123–131, 2020, doi: 10.1007/s10772-019-09665-y.

[91] D. Sztahó, G. Szaszák, and A. Beke, “Deep learning methods in speaker recognition: a review,” pp. 1–12, 2019, doi: 10.3311/ppee.17024.

[92] S. Hourri, N. S. Nikolov, and J. Kharroubi, “A deep learning approach to integrate convolutional neural networks in speaker recognition,” *Int. J. Speech Technol.*, vol. 23, no. 3, pp. 615–623, 2020, doi: 10.1007/s10772-020-09718-7.

[93] S. Sukhbaatar, R. F. preprint arXiv:1406.2080, and undefined 2014, “Learning from noisy labels with deep neural networks,” *Citeseer*, Accessed: Apr. 02, 2023. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ad84f49b2cd1b85a6d7df2304144a093f5b610a8

[94] P. J. Huber, “Robust Estimation of a Location Parameter,” *Ann. Math. Stat.*, vol. 35, no. 1, pp. 73–101, 1964, doi: 10.1214/aoms/1177703732.

[95] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001, doi: 10.1214/aos/1013203451.

[96] C. Hsu, C. L.-I. transactions on N. Networks, and undefined 2002, “A comparison of methods for multiclass support vector machines,” *ieeexplore.ieee.org*, Accessed: Apr. 02, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/991427/

[97] P. Domingos, “A Few Useful Things to Know about Machine Learning,” *Commun. ACM*, vol. 55, 2012.

[98] C. Willmott, K. M.-C. research, and undefined 2005, “Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance,” *int-res.com*, Accessed: Apr. 02, 2023. [Online]. Available: https://www.int-res.com/abstracts/cr/v30/n1/p79-82

[99] R. Girshick, “Fast R-CNN”, Accessed: Apr. 02, 2023. [Online]. Available: https://github.com/rbgirshick/

[100] S. Safavian, D. L.-I. transactions on systems, undefined man, and undefined 1991, “A survey of decision tree classifier methodology,” *ieeexplore.ieee.org*, 1990, Accessed: Apr. 02, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/97458/

[101] S. S. Kamble, A. Gunasekaran, and S. A. Gawankar, “Sustainable Industry 4.0 framework: A systematic literature review identifying the current trends and future perspectives,” *Process Saf. Environ. Prot.*, vol. 117, pp. 408–425,

Jul. 2018, doi: 10.1016/J.PSEP.2018.05.009.

[102] F. Almaguer-Angeles, J. Murphy, L. Murphy, and A. O. Portillo-Dominguez, “Choosing machine learning algorithms for anomaly detection in smart building iot scenarios,” *IEEE 5th World Forum Internet Things, WF-IoT 2019 - Conf. Proc.*, pp. 491–495, Apr. 2019, doi: 10.1109/WF-IOT.2019.8767357.

[103] S. Lalmuanawma, J. Hussain, and L. Chhakchhuak, “Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review,” *Chaos, Solitons & Fractals*, vol. 139, p. 110059, Oct. 2020, doi: 10.1016/J.CHAOS.2020.110059.

[104] Y. Tanaka, K. Oka, T. Ono, and K. Inoue, “Accuracy analysis of machine learning-based performance modeling for microprocessors,” *Proc. 2016 4th Int. Japan-Egypt Conf. Electron. Commun. Comput. JEC-ECC 2016*, pp. 83–86, Jul. 2016, doi: 10.1109/JEC-ECC.2016.7518973.

[105] D. M. W. Powers, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,” Oct. 2020, Accessed: Apr. 03, 2023. [Online]. Available: http://arxiv.org/abs/2010.16061

[106] “(PDF) The truth of the F-measure.” https://www.researchgate.net/publication/268185911_The_truth_of_the_F-measure (accessed Apr. 03, 2023).

[107] X. Liu, G. Meng, and C. Pan, “Scene text detection and recognition with advances in deep learning: a survey,” *Int. J. Doc. Anal. Recognit.*, 2019, doi: 10.1007/s10032-019-00320-5.

[108] B. Pan *et al.*, “Spatio-Temporal Graph for Video Captioning With Knowledge Distillation.” pp. 10870–10879, 2020.

[109] “Encyclopedia of Machine Learning - Google Books.” https://books.google.com.pk/books?hl=en&lr=&id=i8hQhp1a62UC&oi=fnd&pg=PT29&dq=Sammut+and+G.+I.+Webb,+Encyclopedia+of+machine+learning&ots=91q7Csfy9M&sig=kXlYJTlmxvvpN8T_3sJ63MJ6KB8&redir_esc=y#v=onepage&q=Sammut and G. I. Webb%2C Encyclopedia of machine learning&f=false (accessed Apr. 03, 2023).

[110] “Jaccard Index / Similarity Coefficient - Statistics How To.” https://www.statisticshowto.com/jaccard-index/ (accessed Apr. 03, 2023).

[111] “A Simple Explanation of the Jaccard Similarity Index - Statology.” https://www.statology.org/jaccard-similarity/ (accessed Apr. 03, 2023).

[112] “Image classification | TensorFlow Lite.” https://www.tensorflow.org/lite/examples/image_classification/overview (accessed Apr. 17, 2023).

[113] “Image Classification.” http://www.sc.chula.ac.th/courseware/2309507/Lecture/remote18.htm (accessed Apr. 17, 2023).

[114] D. Lu and Q. Weng, “A survey of image classification methods and techniques for improving classification performance,” *https://doi.org/10.1080/01431160600746456*, vol. 28, no. 5, pp. 823–870, 2007, doi: 10.1080/01431160600746456.

[115] “Image Classification Techniques in Remote Sensing.” https://gisgeography.com/image-classification-techniques-remote-sensing/ (accessed Apr. 18, 2023).

[116] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review,” *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017, doi: 10.1162/NECO_A_00990.

[117] M. Sornam Associate Professor, R. Professor, and R. Scholar, “A Survey on Image Classification and Activity Recognition using Deep Convolutional Neural Network Architecture; A Survey on Image Classification and Activity Recognition using Deep Convolutional Neural Network Architecture,” 2017.

[118] W. Wang, Y. Yang, X. Wang, W. Wang, and J. Li, “Development of convolutional neural network and its application in image classification: a survey,” *https://doi.org/10.1117/1.OE.58.4.040901*, vol. 58, no. 4, p. 040901, Apr. 2019, doi: 10.1117/1.OE.58.4.040901.

[119] B. Jena, G. K. Nayak, and S. Saxena, “Convolutional neural network and its pretrained models for image classification and object detection: A survey,” *Concurr. Comput. Pract. Exp.*, vol. 34, no. 6, p. e6767, Mar. 2022, doi: 10.1002/CPE.6767.

[120] J. Naranjo-Torres, M. Mora, R. Hernández-García, R. J. Barrientos, C. Fredes, and A. Valenzuela, “A Review of Convolutional Neural Network Applied to Fruit Image Processing,” *Appl. Sci. 2020, Vol. 10, Page 3443*, vol. 10, no. 10, p. 3443, May 2020, doi: 10.3390/APP10103443.

[121] P. Dhruv and S. Naskar, “Image classification using convolutional neural network (CNN) and Recurrent Neural Network (RNN): A Review,” *Adv. Intell. Syst. Comput.*, vol. 1101, pp. 367–381, 2020, doi: 10.1007/978-981-15-1884-3_34/COVER.

[122] P. Dhruv and S. Naskar, “Image Classification Using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN): A Review BT - Machine Learning and Information Processing,” pp. 367–381, 2020.

[123] K. Simonyan and A. Zisserman, “Two-Stream Convolutional Networks for Action Recognition in Videos”.

[124] M. Carreras, G. Deriu, L. Raffo, L. Benini, and P. Meloni, “Optimizing Temporal Convolutional Network Inference on FPGA-Based Accelerators,” *IEEE J. Emerg. Sel. Top. Circuits Syst.*, vol. 10, no. 3, pp. 348–361, Sep. 2020, doi: 10.1109/JETCAS.2020.3014503.

[125] S. C. Madanapalli, A. Mathai, H. H. Gharakheili, V. Sivaraman, and E. Engineering, “ReCLive : Real-Time Classification and QoE Inference of Live Video Streaming Services,” 2021.

[126] K. Soomro, A. Roshan Zamir, and M. Shah, “UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild,” 2012, Accessed: Apr. 21, 2023. [Online]. Available: http://crcv.ucf.edu/data/UCF101.php

[127] Z. A. A. Ibrahim, M. Saab, and I. Sbeity, “VideoToVecs: a new video representation based on deep learning techniques for video classification and clustering,” *SN Appl. Sci.*, vol. 1, no. 6, pp. 1–7, 2019, doi: 10.1007/s42452-019-0573-6.

[128] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale Video Classification with Convolutional Neural Networks”, Accessed: Apr. 21, 2023. [Online]. Available: http://cs.stanford.edu/people/karpathy/deepvideo

[129] D. Tran, L. Bourdev, … R. F.-P. of the, and undefined 2015, “Learning spatiotemporal features with 3d convolutional networks,” *openaccess.thecvf.com*, Accessed: Apr. 22, 2023. [Online]. Available: http://openaccess.thecvf.com/content_iccv_2015/html/Tran_Learning_Spatiotemporal_Features_ICCV_2015_paper.html

[130] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, “Temporal convolutional networks: A unified approach to action segmentation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9915 LNCS, pp. 47–54, 2016, doi: 10.1007/978-3-319-49409-8_7/TABLES/1.

[131] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, and I. Laptev Marcin Marszałek Cordelia Schmid Benjamin Rozenfeld, “Learning Realistic Human Actions from Movies,” pp. 1–8, 2008, doi: 10.1109/CVPR.2008.4587756ï.

[132] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, “Temporal Convolutional Networks for Action Segmentation and Detection.” pp. 156–165, 2017. Accessed: Apr. 21, 2023. [Online]. Available: https://github.com/colincsl/

[133] C. Zhang, R. Li, W. Kim, D. Yoon, P. P.-I. Access, and undefined 2020, “Driver behavior recognition via interwoven deep convolutional neural nets with multi-stream inputs,” *ieeexplore.ieee.org*, Accessed: Apr. 22, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9233399/

[134] Y. A. Farha and J. Gall, “MS-TCN: Multi-Stage Temporal Convolutional Network for Action Segmentation.” pp. 3575–3584, 2019.

[135] O. Zatsarynna, Y. Abu Farha, and J. Gall, “Multi-Modal Temporal Convolutional Network for Anticipating Actions in Egocentric Videos.” pp. 2249–2258, 2021.

[136] G. Varol, I. Laptev, C. S.-I. transactions on pattern, and undefined 2017, “Long-term temporal convolutions for action recognition,” *ieeexplore.ieee.org*, Accessed: Apr. 22, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7940083/

[137] J. Donahue *et al.*, “Long-term Recurrent Convolutional Networks for Visual Recognition and Description”.

[138] X. Luo, O. Ye, and B. Zhou, “An modified video stream classification method which fuses three-dimensional convolutional neural network,” *Proc. - 2019 Int. Conf. Mach. Learn. Big Data Bus. Intell. MLBDBI 2019*, pp. 105–108, Nov. 2019, doi: 10.1109/MLBDBI48998.2019.00026.

[139] B. Martinez, P. Ma, S. Petridis, and M. Pantic, “Lipreading Using Temporal Convolutional Networks,” *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2020-May, pp. 6319–6323, May 2020, doi: 10.1109/ICASSP40776.2020.9053841.

[140] X. Li, Y. Wang, Z. Zhou, and Y. Qiao, “SmallBigNet: Integrating Core and Contextual Views for Video Classification.” pp. 1092–1101, 2020. Accessed: Apr. 21, 2023. [Online]. Available: https://github.com/xhl-video/SmallBigNet.

[141] H. H. Pham, L. Khoudour, A. Crouzil, P. Zegers, and S. A. Velastin, “Video-based Human Action Recognition using Deep Learning: A Review,” Aug. 2022, Accessed: Apr. 22, 2023. [Online]. Available: https://arxiv.org/abs/2208.03775v1

[142] “Action Recognition | Papers With Code.” https://paperswithcode.com/task/action-recognition-in-videos (accessed Apr. 22, 2023).

[143] K. Host and M. Ivašić-Kos, “An overview of Human Action Recognition in sports based on Computer Vision,” *Heliyon*, vol. 8, no. 6, p. e09633, Jun. 2022, doi: 10.1016/J.HELIYON.2022.E09633.

[144] Z. Yu and W. Q. Yan, “Human Action Recognition Using Deep Learning Methods,” *Int. Conf. Image Vis. Comput. New Zeal.*, vol. 2020-November, Nov. 2020, doi: 10.1109/IVCNZ51579.2020.9290594.

[145] H. Jhuang, T. Serre, … L. W.-2007 I. 11th, and undefined 2007, “A biologically inspired system for action recognition,” *ieeexplore.ieee.org*, Accessed: Apr. 23, 2023. [Online]. Available:

https://ieeexplore.ieee.org/abstract/document/4408988/

[146] B. D. Ripley, *Pattern recognition and neural networks*. 2007. Accessed: Apr. 23, 2023. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=m12UR8QmLqoC&oi=fnd&pg=PR9&dq=B.+D.+Ripley.+Pattern+recognition+and+neural+networks.&ots=aPPmgLZL-g&sig=FdPK7JxgfMLpg1nzDl5c_gEsirk

[147] X. Glorot, Y. B.-P. of the thirteenth, and undefined 2010, “Understanding the difficulty of training deep feedforward neural networks,” *proceedings.mlr.press*, Accessed: Apr. 23, 2023. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a

[148] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, 2015.

[149] D. Tran, L. Bourdev, … R. F.-P. of the, and undefined 2015, “Learning spatiotemporal features with 3d convolutional networks,” *openaccess.thecvf.com*, Accessed: Apr. 23, 2023. [Online]. Available: http://openaccess.thecvf.com/content_iccv_2015/html/Tran_Learning_Spatiotemporal_Features_ICCV_2015_paper.html

[150] “Video Segmentation: Intro, Methods, Tutorial.” https://www.v7labs.com/blog/video-segmentation-guide (accessed Apr. 24, 2023).

[151] D. Liu, D. Yu, C. Wang, and P. Zhou, “F2Net: Learning to Focus on the Foreground for Unsupervised Video Object Segmentation,” *35th AAAI Conf. Artif. Intell. AAAI 2021*, vol. 3A, pp. 2109–2117, 2021, doi: 10.1609/AAAI.V35I3.16308.

[152] T. Zhou, F. Porikli, D. J. Crandall, L. Van Gool, and W. Wang, “A Survey on Deep Learning Technique for Video Segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, Jul. 2021, doi: 10.1109/TPAMI.2022.3225573.

[153] F. Perazzi, J. Pont-Tuset, … B. M.-P. of the, and undefined 2016, “A benchmark dataset and evaluation methodology for video object segmentation,” *cv-foundation.org*, Accessed: Apr. 24, 2023. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Perazzi_A_Benchmark_Dataset_CVPR_2016_paper.html

[154] W. D. Jang and C. S. Kim, “Streaming video segmentation via short-term hierarchical segmentation and frame-by-frame Markov random field optimization,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9910 LNCS, pp. 599–615, 2016, doi: 10.1007/978-3-319-46466-4_36.

[155] I. Budvytis, V. Badrinarayanan, R. C.-C. 2011, and undefined 2011, “Semi-supervised video segmentation using tree structured graphical models,” *ieeexplore.ieee.org*, Accessed: Apr. 24, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5995600/

[156] V. Badrinarayanan, … I. B.-I. transactions on, and undefined 2013, “Semi-supervised video segmentation using tree structured graphical models,” *ieeexplore.ieee.org*, Accessed: Apr. 24, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6475946/

[157] B. Liu, X. H.-P. of the I. conference on computer, and undefined 2015, “Multiclass semantic video segmentation with object-level active inference,” *cv-foundation.org*, Accessed: Apr. 24, 2023. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Liu_Multiclass_Semantic_Video_2015_CVPR_paper.html

[158] R. Yao, G. Lin, S. Xia, J. Zhao, and Y. Zhou, “Video Object Segmentation and Tracking,” *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 4, Jul. 2020, doi: 10.1145/3391743.

[159] C. Wu, S. Shao, C. Tunc, S. H.-2020 I. 17th, and undefined 2020, “Video anomaly detection using pre-trained deep convolutional neural nets and context mining,” *ieeexplore.ieee.org*, Accessed: Apr. 24, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9316538/

[160] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, “Convolutional learning of spatio-temporal features,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6316 LNCS, no. PART 6, pp. 140–153, 2010, doi: 10.1007/978-3-642-15567-3_11.

[161] “Live News BOL | BOL News Live watch online streaming.” https://www.bolnews.com/live/ (accessed Mar. 10, 2023).

[162] “ہم یوز ن - تان سد اک پ سے اردو ں بری خ Hum News - Latest News, Pakistan News.” https://www.humnews.pk/ (accessed Mar. 10, 2023).

[163] “HUM News - YouTube.” https://www.youtube.com/channel/UC0Um3pnZ2WGBEeoA3BX2sKw (accessed Mar. 10, 2023).

[164] “PTV’s Official Web Portal.” https://ptv.com.pk/ptvNews (accessed Mar. 10, 2023).

[165] “PTV News - YouTube.” https://www.youtube.com/c/PTVNewsOfficial (accessed Mar. 10, 2023).

[166] “92 News Live TV, First Hd Plus Channel of Pakistan.” https://92newshd.tv/ (accessed Mar. 10, 2023).

[167] “92 News HD - YouTube.” https://www.youtube.com/@92newshdTv (accessed Mar. 10, 2023).

[168] “Models and pre-trained weights — Torchvision 0.15 documentation.”

https://pytorch.org/vision/stable/models.html#classification (accessed Apr. 04, 2023).

[169] “A schematic view of ResNet architecture [15], decomposed into three... | Download Scientific Diagram.” https://www.researchgate.net/figure/A-schematic-view-of-ResNet-architecture-15-decomposed-into-three-blocks-embedding_fig1_333475917 (accessed Apr. 09, 2023).

[170] “Basics of Machine Learning Image Classification Techniques.” https://iq.opengenus.org/basics-of-machine-learning-image-classification-techniques/ (accessed Apr. 13, 2023).

[171] “Models and pre-trained weights — Torchvision main documentation.” https://pytorch.org/vision/stable/models.html#classification (accessed Mar. 11, 2023).

[172] A. Krizhevsky and G. Inc, “One weird trick for parallelizing convolutional neural networks,” 2014.

[173] “AlexNet — Torchvision main documentation.” https://pytorch.org/vision/stable/models/alexnet.html (accessed Mar. 11, 2023).

[174] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size,” Feb. 2016, Accessed: Mar. 23, 2023. [Online]. Available: https://arxiv.org/abs/1602.07360v4

[175] “[1602.07360] SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size.” https://arxiv.org/abs/1602.07360 (accessed Mar. 23, 2023).

[176] “SqueezeNet — Torchvision 0.15 documentation.” https://pytorch.org/vision/stable/models/squeezenet.html (accessed Mar. 23, 2023).

[177] “[1608.06993] Densely Connected Convolutional Networks.” https://arxiv.org/abs/1608.06993 (accessed Mar. 23, 2023).

[178] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 2261–2269, Aug. 2016, doi: 10.1109/CVPR.2017.243.

[179] M. Steen, S. Downe, N. Bamford, and L. Edozien, “DenseNet:Densely Connected Convolutional Networks arXiv:1608.06993v5,” *Arxiv*, vol. 28, no. 4, pp. 362–371, 2018.

[180] “DenseNet — Torchvision 0.15 documentation.” https://pytorch.org/vision/stable/models/densenet.html (accessed Mar. 23, 2023).

[181] “[1409.1556] Very Deep Convolutional Networks for Large-Scale Image Recognition.” https://arxiv.org/abs/1409.1556 (accessed Mar. 23, 2023).

[182] “VGG — Torchvision 0.15 documentation.” https://pytorch.org/vision/stable/models/vgg.html (accessed Mar. 23, 2023).

[183] “ResNet — Torchvision 0.15 documentation.” https://pytorch.org/vision/stable/models/resnet.html (accessed Mar. 23, 2023).

[184] “resnet18 — Torchvision 0.15 documentation.” https://pytorch.org/vision/stable/models/generated/torchvision.models.resnet18.html#torchvision.models.resnet18 (accessed Mar. 23, 2023).

[185] “resnet34 — Torchvision 0.15 documentation.” https://pytorch.org/vision/stable/models/generated/torchvision.models.resnet34.html#torchvision.models.resnet34 (accessed Mar. 23, 2023).

[186] “resnet50 — Torchvision 0.15 documentation.” https://pytorch.org/vision/stable/models/generated/torchvision.models.resnet50.html#torchvision.models.resnet50 (accessed Mar. 23, 2023).

[187] “resnet101 — Torchvision 0.15 documentation.” https://pytorch.org/vision/stable/models/generated/torchvision.models.resnet101.html#torchvision.models.resnet101 (accessed Mar. 23, 2023).

[188] “resnet152 — Torchvision 0.15 documentation.” https://pytorch.org/vision/stable/models/generated/torchvision.models.resnet152.html#torchvision.models.resnet152 (accessed Mar. 23, 2023).

[189] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” Jan. 2022, Accessed: Mar. 19, 2023. [Online]. Available: http://arxiv.org/abs/2201.03545

[190] Z. Liu, H. Mao, C. Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2022-June, pp. 11966–11976, Jan. 2022, doi: 10.1109/CVPR52688.2022.01167.

[191] “ConvNeXt — Torchvision 0.15 documentation.” https://pytorch.org/vision/stable/models/convnext.html (accessed Mar. 19, 2023).