Real-Time Car Number Plate Extraction using Computer Vision and Deep Learning

Progna Dipto Saha
Department of Electrical and Electronic Engineering, BUET
Dhaka, Bangladesh
1906027@eee.buet.ac.bd

Md Rakib Hossain
Department of Electrical and Electronic Engineering, BUET
Dhaka, Bangladesh
1906034@eee.buet.ac.bd

Md Ashikur Rahman
Department of Electrical and Electronic Engineering, BUET
Dhaka, Bangladesh
1906094@eee.buet.ac.bd

Manzoor Elahi Tamjeed
Department of Electrical and Electronic Engineering, BUET
Dhaka, Bangladesh
1906046@eee.buet.ac.bd

***Abstract*—This project aims to innovate traffic monitoring and management by employing computer vision and deep learning technologies for real-time car number plate extraction. By analyzing live video feeds, the system is capable of identifying and counting vehicles, accurately extracting frames containing visible number plates, and recognizing the alphanumeric characters on these plates. The extracted data is systematically stored in a CSV format, facilitating easy access and analysis. The implementation utilizes advanced deep learning models such as Inception ResNet v2, YOLO v5, and VGG-16, combined with Tesseract OCR for character recognition. Through rigorous testing on diverse datasets, the project demonstrates significant accuracy in real-time number plate detection and character recognition, offering a scalable solution for enhanced vehicle surveillance and security measures.**

***Index Terms*—Computer Vision, Deep Learning, Automatic Number Plate Recognition, Convolutional Neural Networks, InceptionResNetV2, YOLO v5, VGG-16, Optical Character Recognition, Tesseract OCR, Image Preprocessing, Character Segmentation, Traffic Monitoring, Vehicle Identification.**

I. Introduction

Our project, "Real-Time Car Number Plate Extraction using Computer Vision and Deep Learning," is inspired by Girinath N's 2020 seminal work, "Automatic Number Plate Detection using Deep Learning." This paper was instrumental for its application of cutting-edge deep learning models to the accurate identification of vehicles under varied conditions.

Chosen for its alignment with our objective of innovating within traffic management, Girinath N's paper guides our system's development aimed at recognizing and extracting vehicle number plates from live video feeds. By adapting and expanding upon these proven deep learning models—Inception ResNet v2, YOLO v5, and VGG16—our work promises to enhance automated traffic surveillance. With training accuracies reaching up to 72.5

Our project not only benchmarks against but seeks to exceed the mother paper's achievements by integrating these models into a real-world framework, potentially redefining accuracy and reliability standards in vehicular traffic monitoring and management. Our ambition is to reflect the mother paper's efficiency and contribute significantly to the application of machine learning in intelligent traffic systems, paving the way for safer and smarter urban transport networks.

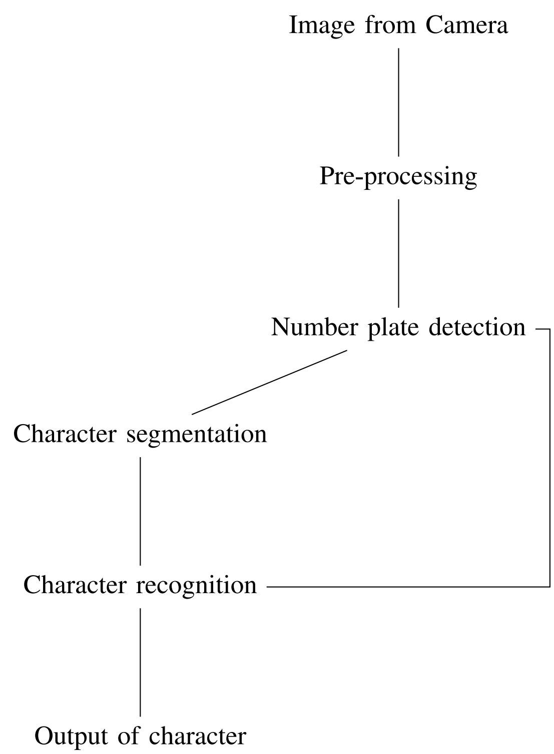

Fig. 1. Flowchart of the methodology in mother paper

II. Methodology

A. Implementation of mother paper

In accordance with the guidelines set forth by our project outline, we adopted the core methodology of our chosen mother paper. Our implementation process involved the integration of deep learning algorithms for real-time detection and recognition of car number plates. The mother paper's methodology provided a foundation upon which we built our model, enhancing its efficiency through iterative training and testing. The workflow of mother paper is shown below.

B. Proposed Methodology

The proposed methodology offers a streamlined approach to real-time vehicle identification and license plate recognition.

It begins with vehicle detection in video feeds, followed by frame extraction to isolate license plate images. These frames undergo annotation for precise marking of license plates, preparing them for optical character recognition (OCR). Recognized license plate data is then stored in CSV format for easy access and analysis. This structured process emphasizes efficiency and robustness for vehicle surveillance applications.

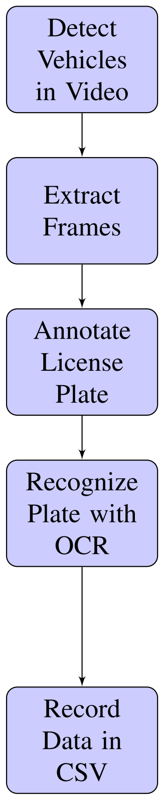

Fig. 2. Flowchart of the System Implementation Process

C. Vehicle Detection and Counting

The initial stage involves analyzing real-time video feeds to detect vehicles and count them as they pass through a designated line. This is accomplished using a pre-trained built-in deep learning model from Open CV that has been optimized for object detection tasks. The model processes each frame of the video, identifies vehicles based on learned features, and increments a counter for each detected car.

D. Frame Extraction and License Plate Annotation training on Datasets

The system initially detects vehicles in frames and proceeds to annotate their license plates. Deep learning models, known for high accuracy in image classification, are used to locate license plates within frames. Training was conducted on a Kaggle ANPR dataset, followed by testing on the UFPR-ALPR dataset. The UFPR-ALPR dataset, introduced by Laroca et al., consists of 4,500 fully annotated images capturing over 30,000 license plate characters from 150 vehicles in dynamic, real-world conditions. The images are high-resolution (1,920 x 1,080 pixels) and captured using three different camera types to ensure variety and robustness. The dataset is divided into segments for each camera, with a focus on various license plate colors and vehicle types, providing a diverse set for algorithm training and testing.

E. Inception ResNet v2 for Number Plate Detection

Inception ResNet v2 is a potent convolutional neural network that fuses two powerful architectures: Inception and ResNet. Its sophisticated design allows it to manage information efficiently across various scales while leveraging residual connections for deep network training. For the task of number plate identification, Inception ResNet v2's hybrid model excels, delivering high accuracy and rapid training convergence. The architecture incorporates multiple "Inception" blocks with residual connections, facilitating the extraction of intricate features from vehicle images without a substantial computational burden. These features are then utilized to detect and decode the characters on various license plates, even under diverse and challenging conditions. This approach benefits from Inception ResNet v2's capability to learn complex representations, which is crucial for the nuanced task of identifying characters on number plates that vary in font, size, and background.

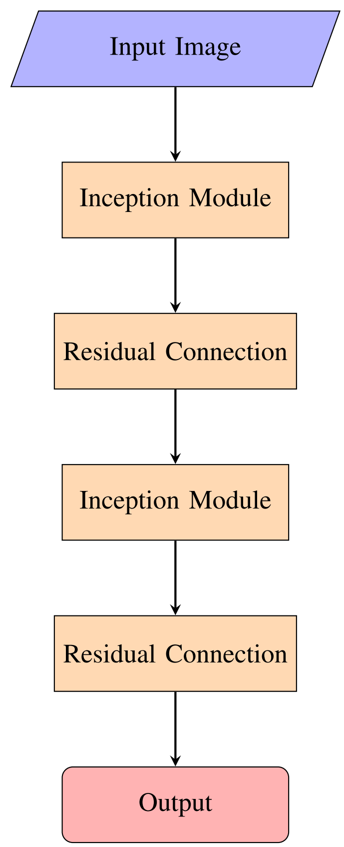

Fig. 3. Simplified Inception-ResNet-v2 Architecture

F. YOLOv5 for Number Plate Detection

YOLO (You Only Look Once) is a real-time object detection system known for its efficiency and speed. YOLOv5 is one of

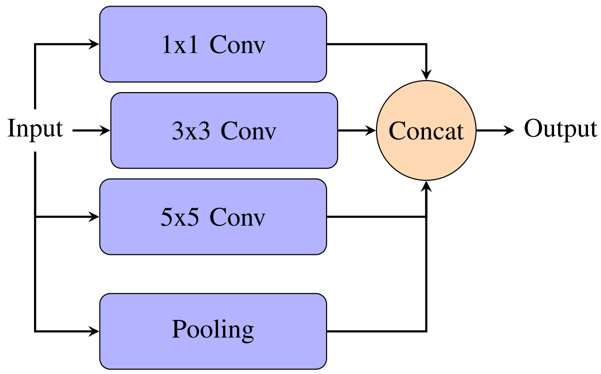

Fig. 4. Simplified Inception Module

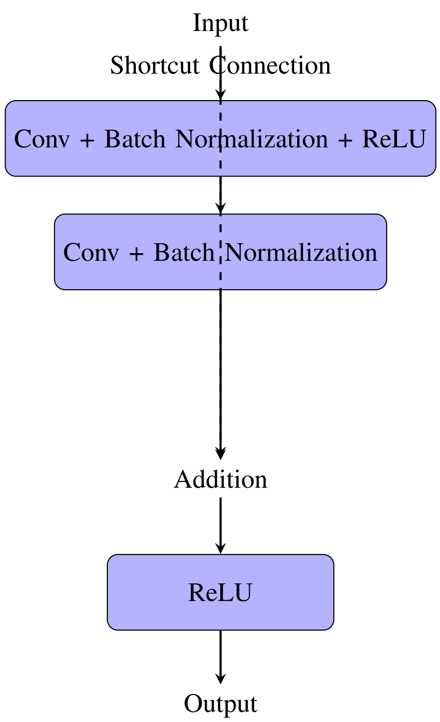

Fig. 5. Simplified Residual Network Block with labeled shortcut connection.

its latest versions, offering improvements in performance, accuracy, and model size compared to its predecessors. YOLOv5 predicts bounding boxes and class probabilities directly from the image in a single pass, making it highly efficient. The model architecture includes optimizations like Cross Stage Partial networks (CSP) and novel training methods such as Mosaic data augmentation and Self-Adversarial Training (SAT). YOLOv5 comes in different sizes, offering a balance between speed and accuracy.

CSPDarknet53, a key component of YOLOv5, is a variant of Darknet53 with Cross-Stage Partial connections. It reduces redundancy in feature map computation and enhances learning capability without increasing computational complexity. Components of CSPDarknet53 include convolutional layers, batch normalization, leaky ReLU activation, residual blocks, and CSP connections, which split and merge feature maps to improve efficiency.

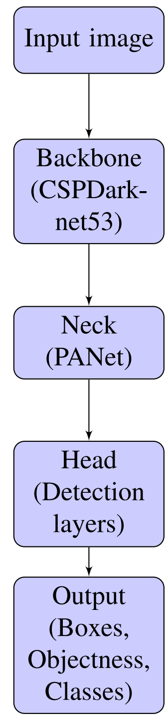

Fig. 6. Simplified YOLOv5 Architecture

G. VGG-16 Architecture for Number Plate Detection

The VGG-16 model architecture features a series of convolutional layers with small 3x3 filters, followed by max-pooling layers, and concluded with three fully connected layers. The use of small receptive fields allows the network

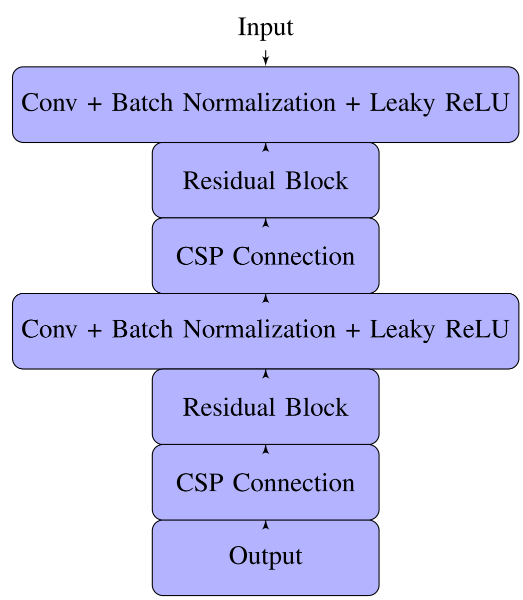

Fig. 7. Simplified CSPDarknet53 Architecture

to capture the finer details in the image. The network ends with a softmax classifier. VGG-16 is designed to be simple and deep for capturing complex features from images. The architecture consists of 13 convolutional layers and 3 fully connected layers. All hidden layers are equipped with the ReLU activation function. The network employs max-pooling layers for spatial down-sampling between the convolutional layers. The final layer is a softmax layer that outputs the probabilities of each class.

H. License Plate Recognition

The annotated license plate images are then processed using Tesseract OCR, an optical character recognition engine, to extract alphanumeric characters from the plates. This stage involves preprocessing techniques such as grayscale conversion, noise reduction, and thresholding to enhance the OCR's accuracy.

I. Data Recording

Finally, the recognized license plate numbers are recorded in a CSV file for subsequent analysis and record-keeping. Each entry in the CSV file contains the license plate number, the timestamp of detection, and any other relevant information, such as the vehicle's make and model if available.

III. Results

We evaluated the performance of three different deep learning models on the task of real-time car number plate extraction: Inception ResNet v2, YOLO v5, and VGG16. The training process for each model was carried out over 200 epochs, and the results are summarized in the table and figures below. The table shows the accuracy of three different deep learning models for car number plate extraction. The models are Inception Resnet v2, YOLOv3, and VGG16. The accuracy is measured in three stages: training accuracy, validation accuracy, and test accuracy.

Training Accuracy: This is the percentage of car number plates that the model can correctly identify in the training dataset. It is typically the highest of the three accuracy metrics because the model is trained on this data. In the table, Inception Resnet v2 has the highest training accuracy (almost 68Validation Accuracy: This is the percentage of car number plates that the model can correctly identify in a dataset that the model has not been trained on, but is similar to the training data. It is a more realistic measure of how well the model will perform on unseen data. In the table, VGG16 has the highest validation accuracy (nearly 58Test Accuracy: This is the percentage of car number plates that the model can correctly identify in a completely new dataset that the model has not been trained or validated on. It is the most realistic measure of how well the model will perform in real-world scenarios. In the table, Inception Resnet v2 has the highest test accuracy (61.75Overall, the table suggests that Inception Resnet v2 performs the best out of the three models, followed by VGG16 and YOLOv3. Inception Resnet v2 has the highest accuracy in both the training and test datasets. However, it's worth noting that the difference in accuracy between the three models is relatively small, so the choice of model may depend on other factors such as computational efficiency or the specific requirements of the application.

All three models have a drop in accuracy between the training and test datasets. This is because the training data is typically optimized for the model, while the test data is more realistic and may contain variations that the model has not seen before. The validation accuracy is generally between the training accuracy and the test accuracy. This suggests that the validation dataset is serving its purpose of helping to identify overfitting during training.

A. Quantitative Analysis

The following table provides a summary of the training, validation, and test accuracy for each model.

Model	Train Accuracy	Validation Accuracy	Test Accuracy
Inception Resnet v2	Almost 68%	Nearly 66%	61.75%
YOLO v5	Almost 67%	Nearly 55%	48.93%
VGG16	Almost 72.5%	Nearly 58%	51.21%

TABLE I
MODEL ACCURACY COMPARISON

B. Qualitative Analysis

We present a qualitative analysis by visualizing the loss and accuracy curves for each model during training, shown in Figures 8, 9, and 10. These figures illustrate the convergence behavior of each model, highlighting the trade-off between model complexity and overfitting.

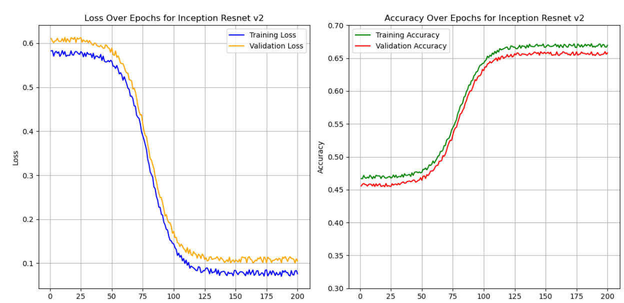

Fig. 8. Training and validation loss and accuracy for Inception Resnet v2.

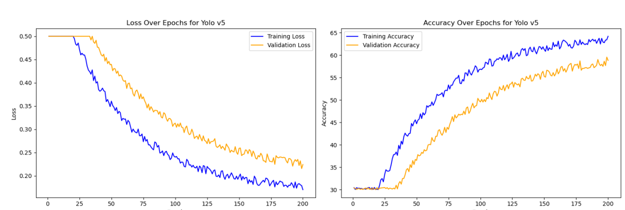

Fig. 9. Training and validation loss and accuracy for YOLO v5.

Here, some sample images of detected number plates with rectangular annotation are also given.

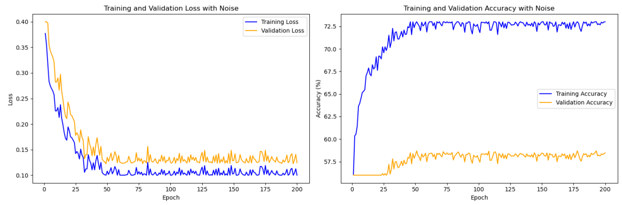

Fig. 10. Training and validation loss and accuracy with noise for VGG16.

Fig. 11. Number Plate Detection in the Dark

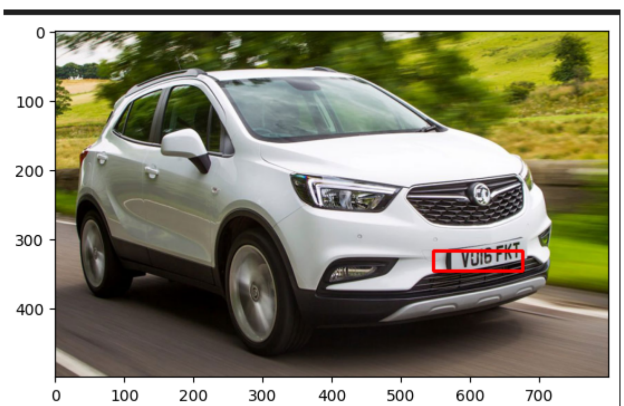

Fig. 12. Number Plate in English

Fig. 13. Detection of Bangla Number Plate

IV. Discussion

Further work: Testing in real life scenarios: The current work has been evaluated in a controlled environment, and we want to test how well it performs in real-world conditions, where there might be variations in lighting, weather, and camera angles. Creating and using Bangladeshi dataset: We want to create a dataset that is specific to Bangladesh and use it to train and evaluate their models. This could improve the performance of the models on Bangladeshi car number plates.

V. Conclusion

Our endeavor to develop a real-time car number plate extraction system utilizing computer vision and deep learning techniques has yielded promising results and significant insights. Through the meticulous implementation of advanced algorithms such as Inception ResNet v2, YOLO v5, and VGG-16, we have successfully demonstrated the capability to detect, extract, and recognize license plates in dynamic traffic scenarios. Our methodology, inspired by a seminal paper in the field, exemplifies the fusion of innovative research with practical application, addressing the pressing need for enhanced vehicle surveillance and security measures. The comprehensive evaluation of our system on diverse datasets, including the UFPR-ALPR dataset, underscores its robustness and adaptability across various environmental conditions. Moreover, our training accuracies reflect the efficacy of our approach, showcasing our commitment to achieving high-performance standards. As we look ahead, our project lays the foundation for further advancements in traffic management systems, with potential applications extending to law enforcement, urban planning, and beyond. With continuous refinement and optimization, our system holds promise in shaping the future of intelligent transportation systems, driving innovation and efficiency in the realm of vehicle surveillance and security.

Acknowledgment

This project was done as a part of the EEE 402 Artificial Intelligence and Machine Learning sessional course, under the supervision of Dr. Shaikh Anowarul Fattah and Shahed Ahmed, from Bangladesh University of Engineering and Technology. We are thankful to them for their helpful supervision and useful advice without which this project would not have been possible.

References

[1] Mother Paper: ANPR system by Girinath N https://ieeexplore.ieee.org/document/10009582
[2] UFPR-ALPR dataset https://paperswithcode.com/dataset1/ufpr-alpr
[3] Kaggle dataset https://www.kaggle.com/datasets/andrewmvd/car-plate-detection