Automatic Helmet-less Biker Detection Using Deep Learning

by

Md. Ibrahim Ratul
18301113
Abdul Karim Ibne Mohon
18301152
Md. Reduan Sarker
18301088

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
School of Data and Sciences
Brac University
September 2023

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing our 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:

Ibrahim Ratul

Md. Ibrahim Ratul

18301113

Karim Ibne Mohon

Abdul Karim Ibne Mohon

18301152

Md. Reduan Sarker

Md. Reduan Sarker

18301088

Approval

The thesis/project titled "Automatic Helmet-less Biker Detection Using Deep Learning" submitted by

1. Md. Ibrahim Ratul (18301113)
2. Abdul Karim Ibne Mohon(18301152)
3. Md. Reduan Sarker (18301088)

Of Summer 2023 has been accepted as satisfactory in partial fulfilment of the requirement for the degree of B.Sc. in Computer Science and Engineering on September 17, 2023.

Examining Committee:

Supervisor:
(Member)

AAR 18/9/23

Annajiat Alim Rasel

Senior Lecturer
CSE Department
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
CSE Department
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi

Chairperson
CSE Department
Brac University

Abstract

When riders don't wear a helmet while driving a motorcycle, they aren't paying attention, leading to crashes and more deaths. The researchers utilized multiple deep-learning models to identify motorcyclists without helmets. We have to identify correctly if a rider is wearing a helmet as we want to reduce risks regarding not wearing a helmet. Our paper proposes a convolutional neural network-based way to decide whether the rider carries a helmet. We utilized pre-trained deep learning models to forecast the outcome because we used a customized dataset. These models include EfficientNetB0, Inception, ResNet50, VGG16, and VGG19, and the results are satisfactory. Later, we put forth our model, combining the CNN, LSTM, and attention models. Our fusion model's foundation is a Dialated CNN layer. Our dilated CNN layer comprises three maximum pool layers and five convolutional layers. LSTM and an attention model layer follow convolutional layers on a five-layer CNN. Additionally, we used the same model to predict three classes on a separate dataset, and both models produced satisfactory outcomes. Our goal is to make greater use of the deep learning technique so that it can detect with incredible speed and precision. The test results indicate that, with a classification accuracy of 92.41%, our proposed method outperforms the alternatives we used. We have used YOLOV8 to detect riders wearing non-helmet headwear, such as caps, hijabs, and turbans, and have classified them as non-helmet wearers with satisfactory results.

Keywords: Helmet, CNN, LSTM, ATTENTION, Dialated CNN, YOLOV8

Acknowledgement

At the outset, it is essential to acknowledge that all credit is attributed to the divine entity, Almighty Allah, whose blessings have been instrumental in facilitating the smooth completion of our task with minimal difficulties. Secondly, we are so grateful to our supervisor, Annajiat Alim Rasel, for allowing us to work for him and his constant support. He gave us fruitful advice and suggestions regarding our research. We would also like to extend our utmost appreciation to ASP Ziaul Haque, DB, for his essential support in authorizing the safety assessments of the helmet, thereby significantly augmenting the efficacy of our research endeavour. Lastly, we want to thank our parents for their constant encouragement, enabling us to reach this point. Their prayers and inspiration have played a significant role in leading us towards our imminent 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

ANN Artificial Neural Network

CNN Convolutional Neural Network

DNN Deep Neural Network

LSTM Long Short Term Memory

ReLU Rectified Linear Unit

RESNET Deep Residual Networks

RNN Recurrent Neural Network

VGG Visual Geometric Group

YOLO You Only Look Once

Chapter 1

Introduction

In a growing nation like Bangladesh, personal vehicles are rising quickly. A sizable portion of these new automobiles are motorcycles. People, especially young people, are drawn to two-wheelers due to the relative affordability and ease of travel. We are counting the growing number of motorcycle accidents and the expanding number of motorcycles. More than 945 people died in motorbike accidents in 2019. The following year, 2020, saw a rise in that number, with 1463+ fatalities recorded. According to a survey conducted in 2021 by a non-governmental group, motorcycle accidents made up 39% of all accidents that occurred in Bangladesh that year, with multiple 2078+ accidents claiming the lives of 2214+ persons. Compared to earlier surveys, the number of motorcycle accidents climbed by 50% between 2020 and 2021, with a more significant fatality rate increase of 51%[28]. According to data gathered by road safety foundations, 830 persons died in the first four months of 2022. 88% of those killed in motorcycle accidents, according to BUET research, were not wearing helmets. The age range of those killed in motorcycle accidents is 14 to 45[25]. According to additional analysis based on that study, 66% of motorcycle riders at the time were not wearing helmets. Motorcycle accidents worsen when drivers and passengers do not wear helmets, frequently leading to severe head injuries. The empirical evidence indicates that the use of helmets is associated with a substantial decrease in the incidence of deaths resulting from motorcycle accidents. Based on available data, it has been shown that motorcycle riders may achieve a mortality reduction rate of up to 37%. At the same time, passengers can get a mortality reduction rate of up to 41%[40]. This alarmingly high rate of motorcycle-related traffic accidents is a global issue as well as a regional one. In addition, it is difficult for the police to determine who does not have a helmet so that they can take the necessary measures. In our case, automating this process will be helpful. Different algorithms are required for this kind of automated system. Several artificial intelligence techniques have been applied in this work, as we have shown. Deep learning will be necessary to implement our method. Our approach can identify motorcycle riders who are not wearing helmets after training. We will outline in our paper how we want to use the video data we've obtained from the roads of Dhaka, Bangladesh and convert them into image data. The deep learning system developed in this study can accurately assess the presence or absence of a helmet on a motorcycle.

1.1 Research Problem

Head injuries are the most frequent cause of bike accidents. Bike riders are particularly vulnerable to dying in accidents. Despite making up only 3% of all registered vehicles in the US, motorbikes were responsible for 14% of all traffic fatalities. Furthermore, throughout the year (2019–2020), the number of fatalities involving motorcycle riders and passengers rose by 11%. Deaths have grown by nearly 20% in the past ten years; there have been 5,579 fatalities, including motorbikes, with a rate of 31.64 per 100 million miles driven by vehicles[27]. Wearing a helmet can lessen the severity of any brain injuries one might sustain in the case of a motorbike accident. As things are, bike riders are legally required to wear helmets, and breaking this rule can result in expensive fines. Despite this, a significant fraction of motorcycle riders violate this law. The physical measures taken by the police to contain this situation are ineffective given the actual situation[39]. Helmets reduce mortality risk and brain injuries. In the case of brain injuries, helmets can help reduce it to 42%, and for mortality rate, it reduces to 69%, respectively. Therefore, all motorcycle commuters and passengers should wear them.[41]. Our paper suggests a program to identify bikers not protecting their heads with helmets and collect their license plates to identify motorcyclists.

1.2 Research Objectives

Our primary goals are to lower the cost of labour and the number of accidents that could be prevented by not wearing a helmet. In congested areas or during emergencies, it might be challenging for traffic officers to identify helmetless bikes manually. Additionally, manually checking involves laborious work. The answer to that issue is to create an automatic detection process to identify a motorcyclist riding without a helmet. Students often die in accidents because they do not wear helmets when preparing to serve their country. Due to not wearing a helmet, a person who provides the only source of income for a family risks untimely death just when they need him the most. To prevent accidents, both the rider and the pillion passenger in the backseat must wear helmets.
According to research conducted by the World Bank and the BUET Institute of Accident Research, helmets have been shown to reduce the likelihood of deaths by 40% and severe injuries by 70%[26]. The present investigation included a sample size of 400 individuals who engaged in riding activities within the geographical regions of Dhaka, Barisal, the Dhaka-Mawa highway, and the Dhaka-Gaibandha highway; the data collection period spanned from July 2019 to July 2021. According to the study, nearly 53% of motorcyclists wear helmets most of the time, with 10% of those riders wearing them rarely and 2% not wearing them at all. The most frightening statistic is that 88% of bikers who died while riding were found to be without helmets. This issue is not exclusive to Bangladesh. Accidents involving a lack of a helmet happen all over the world. The lack of helmet wear was responsible for 83% of the deaths of bikers in Maharashtra in 2020, as reported by the MORTH[31]. According to the cited source, it is evident that the lack of helmet use played a significant role in more than 39,500 fatalities resulting from road accidents in India during the year 2020. Furthermore, it is noteworthy that Maharashtra accounted for 12% of these fatalities. The likelihood of sustaining brain damage or experiencing a fatality is much higher

in bicycle accidents if the rider's head is inadequately safeguarded. Most mishaps occur due to the failure to use an appropriate helmet. However, wearing a helmet may protect individuals from sustaining severe injuries and experiencing untimely mortality.

It was found by an insurance company related to Highway Safety that just in 2014, greater than 60 per cent of bicycle-related fatalities were attributable to riders who were not wearing helmets.

That proportion has been demonstrated to be as high as 97% in other local investigations[18]. According to studies conducted by WHO, it is stated that around 1.3 million individuals succumb to fatal accidents occurring on roads and highways annually. Statistics also reveal a 42% drop in fatalities and a 69% drop in the likelihood of suffering a brain injury.

In addition, a 2015 research by the United Nations found that meeting the necessity of a helmet increases an individual's odds of survival by 42% [29]. Our first goal is to develop a fully automated method for identifying helmetless bikers. This system is intended to enhance traffic legislation and policies, minimize the need for human inspection by law enforcement personnel, and, crucially, mitigate the occurrence of accidents.

Chapter 2

Literature Review

In the article by Vishnu [7], a technique for the automated identification of motorcycle riders operating their vehicles without helmets is described. The moving item was extracted from the recorded video's corners using adaptable subtraction from the backdrop. The motorcycle riders in the film were also identified with the help of CNN. Based on the findings presented, the recommended approach showed a high level of efficacy, successfully identifying 92.87 per cent of the offenders while keeping the average rate of false positives below 0.5%.

In their paper, Rohit [11] did show a similar deep-learning method for helmet detection, which he and his group members developed. The model used for both extraction and detection was the Caffee model. They have seen the result near 86 accuracy. And they have also used another method, the Inception V3 model. Regarding image classification by this model, accuracy was near 74.

In Harsh Nagoriya's [16] article, they presented an automated framework for differentiating motor riders on the time of riding a motorcycle who did not have a helmet. The Model used a Multibox Detector (SSD) model (Single Shot) to differentiate between the bounding boxes of the bike and the rider, as well as to choose motorcyclists from a crowd of moving objects and to recognize motorcyclists without helmets. Additionally, helmetless motorcyclists' license plates were identified using the YOLO framework. Therefore, various system parts have used the SSD, YOLO, and bespoke CNN models. The bicyclists' number plates were then scanned using the YOLO model, and the data was recorded into a linked database. Their model had an accuracy of about 96.97% when trained on the binary classification of with and without helmets.

In the article by Narong [24], an approach to detecting helmets that uses deep learning and is referred to as Single Shot Multibox Detection was developed. The methodology involves using a singular CNN to detect and outline the bounding box area that includes the motorcycle and the rider. In addition, the network can determine whether they have a helmet on. The experiment results were positive, as the use of Deep Learning and CNN approaches proved influential in developing algorithmic solutions for the issues associated with photo recognition.

Bhaskar, P. K., and Yong, S.-P [3] did the study. (2012) centred on the identi-

fication of vehicles using image-processing techniques derived from video frames. The research included using algorithms for the categorization of vehicles, Gaussian mixture models and object detection methodologies for tracking. Using binary processing, we extracted the foreground from the background of the chosen frame. This included the creation of rectangular regions surrounding the identified item as well as the implementation of a foreground detector to identify objects in the foreground. It was observed that their detection accuracy is 91%.

The research conducted by Mistry [6] and colleagues (2019) integrates convolutional neural networks (CNN) with YOLOv2 models. The YOLOv2 model, first trained on the COCO dataset, can detect and classify distinct objects. As a consequence, there will be fewer people going about without conspicuous helmets. The suggested methodology aims to enhance the accuracy of helmet recognition in the input picture by differentiating between the human class and the motorcycle class. The second YOLOv2 model processes the cropped photographs of the person being identified. Images of people while wearing helmets were used to train this model. . The pictures without headgear depict people whose license plates have been obscured. The suggested approach utilizes the COCO and helmet datasets. The experimental evaluation conducted on a diverse set of photos, including both helmeted and unhelmeted individuals, showed that the suggested methodology performs better than other established techniques, achieving a detection accuracy of 94.70%

In Adil Fazal's article [17], they proposed a methodology for motorcycle helmet detection using the Faster R-CNN deep learning model, with two phases: detection using the Region Proposal Network (RPN) and recognition. They set up a camera at the main entrance to an establishment by which they could gain 70 recordings from both a frontal and rear view of motorbikes passing through that main gate. The second data set was gathered from a region in Lahore, Pakistan, with a dense intercross section area. The thirty movies of drone-filmed aerial images that make up the third batch of data were collected from an overcrowded one-way traffic region in Lahore, Pakistan. The place is located in Pakistan. There is a total of 1750 minutes of footage split among three datasets. A total of 16,422 photos were used for both training and testing. Anchor points have been added to the image by RPN so that the helmet may be located. The model under consideration demonstrates a test dataset accuracy of 97.26% after training.

Kunal Dahiya and colleagues [5] devised a technique for identifying motorbikes in video surveillance material. They use SVM to determine whether or not a helmet is being worn by segmenting the item being worn and then subtracting the background from the segmented object. They completed the process of assembling their data collection. They have a 93.80% accuracy rate on data taken from actual surveillance around the globe.

B. Yogameena et al. [10] proposed employing deep learning (R-CNN) technique for automated helmet identification. B. Yogameena proposed a technique for detecting motorcycles in the foreground using the Gaussian Mixture Model to split video frames, label the resulting objects, and then use the Faster R-CNN algorithm. After that, bikers wearing helmets are singled out using the Faster RCNN. They

can determine the motorcycle registration numbers of riders without helmets by using Character Sequence Encoding and Spatial Transformation. They employ low-quality images with distortion, occlusion, a hairless head, and a person with hair of contrasting colours. For training, they used the TCE1 dataset from 2014, the TCE2 dataset from 2017, and the Bangalore1 dataset with the dataset - Bangalore2. They also have used Faster R-CNN.
And Bangalore2 datasets to train and assess the Faster R-CNN model.
From the TCE2 dataset, they retrieved the highest MAP score of 79.5% in the case of helmetless riders as 77.5% for riders with helmets. The TCE2 dataset contained the highest MAP scores, 79.5% for helmetless and 77.5% for helmeted motorcyclists.

Waranusast et al.[2] applied several algorithms as their first goal was detecting moving objects on the road. Firstly, he used the AGMM algorithm to identify the moving object. Then, His section works with backdrop removal, labelling-related components, moving direction detection, and background subtraction. He used geometric information and also detecting colour to classify vehicles. Once an item has been obtained, the KNN classifier is used to determine whether or not it is a motorcycle. The final section involves counting and removing the riders' heads from a motorbike region. Based on KNN, The extracted head's helmet status is determined in the last phase. In this part, we consider the mean circularity, mean intensity, and mean colour of each of the four quadrants of the head. According to his findings, 89% of bikers had their helmets seen.

Dharma Raj [8] used deep convolutional neural networking with image processing to detect on-road motorcycle users who are not wearing helmets. Like other works, they also first identified moving vehicles. They used the CNN model to classify vehicles, and upon classifying, they discarded vehicles other than 2-wheelers as their system only consists of a method detecting helmets on motorcycle users. For identifying helmets on riders, they used their CNN model. Though their method identified helmets, that method also identified hats as a helmet, creating concerning issues. However, they concluded that by adding more photographs of bikers wearing caps, they might precisely remedy this problem.

In their publication, Boonsirisumpun et al.[9] (2016) demonstrated an approach using a combination of VGG16, VGG19, GoogleLeNet (Inception v3), and MobileNets). Their study aimed to classify photos containing motorcycle riders, distinguishing between those wearing helmets and those without helmets. The authors employed datasets comprising helmet-wearing and non-helmet-wearing motorcycle rider images for training and evaluation. He used a deep learning technique for image detection known as SSD to find a rider who isn't wearing a helmet in a video clip. This study used a four-step process: gathering videos and images, experimenting with image classification, detecting, and analyzing the results.

Chiu et al. [1] introduced an algorithm for the computational vision-based system for counting vehicles. Some motorbikes could be blocked by other cars, so the objective would be to locate those blocked motorbikes. Helmet detection technology is used to find motorcycles. The system takes the shape of the helmet region to be circular. To find the helmet, the boundaries of the image are estimated over the potential

area where the motorcycle might be. Next, they determine how many edge points have a spherical appearance.
During system calibration, the result is checked to a pre-defined output, and if it's higher than the threshold, the region is taken as having a helmet.
Upon detecting the helmet, it considers that a rider is present there. The control system must set specific parameters during calibration, including helmet size, camera view angle, and distance from the ground.

Chapter 3

Methodology

3.1 Workflow Diagram

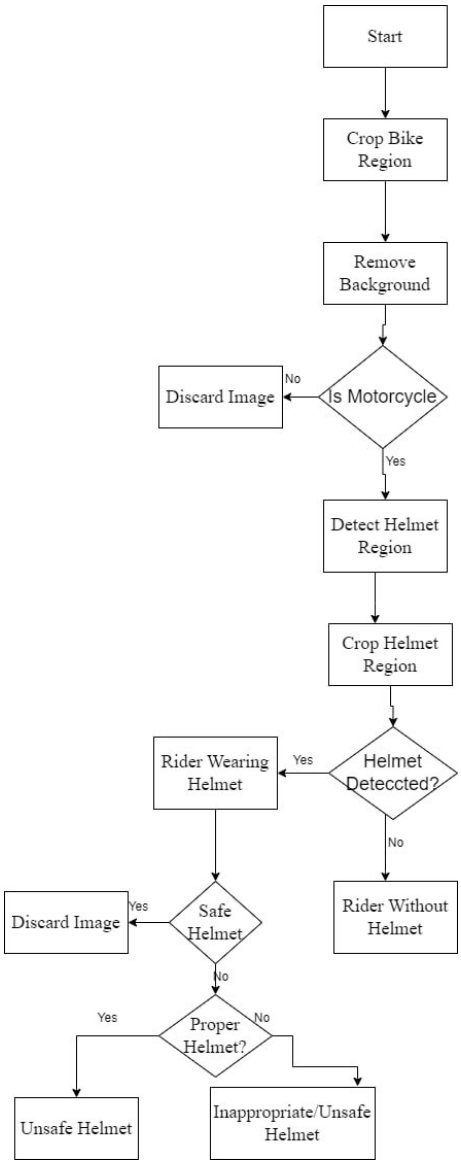


Figure 3.1: Dataflow of LSTM

3.2 Dataset

The efficacy of an automated system may be limited when it is only trained on a singular dataset. Images from diverse datasets are often acquired via scanners, cameras, and similar devices. These images exhibit variations in pixel density, colour intensity, and other relevant characteristics. A more accurate diagnosis can be achieved by training an automated system with various photos; training models using multiple images is essential. We used two separate datasets to train our suggested model and the other five transfer learning models. Our first dataset was used to identify bike riders wearing helmets or not, and our secondary dataset was used to train the models to distinguish between safe, unsafe, and inappropriate helmets (such as the hats, caps, and turbans worn by construction workers). The Canon EOS 80D is furnished with a megapixel count of 24.2. It also uses an APS-C CMOS sensor and Canon's special DIGIC 6, an image processor developed by Canon, hence offering a diverse array of functionalities suitable for both photographic and videography purposes. The camera has a 45-point autofocus system encompasses all cross-types, facilitating accurate focusing. In addition, it can take seven shots per second in continuous mode. The camera can also record high-definition video. The videos were shot close to BRAC University on the streets of Dhaka. The bikers' images were edited from the films. 5237 photos in all were taken from the videos. The majority of the motorcyclists among them were wearing helmets, so we also needed to gather a few pictures of riders without helmets from the internet. These photographs weren't taken from other datasets; instead, we just used Google Photographs and cropped screenshots from online videos of unrelated content. Images of helmetless riders were gathered from the internet at 1028. The models were trained using a total of 6265 photos.

Class type	No of. images
WithHelmet	3,626
WithoutHelmet	2,639

Table 3.1: Primary Dataset

Class type	No of. images
SafeHelmet	1,316
UnsafeHelmet	1,018
InappropriateHelmet	692

Table 3.2: Secondary Dataset

Our primary dataset included 3626 photographs of motorcyclists wearing helmets and 2639 images where the rider lacks helmets. This dataset was utilized to train algorithms to detect helmets. Likewise, our secondary dataset, used to determine whether a helmet was safe, unsafe, or unsuitable, had around 1316 photos of safe helmets, 1018 images of risky helmets, and 692 pictures of inappropriate helmets. ASP Ziaul Haque, DB, has kindly reviewed the safety checks of these helmets.

3.2.1 Dataset Split

The dataset must be split before the photos can be fed into the suggested model. To allocate the dataset for training, validation, and testing, we used a ratio of 80% for training, 10% for confirmation, and 10% for testing. There are 5012 photos in a dataset for the primary dataset, used as the training dataset, 626 in the validation dataset, and 627 in the testing dataset. Our exact ratio was kept for the secondary dataset, which has a training dataset with 2421 images and validation and testing datasets with 302 and 303 images, respectively.

3.2.2 Experimental Setup

This research used an Intel core i5 12600K CPU and 16 gigabytes of random access memory (RAM). The Gigabyte RTX 3060 GDDR6 GPU, having 12 GB of VRAM, is integrated. The Integrated Development Environment (IDE) used for this project was Jupyter Notebook, specifically v. 6.4.12.

3.3 Object Classification Model

Here, we provide the proposed architecture and model for comparison and design. CNN and LSTM networks with conceptual mechanisms were chosen. In our response, we will briefly describe these classes.

3.3.1 CNN

Deep learning often employs a Convolutional Neural Network (CNN) to analyze visual pictures. Since 2012, convolutional neural networks (CNNs) have emerged as the primary method for tackling computer vision issues because of their superior

results. [38] [42]. To autonomously plus dynamically learn spatial hierarchies of input via backpropagation, convolutional neural networks (CNNs) use convolutional, pooling, and fully connected layers.
[36]. CNN has at least one convolution layer to reduce the computational complexity of tasks like picture identification [35].

Input Layer

There is an image in the input layer. Depending on its size and kind, the system divides the photo into an array of pixels and then puts them in the matrix. There is only one pixel in a picture that is grayscale. The three pixels and colour planes that make up an RGB picture set it apart. Various colour space sizing options exist, including HSV, CMYK, and c24. Matrix size is reduced using the Convolutional layer for a more accessible selection of necessary features to make handling easier.

Convolution Layer

The first layer generates a wide range of information from the input photographs. At this level, the convolution process involves performing a mathematical operation on the input picture using a filter of a predetermined size, MxM. Using a convolution operation, the dot product within the filter with different parts of the input picture may be computed, considering the filter size (MxM)c26 c27.
Consequently, the corners and edges of a picture are meticulously delineated and documented inside a designated representation termed the feature map. Subsequently, subsequent layers can access this feature map to acquire novel features in the given picture.
The output from CNN's convolution layer is passed to the following layer once the convolution operation has been applied to the input. CNNs maintain the spatial correlation among pixels.

Non-Linearty

After every Convolutional operation, this layer is added. This layer incorporates an activation function to make the visuals less linear. Numerous activation methods exist, including tanh, ReLU, sigmoid, Leaky ReLU, softmax and Parametric ReLU.[R8]. The network cannot effectively model the class or label without non-linearity.

Pooling Layer

The layer after that is a pooling layer. The pooling layer carries out a max-pooling procedure. With the help of pooling, the most significant traits are extracted whilst discarding unfavourable ones—resulting in a smaller map of features.
The three types of pooling are: 1. The Max Pool 2. Sum Pool 3. Average Pool. [R7]. As a result, the layer takes the convolution layer's feature maps, is fed as input, and gives a feature map (pooled).

3.3.2 LSTM

One Recurrent Neural Network (RNN) that may help with the issue of long-term dependency is the Extended Short-Term Memory Network. To prevent long-term dependence, as seen in Figure 4.5, LSTM comprises four layers integrated in a novel fashion, as opposed to only one neural network layer. The cell state, represented by

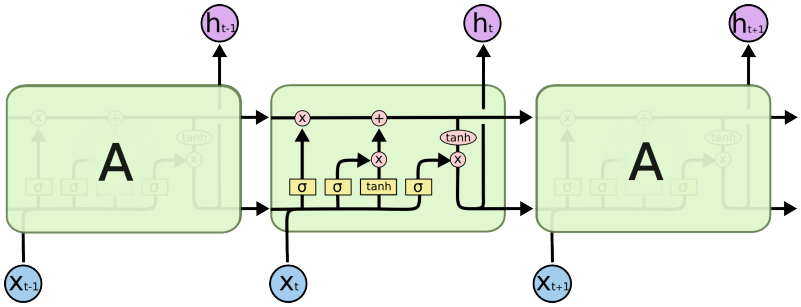


Figure 3.2: Dataflow of LSTM

the continuous horizontal line, functions like a conveyor belt. The figure shows it travels directly through the system with a few minor linear computations.

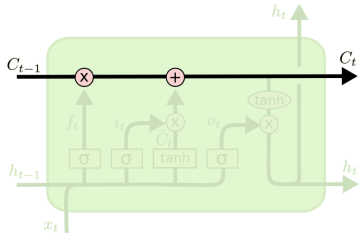


Figure 3.3: Cell State

The gate-controlled cell state seen in Figure 4.7 can be erased and updated by the LSTM. Gates function as a technique of permitting information to flow through only under certain circumstances. The sigmoid layer generates numbers between 0 (not pass data) and 1 (pass data). LSTM standard gates:
1. The forget gate,
2. input gate, and
3. output gate

In the first phase of LSTM, the "forget gate key" is used to pinpoint the specific pieces of data that need to be removed from the cell state.
The forget layer, responsible for evaluating the inputs ht1 and xt and producing a scalar value between 0 and 1 for each element in the cell state, can be considered a sigmoid layer. The ignore layer is shown in Figure 3.4. The following is the candidate memory cell equation:
Input Gate: The final step entails selecting the new data incorporated into the cell's state, consisting of two distinct components. The "input gate layer," a sigmoid layer, is activated as the initial step in updating information, and it is responsible for identifying which pieces of data need to be updated. The state is added to a vector created by a tanh layer containing new values. These two will be joined in the subsequent step to update the state, as shown in Figure 3.6.

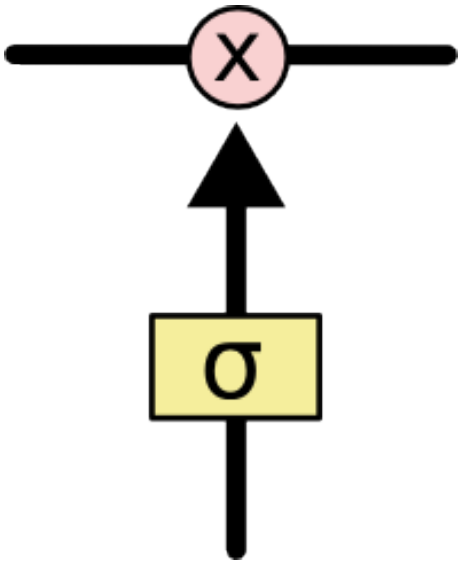
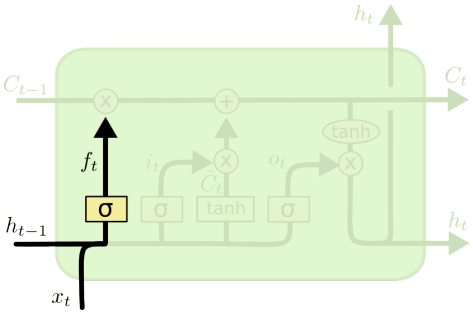
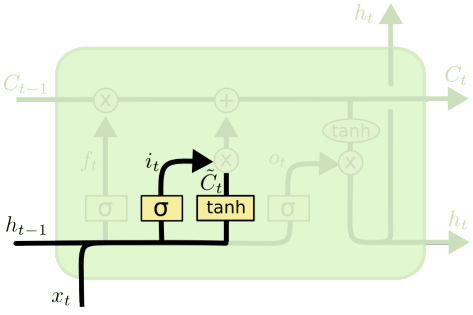


Figure 3.4: Gate Layer



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

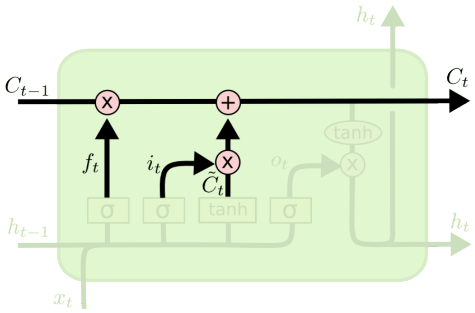
Figure 3.5: Forget Gate



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 3.6: Input Layer, Sigmoid and tanh operation

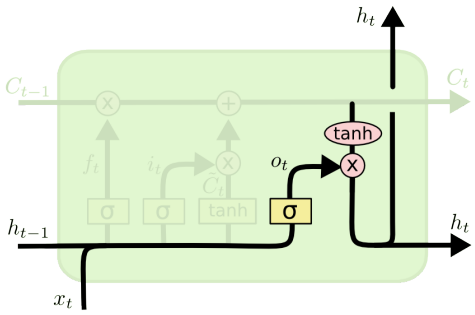
By executing the preceding steps, the values of the old cell state, Ct1, are now promoted to the new cell state, Ct. It multiplies the previous state without regard to the selected values. After that, we add to it—Cet. As seen in Figure 3.7, the present compilation of potential values has been generated by computing the extent to which each state value has been modified.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 3.7: Updated Value

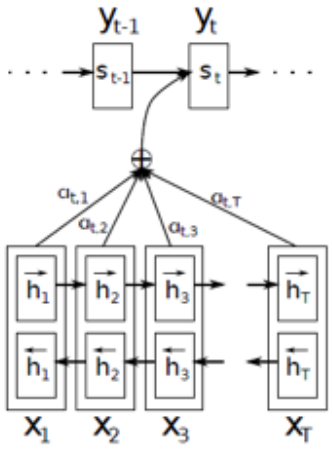
Output Gate: Depending on the cell state, the final state will decide the output that will be filtered. A sigmoid layer selects which parts of the cell state will be transmitted as output. Subsequently, the cell layer will use the tanh form to generate values from 0 to 1. In the end, multiplication will be performed using this result and the sigmoid value to get the outcome seen in Figure 3.8.



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 3.8: Output Layer

3.3.3 Attention



The above visual representation depicts the Attention model as proposed by Bahdanau in their study. In this particular scenario, Bidirectional LSTM produces a single set of footnotes (h1, h2,..., hTx) for every given phrase [37]. The researchers

$$h_j = \left[\overrightarrow{h}_j^{\top}; \overleftarrow{h}_j^{\top} \right]^{\top}$$

in question have effectively combined the secret states of the encoder, both in the forward and reverse directions, to generate the vectors h1, h2, and so on.
Concisely, the vectors h1, h2,..., hTx correspond to the number of syllables, Tx, while the input phrase. At the primary encoder and decoder paradigm context, the encoder LSTM's only state used as the context vector was the most recent state (referred to as hTx in this case). The procedure for calculating weights is currently in question. A feed-forward neural network also learns the weights; their equation is below. Ci; the annotation's weighted sum will generate the context vector from the output word

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

The weights ij are determined using a softmax function, which this equation can mathematically express.:

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T_x} \exp\left(e_{ik}\right)},$$

eij, The terminal value of an FNN measures the degree to which the input at j and the output at i coincide. A function defines it.

$$e_{ij} = a(s_{i-1}, h_j)$$

The feed-forward network's input dimension is reduced by combining the decoder's prior state, which has d dimensions, with the two vectors (Tx, 2d). This is where the encoder creates the Tx "annotations" that represent the d-dimensional hidden state vectors. Scores eij are calculated by performing a matrix multiplication on the input with a matrix Wa of dimensions (2d, 1). These scores have sizes (Tx, 1). Following that, the bias component is included in the scores.
By plugging the eij scores into a tan hyperbolic function, we can get the normalized alignment values for output j. followed by a softmax operation.

$$E = I[Tx * 2d] * Wa[2d * 1] + B[Tx * 1] \tag{3.1}$$

$$\alpha = softmax(tanh(E)) \tag{3.2}$$

$$C = IT * \alpha \tag{3.3}$$

3.4 Proposed Helmet Classification Prediction Model

We start our experiment by combining the CNN, LSTM, and Attention models. A Dilated CNN layer forms the foundation of our fusion model. Our dilated CNN layer comprises three maximum pool layers and five convolutional layers, each with a dilated rate ranging from 1 to 5 with each convolutional layer. A CNN has five layers of convolution, followed by an LSTM and an Attention model layer. The optimizer was Adam, the group size was 16, and the initial learning rate was 0.001. The principal model's loss function was categorical cross-entropy.
Dilated convolutional neural network (CNN) layers comprise three distinct portions. The initial component consists of a convolutional layer and a max-pooling layer. The input component of Block 1 is linked to the first Conv2D layer, which has a 16-bit kernel. Picture loaded in has a depth of colour of 3 and a resolution of 224x224. Following the application of the first max pooling layer, the dimensions of the output from the first Conv2D layer are reduced to 111 x 111. The configurations of Blocks 2 and 3 vary solely in terms of kernel size. Two convolution layers with kernel sizes of 32 and 64 each are present in the second block, along with a maxpool layer. The exact mix of kernel sizes is seen in Block 3. The corresponding maxpool layer follows these blocks' Conv2D layers. The input size approaches (16, 16) after the third max pool layer, which is too little to send down to the LSTM layer; therefore, a reshape layer is added, reshaping the size to 256. Because we seek to focus on the head region of bike riders to detect helmets, the Attention model was the most logical choice because we know it is best to focus on a specific portion of the datasets.

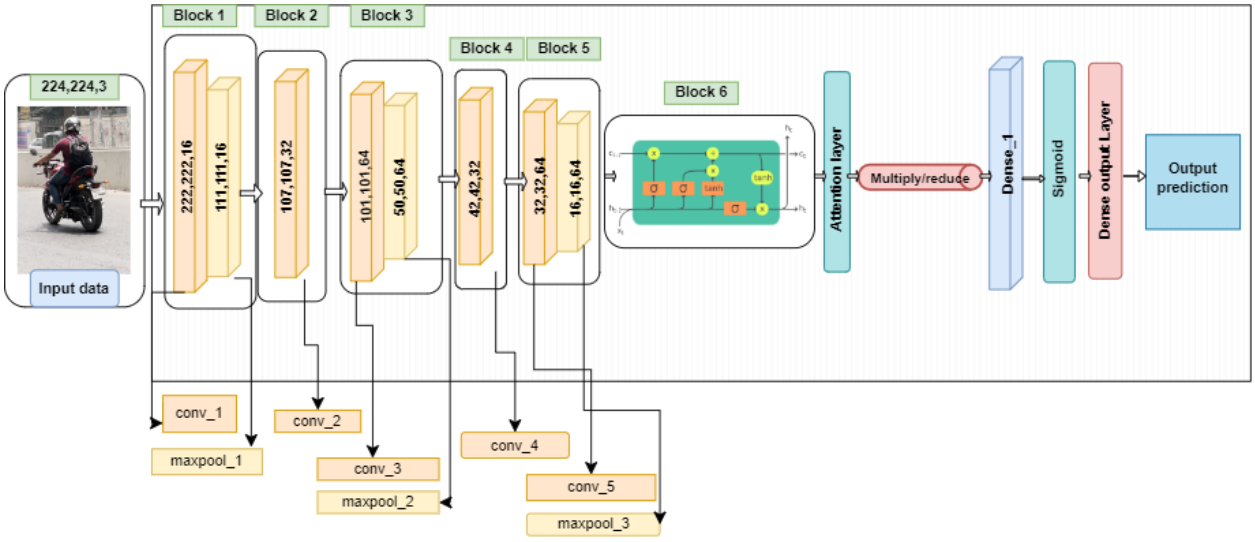


Figure 3.9: Framework of base CNN model

After the three models are fused and multiplied to decrease the number of feature maps and increase computational efficiency, a dense layer is added to enhance output

prediction. The fully connected layer (FC), the classification dense layer, has two neurons for the first dataset's two classes and three classes for the second dataset's three classes. It features Sigmoid activation functionality. All three convolutional layers use kernels of uniform size and utilize the rectified linear unit (ReLU) as the shared nonlinear activation function. Additionally, these layers have a stride size of 1x1. The research showed that the Rectified Linear Unit (ReLU) outperformed the Parametric Rectified Linear Unit (PReLU) within our stated model. Our proposed model has 97,923 trainable parameters. Loss functions are crucial during model training because they allow the trainer to measure the expected and real network output errors. At the same time, the model's starting weights are significant for determining which features to pull out of the data. After each training period, the error rate is collected and analyzed to assess the kernels' performance. This allows for the most refined kernel properties to be extracted and tweaked after each epoch. The input is received by Block-1, which then applies the first convolution layer. The current layer consists of 16 filters, resulting in a cumulative count of 448 parameters. The first convolution layer utilizes some filters (16) to preserve the input image's intrinsic structural and textural characteristics. The input produces a set of sixteen feature maps, which are further processed using the ReLU activation function. The Rectified Linear Unit (ReLU) function maintains the nonnegative values inside the feature maps, therefore rectifying them. The output feature maps derived from the first convolutional layer undergo a halving process via 2 x 2 max pooling. The convolutional layers in Blocks 2 and 3 are composed of 32 and 64 filters, respectively. Block 2 contains a parameter count of 4640 and 18496 for 32 and 64 filters. Similarly, Block-3 has a parameter count of 18464 and 18496 for 32 and 64 filters. The final convolutional layer of Block 3 produces a 32x32-pixel feature map with a total of 64 filters. The next max-pooling layer shrinks the feature maps to 16 by 16 pixels. After the conclusion of block 3, 64 feature maps were generated using cumulative counting. The feature maps include a greater complexity in terms of input data characteristics, including deeper features and more complicated objects and forms compared to the previous blocks. A 16384-value unidimensional vector is created for each input using the multidimensional feature mappings in Block 3.A dropout layer with a dropout rate 0.5 is included after the first fully connected (FC) layer. The second fully connected (FC) layer has a sigmoid activation function and serves as a classification layer. The coating consists of two neurons in the first dataset, but in the second dataset, it consists of three. Using the sigmoid activation function in this particular layer facilitates the gradual abstraction of distinctive characteristics, leading to the generation of predictive results, including all five categories within our datasets.

Figure 3.9 presents a visual representation of the architectural architecture of our proposed model, as recommended.

3.5 Short Description Of Pretrained Model and Architectures Used For Comparison and Composition

EfficientNet-B0:

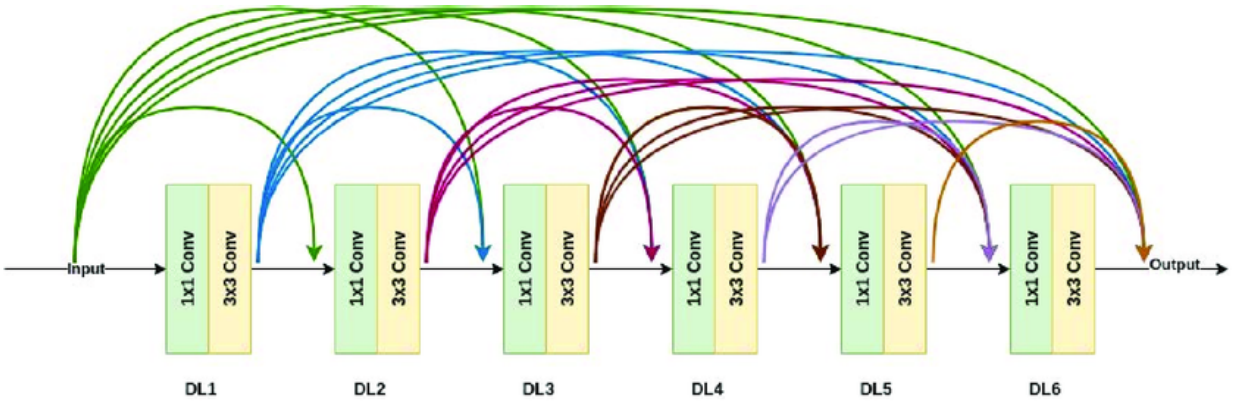


Figure 3.10: EfficientNetB-0

EfficientNet is a model for convolutional neural networks that aims to maximize accuracy while minimizing computation. It accepts specified image resolutions and contains convolution blocks, including a convolution layer, batch normalization, swish activation, and depth-separate convolution (DS Convolution). Using the EfficientNet-B0 compound scaling method, it accurately scales in width and depth. The model gradually increases the number of paths by expanding feature maps on deeper mesh layers. By storing global average pooling channel data, adjusting input image sizes, and reducing parameters, spatial dimensions are reduced to 1x1. The model contains a fully connected (FC) layer that converts extracted features into output classes for particular tasks, preventing overfitting the mouth. Standard optimization algorithms, such as SGD or Adam, can be used to train EfficientNet with adequate learning rates and weight reduction. Data enhancement techniques increase the generalizability and diversity of a dataset. EfficientNet creates a pretrained computer vision model using large data sets such as ImageNet. Standard measures such as precision, accuracy, recall, and F1 scores for performance specifications are used to evaluate performance hypotheses [12].

VGG16:

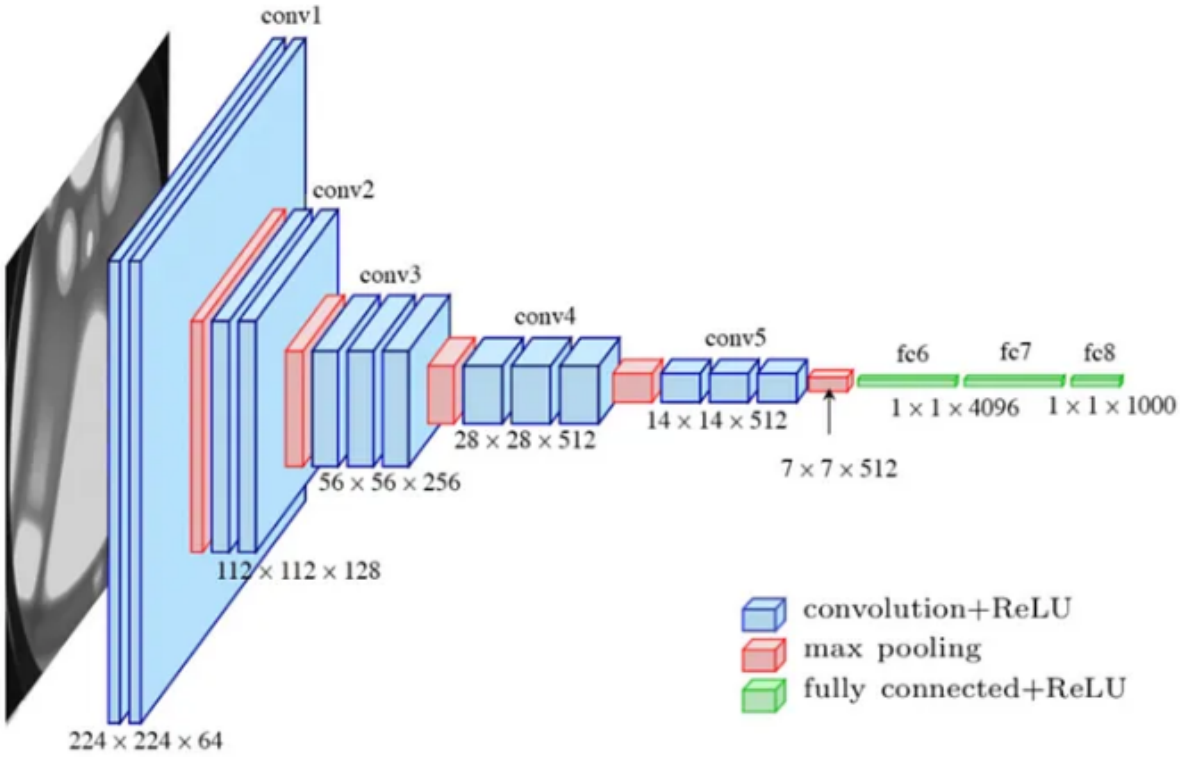


Figure 3.11: VGG16

The VGG16 convolutional neural network architecture achieved first place in the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It is characterized by sixteen weight levels and an input picture size of 224x224x3. The measurement employs thirteen convolutional layers with a three-by-three filter and maintains spatial resolution at levels 64-512. Nonlinear ReLU activation function, convolutional layers with reduced spatial dimensions, and extracting significant features using maximum pooling layers. Layer three contains 4096 neurons and the Softmax activation function for classifying multiple classes. Computer vision applications extensively utilise small-batch stochastic gradient descent (SGD) with kinetic generation strategies like weight loss and learning rate reduction. VGG16 with pre-trained ImageNet weights is widely used for transfer learning. Precision, accuracy, recall, and F1 scores are typical performance measures [13] [4].

VGG19:

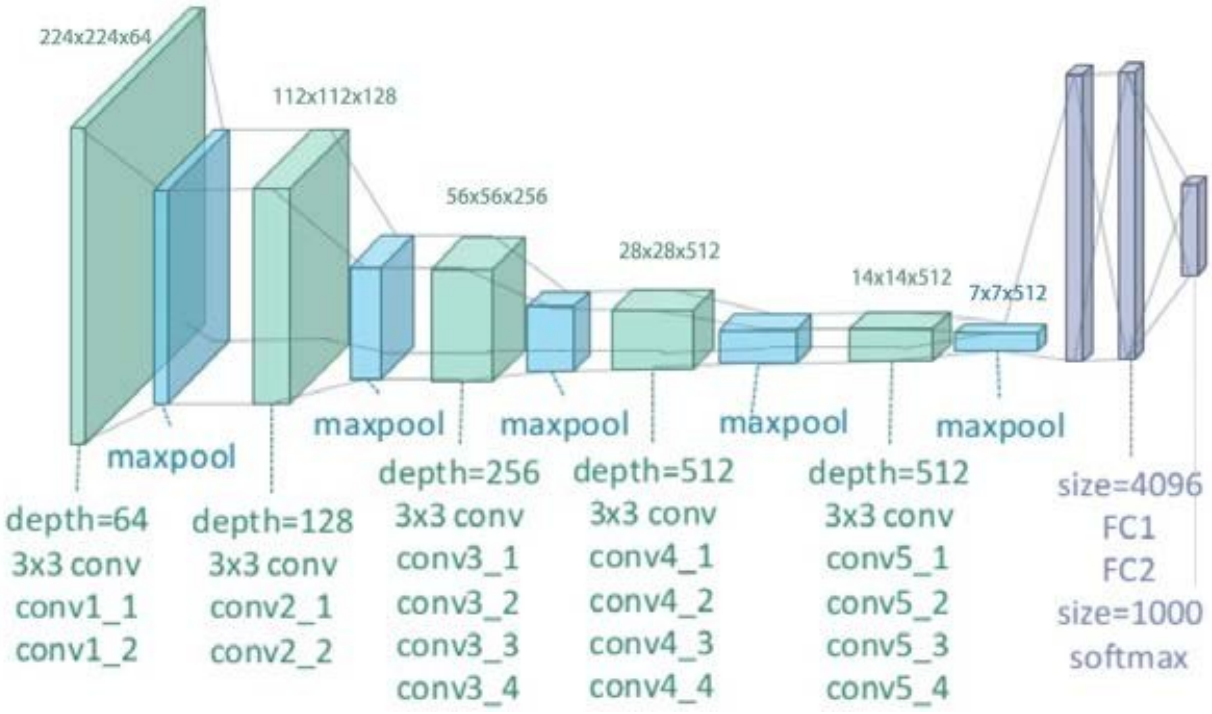


Figure 3.12: VGG19

VGG19 is a segmentation model based on deep neural networks. There are 19 layers, including fully connected and convolutional layers. The model's defining characteristic is using small 3x3 filters in the convolutional layer to detect image features. The model's input is typically a 224x224-pixel image. VGG19 consists of sixteen convolutional layers, with three filters per layer. These layers assist the model in recognizing images, including their borders, corners, and textures. In addition to multiple convolutional layers, maximum pooling layers with 2x2 filters are employed to decrease the spatial dimensions of the learned features. Three lines with complete connections and several veins near their ends. These layers combine the known features into an image content prediction. In the case of big datasets such as ImageNet, which consists of 1000 classes, the output layer is designed to accommodate an equivalent number of neurons. The Softmax activation function transforms the scores at the final level into probabilities, indicating the likelihood of a picture belonging to a particular class. VGG19 is renowned for its simplicity and efficacy and has served as the foundation for numerous other deep-learning programs. It attained competitive results in image classification services and advanced computer vision [14].

ResNet50:

The ResNet50 architecture is a CNN brought by Camming et al. in 2015.The architecture comprises 50 layers, including convolutional layers, batch normalization layers, max-pooling layers, global average pooling layers, and fully linked layers. Residual blocks improve learning efficiency and consist of many layers, including convolutional, batch normalization, and max-pooling. Full-layer connectivity and

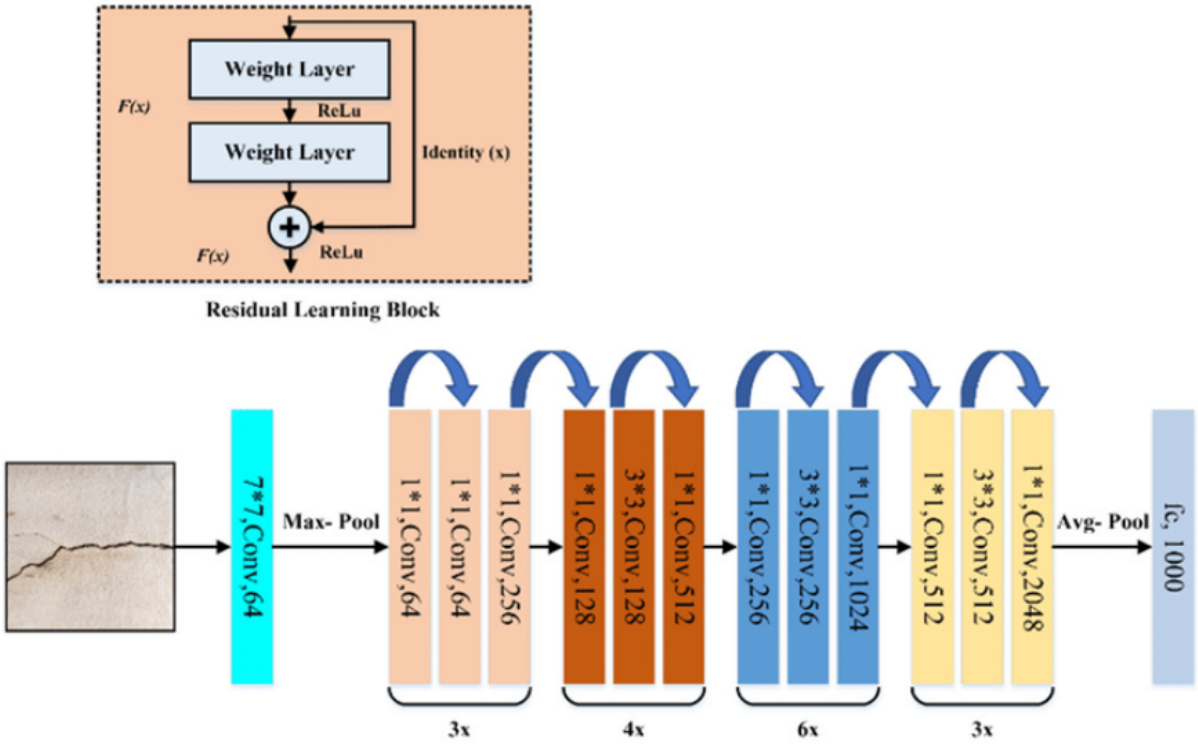


Figure 3.13: ResNet50

global-average pooling both see enhancements. ResNet50 identity mapping accepts as input a 224 by 224 RGB image. The grid consists of three sections: one with 64 filters, another with 128 filters, and a final section with 512 filters. The second section comprises 12 layers, each with 1x1 kernel convolution with 128 filters, 1x3 kernel convolution with 128 filters, and a final 1x1 kernel convolution with 512 filters. The third section consists of 18 layers, each with 256 filters, two kernel convolutions of 3x3, and a final with 1024 1x1 filters. The fourth component of kernel convolution states that ho has nine layers, each with 512 filters, three 3x3 kernel convolutions, and a final 1x1 kernel convolution with 2,048 filters. The final average pooling layer reduces the dimensions from a spatial composition to a linearized shape of 1 by 1, and the softmax activation functions comprise a fully connected layer containing 1000 units. ResNet50 architecture excels at various computer vision tasks, establishing itself as a foundational framework in deep learning.[22] [19]

InceptionV3:

Inception-v3 is an improved iteration of the Inception and Inception-v2 deep neural network frameworks widely used for image recognition. It uses Inception modules, blocks of convolutional neural networks, to extract multiscale image features efficiently. Inception-v3's fundamental architecture consists of Inception modules, which capture context through various filters. Inception module v1 contained 1x1, 3x3, and 5x5 convolutions, whereas subsequent versions included optimizations to reduce computational complexity. This centred distribution includes average pooling,

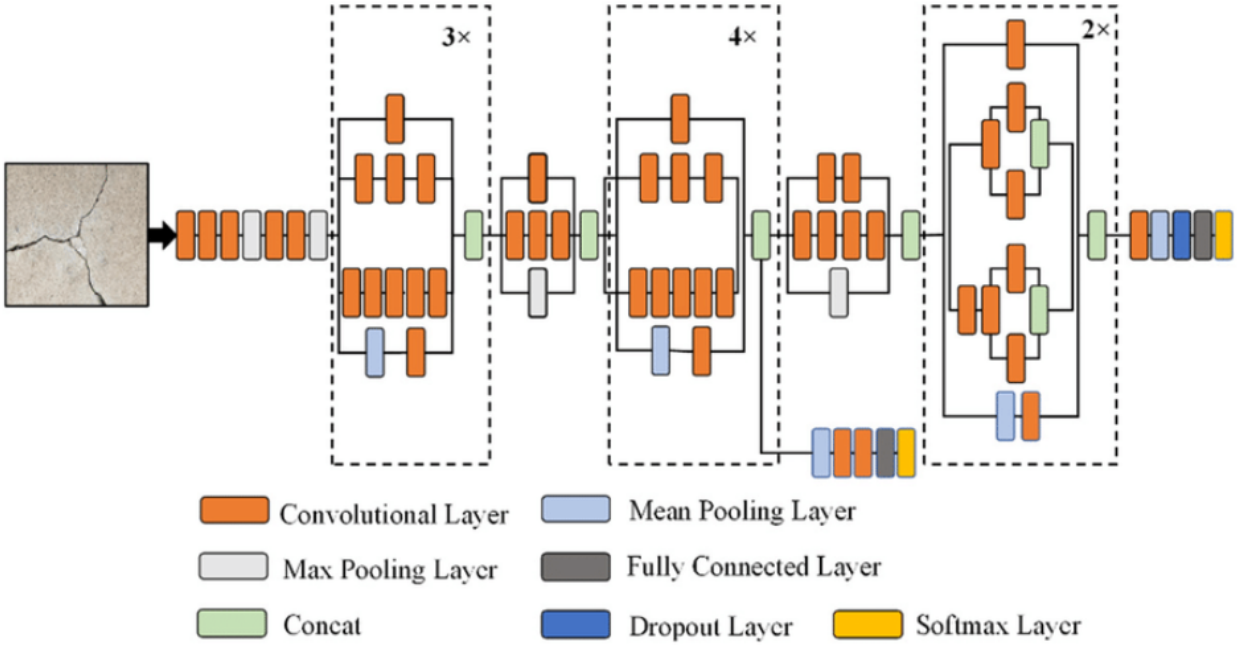


Figure 3.14: InceptionV3

convolutional levels, and levels with complete overlap. Inception-v3 has demonstrated exceptional performance in computer vision tasks and is widely employed as a feature extractor or base model for transfer learning in deep learning experiments.[15][21]

3.6 Optimization

3.6.1 Adam

Adam is one of the most often used algorithms in deep learning due to its effective and reliable optimization. The following features are incorporated to achieve this optimization: Beginning with stochastic gradient computing for optimization problems descent. Moreover, vectorizing minibatch stochastic gradient descent significantly enhances the use of larger sets of observations in a single minibatch. Efficiency. Momentum provided a technique for collecting a set of prior gradients to hasten convergence. Utilizing per-coordinate scaling was Adagrad. to enable an efficient preconditioner in terms of computing—RMSProp distinguished between per-coordinate scaling and a learning rate adjustment. One of Adam's key characteristics employed for estimation is leaky averaging. By using exponentially weighted moving averages and state variables, both momentum and the second moment of the gradient can be calculated in this way:

$$v_t \leftarrow \beta_1 v_{t\text{-}1} + (1 - \beta_1) g_t, \tag{3.4}$$

$$s_t \leftarrow \beta_2 s_{t\text{-}1} + (1 - \beta_2) g_t^2 \tag{3.5}$$

Scale and momentum are self-explanatory in state variables. The terms are redefined because of their definition, which can be altered by altering the initialization and

update condition. The term combo is also quite plain and easy to understand. Finally, we may control the step length by controlling the explicit learning rate to overcome convergence-related issues.

Chapter 4

Proposed Model Result Analysis

4.1 Evaluation Metrics

All classification models were evaluated in this study using several metrics, such as precision, recall, specificity, accuracy (ACC), false positive rate (FPR), false negative rate (FNR), false discovery rate (FDR), and negative predicted value (NPV). Figure 16 shows the best-proposed model's confusion matrix. True positive (TP), true negative (TN), false positive (FP), and false negative (FN) values are employed in the context of the given equations 4.1–4.9 to derive performance measures. Equations (4.10) and (4.11) are utilized to characterize the data, where (4.10) represents the total number of observations by m and (4.11) the predicted value of x by x^p. An equation is a mathematical statement that asserts the equality of two expressions. This is the equation:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

$$recall = \frac{TP}{TP + FN} \tag{4.2}$$

$$specificity = \frac{TN}{TN + FP} \tag{4.3}$$

$$precision = \frac{TP}{TP + FP} \tag{4.4}$$

$$F1score = \frac{2 * precision * recall}{precision + recall} \tag{4.5}$$

$$FPR = \frac{FP}{FP + TN} \tag{4.6}$$

$$FNR = \frac{FN}{FN + TP} \tag{4.7}$$

$$FDR = \frac{FP}{FP + TP} \tag{4.8}$$

$$NPV = \frac{TN}{TN + FN} \tag{4.9}$$

$$MAE = \frac{1}{m} \sum_{J=1}^{m} |x_{\mathrm{j}} - x_{\mathrm{j}}^{p}| \tag{4.10}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{J=1}^{m}(x_{\mathrm{j}} - x_{\mathrm{j}}^{p})^{2}} \tag{4.11}$$

4.2 ANALYSIS OF OPTIMAL MODEL RESULT

4.2.1 BEST MODEL PERFORMANCE METRICS

Specificity, accuracy, recall, ACC, F1-score, FPR, FNR, FDR, and NPV are only a few of the performance measures used to assess the models' usefulness across the two datasets.

According to 4.1 Table, the proposed model's optimal configuration for the first dataset had the following performance metrics: 92.313% specificity, 92.457% precision, 92.313% recall, 92.413% accuracy, 92.385% F1-score, 7.69% FPR, 7.69% FNR, 7.543% FDR, and 92.457% NPV. The FDR, FPR, and FNR readings are positive, and the results are almost all at or above 93

According to the data shown in Table 4.2, it can be seen that the best configuration of the proposed model exhibited certain performance measures for the second dataset. 89.101% specificity, 89.235% precision, 89.111% recall, 89.211% accuracy, 89.163% F1 score, 7.89% FPR, 7.89% FNR, 7.89% FDR, and 89.25% NPV.Memory FNR and FDR are 90

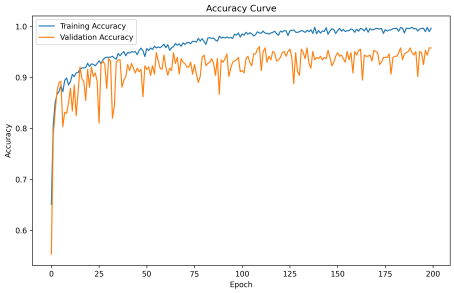
The model we've described performs an excellent job of categorizing the success metrics since all of them are good. The performance metrics for the study's best-case scenario model are shown in Table 4.1. The confusion matrix, AUC, training in contrast to validation accuracy, and loss for the best model on our primary dataset are shown in Fig. 16. Figure 16A shows minimal breaks in the training curve's smooth convergence from the first to the last epoch. There needs to be more evidence indicating overfitting throughout the training process, as seen by the absence of notable disparities between the accuracy profiles for validation and training. As shown in Figure 16B, the loss curve of the training curve exhibits a progressive convergence. The training and loss profiles examined showed no apparent signs of overfitting. The modified model's confusion matrix is shown in Figure 16C. In the test dataset, the row values reflect the actual data, while the column values show what is expected to happen. The diagonal value in the dataset reflects the true positive (TP) value. Without exhibiting any prejudice towards either social class, our proposed model demonstrates a high level of accuracy in predicting outcomes for both groups with the same degree of precision. The ROC curve, from which the AUC may be derived, is also shown for the reader's convenience. Briefly summarizing the model's discriminatory power is the area under the receiver operating characteristic (ROC) curve, also known as the area under the curve (AUC). When the space Under the Curve (AUC) metric approaches a value of 1, the model demonstrates a high level of class identification precision.

Performance study for the optimal design								
ACC	recall	specificity	precision	FPR	FNR	NPV	FDR	F1- score
92.41%	92.31%	92.32%	92.46%	7.69%	7.69%	92.46%	7.54%	92.38%

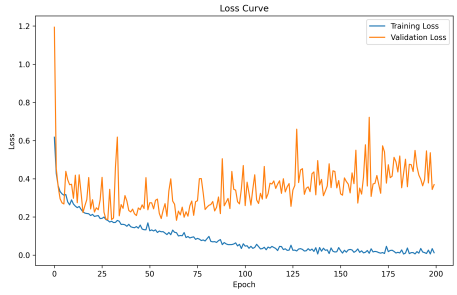
Table 4.1: Performance evaluation matrix for the suggested model's optimum configuration

Performence analysis for the best configuration								
ACC	recall	specificity	precision	FPR	FNR	NPV	FDR	F1- score
89.21%	89.11%	89.10%	89.24%	7.89%	7.89%	89.25%	7.77%	89.16%

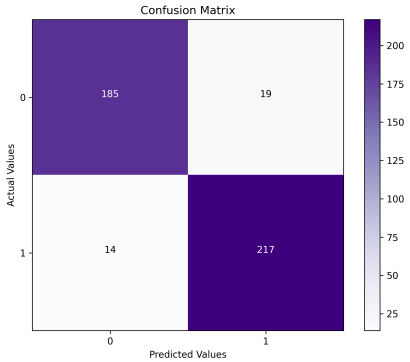
Table 4.2: Analyzing the performance evaluation matrix of the optimal configuration of the proposed Inner model.



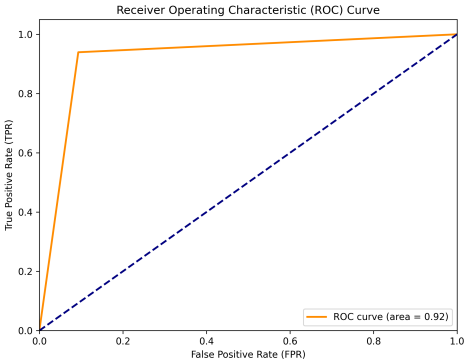
1. Training and Validation Accuracy



2. Training and validation loss



3. Confusion Matrix



4. AUC Curve

Figure 4.1: 1. Accuracy curve 2. Loss curve 3. Confusion matrix 4. AUC curve for the proposed BatNet-10 model's best performance following model optimization.

Chapter 5

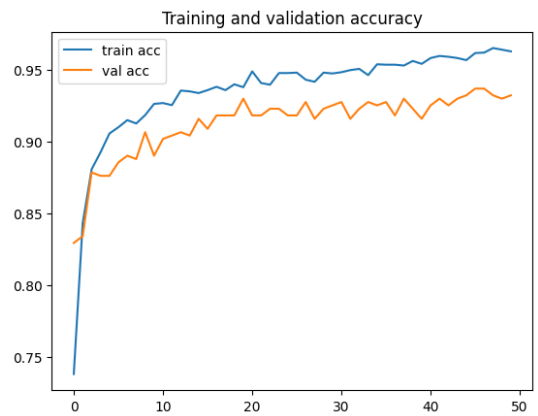
Pre-trained Model Result Analysis

5.1 Pre-Trained Model Implementation

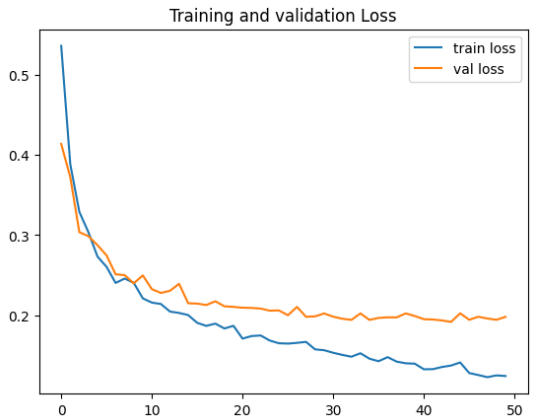
Pre-trained models possess a universal nature and are designed to exhibit high performance across many image-processing applications. Consequently, these models are conveniently available for download, accompanied by pre-trained weights. [9]. We used our dataset and some pre-trained models in this research: Inception, Resnet50, EfficientNetB0, VGG16, and VGG19. Consequently, we must add and modify the output layer to meet our requirements. Five additional nodes were added to the neural network architecture in our investigation. The softmax activation function for the output layer was chosen due to its suitability for classification. This decision was made because the first dataset contained two distinct image categories, and the second dataset contained three specific image categories [18]. This is a classification problem with multiple classes.

5.2 Pre-Trained Model Results

5.2.1 VGG19

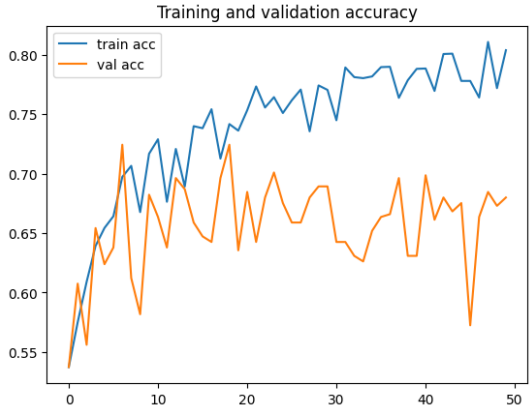


A. Train and Validation Accuracy

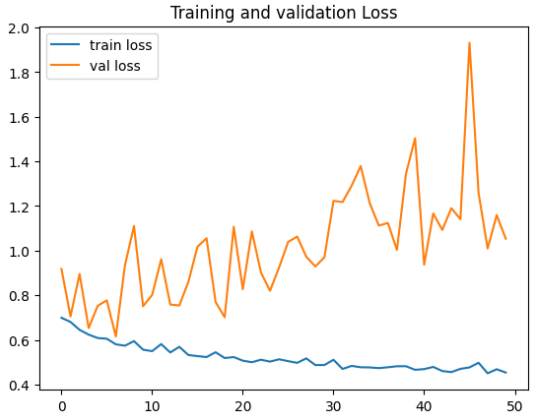


B. Train and validation loss

5.2.2 RESNET50

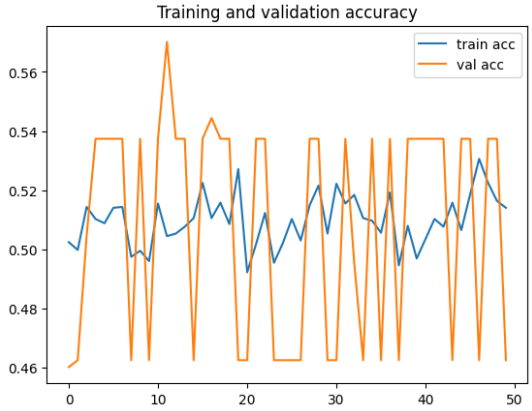


A. Train and Validation Accuracy

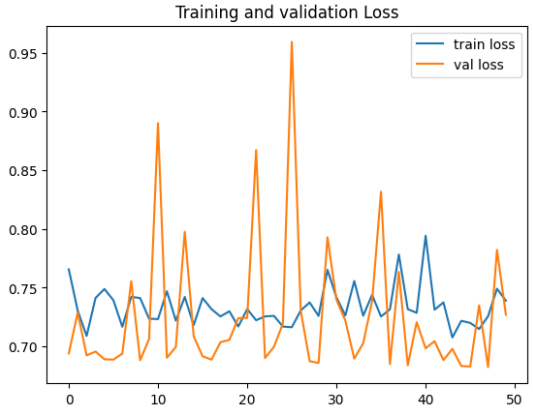


B. Train and validation loss

5.2.3 EfficientNetB0



A. Train and Validation Accuracy



B. Train and validation loss

5.3 An overall comparison between the pre-trained models and the proposed model

Models	No of. Layers	Epochs	Optimizer	Batch size	Image size	Learning Rate	Test accuracy
Efficient NetB0	237	50	Adam	32	224	0.0001	58.06%
Inception	48	50	Adam	32	224	0.0001	91.0%
ResNet50	50	50	Adam	32	224	0.0001	71.66%
VGG16	16	50	Adam	32	244	0.0001	90.32%
VGG19	19	50	Adam	32	224	0.0001	92.17%
Proposed Model	15	200	Adam	32	224	0.0001	92.41%

Table 5.1: result comparison

5.4 EarlyStopping Function

The EarlyStopping callback is commonly employed during the training of neural networks [30]. It offers the benefit of allowing a substantial number of training epochs while simultaneously terminating the training process when the model's performance ceases to improve on the validation Dataset. By utilizing the EarlyStopping callback, it is possible to construct a Neural Network so that the training process terminates when there is no further improvement in the model's performance. In our specific scenario, the training process for the pre-trained models was halted after 50 epochs.

Chapter 6

Object Detection model

6.1 YOLOv8

YOLOv8 is the most recent iteration of the YOLO model, designed to address various applications such as instance segmentation, object recognition, and image classification. The company Ultralytics, which also developed the well-known and industry-defining YOLOv5 model, is the creator of YOLOv8. YOLOv8 has several architectural and developer experience advancements and adjustments compared to YOLOv5. As of this post's writing, Ultralytics is actively working on new features and responding to community input for YOLOv8. Ultralytics provides models with long-term support once released, working with the community to ensure the model is optimized for maximum performance. Author of YOLOv8, Glenn Jocher of Ultralytics, noticed the YOLOv3 repository in PyTorch, Facebook's deep learning platform. Finally, Ultralytics published their model after enhancing training within the shadow repository: YOLOv5.[33]. There are many primary justifications for considering the utilization of YOLOv8 in one's forthcoming computer vision undertaking. According to Roboflow 100 and COCO measurements, YOLOv8 has a high accuracy rate. YOLOv8 has several developer-convenience features, including an easy command-line interface (CLI) and a well-structured Python package. The YOLO framework has garnered a substantial following, with a developing community centred on the YOLOv8 model. Consequently, computer vision professionals are likelier to provide valuable assistance and advice when sought. In the COCO dataset, YOLOv8 demonstrates remarkable levels of accuracy. For example, when assessed on the COCO dataset, the YOLOv8m (medium) model attains a mean average accuracy (mAP) of 50.2%. Upon evaluating the performance of YOLOv5 and YOLOv8 on the Roboflow 100 dataset, it was noted that YOLOv8 demonstrated much better outcomes than YOLOv5. The Roboflow 100 dataset is widely used for assessing model performance across several task-specific domains. We go into more detail about this in our performance study later in the piece[32]. In addition, YOLOv8 has certain important developer-convenience features. In contrast to other models that use a fragmented approach, segregating job functions into many executable Python files, YOLOv8 offers a Command Line Interface (CLI) that enhances the ease and accessibility of model training. Additionally, including a Python package that provides a more seamless development experience compared to earlier models is noteworthy. The YOLO community is notable when selecting a model for use. Yolo's functionality is well-known to computer vision specialists, and

a wealth of internet documentation is available regarding practical YOLO usage. Even though YOLOv8 is new as of this article's writing, many helpful tutorials are available online.

6.1.1 YOLOv8 Architecture

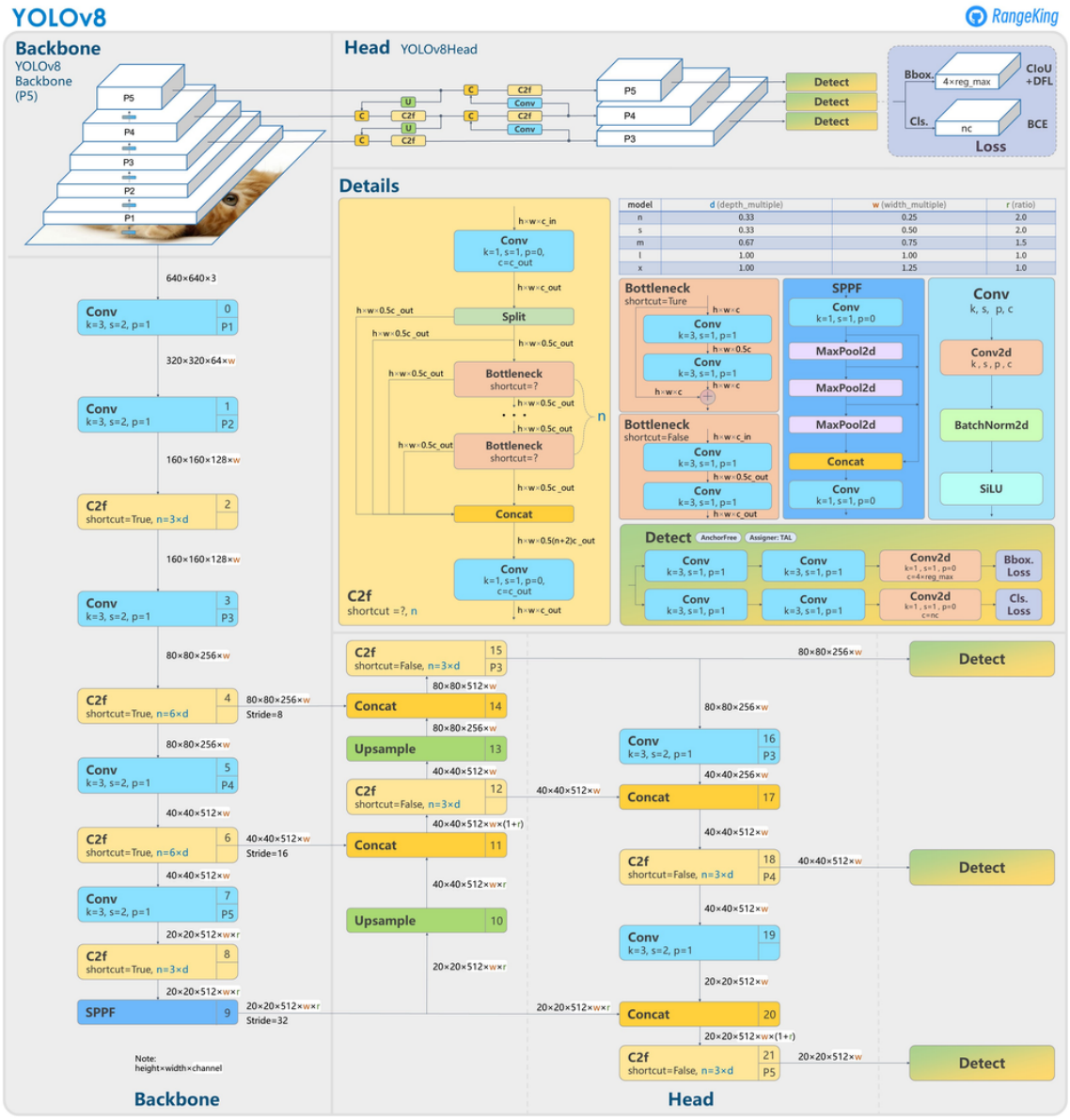


Figure 6.1: YOLOv8 Architecture

Since YOLOv8 hasn't published a paper yet, we can't learn anything about the ablation studies and direct research methodology used to create it. Having stated that, we began describing the new features of YOLOv8 by examining the repository and existing data regarding the model[33]. I recommend exploring the YOLOv8 repository and examining the provided code differential to get insights into the research methodology. This will allow you to go into the technical aspects of the study if you choose to analyze the code independently.After giving a brief overview of significant modelling modifications, we will examine the model's evaluation, which is

self-explanatory.GitHub user RangeKing created the following graphic, which thoroughly depicts the network's architecture.

6.2 Dataset

To identify the data, our detection method was applied to a total of 4446 photos. We categorize those photos into four groups: bikers wearing caps, number plate photographs, and images with and without helmets. For the training phase, we used 80% of the photos, or 3556 images, and 10%, or 445 images, for the validation and test phase.

Dataset Split	No of. images
Training dataset	3,556
Validation dataset	445
Test dataset	445

Table 6.1: Dataset Used for Detection

6.3 Result

For detection, we have employed YOLOV8. For training, we have increased the number of epochs to 50. Our model has been trained, and we have discovered that the test, validation, and training outcomes are all satisfactory.

6.3.1 Results For Training Datasets

We can see from the outcome that recall, precision, mean, and average precision increase with each epoch while loss decreases. This is something that we can observe. And the results are getting steadily better in terms of precision while getting steadily worse in terms of loss. Once more, there is no sign of noise. Therefore, it indicates the model is robust. In the outcomes of our training, we discovered that our precision metrics were 91.6 percent, our recall metrics were 90.7%, and our mAP50 metrics were 93.7%

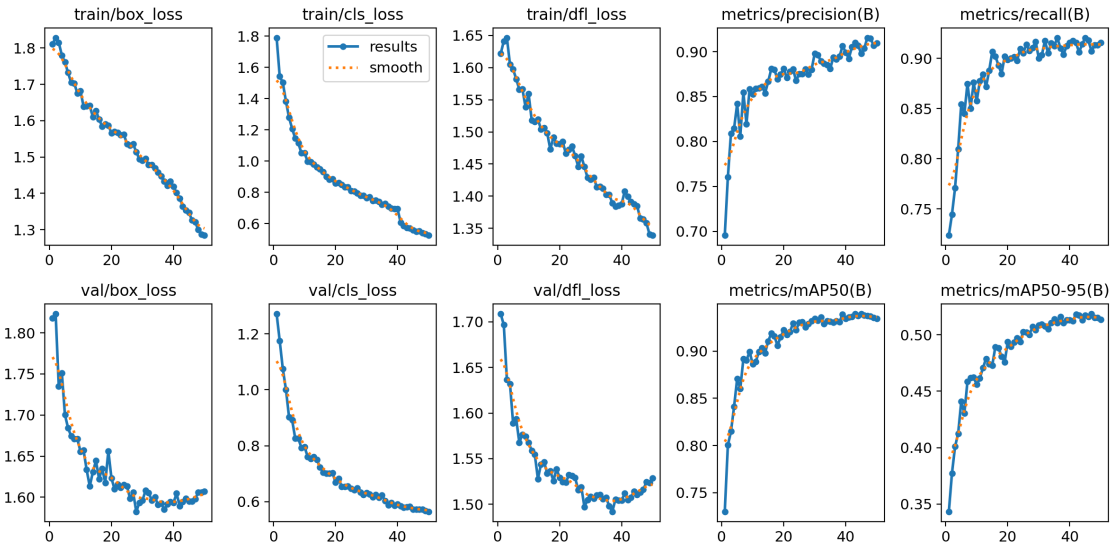


Figure 6.2: YOLOV8 Results

6.3.2 Results For Validation Datasets

F1-Confidence Curve:
The F1 score is a statistical measure that integrates the accuracy and recall of a classifier into a unified metric computed using the harmonic mean. Primarily, it is utilized to evaluate the effectiveness of two classifiers[23]. Our results show an F1 result of 91%

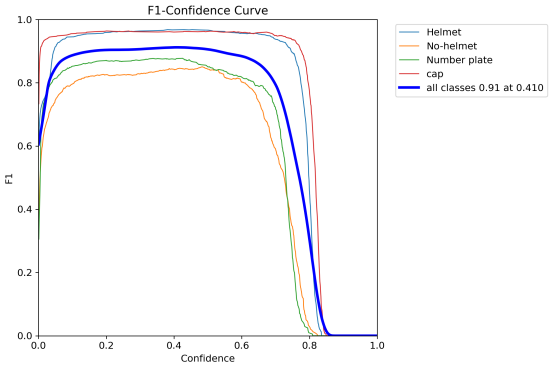


Figure 6.3: F1-Confidence Curve

Precision Curve:
precision curve demonstrates What percentage of the optimistic predictions are accurate[20]. We have found a result of 100% in precision curve accuracy.

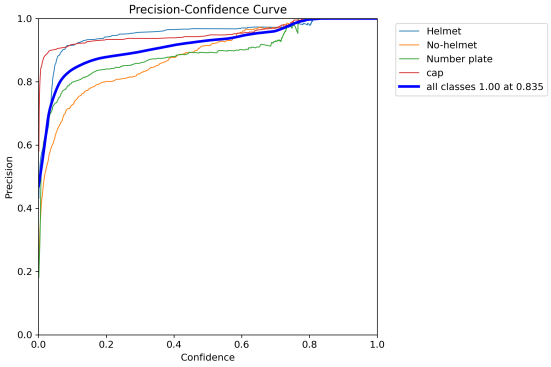


Figure 6.4: Precision Curve

Recall Curve:
The recall curve indicates Positive is predicted as a percentage of the total positive. From our results, we have seen a recall confidence curve of 98%.

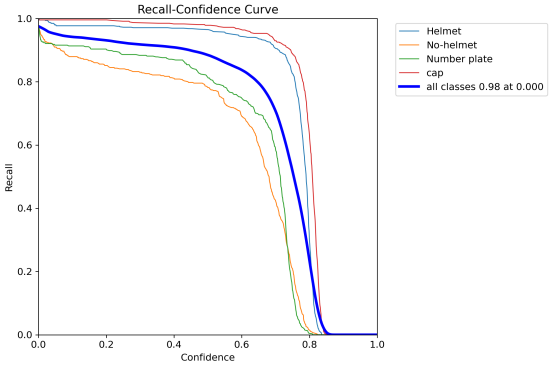


Figure 6.5: Recall Curve

Precision-Recall Curve:
Following the completion of the training process, a precision-recall curve is derived from the validation set. The precision-recall curve illustrates the balance between precision and recall at various levels. [34]. We have got a curve of 93.7% of precision-recall.

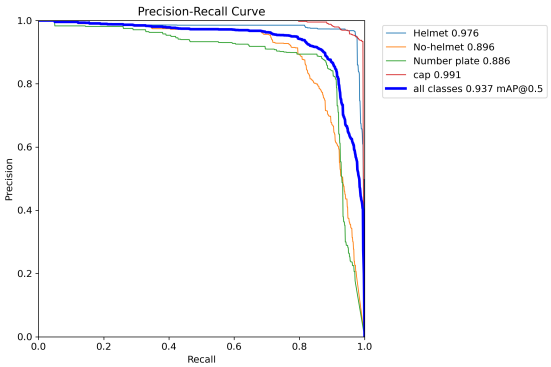


Figure 6.6: Precision-Recall Curve

6.4 Sample Results

To determine whether or not our model was accurately detecting, we examined some unseen data. We found that our model can detect those data after employing those data. As our model can accurately detect, it can be valuable in real-life applications.

Figure 6.7: Detection Visualization

Chapter 7

Conclusion and Future Work

7.1 Conclusion

As motorcycle accidents increase, the world is on the brink of an alarming situation. This number of fatalities or accidents could be decreased if people were more cautious and aware of the significance of a helmet. Hereby, by developing a model to detect people riding motorcycles without helmets or those wearing improper helmets, we advance the safety of ourselves and those around us. We are now developing a model that aims to reliably ascertain if the rider is using a helmet and evaluate the helmet's compliance with safety standards and the inclusion of supplementary safety attributes. This progress is achieved using several components from the latest and most effective image classifier and object recognition algorithm. Individuals often choose helmets that are inexpensive or provide comfort. Nevertheless, it is essential to prioritize the use of high-quality helmets. For this reason, we also endeavoured to identify the protective features of helmets. Consequently, this study aims to demonstrate through experimental analysis that our proposed model is superior to recent computer vision architecture in detecting helmets and detecting the quality of helmets by combining CNN, LSTM, and ATTENTION.

7.2 Future Work

The essential architectural model experimentation can continue after considering this work as a first step in offering guidelines for safety precautions when operating a motorcycle. A significant extension of the standard basic architecture, changes to its internal layers, and testing of various hyperparameter tuning will be part of the primary follow-up to our current effort. Moreover, we use a lab-approved dataset with certification levels for helmets rather than classifying using the technique we used to assume the safety measures of the helmets. The next step is to employ a classification system to determine whether or not any children are riding motorcycles and whether or not they are wearing safety helmets, which will further improve our risk analysis methodology. The existing dataset may be expanded, and the best outcomes for a vast amount of data can be found. This will give us insights into performance in terms of accuracy and scalability.

Bibliography

[1] C. C. C. Chung-Cheng, K. M. Y. Min-Yu, and C. H. T. Hung-Tsung, *Motorcycle Detection and Tracking System with Occlusion Segmentation*, Jun. 2007. DOI: 10.1109/wiamis.2007.60. [Online]. Available: http://dx.doi.org/10.1109/wiamis.2007.60.

[2] W. R. Rattapoom, B. N. Nannaphat, T. V. Vasan, T. C. Chainarong, and P. P. Pattanawadee, "Machine vision techniques for motorcycle safety helmet detection," *2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013)*, Nov. 2013. DOI: 10.1109/ivcnz.2013.6726989. [Online]. Available: http://dx.doi.org/10.1109/ivcnz.2013.6726989.

[3] P. Bhaskar and S. Yong, "Image processing based vehicle detection and tracking method," pp. 1–5, Jun. 2014. DOI: 10.1109/ICCOINS.2014.6868357.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] K. Dahiya, D. Singh, and C. K. Mohan, "Automatic detection of bike-riders without helmet using surveillance videos in real-time," pp. 3046–3051, 2016. DOI: 10.1109/IJCNN.2016.7727586.

[6] J. Mistry, A. K. Misraa, M. Agarwal, A. Vyas, V. M. Chudasama, and K. P. Upla, "An automatic detection of helmeted and non-helmeted motorcyclist with license plate extraction using convolutional neural network," pp. 1–6, 2017. DOI: 10.1109/IPTA.2017.8310092.

[7] C. Vishnu, D. Singh, C. K. Mohan, and S. Babu, *Detection of motorcyclists without helmet in videos using convolutional neural network*, 2017. DOI: 10.1109/IJCNN.2017.7966233.

[8] R. K. C. D. K C Dharma, C. A. Aphinya, T. V. Vasan, D. M. N. Matthew N., and E. M. Mongkol, "Helmet violation processing using deep learning," *2018 International Workshop on Advanced Image Technology (IWAIT)*, Jan. 2018. DOI: 10.1109/iwait.2018.8369734. [Online]. Available: http://dx.doi.org/10.1109/iwait.2018.8369734.

[9] B. N. Narong, P. W. Wichai, and W. P. Phonratichi, "Automatic Detector for Bikers with no Helmet using Deep Learning," *2018 22nd International Computer Science and Engineering Conference (ICSEC)*, Nov. 2018. DOI: 10.1109/icsec.2018.8712778. [Online]. Available: http://dx.doi.org/10.1109/icsec.2018.8712778.

[10] Y. Balasubramanian, K. Menaka, and S. Perumaal, "Deep learning-based helmet wear analysis of a motorcycle rider for intelligent surveillance system," *IET Intelligent Transport Systems*, vol. 13, Jul. 2019. DOI: 10.1049/iet-its.2018.5241.

[11] C. A. Rohith, S. A. Nair, P. S. Nair, S. Alphonsa, and N. P. John, *An efficient helmet detection for mvd using deep learning*, 2019. DOI: 10.1109/ICOEI.2019.8862543.

[12] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, PMLR, 2019, pp. 6105–6114.

[13] M. Tan and Q. V. Le, *Efficientnet: Rethinking model scaling for convolutional neural networks*, May 2019. [Online]. Available: https://arxiv.org/abs/1905.11946v5.

[14] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv.org*, May 2019. [Online]. Available: https://arxiv.org/abs/1905.11946v5.

[15] P. Blog. "Guide to resnet, inception v3, and squeezenet — paperspace blog." Accessed Jun 2020. (2020), [Online]. Available: https://tinyurl.com/bdewme23.

[16] H. Nagoriya, *Live helmet detection system for detecting bikers without helmet*, Sep. 2020. DOI: 10.5281/zenodo.4050483.

[17] A. Afzal, H. Umer, Z. Khan, and M. U. Khan, "Automatic helmet violation detection of motorcyclists from surveillance videos using deep learning approaches of computer vision," pp. 252–257, Apr. 2021. DOI: 10.1109/ICAI52203.2021.9445206.

[18] *Bike Accidents without Helmet — Bay Area Bicycle Law*, Aug. 2021. [Online]. Available: https://bayareabicyclelaw.com/bicycle-accidents/no-helmet/.

[19] D. Info. "Vgg-19 convolutional neural network." (Mar. 2021), [Online]. Available: https://blog.techcraft.org/vgg-19-convolutional-neural-network/.

[20] V. Jayaswal. "Performance metrics: Confusion matrix, precision, recall, and f1 score." Accessed Dec 2021. (2021), [Online]. Available: https://towardsdatascience.com/performance-metrics-v%20confusion-matrix-precision-recall-and-f1-score-a8fe076a2262.

[21] O. I. C. E. Legacy. "Inception v3 model architecture." Accessed Sep 2021. (2021), [Online]. Available: https://iq.opengenus.org/inception-v3-model-architecture/.

[22] D. I. Sec. "Vgg-19 convolutional neural network." (Mar. 2021), [Online]. Available: https://blog.techcraft.org/vgg-19-convolutional-neural-network/.

[23] Educative. "What is the f1-score?" Accessed Sep 2023. (2022), [Online]. Available: https://www.educative.io/answers/what-is-the-f1-score.

[24] S. Gonwirat, A. Choompol, and N. Wichapa, "A combined deep learning model based on the ideal distance weighting method for fake news detection," *International Journal of Data and Network Science*, vol. 6, pp. 1–9, Jan. 2022. DOI: 10.5267/j.ijdns.2022.1.003.

[25] S. Hossain, *Bangladesh counts highest death rates from bike accidents*, Jun. 2022. [Online]. Available: https://en.prothomalo.com/bangladesh/accident/bangladesh-counts-highest-death-rates-from-bike-accidents.

[26] K. M.J. Mohammad Jamil, *88% bikers in road accidents don't wear helmets: Study*, Sep. 2022. [Online]. Available: https://www.thedailystar.net/news/bangladesh/news/helmets-crucial-saving-lives-3118681.

[27] *Motorcycles*, May 2022. [Online]. Available: https://injuryfacts.nsc.org/motor-vehicle/road-users/motorcycles/.

[28] S. Rahman, *830 killed in motorcycle accidents in four months*, May 2022. [Online]. Available: https://en.prothomalo.com/bangladesh/accident/830-killed-in-motorcycle-accidents-in-four-months.

[29] *Road traffic injuries*, Jun. 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

[30] M. S. Rohith. "Keras earlystopping callback to train the neural networks perfectly." Accessed Sep 2022. (2022), [Online]. Available: https://pub.towardsai.net/keras-earlystopping-callback-to-train-the-neural-networks-perfectly-2a3f865148f7.

[31] S. S.N.N.Somit Nitasha Natu, *Bikers make up half of state road crash victims; 83% of them helmetless*, May 2022. [Online]. Available: https://timesofindia.indiatimes.com/city/mumbai/bikers-make-up-half-of-state-road-crash-victims-83-of-them-helmetless/articleshow/91824303.cms.

[32] A. Mehra. "Understanding yolov8 architecture, applications features." Accessed Jun 2023. (2023), [Online]. Available: https://www.labellerr.com/blog/understanding-yolov8-architecture-applications-features/.

[33] J. Solawetz. "What is yolov8? the ultimate guide." Accessed Jan 2023. (2023), [Online]. Available: https://blog.roboflow.com/whats-new-in-yolov8/.

[34] T. A. Team. "Precision-recall curve." Accessed Jan 2023. (2023), [Online]. Available: https://towardsai.net/p/l/precision-recall-curve.

[35] A. Vidhya, "A comprehensive guide to attention mechanism in deep learning for everyone," *Analytics Vidhya*, Jul. 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/11/comprehensive-guide-attention-mechanism-deep-learning/.

[36] *CS231N Convolutional Neural Networks for Visual Recognition*. [Online]. Available: https://cs231n.github.io/convolutional-networks/.

[37] *Deep learning*. [Online]. Available: http://www.deeplearningbook.org/.

[38] M. Gurucharan, *Top 12 Commerce Project Topics Ideas in 2023 [For Freshers]*. [Online]. Available: https://www.upgrad.com/blog/basic-cnn-architecture/.

[39] Meenu R International Journal of Innovative Science and Research Technology, *Detection of Helmetless Riders Using Faster R-CNN*. [Online]. Available: https://www.scribd.com/document/465528837/Detection-of-Helmetless-Riders-Using-Faster-R-CNN.

[40] *Motorcycle Safety — Transportation Safety — Injury Center — CDC*. [Online]. Available: https://www.cdc.gov/transportationsafety/mc/index.html.

[41] *The importance of wearing a helmet — www.personalinjury-law.com.* [Online]. Available: https://www.personalinjury-law.com/auto-accidents/helmet.

[42] *Understanding LSTM Networks – colah's blog.* [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.