VEHICLE NUMBER PLATE DETECTION AND BLURRING USING DEEP LEARNING

MSc Research Project
Data Analytics

Darshan Vijay Shringarpure
Student ID: 20221215

School of Computing
National College of Ireland

Supervisor: Aaloka Anant

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Darshan Vijay Shringarpure
Student ID:	20221215
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Aaloka Anant
Submission Due Date:	15/12/2022
Project Title:	VEHICLE NUMBER PLATE DETECTION AND BLURRING USING DEEP LEARNING
Word Count:	5737
Page Count:	16

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Darshan Vijay Shringarpure
Date:	1st February 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

VEHICLE NUMBER PLATE DETECTION AND BLURRING USING DEEP LEARNING

Darshan Vijay Shringarpure
20221215

Abstract

Data privacy is a vital topic in today's world, nowadays data is everything and everywhere. Every day, millions of sensitive data containing the personal information of users are generated. Many businesses that are handling this data makes reliable profit. User information is retrieve and identify using ANPR and other number plate detection system. This system is useful for collecting tolls and to generate parking ticket and this information can be used for malicious acts or monitor user's behavior. Due to these issues, there must be a system that can hide or obscure data in captured images or recorded videos. This study exclusively focuses on data privacy, which guarantees number plate blurring based on deep learning. To accomplish the blurring objective YOLOv5 is used and the changes were made in the existing functionality of YOLOv5. UFPR dataset which contains 4500 images is used in this research. This model has achieved 99% detection with 100% blurring accuracy for images and 94% blurring accuracy for videos at 30 frames per second. Although detection accuracy is 99% this research has achieved 100% blurring of vehicle number plates to maintain the data privacy.

Keywords: Data Privacy, ANPR, YOLOv5, UFPR

1 Introduction

Data privacy is currently a major concern all around the world. Maintaining user privacy in an intelligent digital world is a difficult task. This task is becoming more difficult due to the increasing use of cellphones and CCTV surveillance cameras. Millions of data generated using the cellphones and CCTV cameras across the world which contains user private information such as faces, vehicle number plates and papers which may contain confidential information etc. One of the key factors that invades user privacy is social media. Due to many social media apps user information transferred from one place to other in seconds. Many Large companies uses user's information like name, email and other sensitive data from multiple apps and websites. These firms eventually use this information to track customers behavior or interest in specific products, and they subsequently sell this sensitive data to other businesses in order to generate significant profits. As a result, the European Union (EU) adopted the GDPR (Global Data Privacy Regulation) principles recently Finck and Biega (2021) to minimize the data privacy issues. The GDPR's primary objective is to limit the use of personal data. Data minimization is one of the main objectives of GDPR. The vehicle license plate, which contains the vehicle registration number which can retrieve user's private information and due to this

the user's has the right to protect it. ANPR (Automatic Number Plate Recognition) is currently working in most of the countries to collect the toll and for parking ticket system and in Ireland it is currently working in M50 highway for toll collection. The data retrieved from the user's license plate is used to automatically collect the toll. Number plate identifies the user, and the appropriate toll value is deducted after checking the user's bank account. When the system collects the toll, the database stores the user's vehicle license plate information. ANPR is questioned after accepting GDPR as ANPR uses the vehicle number plate which is private data of user. ANPR reads the car number plate and stores it for up to two years but some nations wants to alter it for up to seven years(Woods; 2017) in order to settle disputes over toll payments. Storing such kind of sensitive information is big concern as it can be use for any malicious activity. Blurring the vehicle number plate to protect the users data is the main business objective of this research project. Blurring is necessary as it can be helpful to protect the user's sensitive data and maintain the data privacy.
The objective of this research is to minimize the violation of user data privacy caused by car number plates.
The research question posed in this research is:

How accurately we can detect and blur the vehicle number plate using deep learning?

This research uses YOLOv5 one of the fastest object detection algorithm (Yin et al.; 2020) to detect the vehicle number plate. YOLO firstly get train with the car number plate images and then detect the vehicle number plate which can be later blur or hide to maintain data privacy. ANPR using YOLO and CV2 has around 99 percentage of accuracy for vehicle number plate detection but with the blurring or hiding the number plate we can achieve 100 percent of data privacy. The major contribution of this research is to maintain the user's data privacy.

2 Related Work

This section is mainly divided into 3 categories. Firstly number plate blurring is explained which has very limited paper on it. Next section is all about the YOLO algorithm which is used for the object detection. The main objective of this research is to resolve the business problem of vehicle number plate data privacy hence the ANPR algorithms which are currently used for vehicle number plate detection are studied in the next section and last section is about the how to verify the blurring accuracy.

2.1 Number Plate Blurring

In this paper (Chan et al.; 2020) which is done the blurring of the European vehicles due to GDPR. The dataset used in the paper was initially verified for sufficient data. Various datasets including UFPR and THI were examined. Author explained that the UFPR and other datasets have a large number of images, however they were created specifically for a few nations, such as Brazil, and working with them for European license plate detection may have low accuracy this approach helped in this research as using particular nations number plate could have give less accuracy hence decide to use mixed number plate dataset. Here they have created the dataset from video taken from phones with high resolution. This dataset includes 18 to 19 thousand images taken at various angles and in various weather situations. Dataset creation is an useful solution, but it is still subject to GDPR guideline because it violates user privacy to capture a license plate without consent. The biggest task following the creation of the dataset and one that this research also encountered was labeling the vehicle number plate. Despite employing a labeling tool to draw a box to detect the number plate on each image, it is still a time-consuming operation. Later, the car number plates were detected using the YOLOv3 algorithm. Due to the large amount of input images, 30000 epochs were run to train the model, and the batch size was increased to 512 to improve the model learning. For YOLOv3, the data was divided into 60 15 and 25 ratios. Despite using photos that were 1920 * 1080 in size to train the model, the author chose to use an image size of 608, which is not explained. In this work, precise tuning was also applied to enhance detection. They have used images to achieve an accuracy of about 90%. The goal of this research is to increase the precision and work with videos to detect and blur the license plate.

2.2 YOLO (You Only Look Once)

One of the fastest object detection algorithms is YOLO (You Only Look Once). From YOLOv1 to YOLOv7, there are different versions of it. In contrast to other YOLO models, the author of this work selected YOLOv5, which is the most stable and user-friendly model for object detection. In this paper (Qin and Yan; 2021), deep learning is used to detect traffic signs. For the purpose of detecting traffic signs, a variety of algorithms like CNN (Convolutional Neural Network), SSD (Single Shot Detection), and YOLOv5 were applied. In this study, the traffic signs were located using 2182 images. In this 80/10/10 ratio comparable technique used and separated the images into train, test, and validation as same has been used in current research. The following step was labeling the dataset images with a labeling tool. Here, the images were 1128 × 2016 and 1536 × 2048 in size. However, the model's basic configurations as the image size were maintained while training and the 200 epoch size and 16 batch size achieved 100% recall and precision for various signs used in this paper. Object detection was performed using SSD, CNN, and YOLOv5, but YOLOv5 performed better than the other two for all sign images. All three algorithms were tested on videos, and in those tests, YOLO outperformed SSD since it detected video frames at a rate of 30 frames per second while SSD detected just 3.79 frames per second (FPS). YOLO performed better with high accuracy despite having a varied number of signs.
AThis paper (Laroca et al.; 2018) employs YOLOv5 for the detection of vehicle number plates. In this research, the author used YOLOv5 to detect the vehicle's license plate following character segmentation and recognition. This paper used the UFPR-ALPR dataset, which includes 150 videos with 30 frames contains 4500 images of automobiles, motorcycles, trucks, and buses in a variety of weather conditions, including snow and rain. Such a dataset provides accurate results and will not skewed during model training. This study also used the SSID dataset, which includes 2000 frames from 101 videos. With 47 frames per second (FPS), the SSID dataset has a 93 percent accuracy rate. On other hand, UFPR-ALPR achieved 79% accuracy with 35 FPS(Frames per Seconds). Utilizing any of the datasets is appropriate for current study because both dataset contains high-quality images and datasets are large enough. YOLOv5, which outperformed openALPR and another trial version of a commercial system, provides both accuracy and reliability. Both datasets begin by detecting license plates and vehicles, and then OCR is used

to segment and recognize characters. Large datasets were required for vehicle detection because the background vehicles were missed by the model. To detect all vehicles, the threshold value of confidence changed to 0.25, and to the test dataset, it was set at 0.125 half of it to do so. To achieve high accuracy for the model this approach is useful. The model accuracy has been improved by adopting this strategy, which has a 99% recall rate and a 100.0% precision rate. In future work they have mentioned to create the robust system which can detect the vehicle numbers plate from all countries.
Number plate detection is an important factor of this study since a high detection rate can lead to more precise blurring. In this paper (Xie et al.; 2018) vehicle number plate is detected by using various algorithms and later they are compared using different evaluation matrix. In this number plate images were rotated or captured in such a way that the model could recognize the plate from a variety of angles. The Taiwanese car dataset, which includes 2049 photos of cars, was used in this study. In this work, several other algorithms, including faster-RCNN SSD and YOLO with different versions, were used. YOLO's ALMD YOLO model outperformed others with 99 percent accuracy. The ALMD model determines the angle of rotation in addition to the bounding box's center coordinates with height and width, improving detection for locating number plates from a variety of angles. Intersection over Union (IoU) and bounding boxes perform well when the rotation angle is zero, but when the area of rotation varies to a certain extent, the IoU value abruptly decreases, leading to poor accuracy. As a result, when we read the number plate from different angles, our accuracy is lower than it would be if we were only reading it straight. By utilizing multi-directional car number plates and less computing techniques than those previously employed by CNN, this research claimed to be able to tackle the challenge of detecting license plates from different angles. The unique model in this work is also based on the YOLO method, which is now being applied in current research. However, there are several issues that must be resolved to enhance the accuracy of these paper model. These issues include the lack of multidimensional number plate data along with low quality photos. Security cameras are present in toll booths, parking garages, and a few other public locations. For security purposes, this camera records everything, hence the objective of this study Gnanaprakash et al. (2021) is number plate detection using CCTV surveillance cameras. This study made use of toll booth data, which records every car that passes through a toll. Then, using video frames, this video collection is transformed into photos. After receiving the image dataset, a labeling tool was used to label each and every image. As we have seen in the past, each author used a different type of labeling tool to label the YOLO dataset, and the author of this research did the same. After labelling dataset next step is to train the model. In this paper custom object car is detected using using Jason and H5 files. Jason files contains all the features of the car and H5 files having weights of the model which is important to detect the car number plate. After detection of character segmentation and recognition were done using ImageAI.

2.3 ANPR algorithms

The OpenALPR algorithm can be used for a variety of tasks, including parking ticketing and toll collecting. This paper (Kl et al.; 2021)demonstrates how the parking ticket system uses OpenALPR. Here, the free parking space is checked by analyzing the frames using the openCV package of Python. The box is drawn over the empty space in the frame after the frame has been taken by the camera. These coordinates are later pre-

served in the yaml file. After a video is sent to a motion detector, and each slot in the frame is checked. Each pixel in this image is averaged, and if the average is high, it is assumed that there are no cars present and the parking space is therefore unoccupied and if average is low we can determine parking space occupied.Once the car is located, the camera snaps a picture and feeds it to OpenALPR. From the images of parking and leaving that are fed into OpenALPR, the in and out time is updated, and the appropriate toll is then determined based on the in and out time. This entire system basically worked on OpenALPR as recognizing and updating in and out time is important task for parking ticket.
In this paper (Pourhadi et al.; 2022), the Super Resolution Generative Adversarial Network (SR-GAN) and the YOLO method are utilized to recognize the license plate. They have raised the ALPR accuracy by 18% as a result of this combination. Here, YOLOv5 takes the input image and performs localization before using the SR-GAN network to identify the license plate for the YOLO model, which then completes character segmentation and recognition at the end to get the desired result. Here, a dataset with 6000 registered plates is employed, which operates on various models. The YOLOv5 model has the best accuracy among all others, which is 97%.

2.4 Blurring verification

Detection can be done with the use of above algorithm which were used by various authors but while checking about the blurring accuracy author came across this paper (Abbasi et al.; 2022). In order to preserve anonymity, face blurring is done in this paper using Gaussian blurring. Here, researchers blurred face picture data before passing it to multiple object identification models, which checked the faces for accuracy using face alignment, face detection, face verification, and face detection. In this method, the author first blurs the faces before trying to recognize them using various models and facial recognition parameters in order to check the blurring accuracy. The negative accuracy defines the effective blurring. This approach is extremely helpful for validating blurring, and it is being used in the current study to verify blurring of license plates.

3 Methodology

This research uses the deep learning model for the object detection. First step for any model development is to check the data thoroughly which has been done here for the model building. There are lots of methods using which we can do the object detection like Faster CNN(F-CNN), SSD (Single Shot Detection) and YOLO (You Only Look Once) with different version YOLOv3, YOLOv4, YOLOv5 etc. YOLO is the one of fastest algorithm for object detection (Yin et al.; 2020). YOLOv5 pre-trained model is used in this research for object detection and later results were explained to support the successful object detection. In the process of object detection existing model functionalities has been changed to hide the object after detection to maintain the data privacy.

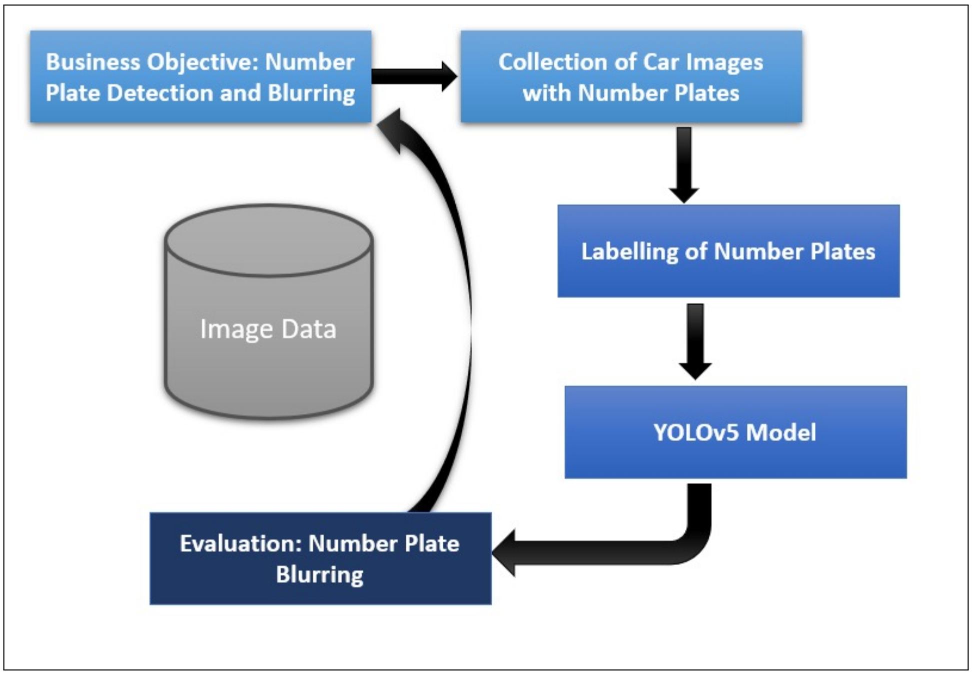

Figure 1: Methodology for Model Implementation

In Figure 1, FolloTwing steps has been taken to built and verify the model.

- Understanding of Data
- Number Plate Detection & Blurring
- Labelling and Data Augmentation
- Developing the YOLOv5 model
- Number Plate Blurring Results and Evaluation

3.1 Data Understanding

In this study, two datasets were used. The first dataset [1] is accessible to everyone on Kaggle. There are 432 vehicle images with number plates in PNG file format. It includes different types of car license plates that can be used for training and creating a model. Vehicle number plate data collected and generated by (Laroca et al.; 2018) is the second dataset [2] used in this research. Although the dataset is not accessible publicly, the author shared it for study purposes. Number plates are visible from different angles and in varied weather situations, thus the model will be good at recognizing the number plates in various conditions. The dataset contains 4500 vehicle images, including cars, buses, trucks, and motorbikes, which were obtained from video files at 30 FPS.

3.2 Vehicle Number plate Detection and Blurring

The first step is to find the business objective of this study. The main business objective of this research is to identify and blur the vehicle number plate to protect the privacy of

[1] https://www.kaggle.com/datasets/andrewmvd/car-plate-detection?resource=download
[2] https://drive.google.com/drive/folders/1RUOXhw6vlyIYWXC0T2L6-qAd5wZDkbs4?usp=sharing

the data. This is done using the YOLOv5 model.

3.3 Labelling and Data Augmentation

Labelling is the important part of the YOLOv5 model building. Labelling is done using the below labeling tool [3]. In this tool images were selected one by one to locate the number plate and draw the bounding box to get the labels.
In this research labels are created for train and val images and now YAML file is needed which contains the path of the train and val images. While training the model these images use the default augmentation technique which is based on image-space and color-space. It is only applied on training dataset. The mosaic augmentation basically adds original image plus 3 other images rotated at certain degree and the one image is never present same way while training the model. This augmentation is present in hyp-scratch yaml file. The file can be changed as per the requirement, but YOLO official website recommends it to use the default file and this research used the default augmentation as per above yaml file.

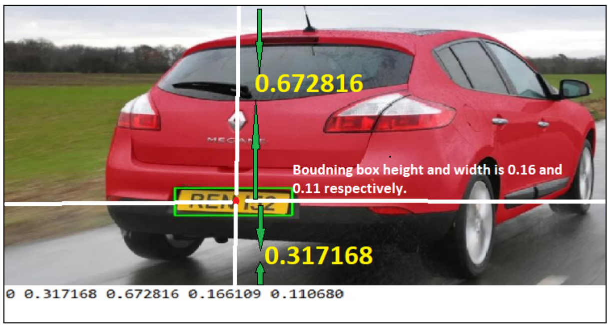

Figure 2: Number Plate Located in Car Images using Labelling Tool

In Figure 2 the labelling is done using the YOLO Labelling tool. Here the first 0 represents the number of objects which is only one (Number Plate).
Number of Objects = Total Objects - 1
Next two numbers represents the distance of the bounding box center from the image height and width and last two number represents the bounding box total width ad height. Both datasets doesn't have the labels and generated manually using the YOLO labeling tool. The labeling for the first dataset, which contains about 432 images, took about an hour, whereas the labeling for the second dataset required about ten hours of manual effort.

3.4 Developing the YOLOv5 model

After making the labels, the next stage is to develop the YOLOv5 model. In this case, git cloning [4] is utilized to create the model using the YOLOv5 pretrained model. The model is trained using 80 percent of the images, and changes were later made to the YOLOv5's existing functionality that would detect and blur the vehicle number plate, which is the key objective of the research project.

[3]https://github.com/developer0hye/Yolo_Label/releases/download/v1.1.1/YoloLabel_v1.1.1.zip
[4]https://github.com/ultralytics/yolov5.git

3.5 Number Plate Blurring Results and Evaluation

Number plate blurring is done by changing the existing functionality of the YOLOv5 model. Results are explained using the precision, recall and mean average average precision (mAP).

4 Design Specification

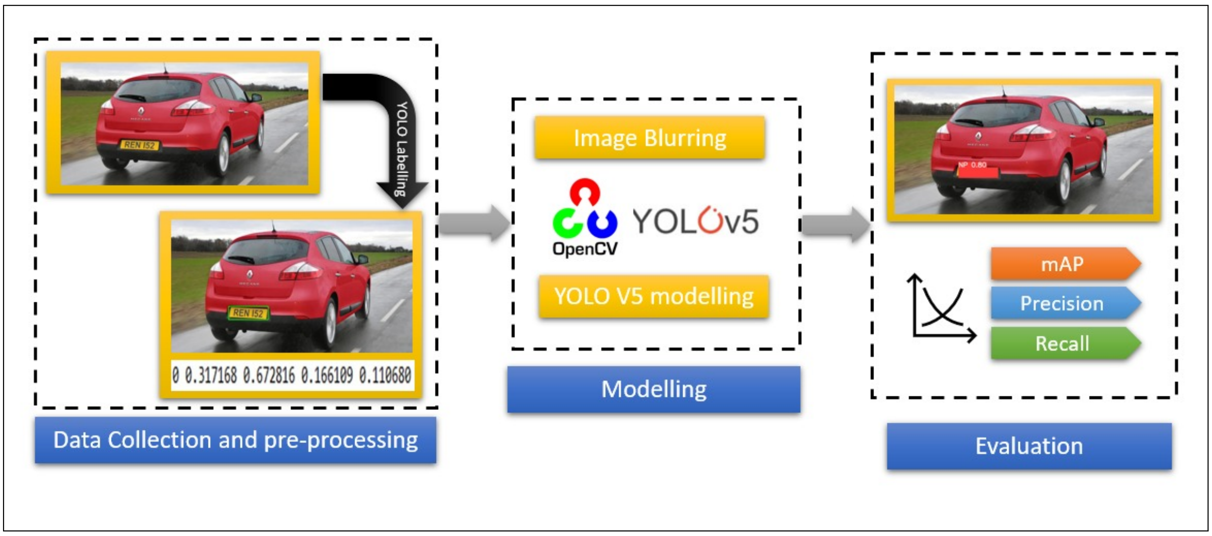

Figure 3: Number Plate Detection and Blurring Architecture Using YOLOv5

The architecture in Figure 3 is mainly divided into 3 categories.

- Data Collection and Preprocessing:
- Modeling
- Evaluation

We have already seen data collection and pre-processing in above section. Modeling is the next section which describes the implementation of the models followed by evaluation.

5 Implementation

The following steps are involved in implementation: collecting data for the yolov5 model to run; defining the model architecture and functionality with the appropriate parameters; training the model with these parameters; changing the functionality of YOLOv5 to blur or hide the number plate; and, finally, evaluating model results and graphs.
This research based on image data that has been split into train, validation and test data, with labels for each generated manually during the data pre-processing stage for both datasets. The ratio used to divide the images into train, test, and validation groups is 80/10/10. Data is utilized to train the model in an 80% ratio, with the remaining

10% being used to validate the model. Several parameters are utilized to train the model, and they are described below.

5.1 Model training

In order to train the model for this study, label quality with object coordinates was a required for accurate object detection. These labeled input images are processed by YOLOv5 during training. The model is initially run using the default settings recommended in the yolov5 official website [5]. The model was initially ran with 10 epochs, but because of the poor number plate detection, the epoch size was expanded to 25 epochs. Here, training and testing sizes of the images should be kept same for better detection as described in above official website. Initially, the image size was set to be the default, which is 640, where the model would perform on 640 * 640 image size. As a result, the size is enlarged to 1980 pixels, which is useful for identifying things in high-resolution videos and images that cameras typically take. The last parameter is the batch size, which indicates how many groups of images will be used to train the model at the initial epoch. As recommended in the paper (Ting et al.; 2021), the batch size employed in this study to train the model was 32. Considering all these parameters, the model has been designed and built. In this step the images first passed to the 3 architectural blocks, i.e. Backbone (CSPDarkent), Neck (Panet) and Head. In the backbone layer CSPDarknet is used for the feature extraction from the images. In this stage bounding box is drawn in the image based on image size and later it checks where exactly number plate is in the image. once number plate is found all the features are extracted and pass it to the next stage. Next stage is Panet which takes the input from the head and perform the aggregation on the features and pass it to next stage for prediction.In the last stage anchor-box is predicted from the generated predictions for the number plate detection.

5.2 Number Plate Detection

The model generated using the above parameters will then be tested. These images are sent to the trained models, and the model then determines where the number plate is located in the image. In accordance with this, if it is recognized based on model training, the box is drawn on the license plate. YOLOv5 also supports video and YouTube's video streaming, so if a video file or YouTube video link sent to the trained model, the model would try to recognize the license plate in each frame and draw a bounding box over it.

5.3 Blurring or Hiding

The main purpose of this step is to hide or blur the number plate that the model can detect. Blurring is entirely dependent on detection; if detection is successful, blurring will also be successful since after detecting the number plate blurring is done. In this phase, blurring or hiding is accomplished by modifying the YOLOv5's designed functionality.
To test the model the detect.py file of python is used which will take the image or video as an input and try to recognize the number plate in the images or videos.
This detect.py file contains run method which will take input as image size, image or video and some other parameters like confidence threshold. To draw the bounding box over the object this method has Annotator.box_label method where Annotator is class

[5]https://docs.ultralytics.com/tutorials/training-tips-best-results/

imported from utils.plots file.
In Utils.plots.py file Annotator class box_label method is defined inside which CV2 library is used to draw the rectangle.
CV2.rectangle method is responsible to draw the bounding_box over the object. This method takes the image size and bounding_box co-ordinates as well as color and most important parameter thickness.
Here changes are made in the thickness parameter which will fill the bounding_box with the same color of bounding_box border.
As per the mentioned changes number plate is totally blurred.

5.4 Verify Blurring or Hiding of Number Plate

The model checked the number plate for detection in this step again after the number plate blurring and hence the model was not identified any number plates, indicating that 100% blurring had been completed.

As per above implementation, model should be good enough to detect the number plate in first occurrence so the blurring accuracy will be good. This process can be repeated to achieve best result of blurring or hiding.

6 Evaluation

This section is all about the evaluation of models implemented in model designing phase. To explain the results tensorboard has specific library tensorboard which is preinstalled in YOLOv5 requirement file.
Below are the terms which are important in the YOLOv5 Evaluation.

6.0.1 Intersection over Union (IoU))

Intersection over Union (IoU) is calculated as ratio bounding box of the image to the predicted box of the image.

6.0.2 Mean Average Precision (mAP)

Average precision is the area under the precision-recall curve. mAP is the mean value of average precision.

6.1 Experiment / Case Study 1

Here, the first experiment was conducted using kaggle car dataset which contains 432 images. It takes around half n hour to execute the code in Google Colab with GPU.
After execution of first experiment, below are the model results:

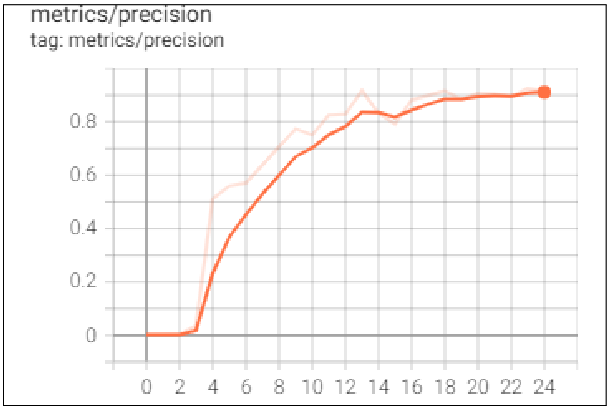

Figure 4: Precision

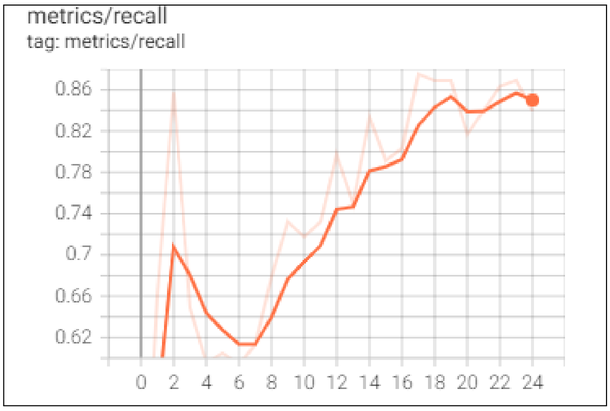

Figure 5: Recall

Here in Figure 4, After execution of half epochs precision value increased upto 80% and with increasing epoch size 92% precision is achieved for current dataset.
In Figure 5,Recall value reached at 70% at the end of the second epoch and dropped to 60% from epochs 2 through 8. From epoch 9 to 18 it reaches 96%, following which it remained stable until the end of 25th epoch.

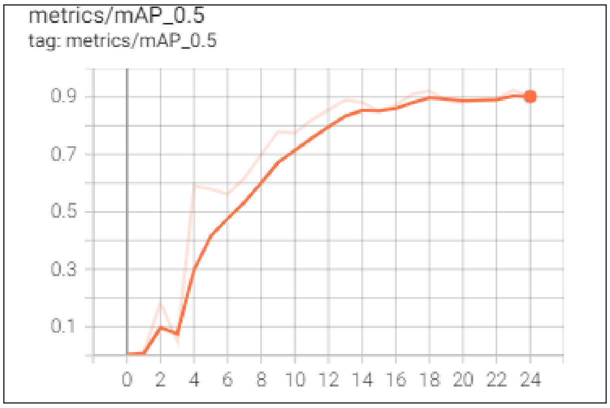

Figure 6: mAP 0.5

In Figure 6, Starting at 0, the mAP value increased by up to 90% by the ending of the 18th epoch before remaining constant until the completion of the last epoch.

Figure 7: Irish Number Plate Hidden for Privacy

This Figure 7, clearly demonstrates that this model is capable of detecting and hiding the Ireland vehicle number plate. The blur image is verified on the train model, which shows no number plate detection, to confirm blurring accuracy.

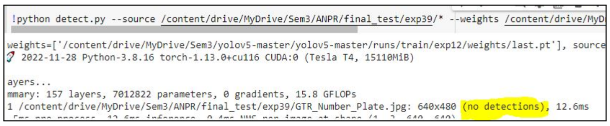

Figure 8: Blur Validation

The Figure 8 clearly demonstrates that the model did not recognize the blurred number plate, hence blurring of object is completely achieved.

6.2 Experiment / Case Study 2

Here the UFPR dataset is used which contains 4500 images. With the Google colab GPU it takes around 90 minutes to run the code and get desired output.

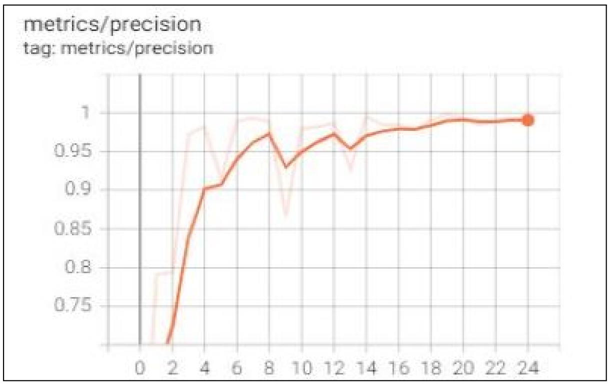

Figure 9: Precision

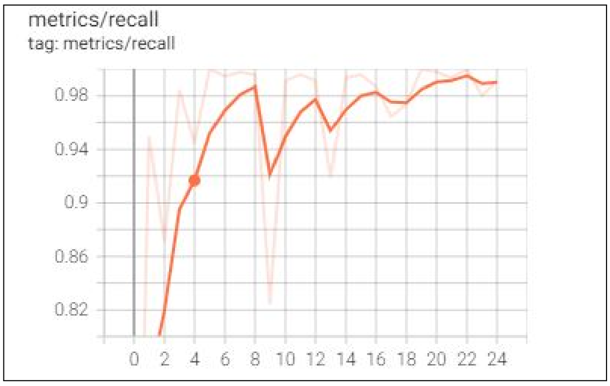

Figure 10: Recall

In Figure 9, we can see that the precision value increases with an increasing number of epochs and reaches 99% after 25 epochs.
In Figure 10, between 6-8 epoch recall value decreased but with increasing epochs it reached up-to 99% at the end of 25th epoch.

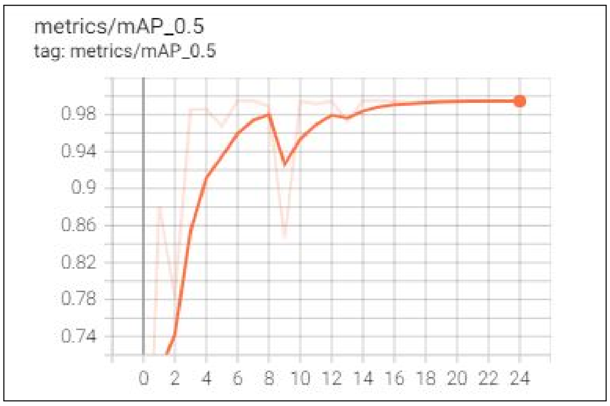

Figure 11: mAP 0.5

Figure 11, Starting with 0.70 mAP, the value grew up to 98 percent until the eighth epoch, dipped to 92 percent until the tenth epoch, and then increased up to 99 percent until the end of the last epoch.

Figure 12: Number Plate Hidden fro Data Privacy

The Figure 12 illustrates that the model can accurately detect and blur the vehicle number plate.

```
Fusing layers...
Model summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
image 1/450 /content/drive/MyDrive/Sem3/ANPR/final_test/exp41/track0136[01].png: 384x640 (no detections)
image 2/450 /content/drive/MyDrive/Sem3/ANPR/final_test/exp41/track0136[02].png: 384x640 (no detections)
image 3/450 /content/drive/MyDrive/Sem3/ANPR/final_test/exp41/track0136[03].png: 384x640 (no detections)
image 4/450 /content/drive/MyDrive/Sem3/ANPR/final_test/exp41/track0136[04].png: 384x640 (no detections)
image 5/450 /content/drive/MyDrive/Sem3/ANPR/final_test/exp41/track0136[05].png: 384x640 (no detections)
image 6/450 /content/drive/MyDrive/Sem3/ANPR/final_test/exp41/track0136[06].png: 384x640 (no detections)
image 7/450 /content/drive/MyDrive/Sem3/ANPR/final_test/exp41/track0136[07].png: 384x640 (no detections)
image 8/450 /content/drive/MyDrive/Sem3/ANPR/final_test/exp41/track0136[08].png: 384x640 (no detections)
image 9/450 /content/drive/MyDrive/Sem3/ANPR/final_test/exp41/track0136[09].png: 384x640 (no detections)
image 10/450 /content/drive/MyDrive/Sem3/ANPR/final_test/exp41/track0136[10].png: 384x640 (no detections
```

Figure 13: Blur Verification

After blurring of all the number plates, model will try to detect the hidden number plate again to verify blur accuracy.
Since there is no detection for the number plate in the Figure 13, we can say that blurring has been completely achieved.

6.3 Discussion

Irish and Brazilian license plates have both been successfully blurred using this research. The number plates for buses, trucks, motorcycles, and cars were all blurred for the Brazil dataset, as can be seen in the Figure 14

Figure 14: All types of vehicle blurred for privacy

YOLOv5 model required more images for training, First dataset has only 432 images therefore accuracy was low to 92% but after using the Brazilian number plate dataset accuracy increased upto 99%.

7 Conclusion and Future Work

The main objective of this study was to blur or hide the vehicle number plate. The number plates were recognized and blurred by the model, which was trained using YOLOv5, in both images and videos. We have 99 percent accuracy rate for images, and a 94 percent accuracy rate for videos at 30 frames per second. Despite a detection rate of 99 percent, the blurring obtained in this case is 100 percent after detection because no number plate can be identified after blurring, thus we can say that the objectives were achieved.

In this study, number plates were blurred for images and videos using two datasets; one was able to detect and blurr the number plates for all countries, but accuracy was poor, and the other dataset is specific to Brazilian number plates; therefore, in the future, if one mixed dataset that would be large enough to be used for blurring, then blurring can be achieved for all types of number plates. Since object can be blurred using YOLOv5 method with this approach, this research will likely benefit to data privacy in the future.

Acknowledgement

I would like to acknowledge and give my warmest thanks to my professor Aaloka Anant to guide me and help me to complete this research. I would also like to give my sincere thanks to all NCI staff which help me knowingly or unknowingly to complete this research. Finally I would like to thank my family members and friends who support me throughout the research.

References

Abbasi, W., Mori, P., Saracino, A. and Frascolla, V. (2022). Privacy vs accuracy trade-off in privacy aware face recognition in smart systems, *2022 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–8.

Chan, L. Y., Zimmer, A., da Silva, J. L. and Brandmeier, T. (2020). European union dataset and annotation tool for real time automatic license plate detection and blurring, *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)* pp. 1–6.

Finck, M. and Biega, A. (2021). Reviving purpose limitation and data minimisation in personalisation, profiling and decision-making systems, *CoRR* **abs/2101.06203**.
URL: *https://arxiv.org/abs/2101.06203*

Gnanaprakash, V., Kanthimathi, N. and Saranya, N. (2021). Automatic number plate recognition using deep learning, *IOP Conference Series: Materials Science and Engineering* **1084**(1): 012027.
URL: *https://dx.doi.org/10.1088/1757-899X/1084/1/012027*

Kl, T., Yadav, U., Krishnan, V., Ks, V. and Pm, M. (2021). Application based smart parking reservation system using openalpr, **83**: 5004–5008.

Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R. and Menotti, D. (2018). A robust real-time automatic license plate recognition based on the yolo detector, *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10.

Pourhadi, N., Shafizadeh, B., Eshghi, F. and Kelarestaghi, M. (2022). Yolov5-based alpr improvement using selective sr-gan, *2022 2nd International Conference on Computing and Machine Intelligence (ICMI)*, pp. 1–6.

Qin, Z. and Yan, W. Q. (2021). Traffic-sign recognition using deep learning, *International Symposium on Geometry and Vision*, Springer, pp. 13–25.

Ting, L., Baijun, Z., Yongsheng, Z. and Shun, Y. (2021). Ship detection algorithm based on improved yolo v5, *2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE)*, pp. 483–487.

Woods, L. (2017). Automated number plate recognition: data retention and the protection of privacy in public places, *Journal of Information Rights, Policy and Practice* **2**(1): 1–21.

Xie, L., Ahmad, T., Jin, L., Liu, Y. and Zhang, S. (2018). A new cnn-based method for multi-directional car license plate detection, *IEEE Transactions on Intelligent Transportation Systems* **PP**: 1–11.

Yin, Y., Li, H. and Fu, W. (2020). Faster-yolo: An accurate and faster object detection method, *Digital Signal Processing* **102**: 102756.
URL: *https://www.sciencedirect.com/science/article/pii/S1051200420301019*