DOI: 10.24996/ijs.2023.64.10.40

ISSN: 0067-2904

Transfer Learning Based Traffic Light Detection and Recognition Using CNN Inception-V3 Model

Mohammed Qader Kheder*, Aree Ali Mohammed

Department of Computer Science, College of Science, University of Sulaimani, Sulaimaniyah, Iraq

Received: 2/8/2022 Accepted: 18/12/2022 Published: 30/10/2023

Abstract

Due to the lack of vehicle-to-infrastructure (V2I) communication in the existing transportation systems, traffic light detection and recognition is essential for advanced driver assistant systems (ADAS) and road infrastructure surveys. Additionally, autonomous vehicles have the potential to change urban transportation by making it safe, economical, sustainable, congestion-free, and transportable in other ways. Because of their limitations, traditional traffic light detection and recognition algorithms are not able to recognize traffic lights as effectively as deep learning-based techniques, which take a lot of time and effort to develop. The main aim of this research is to propose a traffic light detection and recognition model based on a transfer learning-based model that uses the Inception-V3 network model to significantly reduce the amount of training data and computing costs. The proposed model was trained and tested in the laboratory for the intelligent and safe automobiles (LISA) traffic light dataset, which has been augmented by several pre-processing methods. Then, using different convolution and pooling techniques, the retrieved layer-wise features were compared and analyzed. Lastly, great reliability and repeatability are seen based on statistical analysis when the transfer learning-based model is frequently retrained utilizing precise tuning parameters. The results demonstrate that a transfer learning-based model is capable of high-level recognition performance in the proposed model, with an accuracy rate of 98.6%.

Keywords: Transfer Learning; Convolution Neural Network; CNN; Inception-V3; Traffic Light Detection and Recognition; LISA Traffic Light Dataset.

الكشف والتعرف على إشارات المرور القائمة على تعلم النقل بإستخدام نموذج-CNN Inception V3

محمد قادر خدر*, ئارى على محمد

قسم علم الحاسبات، كلية العلوم، جامعة السليمانية, السليمانية, العراق.

الخلاصة

بسبب عدم توفر الإتصال من مركبة إلى البنية التحتية (V2I) في أنظمة الإتصالات الحالية، يعد الكشف والتعرف على إشارات المرور شيء ضروري لأنظمة مساعدة السائق المتقدمة (ADAS) وإستطلاعات البنية التحتية للطرق. علاوة على ذلك، لدى المركبات المستقلة القدرة على تغيير النقل الحضري بجعله آمنا واقتصاديا ومستداما وخاليا من الازدحام وبالمقدور نقلها بطرق أخرى. بسبب القيود عليها، فإن خوارزميات الكشف والتعرف

*Email: mohammed.kheder@univsul.edu.iq

على إشارات المرور التقليدية غير قادرة على التعرف على إشارات المرور بشكل فعال مثل التقنيات القائمة على التعلم العميق، والتي تتطلب الكثير من الوقت والجهد بغية تطويرها. إن الهدف الرئيسي من هذا البحث هو إقتراح نموذج قائم على التعلم للإتصال تستند على إستخدام نموذج شبكة Inception−V3 من أجل تقليل كمية بيانات التدريب بشكل ملحوظ وتقليل تكاليف الحوسبة. يتم تعليم وإختبار النموذج المقترح في المختبر لمجموعة بيانات إشارات مرور السيارات الذكية والآمنة (LISA)، والتي تم تعزيزها بعدة طرق معالجة مسبقة. بعد ذلك، تتم مقارنة وتحليل ميزات الطبقة المسترجعة عن طريق إستخدام تقنيات الإلتفاف والتجميع المختلفة. أخيرا، يتم رؤية موثوقية كبيرة وإمكانية التكرار بالإستناد إلى التحليل الإحصائي عندما يتم إعادة تدريب نموذج النقل القائم على التعلم بشكل متكرر بإستخدام معلمات ضبط دقيقة. تظهر النتائج بأنه بمقدور النموذج القائم على التعلم والنقل أداء تعرف عالي المستوى في النموذج المقترح، وذلك بمعدل دقة٩٨.٦٪.

1. Introduction

Road accidents have increased recently over the past several years due to the increased use of vehicles for transportation. According to Symeonidis et al. [1], one of the major factors contributing to traffic accidents is drivers who ignore traffic lights, including red lights. Many governments have installed video cameras to monitor the road and record vehicle accidents to decrease the number of accidents caused by disobeying traffic lights. However, implementing this approach would be costly and would not provide a true solution to avoiding accidents. As a result, many vehicle manufacturers attempted to integrate advanced driver assistant systems (ADAS), including the traffic light detection and recognition system, into their vehicles to assist drivers while driving [2, 3]. These systems are made to be integrated into vehicles so that they may provide drivers with advice while being ready to correct several driving mistakes to prevent accidents without bothering the driver.

There are two phases in a traffic light detection and recognition systems: detection and recognition. Localizing traffic lights in an image is called traffic light detection, while correctly labelling them is called traffic light recognition [4]. Moreover, traffic lights, often referred to as traffic control signals can be found at road intersections, pedestrian crossings, and several other places. Red, yellow, and green traffic lights are the three main types.

The new version of the Microsoft Common Object in Context (MS COCO) dataset, which is divided into 80 object classes and different super categories like a person, vehicle, traffic light, bird, laptop, and other objects [5], was used in this study along with the Object Detection API of TensorFlow to detect traffic lights in road images. According to Gautam and Kumar [3] and Mahesh and Satich [6], traffic light detection and recognition is a challenge due to several factors, such as:

1. the size of very small traffic lights,
2. the color of traffic lights, particularly red ones, will coincide with the potential vehicles' backlight, streetlights, or decorative lights in night vision.
3. orientation of the traffic lights, and others.

Prior studies on traffic light detection and recognition have used map information [7], template matching [8], spotlight detection and color thresholding [9], among other techniques. All these systems depend on assumptions. They usually require a recognizable background and traffic light images that are at least a certain size for the algorithm to operate or assume the existence of maps that already know where every traffic light in the area is placed. Machine learning has reached a golden time of development because of the continuous development of computer technology, which has significantly reduced the limitations of artificial neural networks (ANN) [10]. In addition, significant advancements were achieved in several applications, especially computer vision. Considering the current improvement of deep neural networks, which are types of machine learning. As mentioned in [2, 6, 11], deep neural network techniques have been used for image classification, end-to-end object detection, pixel-precise object segmentation, and other vision applications. A deep neural network model mimics the way the human brain processes information [6]. Therefore, the robustness and generalization of the algorithms might be improved by using deep neural network models rather than the traditional traffic light detection and recognition techniques to extract useful features from the road images.

In this study, a transfer learning-based model for the precise detection and recognition of traffic lights is proposed.

The proposed model uses TensorFlow’s Inception-V3 Convolutional Neural Network (CNN) model for transfer learning. Moreover, it was trained using the laboratory for intelligent and safe automobiles (LISA) dataset, which contains a variety of images (both day and night vision). The proposed model achieves its goal by classifying the traffic light colors when they are detected.

Even though the results from deep convolutional neural networks outperform feature-based techniques on traffic light detection and recognition, they still have two weaknesses. The first is that deep learning models are typically created through an iterative, convoluted process, in which the training phase necessitates a large amount of labelled data [12–14]. Second, there would be a substantial cost associated with computing due to the vast number of neuron connections [3, 12]. The transfer learning approach for traffic light detection and recognition is introduced in this research to address these two weaknesses.

Hussain et al. [12] stated that the main goal of transfer learning is to study how individuals might transfer knowledge from one situation to another with similar situations. Transfer of learning is now regarded to be the process by which earlier experiences enhance learning results in a new context [3, 12–14]. This indicates that a pre-trained model may be used to carry out the same task by learning new data distribution and adjusting model parameters at each layer. The proposed model is trained and evaluated using Google Colab Notebook and GPU (Graphical Processing Unit) hardware. As a result, this research paper’s reported results are GPU-based.

The remainder of this paper is organized as follows: In Section 2, a number of related works in the field of traffic light detection and recognition are reviewed. Several theoretical concepts related to the proposed model are provided in Section 3. Section 4 provides a detailed discussion of the proposed model’s architecture while the results are assessed in Section 5. Finally, the study’s conclusion is presented in Section 6.

2. Related Works

Detection and recognition of objects in an image have been the subject of several ongoing studies. Thus, a literature review was carried out to examine the ongoing research and review current developments.

Traffic light detection and recognition techniques may be divided into three categories: image processing-based, machine learning-based, and map-based techniques. As stated by [1, 7, 15], each of these techniques makes reliable predictions. A single or several processes will be applied to the image in the image processing-based technique to achieve the desired result.

Widyantoro and Saputra [4] accomplished traffic light detection and recognition using color segmentation and Circle Hough Transform (CHT), while Mu et al. [15] were able to do the same thing by converting RGB (Red, Green, and Blue) to HSV (Hue, Saturation, and Value), filtering, employing histogram of oriented gradient (HOG) features, and using the support vector machine (SVM) deep learning technique.

To analyze traffic lights at night vision, Pillai et al [9] developed a system that detects taillights using different morphological operations, including filtering extraction, thresholding, and other operations. On the other hand, adaptive background suppression filters were proposed by Shi et al. [16] as a fast and reliable solution for traffic light detection under various lighting situations.

Furthermore, De Charette and Nashashibi [17] developed a real-time traffic light detection and recognition system for intelligent vehicles. The proposed system is fully based on image processing. Spotlight detection is used to accomplish the detection phase in grayscale images, and the regions that show traffic light detection are then chosen. Subsequently, using general adaptive templates, the recognition step is achieved.

Touma and Abbas [18] suggested a traffic light detection and recognition system based on digital image processing by using three steps: image subtraction, traffic light segmentation, and traffic light recognition (using two methods CHT and morphological operations). Image cropping, grayscale conversion, image subtraction, and other image processing techniques have all been used. the system was evaluated based on day and night vision. Even though the steps to image processing are quite straightforward and uncomplicated, it passes through crucial stages including thresholding and filtering. In these steps, small errors in computation or deviations from standards may result in unclear results that are highly undesired in sensitive traffic light detection and recognition situations. To prevent these challenges, machine learning-based solutions are employed separately or in combination with several processing techniques to eliminate confusing instructions, such as [1, 3, 13, 19, 20] proposed traffic light detection and recognition systems based on deep learning methods.

Current traffic light recognition methods consist mostly of two steps: extracting image features and using a high-quality classifier or matching a template for classification. The primary distinctive features of a traffic light are its color and intensity, which may be utilized to identify its location in an image. A system that can locate traffic lights and recognize their status in a video sequence was proposed by Symeonidis et al. [1]. The detection phase was accomplished by applying several image processing techniques, such as image cropping, the Gaussian low-pass filter, color transformation, segmentation, morphological dilation, Canny edge detection, and CHT. Additionally, a CNN-based technique was used for the recognition phase. Since the LISA traffic light dataset includes both daytime and night-time video sequences, it was used to evaluate the proposed system.

Wang et al. [19] developed a deep learning-based traffic light detection and recognition system that can automatically extract representation and robust features from the input image without the need for artificial features. The system is divided into two phases: regional proposal, which is achieved using several image processing techniques, including the Gaussian filter, color conversion (RGB to HSI), Top-hat transform, and traffic light classification using CNN-based techniques.

Moreover, a deep learning-based system for recognizing traffic lights was proposed by Madhu and Nair [20], and trained and evaluated using data from the LISA traffic light dataset. The system comprises of Single Shot MultiBox Detection (SSD) for detection and Faster R-CNN inception-V2 for recognition processes.

As has already been stated, adopting deep learning-based systems will present several challenges, including the need for a huge amount of data during the training phase, as well as a large number of connections between neurons, both of which might result in high computing costs. As a result, some approaches, like this study, used transfer learning as the basis for their solutions. Pathak and Elster [13] created a traffic monitoring model to prevent accidents. On their own-created VOC dataset, they used YOLO-V2, a homogeneous convolutional architecture that speeds up bounding box prediction. The proposed model is based on the transfer learning approach.

In addition, Gautam and Kumar [3] proposed a traffic light detection and recognition system using a variety of CNN-based transfer learning models that have been pre-trained, including VGG16, Inception-V3, AlexNet, ResNet50, DensNet121, and Xception, using the fully accessible LISA traffic light dataset. To determine the color of the observed traffic light image, they also applied a random forest classifier. Finally, they claimed that transfer learning speeds up the training process and that using a transfer learning-based model was superior to building the model from scratch in terms of results.

Based on a detailed analysis of transfer learning and TensorFlow’s object detection APIs, which are helpful to detect and classify objects in images and videos, there are several gaps that other relevant studies have not considered. Finding the best solution for these gaps is the contribution of this study, and they are as follows: (1) Using a transfer learning-based model instead of traditional learning. Machine learning operates in isolation from traditional learning. It learns how to perform a certain task when given a large enough dataset. When assigned to solving a new issue, it cannot return to any previously acquired information. On the other hand, transfer learning depends on previously acquired tasks to learn new ones. The algorithm can store and retrieve information. (2) Using a small amount of training data and reducing computational costs due to the transfer learning-based model in proposing the traffic light detection and recognition system. As mentioned in [3, 12, 14], transfer learning-based models speed up learning, produce high-quality results, and reduce computational costs. (3) Considering the proposed system may be implemented in other real-time applications, it must utilize a quick and reliable deep learning algorithm. In comparison with previous traffic light detection and recognition systems, the proposed system, which is based on the Inception-V3 CNN architecture, delivers quick and accurate results.

3. Theoretical Background

The proposed system uses a CNN-based model and the pretrained Inception-V3 architecture. Furthermore, the system is learned and evaluated using the transfer learning-based technique, these popular deep learning topics are explained more below.

3.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks, often known as CNNs or ConvNets, are multi-stage deep architectures that combine convolutional layers with pooling or subsampling layers, followed by one or more fully connected layers. As stated by [10, 21, 22], its hierarchical network architecture makes it simpler to acquire invariant features and gather layer-by-layer representations from lower to higher layers. A classic CNN architecture that was used to recognize digits is shown in Figure 1 [24].

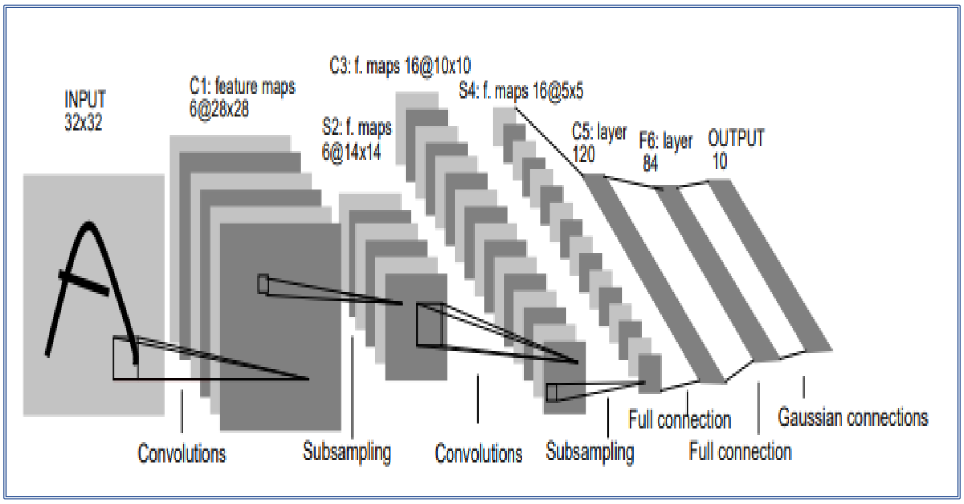

Figure 1: Standard architecture of the CNN [24]

In the illustration above, inputs are fed to develop a representation of the features using two-phase convolutional and subsampling processes, and a Gaussian classifier is then used to produce a probabilistic distribution. The CNN is made up of three main components [14, 23]:

1. *Convolutional operation*: it slides a weighted window over the whole image to calculate the weighted total of the input pixel values. To prevent the input from being used to learn useless linear representations, a non-linear activation operation known as the activation function is then implemented [11]. There are several activation functions used for this, such as the Rectified Linear Unit (ReLU), Softmax, tanH, and Sigmoid. Each of these functions has a unique purpose. [11, 10] stated that for a CNN model with binary classification, sigmoid and softmax functions are preferred, whereas softmax is typically applied for multiclass classification. According to [11, 14, 23], ReLU is one of the most efficient activation functions, where a non-negative piecewise function always returns the largest value between 0 and the input, to compute, it uses the formula below.

$$f(x) = \max(0, x) \begin{cases} 0, \; for\; x < 0. \\ \quad .. \\ x, \; for\; x \geq 0. \end{cases} \tag{1}$$

Convolutional operation is defined as the input *X* convolved with a filter *W* of dimension (*Nx*, *Ny*). The outcome of *Y* is mathematically described in the following formula:

$$Y \; = f(\textstyle\sum_{i=1}^{n} X_i * W + b_i) \tag{2}$$

Where *f* represents the activation function, *n* stands for the number of elements and b_i is the bias of output. In conclusion, the element-wise multiplication of the input and weight matrix yields the weighted sum of the input pixel values in a 2D convolutional process.

2. *Pooling operation*: this operation often comes after the convolutional operation. By moving a weighted window over the pixel matrix, it accumulates small pixel pitches and subsamples the features of the preceding layer. Therefore, the main objective of this operation is to reduce the size of the convolved feature map to reduce computational costs [23]. There are three common types of pooling operations:

- *Max pooling*: the largest feature is extracted from the feature map.

- *Average pooling*: calculates the average of the features in a region of a predefined size inside an image.
- *Sum pooling*: calculates the sum of the features inside an area of a predefined size within an image.

This layer often acts as a connection between the convolutional layer and the fully-connected layer.

3. *Fully-connected layer*: it generally uses a sequence of affine transformations to convert feature mappings into 1D feature vectors before adding a classifier to generate class-specific probability distributions. The Softmax classifier is widely used to normalize the label probability in object recognition, as shown mathematically below.

$$softmax(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{n} e^{y_i}} \tag{3}$$

Where the elements of the input vector are all size y_i values. The normalization term appears at the bottom of the formula, which ensures that the function’s output values all sum to 1 and are inside the range (0, 1), producing a valid probability distribution. Additionally, *n* indicates the number of classes in the multi-class classifier, in this study *n* is equal to 4.

The local connection, pooling operation, shared weight, and hierarchical design of the CNN are its four main features [22, 24]. According to previous research on the function of the visual cortex, human cognition of the actual world extends from local to global. Since each neuron records local features and combines them with local information to represent the entire image in a higher layer, CNNs are created to mimic human visual mechanisms. Through a local connection, convolutional and pooling processes may obtain significant and unique feature representations of a given set of data. Furthermore, sharing weights is used in convolution and pooling operations using sliding weighted windows. The concept of sharing weight illustrates how spatial identity, which is a statistical feature for the entire image [10], can be shared. As a result, features may be extracted from the image at each pixel point using the same weights. In addition, hierarchical architecture is used to investigate correlations between neurons in adjacent layers to extract layer-wise features. In contrast to a fully connected network, CNN's distinctive architecture makes the extraction of notable features simpler while reducing parameter counts and neuron connections. In summary, CNNs, especially for the Inception-V3 model, are the most advanced technologies in the computer vision field and are often used for image recognition and classification tasks [12, 14, 25].

3.2 Inception-V3 Model

CNN classifiers achieved a significant milestone with the invention of the Inception network. Before its establishment, the majority of popular CNNs stacked convolutional layers deeper and deeper to improve performance [25]. Szegedy et al. [25] and Xia et al. [26] stated that despite the complexity of the Inception neural networks, they used a variety of techniques to enhance performance, both in terms of speed and accuracy. Multiple versions of the Inception: Inception-V1 or GoogleNet (2014), Inception-V2 (2015), Inception-V3 (2015), Inception-V4 and Inception-ResNet (2016) were developed due to its continuous development [25]. Each new version is an improvement over the previous one. Inception-V3 was trained on a dataset consisting of 1,000 classes from the original ImageNet dataset, which was trained with over one million training images. The traffic light is one of these classes on which the model has been trained to classify the color states. Annually, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is a significant image recognition and classification competition [27]. Eventually, several convolutional models were developed to decrease the Top-5 error rate, which is the percentage of times the classifier failed to include the correct class among its five predictions while classifying objects. Figure 2 depicts CNN object recognition Top-5 error rates using ImageNet [14], and GoogleNet (Inception-V1) has achieved remarkable recognition achievements. According to the results, the recognition performance can be improved with a deeper model layer.

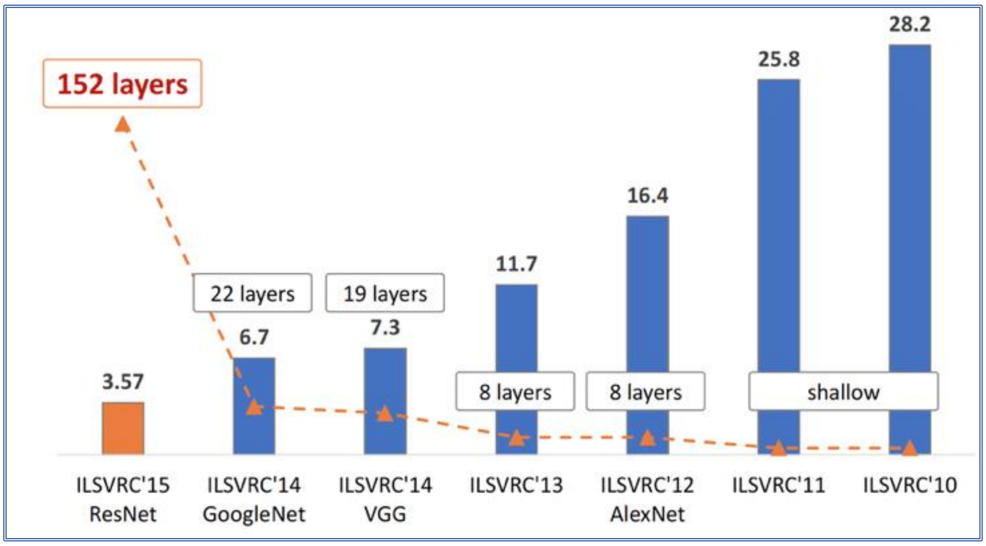

Figure 2: Top-5 error of representative CNN algorithms [14] (*Abbreviations*: ResNet (Residual Network), GoogleNet (Google Inception Net), and VGG (Visual Geometry Group))

As discussed in [12, 14], the Inception-V3 model performs better in terms of object recognition than GoogleNet (Inception-V1). Inception-V3 consists of a basic convolutional block, an enhanced Inception module, and a classifier [14]. For feature extraction, the basic convolutional block, which alternates between convolutional and pooling layers, is applied. Besides, Network In Network (NIN) [28], where multi-scale convolutions are performed concurrently, is used to construct the enhanced Inception module and the results of each branch's convolution are further combined. In addition to improving gradient convergence and stabilizing training outcomes, the inclusion of auxiliary classifiers also reduces the risk of overfitting and vanishing gradients.

The 1x1 convolutional kernel is widely used in Inception-V3 to decrease the number of feature channels and speed up the training process. More to say, the large convolution is decomposed into smaller convolutions, which reduces the calculation cost and the number of parameters. In conclusion, Inception-V3 provides the most advanced object recognition performance because of its distinctive Inception architecture. Consequently, this type of CNN model is often utilized for transfer learning-based models.

3.3 Transfer Learning Model

Transfer learning is the reuse of a previously learnt model on a new task [3, 12-14, 30]. It is popular in deep learning since it can build deep neural networks with little training data. In most cases, training a fully convolutional network from start is time-consuming and requires a large training data [12, 29, 30]. Consequently, this issue may be resolved using the benefits of transfer learning with a pre-trained model. As demonstrated in [26], the Inception-V3 model may be used to recognize and categorize a new image by changing the design of fully-connected layers and changing the parameters for all convolution layers. The architecture of the transfer learning-based model (proposed model) is shown in Figure 3.

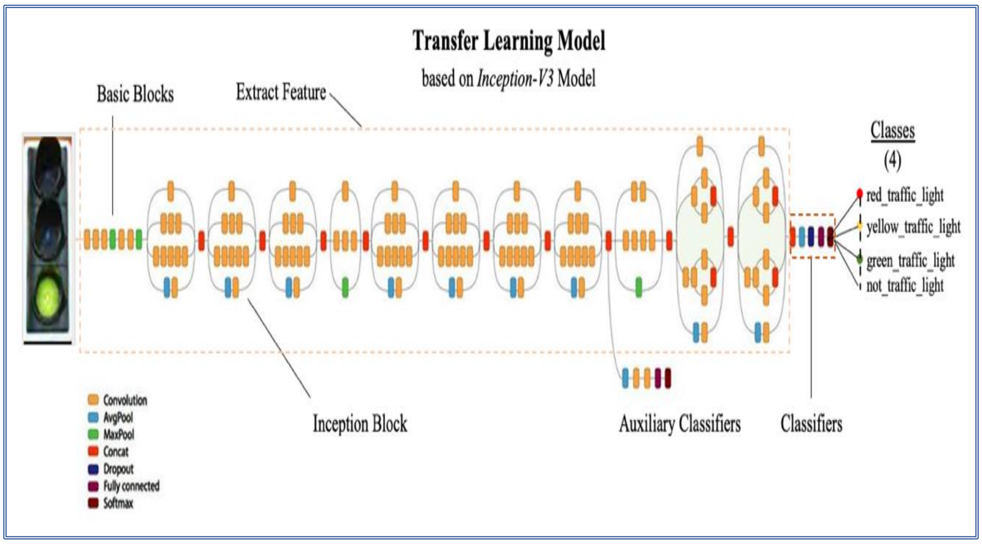

Figure 3: Inception-V3 model for the proposed system based on transfer learning

The Inception-V3 model-based sequential concatenation of the basic convolution block, enhanced Inception module, and task-specific classifiers. Especially, low-level feature mappings are trained using 1x1 and 3x3 kernel basic convolutional operations. To enhance convergence performance, multi-scale feature representations are combined in the Inception module and fed into an auxiliary classifier using a variety of convolution kernels, namely 1x1, 1x3, 3x1, 3x3, 1x5, 5x1, 5x5, 1x7, 7x1, and 7x7 filters. After implementing the 1x1 Inception module, multi-scale feature vectors are converted to 1D using a fully-connected layer. Lastly, a one-hot vector that is compatible with 4-classes (red_traffic_light, yellow_traffic_light, green_traffic_light and not_traffic_light) probability is produced using the Softmax classifier. Depending on the maximum 4-class probability value, the final classification outcome may be calculated. Subsequently, the performance of a transfer learning-based model for the proposed model is assessed using a case study on the LISA traffic light dataset.

4. Proposed Traffic Light Detection and Recognition Model

With the availability of huge volumes of data, faster GPUs, and improved algorithms, it is easier to teach machines to detect and classify many objects inside an image with high accuracy. The development of autonomous vehicles has increased rapidly due to technological developments. The accurate detection and recognition of traffic lights are essential to the development of such vehicles. The concept includes allowing autonomous vehicles to recognize traffic lights automatically without human intervention, which will help to reduce the number of road accidents.

In this study, the traffic light detection and recognition model for improving traffic light recognition is presented, which can be easily applied to autonomous vehicles. The proposed model was built using a transfer learning-based model of the pretrained MS COCO model and the Object Detection API of TensorFlow. Initially, several data pre-processing algorithms were performed on the LISA traffic light dataset, including image cropping, color conversion, and image resizing. And then, data augmentation was used to increase the amount of training data to avoid the overfitting problem. In addition, the pre-trained Inception-V3 model was implemented and trained using the LISA traffic light dataset. Figure 4 illustrates the block diagram of the proposed model, and how it works.

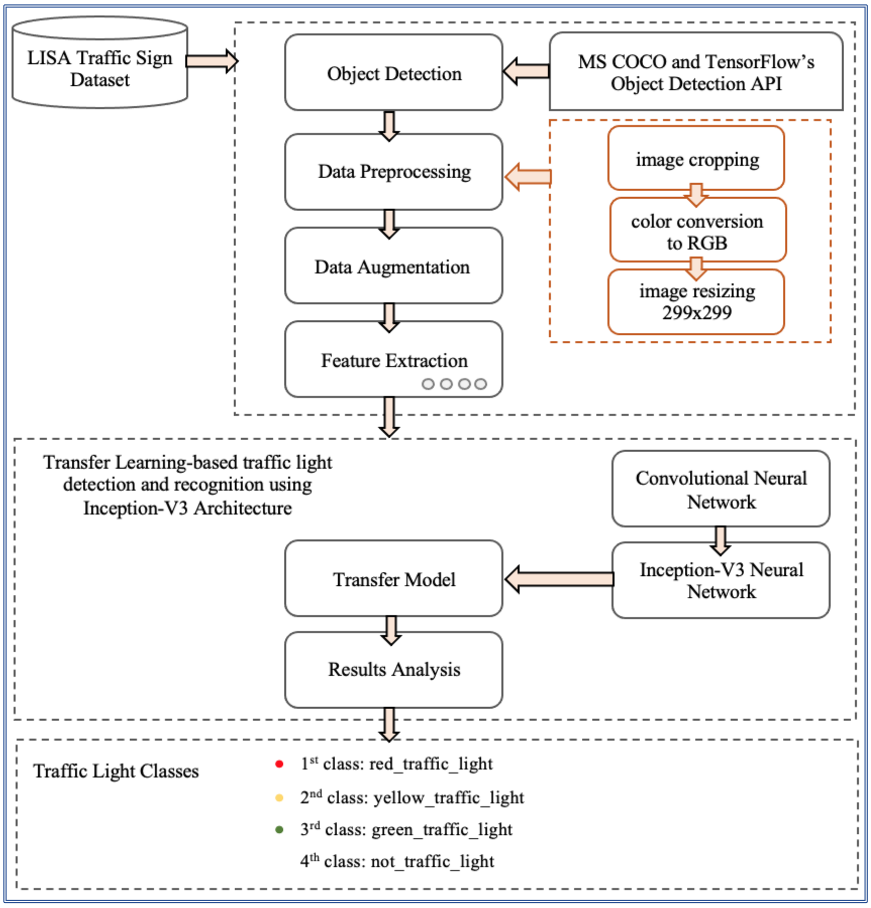
LISA Traffic Sign Dataset
Object Detection
MS COCO and TensorFlow's Object Detection API
Data Preprocessing
image cropping
color conversion to RGB
image resizing 299x299
Data Augmentation
Feature Extraction
Transfer Learning-based traffic light detection and recognition using Inception-V3 Architecture
Convolutional Neural Network
Inception-V3 Neural Network
Transfer Model
Results Analysis
Traffic Light Classes
1st class: red_traffic_light
2nd class: yellow_traffic_light
3rd class: green_traffic_light
4th class: not_traffic_light

Figure 4: Block diagram of the proposed model

4.1 Dataset Preparation

For object detection and evaluating the proposed model, both datasets: MS COCO and LISA traffic light were used, respectively.

4.1.1. The Microsoft Common Objects in Context (MS COCO)

The MS COCO dataset is large-scale object detection, segmentation, key-point detection, and captioning dataset published by Microsoft [27, 31]. In 2014, the initial version of the MS COCO dataset was released. It includes 164,062 images grouped into training, validation, and test sets of 82,783, 40,504, and 40,775 images, respectively. In 2015, an additional test set of 81, 434 images, containing all the prior test images and a further 40,659 images, was provided. In 2017, the training and validation split was changed from 82, 783 and 40, 504 to 118, 287 and 5,000, respectively, based on feedback from the community. In addition, the 2017 test set is a subset of the 2015 test set consisting of 40,670 images [31]. The MS COCO dataset contains images of 80 object classes divided into 11 super-categories, as seen in Figure 5. The first 14 classes are transportation-related, including bicycle, vehicle, bus, traffic lights, etc. [32]. The ID of the traffic light is 10.

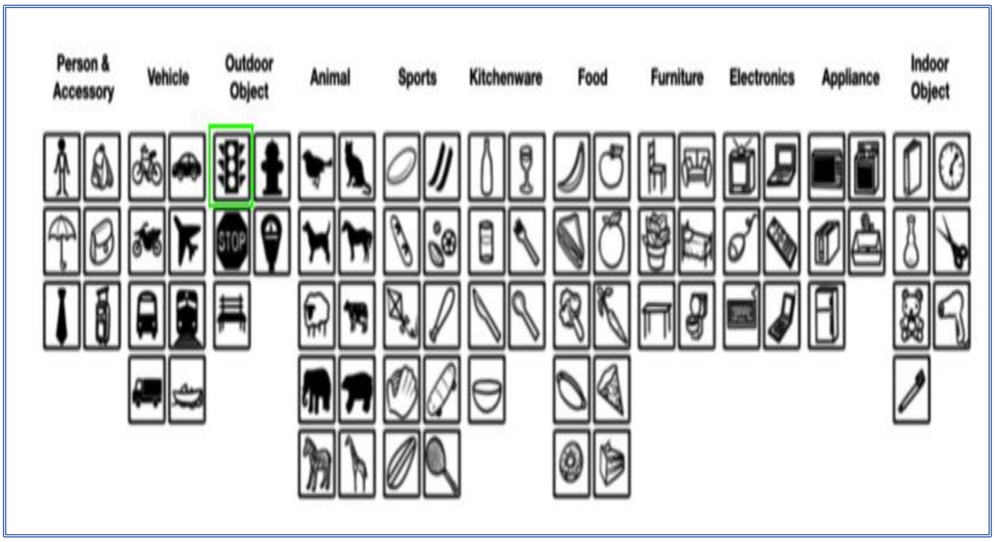

Figure 5: Samples of images with their MS COCO categories

The MS COCO dataset is often used by machine learning and computer vision applications [27, 32]. Thus, this dataset can be used to train algorithms for object detection and object classification, as this study is done for the traffic light detection and recognition model. This dataset was used for several reasons, including: (1) this is one of the most prominent benchmark datasets for object detection, scene understanding, and visual reasoning. (2) As reported by Lin et al. [31], the 80 object classes are chosen by subject matter specialists after deep consideration. (3) Each object category has a large amount of data, thereby the average number of objects per category is 27,473. In addition, according to [32] this is the most comprehensive dataset for objects in this context at the time this work was conducted.

4.1.2 LISA Traffic Light Dataset

All training, validation, and test images for training and evaluating in the proposed model were extracted from the LISA traffic light dataset. This is one of the most used datasets concerning traffic light systems. The LISA dataset contains traffic lights found in San Diego, California, United States [33]. The dataset contains two daytime sequences and two nighttime sequences. It is comprised of test and training video sequences with a total of 43,007 frames and 113,888 traffic light annotations. The sequences are captured at a resolution of 1280 x 960 by a stereo camera installed on the roof of a moving car at night and during the day under varying lighting and weather conditions. [33]. Details regarding this dataset are given in Table 1.

Table 1: The LISA traffic light dataset in detail

Camera	Stereo Camera (two lenses)
Video Length	23 minutes and 25 seconds
No. of Frames	43, 007
Resolution [WxH]	1280x960
Depth [bit]	8
Frame Rate [Hz]	16
Annotations	113,888
No. of Cities (US Cities)	2

4.2 Data Pre-processing

In this work, data pre-processing is performed before training and testing the proposed model to obtain higher accuracy in the proposed model. Accordingly, several image processing techniques were applied as follows:

- *Image cropping*: once the object (traffic light) has been detected inside the image frame. The traffic light object was then extracted from the other objects in the frame using a cropping image.
- *Color conversion*: the color conversion's goal is color-coded information that is sensitive to lighting conditions, noise, and captured equipment quality [4]. In this study, this conversion is used to convert the color to the RGB color system, because the proposed model can classify traffic lights based on traffic colors.
- *Image resizing*: the size of the cropped image in the LISA traffic light dataset varies after image cropping. Zhu et al. [32] have shown that many of the architectures of deep learning models require input images of the same size. As a result, all cropped images are resized to 299x299 pixels following the shape that was inputted into the Inception-V3 model, which should be (299, 299, 3). The preprocessing steps are presented in Figure 6.

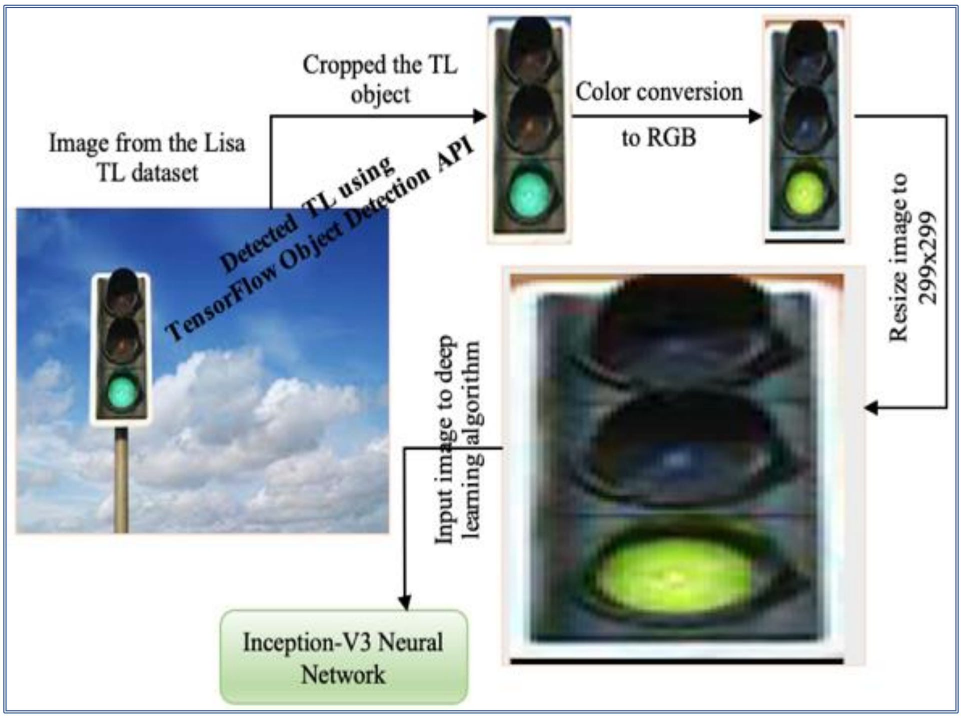
Image from the Lisa TL dataset
Detected TL using TensorFlow Object Detection API
Cropped the TL object
Color conversion to RGB
Resize image to 299x299
Input image to deep learning algorithm
Inception-V3 Neural Network

Figure 6: Pre-processing step's results

4.3 Data Augmentation

Data augmentation is the use of a variety of techniques to generate new data from current data to increase the amount of data [7, 12, 14]. This process is performed on the training data to avoid an overfitting issue, which occurs when the model fits well on the training data but works poorly on new data, and unknown data [23]. Moreover, it ensures that the proposed model never sees the same traffic light image twice, hence it enhances the robustness of the model. Rotate, scale, and translate were the operations that were used to manipulate traffic light images.

The original training images were randomly (1) rotated, (2) zoomed-in/zoomed-out, (3) vertically shifted, (4) horizontally shifted, and (5) horizontally flipped to increase the number of the training images. Furthermore, the parameters used for augmentation were randomly chosen from 15^{o} angle, [-10.0, 10.0] %, [-5.0, 5.0] %, [-5.0, 5.0] %, and true for rotate, zoom-in/zoom-out, horizontal-shift, and vertical-shift, horizontal flip, respectively. An example of traffic light images after the augmentation process is shown in Figure 7.

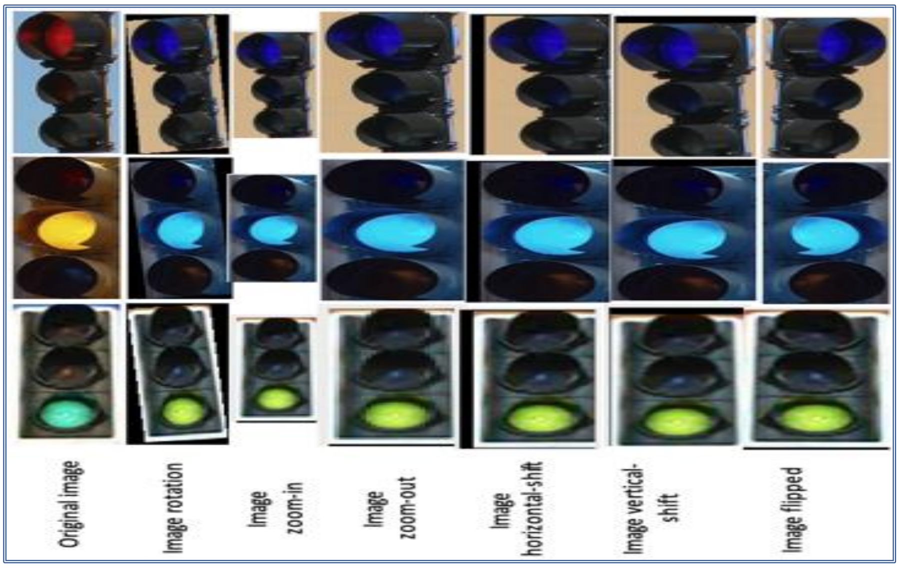

Figure 7: Augmented traffic light images from the LISA dataset

4.4 Feature Representation

An RGB image with a resolution of 299x299 pixels was the input to the proposed model. The three feature mappings stand in for the red, green, and blue color channels, respectively. As mentioned before, the transfer learning-based model is composed of three primary components: the fundamental convolutional block, the Inception modules, and the Softmax classifier.

In this study, five convolution layers numbered conv2d_1 until conv2d_5, make up the fundamental convolutional block. The conv2d_1 layer uses 32 filters of size 3x3 to extract low-level features, resulting in 32-channel feature maps with a size of 149x149. Likewise, 32, 64, 80, and 192-channel feature representations with sizes of 147x147, 147x147, 73x73, and 71x71, respectively, are achieved in the subsequent convolution layers, as shown in Table 2.

Moreover, the core of the proposed model is the Inception module. In the first Inception module (mixed0), 256-channel feature representations are given with a 35x35 resolution. The layer-wise convolutional processes are then improved, and the ensuing Inception modules achieve multi-scale feature representations. Typically, more abstract feature representations are generated as the number of convolutional layers rises [12]. The last Inception module (mixed9) presents 2048-channel feature mappings with a size of 8x8, as seen in the table below. Finally, after flattening 1D vector representations of multi-scale feature representations, 4-neuron output correlates to 4-class probability. Therefore, the class label that the tested traffic light belongs to would thus be the neuron with the greatest probability.

Table 2: The proposed model's architecture configuration

No.	Layers	Output Shape	Kernel Size	No.	Layers	Output Shape	Kernel Size
1	input	(299, 299, 3)	-	**10**	mixed3	(17, 17, 768)	(1, 1), (3, 3)
2	conv2d_1	(149, 149, 32)	(3, 3)	**11**	mixed4	(17, 17, 768)	(1, 1), (3, 3), (1, 7), (7, 1)
3	conv2d_2	(147, 147, 32)	(3, 3)	**12**	mixed5	(17, 17, 768)	(1, 1), (3, 3), (1, 7), (7, 1)
4	conv2d_3	(147, 147, 64)	(3, 3)	**13**	mixed6	(17, 17, 768)	(1, 1), (3, 3), (1, 7), (7, 1)
5	conv2d_4	(73, 73, 80)	(1, 1)	**14**	mixed7	(17, 17, 768)	(1, 1), (3, 3), (1, 7), (7, 1)
6	conv2d_5	(71, 71, 192)	(3, 3)	**15**	mixed8	(8, 8, 1280)	(1, 1), (3, 3), (1, 7), (7, 1)
7	mixed0	(35, 35, 256)	(1, 1), (3, 3)	**16**	mixed9	(8, 8, 2048)	(1, 1), (3, 3), (1, 3), (3, 1)
8	mixed1	(35, 35, 288)	(1, 1), (3, 3), (5, 5)	**17**	mixed10	(8, 8, 2048)	(1, 1), (3, 3), (1, 3), (3, 1)
9	mixed2	(35, 35, 288)	(1, 1), (3, 3)	**18**	output	(4, 4, 1)	-

5. Results and Discussion

To train and evaluate the proposed model, a portion of the LISA traffic light dataset was used. Consequently, the updated dataset contains approximately 2,000 images of traffic lights, which are classified into four classes: red_traffic_light, green_traffic_light, yellow_traffic_light, and not_traffic_light classes for training and validation, such as 80% and 20% of the images to learn the model and validation set, respectively. The proposed system did not require a high volume of data in the training phase since it used a transfer learning-based and pre-trained model, as discussed in [3, 13, 29]. In addition, the proposed model used the following hyperparameters for training and validation, as given in Table 3.

Table 3: The proposed model's hyperparameters

Epoch	94
Batch Size	32
Loos Function	categorical_crossentropy
Optimizer	Adam and learning rate=0.005
Activation Functions	ReLU and Softmax
Dropout	50%

The chosen dataset is used to retrain transfer learning using the TensorFlow machine learning framework. The model is repeatedly trained across 250 epochs, but the highest accuracy is reached at epoch 94 due to the early stopping method, which is a type of regularization used to prevent overfitting when training a learner using an iterative technique like gradient descent. Every porch depicts a cycle of propagation in both the forward and backward directions. In addition, various learning rates are used to evaluate the model's performance; the one where the model performs best in terms of accuracy is 0.005. To determine the appropriate batch size, which refers to the number of samples that will be propagated through the network [23], (16, 32, 64, and 128) were tested. The result was that the model could offer the highest accuracy with batch size 32. In this work, the initial learning rate of 0.005 provides the highest test accuracy of 98.6%. The proposed model's loss and accuracy graph for the LISA traffic light dataset is shown in Figure 8.

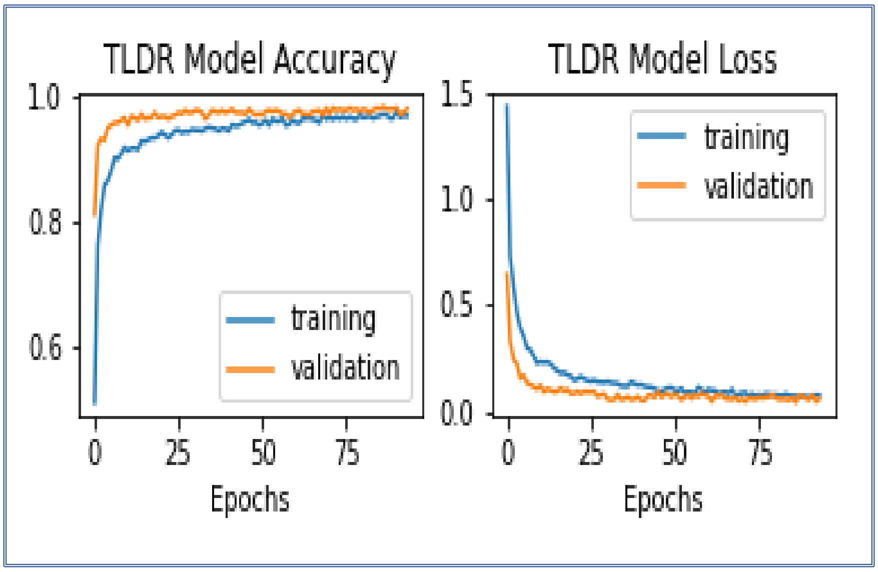

Figure 8: Graphs representing the model's accuracy and loss

There have been several prior implementations for image classification, particularly for image classification of road images, including traffic lights, traffic signs, pedestrians, vehicles, and other objects. These works were chosen based on the use of transfer learning-based and deep learning-based models, but it is challenging to find many papers that are transfer learning-based for comparison because they deal with cutting-edge topics in the construction of deep learning models that depend on previously learned models. Accordingly, some of these competitive studies use deep learning-based models. In addition, as demonstrated in Table 4, the proposed model's outcomes are superior when compared to other models.

Table 4: Comparison of the proposed model's accuracy to relevant previous studies.

No.	Paper	Dataset(s)	Deep learning-based model	Accuracy Result(s)
1	[21]	Own Dataset	CNN-based	94.70%
2	[6]	Bosch Traffic Light Dataset	CNN-based	90.4%
			DCIGN	98.2%
3	[34]	MS COCO Dataset	Faster R-CNN	97.015%
4	[2]	LISA Dataset	Transfer learning-based VGG16	97.17%
5	[35]	LISA Dataset	CNN	92.67%
6	[36]	LISA Dataset	Faster R-CNN and YOLOv4	97.58%
7	**Proposed Model**	**MS COCO and LISA Datasets**	**Transfer learning-based Inception-V3**	**98.6%**

Abbreviations: Faster R-CNN (Region-based Convolutional Neural Network), YOLO (You Only Look Once), DCIGN (Deep Convolution Inverse Graphics Network).

Following the completion of the training model, the proposed model was built, evaluated, and then used on several new traffic light images that had never been seen before. As can be seen in the figure below, it successfully detected and classified the traffic light colors.

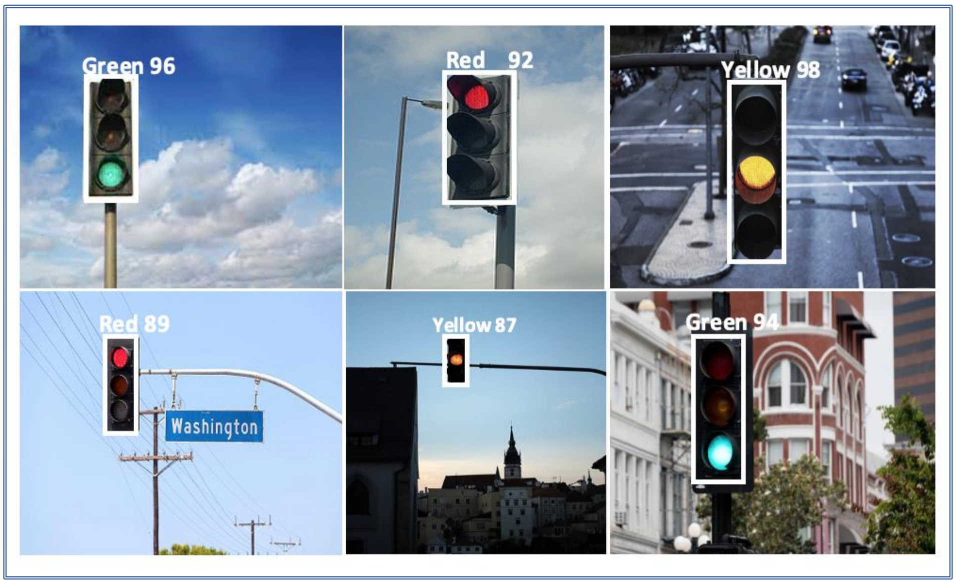

Figure 9: Test results of the proposed model.

6. Conclusion

This study proposed a transfer learning-based model for traffic light detection and recognition. Images from the LISA traffic light dataset were prepared and improved using data pre-processing techniques. Additionally, a data augmentation technique was used to augment the number of training images, which may improve the model's robustness. TensorFlow Object Detection API, which is pre-trained on the MS COCO dataset [31], is used to detect traffic lights in LISA traffic light images [33]. The detected traffic lights are then cropped and prepared to feed to the CNN Inception-V3, which acts as the proposed model's base model.

At various learning rates, the proposed model was retrained across 94 epochs. With the best recognition accuracy of 98.6% at the learning rate of 0.005, the accuracy test results demonstrate that the transfer learning-based model is efficient for detecting and recognizing traffic lights. More to say, according to the proposed model's results, a transfer learning-based model may provide reliable, repeatable outcomes. This makes it useful for maintaining various types of traffic infrastructure, including facilities for lane marking, traffic sign recognition, and roadside prediction. The CNN Inception-V3 model, which is based on transfer learning, enhances accuracy in the field of autonomous driving and qualifies the system for real-time applications.

On the other hand, the proposed model still requires to be improved in the future from the following aspects. (1) The model will become more reliable as a result of training and testing it using additional traffic light datasets that are widely available. (2) The architecture of the proposed model can be enhanced using newly developed techniques in the transfer learning-based model. (3) Using real-world data to test and evaluate the model allows it to be more realistic.

Compliance with Ethical Standards

Conflicts of Interest/Competing Interests: The authors declare that there are no conflicts of interest related to the publication of this study.

References

[1] G. Symeonidis, P. P. Groumpos and E. Dermatas, "Traffic Light Detection and Recognition Using Image Processing and Convolution Neural Networks," *In Conference on Creativity in Intelligent Technologies and Data Science,* Springer, Cham, pp. 181-190, 2019.

[2] A. Gupta and A. Choudhary, "A Framework for Traffic Light Detection and Recognition using Deep Learning and Grassmann Manifolds," *In 2019 IEEE Intelligent Vehicles Symposium (IV),* IEEE, pp. 600-605, 2019.

[3] S. Gautam and A. Kumar, "Automatic Traffic Light Detection for Self-Driving Cars Using Transfer Learning," *In Intelligent Sustainable Systems,* Springer, Singapore, pp. 597-606, 2022.

[4] D. H. Widyantoro and K. I. Saputra, "Traffic Lights Detection and Recognition based on Color Segmentation and Circle Hough Transform," *In 2015 International Conference on Data and Software Engineering (ICoDSE),* IEEE, pp. 237-240, 2015.

[5] R. Gokul, A. Nirmal, K. M. Bharath, M. P. Pranesh and R. Karthika, "A Comparative Study between state-of-the-art Object Detectors for Traffic Light Detection," *In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE),* IEEE, pp. 1-6, 2020.

[6] G. Mahesh and T. Satich Kumar, "Real Time Traffic Light Detection by Autonomous Vehicles using Artificial Neural Network Techniques," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 10, pp. 2129-2133, 2019.

[7] M. Hirabayashi, A. Sujiwo, A. Monrroy, S. Kato and M. Edahiro, "Traffic Light Recognition using High-Definition Map Features," *Robotics and Autonomous Systems*, vol. 111, pp. 62-72, 2019.

[8] V. John, K. Yoneda, Z. Liu and S. Mita, "Saliency Map Generation by the Convolutional Neural Network for Real-Time Traffic Light Detection using Template Matching," *IEEE Transactions on Computational Imaging*, vol. 1, no. 3, pp. 159-173, 2015.

[9] S. S. Pillai, B. Radhakrishnan and L. P. Suresh, "Detecting tail lights for analyzing traffic during night using image processing techniques," *In 2016 International Conference on Emerging Technological Trends (ICETT),* IEEE. pp. 1-7, 2016.

[10] S. Kulkarni, S. Harnoorkar and P. E. Pintelas, "Comparative Analysis of CNN Architectures," *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, no. 6, pp. 1459-1464, 2020.

[11] A. Khan, A. Sohail, U. Zahoora and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial intelligence review,* vol. 53, no. 8, pp. 5455-5516, 2020.

[12] M. Hussain, J. J. Bird and D. F. Faria, "A Study on CNN Transfer Learning for Image Classification," *In UK Workshop on Computational Intelligence,* Springer, Cham, pp. 191-202, 2018.

[13] A. R. Pathak and A. C. Elster, "Applying Transfer Learning to Traffic Surveillance Videos for Accident Detection," *In 2022 International Conference on Applied Artificial Intelligence (ICAPAI),* IEEE, pp. 1-7, 2022.

[14] C. Lin, L. Li, W. Luo, K. C. Wang and J. Guo, "Transfer learning-based Traffic Sign Recognition using Inception-v3 Model," *Periodica Polytechnica Transportation Engineering,* vol. 47, no. 3, pp. 242-250, 2018.

[15] G. Mu, Z. Xinyu, L. Deyi, Z. Tianlei and A. Lifeng, "Traffic light detection and recognition for autonomous vehicles," *The Journal of China Universities of Posts and Telecommunications*, vol. 22, no. 1, pp. 50-56, 2015.

[16] Z. Shi, Z. Zou and C. Zhang, "Real-Time Traffic Light Detection with Adaptive Background Suppression Filter," *IEEE Transactions on Intelligent Transportation Systems,* vol. 17, no. 3, pp. 690-700, 2015.

[17] R. De Charette and F. Nashashibi, "Traffic Light Recognition using Image Processing Compared to Learning Processes," *In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, pp. 333-338, 2009.

[18] T. A. Touma and H. K. Abbas, "Traffic Light Detection in Autonomous Vehicles Using Image Processing Methods," *Journal of the College of Basic Education,* vol. 2, pp. 97-109, 2021.

[19] C. Wang, G. Zhang, W. Zhou, Y. Rao and Y. Lv, "Traffic Lights Detection Based on Deep Learning Feature," *In International Conference on Internet of Things as a Service*, Springer, Cham, pp. 382-396, 2019.

[20] A. Madhu and V. Nair, "Traffic Sign Detection and Recognition for Automated Driverless Cars Based On SSD," *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)*, vol. 9, no. 7, pp. 5550-5553, 2020.

[21] D. Wang, H. Bao and F. Zhang, "CTL-DNNet: Effective Circular Traffic Light Recognition with a Deep Neural Network," *International Journal of Pattern Recognition & Artificial Intelligence*, vol. 31, no. 11, pp. 1-15, 2017.

[22] H. K. Dhahir and N. H. Salman, "A Review on Face Detection Based on Convolution Neural Network Techniques", *Iraqi Journal of Science*, vol. 63, no. 4, pp. 1823–1835, 2022.

[23] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie and L. Farhan, "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions," *Journal of Big Data*, vol. 8, no. 1, pp. 1-74, 2021.

[24] Y. Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541-551, 1989.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818-2826, 2016.

[26] X. Xia, C. Xu and B. Nan, "Inception-v3 for Flower Classification," *In 2017 2nd International Conference on Image, Vision, and Computing (ICIVC)*, IEEE, pp. 783-787, 2017.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein and A. C. Berg, "Imagenet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, 2015.

[28] H. Alaeddine and M. Jihene, "Deep Network in Network," *Neural Computing and Applications*, vol. 33, no. 5, pp. 1453-1465, 2021.

[29] S. Gupta, M. Rawat and A. S. Rao, "Accident Detection and Prediction with Notification Alert System," *In Advances in Mechanical Engineering and Technology*, Springer, Singapore, pp. 281-289, 2022.

[30] H. A. Ahmed and E. A. Mohammed, "Detection and Classification of The Osteoarthritis in Knee Joint Using Transfer Learning with Convolutional Neural Networks (CNNs)," *Iraqi Journal of Science*, vol. 63, no. 11, pp. 5058–5071, 2022.

[31] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," *In European Conference on Computer Vision*, Springer, Cham, pp. 740-755, 2014.

[32] L. Zhu, F. Lee, J. Cai, H. Yu and Q. Chen, "An Improved Feature Pyramid Network for Object Detection," *Neurocomputing*, vol. 483, pp. 127-139, 2022.

[33] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund and M. M. Trivedi, "Vision for Looking at Traffic Lights: Issues, Survey, and Perspectives," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1800-1815, 2016.

[34] T. V. Janahiraman and M. S. M Subuhan, "Traffic Light Detection using Tensorflow Object Detection Framework," *In 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, IEEE, pp. 108-113, 2019.

[35] D. Vitas, M. Tomic and M. Burul, "Traffic Light Detection in Autonomous Driving Systems," *IEEE Consumer Electronics Magazine*, vol. 9, no. 4, pp. 90-96, 2020.

[36] Q. Wang, Q. Zhang, X. Liang, Y. Wang, C. Zhou and V. I. Mikulovich, "Traffic Lights Detection and Recognition Method based on the Improved YOLOv4 Algorithm," *Sensors*, vol. 22, no. 1, p.200, 2021.