A MINI PROJECT REPORT ON

ENSURING SAFETY WITH MOTORCYCLE HELMET DETECTION

BACHELOR OF TECHNOLOGY
IN
Artificial Intelligence & Machine Learning

SUBMITTED BY

GROUP NO: 7

Name of the student	PRN
RUSHIL SHAH	22070126091
SAHARSH MEHROTRA	22070126093
SAMARTH PATEL	22070126098
SAMEER KHATWANI	22070126099

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE – 412115

(A CONSTITUENT OF SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY))

2023-24

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviated Word	**Expansion**
AIML	Artificial Intelligence & Machine Learning
CNN	Convolutional Neural Network
ORB	Oriented FAST and Rotated BRIEF
FLANN	Fast Library for Approximate Nearest Neighbors
SIFT	Scale-Invariant Feature Transform
ReLU	Rectified Linear Unit
RGB	Red Green Blue
Adam	Adaptive moment estimation
KDTree	K-Dimensional Tree
YOLO	You Only Look Once
R-CNN	Region-based Convolutional Neural Networks
RPN	Region Proposal Network
COCO	Common Objects in Context
ALPR	Automatic License Plate Recognition
SVM	Support Vector Machine

CHAPTER I: INTRODUCTION

1.1 Motivation

Assuring safety in various situations has become a top priority in today's technological environment which is continually expanding. In the past couple of years, there has been a significant number of injuries and deaths of motorcyclists due to non-wearing of helmets. In order to reduce these road accidents, the detection of helmetless motorcycles is essential. Wearing of helmets is a crucial in abiding traffic rules and regulations. In spite of these orders, a massive number of motorcycles still do not adhere to these rules.

As a result of these traffic violations, surveillance cameras have been set up by the Government to maintain protocol. Helmets are an essential piece of safety equipment that greatly lowers the risk of head injuries and also saves lives. In counties like Dubai and Pakistan, all major cities have already deployed large scale camera networks for video surveillance purposed to keep watching on this crisis. A recent study has discovered that human surveillance is highly inefficient and causes error as the time period increases. Therefore, we find it vital to have a reliable and robust system which can detect the rule-violators from surveillance images and videos.

1.2 Objectives

Earlier, various researchers have applied traditional image processing methods for feature extraction for detecting helmets. Nowadays, we have been introduced to a new and upcoming technology known as Deep Learning. In the field of computer vision, deep learning can be implemented with the help of Convolutional Neural Networks (CNN). CNN along with a quality dataset can be tuned optimally in order to create a well-structured ‘Helmet Detection’ Model. Automatic feature extraction with the help of CNN will help boost the accuracy of model.

This project primarily aims to ensure safety of motorcyclists by detecting whether they are wearing a helmet or not. Detecting the absence of helmets helps reduce the risk of head injuries which can lead to extreme consequences. In addition to this, our project also aims to the streamline surveillance process. When a helmet is not detected on a rider, an alert or notification can be triggered and this information can be sent to the rule-enforcing authorities. By promoting safety and alleviating human errors, our model can create a safer environment on the roads with the help of CNN.

1.3 Constraints and Feasibility

When it comes to constraints and feasibility, the implementation of a helmet detection system comes with a certain number of challenges. Firstly, the availability of data plays a crux role in the success of model creation. A well-curated and diverse dataset is required for training, validation and testing especially with techniques such as K-fold cross validation. Gathering a comprehensive and balanced dataset can be very challenging. Besides this, the model requires high-performance hardware such as GPUs and TPUs along with suitable frameworks such as TensorFlow and Keras. Fine-tuning of the model with the correct hyperparameters also requires a lot of repetitive training. The scalability is all dependent upon the most optimized and important factors while creation of the model. Even after all these efforts, the accuracy of CNN-based detection systems can be misleading and resulting in a number of false positives and false negatives. Striking the balance between a perfectly fitted model and an over fitted one by applying the appropriate learning rates and activation functions is key.

1.4 Literature Review

Authors	Methods	Database	Results
Madhuchhanda Dasgupta, Oishila Bandyopadhyay, Sanjay Chatterji [1]	The research employed a two-stage approach, using YOLOv3 for motorcycle and rider detection and a custom CNN for helmet detection in traffic surveillance videos. Testing on real traffic data validated the framework's ability to accurately identify motorcycle riders not wearing helmets, contributing to enhanced road safety	The paper uses two datasets those being for Motorcycle and Person Detection and Helmet Detection. For Helmet Detection , the dataset was created using internet helmet images and real-time traffic helmet images	The research achieved promising results, with an accuracy of 91.08% in detecting motorcycle riders (both with and without helmets) using the YOLOv3 model for object detection. In the second stage, the custom Convolutional Neural Network (CNN) achieved an impressive accuracy of 96.23% in detecting whether riders were wearing helmets or not
C. Vishnu, Dinesh Singh, C. Krishna Mohan and Sobhan Babu [2]	The methodology combines adaptive background subtraction and CNNs to automatically detect helmetless motorcyclists in surveillance videos. It involves initial object classification and a head region analysis using two CNN models	This Research used two Datasets IITH Helmet 1 and IITH Helmet 2 which contain video data with sparse and dense traffic respectively	This CNN based approach achieved 99.24% accuracy for sparse traffic and 91.81% accuracy for dense traffic. Moreover, it achieved a 98.63% accuracy in a Helmet vs Non-Helmet Detection.
Adil Afzal, Hafiz Umer Draz, Muhammad Zeeshan Khan, Muhammad Usman Ghani Khan [3]	The proposed approach uses the Faster R-CNN model, consisting of a Region Proposal Network (RPN) for anchor-	It is mentioned that dataset was self-generated from different locations in Lahore.	It can be concluded that their model is efficient in detecting helmets with an accuracy of 97.26% . Mentioned future works includes

	based object detection and classification.		exploring addition features such as number plate recognition and generating fine for non-helmet wearing motorcyclists
Jimit Mistry, Aashish K. Misraa, Meenu Agarwal, Ayushi Vyas, Vishal M. Chudasama, Kishor P. Upla [4]	The authors propose two stages of YOLO model , first one for detecting person in image and second one trained for helmet detection	The paper mentions use of two papers , COCO and Helmet Dataset. COCO dataset was used for initial stage of person detection.	The author's approach provides a very robust and reliable method for helmet detection achieving an accuracy of 94.70% .
Rongali Lalith Vardhan , Voora Uday Bhaskar, Yalipi Sushanth , Vajja Karthik and Dr. Prajwala T.R [5]	This research employs a multi-phase methodology using the YOLO machine learning algorithm for helmet detection and an ALPR system for license plate reading. It includes image labelling, model training, and real-time video stream processing to detect two-wheelers, identify helmets and extract license plates from non-helmeted riders, facilitating immediate enforcement of traffic safety regulations	The document does not specify a particular dataset or database used for this research	The YOLO architecture was used to detect motorcycles, people, helmets, and license plates, aiding in identifying helmet use and imposing fines for violations to promote helmet safety. Future work includes creating a user-police interface for fine management and payment processing, enhancing enforcement
Arvind S Kapse, Shreevamshi, Ravichandra P, Ranadheer Reddy and Revanth Reddy [6]	The methodology involves a two-phase process. In Phase 1, computer vision techniques and SVM classifiers are used to detect bike riders in	Data was gathered from the Indian Institute of Technology Hyderabad's monitoring	This system is accurate with precision of 98.88% and 93.80% , with potential for automated law enforcement.

	video footage, and in Phase 2, the system examines the area around the rider's head to determine helmet usage, all in real-time for automated enforcement		Additionally, with a minor adjustment, the suggested structure easily adjusts to conditions. This structure may be expanded to track down criminals' registration plates

Table 1.1: Literature Review

CHAPTER II: METHOD

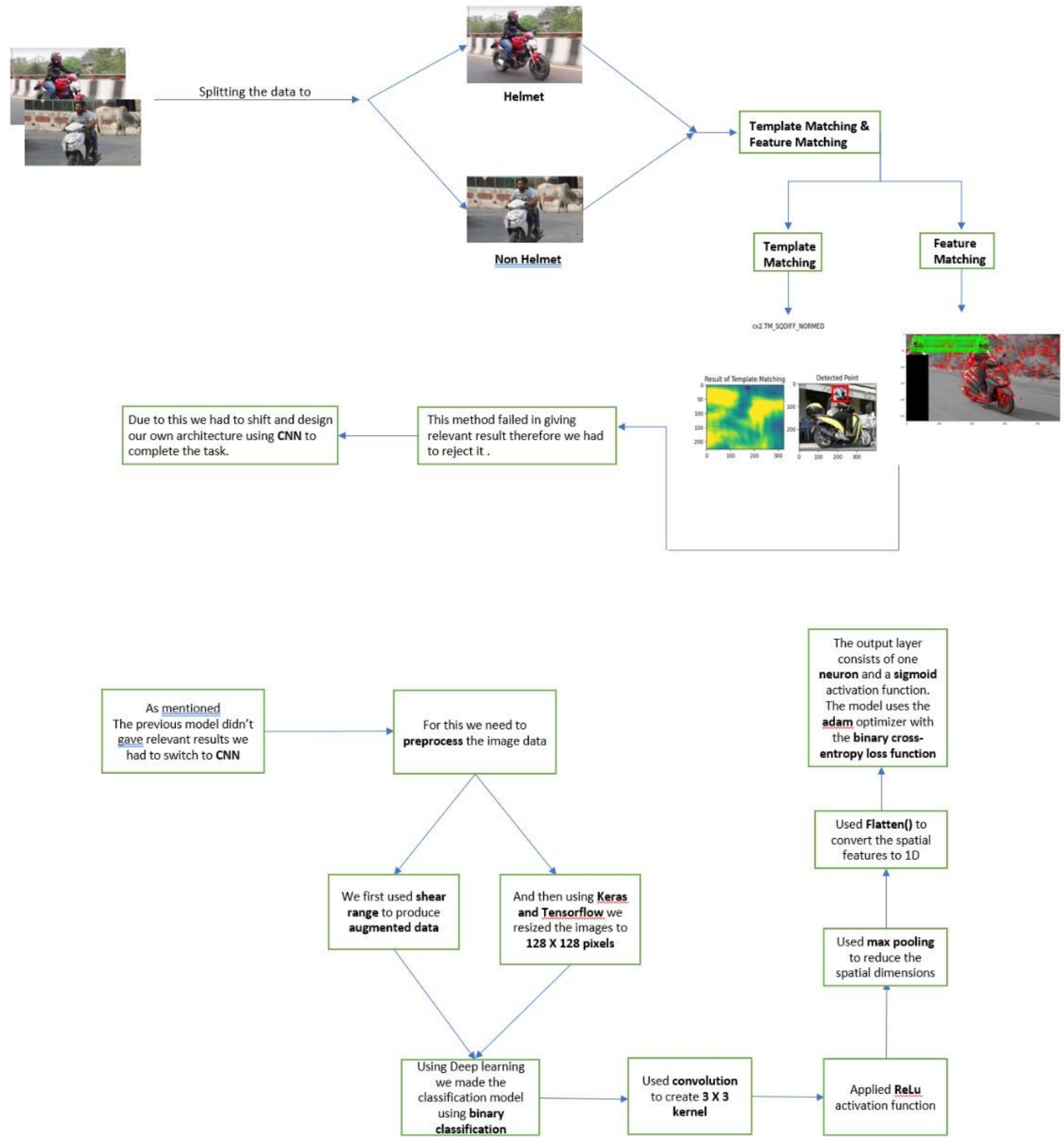

Table 2.1: Block Diagram of Helmet Detection Workflow

2.1 Implementation

In this project, we have applied several computer vision techniques for our helmet detection model. To begin with, we initially applied traditional image processing techniques which includes segmentation of the input images in the dataset. For this purpose, we used edge detection using the Canny edge detection function. Using canny() we found edges and contours in the images by detecting areas of rapid intensity change. However we were unsuccessful in finding the correct shape of the helmet.

Furthermore, using certain functions of OpenCV's library we applied template matching. Using the matchTemplate() function paired along with minMaxLoc() the desired result was obtained. The TM_SQDIFF and TM_CCOEFF methods were vital in order to match the input images matrix with the matrix of the actual image. Once the images were matching, a red box or rectangle was drawn on top of the image with the helmet detected. Out of the 6 different methods, 5 were able to give the necessary results.

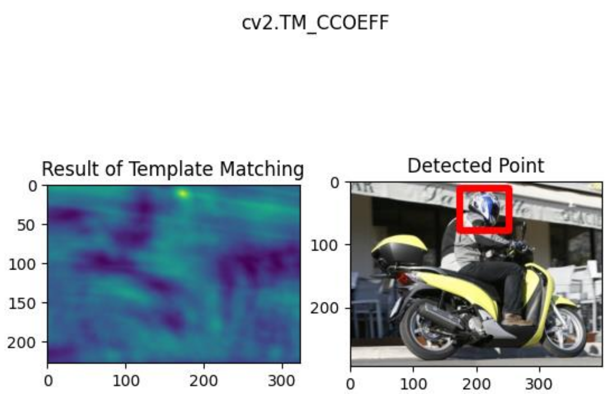

Fig 2.1: Template matching using CCOEFF

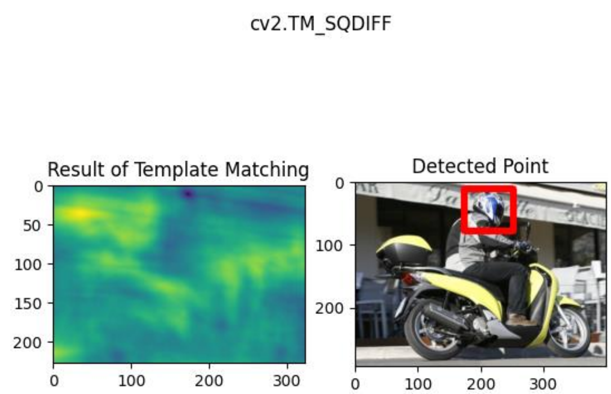

Fig 2.2: Template matching using SQDIFF

Moreover, we delved deeper into feature extraction by using multiple feature extraction algorithms. Primarily, we converted the images to grayscale and then applied the Brute Force Detection algorithm with ORB Descriptors. Using the ORB detector it was able to find the key points and feature vectors in the helmet image. Likewise, was done for the input image. By creating a BFMatcher object we were able to match the descriptors based upon its 'Hamming distance'. Hamming distance is suitable for binary descriptors and was useful for two-way matching. Next, the matches were sorted on the basis of their distances. At last, our code drew the best 25 matches on the new image and connected the lines to visualize the helmet matches.

Fig 2.3: Brute Force Detection using ORB Descriptors

However, this wasn't the only feature extraction technique we applied. Adding to this, we used the SIFT detector in order to detect and compute the key points and descriptors. Using FLANN based Matcher we could achieve more satisfactory results. After defining parameters such as KDTree we implanted the FLANN Based Matcher. The matcher was able to find the k-nearest matches between the descriptors of the helmet and input images. This resulted in a list of potential matches by showing all the objects identified in the image. Thereafter, we produced a mask and compared it periodically. Using OpenCV's drawMatchesKnn() function, we were clearly able to visualize the good matches and colored them green. The wrong matches were highlighted in red.

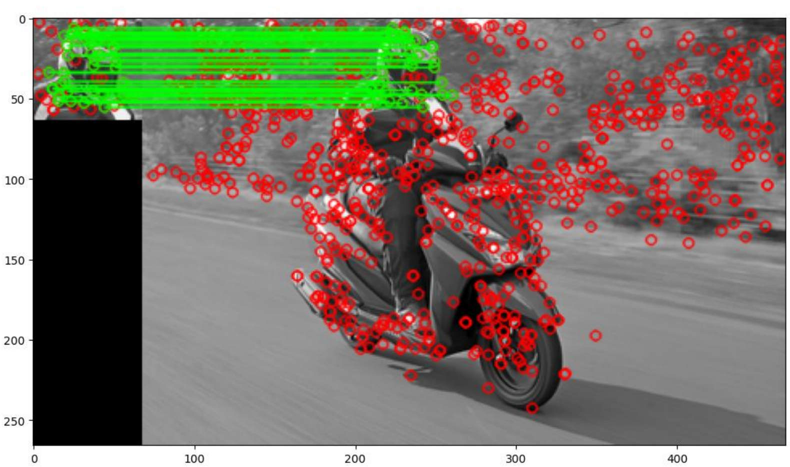

Fig 2.4: Helmet Feature Matching with SIFT features using FLANN based matcher

2.2 Implementation using CNN

Unfortunately, the primary objective of our model was still not achieved. The ML model was still not able to identify the helmet object with its own intelligence. Hence, we finally resorted to deep learning using CNN. The dataset was split into two primary classes: helmet and non-helmet. A convolutional neural network (CNN) is a variant of feed forward neural networks using back propagation algorithm. It learns high-level features from the spatial data like image. The recent widespread success of convolutional neural networks is in its ability to extract inter-dependant information from the images i.e localization of the pixels which are highly sensitive to other pixels. The Convolutional neural network training firstly requires preprocessing. All the training images' pixel values were rescaled to a range of 0 to 1 and then they were augmented with the help of shearing. Since there was a scarcity in the number of input images, we had to increase the data with data augmentation. Using Tensorflow and Keras libraries we resized the images to have a target size of 128 x 128 pixels. The data was organized into batches of 32 images at a time, which is a common practice in mini-batch training for deep learning. The classification of our data is a binary classification.

To begin with, we initialized a blank sequential model which is a linear stack of layers where you can add one layer at a time. In the first convolutional layer we applied 32 filters of size 3x3 with 3 color channels (RGB) and it applies the ReLU(Rectified Linear Unit) activation function. ReLU is a non-linear function that will output the input directly if it is positive, otherwise, it will output zero.

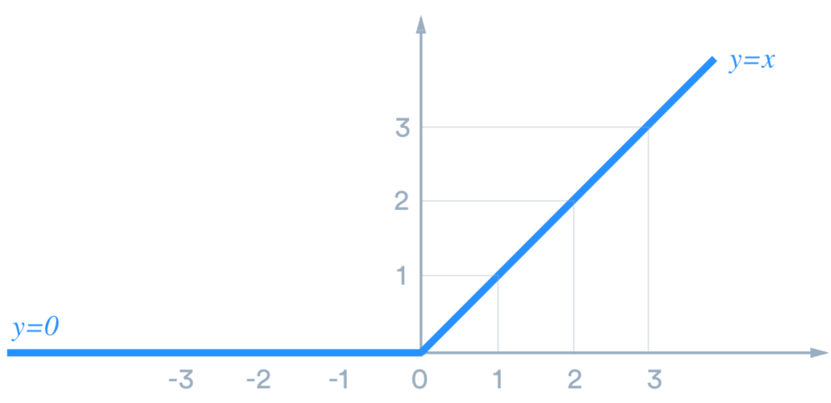

Fig 2.5: ReLU activation function

After this we have applied the first max-pooling layer with a pool-size of 2x2. This reduces the spatial dimensions of the feature maps. In the dropout layer, we have applied a 20% dropout rate which helps in preventing overfitting by randomly deactivating a fraction of least useful neurons during training. In the second and third convolutional layers, we have applied 64 and 128 filters respectively. The same process has been completed until 3 hidden layers were formed with dropouts of 0.2,0.3 and 0.4 respectively. The flattening layer flattens the 2D features and maps it into a 1D vector to prepare the fully connected layers. This output layer contains one neuron and a sigmoid activation function which is used for binary classification tasks. The code uses an Adam(adaptive moment estimation) optimizer. To conclude, the model is compiled and the binary cross-entropy loss function is used, which helps in minimizing the gradient(derivative) of the loss.

Layer	**Filters**	**Size**
C1	16	3x3
Max-Pooling		2x2
C2	32	3x3
Max-Pooling		2x2
C3	64	3x3
Max-Pooling		2x2

Table 2.2: Hyperparameters of proposed CNN

CHAPTER III: EXPERIMENTATION

3.1 Description of the Dataset.

To assist in the development of computer vision models that can determine whether people are wearing helmets in pictures we have used a "Helmet Detection" dataset on Kaggle. Applications like road safety, construction workplace safety, sports monitoring and several others might benefit from using this kind of dataset.

The dataset would include a selection of photos or videos showing various situations where people might be wearing helmets which includes work locations, cycling competitions, motorcycling, etc. Annotations indicating the location of helmets would be present with each image or frame of a film. Bounding boxes, which specify the coordinates of a rectangle surrounding the helmet, are frequently used to represent annotations. This aids in teaching object recognition software how to spot helmets in photos.

If a person wearing a helmet is associated with each of the bounding boxes, it will be indicated by a class label on the bounding boxes. To ensure the model's resilience, the dataset contains a wide variety of scenarios. This covers changes in background, camera angle and illumination among other things. The dataset has been divided into two subsets: a training set for training the model and a testing set for assessing the generalizability of the final model to untested data. Using the testing data, we will calculate the precision and the accuracy of the model using F1-score. The parameters will be fined-tuned accordingly on the basis of the final output.

3.2 Images present in the Dataset.

Fig.3.1: Person wearing helmet

Fig 3.2: Person not wearing helmet

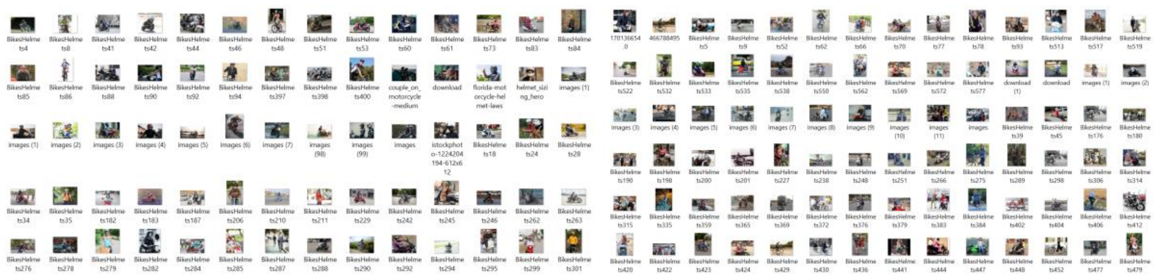

Fig 3.3: Helmet and non-Helmet images

3.3 Results

As for the results we finally fitted our model using our training set data. After tinkering with various different values for the number of epochs, we eventually settled on epochs =100. Once the model was created we plotted a history graph which consisted of model accuracy vs number of epochs.

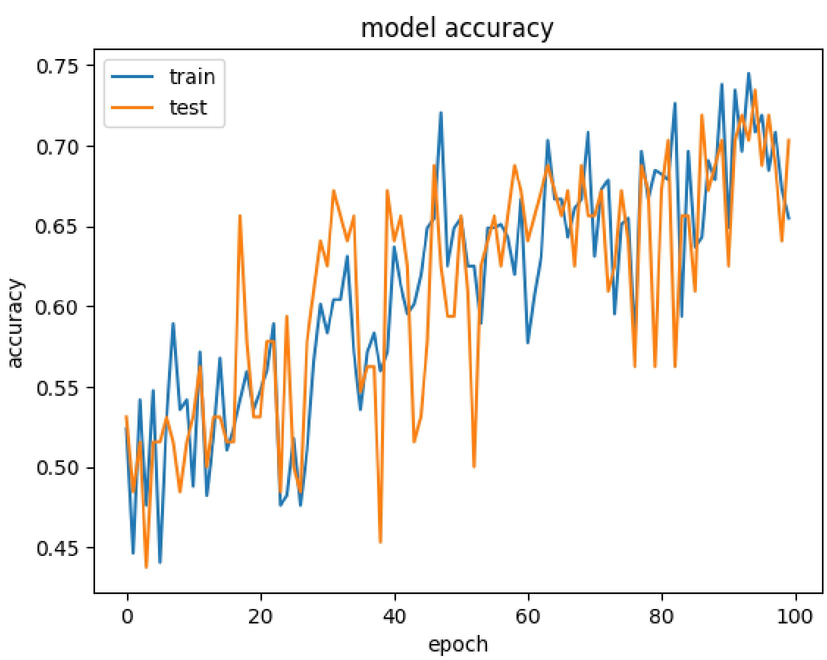

Fig 3.4: Model Accuracy Graph: Accuracy vs epochs

From Fig 3.4 we can conclude that the model was best fitted at around 100 epochs. The accuracy of the model was found to be 70.31%. Although this may seem slightly below industrial requirements, an increase in the amount of image data will definitely be more beneficial in increasing the model’s preciseness. A further extension of this would be to detect the license plate number of motorcyclists who are not wearing a helmet. Using this data, the rule-enforcing authorities can raise tickets and hand out fines for the travelers who aren’t abiding to the traffic rules. Furthermore, helmet detection is also very important in construction sites since the safety of the workers is vital. This can be implanted by adding worksite images to the CNN architecture. Using Streamlit we have created a global deployment which can be used to detect the helmet’s presence. Figure 3.5 contains the front-end model which has been produced as a result. In summary, helmet detection using CNNs holds great promise for improving safety with its continued development and integration into various applications, it can significantly reduce the risks associated with non-compliance and accidents in helmet-dependent environments.

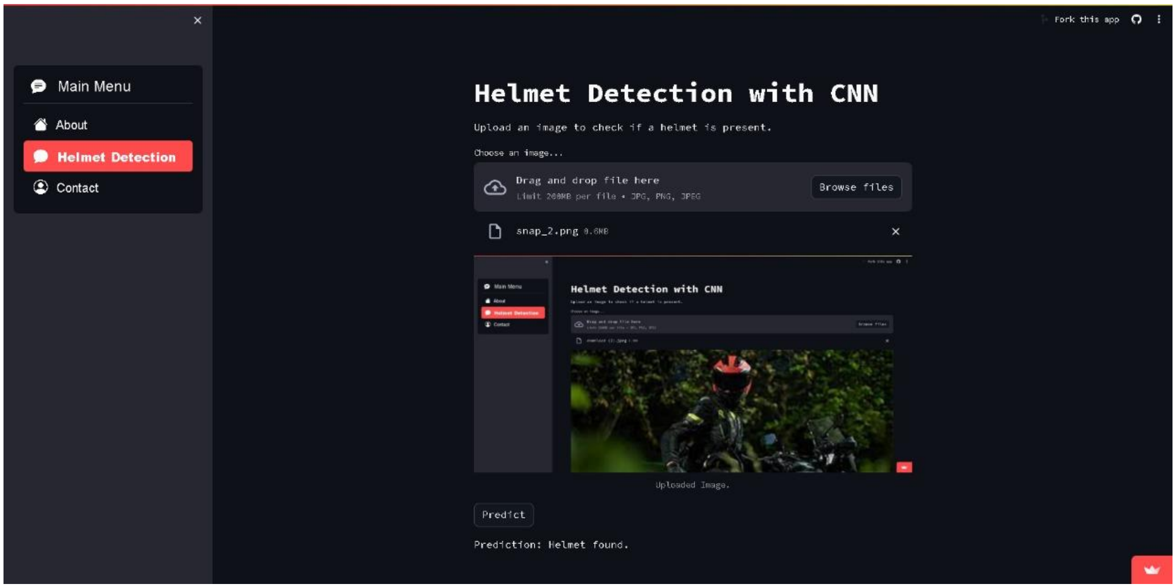

Fig 3.5: Model Deployment using Streamlit

REFERENCES

[1] M. Dasgupta, O. Bandyopadhyay and S. Chatterji, "Automated Helmet Detection for Multiple Motorcycle Riders using CNN," 2019 IEEE Conference on Information and Communication Technology, Allahabad, India, 2019, pp. 1-4, doi: 10.1109/CICT48419.2019.9066191.

[2] C. Vishnu, D. Singh, C. K. Mohan and S. Babu, "Detection of motorcyclists without helmet in videos using convolutional neural network," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 2017, pp. 3036-3041, doi: 10.1109/IJCNN.2017.7966233.

[3] A. Afzal, H. U. Draz, M. Z. Khan and M. U. G. Khan, "Automatic Helmet Violation Detection of Motorcyclists from Surveillance Videos using Deep Learning Approaches of Computer Vision," 2021 International Conference on Artificial Intelligence (ICAI), Islamabad, Pakistan, 2021, pp. 252-257, doi: 10.1109/ICAI52203.2021.9445206.

[4] 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," 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Montreal, QC, Canada, 2017, pp. 1-6, doi: 10.1109/IPTA.2017.8310092.

[5] Vardhan, Rongali & Bhaskar, Voora & Sushanth, Yalipi & Karthik, Vajja & T R, Prajwala. (2023). Helmet Violation and Number Plate Detection System.

[6] Kapse, Arvind & Shreevamshi, & P, Ravichandra & Reddy, Ranadheer & Reddy, Revanth. (2023). A Survey on Helmet Detection by CNN Algorithm. ITM Web of Conferences. 56. 10.1051/itmconf/20235605004.

[7] Telen, Mary Ann & Espinosa, Aljoe. (2023). A Deep Learning Approach for Real-Time Detection of Motorcyclists Without Helmets Using Convolutional Neural Network and YOLOv2. Science International. 35. 441-447.

[8] Guo, Shuqiang & Li, Dongxue & Wang, Zhiheng & Zhou, Xinxin. (2020). Safety Helmet Detection Method Based on Faster R-CNN. 10.1007/978-981-15-8086-4_40.