DETECTING HELMETS OF THE BIKE RIDERS USING DEEP LEARNING ALGORITHMS

BY

Yousuf Mia
ID: 191-15-2360
And
Animesh Basak
ID: 191-15-2361

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

MD.SABAB ZULFIKER
Lecturer (Sr. Scale)
Department of CSE
Daffodil International University

Co-Supervised By

Saima Afrin
Lecturer
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
January 2023

APPROVAL

Our Project titled "**DETECTING HELMETS OF THE BIKE RIDERS USING DEEP LEARNING ALGORITHMS**", was submitted by Yousuf Mia ID: 191-15-2360, & Animesh Basak ID: 191-15-2361 to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 25 January 2023.

BOARD OF EXAMINERS

Chairman

Professor Dr. Touhid Bhuiyan
Professor & Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Md. Atiqur Rahman
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Shayla Sharmin 25.1.23
Shayla Sharmin
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

External Examiner

25-01-23
Dr. Dewan Md Farid
Professor
Department of Computer Science & Engineering
United International University

©Daffodil International University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Md. Sabab Zulfiker, Lecturer (Sr. Scale), Department of CSE, Daffodil International University**. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised By:

24.01.23

Md. Sabab Zulfiker
Senior Lecturer
Department of CSE
Daffodil International University

Co-Supervised by:

Saima Afrin
Lecturer
Department of CSE
Daffodil International University

Submitted by:

Yousuf Mia
ID: 191-15-2360
Department of CSE
Daffodil International University

Animesh Basak
ID: 191-15-2361
Department of CSE
Daffodil International University

©Daffodil International University II

ACKNOWLEDGEMENT

First, we would like to express our heartiest thanks and gratefulness to Almighty Allah for His divine blessing makes it possible to complete this final year research project successfully.

We are really grateful and wish our profound indebtedness to **Md. Sabab Zulfiker, Lecturer (Sr. Scale), Department of CSE Daffodil International University, Dhaka, and Cosupervisor Saima Afrin, Lecturer, Department of CSE, Daffodil International University, Dhaka**. We would like to thank the deep knowledge & keen interest of our supervisor in the field of Machine Learning to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Professor Dr. Touhid Bhuiyan, Professor and Head, of the Department of CSE,** and **Dr. S. M. Aminul Haque, Associate Professor and Associate Head, of the Department of CSE** for their kind help to finish our project and also to all of the faculty members and the staff of the CSE department of Daffodil International University for kind support and help on the technical and the administrative aspect of the study.

We would like to thank all of our course-mates at Daffodil International University, who took part in this discussion while completing the course work.

At last, we'd want to give thanks to our parents for their unending love and tolerance. They have been there our whole lives to back us up and cheer us on in all we do.

ABSTRACT

In our paper, we propose an approach for the automatic detection of helmets of bike riders in real-time. In our country, there are a large number of people who use bikes for their daily rides. Motorbikes are more popular than cars because they are less expensive to maintain, take up less parking space, and offer greater mobility and versatility in urban environments. Riding a bike is a lot of fun, but it can also be dangerous. Complete safety for bicycle riders is a primary goal of the proposed system. Despite the fact that helmet use is now legally required, many motorists continue to avoid donning them. Especially in developing nations, the mortality toll has been rising steadily over the past few years. To ensure the safety of the public, it is necessary to implement a system of helmet detection that can identify drivers who are not wearing protective headgear. For this mechanism, we use some real-time dataset about 3202. Here we collect wearing helmet 1911 and no helmet 1291 data and use algorithms like VGG16, Resnet50, MobileNet v.02, Inception V3, EfficientNet, and CNN. We got the highest accuracy from the EfficientNet about 98%. The implementation portion of the article also contains the methods used in each person's comparison statements. Model validation methods are also used in this investigation to create the best possible model for the given circumstances.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Our project aims to provide total safety for bike riders. In our country, there are a large number of people who use bikes for their daily rides. Motorbikes are more popular than cars because they are less expensive to maintain, take up less parking space, and offer greater mobility and versatility in urban environments. Riding a bike is a lot of fun, but it can also be dangerous. Complete safety for bicycle riders is a primary goal of the proposed system. Despite the fact that helmet use is now legally required, many motorists continue to avoid donning them. The annual death toll has been rising. The majority of accidents in the last few years have been bike accidents. As a result, our goal is to increase the efficiency of a traffic flow system while reducing the causes of accidents. One of the significant issues is that some motorcycle riders do not wear helmets. Every biker in Bangladesh is required by law to wear a protective headgear while operating a motorbike. A large number of motorcyclists are there who have disobeyed the rules and have failed to wear a helmet as well as other safety equipment. There are numerous bikes on the road, but the traffic cops are unable to see everyone's helmets and license plates. As a result, we came up with a solution to this problem; now it's time to move on to electronic equipment that will be connected to a road CC camera or a police box and will recognize whether bikes have helmets and proper licenses. To ensure the safety of the public, it is necessary to implement a system of helmet detection that can identify drivers who are not wearing protective headgear. For this mechanism, we use some real-time data-set about 3202 and use algorithms like VGG16, Resnet50, MobileNet v.02, Inception V3, EfficientNet, and CNN. We got the highest accuracy from the EfficientNet about 98%. As a result, in the present day, there is a critical need for a method that can reliably predict the likelihood of acquiring helmet detection. Several lives may be saved with early detection of this mechanism. In-depth study and demonstration of numerous classification and modeling techniques have been provided by authors of the several helmet detection systems now on the market, but each has its own set of limitations. Our primary focus is on devising a system that can reliably predict the occurrence of cardiac arrest with high accuracy and minimal overhead while circumventing these limitations.

1.2 Motivation

Motorbike accident is one of the most significant problems all over the world. Motorcyclists represent more than 380,000 annual deaths worldwide. Motorcycles accounted for 38.93 percent of total deaths and 42.18 percent of total accidents, the report said. A total of 524 people, including 68 women and 73 children, have died and 821 others were injured in 467 road accidents reported in Bangladesh on June 22, according to the study. If we look at death history we will see that the maximum cases of death were happening because of biker's head injuries. To protect against heavy head injury bikers should ware safety bike helmets. Helmet usage is mandated under Bangladeshi traffic rules, and this includes both drivers and passengers, although quality standards aren't addressed. An anonymous Pathao user said that the helmets offered by ridesharing services are too lightweight to be effective.

1.3 Research Questions

Before we started the investigation, we had a lot of questions. In this study, we employ deep-learning algorithms to detect bikers' helmets. A few of the most crucial research questions included:

- In what way might helmets be detected most effectively?
- How can we detect helmets?
- What methodology should we apply?
- Why do we use deep learning for detecting helmets?
- How can we get a better result?

1.4 Main Objective

This project's objective is to demonstrate a Wearing Helmet Detection System based on image processing and deep learning.

- To find out if there is a helmetless biker and if there is a bike rider.
- To make a traffic flow system more efficient.
- To reduce the causes of traffic collisions.
- To demonstrate a novel and useful helmet-wearing detection system.

1.5 Report Layout

The final report for the project follows this structure:

- Chapter One contains the introduction of our research, motivation for this study, objectives, and expected outcome of this study.
- Chapter Two includes the background of the research, related works, research summary, and difficulties encountered during this research.
- Chapter Three includes the research methodology, the proposed systems, datasets, the implementation procedure, data preprocessing, and the improved model
- Chapter Four includes Experimental Results and Discussion including experimental setup, confusion matrix, performance, and comparative analysis.
- Chapter Five includes the Conclusion and Future Scope of this project.

CHAPTER 2
BACKGROUND

2.1 Introduction

In our studies, we employed VGG16, Resnet50, MobileNet v.02, Inception V3, EfficientNet, a deep CNN architecture that we improved, and Transfer Learning. The algorithms all have a common technological denominator: they all use neural networks to process picture collections. All of these techniques are often used in the areas of image detection and classification. Every one of these methods performs well when applied to the task of processing images. Long-standing examples of convolutional neural network architecture include VGG16, Resnet50, MobileNet v.02, Inception V3, and EfficientNet. A few examples of item recognition models that our algorithm can accommodate include VGG16, which can have up to five layers, Resnet50, which can have up to five layers, MobileNet v.02, which can have up to five layers, Inception V3, which can have up to five layers, EfficientNet, which can have up to five layers, and CNN, which can have up to eight layers.

2.2 Related Works

Numerous writers have previously produced a large number of articles, studies, and research papers on the topic of life expectancy. The following are some job evaluations that are pertinent to our work:

Ramesh et al. [1] wanted to identify two-wheeler riders who were not wearing helmets. The system detects moving items like trees, pathways, buildings, and other dangerous objects using an input video of traffic on public roadways. They collected vehicle classification training data themselves. The front and back views of the vehicle were both shown in the training photographs. The classifiers were not trained on 20% of the total amount of data that was obtained. Instead, it was utilized to evaluate the helmet detection algorithms. In image recognition, a deep neural network is predicted to outperform a random forest, but this was not the case due to a lack of data. When there were several automobiles present, the system was inoperable. Their primary concern was evaluating how various machine-learning algorithms performed in that situation. Maharsh et al. [2] project's primary goal was to reduce accidents by employing techniques including fall detection, helmet authentication, and alcohol detection. If a person was not wearing a helmet, the license plate should be extracted using the background subtraction approach, and the vehicle's license plate should be obtained using the optical recognition method. An automated system that connects to a GPS unit to obtain information about time and position. The system extracted the license plate from the camera frame and search the central database if it detects any falls. In an emergency, the system would alert surrounding hospitals, family members, and law enforcement. Mario et al. [3] proposed an algorithm that used image-processing methods to identify motorcycles and determine whether riders were wearing helmets. Motorcycle categorization accuracy was 97.14%, while helmet detection accuracy was 85.29%. For the purpose of detecting motorcyclists in the path of moving vehicles, traffic authorities need a reliable, adaptable, and affordable method. Due to the significant variations in the shape, color, and size of both objects, it was particularly challenging to detect motorcycles and safety helmets. In Bogot, Colombia, pictures were captured at a frontal angle of the moving vehicles. Moment particle analysis was used to threshold images after background subtraction, ROI selection, contrast improvement, and ROI thresholding. The morphology of the objects showed in the photograph enables the classification of such objects used artificial intelligence models. Below were the outcomes of the binary classifier for motorcycle detection that used boosting trees. The paper's review indicates that the best motorcycle classifier has an accuracy rate of 94.9%. The suggested classifier achieves a precision that was pretty comparable. Rohith et al [4] intend to construct a real-time classifier. Using video footage captured by a webcam, they created a model to recognize two-wheeler vehicles using transfer learning and fine-tuning techniques. To found people riding two-wheelers, they employed real-time object detection (Caffe). A bounding box was built around a two-wheeled individual if the model identifies their existence. After being clipped from the current frame, this region was then sent to an image classifier. The InceptionV3 model was adjusted to meet specifications and trained on the fresh dataset. If the image was designated as having no headgear, it was saved to the directory for further use. The MVD might employ these specifics to penalize violators. The scores for the two categories were obtained using transfer learning strategies. They would like to extract pictures of people operating two-wheelers wearing helmets and without them. The Caffe model was the one employed for categorization and extraction. For this specific binary classification test, the suggested model has an accuracy score of 86%.Apoorva et al. [5] presented a framework to aid traffic police in identifying offenders under unusual environmental situations, such as extreme heat. A 10-fold cross-approval analysis was used to determine the system's final verdict on the entire picture. A number of components of the detection system were built using OpenCV libraries. To assess the data set containing information about motorcyclists and whether or not they were wearing helmets, YOLOv3 attempted to perform an image grouping. The technique used a 10-fold cross-approval analysis to determine the overall quality of the photograph. Narong et al. [6] used various types of algorithms where MobileNets was the winner, detecting 421 (85.40%) valid biker classes out of a total of 493 video pictures. Only 416 (84.38%) proper images were recognized by Inception V3. Both algorithms effectively detect all 493 bikers in the video datasets, leaving only 0 unidentified. The area of a motorcycle with a cyclist who did not wear a helmet was designated as "Biker with no helmet." Meenu et al. [7] demonstrated by using CCTV Videos how a riders' head protection status may be determined. Open-ALPR detected the vehicle's license plate and issued an alert to the nearest police station. This project yielded a 92% accurate outcome. The effectiveness of a CCTV camera system is a key factor. Kunal et al. [8] suggested an approach that takes place in real-time and requires minimal processing resources. A total of 40 persons and 13 vehicles can be seen in the footage. For a wide variety of feature sets and kernels, the HOG kernel consistently produces the best results. The detection accuracy of the experiments was 93.80% when applied to real-world surveillance data. C. Vishnu et al. [9] used adaptive background subtraction to find moving things in the proposed system. Then, these moving things were put into a CNN classifier, which put them into two groups: motorcycles and everything else. On the IITH Helmet 1 dataset, the proposed CNN method does better than the existing HOG-SVM method by 0.36% and by 9.97% on the iITH Helmet 2 dataset. The accuracy is 95.24%, and there are less than 0.5% false alarms. Kavyashree et al. [10] showed by capturing the video feed, the system hoped to perform continuous helmet detection. Every vehicle on the road inspected, and a database of offenders will be created in real-time. A video camera installed on the road to provide input to the system. Feature extraction and feature matching used to label objects on the streets. In order to ensure that no helmets were overlooked throughout the training process, a CNN model was given information on all of the helmets that are currently on the market. Dasgupta et al. [11] suggested employing the SSD or YOLOv3 technique in order to locate the area occupied by the motorcycle, after which the upper portion of the image would be extracted, and the classification algorithm would be used in order to distinguish between helmets and other head coverings. In a similar vein, the categorization system will be rendered useless whenever there is more than one person riding on the motorcycle. Khan et al. [12] suggested using the YOLOv3 algorithm to determine whether or not a motorcyclist was wearing a helmet. However, the detection of motorcycles is not reported by the algorithm. Khan then proposed using the overlapping area of the bounding box between the motorcycle and the person to determine who was riding the motorcycle. Finally, the YOLOv3 algorithm was applied in order to determine whether or not the motorcyclist was wearing a helmet. Nevertheless, from the perspective of monitoring traffic, motorcyclists and motorcycles share a lot of similarities, which means that distinguishing motorcyclists from bikes is not necessary. Waranusat et al. [13] demonstrated a system capable of detecting moving objects by positioning a k-NN classifier above the head of a motorbike to determine the sort of helmet being worn by the rider. The amount of accuracy that could be reached with these models, which were suggested based on statistical information from images, was limited; hence, they were not generally adopted.

2.3 Research Summary

The above demonstrates that several writers have explored this area of study in the past. A 95% accuracy threshold was set as sufficient in their study; we raise it to 98%. Most of the authors used the CNN algorithm we also use that algorithm and also we use the latest version of CNN that’s why we get more accurate results. For detecting helmets on bikers A higher level of precision allows for more reliable predictions than does a lower level. The data may have had some errors or missing values and another reason can happen that they can’t use the proper algorithm or proper way that’s why they can’t get a better result. But we check our dataset and preprocess that data then we apply deep learning and also use some latest algorithms. In this study, we employed all of the models to categorize and identify the helmets worn by motorcyclists in real time, producing both a visual and an audible output. As a consequence, we achieved a more consistent level of identification accuracy while also reducing the amount of processing time required.

2.4 Scope of the Problem

The traffic authorities need a reliable, adaptable, and inexpensive technology for detecting motorbikes within the vehicle flow and for detecting the wearing of safety helmets by riders so that they may begin to solve the aforementioned road safety issue. Intelligent transportation systems (ITS) were developed for this reason; nonetheless, accurate detection in our nation requires the widespread deployment of surveillance cameras. These innovative programs integrate electrical components with other technologies such as communications networks, computers, and sensors. To improve public and private traffic safety, efficiency, and comfort, ITS integrates cars, people, and road information and supplies this data in real-time. A high-quality, high-efficiency gadget is essential for managing our system since doing so requires a complex procedure and the skills of trained staff. This was a significant challenge as we gathered information since we needed to take several photographs from a variety of perspectives.

2.5 Challenges

To make one of these devices for people who can't see, we want to make a system that is reliable, accurate, and easy to use. Some of the difficult situations that could come up in the development of this research based on the whole system are:

- Data transfer: It is difficult to preserve a large amount of data from a mobile device, as seen by the photos we've shot with our phone. We also planned to use a USB cord to transfer the photos from our phone to the computer. We had so many different pictures on our phones that it took forever to swap them all across.
- Time Complexity: Because of the alleged device is a real-time detection system, we want to identify it as soon as possible to reduce the detection's processing time.
- Processing data in preparation for intensive analysis.

- Hardware obstacles: It's possible that there were delays in the background of the acquired photographs due to limitations in our processing gear. This can be easily fixed by including more powerful hardware into our design, such as a more powerful central processing unit, hard disk drive, or graphics processing unit.

- Collecting enough data for the neural networks to learn.
- Data collection: Collecting our Dataset has been fraught with difficulties. We had to take pictures of each person with and without a helmet separately, from a variety of different perspectives. We wasted a lot of time on this.
- We need many camera on every road for the detection in our country that is very difficult.

CHAPTER 3
RESEARCH METHODOLOGY

3.1 Introduction

The intent of this initiative is to identify bike riders who are wearing protective headgear. To find a solution, we used a wide variety of neural network architectures, including VGG16, Resnet50, MobileNet v.02, Inception V3, EfficientNet, and deep CNN. We've built our own versions of CNN, Inception V3, EfficientNet, and Resnet50 in the Tensorflow framework. We created our own proprietary, pretrained algorithms and utilized a raw collection of photos for our Training dataset.

3.2 Proposed System

In order to do this, we have utilized a combination of custom CNN, Transfer learning, and pre-trained models like VGG16, Resnet50, MobileNet v.02, Inception V3, and EfficientNet. The last step was to evaluate the six models we had developed. We started with a bespoke CNN model that had two filters and an activation layer configuration that included MaxPooling2D, Conv2D, MaxPooling1, Conv2D, MaxPooling2, Flatten, and Dense1. We then trained a pre-trained model using the extracted characteristics from the photos (using models such as VGG16, Resnet50, MobileNet v.02, Inception V3, and EfficientNet) and analyzed its results. As a final step, we developed a trainable model based on our proprietary CNN and a feature extractor based on our pre-trained VGG16, Resnet50, MobileNet v.02, Inception V3, and EfficientNet model.

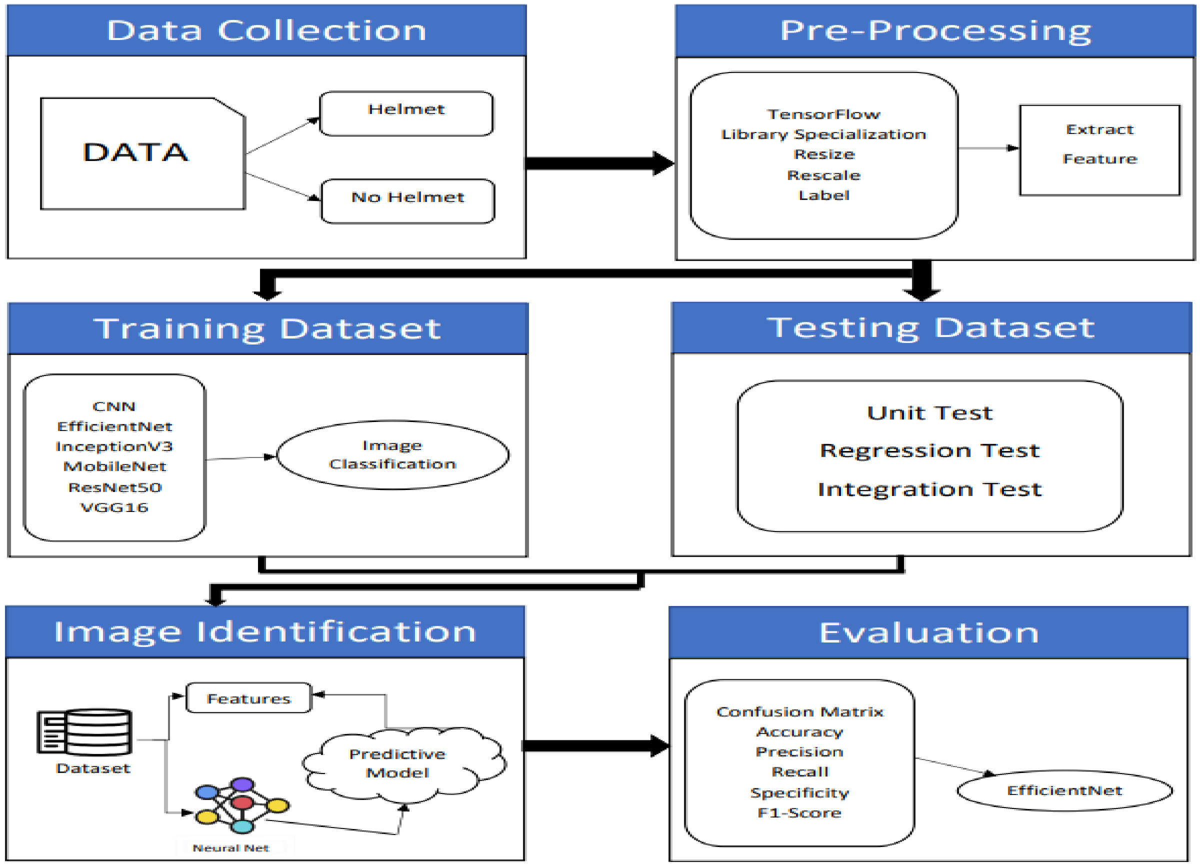

Figure 1: Working method of the proposed methodology.

3.3 Dataset

The first stage of putting our concept into action was data collection. An picture database served as the basis for our model. In other words, we were using our own mobile devices to gather raw data, takas, and photographs. Bicyclists were gathered from various routes, including those who were and were not wearing helmets. Once the picture data, in its raw form, has been captured. So that our model could quickly load new data, we decreased the pixel size. There were a total of 3202 photographs in the collection, all of which were at least 224 pixels on the longest side. The dataset included two groups.

3.4 Implementation Procedure

3.4.1 Data preprocessing

The importance of thorough pre-processing of data cannot be overstated. Because photos of varying sizes are included in our dataset. They won't be able to be trained properly if the photos are of various sizes. To prevent this issue, we have shrunk and reshaped the photos such that they are all 50 pixels on the longest side. Moreover, we employed the CNN, VGG16, Resnet50, MobileNet v.02, Inception V3, and the EfficientNet model to train our dataset, all of which need a large number of pictures. To train our dataset, we had access to sufficient data. We've utilized 3 channel color photos and tagged our image in the data pre-processing phase as well (RGB). We were able to train the dataset with much higher-quality photos and improved performance thanks to this.

3.4.2 Convolutional Neural Networks (CNNs)

What we mean by "convolutional neural networks" is a subset of machine learning that consists of networks that learn to "see" images. It's a kind of artificial neural network, one of several that may be used with a wide variety of tasks and information sets. A The Convolutional Neural Network (CNN) is a network architecture for deep learning algorithms often used in image recognition and pixel data processing. To sum up, convolutional neural networks (CNNs) are the top choice. In this way, they excel at computer vision (CV) tasks and applications like autonomous vehicles and face recognition that need object identification.

We implement it on code-

1. Prepare the training and testing data.
2. Build the CNN layers using the Tensorflow library.
3. Select the Optimizer.
4. Train the network and save the checkpoints.
5. Finally, we test the model.

Here we used 8 layers. They are-

1. Max_pooling2d
2. Conv2d_1
3. Max_pooling2d_1
4. Conv2d_2
5. Max_pooling2d_2
6. Flatten
7. Dense
8. dense_1

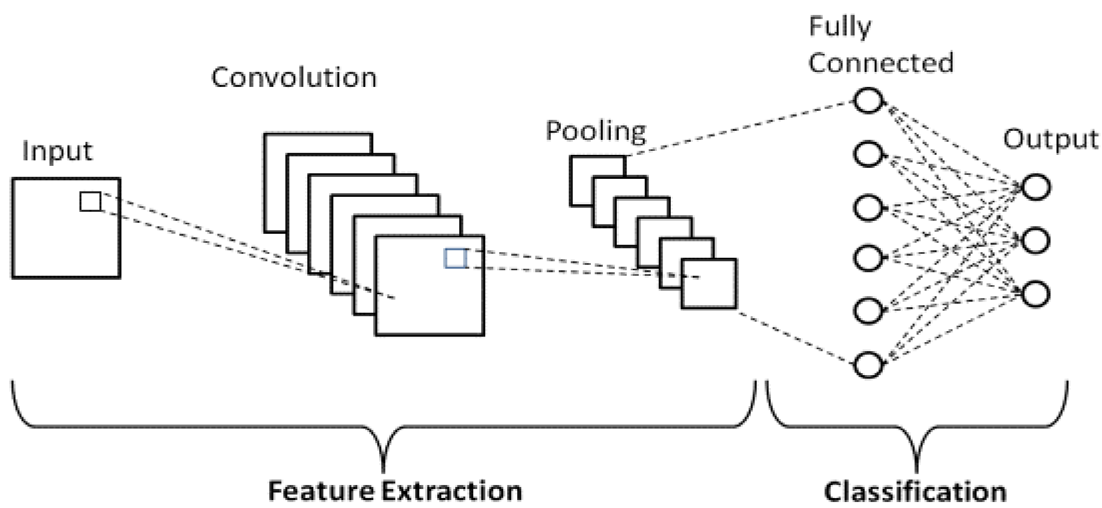

Figure 2: Working method of CNN

3.4.3 Pre-Trained Model (EfficientNet)

EfficientNet is a method for designing and efficiently scaling convolutional neural networks, whereby a compound coefficient is used to uniformly scale the depth, breadth, and resolution of the network. The EfficientNet scaling method uniformly grows network width, depth, and resolution according to a preset set of scaling coefficients, in contrast to the existing practice of randomly scaling these components. Extending the network depth by, breadth by, and image size by, where are constant coefficients acquired from a modest grid search on the original little model, is one simple way to use many times the computer resources. By using a compound coefficient, EfficientNet is able to expand network width, depth, and resolution proportionally. The theory behind the compound scaling method is that a bigger input picture will need a network with more layers to increase the receptive field and more channels to detect the finer details in the image.

We implement it on our code-

1. The Custom Classification Task.
2. Import EfficientNet Dependencies.
3. Import EfficientNet and Choose EfficientNet Model.
4. Creating a Custom EfficientNet Training Dataset.
5. Creating a Custom EfficientNet Training Job.

Here we used 5 layers. They are-

1. Flatten
2. Dropout
3. Dense
4. Dense_1
5. Dense_2

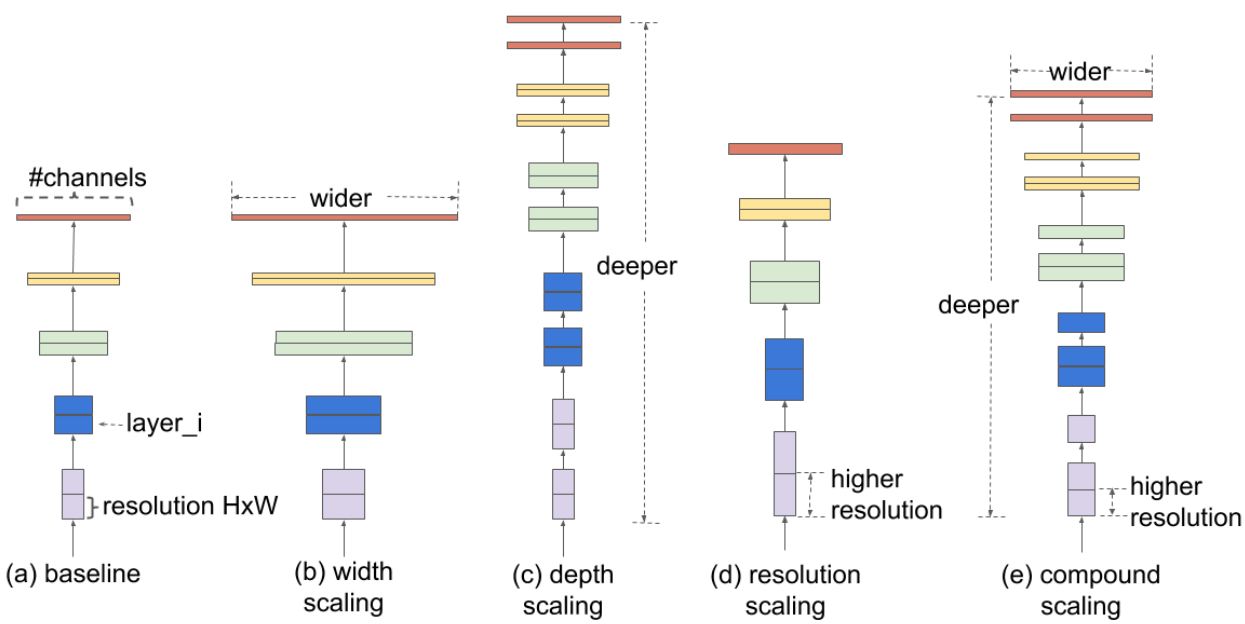

Figure 3: Working method of EfficientNet

3.4.4 Pre-Trained Model (InceptionV3)

Convolutional neural networks provide the basis of the Inception V3, a deep learning model used for the classification of images. When numerous deep layers of convolutions were utilized in a model, the data was overfitted.

We implement it on our code-

1. Installing Monk.
2. The Dataset.
3. Training the models.
4. Results of Training.
5. Deploy the models through API.
6. Running the API.
7. Conclusion.

Here we used 5 layers. They are-

1. Flatten_1
2. Dropout_1
3. Dense_3
4. Dense_4
5. dense_5

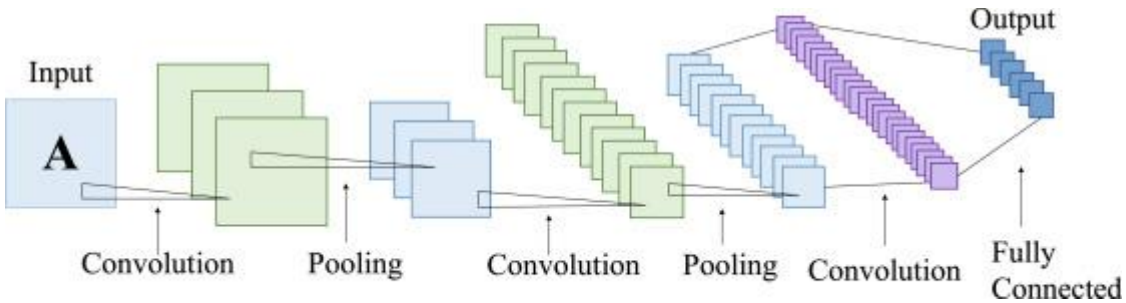

Figure 4: InceptionV3 model's architecture.

3.4.5 Pre-Trained Model (MobileNet)

MobileNet is a CNN-based image classification and mobile vision architectural paradigm. There are several models, but MobileNet stands out since it uses relatively minimal computing power to carry out or apply transfer learning. This makes it perfect for mobile devices, embedded systems, and computers without compromising a lot of accuracy since they lack a GPU or have poor processing performance. It works well with web browsers since they have better computational, visual processing, and storage capabilities.

We implement it on our code-

1. Getting the dataset
2. Transfer Learning With MobileNet
3. Importing Tensorflow and necessary libraries
4. Preparing the training data
5. Creating training and validation sets
6. Preprocessing the Image data
7. Initializing the base model
8. Adding Extra layers to Pre-trained Model
9. Compiling the model
10. Training the model
11. Visualizing the training and Validation performance

Here we used 5 layers. They are-

1. flatten_8
2. Dropout_8
3. dense_24
4. Dense_25
5. dense_26

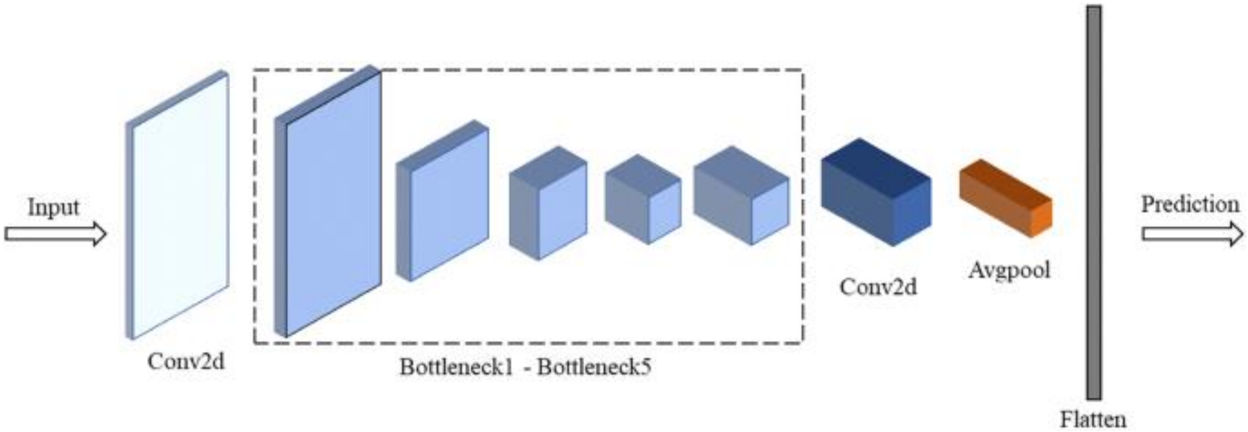

Figure 5: MobileNet model architecture.

3.4.6 Pre-Trained Model (Resnet50)

Residual Network (ResNet) is a deep learning model used in computer vision. It has an architecture made up of hundreds or thousands of convolutional layers and is a convolutional neural network (CNN). Previous CNN designs performed poorly because they couldn't scale to a large number of layers. But when additional layers were added, the "vanishing gradient" issue cropped up for the researchers.

We implement it on our code-

1. Import the required libraries and dataset for the problem
2. Partition and Visualize Data
3. Import the pre-trained machine learning model
4. Train and Evaluate Model
5. The model and classify images

Here we used 5 layers. They are-

1. Flatten
2. Dropout
3. Dense

4. Dense_1
5. dense_2

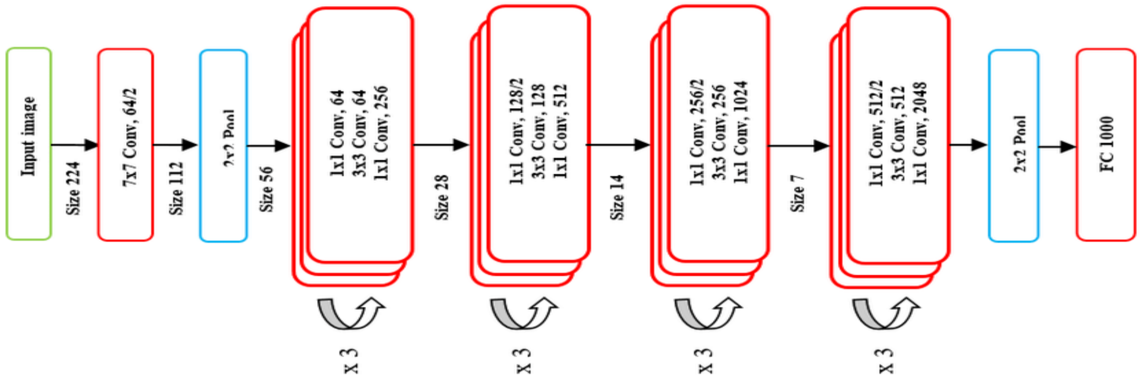

Figure 6: Resnet50 model Workflow.

3.4.7 Pre-Trained Model (VGG16)

A 16-layer deep convolutional neural network is represented by the VGG-16. The ImageNet image collection contains over one million different pictures, which may be used to load a network that has already been pretrained. The pretrained network is capable of recognizing one thousand unique item categories inside pictures. These categories include animals, keyboards and mouse, pens, and other writing implements. As a direct consequence of this, the network has learned complex feature representations for a large variety of different pictures.

We implement it on our code-

1. Feature Extraction Approach
2. Fine-Tuning Approach
3. Preparing the training and testing data
4. Pre-trained Layers for Feature Extraction

Here we used 5 layers. They are-

1. Flatten_1
2. Dropout_1
3. Dense_3

4. Dense_4
5. dense_5

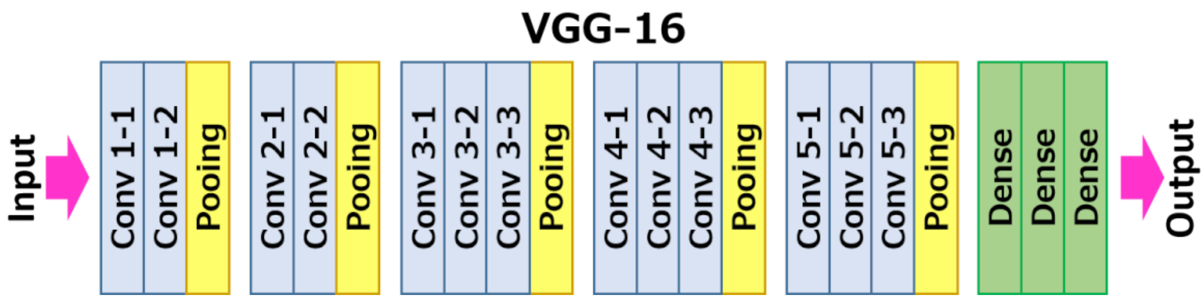

Figure 7: VGG16 model Workflow.

3.4.8 Model Tuning.

Tuning entails experimenting with different values for a set of hyperparameters. We have raised the batch size to 32 for all of the models in our project. To avoid any potential confusion in our dataset, we did not modify any of the classes. When we increase the epoch to 30, our models significantly improve.

3.4.9 Model Training

Our study highlights the significance of training for successful model implementation. Without further adjustment of the pre-trained VGG16, Resnet50, MobileNet v.02, Inception V3, EfficientNet model, we have implemented the CNN model using the Keras and Tensorflow framework. For the training in question, we utilized a batch size of 32 and 30 training iterations. We had three different courses to choose from. Training the model involves adjusting several hyperparameters from a configuration file, such as the learning rate, batch size, image filtering, and number of epochs. The model's training parameters and structure are stored so that they may be evaluated and improved upon in the future. Within the training loop, we load the training and validation datasets. In order to train the model, we employ the Adam Optimizer with Cross-Entropy Loss. Models are assessed at each timepoint in the validation set, with the most accurate model being retained for further testing. When the training is done, the plot of error and loss overtraining, as well as the training and validation error and loss, are preserved.

3.4.10 Experimental Environment

The test was run on a PC outfitted with a 2.90GHz 4.10GHz Intel(R) Core(TM) i5-9400U CPU, a 4GB NVIDIA HD Graphics 630 GPU, 12 GB of RAM, and Windows 10 as the operating system. This model is run using the free and open-source Google Colab Notebook. Tensorflow and Keras are used to create the model.

CHAPTER 4
RESULTS AND DISCUSSIONS

4.1 Confusion Matrix

Research into methods of identifying bikers who are wearing helmets has gained significant traction in recent years. To maximize our model's performance, we employed it as the final layer in an EfficientNet model that was trained using several image classes. Our model's efficacy was measured using confusion matrix.

We have displayed the confusion matrix of the custom CNN model that was trained with our provided images in Figure M1. The number of successfully identified photos is represented along the diagonal.

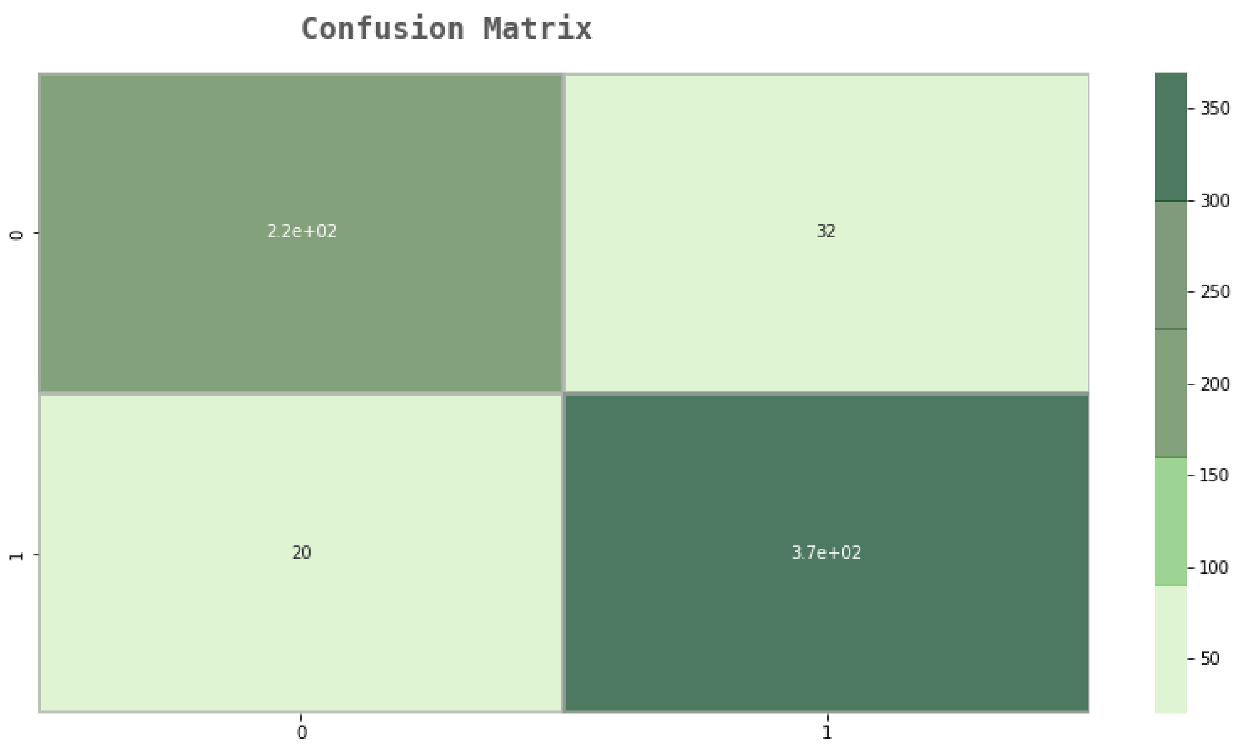

Figure M1: Confusion matrix of CNN

We have displayed the confusion matrix of the VGG16 model that was trained with our provided images in Figure M2. The number of successfully identified photos is represented along the diagonal.

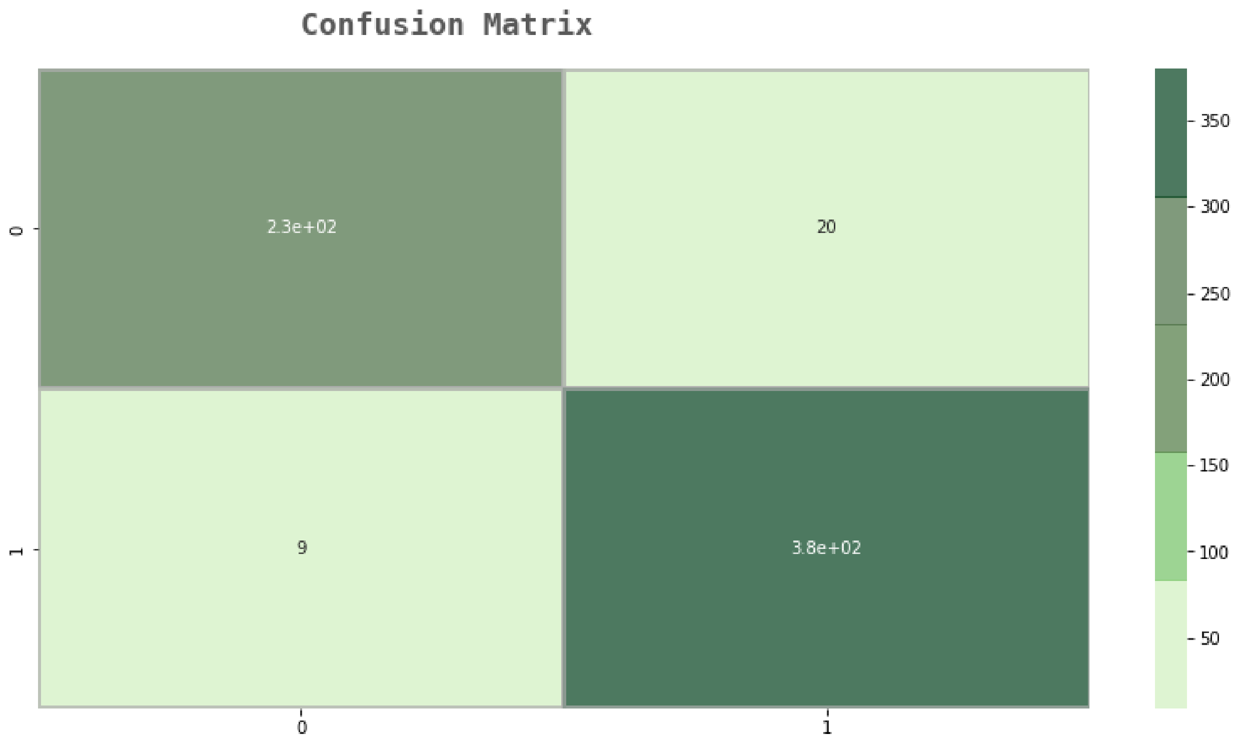

Figure M2: Confusion matrix of VGG16

We have displayed the confusion matrix of the EfficientNet model that was trained with our provided images in Figure M3. The number of successfully identified photos is represented along the diagonal.

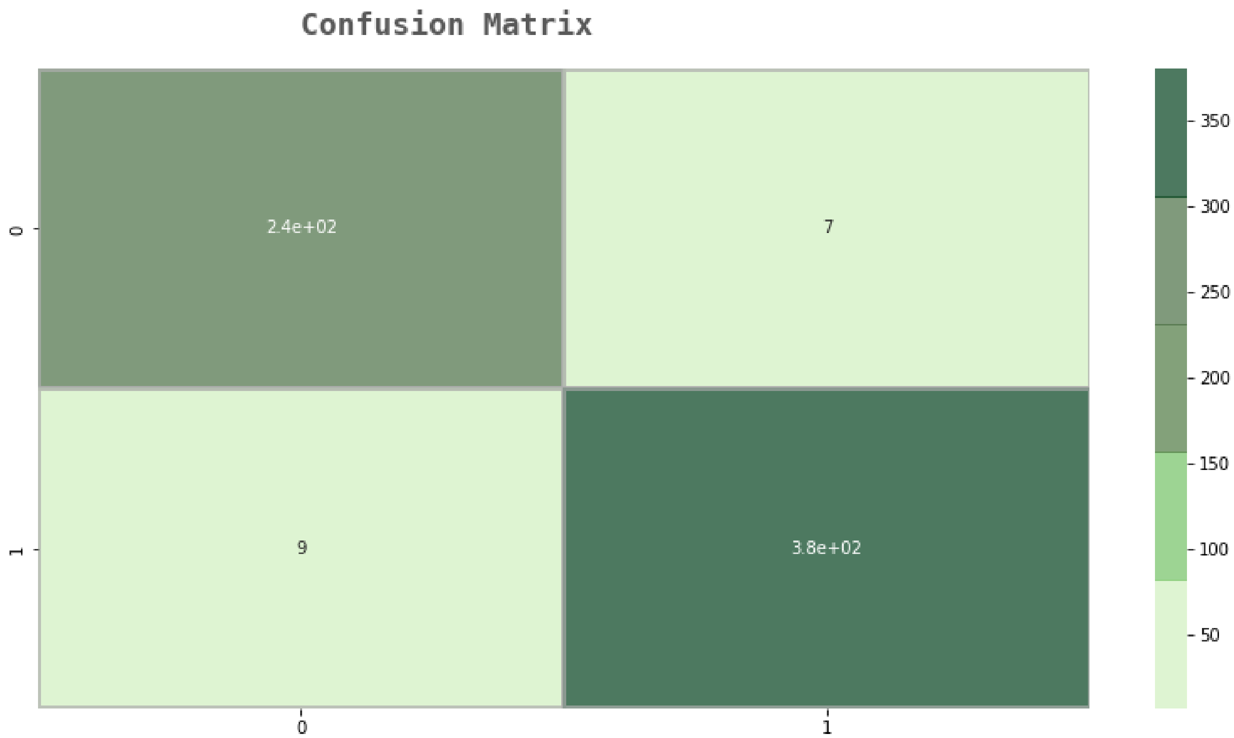

Figure M3: Confusion matrix of EfficientNet

We have displayed the confusion matrix of the InceptionV3 model that was trained with our provided images in Figure M4. The number of successfully identified photos is represented along the diagonal.

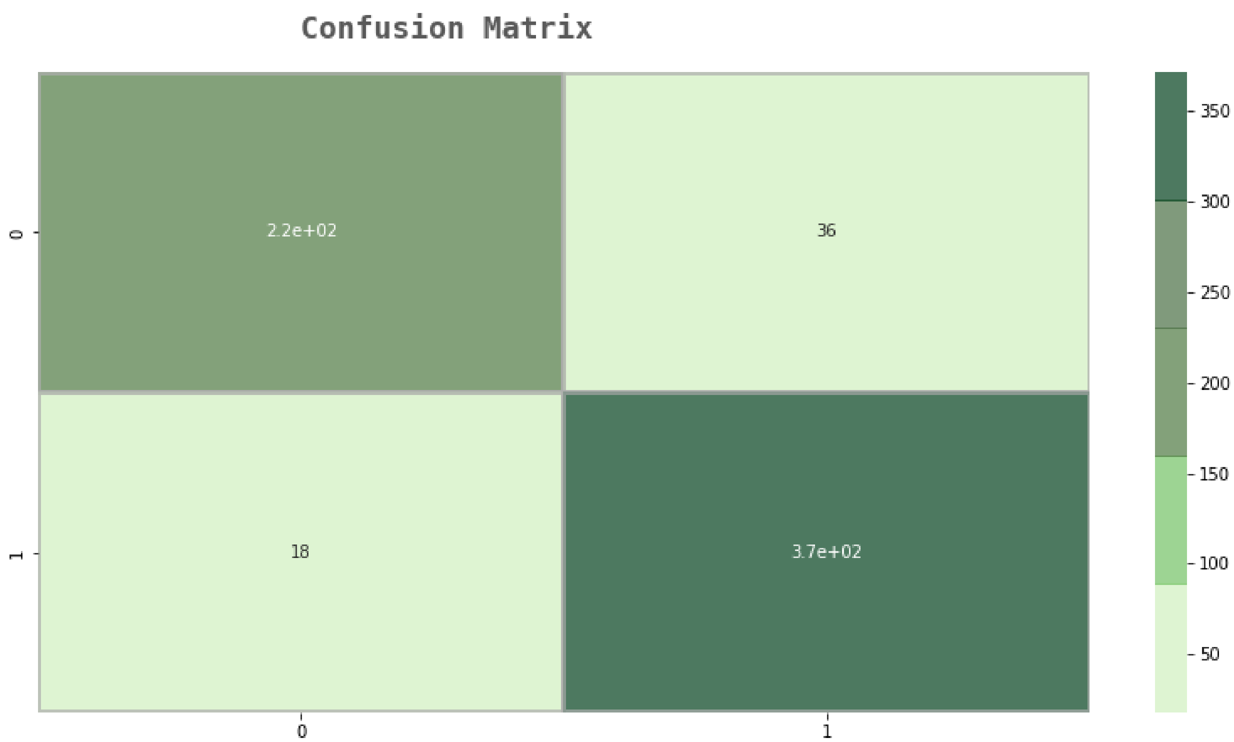

Figure M4: Confusion matrix of InceptionV3

We have displayed the confusion matrix of the MOBILENET model that was trained with our provided images in Figure M5. The number of successfully identified photos is represented along the diagonal.

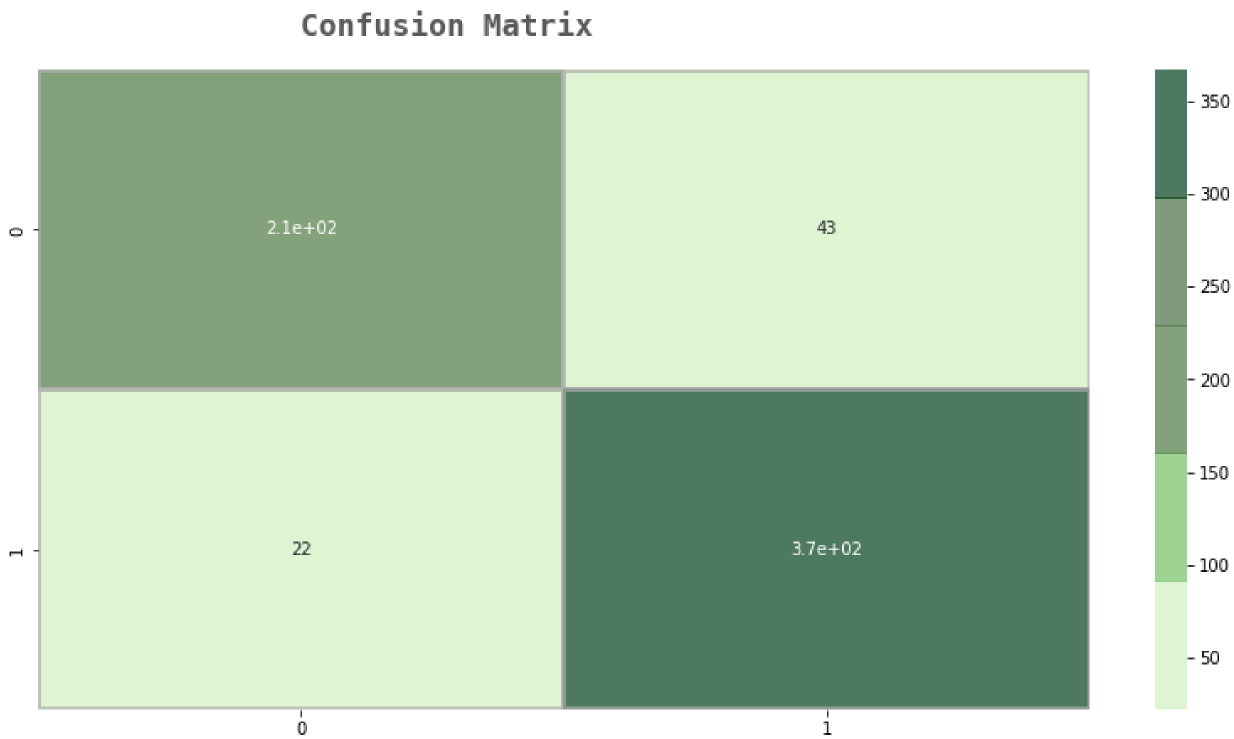

Figure M5: Confusion matrix of MOBILENET

We have displayed the confusion matrix of the Resnet50 model that was trained with our provided images in Figure M6. The number of successfully identified photos is represented along the diagonal.

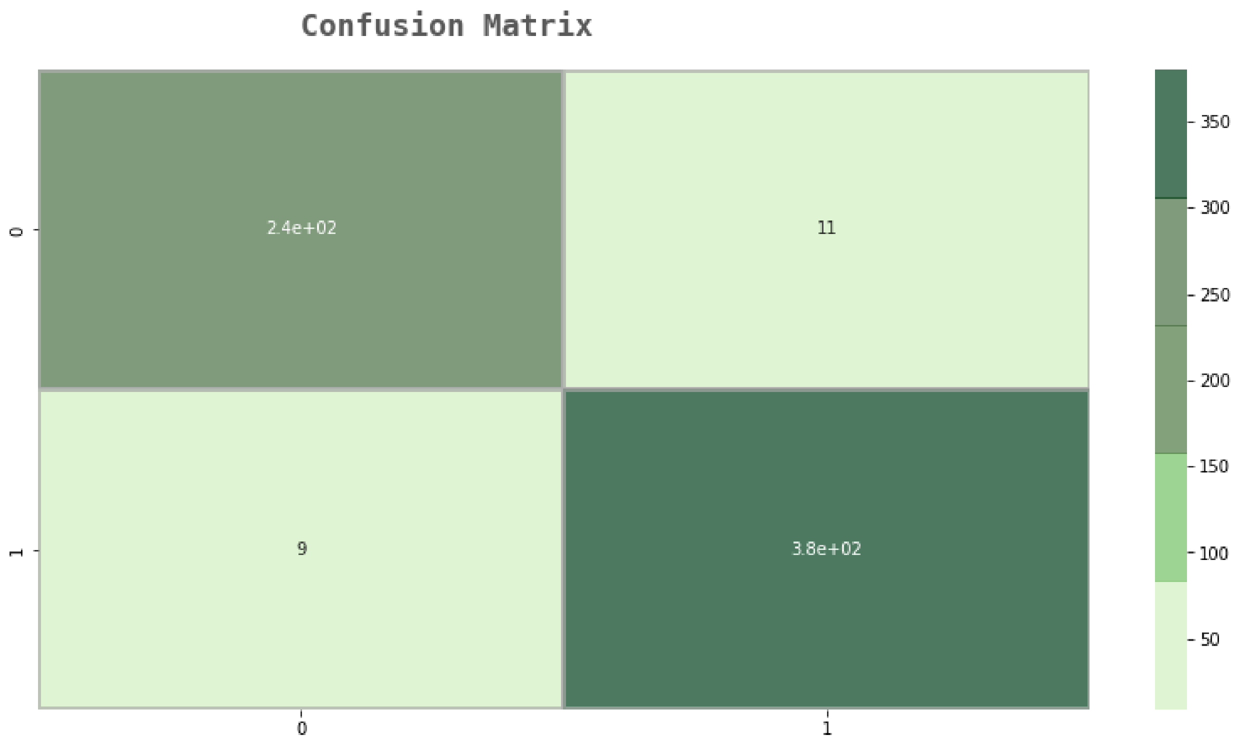

Figure M6: Confusion matrix of Resnet50

4.2 Performance Matrix

Table 1: Performance after training with CNN

CNN			
Accuracy	TP+TN / TP+TN+FP+FN	369+220 / 369+220+32+20	92%
Precision	TP / TP+FP	369 / 369+32	92%
Recall	TP / TP+FN	369 / 369+20	95%
Specificity	TN / TN+FP	220 / 220+20	92%
F1-Score	2TP / 2TP+FP+FN	2*369 / 2*369+32+20	93%

Table 1 shows that the model's validation loss was somewhat high, despite the fact that its accuracy was adequate. Consequently, there are still ways in which our model could be enhanced.

Table 2: Performance after training with VGG16

VGG16			
Accuray	TP+TN / TP+TN+FP+FN	380+232 / 380+232+20+9	95%
Precision	TP / TP+FP	380 / 380+20	95%
Recall	TP / TP+FN	380 / 380+9	98%
Specificity	TN / TN+FP	232 / 232+20	92%
F1-Score	2TP / 2TP+FP+FN	2*380 / 2*380+20+9	96%

As you can see in Table 2, we reduced our training loss and increased our accuracy as a function of various other performance indicators.

Table 3: Performance of EfficientNet

EfficientNet			
Accuracy	TP+TN / TP+TN+FP+FN	380+245 / 380+245+7+9	98%
Precision	TP / TP+FP	380 / 380+7	98%
Recall	TP / TP+FN	380 / 380+9	98%
Specificity	TN / TN+FP	245 / 245+7	97%
F1-Score	2TP / 2TP+FP+FN	2*380 / 2*380+7+9	98%

As you can see in Table 3, we reduced our training loss and increased our accuracy as a function of various other performance indicators.

Table 4: Performance after training with InceptionV3

InceptionV3			
Accuracy	TP+TN / TP+TN+FP+FN	371+216 / 371+216+36+18	92%
Precision	TP / TP+FP	371 / 371+36	91%
Recall	TP / TP+FN	371 / 371+18	95%
Specificity	TN / TN+FP	216 / 216+36	86%
F1-Score	2TP / 2TP+FP+FN	2*371 / 2*371+36+18	93%

As you can see in Table 4, we reduced our training loss and increased our accuracy as a function of various other performance indicators.

Table 5: Performance after training with MobileNet

MobileNet			
Accuracy	TP+TN / TP+TN+FP+FN	367+209 / 367+209+43+22	90%
Precision	TP / TP+FP	367 / 367+43	90%
Recall	TP / TP+FN	367 / 367+22	94%
Specificity	TN / TN+FP	209 / 209+43	83%
F1-Score	2TP / 2TP+FP+FN	2*367 / 2*367+43+22	92%

As you can see in Table 5, we reduced our training loss and increased our accuracy as a function of various other performance indicators.

Table 6: Performance after training with ResNet50

ResNet50			
Accuracy	TP+TN / TP+TN+FP+FN	380+241 / 380+241+11+9	97%
Precision	TP / TP+FP	380 / 380+11	97%
Recall	TP / TP+FN	380 / 380+9	98%
Specificity	TN / TN+FP	241 / 241+11	96%
F1-Score	2TP / 2TP+FP+FN	2*380 / 2*380+11+9	97%

As you can see in Table 6, we reduced our training loss and increased our accuracy as a function of various other performance indicators.

4.3 Result

Table 7: Performance of all algorithms

Algorithm name	Accuracy	Precision	Recall	Specificity	F1-Score
CNN	92%	92%	95%	92%	93%
EfficientNet	98%	98%	98%	97%	98%
InceptionV3	92%	91%	95%	86%	93%
MobileNet	90%	90%	94%	83%	92%
Resnet50	97%	97%	98%	96%	97%
VGG16	95%	95%	98%	92%	96%

All algorithms and their respective accuracies, precisions, recalls, and specificities, as well as f1-scores, are displayed in above table. Here we can easily see that CNN provide 92%, InceptionV3 provide 92%, MobileNet provide 90%, VGG16 provide 95%, Resnet50 provide 97%, and EfficientNet give the highest value 98% and also it is our best accuracy.

4.4 Learning curve

A learning curve, in its simplest form, is a graph showing the progression of a model's performance with respect to either time or experience. It is common practice in machine learning to use learning curves as a diagnostic tool for algorithms that make incremental improvements after being exposed to a training dataset. Both train and validation accuracy, as well as train and validation loss, are depicted in the provided learning curve figure.

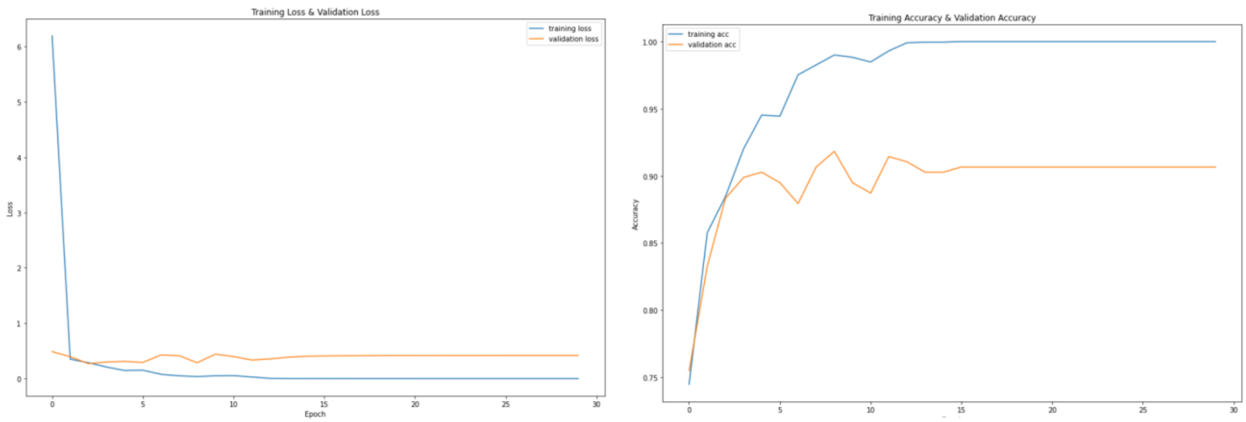

Figure 8: The learning curve of the custom CNN model.

In figure 8 shows the training and validation loss as well as the training and validation accuracy of CNN. Both figures are colored blue to represent training accuracy and training loss. On the other hand, orange represents validation accuracy and validation loss. Initially, we see that the training loss was approximately 6.1878; however, as epochs are increased, the amount of loss decreases. After a few epochs, the amount of loss is stable and close to 0.0436. Initially, we see that training accuracy was approximately 0.7448. After that, as epochs are increased, so does the amount of accuracy. After a few epochs, the amount of accuracy is stable and close to 0.9906. On the other hand, validation loss is almost identical from first to last when passing the epochs, and the amounts of loss range from 0.4885 to 0.3864, while validation accuracy was initially 0.7549. When epochs are increased, validation accuracy increases, but only occasionally. Finally, as the epochs increase, the validation accuracy remains stable, and the final value approaches 0.9189.

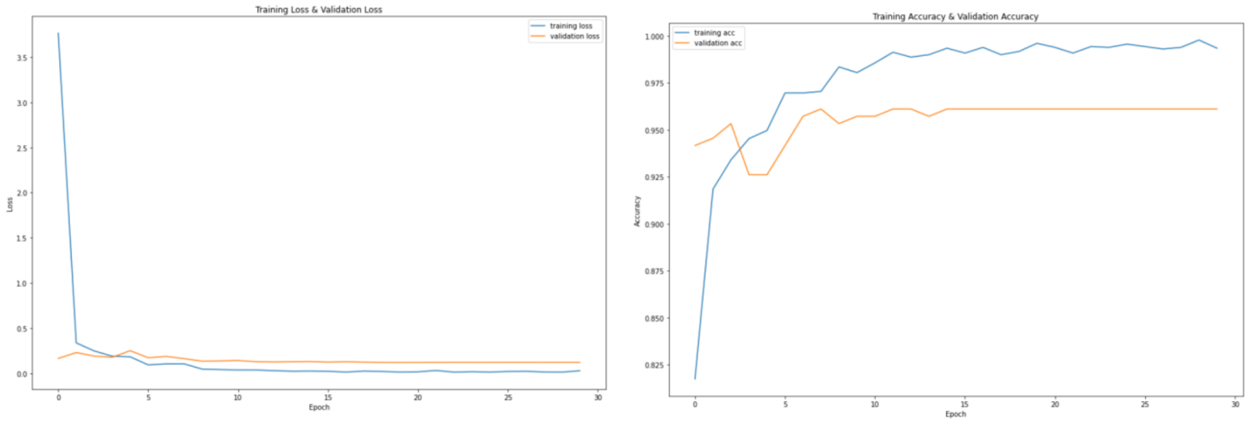

Figure 9: The learning curve of the VGG16 model

In figure 9 shows the training and validation loss as well as the training and validation accuracy of VGG16. Both figures are colored blue to represent training accuracy and training loss. On the other hand, orange represents validation accuracy and validation loss. Initially, we see that the training loss was approximately 3.7653; however, as epochs are increased, the amount of loss decreases. After a few epochs, the amount of loss is stable and close to 0.0130. Initially, we see that training accuracy was approximately 0.8173. After that, as epochs are increased, so does the amount of accuracy. After a few epochs, the amount of accuracy is stable and close to 0.9961. On the other hand, validation loss is almost identical from first to last when passing the epochs, and the amounts of loss range from 0.1650 to 0.2617, while validation accuracy was initially 0.9416. When epochs are increased, validation accuracy increases, but only occasionally. Finally, as the epochs increase, the validation accuracy remains stable, and the final value approaches 0.9548.

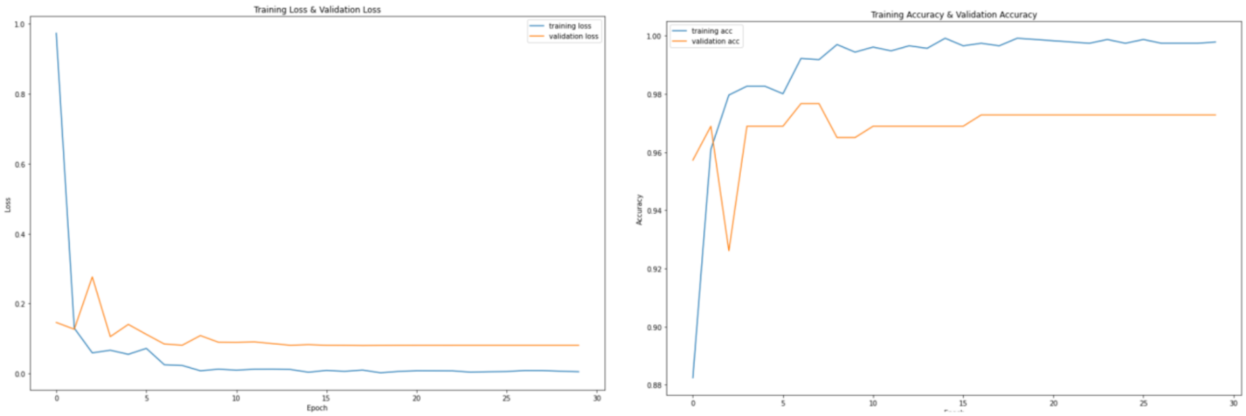

Figure 10: The learning curve of the EfficientNet model

In figure 10 shows the training and validation loss of EfficientNet on the left and the training and validation accuracy on the right. Both figures are two colors, with blue representing training accuracy and red representing training loss. Orange represents validation accuracy and validation loss on the other side. Initially, the training loss was approximately 0.9734; however, as the number of epochs increased, the amount of loss decreased. After a few epochs, the amount of loss becomes stable and approaches 0.0081. Initially, we see that training accuracy was around 0.8824. Following that, as epochs are increased, so does the amount of accuracy. After a few epochs, the amount of accuracy becomes stable and approaches 0.9973. On the other hand, validation loss is almost the same from first to last when passing the epochs, and the amounts of loss are 0.1457 to 0.1328, but initially, when passing the epochs, the value of the loss was up and down. Initially, the validation accuracy was 0.9572. When epochs are increased, validation accuracy increases as well, but it was up and down a few times. Finally, as the epochs increase and the validation accuracy remains stable, the final value approaches 0.9750.

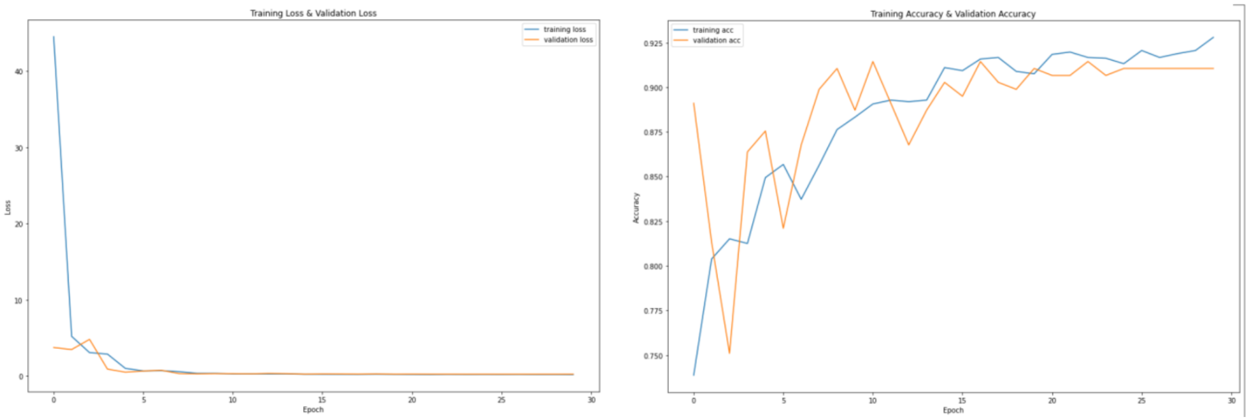

Figure 11: The learning curve of the InceptionV3 model

In figure 11 shows the training and validation loss on the left and the training and validation accuracy on the right for InceptionV3. Both figures are two colors, with blue representing training accuracy and red representing training loss. Orange represents validation accuracy and validation loss on the other side. Initially, the training loss was approximately 44.5378; however, as the number of epochs increased, the amount of loss decreased. After a few epochs, the amount of loss becomes stable and approaches 0.1258. Initially, we see that training accuracy was around 0.7387. Following that, as epochs are increased, so does the amount of accuracy. After a few epochs, the amount of accuracy becomes stable and approaches 0.9547. However, accuracy fluctuated throughout the passing epoch. On the other hand, validation loss is shown first to last when passing the epochs, and the amounts of loss are 3.7354 to 0.2531, but initially, when passing the epochs, the value of the loss was up and down. Initially, the validation accuracy was 0.8911. When epochs are increased, validation accuracy increases as well, but it was up and down a few times. Finally, as the epochs increase and the validation accuracy remains stable, the final value approaches 0.9158.

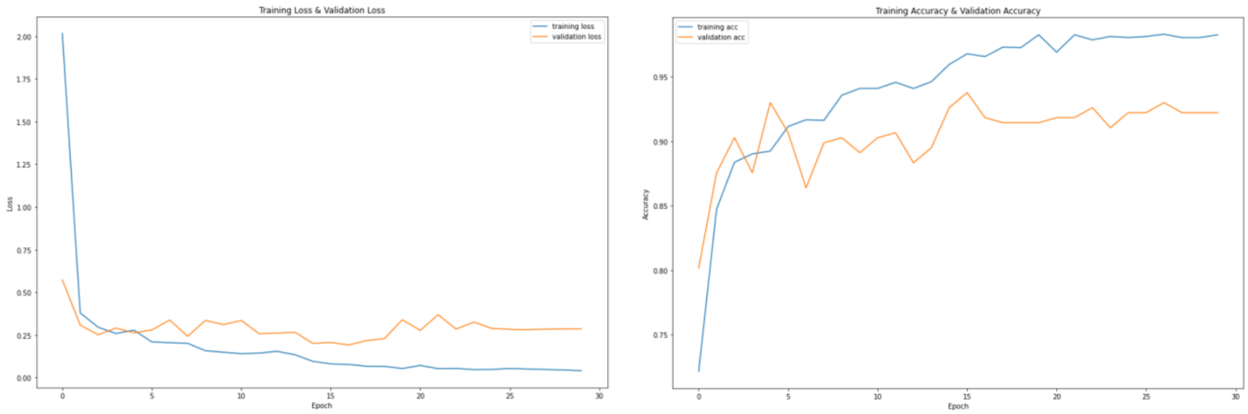

Figure 12: The learning curve of the MOBILENET model

In figure 12 shows the training and validation loss on the left and the training and validation accuracy on the right for MobileNet. Both figures are two colors, with blue representing training accuracy and red representing training loss. Orange represents validation accuracy and validation loss on the other side. Initially, the training loss was approximately 2.0154; however, as the number of epochs increased, the amount of loss decreased. After a few epochs, the amount of loss becomes stable and approaches 0.0337. Initially, we see that training accuracy was around 0.7214. Following that, as epochs are increased, so does the amount of accuracy. After a few epochs, the amount of accuracy becomes stable and approaches 0.9922. However, accuracy fluctuated throughout the passing epoch. On the other hand, validation loss is shown first to last when passing the epochs, and the amounts of loss are 0.5734 to 0.4266, but initially, when passing the epochs, the value of the loss was up and down. Initially, the validation accuracy was 0.8016. When epochs are increased, validation accuracy increases as well, but it was up and down a few times. Finally, as the epochs increase and the validation accuracy remains stable, the final value approaches 0.8986.

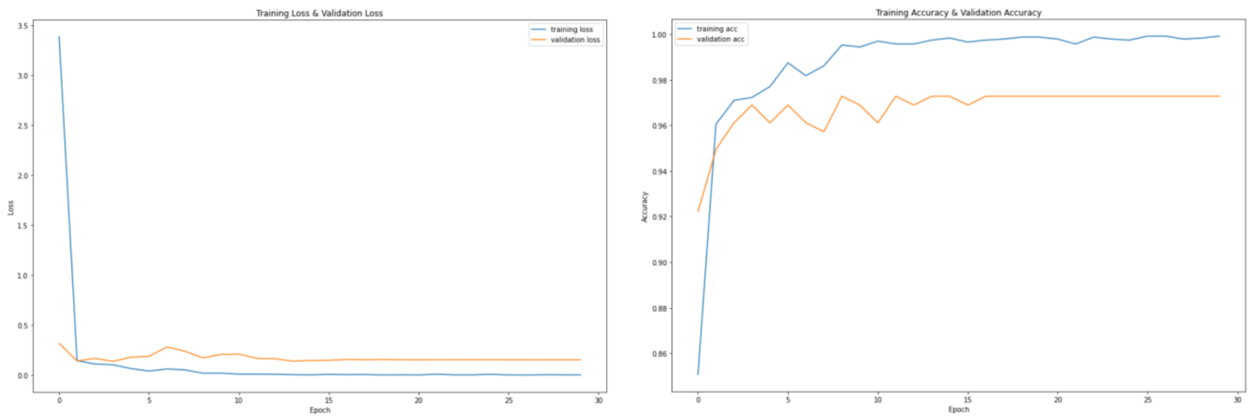

Figure 13: The learning curve of the Resnet50 model

In figure 13 shows the training and validation loss on the left and the training and validation accuracy on the right for ResNet50. Both figures are two colors, with blue representing training accuracy and red representing training loss. Orange represents validation accuracy and validation loss on the other side. Initially, the training loss was approximately 3.3832; however, as the number of epochs increased, the amount of loss decreased. After a few epochs, the amount of loss becomes stable and approaches 0.0157. Initially, we see that training accuracy was around 0.8507. Following that, as epochs are increased, so does the amount of accuracy. After a few epochs, the amount of accuracy becomes stable and approaches 0.9973. However, accuracy fluctuated throughout the passing epoch. On the other hand, validation loss is shown first to last when passing the epochs, and the amounts of loss are 0.3192 to 0.1951, but initially, when passing the epochs, the value of the loss was up and down. Initially, the validation accuracy was 0.9222. When epochs are increased, validation accuracy increases as well, but it was up and down a few times. Finally, as the epochs increase and the validation accuracy remains stable, the final value approaches 0.9688.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

Because of the methods described in this study, bikers' helmets may be identified. The geometry of the picture capture employed makes it simple to repeat this experiment in other locations, which will help researchers better understand the diversity of motorcyclists' attitudes about risk and how they could classify it. Because of this, the appropriate authorities will be able to implement the public policy changes needed to lower the astronomical costs associated with vehicular collisions. The idea for this project came from the need for a system that could automatically identify helmets and notify any incidents. This way, biker’s safety can be guaranteed. For Safety, we want to extract photos of a driver with and without a helmet. First, we categorize photographs of people riding bikes. After classifying the photos, we acquire the condition-based dataset. Non-compliant frames will be destroyed or ignored. We take 3202 row dataset and use six algorithms and get a good accuracy about 98%. Our raw dataset has various restrictions that we discuss in the study, including a maximum of true values and a smaller but non-zero number of erroneous ones. A better outcome is impossible to achieve given that comparable. The precision will improve if we can find a comparable value.

5.2 Future Work

The investigation is complete by 3202 raw Data. We used those data to identify bikers who are wearing helmets or not. We want to continue this study in the future. We predict that in the not-too-distant future, more techniques will be utilized to reduce the amount of features while simultaneously improving accuracy. More tweaks can be tried out to see whether they improve the technique's predictability and scalability. Finally, we plan to conduct experiments with several distinct decision-making approaches, multi-class voting schemes, and several forms of cutting-edge CNN, including those concerned with data collecting and reward analysis. openness to testing the waters with new regulations, such as collective regulations and data-gathering techniques.

Reference

[1] Babu, R., Rathee, D. A., Kalita, K., & Deo, M. S. (2018, September). Helmet detection on two-wheeler riders using machine learning. In ARSSS Int. Conf (Vol. 41, No. 4, pp. 603-623).

[2] Desai, M., Khandelwal, S., Singh, L., & Gite, S. (2016). Automatic helmet detection on public roads. International Journal of Engineering Trends and Technology (IJETT), 35.

[3] Forero, M. V. (2018). Detection of motorcycles and use of safety helmets with an algorithm using image processing techniques and artificial intelligence models.

[4] Rohith, C. A., Nair, S. A., Nair, P. S., Alphonsa, S., & John, N. P. (2019, April). An efficient helmet detection for MVD using deep learning. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 282-286). IEEE.

[5] Saumya, A., Gayathri, V., Venkateswaran, K., Kale, S., & Sridhar, N. (2020, September). Machine learning based surveillance system for detection of bike riders without helmet and triple rides. In 2020 International Conference on Smart Electronics and Communication (ICOSEC) (pp. 347-352). IEEE.

[6] Boonsirisumpun, N., Puarungroj, W., & Wairotchanaphuttha, P. (2018, November). Automatic detector for bikers with no helmet using deep learning. In 2018 22nd International Computer Science and Engineering Conference (ICSEC) (pp. 1-4). IEEE.

[7] Munigala, M. S., Tahniyath, W., & Shaziya, J. (2019). A Hybrid Approach for Helmet Detection for Riders Safety using Image Processing Machine Learning Artificial Intelligence. International Journal of Computer Applications, 182, 50-55.

[8] Dahiya, K., Singh, D., & Mohan, C. K. (2016, July). Automatic detection of bike-riders without helmet using surveillance videos in real-time. In 2016 International Joint Conference on Neural Networks (IJCNN) (pp. 3046-3051). IEEE.

[9] Vishnu, C., Singh, D., Mohan, C. K., & Babu, S. (2017, May). Detection of motorcyclists without helmet in videos using convolutional neural network. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 3036-3041). IEEE.

[10] Devadiga, K., Gujarathi, Y., Khanapurkar, P., Joshi, S., Deshpande, S., Devadiga, K., ... & Deshpande, S. (2018). Real time automatic helmet detection of bike riders. International Journal, 4, 146-148.

[11] Dasgupta, M.: Automated helmet detection for multiple motorcycle riders using CNN. In: 2019 IEEE Conference on Information and Communica tion Technology, Allahabad, India, 6–8 Dec. 2019.

[12] Khan, F.A.: Helmet and number plate detection of motorcyclists using deep learning and advanced machine vision techniques. In: 2020 Second International Conference on Inventive Research in Computing Applica tions (ICIRCA), Coimbatore, India, 15–17 July 2020.

[13] A. Hirota, N. H. Tiep, L. Van Khanh, and N. Oka, Classifying Helmeted and Non-helmeted Motorcyclists. Cham: Springer International Publishing, 2017.

APPENDIX

Research Reflections:

We had a hard time identifying issues and situations while we worked on this project. We started by picking the best algorithms out of the bunch for optimal performance. In addition, everyone had to get a thorough understanding of that with the use of machine learning and python. It wasn't as simple as we thought it would be to gather and collect such a massive dataset. At long last, we've accomplished our goal.

For the CSE-499 Project/Internship Capstone course, students will also be required to complete this project.

DETECTING HELMETS OF THE BIKE RIDERS USING DEEP LEARNING ALGORITHMS

ORIGINALITY REPORT

14%	10%	6%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	6%
2	Submitted to Daffodil International University Student Paper	2%
3	Submitted to University of Newcastle upon Tyne Student Paper	<1%
4	Wei Jia, Shiquan Xu, Zhen Liang, Yang Zhao, Hai Min, Shujie Li, Ye Yu. "Real‐time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector", IET Image Processing, 2021 Publication	<1%
5	Animesh Basak, Md. Solaimanur Rahman, Mushfiqur Rahman. "Prediction of Heart Disease Using an Approach Based on Machine Learning", 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2022 Publication	<1%