Criminal Activity Detection Using Deep Learning Algorithms

by

Zarin Tasnim
16201027
Syeda Sanjana Shahid
16201082
Sofana Quayum
17201143
Umme Habiba Barsha
17301211

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
January 2021

© 2021. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Zarin Tasnim
16201027

Syeda Sanjana Shahid
16201082

Sofana Quayum
17201143

Umme Habiba Barsha
17301211

Approval

The thesis titled "Criminal Activity Detection Using Deep Learning Algorithms" submitted by :

1. Syeda Sanjana Shahid (16201082)
2. Zarin Tasnim (16201027)
3. Sofana Quayum (17201143)
4. Umme Habiba Barsha (17301211)

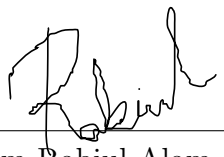
Of Fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 15, 2021.

Examining Committee:

Supervisor:
(Member)

Amitabha Chakrabarty, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:



Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Mahbubul Alam Majumdar, PhD
Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Criminal activities using guns and knives occur very frequently. The quick and accurate detection of a criminal activity is paramount to securing a place where people usually gather every day. More and more security systems are being developed as the number of cities are growing rapidly. This creates a backlog of video data that is being monitored under human supervision but usually human error happens in such cases. This also creates a huge amount of workload for the supervising team. There are several solutions in computer science that can be implemented for immediate and accurate criminal activity detection without any human intervention. Human behavior and pattern recognition is a challenge when it comes to criminal detection as there are several people who act in an abnormal way but aren't suspicious. In such cases it might generate a false alarm. As we proceed further with our research, most of the crimes take place with the use of handguns or knives. There are many more studies from different countries that show that, most dangerous crimes took place using weapons of different sorts. So, in order to detect a criminal from a live crime scene, the first and the quickest step is to determine whether a person is carrying an arm or not. For such detection method, Convolutional Neural Networks is very useful. That's why among all different types of Deep Learning approaches, we opted for Convolutional Neural Network (CNN) to identify a criminal using object detection method. The major challenge of our research was the unavailability of datasets. We created our own image dataset and classified them into four different classes in order to train our model. We have 4,180 images in our dataset which are collected from different crime scenes. There are several CNN models that give efficient results in terms of object detection from image datasets. In our work, we implemented five different CNN models which are MobileNetV2, Inception-v3, Xception, VGG16, ResNet50 and as a result accuracy for each model is 98%, 98%, 94%, 70% and 60% respectively. The accuracy in MobileNetV2 and Inception-v3 was the highest.

Keywords: Criminal identification; Crminal Activity Detection using Deep Learning; Convolutional Neural Network; VGG16; ResNet50; MobileNetV2; InceptionV3; firearm detection; knife detection.

Dedication

We would like to dedicate this thesis to our loving parents,Supervisor, friends and everyone that helped us with this paper.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our advisor Dr.Amitabha Chakrabarty sir for his kind support and advice in our work. He helped us whenever we needed help.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI Artificial Intelligence

CNN Convolutional Neural Network

DCNN Deep Convolutional Neural Network

HAR Human Activity Recognition

HDL Hybrid Deep Learning

IMFDB Internet Movie Firearm Database

NN Neural Network

R-CNN Region Based Convolutional Neural Network

ReLu Rectified Linear Units

ResNet Residual Network

RGB Red Green Blue

SSD Single Shot Detector

VGG Visual Geometry Group

YOLO You Only Look Once

Chapter 1

Introduction

Activity recognition is a process of identifying the actions from a series of observations involving a person's regular activities. Once a series of observations of human movement and actions recorded a similar pattern can be identified through which we can differentiate different human activities. When a study is done on human activities we call it human activity recognition. Human activity recognition is a broad field which involves image processing using machine learning. For example, a human activity could be either sleeping, walking or running. In all of those activities a kind of similar pattern is followed. A particular set of activities can be detected and identified manually by HAR [5]. As sleeping involves and idle in the human body or walking involves a regular movement of the limbs, criminal activities also have some common patterns. Of the most common patterns regarding criminal activities in activity recognition is a process of identifying the actions from a series of observations involving a person's regular activities. Once a series of observations of human movement and actions is laves the carriage of melee weapons or firearms and carriage of handguns and pistols is very common in performing a wide range of criminal activities [14]. Starting from a local thug to a worldwide terrorist organisation all the criminals involve the usage of firearms. These firearms are used in various crimes such as robbery, murder , mass terrorism etc. For this reason a strong security and prevention system is required to minimize criminal activities and assistance from technological advancements are very essential to improvise security systems. The most widely used device for improving security is video footage which consists of a system for video surveillance in different areas like schools, shopping malls, public places, residential and commercial areas. But the effectiveness of video or image data-set is questionable as it requires train supervisors and human attention. Several researches state that 90% of the footage is not watched or focused on attentively. In such scenario it is of no use .This is where human activity recognition based on image processing and machine learning can play a crucial role to facilitate the security system. With the increasing amount of crime and terrorism rate globally an automated surveillance system is a burning need. As we live in a society, we are bound to interact with the other members of the society. Within this interaction it is extremely difficult to identify normal and criminal activities [20].

1.1 Motivation

Problems are increasing with increasing amounts of technological facilities. Solution is also technology. Murders, robbing, terrorism, rape - everything is increasing day by day. Security is the main concern in the modern world and people are now fully dependent on computers for their security. Terrorists or other criminals can attack anywhere and any time. So only guards are not able to cover the full situation. Innocent people are dying because of the criminals. Underdeveloped as well as developed countries are suffering from these criminal activities. If machine learning can detect the criminal by its weapon it can alert the security or authority of that place which can prevent that crime. Image processing is doing great in this matter. Lots of projects and researches are held to solve this problem. It can detect both criminal and non criminal activities by comparing them individually [2] .

1.2 Major Contribution

The main aim of our research work is to reduce the pressure on the controller who are appointed in areas that need high security system like airports, bus stations , metro station, multinational companies, railway stations, cantonment areas, shopping malls entrances and similar places. Our models will identify the criminals from the images and we have trained our models like that so that it can differentiate between criminal and non criminal activity . It will be very easy to detect crime or protect our society if we make machine learning do the work for us .

1.3 Thesis Orientation

In Chapter 1 we gave a introduction of our thesis. Reviews of the papers which have been reviewed by us are in the Chapter 2. We have reviewed various papers related to our work and literature review of similarities,dissimilarities of the papers between our methodology of work. In chapter 3, we have discussed background analysis of our thesis. Chapter 4 is all about CNN which is an important part in our thesis.We have described our methodologies in chapter 5. The result and analysis of our work is in Chapter 6. Finally our entire paper is concluded and summarized in Chapter 7.

Chapter 2

Literature Review

Many recent researches related to image datasets, have been done using deep learning (CNN) models. For example, AlexNet [6], ImageNet [4], GoogleNet [10], VGG-Net for object detection of a specific class. Generally, a CNN model consists of multiple layers and the last layer of CNN generates the activation which is basically, used for object detection. In order to classify images based on object detection a lot of supervised and unsupervised learning method is being used till date. Recently most advancements regarding object detection using deep learning techniques are implemented due to region proposal methods [7] and region-based convolutional neural networks (R-CNNs) [8]. Lately, in some work webcam-based deep learning approach is used for object detection. Such as, two models of Faster R-CNN: Inception V2 and Resnet50 is used to train a model to detect specific objects from real time data that is viewed through a webcam [35].

Furthermore, in a research done by Navalgund et al [23] used CCTV camera based criminal activity detection method where pre-trained deep learning model VGGNet-19 was used. Their researches were more focused on detecting weapons like guns and knives in criminal's hand. The result of VGGNet-19 was later on compared with other two deep learning algorithms which are GoogleNet and Inception V3 but the accuracy in terms of training, was better in VGG19. In this research, images of people holding objects which look similar to guns and knives are used as negative images and they are classified in not a weapon class to train the model. The accuracy of VGG19 was 100% for crime datasets whereas in GoogleNet inception the accuracy is 87%. VGG19 classifies the objects present in an image at FC layer which is good in terms of classification whereas in Googlenet object classification happens in Max layer where the accuracy is less as it uses one FRCNN algorithm only, on the other hand VGGnet19 uses FRCNN and RCNN both for object localization in an image. VGGNet19 converts raw datasets into 244×244 image size which is a standard one.

Recently, models like auto regressive are being used in terms of forecasting any criminal act but there are some shortcomings. That's why many works are being done using Neural Networks. Another paper by Steven Schmitt et al [22] proposed another method for criminal activity detection using neural networks in combination with a Hybrid Deep learning algorithm which analyzes data from video streams. The HDL algorithm is used for high level relational feature extraction from each and

every frame for face identification. For the implementation process HDL, DCNN and RNN are used in this work. In this work, processed video frames are fed into the model and then it detects any object or human being present in the frames then any abnormal behavior detected is being monitored. HDL algorithm is used for facial recognition through extracting features of high performance from each frame. Facial recognition systems with high accuracy rate is built using multiple stages. First facial landmarks are being detected. The convolution is being performed for detailed features extraction from those facial landmarks. In this paper DNN is also used for object tracking and detection in crowd. The multilayer perceptron in DCNN is used for face recognition which is designed to work with HDL. RNN is used for temporal behavioral feature extraction from the streaming data. This is also used in conjunction with the HDL algorithm to build a model for crime detection.

Handgun and knife detection indeed are one of the most challenging tasks due to variation in terms of viewpoint, background cluttering and occlusion that occurs very much frequently in many scenes. In this study by Arif Warsi et al [34] categorized various algorithms for detecting guns and knives. This paper basically explains the classification of algorithms in terms of handgun and knife detection, illustrates deep learning and Non-Deep learning algorithms for criminal identification.

A study conducted by Michal Grega et al [12] worked on various algorithms to detect a firearm which offers a very low rate of false alarm which will alert human supervisor whenever guns and knives are identified in an image. These sort of studies focuses on image datasets to detect weapons that might cause a criminal action. The results of this research are presented in 4 table where edge histogram descriptor and homogeneous texture descriptor is used. In terms of knife detection edge histogram was proved to be better the true negatives were in large number and false positive is just 5% meaning that the false alarm rate is minimum and the accuracy is 97% in terms of edge histogram. There are other algorithms which are being tested on the same datasets but the accuracy was less. On the other hand, the homogeneous texture descriptor gave worse result but according to their research it still can be used for filtering false alarm out as it has a lower false alarm rate which is 7% with high specificity which is 93%. In terms of knife detection algorithm the specificity is 94.93% and sensitivity is 81.18%. This paper deals with image datasets which are poor in quality with low resolution as the images from CCTV footage do not have high resolution. In terms of firearm detection, the specificity and sensitivity are 96.69% and 35.98% respectively for video datasets that contain harmful objects whereas they found 100% specificity in terms of video dataset that does not contain harmful objects. Their implementation is for different types of places like "banks" with good lighting but less distance between the CCTV camera and the criminal holding dangerous objects as it's a confined space and "streets" where they have to deal with poor lighting also high distance between the CCTV camera and the person.

Convolutional Neural Network has been proved to be the most efficient Deep Learning algorithm when it comes to object detection using image datasets. A study by Gyanendra et al [21] used Deep Convolutional Network (DCN), Faster Region-based CNN model through transfer learning to detect harmful objects like guns from clut-

tered scene. The detection process evaluation was done over IMFDB, a benchmark gun database. This paper used VGG16 which is pre-trained on the ImageNet data set and gained 92% accuracy.

Many datasets are built using images from CCTV footage. In this paper by Jose Luis et al [31] similar datasets were built and trained using Faster R-CNN. They applied Faster R-CNN with the help of Feature Pyramid Network along with ResNet-50 which resulted in a model that detect harmful weapons in order to detect guns from an ongoing crime scene. They have not included knives or other dangerous objects in their study. In this paper 14 models were performed with different types of combination over four test datasets which helped them analyze the comparison of performance for all different models.

Another research has been made by Tharinda Dilshan et al [37] which proposes concealed weapon detection. The proposed system uses image fusion to fuse RGB and IR images to create a detailed image of the people visible in the crime scene along with hidden firearms. For such detection phase two convolutional models are being used, YOLO followed by VGG Net. In this case, YOLO is applied for feature extraction purposes in order to identify people present in the image which is sent to VGG Net as inputs to detect firearms. The major Limitation of this paper was the absence of proper datasets. In this paper they had to focus on datasets like IR images so that the proposed model can detect concealed weapons with more accurate details when combined with the corresponding RGB image.

Another research by Mohammad S. Hasan et al [19] has been done on feature extraction which includes four different algorithms to detect a specific dangerous object. In this particular paper, knives are being targeted so that criminal activity that includes knife can be detected within a short period of time. Among all different deep learning algorithms best results are being found in HOG-SVM, Bag of words, pre-trained Alexnet CNN and CNN. In the dataset, they used 1000 images that include knife and 1000 background images that does not include knife. They used random dataset where the algorithm picks 500 images randomly to train and 500 images to test from every category. For further analysis two-times-two-folds validation is being done. The highest accuracy was obtained from Alexnet+SVM. However, BoW also gave high accuracy in addition to that when it is a matter of time consumption, this method is noticeably ahead of the neural networks.

Chapter 3

Background Study

In this section, we have discussed the background analysis of crime and security and also image processing techniques and its applications.

3.1 Crime vs Security

Human Civilization began to evolve long ago. With the passage of time society evolved as well as the interaction between the people. There has always been law and order to maintain the sanctity of a good society. In spite of the existence of law there has been crime and other activities that are not supported by either society or religion. With advancements in all sectors and increasing population criminal activities have also increased in a vast number. A significant portion of crime around the globe is committed with the help of different firearms. Criminal activities such as terrorism, murder, robbery etc concern a high involvement of firearms. In this modern world of the 21st century possession of a handgun or procurement of bigger firearms such as rifles from the black market is not a difficult task. So criminal activities involving the aforementioned items has become a quite common scenario. With the evolution of technology firearms have become so compact that one can easily keep a handgun even in a cloak's pocket. As a result the situation right now is very alarming. But the aggression of this technology can be countered by evenly growing technologies that can assist growing a better security and prevention system. In this era of technology we are completely dependent on computers for maintaining our security system that can be upgraded to be aware of threats posed by someone who is in possession of a gun. It's not possible for security personnel to be fully aware of everyone's activities. But machine learning can play a crucial role in identifying who is a threat or who is not. Taking coverage from CCTV footage a machine learning platform can alert us about someone who is posing a threat with a gun. Nowadays both developing and underdeveloped countries are also coming under CCTV surveillance in part if not in full. So a criminal carrying a gun who is about to commit some crime will definitely catch the eye of the camera trained with machine learning and this platform will understand and let the supervisors know that a crime is about to take place. In this regard human activity recognition can make the maintenance of security easier. It can detect both the criminal and the general people by comparing their activities individually. Show the incorporation of machine learning and human activities recognition into the surveillance system can make the identification of criminals a lot easier and less time consuming [11]. If

the threat is identified ahead of time the criminal activities can be stopped before it takes place. The implementation of this idea not only helps us stop crime but its impact on the society results in such a manner that criminal will think twice before committing a crime such as robbery, murder, terrorism with the help of guns and other firearms.

3.2 Image Processing

Image processing is one of the most crucial parts of AI. AI has seen an exponential growth in recent days even in the smartphone industry. A survey has shown that AI has grown a value of 2.7 trillion US dollar within this 2020 in the sector of sales and marketing also in supply chain planning and manufacturing. Image processing has played a great role in the flourishing of AI. Image processing is the system of analysing the technical details of an image. It involves a complex algorithm which uses the image as the input and uses its useful information to return output. It is projected that the image processing industry will reach a revenue of 38.9 billion US dollar by 2021. Discrete targets are examined in image processing. AI and machine learning provides a different dimension in the world of image processing. A very well known complex algorithm is Google lens which analyzes complex images involving deep learning and AI to process them. Google lens is so useful that it even helps translation of foreign languages within every instance. Google lens is a service of Google which involves AI and machine learning to help us process complex images to identify different things and writings as well as translation and illustrations. It is even more useful that it can suggest actions based on the details of the image [29]. A further integration of AI with Google assistant can make our queries get resolved in a very easy and expeditious manner. As a result, deep learning and image processing efforts are saved. AI and machine learning has been accomplishing wonders in the world of image processing [2]. Image archives contain a large number of images. Hence it's very difficult to find out the required or targeted image from such a vast collection. Therefore a complex algorithm is required to filter images so that we can find out our required one. In order to find out the real solutions to our queries images are to be e categorised on the basis of their details and content based annotations. If this process is attempted manually it will be very expensive and time consuming. If the images are classified according to their shape, texture, position, colours etc it becomes easier to find out the targeted one. In this regard the details of the subject in the image is analysed through various algorithms. It not only classify the images But also find out various details and characteristics of them. This type of advancements started quite a long ago with the hands of speech recognition. Now it is resulting in very much precise image classification through image processing and it has such a bright future that dedicated neural processing units are deployed to handle this kind of work. Deep learning techniques can be very useful to extract complicated information from the images taken as input through its different multilevel structures. Deep learning not only provides image classification but also helps us with data labelling and in some cases suggested actions. A neural network architecture is incorporated in successful deep learning. Now a days we can see e a wide range of applications of deep learning such as self driven cars, news aggregation and detection of 13 fraud news, language processing, entertainment sector, visual recognition, health care, colouring black and white images, adding

sound to videos, handwriting generation, game playing micro, language translation, pixel restoration, image description, demographic surveys etc. In this large world of information and technology retrieval of data from a huge database is a very time consuming act. And extracting useful information in an effortless and time-saving manner can lead to a generational success in all the industries based on or related to technology. Incorporation of machine learning and AI can provide the security system with an extra layer of protection by warning the incoming threats with the help of deep learning algorithms to identify them ahead of time. Hence deep learning and image processing creates a shield for the security systems by saving both labour and time whilst extracting required and useful information and leading features.

Image processing is divided into two different methods :

1.Analogue image processing is the method which processes physical photographs or images of hard copies. Input and output both are images.

2.Digital image processing is the process which manipulates digital form of images or we can say a soft copy of images or any other digital version of images by computer algorithms. Here output might be an image or any other digital information or features connected to the image.

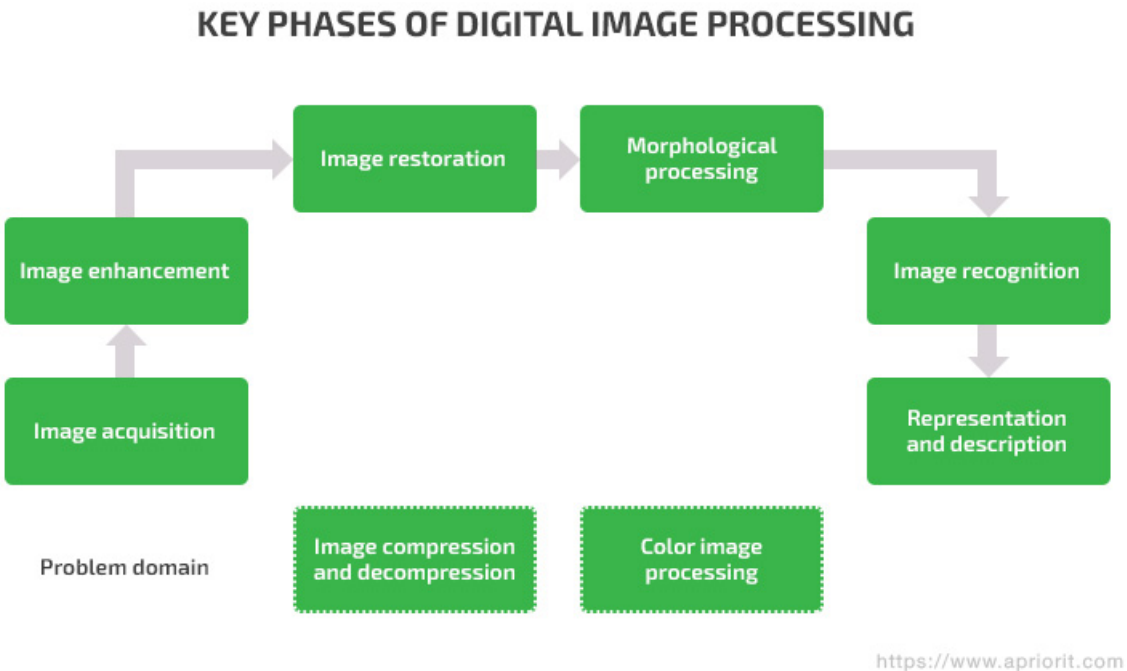


Figure 3.1: Phases Of Image Processing [33]

Image processing has 8 phases. They are :

1.Image Acquisition is the process where the image is being captured by a sensor like a camera. Then it converts the image into possible digital image files.

2.Image enhancement is a process which helps to improve the quality of an image so

that it can be extracted hidden features and formation from the image which will be needed for further processing.

3.Restoration is the second phase of quality improvement process which improves the quality by removing possible noise to get the cleaner version of that file. There are some features like blur, watermarks, noise, miss focusing which are very common for an image. Restoration works to get rid of these noises from an image to get a clearer version of an image.

4.Colour image processing is the method of processing any coloured image. It is mainly the process of processing pseudo colour, RGB etc.

5.Image compression and decompression is the process where the image is being compressed or resized to get the flexible pattern of an image that is needed for any particular project.

6.Morphological processing is the system where it can detect or explain the shape, size, structure of the objects of that particular image. This method is mainly used for creating data sets for training machine learning algorithms or any other AI models. When it can train a model in a particular system so that the model can recognize the expected object from an image.

7.Image Recognition is the process of identifying specific features or objects from a particular image. This is a very important process which is often used in various AI and machine learning researches and projects.

8. Representation And description is the process of describing any process data from an image input. If we look into the raw output we can say numbers of arrays which represents digital information of the model which was trained by the image data set.

Chapter 4

Convolutional Neural Network, Dataset and Tools

In this chapter, we have explained about the convolutional neural network and how it works. The dataset used and the main libraries that we have used to implement the models are also discussed in this section.

4.1 Convolutional Neural Network

The deep neural network has been recognised as the most important technique in recent years and has been widely popular in literature as it can manage massive quantities of data. The models that we have used for our project fall under CNN, one of the most prominent algorithms of Deep Learning for image classification, recognition and object detection. CNN is a type of artificial neural network that is specifically designed for the processing of pixel data in image recognition and processing. Classifications of CNN images work by taking an image, analyzing it labelling it under defined categories. It consists of many hidden layers which take inputs, detect patterns using filters, then transform the inputs and send it to the next layer [9]. CNN has five main layers which are discussed below.

4.1.1 Input Layer

The input layer is an image with the resulting measurements of width, height, and depth. As for an example, the input is $64 \times 64 \times 3$ where the width=64, height=64 and depth=3, the depth represents RGB channels here. If the image is 224×224, it needs to be converted to 50176×1. If the input is X, all the samples looking like X should be observed and categorized by CNN.

4.1.2 Convolutional Layer

Convolution is the first layer to extract features from an input image. By the use of restricted input data squares to learn object functions, Convolution retains the relationship between pixels. Minimal input squares are used for learning image features which helps Convolution preserve the relationship between pixels. This process takes two inputs, such as the image matrix and a kernel. Output neurons are connected by computation to local areas. By choosing 1 or more characteristics

of an image and creating one or more matrix and dot product utilizing image 17 matrix,it will eventually give the result that is the convolution layer. If the size of the image is $M \times M$ and the filter size is $E \times E$ after convolution ,the equation is:

$$(\mathrm{M} \times \mathrm{M}) \times (\mathrm{E} \times \mathrm{E}) = (\mathrm{M} - \mathrm{E} + 1) \times (\mathrm{M} - \mathrm{E} + 1) \tag{4.1}$$

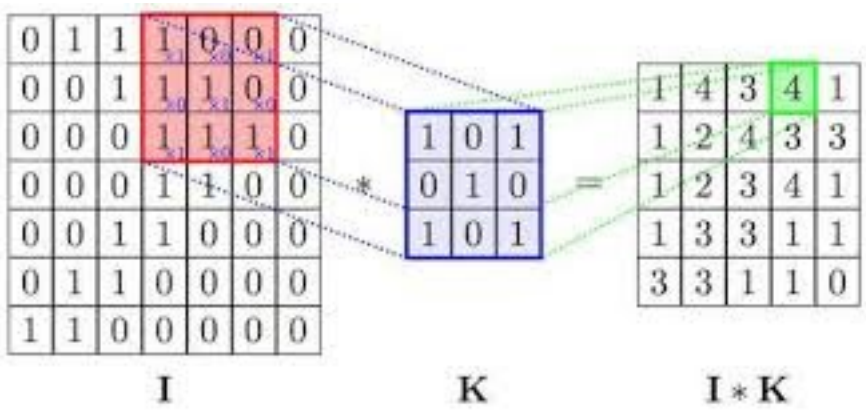


Figure 4.1: Convolutional Layer [27]

4.1.3 Pooling Layer

This layer aims to decrease the image's spatial features, network parameters and overall computation. Each feature map has its own pooling layer. Max pooling is the most common approach as it takes the largest element. At first the RELU layer is converted into a 4×4 matrix which then reduces the data into a 2×2 matrix through repeated iterations of this process.

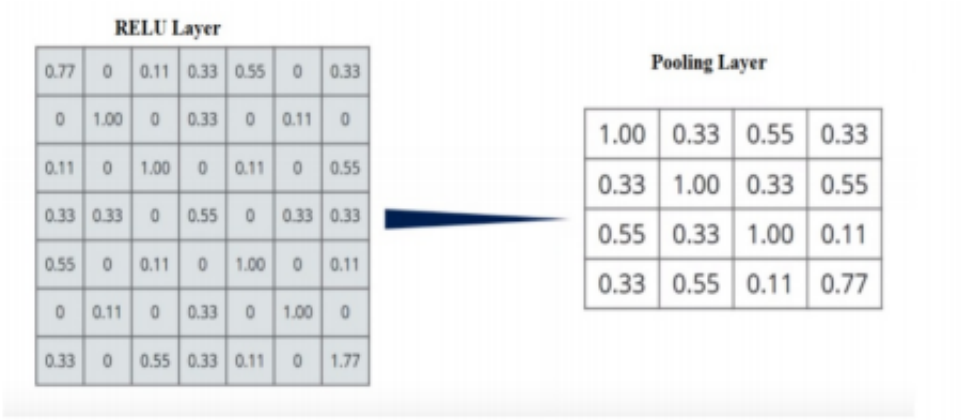


Figure 4.2: Pooling Layer [27]

4.1.4 Fully Connected Layer

The matrix transforms into a vector after the input image goes through the pooling layer and then it goes through a fully connected layer like a neural network. Fully Connected Layer calculates class scores for the column $1 \times 1 \times 12$ for the following picture because there are 3 functions selected and a matrix was generated in the pooling layer for each function. When the number of functions selected was 2, the matrix $1 \times 1 \times 8$ for the same image would have been created.

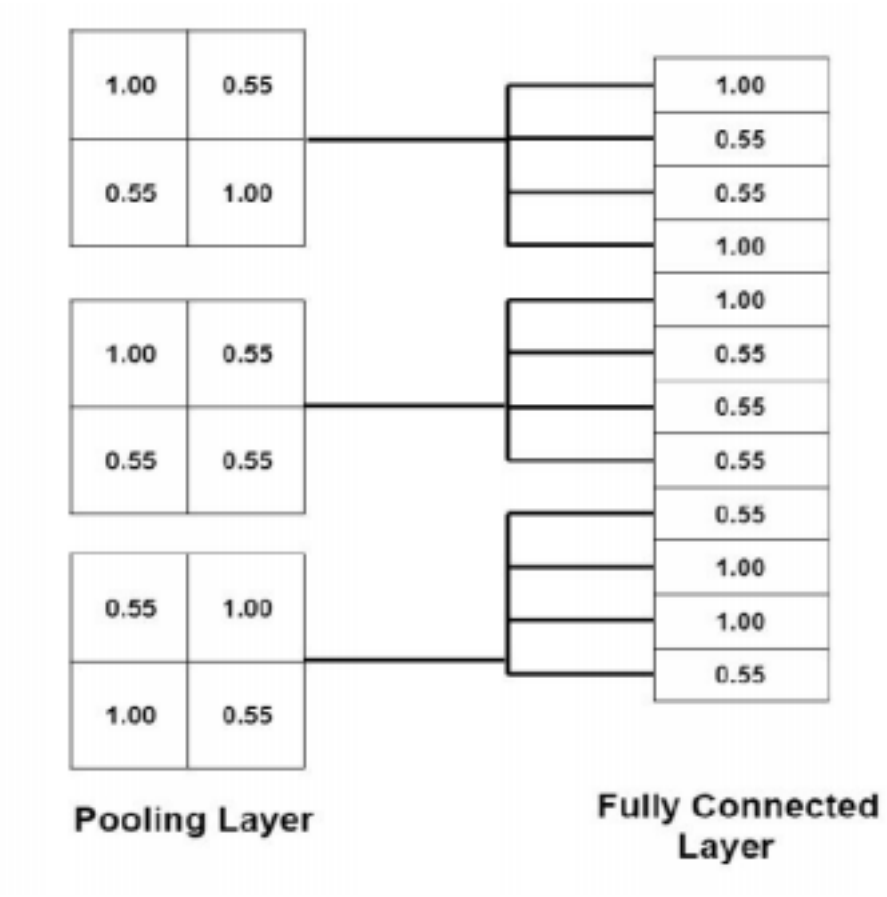


Figure 4.3: Fully Connected Layer [27]

4.1.5 Output Layer

To determine the final output, we need to apply a fully connected layer to generate an output which is equal to the classes. The layer output contains the 1 dot programmed label. All the data is saved and it is marked as X. It checks how many similarities when another new image is given and then detects whether or not the image is X by providing the data that is saved in its memory. Likewise, CNN then transforms the original pixel image to the ultimate class scores from the original pixel values.

Figure 4.4 shows the entire CNN architecture with all the layers.

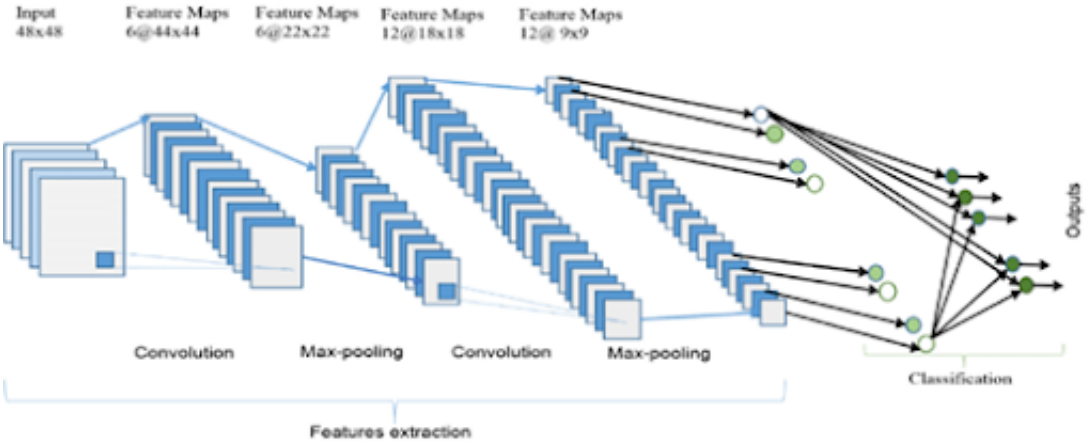


Figure 4.4: Entire CNN Architecture [27]

4.2 Dataset

In our research work we have worked with image datasets. We classified our dataset into four classes which are Criminal with gun, Criminal with knife, Police and Military. Datasets in most research works which are quite similar to ours, are built using images from television shows, video games, anime and so on (IMFDb) but this caused lower accuracy in terms of real-life application. That's why we opted for more realistic image datasets that represent the real-life scenario. For criminal with guns and criminal with knife categories, we took images from various crime scenes where criminals are holding guns or knives in hand or pointing it towards normal people. For the police and military category, we collected images of Bangladesh police and army officers from authentic sources. Right now, we have 4180 images in our dataset. The data collection process will continue for the betterment of our research work. We fed our datasets as inputs into five different CNN models which are VGG16, ResNet50, MobileNetv2, Xception and Inceptionv3.

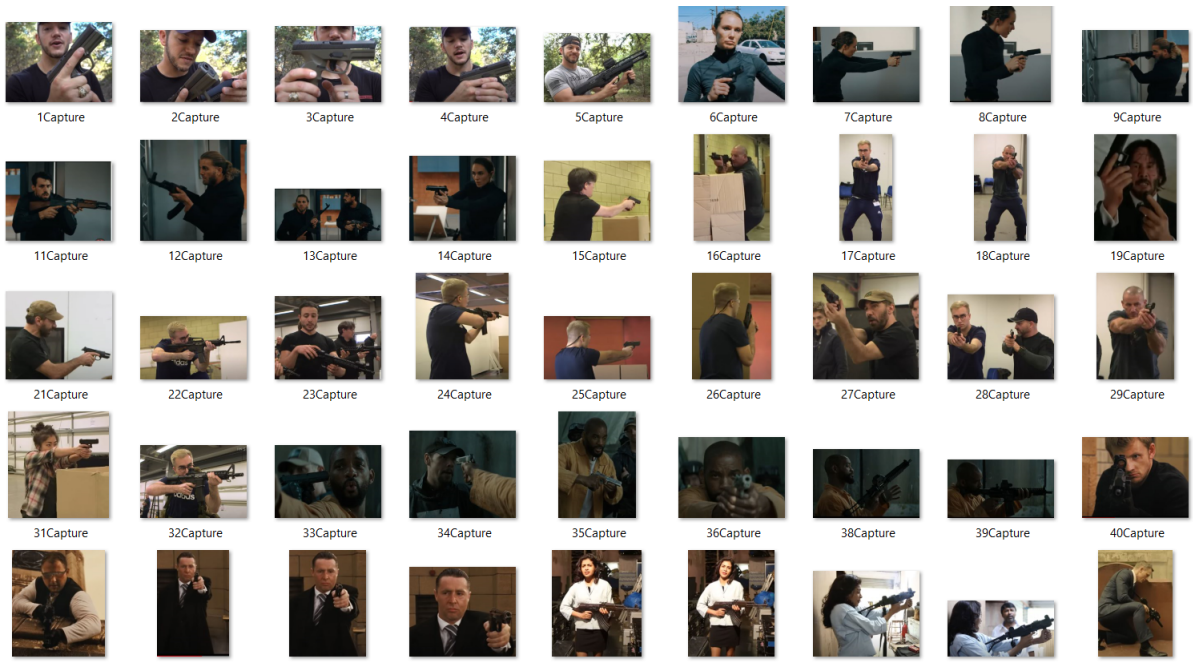


Figure 4.5: Criminal with knife

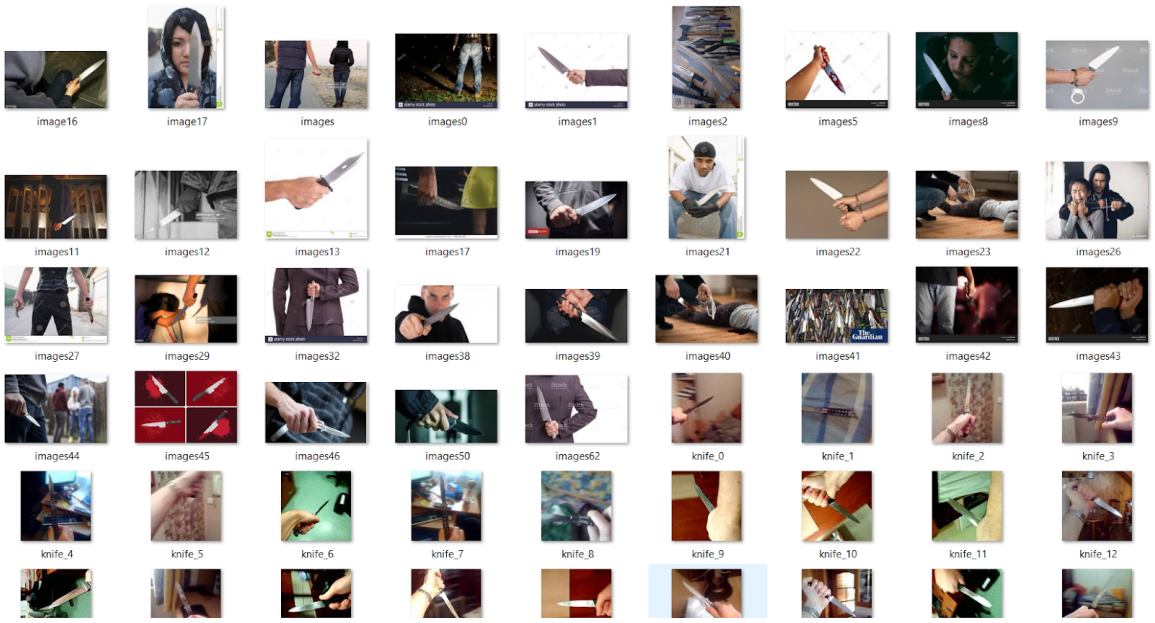


Figure 4.6: Criminal with gun

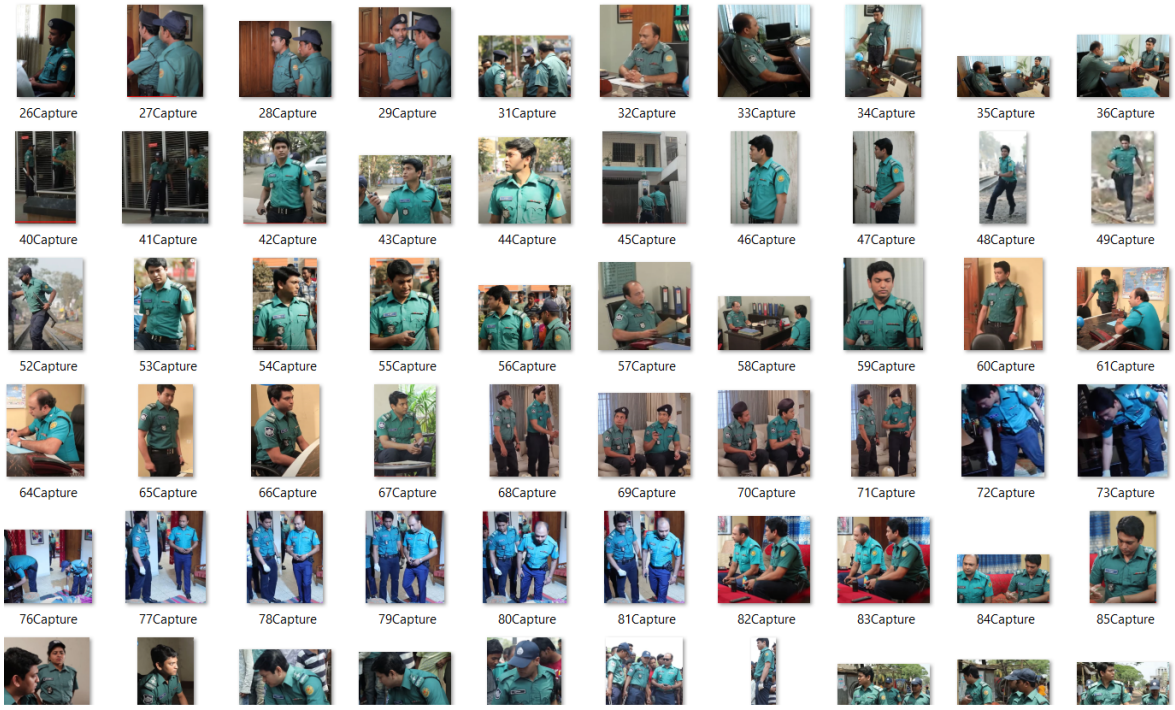


Figure 4.7: Bangladesh police

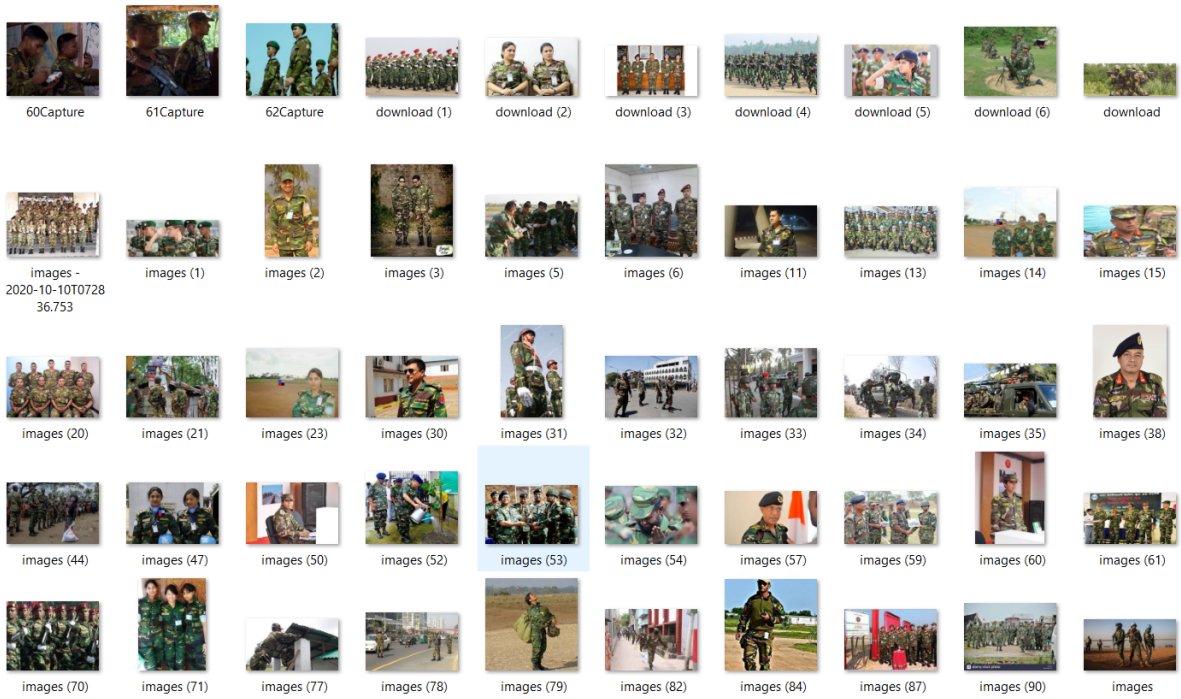


Figure 4.8: Bangladesh Army

4.3 Data Processing

In the area of deep learning, image classification and recognition is a rapidly evolving field where object recognition plays a crucial role. When we use data planning and data augmentation for our image datasets, Keras image processing evaluates and constructs deep learning models. This generator of images provides lots of image data with an increase of real data [17]. In our image dataset, images of various resolution pixels in grayscale format are used. In addition, we used the standard input required for the pre-trained model of CNN in keras which is a resolution of 224×224 pixels 8 bit RGB format. Our dataset consists of about 4180 images of criminal activity divided into four classes. We have used 80% of the images for training and the rest 20% for testing our model. Then the 80% of our training images were again splitted into 25% for validation images and 75% for training images. Our dataset consisted of images of different sizes so our first task was to convert the images into size 224 × 224 according to the CNN models requirements. Figure 4.9 shows the original size of a sample image of our dataset. Figure 4.10 shows the resized image after processing was completed.

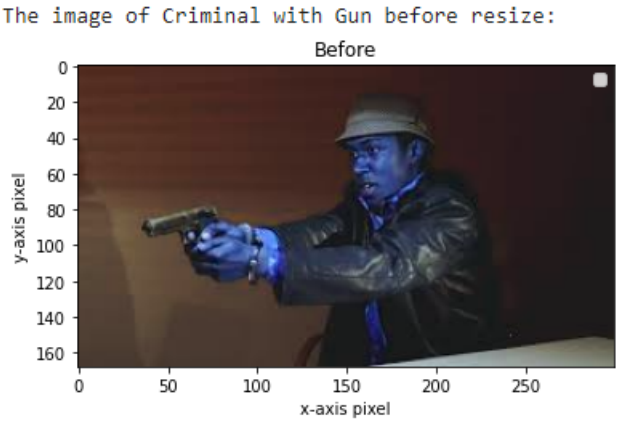


Figure 4.9: Default Image Size Before Processing

From the figure 4.9 we can see that the image size is around 300×160 and just like this image, all the images are of different shapes so we resized them as the models we have used require 224×224 image size.
Figure 4.10 shows the resized image after processing was completed. Here we can see that the image is resized to 224×224.

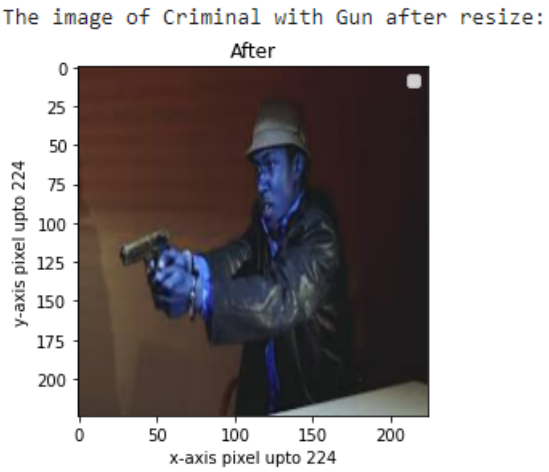


Figure 4.10: Image Size After Processing

4.4 Libraries

Libraries are essential tools in image classification and processing. The most important libraries used for our research are Keras, Tensorflow and OpenCV which are explained in the following sections.

4.4.1 Keras

It is a high level Python based neural network API and its biggest advantage is that it can run on top of TensorFlow and other deep learning libraries. It is not limited to use CPU power only but also uses Nvidia GPUs together with the CPU to provide maximum throughput [25]. Keras Applications supports ten renowned pre-trained models against ImageNet. It can be used to forecast the classification of images, derive features from them, and fine-tune a separate set of classes for the models. Thus, we have used this library along with TensorFlow to create, train and evaluate the models.

4.4.2 Tensorflow

TensorFlow is an artificial intelligence library that is written in Python, C++ and CUDA programming language which is also the most popular open-source library. It can operate in large-scale environments and use data flow diagrams to train models.TensorFlow is programmed to use different software components for the backend (GPUs, ASIC), etc. It utilizes Python to provide a simple front-end API for the application to create apps when running high-performance apps [7]. It has been used to create the models as well as to evaluate the accuracy and loss graphs of the four models.

4.4.3 OpenCV

It is a Python binding library intended to address computer vision tasks such as image and video processing. It is faster and requires low RAM usage. OpenCV is supported by all operating systems and is portable. It has several modules that we can use, such as Core features, it is a module that determines the basic structure of data, such as dense multi-dimensional Mat array and main functions utilized by some of the other available modules [36]. Linear and non-linear image filtering, geometric image modifications (resize, affine and perspective warping, generic table-based restoration), color space transformation, histograms are some of the modules of OpenCV [36]. The image processing module was necessary since we mainly deal with images and use OpenCV to process them. We have used this library to read and write the pixel values of the images.

Chapter 5

Methodology

In this paper, several steps were performed for accomplishing better accuracy. In the following sections, we have described each step sequentially. A brief overview of the models are also provided in separate sections for each of the models.

5.1 Workflow

First of all, we have collected the data set then the next step is to divide our data for training and testing. Our train data contains 3640 images of 4 categories and the test data contains 540 images of 4 categories. After training and validating it will generate data from images and on the other side, the test data will also generate data from the test images. Finally, the generated data and the pre-trained models we are using will go to the prepared model where it will compare the trained data with the test data and will give the accuracy based on the analysis.

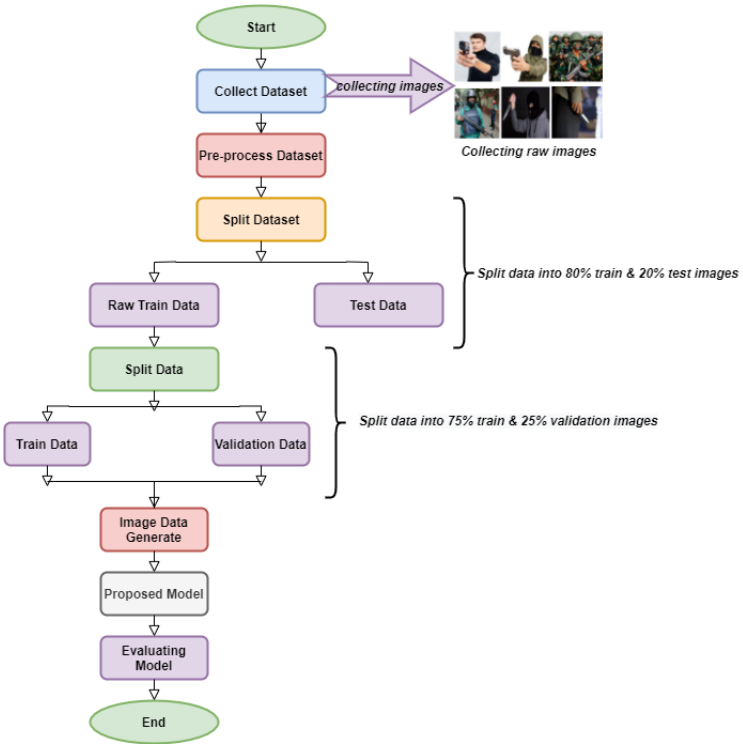


Figure 5.1: Workflow Diagram

5.2 Transfer Learning

Transfer learning is a machine learning approach in which a model built for a task is replicated as a starting place for a second related task. Due to the immense computing and time consuming resources required to train a neural network model from scratch, pre-trained models are becoming increasingly popular and are commonly used as the initial point in computer vision and natural language processing nowadays. It only retrains the later layers which helps utilize the labelled data of the task it was initially trained on. Thus, it is particularly useful in these fields to solve real-world problems which usually do not have millions of data points needed to train the complex models.

5.2.1 VGG16

VGG was created by Karen Simonyan and Andrew Zisserman and named after their lab the Visual Geometry Group at Oxford. The use of multiple convolutional layers before a max pooling layer is what separates it from the previous imagenet models. The vast number of filters used is another significant distinction.The number of filters starts at 64 and gradually rises to 128, 256, and 512 filters after the completion of the feature extraction process of the model [3]. A wide range of architecture versions have been designed and tested, but two are most frequently referred to due to their efficiency and depth. They are VGG-16 and VGG-19 named according to the number of layers used in their respective architectures. VGG-16 is a simplified architecture model since it doesn't use a lot of hyper parameters. It uses 3×3 filters with stride of 1 in the convolution layer and uses a similar padding with a stride of 2 in the 2×2 pooling layers [30]. VGGNet-16 consists of 16 convolution layers and is quite desirable mostly due to its consistent architecture. Compared to AlexNet, it has just 3×3 convolutions, but a ton of filters. It can also be trained on 4 GPUs for two or three weeks [30]. It is currently the most common alternative for extracting features from images in the crowd. The weight configuration of the VGGNet is openly accessible and has been used as a standard feature extractor in several other application domains. VGGNet, however, is made up of 138 million parameters, which can be a little difficult to maneuver. Using transfer learning VGG can be obtained in which the model is pre-trained and the parameters are modified for greater precision on a dataset.

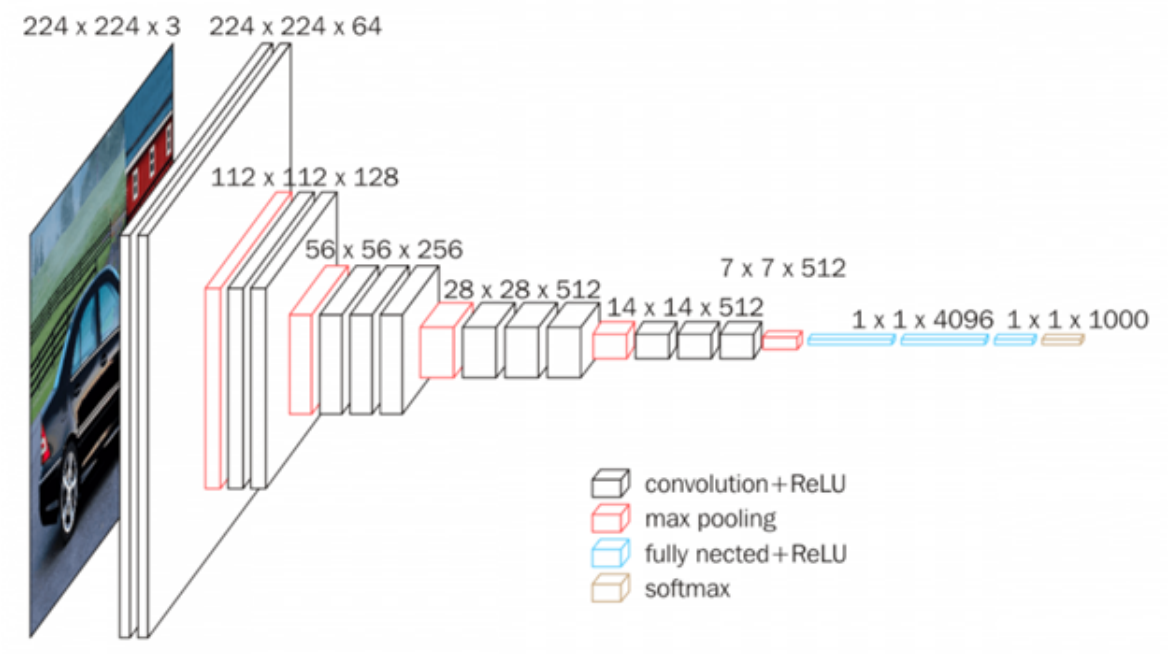


Figure 5.2: VGG16 Architecture [28]

5.2.2 Inception V3 Model

Inception-v3 belongs to the Inception family which is also a convolutional neural network architecture that allows many modifications, including the use of Label Smoothing, Factorized 7 × 7 convolutions, and the use of an auxiliary classifier to relay label information through the network. It can be the best architecture to be implemented into the devices with low processing units. It gives a good accuracy when the images are of low resolution [16].

Inception Networks (GoogLeNet/Inception v1) have proven to be more computationally efficient compared to VGGNet, both in terms of the number of network parameters produced and the financial cost incurred (memory and other resources) [32]. Therefore, due to the complexity of the new network's effectiveness, the adaptation of an Inception network for multiple use cases turns out to be a challenge. Factorized convolutions, regularization, dimension reduction and parallelized computations are some examples of the techniques used [32]. The model consists of two parts. In the first step, it extracts basic features from input data with a convolutional neural network and then classifies them based on the newfound features with fully-connected and softmax layers in the second part.

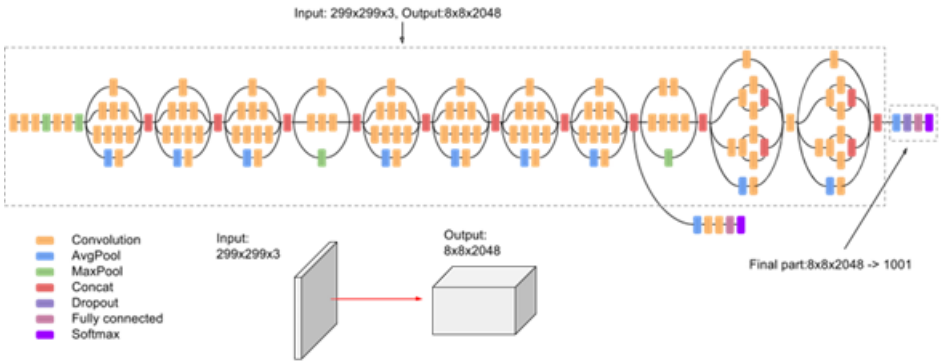


Figure 5.3: InceptionV3 Architecture [1]

5.2.3 ResNet50

Deep Residual Network (ResNet) is by far the most innovative work in the last few years in the area of computer vision / deep learning. The ResNet architecture makes it feasible to train ultra-deep neural networks with hundreds or thousands of layers and yet achieve high performance [13]. Initially, the ResNets were added to the image recognition task but also eventually earned recognition in non-computer vision tasks to achieve better accuracy.

There is a common misconception in the research community that stacking more layers in the network architecture can increase the accuracy of the model. However, as the network deepens, its output becomes saturated or even starts to degrade rapidly due to the infamous problem of vanishing gradients. This is where ResNet comes in handy. ResNet's central concept is to implement shortcut connections which simply carry out identity mapping that skips one or more layers. The authors of [13] explicitly let the layers fit a residual mapping and denoted that as H(x) and they let the nonlinear layers fit another mapping $F(x) := H(x) - x$ so the original mapping becomes $H(x) := F(x) + x$ as can be seen in Figure 5.4. The benefit of residual mapping is that the computation becomes easier and produces the same output without degrading the performance.

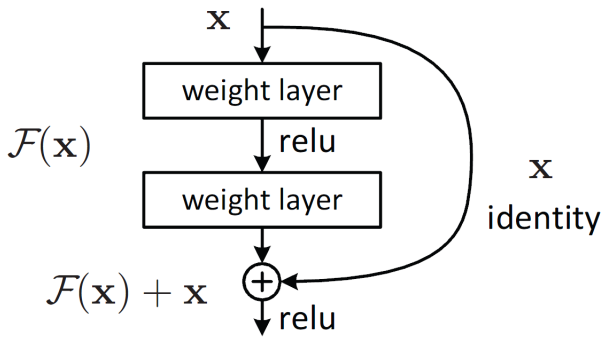


Figure 5.4: Residual Mapping [24]

Following ResNet's popularity, more variations of it were introduced of which ResNet 50 has the best overall performance. ResNet50 is slightly different from its predecessors. Unlike the previous versions which skipped two layers, it skips three layers and also added 1 x 1 convolution layers with the ResNet50 architecture. It consists of 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer and has $3.8 * 10^9$ floating points operations. As a result, ResNet50 has higher performance and the convergence speed and minimum value are better than its other variants.

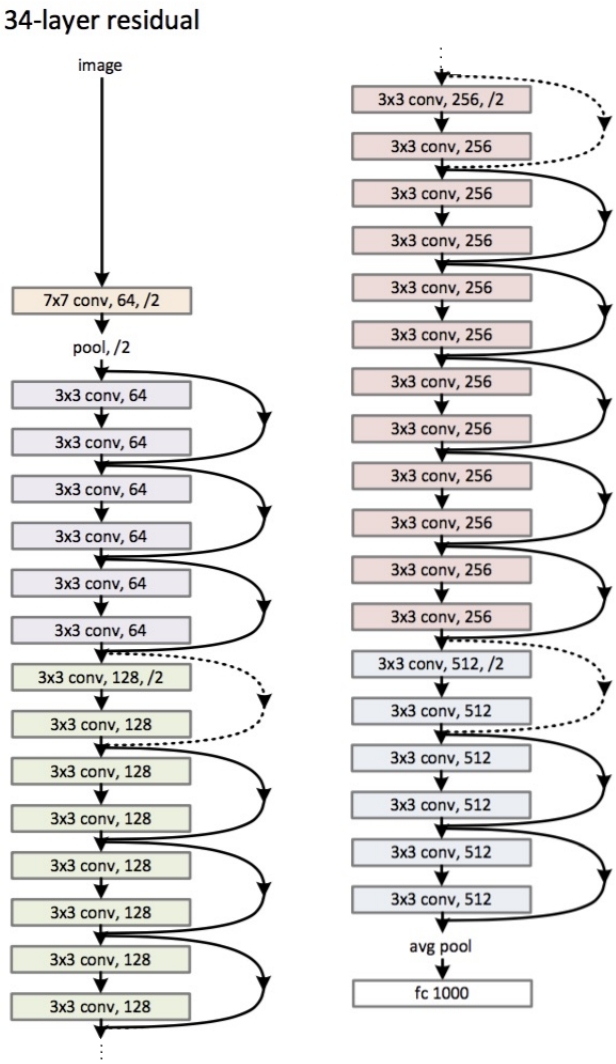


Figure 5.5: ResNet50 Architecture [24]

5.2.4 MobileNet V2

For TensorFlow, MobileNet was introduced as the family of the first portable computer vision models. It was designed to optimize precision efficiently while bearing in mind the minimal resources for an on-board or embedded device. The core concept behind MobileNet is that the convolutional layers, which are central to computer vision problems but are also very costly to replace, can be supplanted by depth wise distinct convolutions. The complete architecture of this model consists of a

standard 3×3 convolution as the first layer, accompanied by a 13-fold convolution of the building block above. MobileNet-v2 has a slight difference in the architecture. It consists of three convolutional layers in the architecture where the inputs are filtered through the last two depth wise convolution layers followed by a 1×1 point wise convolution layer. The entire MobileNet-v2 architecture is made up of 17 of these blocks placed consecutively. This is followed by a regular 1×1 convolution, a global average pooling layer, and a classification layer. The point-wise layer makes the number of channels smaller than that of the V1 which retains the same number of channels. The depth wise layer is also referred to as a bottleneck layer because it decreases the quantity of data that can move through the network. Another addition to MobileNet-v2 is the residual connection similar to that of ResNet which helps with the flow of gradients in the network. Thus, MobileNet-v2 is cheaper, faster and easier to compute than its predecessor MobileNet-v1.

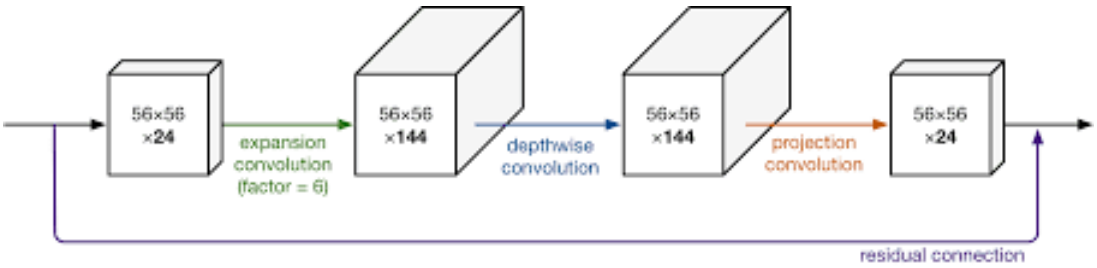


Figure 5.6: MobileNet V2 Architecture [18]

5.2.5 Xception Model

Xception represents the extreme version of Inceptionv3. This is superior to Inception-v3 with an updated depth wise, separable convolution. The modified depth wise separable convolution is the point wise convolution preceded by a wise profound convolution. This modification is inspired by the origin module in Inception-v3 that 1x1 convolution actually took place before any nxn field of spatial convolutions. Hence, it is somewhat different than the first one.

There are two minor differences between inceptionv3 and Xception. Initial depth wise separable convolutions execute channel-wise spatial convolution at first and then perform 1×1 convolution, while updated depth wise separable convolution executes 1×1 convolution first instead of channel-wise spatial convolution [15]. Inception Module is not linear after the first operation whereas Xception does not have any transitional ReLU nonlinearity. SeparableConv is the revised depth wise separable convolution. In Figure 5.7, we can see that SeparableConvs are identified as Inception Modules and are placed all over the entire deep learning architecture. It also consists of residual connections which were originally present in the ResNet architecture. The residual connection increases the accuracy of the Xception model so it is an integral part of the architecture.

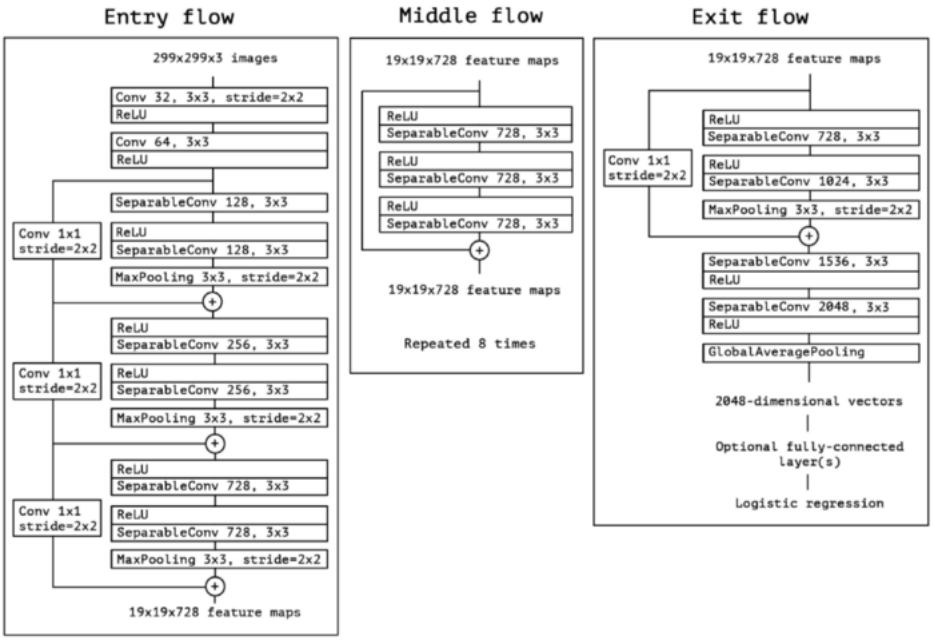


Figure 5.7: Xception Architecture [17]

5.3 Implementation

We have used our own dataset to train five pre-trained models VGG16, Inception-v3, MobileNet-v2, ResNet50 and Xception. These models were then evaluated based on their accuracies to find out the best model which can predict the criminal activity most precisely. The VGG16, Resnet50, Inception-v3, Xception, MobileNet-v2 are CNN models which are trained on 1000 different categories.

Firstly, we prepared our model then we fit our train data and validation data in our model. Here, we used 32 images as batch size per epoch with a specific number of epochs to observe the difference in the outputs. At each epoch the validation data evaluates the loss. We noticed that steps per epoch affects the accuracy rate. If we increase the batch size then the accuracy rate increases because of increased steps. As a result the model gets more data to train at a particular point fixed by the models will be concentrated. Here, we have used pre-trained models for comparison. So, using imagenet will help to take the weight from the pre-trained models that have already been found to work perfectly. After declaring the include top true, the fully connected layer that is at the top of the network was included. The fully connected network performs well when it comes to the entire image, but the specific image cannot be accurately classified in the case of a cropped image.

Rather than using these pre-trained models as feature extractors we fine-tuned the five models with image augmentation by training the last dense and softmax layer and leaving the other layers frozen. The softmax activation function assigns decimal probabilities to every possible class in a multi-class problem. Figure 5.8 shows the mentioned process of our research work implementation.

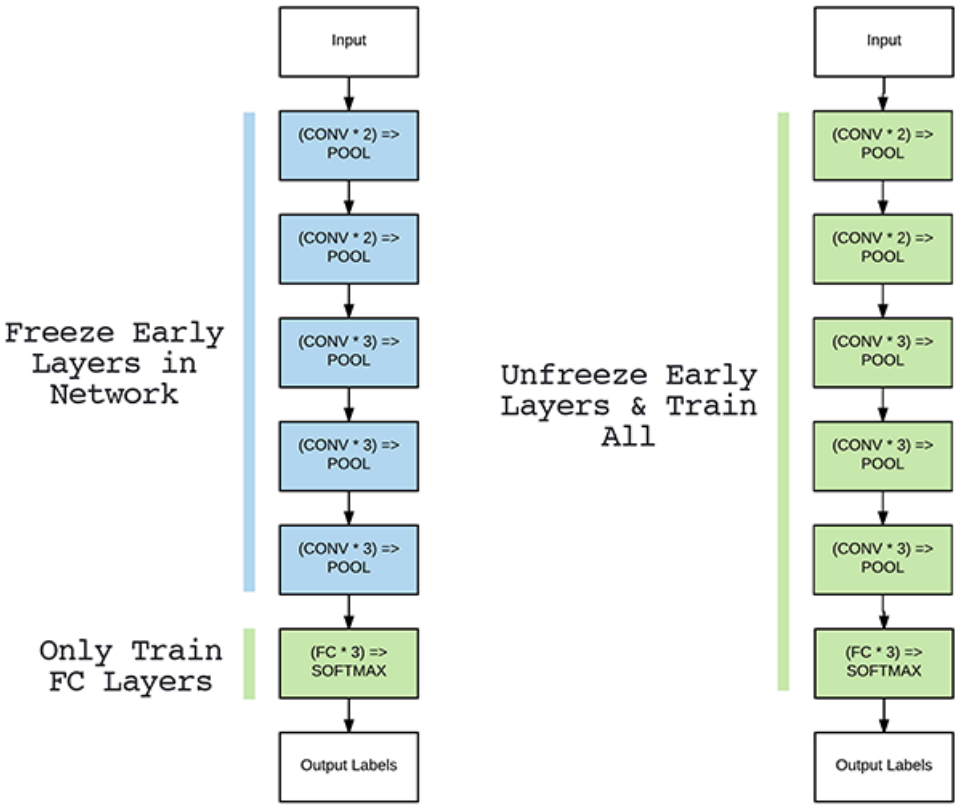


Figure 5.8: Proposed Method of Implementation

We have used Relu(Rectified Linear Unit) activation to each layer so that all the negative values are not passed to the next layer. We have used RELU activation for the dense layer so that we can stop forwarding negative values through the network. We have used ModelCheckpoint in order to save the model by tracking validation accuracy by forwarding valacc to ModelCheckpoint. If the validation accuracy in the current epoch is higher than in the previous epoch, the model will be saved. A 2D avgpool was used for our pooling layer. However, max pooling could have been used here but it only takes the maximum value from the matrix whereas average pooling takes the average of the matrix. Max pooling rejects a big chunk of data and it retains at max 1/4th whereas average pooling uses all of it and more information can be retained from average pooling compared to max pooling.

In addition, we have used EarlyStopping to stop the model training early when there is no improvement in the validation accuracy or the parameter. We set patience at 2, which means that if there is no increase in validation accuracy in 2 epochs, the model will stop learning. Model.fitgenerator has been used and by using ImageDataGenerator it passes data(it rescale, rotate, zoom, flip the images) to the desired model. We passed our train and test data to fitgenerator. A specific batch size has been used to pass the training data to the model. After training the model we visualised training and validation accuracy and loss using matplotlib. We are passing the output of mode.fitgenerator to r variable where accuracy and loss are stored. To predict on the trained model at first we loaded the saved model and pre-processed the image and passed the image to the model for output. We have loaded the image and converted it to a numpy array and resized it similar to the

input images. The one with the highest probability will be the output chosen from four classes epoch which improves the accuracy.

5.3.1 Inception-v3

The output layer of Inception-v3 network contains 1,000 classes whereas we have only 4 classes and those are 'Army', 'Police', 'Criminal with Guns', 'Criminal with knife'. We have trained the last fully connected/dense layer and the softmax layer of the model and frozen the rest of the layers. The Relu function was used in the last fully connected layer and it had 8196 nodes which was followed by a softmax classifier. The softmax layer has output channels from 1,000 which we converted into four classes. The prediction is made by a softmax classifier. The final classifier has 1000 neurons but since we have four classes so we are just training the two last layers. By using the last two layers only, we have reduced the number of trainable parameters significantly. Trainable parameters before freezing were 23,817,352 which dropped to just 8196 after freezing. We have used adam as the optimizer algorithm to change the attributes of the neural network such as weights and learning rate to reduce the losses. In addition, categorical cross entropy was used since we have four classes and also to train the model to output a probability over the four classes for each image. The softmax layer will output the probability of each class and the class with the highest probability will define which class the images belong to. Figure 5.9 shows the correct prediction of criminal with knife by the Inception-v3 model.

Figure 5.9: Successful detection of Criminal with Knife Using Inception-v3

5.3.2 VGG16

The output layer of VGG16 network contains 1,000 classes whereas we have only 4 classes and those are 'Army', 'Police', 'Criminal with Guns', 'Criminal with knife'. We have trained the last fully connected/dense layer and the softmax layer of the model and frozen the rest of the layers. The Relu function was used in the last fully connected layer and it had 4096 nodes which was followed by a softmax classifier. The softmax layer has output channels from 1,000 which we converted into 4 classes. The prediction is made by a softmax classifier. The final classifier has 1000 neurons but since we have four classes so we are just training the two last layers. By using the last two layers only, we have reduced the number of trainable parameters significantly from 138,357,544 to just 16,797,700. We have used adadelta as the optimizer algorithm to change the attributes of the neural network such as weights and learning rate to reduce the losses. In addition, categorical cross entropy was used since we have four classes and also to train the model to output a probability over the four classes for each image. The softmax layer will output the probability of each class and the class with the highest probability will define which class the images belong to. Figure 5.10 shows the correct prediction of criminal with gun by the VGG16 model.

Figure 5.10: Successful Detection of Criminal with Gun Using VGG16

5.3.3 MobileNet V2

The output layer of MobileNer-v2 network contains 1,000 classes whereas we have only 4 classes and those are 'Army', 'Police', 'Criminal with Guns', 'Criminal with knife'. We have trained the last fully connected/dense layer and the softmax layer of the model and frozen the rest of the layers. The Relu function was used in the last fully connected layer and it had 1280 nodes which was followed by a softmax classifier. The last layer has output channels from 1,000 which we converted into

4 classes. The prediction is made by a softmax classifier. The final classifier has 1000 neurons but since we have four classes so we are just training the two last layers. By using the last two layers only, we have reduced the number of trainable parameters significantly from 1,281,000 to just 5124. We have used RMSprop as the optimizer algorithm to change the attributes of the neural network such as weights and learning rate to reduce the losses. In addition, categorical cross entropy was used since we have four classes and also to train the model to output a probability over the four classes for each image. The softmax layer will output the probability of each class and the class with the highest probability will define which class the images belong to. Figure 5.11 shows the correct prediction of Bangladesh Police by the MobileNet-v2 model [26].

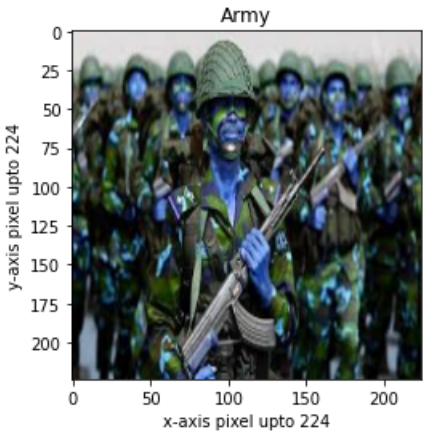
Figure 5.11: Successful Detection of Bangladesh Police Using MobileNet-v2

5.3.4 Xception

The output layer of the Xception network contains 1,000 classes whereas we have only 4 classes and those are 'Army', 'Police', 'Criminal with Guns', 'Criminal with knife'. We have trained the last fully connected/dense layer and the softmax layer of the model and frozen the rest of the layers. The Relu function was used in the last fully connected layer and it had 8196 nodes which was followed by a softmax classifier. The last layer has output channels from 1,000 which we converted into 4 classes. The prediction is made by a softmax classifier. The final classifier has 1000 neurons but since we have four classes so we are just training the two last layers. By using the last two layers only, we have reduced the number of trainable parameters significantly from 2,049,000 to just 8196. We have used adam as the optimizer algorithm to change the attributes of the neural network such as weights and learning rate to reduce the losses. In addition, categorical cross entropy was used since we have four classes and also to train the model to output a probability over the four classes for each image. The softmax layer will output the probability

of each class and the class with the highest probability will define which class the images belong to. Figure 5.12 shows the correct prediction of Bangladesh army by the Xception model.

```
Predictions:
Accuray of input image [[9.5976728e-01 9.9433094e-20 4.0232714e-02 2.5131112e-14]]
0
```




```
print(
    "This image most likely belongs to {} with a {:.2f} percent accuracy."
    .format(Categories[np.argmax(prediction)], 100 * np.max(prediction)))
```

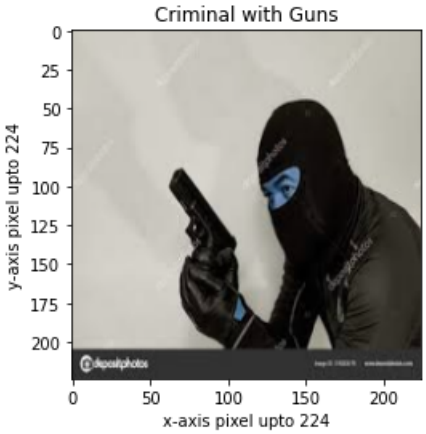
```
This image most likely belongs to Army with a 95.98 percent accuracy.
```

Figure 5.12: Successful Detection of Bangladesh Army Using Xception

5.3.5 ResNet50

The output layer of the ResNet50 network contains 1,000 classes whereas we have only 4 classes and those are 'Army', 'Police', 'Criminal with Guns', 'Criminal with knife'. We have trained the last fully connected/dense layer and the softmax layer of the model and frozen the rest of the layers. The Relu function was used in the last fully connected layer and it had 2048 nodes which was followed by a softmax classifier. The last layer has output channels from 1,000 which we converted into 4 classes. The prediction is made by a softmax classifier. The final classifier has 1000 neurons but since we have four classes so we are just training the two last layers. By using the last two layers only, we have reduced the number of trainable parameters significantly from 2,049,000 to just 8196. We have used RMSprop as the optimizer algorithm to change the attributes of the neural network such as weights and learning rate to reduce the losses. In addition, categorical cross entropy was used since we have four classes and also to train the model to output a probability over the four classes for each image. The softmax layer will output the probability of each class and the class with the highest probability will define which class the images belong to. Figure 5.13 shows the correct prediction of criminal with gun by the ResNet50 model.

```
Predictions:
Accuray of input image [[8.3371199e-24 9.9974316e-01 2.5685725e-04 3.3988840e-10]]
1
```




```
print(
    "This image most likely belongs to {} with a {:.2f} percent accuracy."
    .format(Categories[np.argmax(prediction)], 100 * np.max(prediction)))
```

Figure 5.13: Successful Detection of Criminal with Gun Using ResNet50

From the above implementation process discussed and the images of our successful predictions it can be said that in our paper our proposed models have achieved quite satisfactory results. The next chapter is going to further explain the accuracy and loss curves and detailed analysis of the five models.

Chapter 6

Result and Analysis

6.1 Result

To restate, we have trained the five models with our own dataset consisting of 4,180 images where Test data included 540 images belonging to 4 classes and Train data included 3640 images belonging to 4 classes.

Name Of Models	Accuracy	Validation Accuracy	Loss	Validation Loss
MobileNet-v2	98%	91%	5%	25%
Inception-v3	98%	95%	5%	16%
Xception	94%	84%	15%	44%
VGG-16	70%	67%	76%	85%
ResNet50	60%	67%	98%	85%

Table 6.1: Accuracy and Loss Distribution of the Models

From the table we can see that the best accuracy was achieved by the MobileNet-v2 and Inception-v3 which is 98% and the reason for acquiring this high accuracy is the number of parameters that we have used and the dimension of the nodes of the layer.The worst accuracy was achieved by the ResNet50 model which is 60%. The best and worst validation accuracy were also gained by Inception-v3, MobileNet-v2 and ResNet50 respectively. Before the data is fed as input, accuracy on both of the training set and test set should be performed in order to attain the desired accuracy, which would decide whether the dataset is vulnerable to overfitting or underfitting. The model has an overfitting problem if the accuracy of the training sample is greater than that of the test set.

We have also plotted the graphs of the training loss vs. validation loss and training accuracy vs. validation accuracy against the number of epochs. Train and Validation of loss and accuracy for the five models were trained on our own dataset.

From figure 6.1 , we can see that VGG16 achieved an accuracy of 70% which is quite satisfactory. The model did not overfit as the validation accuracy is greater than the training accuracy.

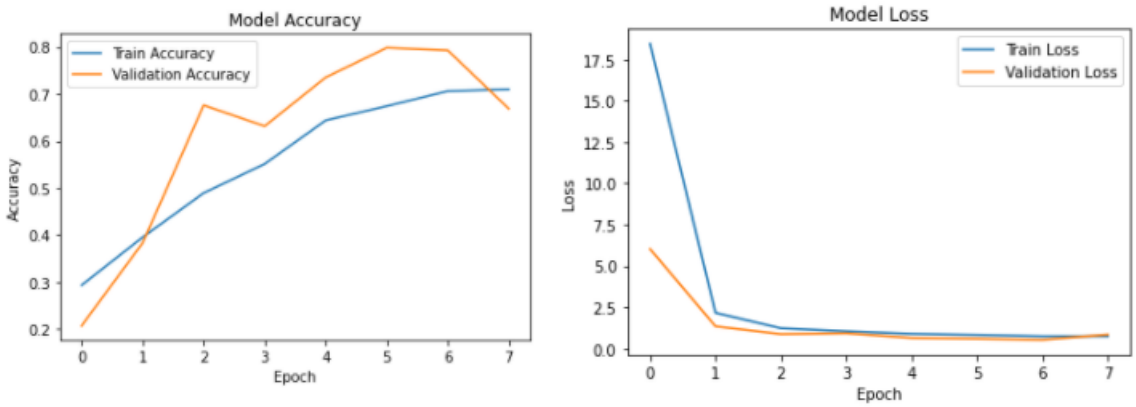


Figure 6.1: VGG16 Accuracy and Loss

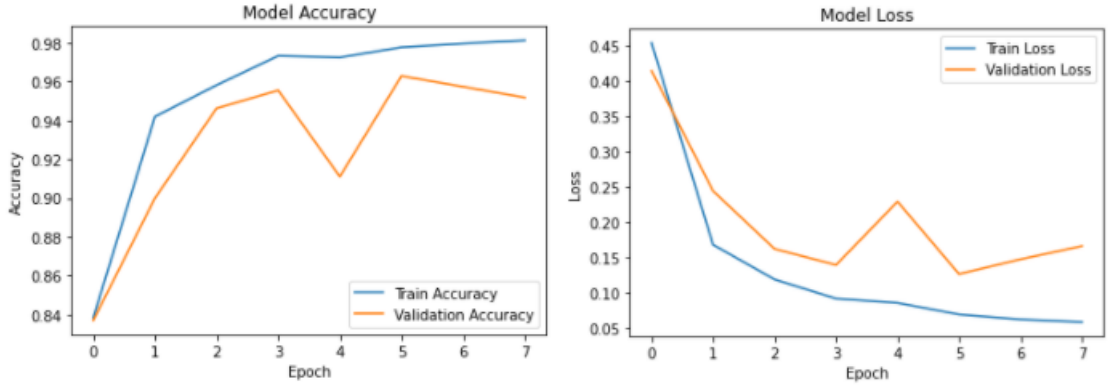


Figure 6.2: InceptionV3 Accuracy and Loss

It can be seen in figure 6.2 that the model begins to overfit on the training data after two to three epochs. The accuracy is around 98%. The reason behind this overfitting is due to the limited training data so the model tends to see the same cases over time across each epoch. To solve this problem, we plan to use an image augmentation technique in future to increase our current training data with images that have small modifications compared to the existing images.

It can be seen from figure 6.3 and 6.4, the approximate accuracy for MobileNet-v2 and Xception is around 98% and 94% which is remarkable but the models begin to overfit on the training data after two to three epochs. The reason behind this overfitting is due to the limited training data so the model tends to see the same cases over time across each epoch. We hope to eliminate this problem in our future research work by using an image augmentation technique to increase our current training data with images that have small modifications compared to the existing images.

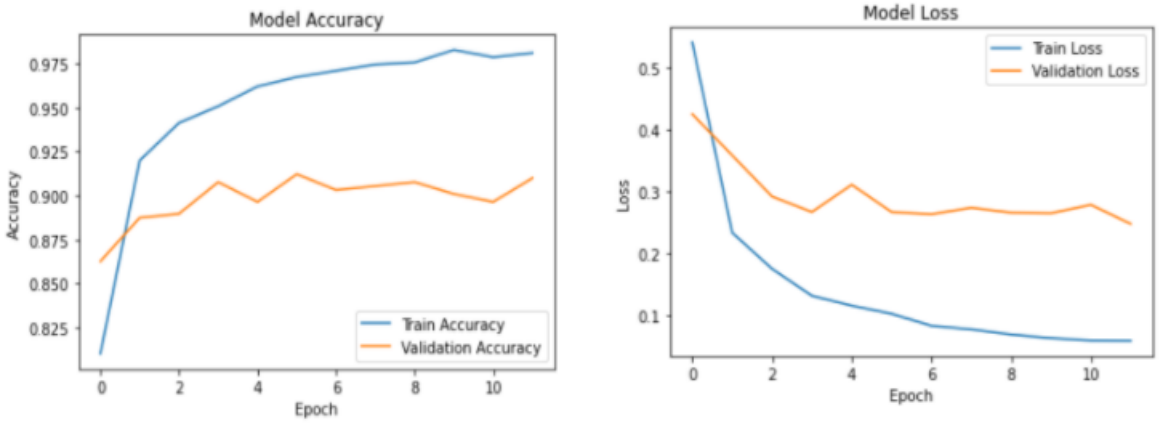


Figure 6.3: MobileNetV2 Accuracy and Loss

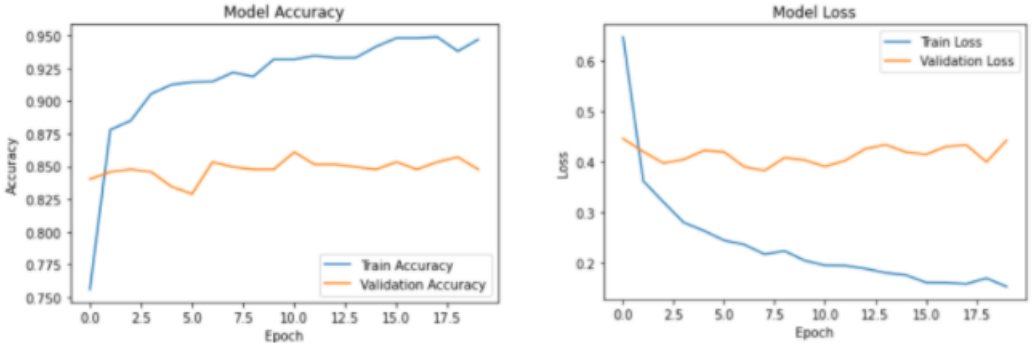


Figure 6.4: Xception Accuracy and Loss

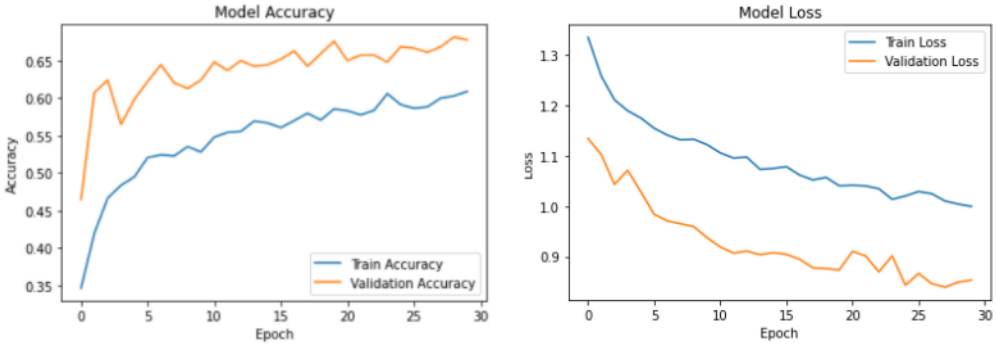


Figure 6.5: ResNet50 Accuracy and Loss

It can be seen from figure 6.5 that the accuracy is 60%. However, there is a fluctuation in the graph. One of the possible reasons can be due to the high variance of our existing dataset. The dataset used for training is not the optimal representation of the entire dataset since there are not enough examples of criminal photos of the same class. We believe that if the images are improved, the shifts and fluctuations will decrease.

The proposed models have been successful in predicting the four classes in most cases but it is still not satisfactory as the models have also predicted false positives. False positives occur when the model identifies a data into a class which is not true in reality. Figure 6.6 and 6.7 show images of a Bangladesh police and criminal with knife respectively but the model is predicting both images as a criminal with gun which means these are false positives. This happened since our images of all the classes are quite similar as all the classes have weapons. Thus, the models can get confused and give wrong predictions.

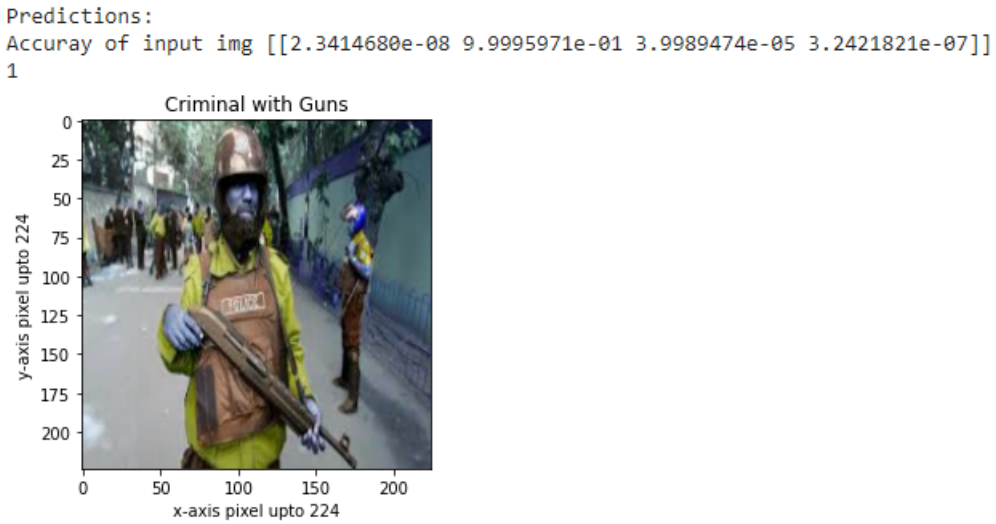


Figure 6.6: False Positive of Criminal with Guns

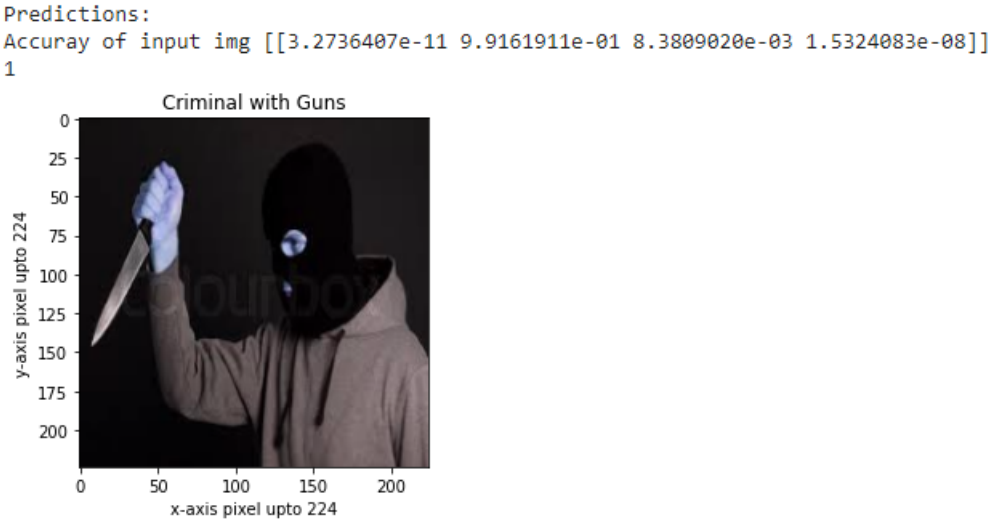


Figure 6.7: False Positive of Criminal with Guns

6.2 Analysis

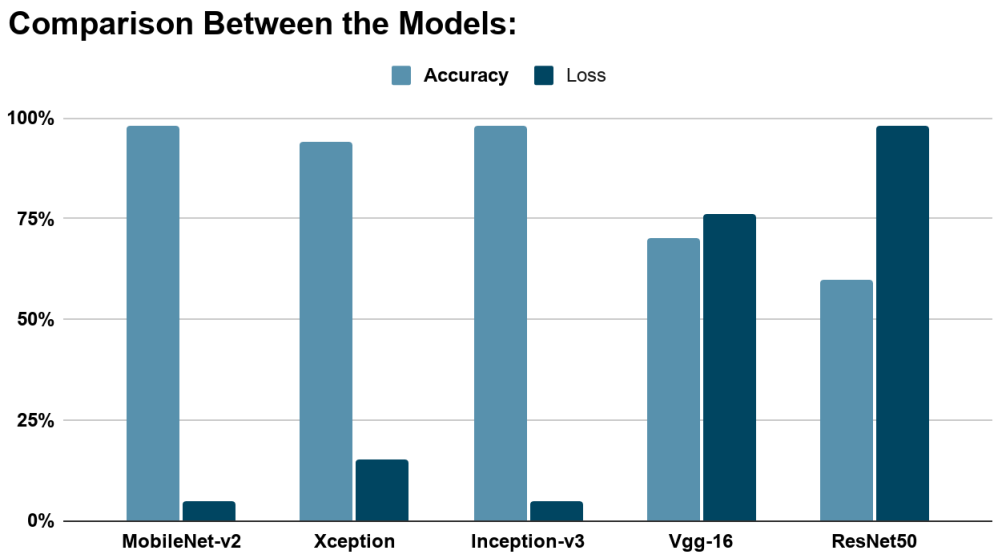


Figure 6.8: Comparison Between the Models

From figure 6.8, it can be seen MobileNet-v2 and Inception-v3 gave the best accuracy among the five models. It has an accuracy of 98% and loss of 5% for both the models. These have performed very well than the other models because of less parameters and its high non trainable parameters. The reason behind this is its depth wise convolution layers and also the pooling layer shape and the convolution shape are similar which has caused less data loss. Inception-v3 outperformed VGG-16 since it has less parameters than VGG-16 which made the computation more easier and gave more accuracy. Xception's accuracy is also very close to Inception and MobileNet which is 94%. ResNet50 has achieved the lowest accuracy than the rest of the models which is 60%. This model has less convolutional parameters than the others which results in huge value loss which affects the accuracy rate. We found that the accuracy rises when the model has more parameters.

Chapter 7

Conclusion

7.1 Conclusion

With the rapid growth of smart cities, the necessity to build a better security system which does not involve any human intervention has become the first priority for every work field and residence. Security systems managed by human officials entirely may cause human error at a high rate which is never expected. That's why our research is about building such a security system that does not need any human intervention. We built our own image datasets that include four classes which are criminal with gun, criminal with knife, police and army. These four classes of images are given as inputs to train five different types of CNN models which are, VGG16, ResNet50, MobileNetv2, Xception and Inceptionv3. The accuracy in MobileNetV2 and Inception-v3 was the highest which is 98%.

7.2 Future Work

This Research work done by us, is entirely based on image datasets where we used images of criminals holding harmful weapons and trained our model to identify the criminal through detecting the weapons the criminal is holding in the image. However, from our future planning point of view, our goal is to establish a security system for crime prone places, where CCTV cameras will be monitored without any human intervention to avoid human errors. For that we need to build raw CCTV footage datasets which will contain videos of previous crime scenes captured in various CCTV cameras from public places or residences. Those videos will be used to train our future proposed model. The whole implementation process of our current research work is entirely based on a supervised learning approach. Since our work will be progressing towards video datasets where continuous video footage from CCTV cameras will be fed into the model, we need to switch to an unsupervised learning approach. In such cases the system updates dynamically once its training stage starts and can generate new data based on the previous training sets. While working with video footage there can be many possible situations in the video which might not be trained in our model. So, for such situations, we need to implement unsupervised learning algorithms which will work for many possible scenarios so that whenever a criminal appears in the footage an alarm is generated and sent to the nearby police station or office authority so that any criminal activity can be prevented and the criminal gets caught. Furthermore, our plan is to push our model

even further and create a better Deep Learning Hybrid algorithm where the human behavior and pattern of the criminal will be recorded so that even if for the first few times the model fails to recognize the criminal but due to the pattern memorization system, in next occurrences it can recognize the criminal very well and alert the authority. This will also be very useful for cases where criminals cover their entire face and are extremely clever that they run away easily after committing a crime. In such cases, pattern recognition might help to recognize the criminal when appeared the next time in front of the CCTV camera.

Bibliography

[1] W. J. Schroeder, L. S. Avila, and W. Hoffman, "Visualizing with vtk: A tutorial," *IEEE Computer graphics and applications*, vol. 20, no. 5, pp. 20–27, 2000.

[2] L. Davis, "Real time computer surveillance for crime detection," Tech. Rep. 192734, US Department of Justice, Tech. Rep., 2002.

[3] U. Akdemir, P. Turaga, and R. Chellappa, "An ontology based approach for activity recognition from video," in *Proceedings of the 16th ACM international conference on Multimedia*, 2008, pp. 709–712.

[4] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[5] E. B. Nievas, O. D. Suarez, G. B. Garcıa, and R. Sukthankar, "Violence detection in video using computer vision techniques," in *International conference on Computer analysis of images and patterns*, Springer, 2011, pp. 332–339.

[6] G. E. Hinton, A. Krizhevsky, and I. Sutskever, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1106–1114, 2012.

[7] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[9] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in neural information processing systems*, vol. 27, pp. 568–576, 2014.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[11] A. Taha, H. Zayed, M. Khalifa, and E.-S. El-Horbarty, "Human activity recognition for surveillance applications," May 2015. DOI: 10.15849/icit.2015.0103.

[12] M. Grega, A. Matiolański, P. Guzik, and M. Leszczuk, "Automated detection of firearms and knives in a cctv image," *Sensors*, vol. 16, no. 1, p. 47, 2016.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[14] S. Mohammadi, A. Perina, H. Kiani, and V. Murino, "Angry crowds: Detecting violent events in videos," in *European Conference on Computer Vision*, Springer, 2016, pp. 3–18.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. DOI: 10.1109/cvpr.2016.308.

[16] ——, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[17] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[19] S. B. Kibria and M. S. Hasan, "An analysis of feature extraction and classification algorithms for dangerous object detection," in *2017 2nd International Conference on Electrical Electronic Engineering (ICEEE)*, 2017, pp. 1–4. DOI: 10.1109/CEEE.2017.8412846.

[20] V. Lam, S. Phan, D.-D. Le, D. A. Duong, and S. Satoh, "Evaluation of multiple features for violent scenes detection," *Multimedia Tools and Applications*, vol. 76, no. 5, pp. 7041–7065, 2017.

[21] G. K. Verma and A. Dhillon, "A handheld gun detection using faster r-cnn deep learning," in *Proceedings of the 7th International Conference on Computer and Communication Technology*, 2017, pp. 84–88.

[22] S. Chackravarthy, S. Schmitt, and L. Yang, "Intelligent crime anomaly detection in smart cities using deep learning," in *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, 2018, pp. 399–404. DOI: 10.1109/CIC.2018.00060.

[23] U. V. Navalgund and P. K., "Crime intention detection system using deep learning," in *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, 2018, pp. 1–6. DOI: 10.1109/ICCSDET.2018.8821168.

[24] F. Palsson, J. R. Sveinsson, and M. O. Ulfarsson, "Sentinel-2 image fusion using a deep residual network," *Remote Sensing*, vol. 10, no. 8, p. 1290, 2018.

[25] S. Yegulalp, "What is tensorflow? the machine learning library explained," *Infoworld. June*, vol. 6, 2018.

[26] S. Das, *Cnn architectures: Lenet, alexnet, vgg, googlenet, resnet and more.... medium, november 2017*, 2019.

[27] *Deep learning course with tensorflow: Ai deep learning training: Edureka*, 2019. [Online]. Available: https://www.edureka.co/ai-deep-learning-with-tensorflow.

[28] R. Thakur, "Step by step vgg16 implementation in keras for beginners," *Medium*, 2019.

[29] X. Zhang and W. Dahu, "Application of artificial intelligence algorithms in image processing," *Journal of Visual Communication and Image Representation*, vol. 61, pp. 42–49, 2019.

[30] A. K. Dash, *Vgg16 architecture*, Nov. 2020. [Online]. Available: https://iq.opengenus.org/vgg16/.

[31] J. L. S. González, C. Zaccaro, J. A. Álvarez-Garcıa, L. M. S. Morillo, and F. S. Caparrini, "Real-time gun detection in cctv: An open problem," *Neural networks*, vol. 132, pp. 297–308, 2020.

[32] V. Kurama, "A review of popular deep learning architectures: Resnet, inceptionv3, and squeezenet," *Consulted on August*, vol. 30, 2020.

[33] S. User, *How to implement artificial intelligence for solving image processing tasks*, Nov. 2020. [Online]. Available: https://www.apriorit.com/dev-blog/599-ai-for-image-processing.

[34] A. Warsi, M. Abdullah, M. N. Husen, and M. Yahya, "Automatic handgun and knife detection algorithms: A review," in *2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2020, pp. 1–9. DOI: 10.1109/IMCOM48794.2020.9001725.

[35] S. Chowdhury and P. Sinha, "Real time object detection using deep learning: A webcam based approach,"

[36] *Introduction*. [Online]. Available: https://docs.opencv.org/master/d1/dfb/intro.html.

[37] T. D. Piyadasa, "Concealed weapon detection using convolutional neural networks,"