Traffic Congestion Prediction using Deep Convolutional Neural Networks: A Color-coding Approach

By

Imrez Ishraque (180021211)

Md. Sumit Hasan (180021239)

Md. Sifath Al-Amin (180021319)

A Thesis Submitted to the Academic Faculty in Partial Fulfillment of the Requirements for the degree of

Bachelor of Science in Electrical and Electronic Engineering

Department of Electrical and Electronic Engineering

Islamic University of Technology (IUT)

Gazipur, Bangladesh.

June 203

CERTIFICATE OF APPROVAL

The thesis titled "Traffic Congestion Prediction using Deep Convolutional Neural Networks: A Color-coding Approach" submitted by Imrez Ishraque (180021209), Md. Sumit Hasan (180021239) and Md. Sifath Al-Amin (180021319) has been found as satisfactory and accepted as partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering on

Approved by:

____

(Signature of the Supervisor)

Mirza Fuad Adnan

Assistant Professor

Department of Electrical and Electronic Engineering (EEE)

Islamic University of Technology (IUT)

Declaration of Authorship

This is to certify that the work presented in this thesis paper is the outcome of research carried out by the candidate under the supervision of Mirza Fuad Adnan, Assistant Professor, Department of Electrical and Electronic Engineering (EEE), Islamic University of Technology (IUT). It is also declared that neither this thesis paper nor any part thereof has been submitted anywhere else for the reward of any degree or any judgment.

Authors

____

Imrez Ishraque

ID-180021211

____

Md. Sumit Hasan

ID-180021239

____

Md. Sifath Al Amin

ID-180021319

Dedicated to

Our beloved parents & teachers whose support made it all possible for us

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

ReLU	Rectified Linear Unit (ReLU)
SBM	Stochastic block model (SBM)
SBO	Structured Bayesian optimization (SBO)
SBSE	Search-based software engineering (SBSE)
SCH	Stochastic convex hull (SCH)
SGD	Stochastic Gradient Descent (SGD)
SGVB	Stochastic Gradient Variational Bayes (SGVB)
SMBO	Sequential Model-Based Optimization (SMBO)
SSVM	Smooth support vector machine (SSVM)
SVD	Singular Value Decomposition (SVD)
SVM	Support Vector Machine (SVM)

ACKNOWLEDGMENTS

Foremost, we would like to express our sincere gratitude and gratefulness to the Almighty Allah; without His graces and blessings, this study would not have been possible.

Acknowledging all who helped us complete this work, we wish to compliment the university's significant role and the department that has been very amiable to us during the entire period of our research

We are indebted to our honorable supervisor, Mirza Fuad Adnan sir, for his selfless support, motivation, patience, enthusiasm, and extensive knowledge of the relevant fields. His continuous guidance and careful supervision kept us going even during the hardest of hours. Lastly, our warmest tribute to our parents, family members, and friends, whose moral support and well wishes benefitted us spiritually in achieving our goals

ABSTRACT

Traffic video data has become a critical factor in limiting traffic congestion due to recent advancements in computer vision. This work proposes a unique technique for traffic video classification using a color-coding scheme before training the traffic data in a deep convolutional neural network. At first, the video data is transformed into an imagery data set, and vehicle detection is performed using the You Only Look Once algorithm. A color-coded scheme has been adopted to transform the imagery dataset into a binary image dataset. These binary images are fed to a deep convolutional network. Using the UCSD dataset, we have obtained a classification accuracy of 98.2%.

CHAPTER-1

Introduction

Fig 1.1: Traffic Congestion

Urbanization's acceleration speeds up traffic problems, resulting in economic losses and the immobilization of urban functions [1]. The effects of traffic congestion extend to individuals as well. Some significant effects of traffic congestion are the abundant waste of time, particularly during peak hours, mental fatigue, and additional pollution, which contribute to catastrophic natural outcomes. The cost of traffic congestion in the four nations of France, Germany, the United Kingdom, and the United States is expected to increase by 55 billion US dollars by 2030 [2]. A nation cannot progress without ensuring economic growth and the comfort of its road users, which is impossible without efficient traffic flow. The ability to foresee traffic congestion

gives officials and consumers the necessary time to allocate resources to ensure that travelers' journeys go smoothly. Consequently, a generic and widely applicable traffic congestion-detecting system is currently required [3].

1.1 Spot-based sensor

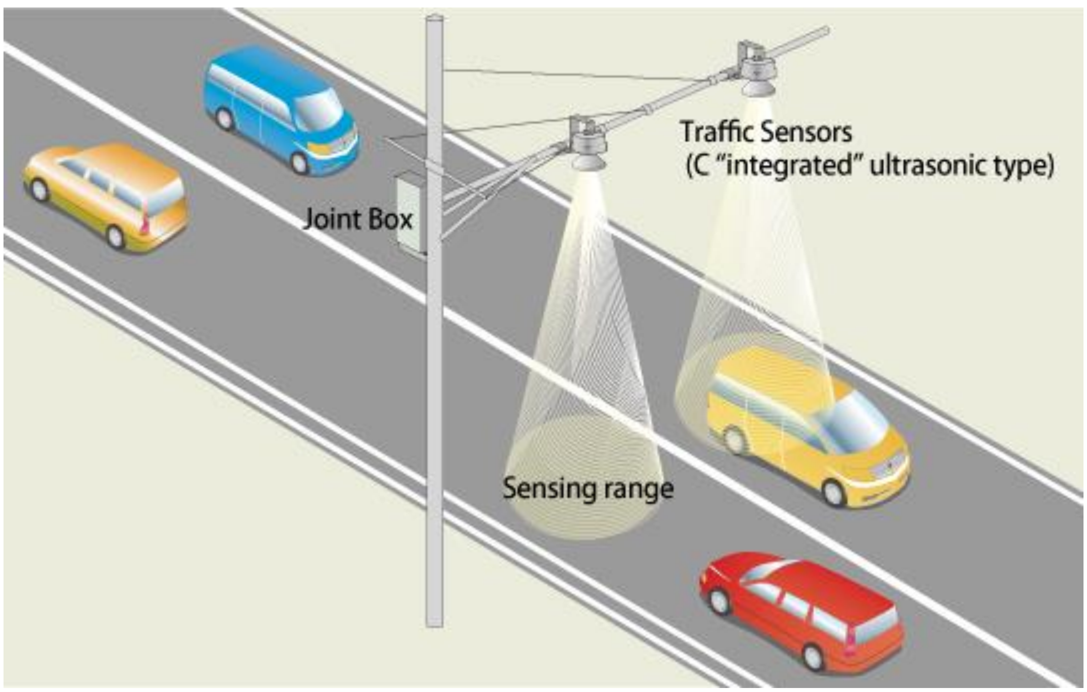

Fig 1.2: Spot-based sensor

Modern information and communication technologies and the Internet of Things have contributed to the development of intelligent transportation systems (ITSs), which have enabled the application of traffic forecasting (TF) techniques [4]. Starting with a theoretical perspective and progressing to data-driven methodologies, an abundance of research has been undertaken on traffic congestion forecasting and relevant topics. Traditionally, spot-based sensors are used for traffic estimation. Whenever a vehicle stops over a loop or goes by the loop, the sensor will count the number of vehicles that have passed. Inductive loop sensors, piezoelectric sensors, and magnetic loops are widely used technologies in the traffic estimation eco. system [5], But these sensors are costly, which makes them hard to implement on a wide scale. Also, these spot

sensors can only quantify or measure traffic flow or address certain subtasks (e.g., traffic queue measurement and traffic density detection). As a result, this cannot be a sustainable generic model to apply [6]. On top of that, recent developments in infrared and laser radar sensors have prompted the gradual replacement of conventional spot-based sensors with most of these devices [5].

1.2 GPS

Fig 1.3: GPS in cars

Apart from these sensor-based technologies, a more affordable method of gathering network-wide traffic data is emerging, utilizing Global Positioning System (GPS) devices. Through tracking vehicle trajectories, in-vehicle GPS technology enables recording vehicle speed and location at a given time. It allows them to follow vehicle trajectories and evaluate traffic condition performance in a broad network at a reasonable cost, making them popular in large-scale research. However, this approach has certain disadvantages; firstly, speed is the only considered parameter, which can sometimes lead to the loss of confidential information from unprocessed GPS data and an inaccurate assessment of traffic congestion [7]. Secondly, as

GPS-based intelligent traffic systems depend on the number of vehicles employing GPS, the detection precision of traffic updates decreases drastically when the number of GPS-equipped vehicles starts to decline significantly [1]. Additionally, Applying GPS-based data to detect traffic congestion on arterial roads will become a complicated scenario because of manual manipulations [7].

1.3 Vision-Based Detection:

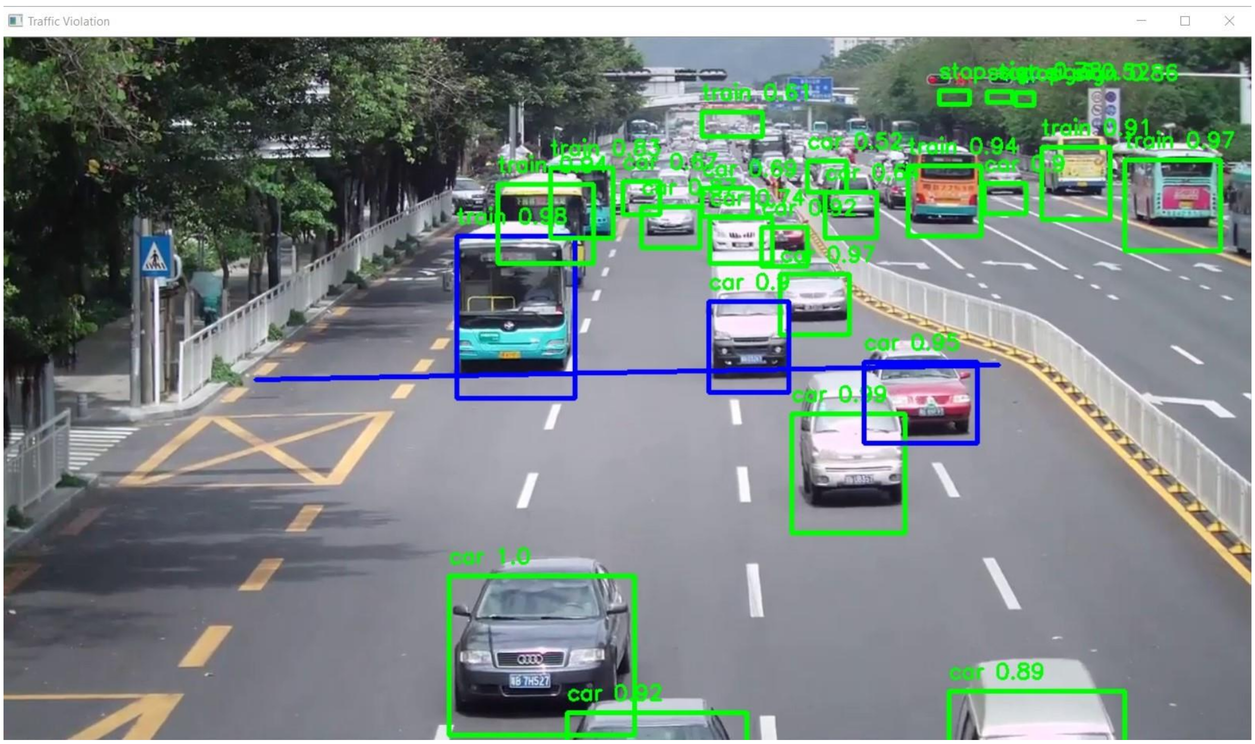

Fig 1.4: Vision-Based Traffic Detection

On the other hand, TF has shifted from a traffic theory-based approach to a data-driven standpoint due to the increased diversity and number of available traffic data provided by ITS [4]. Vision-based detection technologies

have improved significantly in recent years. Advanced image processing algorithms and object detection techniques have created a new opportunity for vision-based intelligent traffic management systems [8]. Different statistical forecasting approaches, shallow machine learning algorithms, and deep learning methods have achieved impressive precision in these classification tasks [3]. Principal benefits of these vision-based approaches:

• They do not rely on picking features manually. Consequently, this eliminates the restrictions on systems utilizing camera images for traffic-state evaluation and forecasting [5].

• Short-term to long-term estimation of traffic congestion (starting from a few minutes to even for multiple hours).

• This detection system can automatically communicate with other entities of a traffic network, thereby contributing to a more optimal environment for traffic management [9].

• More information and parameters about traffic can be considered to increase the system's efficacy [3].

• A higher number of generic data solutions with minimized cost and efficient performance [3].

• Easy maintenance and can continuously be updated and driven into a more authentic version with relevant sets of recent data [1].

CHAPTER-2

Literature Review

2.1 Relevant Research

Research has been conducted to estimate traffic situations by fusing multiple data sources for traffic state estimation with visual inputs integrated with machine learning and deep learning approaches. Akhtar and Moridpour [3] show a direct comparison between shallow machine learning (SML) algorithms and deep machine learning (DML) algorithms by analyzing several notable research works. Most of these works evaluated five parameters: *traffic occupancy, congestion volume, and vehicle density, along with the traffic congestion index and total travel time,* predicting and analyzing the overall traffic occupancy. After analyzing these works, it is evident that deep learning algorithms can assess large datasets efficiently. As a result, they proved it to be more effective tshallowhan In this field, shallow machine learning (SSML), neural networks (ANN), and machine learning vector machines (SVM) are used. P. Chakraborty et al. [5] used two deep neural networks: deep convolutional neural networks (DCNNs) and you only look once (YOLO) and compared these results with the port vector machine (SVM) model to analyze the advantages of deep learning models. To produce both short-term and long-term forecasts (from 5 minutes to up to 4 hours), T. Bogaerts et al. [4] built a hybrid deeper network that concurrently extracts the spatial aspects of traffic using graph convolution and its temporal features using Long Short Term Memory (LSTM) cells. In addition, they selected the most appropriate road linkages for both short- and long-term TF using a data reduction technique, which increased their efficiency. M.A.A. Al-qaness et al. [8] demonstrated an intelligent video surveillance-based vehicle tracking system that can recognize, track, and count vehicles in various situations by combining neural networks, image-based tracking, and You Only Look Once (YOLOv3). H. Cuii et al. [9] employ two convolutional networks, AlexNet and GoogLeNet, to characterize traffic congestion situations. The images in the dataset here were taken from traffic surveillance images. Despite having obscured visual features in some pictures

in the dataset, AlexNet and GoogLeNet showed a convincing result by successfully classifying the images and recognizing highway traffic congestion. In [1], leveraging a gray-level co-occurrence matrix, the multi-layered detection method first determines the density of surrounding items. Then the velocity of moving objects is determined by incorporating the Luca-Kana-e optical flow along with pyramid implementation. Then a Gaussian mixture model is applied to demonstrate the model, which was tuned by CNN. Crc3d is a suggested mapping in [7] to the cube framework to forecast the urban traffic pattern for the holistic network utilizing 3-dimensional convolution networks, convolutional neural networks (CNN), and recurrent neural networks (RNN). The architecture incorporates spatial and temporal dimensions, combining C3D and CNN-RNN. H. Nguyen [10] suggests an enhanced vehicle detection scheme based on an accelerated R-CNN. It used the mobile net architecture to build the convolution layer. A soft NMS algorithm was deployed to address the problem of redundant proposals. A Contcontextre ROI pooling layer was used to scale the suggestions to the required dimensions, and MobileNet architecture was used for constructing the classifier as the method's final step. A hybrid 2D–3D CNN model-based driving assistance system was developed in [11], which uses a transfer learning paradigm. In the first architecture, hybrid-TSR is designed to perform the duty of traffic sign recognition efficiently. The second framework, called hybrid-SRD, enables semantic road space class ending up-sampled of deconvolutional methods. D. Impedovo et al. [12] used feature extraction to classify the congestion Various objection detection algorithms were used for vehicle detection and feature extra extraction, which were then compared side by side by applying machine learning classifiers and deep learning methods to see what the best output was. It is visible in[13] ITSC,t ITSC or intelligent traffic control, control can be used to solve the traffic congestion problem efficiently Here, reinforcement learning was applied to vehicles detected by the ITSC system to reduce the average waiting time. R. Cucchiara et al. [6] propose a 2-level traffic monitoring system called VTTS (vehicular accident tracking system) based on vehicle detection and tracking. Vehicle detection is done using Dedicated Short-Range Communication

(DRSC) structures in intersections rather than a camera or loop detector. The reduced image processing components harvest visual data under different situations, while the elevated units track the vehicles.

2.2 Comparative Analysis of Relevant Research

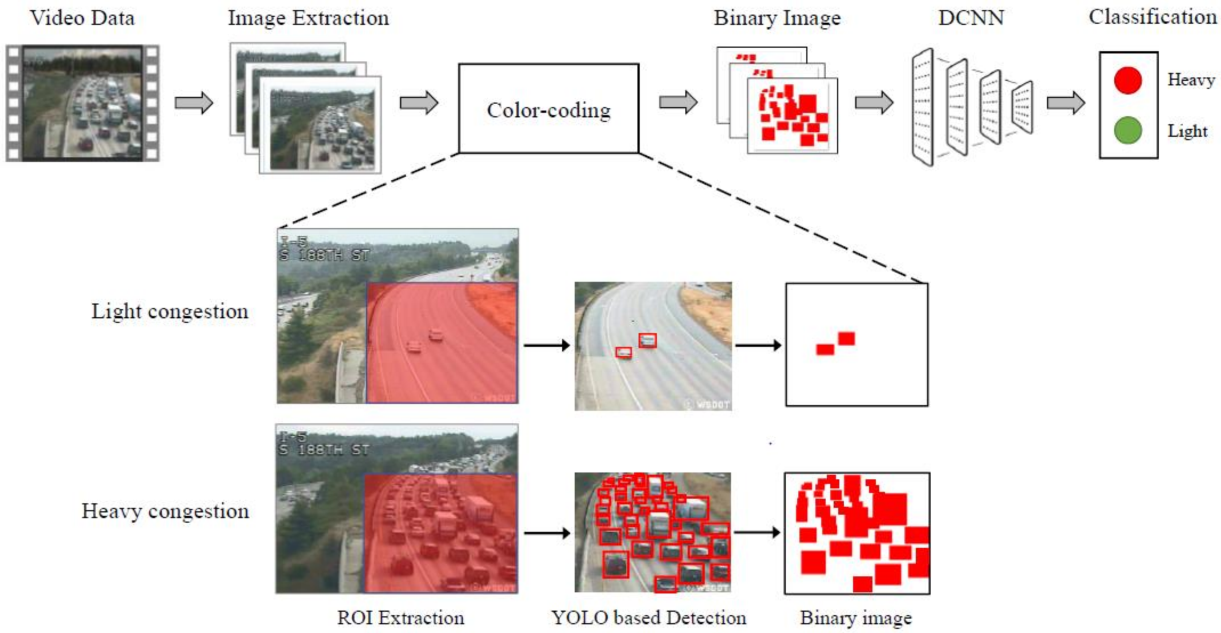

Fig 2.1: Framework of color coding-based Traffic congestion Detection

In this paper, we are interested in investigating a new color-coding-based scheme to train a deep convolutional neural network (DCNN) and evaluate its performance in forecasting traffic congestion at sites that differ significantly from the training data set. The general outline of our proposed scheme is illustrated in Fig. 1.

We have used the UCSD highway traffic data to train our proposed DCNN. A binary imagery dataset of highway traffic was built and labeled with around 3000 congestion and 1000 non-congestion images. The performance of DCNN has been investigated using the binary image dataset. DCNN achieved 98.2% classification accuracy. We also validate our scheme by feeding test images from diverse sites; almost 97% of the cases have been precisely recognized. Our proposed color-coding scheme carmaker this congestion challenge more independent of the dataset.

CHAPTER-3

Methodology

3.1. Basic methodology

To train a deep convolutional neural network (DCNN) and assess how well it performs in predicting traffic congestion at sites that considerably deviate from the training data set, we are interested in researching a novel color-coding-based method. Fig. 1 shows the main layout of the suggested strategy we have created.

We trained our suggested DCNN using the UCSD highway traffic data. Approximately 3000 photos of traffic congestion and 1000 photographs of non-congestion were used to create a binary imaging collection of highway traffic. The binary image dataset has been used to examine DCNN performance.

Vehicle detection is the most crucial step in this congestion detection framework. The purpose is to portray the number of vehicles in the road segment accurately. Vehicle detection from Images consists of two parts, which include image classification and object localization. Using image classification, we can classify the image into one of the classes, and using object localization, we can find the location of the object within the image. So, our X_train data are the images we obtained from the dataset after preprocessing and Y_train data are vectors each of size 7 for each image that will be used as input for our CNN. The output we will get is a vector of size 7 which will specify whether the object we want to detect is in the image are not, the bounding box size, and the center of the location of the detected object To handle multiple vehicles in an object, Yolo divides the image into a grid of smaller images, where detection will occur only if the center of the bounding box is within a particular grid. This will cancel out all other grids except the ones containing the object for detection. After the vehicle detection, we employ color coding to turn the image into a binary image, where the red boxes will indicate the vehicles' bounding boxes and the rest of the portion will be turned white. The study utilized a traditional convolutional neural network (ConvNet) architecture, specifically a deep convolutional neural network (DCNN), which is considered a leading approach for image classification. The UCSD dataset used in the study consists of 254 highway traffic videos, each containing 40–50 frames with a resolution of 320 x 240 pixels.

To extract relevant data, 4-5 frames from each video were selected, and a region of interest (ROI) was manually defined to remove unrelated external objects from the scenes. This step aimed to focus on the traffic-related information within the images. Subsequently, the trimmed images were resized to 180x180 pixels to mitigate memory allocation challenges during the model training process

3.2. Description of basic methodology

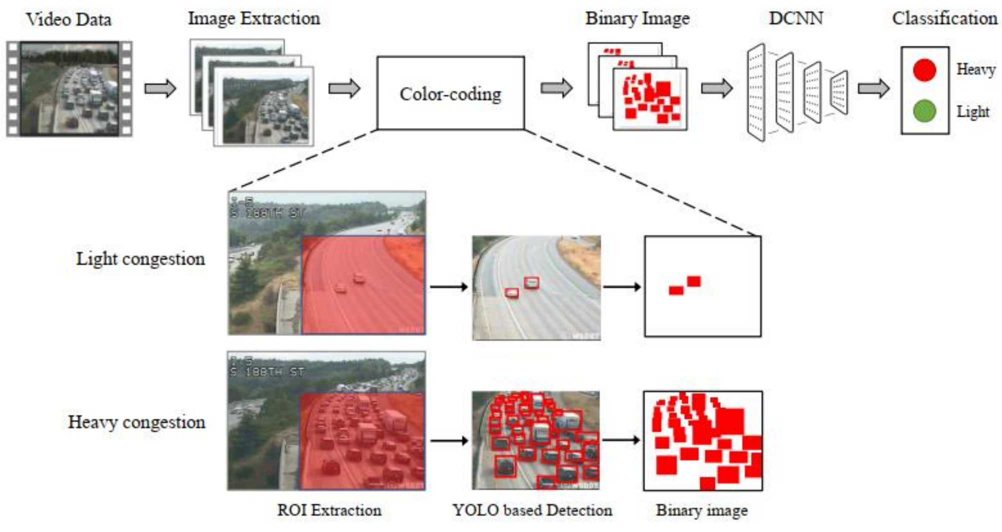

Fig 3.1: Framework of color-coding based Traffic congestion Detection

Our team embarked on training a deep convolutional neural network (N) to address the issue of traffic congestion using highway traffic data. To create a comprehensive binary image dataset specifically focused on highway traffic, we gathered approximately 3000 photos depicting instances of traffic congestion and 1000 photographs capturing non-congested traffic scenarios. Utilizing this dataset, we trained our DCNN model to classify images as either congested or non-congested traffic. The goal was to develop a robust system to accurately identify and differentiate between the two conditions.

Remarkably, our DCNN model achieved an impressive classification accuracy of 98.2% during the training phase. This accuracy metric demonstrates the model's capability to effectively discern between congested and non-congested traffic instances based on the visual features captured in the images.

To further evaluate and validate the performance of our system, we conducted testing using a diverse set of test photos sourced from various sites. Through this process, we aimed to assess the model's generalization and ability to accurately classify instances of congestion in real-world scenarios.

The results of our testing were highly encouraging. Over 97% of the instances presented in the test photos were correctly identified by our DCNN model. This outcome suggests that our trained model demonstrates a strong ability to generalize its learnings to new and diverse traffic situations, allowing it to effectively classify congestion accurately.

One notable aspect of our approach is the development of a suggested color-coding system. This system complements the DCNN model and aims to enhance the dataset independence of the congestion problem. By utilizing specific color codes, we can create a standardized representation of traffic congestion across different datasets and environments.

The suggested color-coding system offers a practical way to make the analysis of congestion more consistent and comparable across different sources of data. It provides a common visual language that can be applied universally, enabling researchers, transportation planners, and stakeholders to assess congestion levels in a dataset-independent manner.

By incorporating this color-coding system into our approach, we aim to establish a standardized framework for analyzing traffic congestion. This framework, combined with the high classification accuracy of our DCNN model, enhances the reliability and applicability of our solution across various datasets and real-world scenarios.

In summary, through training a DCNN model using the UCSD highway traffic data, we have achieved a remarkable classification accuracy of 98.2%. By further validating the system with diverse test photos, we have demonstrated its ability to accurately identify instances of congestion, surpassing an accuracy rate of 97%. The introduction of our suggested color-coding system adds a layer of dataset independence, enabling consistent analysis and comparison of congestion levels across different sources of data.

In the congestion detection framework, vehicle detection plays a crucial role as it accurately portrays the number of vehicles present in a road segment. Vehicle detection from images typically involves two main components: image classification and object localization. Image classification enables the classification of an image into predefined classes, while object localization helps determine the location of the detected object within the image.

In this study, the preprocessed images obtained from the dataset serve as the X_train data. These images undergo preprocessing steps such as resizing and region of interest (ROI) selection to focus on relevant traffic information. On the other hand, the Y_train data consists of vectors, with each vector having a size of 7. These vectors are used as input for the Convolutional Neural Network (CNN) model.

The output of the model is also a vector of size 7, which provides information about the presence of the detected object in the image, the bounding box size, and the center of the detected object's location. To handle multiple vehicles in an image, the study utilizes the YOLO (You Only Look Once) approach, which divides the image into a grid of smaller regions. Detection occurs if the center of the bounding box falls within a particular grid. This approach ensures that only the grids containing the object of interest are considered for detection, ignoring the other grids.

After vehicle detection, the study employs color coding to transform the image into a binary representation. In this representation, the vehicles' bounding boxes are highlighted in red, while the rest of the image is turned into white. This color-coded visualization enhances the understanding of the vehicle's presence within the image and aids in congestion analysis.

Overall, the study combines the traditional ConvNet architecture, particularly a Deep Convolutional Neural Network (DCNN), with vehicle detection techniques to classify and localize vehicles accurately in highway traffic images. The utilization of the UCSD dataset, comprising highway traffic videos with 40-50 frames per video and a resolution of 320x240 pixels, ensures the availability of relevant data for training and evaluation. The preprocessing steps, such as frame selection, ROI definition, and image resizing, contribute to addressing memory allocation challenges and improving the model's performance during training.

3.3. Description of performance matrices in ml/dl

To evaluate the performance of the model, traditional performance metrics such as precision, recall, and accuracy were used. To understand these metrics, it is important to define the concepts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

True positive (TP) refers to cases where a congested image was correctly labeled as congested by the model. These are instances where the model successfully identified and classified the presence of traffic congestion.

True negative (TN) represents the cases where a non-congested image was correctly labeled as non-congested by the model. These are instances where the model accurately recognized and classified the absence of congestion.

False positive (FP) occurs when a congested image is predicted to be non-congested by the model. In other words, the model incorrectly identifies a non-congested image as congested.

False negative (FN) happens when a non-congested image is predicted to be congested by the model. It means the model fails to recognize and correctly classify a non-congested image as non-congested.

These definitions allow us to calculate the performance metrics for the model. Precision is the ratio of true positives (TP) to the sum of true positives and false positives (TP + FP). It measures the accuracy of the model's positive predictions (congested) among all the instances it labels as positive.

Recall, also known as sensitivity or true positive rate, is the ratio of true positives (TP) to the sum of true positives and false negatives (TP + FN). Recall measures the ability of the model to correctly identify all the positive instances (congested) among all the positive instances.

Accuracy is the overall correctness of the model's predictions and is calculated as the ratio of the sum of true positives and true negatives (TP and TN) to the total number of instances.

By evaluating the model's performance using these metrics, it provides a quantitative assessment of its ability to accurately classify congested and non-congested images. These metrics help assess the model's effectiveness in capturing and predicting instances of traffic congestion, allowing for further analysis and improvement if necessary.

Confusion Matrix		Predicted	
		True	False
Actual	True	True Positive (TP)	False Negative (FN)
	False	False Positive (FP)	True Negative (TN)

3.4 formula for performance matrices:

Now, Precision, Recall, and Accuracy can be found using following equations:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP}{TP + FP + TN + FN}$$

CHAPTER-4

Dataset Overview

4.1 Dataset:

The UCSD dataset comprises 254 daytime highway traffic videos, from which still images are extracted. The dataset includes diverse traffic patterns, excluding medium congestion types for consistent analysis. With a total of 4,110 labeled examples, the dataset offers a sufficient number of uncongested and congested images for training and evaluating congestion detection models.

4.2 Image extraction:

The UCSD dataset used in this study consists of 254 daytime highway traffic videos, which were recorded using a stationary camera. As the focus of the paper is on processing still images, images are extracted from these videos for analysis and model training. The dataset exhibits diverse traffic patterns, including light, medium, and heavy congestion scenarios.

To establish the ground truth for the dataset, hand-labeling has been performed, providing annotations that describe each video sequence. The labeling indicates whether the traffic in each image is congested or non-congested. However, in cases where it was challenging to determine whether the traffic was congested or not, the medium congestion types were excluded from the analysis.

After removing the medium congestion types, the dataset contains a total of 4,110 labeled examples. Among these, 2,990 images are labeled as uncongested, while 1,120 images are labeled as congested. This labeling process was conducted by two different annotators to ensure consistency and reduce subjective bias.

The resulting dataset provides a substantial number of labeled examples for training and evaluation of the models. It covers various levels of congestion, allowing for a comprehensive analysis of traffic patterns and the development of effective congestion detection models

4.3 Image Transformation and Class Expansion:

Data classification:

Vehicle detection is a crucial step in the congestion detection framework, as it aims to accurately determine the number of vehicles in a given road segment. Various techniques have been proposed for accurate vehicle detection, including frame differencing, optical flow, region-based convolutional neural networks (RCNN), and You Only Look Once (YOLO).

Previous vehicle detection algorithms typically utilized regions or bounding boxes to identify vehicles within an image. However, YOLO takes a different approach by using regression to predict classes and bounding boxes for the entire image in a single run. This makes YOLO one of the fastest vehicle detection techniques, as it requires only one round of image processing.

It's important to note that the accuracy of YOLO may decrease when two different vehicles are close to each other, as the model may have difficulty distinguishing between them. In this study, the YOLOv3 object detector is adopted, which consists of a convolutional neural (CNN) called Darknet. The Darknet CNN architecture comprises 24 convolutional layers that serve as feature extractors, along with two dense layers for prediction. By leveraging YOLOv3, which incorporates the Darknet CNN architecture, the study aims to achieve efficient and accurate vehicle detection for congestion analysis.

4.4 Image augmentation:

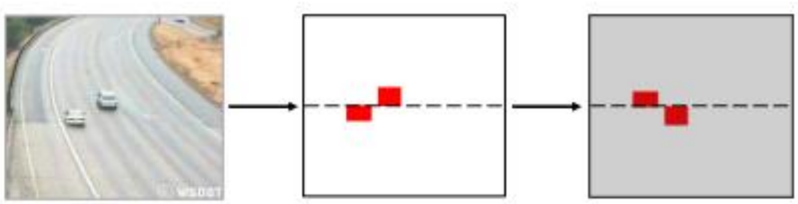

Fig. 4. Data Augmentation (Vertical Flip)

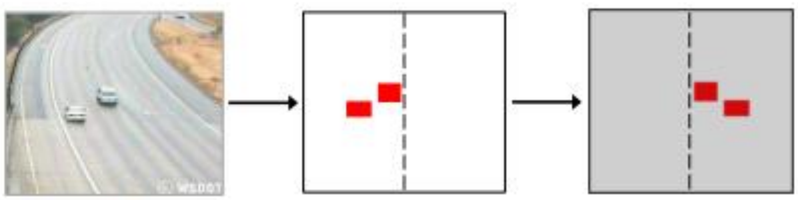

Fig. 5. Data Augmentation (Horizontal Flip)

Fig: 4.1: Data augmentation (vertical flip) and Fig 4.2: Data augmentation (horizontal flip)

Deep Convolutional Neural Networks (DCNNs) are known to be computationally expensive and typically require a large number of images for training in order to prevent overfitting. However, in this particular study, the researchers encountered a limitation in the amount of available data. Only 1,170 images could be generated from the 195 videos through frame extraction, resulting in around 1,200 images for training.

To address the limited dataset, the researchers employed data augmentation techniques and dropout regularization to mitigate the risk of overfitting. Data augmentation involves applying transformations or modifications to existing images to create additional training samples. In this study, horizontal flipping (as illustrated in Figure 5) and vertical flipping (as illustrated in Figure 4) were randomly applied to the sample images to expand the dataset. By flipping the images horizontally and vertically, the researchers introduced variations in the dataset, effectively increasing the diversity of the training samples. This helps to enhance the model's generalization capabilities and reduce the risk of overfitting the limited dataset.

By combining data augmentation techniques with dropout regularization, which randomly deactivates certain neurons during training, the researchers aimed to prevent the model from memorizing the specific training examples and instead encourage it to learn more robust and generalized features.

Overall, in this study, where the available dataset was limited, the researchers addressed the issue by applying data augmentation techniques such as horizontal and vertical flipping to generate additional training samples. This approach, along with dropout regularization, aimed to mitigate overfitting and improve the generalization capabilities of the deep convolutional neural network.

<u>CHAPTER-5</u>

Introduction to Algorithms

5.1: Vehicle Detection

Vehicle detection plays a pivotal role in the congestion detection framework as it aims to accurately portray the number of vehicles present in a given road segment. To achieve this, several techniques have been proposed in the literature. These include frame differencing [14], which compares consecutive frames to detect moving objects; optical flow [15], which tracks the apparent motion of objects; region-based convolutional neural network (R-CNN) [16], which uses region proposals to identify vehicles; and You Only Look Once (YOLO) [17].

While all the vehicle detection algorithms arms employ region-based approaches to identify vehicles within an image, YOLO takes a different approach by utilizing regression. YOLO predicts the classes and bounding boxes for the entire image in a single run, making it one of the fastest vehicle detection techniques. Its efficiency lies in requiring only one pass of image processing, thereby significantly reducing computational overhead. YOLO (You Only Look Once) v3 is a widely acclaimed object detection algorithm that has significantly advanced in real-time object detection. YOLO v3, developed by Yoloeph Redmon and his team, builds upon the success of its predecessors by introducing several key improvements. One notable enhancement in Yolo v3 is its ability to detect objects at different scales and resolutions. The network architecture incorporates a feature pyramid network, allowing the detection of objects at various levels of detail. This multi-scale approach enables Yolo v3 to detect both small and large objects accurately, making it highly effective in complex scenes with things of different sizes. Furthermore, YOLO v3 supports training on large-scale datasets, such as COCO (Common Objects in Context), enabling it to learn from diverse object categories. This extensive training allows YOLO v3 to generalize well to different object classes and perform excellently on various object detection tasks.

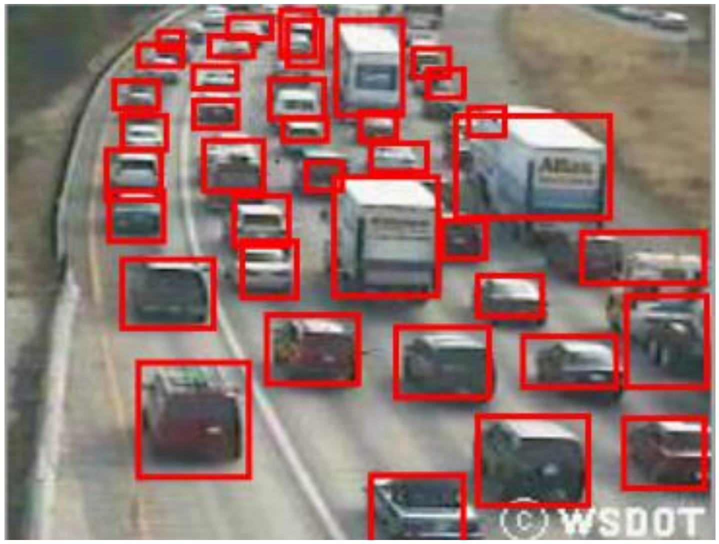

Fig 5.1: Vehicle Detection Using YOLOv3

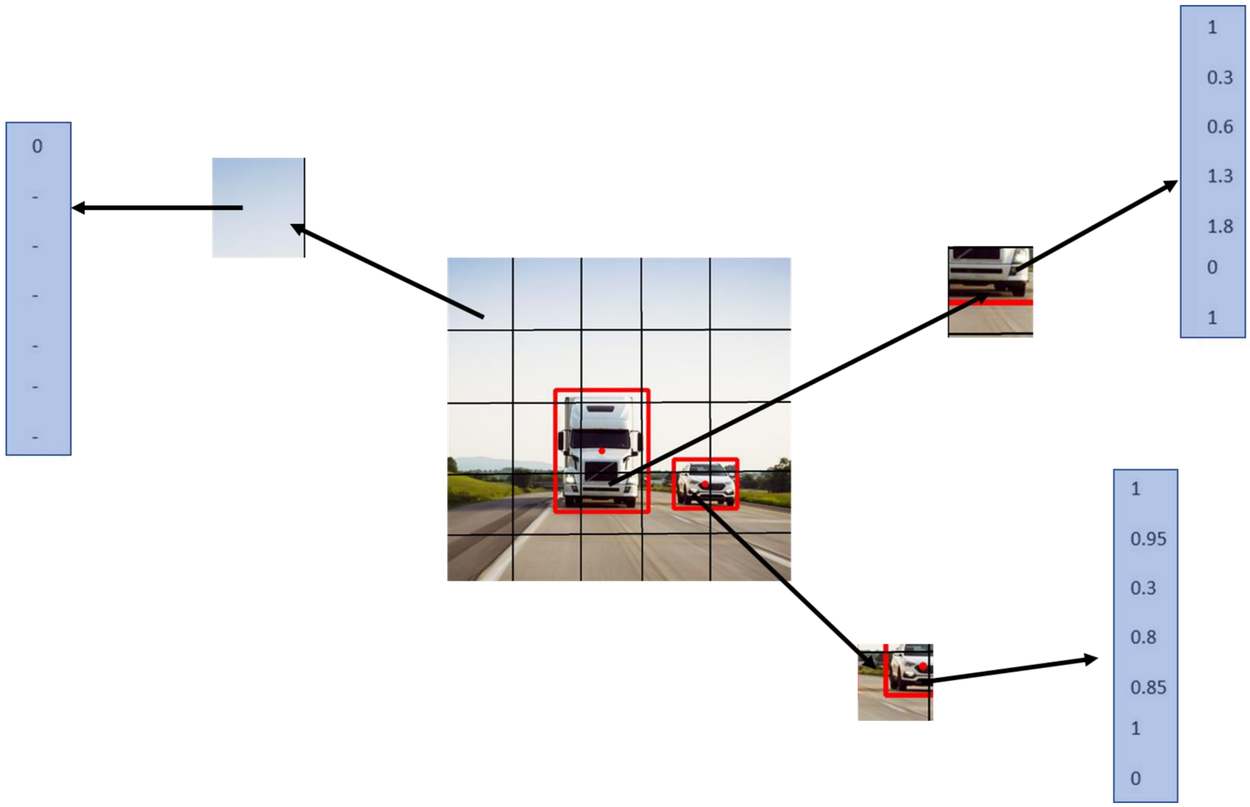

Fig 5.2: Detailed Object Detection Using YOLOv3

However, it is worth noting that YOLO's accuracy can be affected when two different vehicles are close to each other, as the bounding boxes may overlap and lead to incorrect predictions. Despite this limitation, YOLO, specifically YOLOv3, remains a popular choice for vehicle detection due to its speed and efficiency. YOLOv3 utilizes a CNN architecture called Darknet, comprising 24 convolutional layers that serve as feature extractors and two dense layers for prediction. The Darknet architecture, in conjunction with YOLOv3, enables accurate detection and classification of vehicles in real-time scenarios.

In the present study, we have adopted the YOLOv3 object detector with its underlying Darknet CNN architecture for vehicle detection. By leveraging the power of Darknet's 24 convolutional layers for feature extraction and the subsequent dense layers for prediction, our research aims to achieve precise and efficient vehicle detection in congested traffic scenarios. This methodology allows us to capture relevant features and accurately predict the presence and count of vehicles within a given road segment, thereby contributing to an improved understanding of traffic congestion dynamics.

Overall, using YOLOv3 and Darknet in our research enables us to address the critical vehicle detection task within the congestion detection framework. By incorporating this state-of-the-art technique, we aim to provide an accurate and efficient solution for quantifying the number of vehicles on the road, contributing to better traffic management and congestion analysis. So, YOLOv3 represents a significant advancement in object detection. With its multi-scale detection, Darknet-53 backbone, skip connections and non-maximum suppression, Yolo v3 offers improved accuracy and robustness in detecting objects of different sizes and complexities. Its real-time capabilities make it a valuable tool for many applications requiring efficient and accurate object detection.

5.2: Road Congestion Estimation

Deep Convolutional Neural Networks (DCNNs) are widely recognized as the state-of-the-art approach for image classification tasks. In this study, we employed a traditional ConvNet architecture consisting of convolutional and pooling layers. The UCSD dataset, comprising 254 highway traffic videos, served as the foundation for our research. Each tape contained 40–50 frames with a resolution of 320 x 240 pixels. We selected 4-5 frames from each video to extract the most informative frames and manually defined a region of interest (ROI) to exclude irrelevant external objects from the scenes.

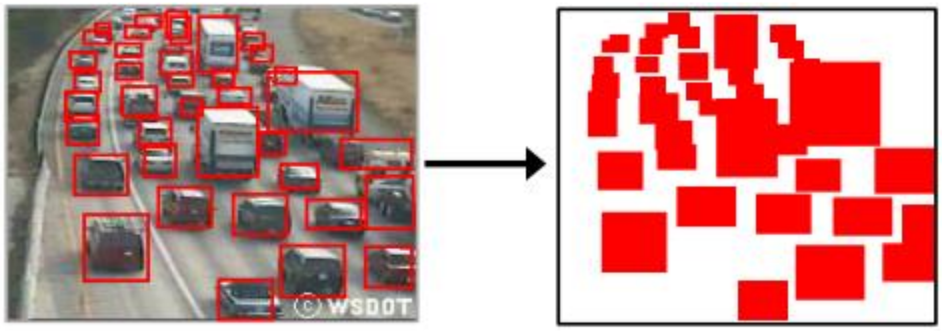

Fig. 5.3: Heavy Congestion Transformation to Binary image

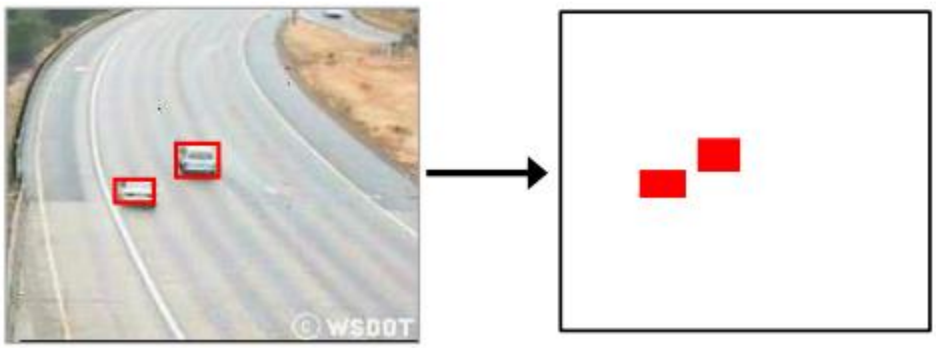

Fig. 5.4: Light Congestion Transformation to Binary image

To ensure efficient memory allocation during model training, the trimmed images were resized to 180 x 180 pixels. These resized images were then used as input for our model, consisting of two successive convolutional layers of 32x3x3, followed by a max pooling layer of size 2x2. The network was further extended with two additional convolutional layers of size 64x3x3, followed by another max pooling layer of the same size. To mitigate the risk of overfitting, a dropout layer with a probability of 0.25 was introduced after each max pooling layer. The rectified linear unit (ReLU) activation function was utilized throughout the model.

Deep Convolutional Neural Networks (DCNNs) are computationally demanding and typically require many images for practical training and to mitigate overfitting. However, in our study, we needed more data. Only 1170 images could be generated from the 195 videos through frame extraction, making approximately 1200 images available for training. To overcome this limitation, we incorporated the data augmentation technique [19] in conjunction with dropout regularization to enhance the generalization capabilities of the model and prevent overfitting. Sample images were randomly flipped horizontally and vertically as part of data augmentation. These augmentations increased the diversity and variability of the training data, enabling the model to learn robust and generalized representations of the traffic scenes.

The model training was conducted on an NVIDIA Quadro M1000M GPU with 8GB of RAM, taking approximately 30 minutes to complete. This powerful computational resource expedited the training process, allowing us to optimize the model's performance and validate its efficacy in traffic image classification.

In summary, our research leveraged a ConvNet architecture within the domain of traffic image classification. Using the UCSD dataset, we extracted relevant frames and defined a region of interest (ROI) to focus on the traffic scenes. The model architecture comprised convolutional and pooling layers, with dropout regularization implemented to mitigate overfitting. Despite the limited availability of training data, we adopted data augmentation techniques and employed a powerful GPU for efficient model training. These efforts contribute to advancing image classification methods for traffic analysis, offering insights into congestion patterns, and facilitating effective traffic management strategies.

5.3 Classification Using Convolutional Neural Network (CNN)

5.3.1. Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is a powerful deep learning model designed explicitly for processing and analyzing visual data. CNNs are inspired by the biological optical system, mimicking the behavior of neurons in the human brain. It has revolutionized computer vision tasks such as image classification, object detection, and image segmentation.

The fundamental building block of a CNN is the convolutional layer. In this layer, a set of learnable filters, or kernels, are convolved with the input image to extract spatial hierarchies of features. The convolution operation is defined as follows:

$$H_{ij}=\sigma(\sum_{m,n} W_{mn} \cdot X_{i+m,j+n}+b)$$

Here, H_{ij} represents the output feature map at position (i,j), W_{mn} denotes the filter weights, $X_{i+m,j+n}$ is the input pixel value at position $(i+m,j+n)$, and b is a bias term. The activation function σ introduces non-linearity, allowing the network to model complex relationships within the data.

CNNs leverage the concept of parameter sharing, which significantly reduces the number of learnable parameters compared to fully connected networks. Each filter is applied across the entire input, generating a feature map that captures a particular visual pattern. Multiple filters can detect different features simultaneously, forming diverse feature maps.

Pooling layers are commonly employed to capture spatial relationships and reduce spatial dimensionality. A popular pooling operation is max pooling, which partitions the input into non-overlapping regions and selects the maximum value within each area. This down samples the feature maps while retaining the most salient features.

Finally, fully connected layers are typically added at the network's end to perform classification or regression tasks. These layers connect every neuron to the neurons in the previous layer, allowing the web to learn complex combinations of features. For example, a CNN can be trained on a large dataset of labeled images for image classification. During training, the network learns to recognize various visual patterns and classify pictures into different categories. Once trained, CNN can accurately organize unseen images by leveraging known representations and hierarchical features.

In summary, CNNs are a highly effective approach for visual data analysis. They can automatically learn and extract meaningful features from images using convolutional layers, parameter sharing, and pooling operations. These networks have propelled significant advancements in computer vision, enabling tasks such as image recognition, object detection, and semantic segmentation to achieve state-of-the-art performance.

5.3.2. Application methods

Due to their remarkable performance, deep convolutional neural networks (DCNNs) have established themselves as the leading approach for image classification tasks. In this study, we employed a traditional ConvNet architecture comprising convolutional and pooling layers. Our research utilized the UCSD dataset, which comprises 254 highway traffic videos. Each video contained 40–50 frames with a resolution of 320 x 240 pixels. We extracted 4-5 frames from each video to focus on the most informative frames and manually defined a region of interest (ROI) to exclude unrelated external objects from the scenes.

To address memory allocation challenges during model training, we resized the trimmed images to 180 x 180 pixels. These resized images served as the input for our model, structured with two successive convolutional layers of size 32x3x3, followed by a max pooling layer of size 2x2. The network was further expanded with two additional convolutional layers of size 64x3x3, followed by another max pooling layer of the same size. To mitigate the risk of overfitting, a dropout layer with a probability of 0.25 was applied after each max pooling layer. The rectified linear unit (ReLU) activation function was employed throughout the model to introduce non-linearity and enhance the model's ability to capture complex patterns and features. The architecture of our DCNN model, as depicted in Table 1, demonstrates the sequential arrangement of the layers, providing a clear overview of the model's structure and the parameters associated with each layer. This architecture highlights the successive nature of the convolutional layers, followed by the pooling layers, emphasizing the hierarchical feature extraction process inherent in ConvNet architectures.

By utilizing this carefully designed DCNN architecture, we aimed to leverage the power of deep learning to classify and analyze highway traffic images accurately. Including convolutional and pooling layers enables the model to effectively learn hierarchical representations of the traffic scenes, simultaneously capturing local details and global patterns. Incorporating dropout layers helps mitigate overfitting, enhancing the model's generalization capabilities.

In conclusion, our research contributes to image classification by utilizing a ConvNet architecture, specifically a DCNN, for highway traffic analysis. As outlined in Table 1, the architecture provides a comprehensive representation of the model's structure, highlighting the arrangement of convolutional and pooling layers. Using the UCSD dataset, we extracted informative frames and defined regions of interest to focus on the traffic scenes. The model's ability to learn hierarchical features and prevent overfitting, coupled with deep learning, allows for accurate image classification and paves the way for improved traffic analysis and management strategies.

TABLE 5.1: DCNN MODEL ARCHITECTURE USED

Layer	Kernel	Stride	Output shape
Input			[180, 180, 3]
Convolution	3x3	1	[180,180, 32]
Convolution	3x3	1	[178, 178, 32]
Max Pooling	2x2	2	[89, 89, 32]
Dropout			[89, 89, 64]
Convolution	3x3	1	[89, 89, 64]
Convolution	3x3	1	[87, 87, 64]
Max Pooling	2x2	2	[43, 43, 64]
Dropout			[43, 43, 64]
Dense			512
Dropout			512
Dense			2

CHAPTER-6

Results and Analysis

We achieved 98.2% classification accuracy through our proposed color-coded scheme. We have employed two deep learning models: one (YOLO) for vehicle detection and a DCNN for classification. The model's performance was evaluated using traditional performance metrics of precision, recall, and accuracy. To assess these metrics, we need to define the concepts of "true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

TP: When a congested image was correctly labeled
TN: If a non-congested image was correctly labeled
FP: If a congested image was predicted as non-congested
FN: if a non-congested image accuracy predicted as congested

Now, precision, recall, and accuracy can be found using the following equations:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP}{TP + FP + TN + FN}$$

98.2 Recall 95.6 Accuracy 98.2
The standard metrics of the proposed model are shown in Table II.

TABLE 6.1: STANDARD METRICS

Performance Metrics	**Result(%)**
Precision	98.2
Recall	95.6
Accuracy	98.2

Furthermore, we have validated our model with test images of eclectic sites without pre-training on those sites. Our model can perform congestion prediction for a different road segment or location. To validate this, we feed the ROI images before applying the color coding to the DCNN. Without color coding, the model failed to detect the validation set properly (10% accuracy only).

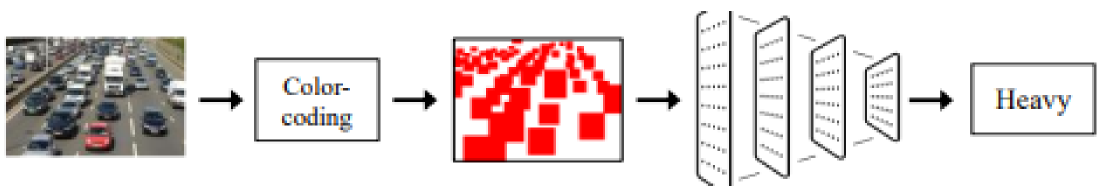

Fig. 6.1: Validation set using color-coded scheme (correct prediction)

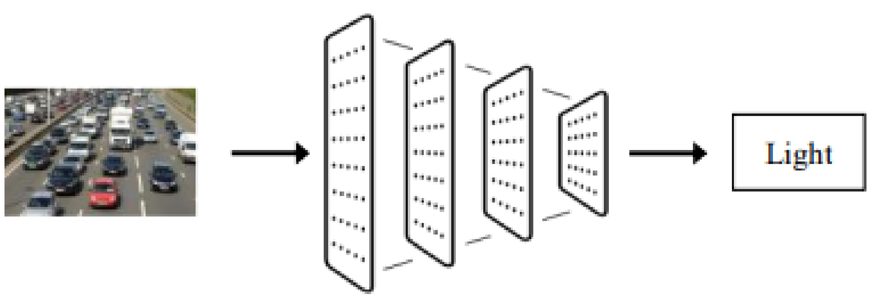

Fig. 6.2: Validation set without a color-coded scheme (false prediction)

From Fig. 6 and Fig. 7, it is evident that a color-coded scheme enhances the performance of DCNN regarding data dependency. The validation set is a collection of images of diverse traffic congestion. In Fig. 7, as the input image is from the different road segments by which the DCNN has been trained, it failed to provide the correct classification. On the contrary, in Fig. 6, due to the color-coded scheme, the prediction is accurate.

CHAPTER-7

Conclusion

This research paper introduces a novel system that utilizes a color-coding scheme to train a Deep Convolutional Neural Network (DCNN) for accurately predicting traffic congestion with an impressive accuracy rate of 98.2%. Previous studies in the field have often relied heavily on specific datasets, each containing different road segments and scenes, resulting in limited diversity. In contrast, our proposed color coding of vehicles offers a data-independent segmentation approach. By converting the images into color-coded binary representations, we establish a clear relationship between occupied space (red color) and vacant space (white color) within the road segment. This color-coding technique enhances the diversity and richness of our model, as it captures the occupancy patterns more comprehensively.

The work highlights the potential for using color-coded binary images to accumulate diverse traffic congestion data. This accumulation could serve as a benchmark for future traffic congestion classification challenges. It also opens up possibilities for exploring advanced models such as AlexNet, GoogLeNet, and VGGNet to enhance the detection and classification of traffic congestion scenarios. By leveraging these state-of-the-art models, we can leverage their advanced architectures and capabilities to achieve even higher accuracy and performance in traffic congestion prediction tasks.

The introduction of the color-coding scheme to enhance the training of DCNN models marks a significant advancement in traffic analysis and congestion prediction. By focusing on the occupancy and vacancy patterns within the road segment, our system offers a more nuanced understanding of traffic congestion. This research provides a strong foundation for further investigation into color-based segmentation techniques and the utilization of diverse datasets for training robust traffic congestion prediction models. The proposed system has the potential to revolutionize traffic management strategies by enabling more accurate and timely predictions of congestion, leading to improved traffic flow and enhanced transportation efficiency.

REFERENCES

[1] X. Ke, L. Shi, W. Guo, and D. Chen, "Multi-Dimensional Traffic Congestion Detection Based on Fusion of Visual Features and Convolutional Neural Network," IEEE Trans. Intell. Transp. Syst., vol. 20, no. 6, pp. 2157–2170, 2019, doi: 10.1109/TITS.2018.2864612

[2] M. Pi, H. Yeon, H. Son, and Y. Jang, "Visual Cause Analytics for Traffic Congestion," IEEE Trans. Vis. Comput. Graph., vol. 27, no. 3, pp. 2186–2201, 2021, doi: 10.1109/TVCG.2019.2940580

[3] M. Akhtar and S. Moridpour, "A Review of Traffic Congestion Prediction Using Artificial Intelligence," J. Adv. Transp., vol. 2021, 2021, doi: 10.1155/2021/8878011
T. Bogaerts, A. D. Masegosa, J. S. Angarita-Zapata, E. Onieva, and P.

[4]Hellinckx, "A graph CNN-LSTM neural network for short and longterm traffic forecasting based on trajectory data," Transp. Res. Part C
Emerg. Technol., vol. 112, no. January, pp. 62–77, 2020, doi: 10.1016/j.trc.2020.01.010

[5] P. Chakraborty, Y. O. Adu-Gyamfi, S. Poddar, V. Ahsani, A. Sharma, and S. Sarkar, "Traffic Congestion Detection from Camera Images using Deep Convolution Neural Networks," Transp. Res. Rec., vol. 2672, no. 45, pp. 222–231, 2018, doi: 10.1177/0361198118777631

[6] R. Cucchiara, M. Piccardi, and P. Mello, "Image analysis and rulebased reasoning for a traffic monitoring system," IEEE Conf. Intell.
Transp. Syst. Proceedings, ITSC, vol. 1, no. 2, pp. 758–763, 1999, doi: 10.1109/itsc.1999.821156

[7] J. Guo, Y. Liu, Q. Yang, Y. Wang, and S. Fang, "GPS-based citywide traffic congestion forecasting using CNN-RNN and C3D hybrid

model," Transp. A Transp. Sci., vol. 17, no. 2, pp. 190–211, 2021, doi: 10.1080/23249935.2020.1745927
[8] M. A. A. Al-qaness, A. A. Abbasi, H. Fan, R. A. Ibrahim, S. H. Alsamhi, and A. Hawbani, "An improved YOLO-based road traffic monitoring system," Computing, vol. 103, no. 2, pp. 211–230, 2021, doi: 10.1007/s00607-020-00869-8
[9] H. Cui, G. Yuan, N. Liu, M. Xu, and H. Song, "Convolutional neural network for recognizing highway traffic congestion," J. Intell. Transp. Syst. Technol. Planning, Oper., vol. 24, no. 3, pp. 279–289, 2020, doi: 10.1080/15472450.2020.1742121
[10] H. Nguyen, "Improving Faster R-CNN Framework for Fast Vehicle Detection," Math. Probl. Eng., vol. 2019, 2019, doi: 10.1155/2019/3808064
[11] K. Bayoudh, F. Hamdaoui, and A. Mtibaa, "Transfer learning based hybrid 2D-3D CNN for traffic sign recognition and semantic road detection applied in advanced driver assistance systems," Appl. Intell., vol. 51, no. 1, pp. 124–142, 2021, doi: 10.1007/s10489-020-01801-5
[12] D. Impedovo, F. Balducci, V. Dentamaro, and G. Pirlo, "Vehicular traffic congestion classification by visual features and deep learning approaches: A comparison," Sensors (Switzerland), vol. 19, no. 23, 2019, doi: 10.3390/s19235213
[13] R. Zhang, A. Ishikawa, W. Wang, B. Striner, and O. K. Tonguz, "Detection for Intelligent Traffic Signal Control," vol. 0, pp. 1–12, 2020.
[14] A. Sobral, L. Oliveira, L. Schnitman, and F. De Souza, "Highway traffic congestion classification using holistic properties," Proc. IASTED Int. Conf. Signal Process. Pattern Recognit. Appl. SPPRA 2013, no. February, pp. 458–465, 2013, doi: 10.2316/P.2013.798-105
[15] H. M. Soleh M, Jati G, "Multi object detection and tracking using optical flow density–Hungarian Kalman filter (Ofd-Hkf) algorithm for vehicle counting," J Ilmu Komput. dan Informasi. 2018;11(1)17-26,

vol. 1, no. 506, pp. 80–94, 2018.
[16] V. Murugan, V. R. Vijaykumar, and A. Nidhila, “A deep learning RcNn approach for vehicle recognition in traffic surveillance system,” Proc. 2019 IEEE Int. Conf. Commun. Signal Process. ICCSP 2019, pp. 157–160, 2019, doi: 10.1109/ICCSP.2019.8698018
[17] B. Xu, B. Wang, and Y. Gu, “Vehicle Detection in Aerial Images Using Modified YOLO,” Int. Conf. Commun. Technol. Proceedings, ICCT, pp. 1669–1672, 2019, doi: 10.1109/ICCT46805.2019.8947049
[18] “SVCL - Analysis of Traffic Video.” http://www.svcl.ucsd.edu/projects/traffic/ (accessed Aug. 20, 2022).
[19] E. Ahmed, M. Jones, and T. K. Marks, “An improved deep learning architecture for person re-identification,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 07-12-June, pp. 3908–3916, 2015, doi: 10.1109/CVPR.2015.7299016