ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 12, Issue 1, January 2025

Traffic Light Detection and Recognition Using Machine Learning Algorithms

[1] Mutiq Katib Alrashidi, [2] Asala Ali, [3] Ahmed Al Ahdal

[1] Department of Artificial Intelligence, University of Hail, Saudi Arabia
[2] Department of Computer Sciences & Information Systems, Najran University, Saudi Arabia
[3] Department of Computer Science and Engineering, Lovely Professional University, Phagwara, Jalandhar, India
Corresponding Author Email: [1] Mutiq.AlRashidi@uoh.edu.sa, [2] Asalaalii10@gmail.com, [3] Ahmed.alahdal21@gmail.com

Abstract— ***Research on traffic light detection and recognition (TLR) is growing gradually every year. Additionally, Machine Learning (ML) has been extensively applied, not solely within TLR investigations but across various fields where data generalization and the automation of human behavior offer practical benefits. This study discusses many artificial intelligence and machine learning techniques for identifying and recognizing traffic lights. It adopts a strategy of initially classifying and then identifying. The strategy begins by locating the traffic light region, extracting it, conducting image processing tasks, and then delivering the processed image to the recognition method, The LISA dataset, an open-source resource utilized in this research, comprises 43,007 frames of continuous video sequences designated for both testing and training, along with 113,888 annotated traffic lights, To collect the dataset, a stereo camera was installed on the roof of a car, which was driven at various times, including both day and night, while encountering a range of lighting and weather conditions. The accuracy of the suggested algorithms was impressive, with the decision tree scoring 97%, logistic regression scoring 98%, and the support vector machine (SVM) achieving the highest score of 98.62%. These outcomes reflect the successful application of artificial intelligence and machine learning algorithms in effectively recognizing and predicting traffic light signals in different lighting and weather scenarios.***

Keywords - traffic light detection and recognition, machine learning, support vector machines, Autonomous.

I. INTRODUCTION

Autonomous driving is a significant research concern in autonomous vehicle technology. The principal objective of self-driving vehicles is to be an alternative to human drivers with advanced computer systems while preserving and improving safety and efficiency [1]. A significant part of achieving this goal is the computer system's capability to perceive and recognize its surrounding environment, which includes the road, pedestrians, traffic signs, and other vehicles. In such an atmosphere, mimicking human visual perception and behavioural responses is critical [2]. In autonomous driving, traffic signal recognition is incredibly crucial. Autonomous terrestrial vehicles must reliably logically sense traffic lights and determine their present circumstance, such as red, yellow, or green. Academic studies on autonomous vehicles usually refer to this problem as Traffic Light Recognition (TLR). TLR is the process of creating models and algorithms that enable autonomous vehicles to identify, categorize, and detect [3–4] traffic lights under actual conditions. The defined methodologies are based on computer vision techniques such as image processing, feature extraction, and artificial intelligence. Suggested approaches include traditional machine learning models like Support Vector Machines (SVM) and Boosting, as well as advanced deep learning models that use convolutional neural networks. The goal is to correctly evaluate visual data from integrated cameras or sensors and make intelligent decisions based on recognized traffic light positions. The constantly changing vehicle movement dynamics and the variety of surrounding environments create significant issues for accurate traffic light recognition [6]. These challenges include unfamiliar environments, interference from other light sources, weather and illumination impacts, changes in viewing angles and sizes, various appearances of traffic lights, different types of traffic lights, autofocus and automatic white balance issues, and the need for real-time processing capabilities [7-8]. Overcoming these challenges requires advanced algorithms that can adapt to varying environments, handle interference, account for weather conditions, cope with changing perspectives, accommodate different traffic light configurations, address other traffic lights, mitigate camera-related factors, and ensure prompt real-time decision-making. Ongoing research efforts aim to develop robust computer vision techniques and machine learning frameworks that enhance the accuracy and reliability of recognition systems for traffic lights in complex driving environments [9]. Both autonomous vehicles and human operators must accurately identify the traffic signals relevant to their route. While human drivers can effortlessly discern these signals, an effective algorithm or criteria set is essential for distinguishing traffic lights in various scenarios. To address this challenge, traffic light detection is frequently integrated with deep learning methodologies[10]. These models can detect information about the traffic lights' position, direction, and other properties. By utilizing machine learning and deep learning models, an autonomous vehicle can proactively determine the presence of traffic lights in its vicinity [11]. This integration uses real-time sensor inputs, including camera images, to recognize and classify traffic lights in an instance. This information helps identify appropriate traffic signals, improve decision-making, and ensure more precise responses at intersections [12]. Using real-time sensor data with machine learning and deep learning improves the accuracy and reliability of identifying traffic lights, leading to safer and more efficient autonomous driving systems [13]. Advancements in sensor technology, including LiDAR and radar systems, have significantly improved traffic light identification in autonomous cars. Sensors supplement camera data and improve vehicle perception. Autonomous vehicles can make more reliable decisions about traffic lights by incorporating data from many sensors. Developing and deploying autonomous driving technology requires overcoming technological and regulatory challenges.

To integrate autonomous vehicles safely within current transportation systems, they must comply with traffic laws and regulations, including traffic light recognition. Collaborations between researchers, industry stakeholders, policymakers, and regulatory bodies are necessary to establish standards and guidelines for autonomous vehicles' behavior at intersections and traffic lights.

A. Motivation

Traffic light recognition (TLR) has been investigated in extensive detail. However, there are still several issues. Current methods emphasize identifying and classifying any traffic light in a location without considering the relevance of certain lights for specific conditions, such as those in the direction the vehicle is heading. Other challenges with recognition include various illumination environments, such as at night, recognition right away at longer distances, and recognition under challenging conditions (such as rain or snow). While deep learning methods for common recognition of objects, such as YOLO, have demonstrated promise, a large number of training samples are needed for deep learning techniques, but the datasets for yellow traffic lights are frequently small, reducing performance in that group.

B. Research Objectives

The This research aims to evaluate machine learning algorithms for traffic light detection and recognition (TLR) under varied environmental conditions. It seeks to investigate suitable machine learning techniques for effective TLR, develop a sequential TLR strategy including detection, extraction, processing, and recognition, assess the proposed approach using the LISA dataset, provide insights into algorithm performance for automated traffic management, and offer recommendations for practical TLR application and further research. In addition, 1- To investigate the use of machine learning algorithms for traffic light detection and recognition (TLR). 2- Develop a systematic approach for TLR by classifying traffic light regions, extracting critical features, applying image processing techniques, and passing processed images to recognition algorithms. 3- Evaluate machine learning techniques for TLR using the LISA open-source dataset.

II. LITERATURE REVIEW

This section covers studies investigating the use of AI and machine learning for traffic light detection and recognizing (TLR). Furthermore, research and a review of literature extending machine learning approaches are needed for an exhaustive assessment. Deep learning is growing in prominence for recognizing traffic lights. Weber et al. [14] developed DeepTLR, a neural network modeled after AlexNet, to identify and classify traffic lights. With nine convolutional layers, the network can handle 640x480 RGB images at 35 FPS [14]. The network consists of nine convolutional layers and achieves a processing speed of 35 frames per second (FPS) when working with 640x480 RGB images [15].

Behrendt et al. [16] modified the YOLO (You Only Look Once) model to detect traffic light candidates. They divided the upper part of the camera frames into three crops and independently fed them into YOLO for detection. To improve the detection of distant traffic lights, they increased the detection grid size from 7x7 to 11x11. The classification task was removed from YOLO, and a custom CNN was used to classify detected objects as red, yellow, green, or background (non-traffic light objects) [17]. To reduce false positives, the projected bounding boxes were extended and rescaled. To track traffic lights, an odometry-based motion model was used in conjunction with another neural network. The models were trained and tested on the publicly accessible Bosch Small Traffic Lights dataset.

Omachi et al. [18] investigated the influence of normalized RGB images from vehicle-mounted cameras on traffic light detection. They employed threshold conditions and edge detection to recognize traffic signals in photographs. Charette et al. [19] used grayscale images, spotlight detection, and template matching to identify traffic lights, with a focus on comprehending their lighting patterns. Fairfield et al. [20] and Huang et al. devised a three-dimensional traffic light mapping technique to forecast light locations in images captured by vehicle-mounted cameras. After determining the positions, they employed blob segmentation to evaluate the operational state of traffic signals. [21] This study compared network models for detecting general objects using the COCO dataset. The study investigated 14 meta-architectures with different feature extractors and network models. The feature extractors included VGGNet [22], MobileNet [23], Inception-v2 [24], Resnet-101 [25], and Inception-Resnet-v2. The study analyzed three network models: Faster-RCNN, R-FCN, and SSD. SSD, like YOLO, improved performance for medium and big objects [26].

However, it performed less effectively than Faster R-CNN and R-FCN for small objects [27]. Omachi et al. [28] established a methodology for detecting and recognizing traffic lights. This method involves constructing a mathematical model for traffic lights and categorizing pixels in the input image into five kinds. Pixels with traffic light colors underwent Sobel edge detection for validation subsequently a voting method was used, combining the model and identified edges to reliably identify traffic lights, resulting in an 89% accuracy rate.

Zhang et al. [29] used a comprehensive approach to traffic light identification, including color segmentation, speckle detection, and structural feature extraction approaches. Initially, these algorithms were used for identifying potential locations within the image that might correlate to traffic signals. The discovered points and blocks inside those potential locations were then scored to further enhance the detection procedure. With this multiple phases approach, the system achieved an excellent average detection accuracy of up to 96.07%.

Deng et al. [30] suggested an approach for recognizing traffic lights that blends deep learning techniques with contextual data. They used convolutional neural networks (CNNs) to extract visual features from traffic light images and added contextual signals, such as the traffic light's position in the scene and the presence of other objects, to improve accuracy and resilience.

Simonyan and Zisserman [31-32] developed a real-time system for detecting and recognising traffic lights using deep neural networks. They used a sliding window strategy in conjunction with deep learning methodologies, notably CNNs, to enable accurate traffic signal detection and recognition in real-time scenarios. The system's high level of precision and efficiency make it an ideal solution for time-sensitive problems.

Zhou et al. [33] developed the region-based CNN (R-CNN) framework for traffic light identification by including feature fusion techniques. They employed feature fusion to combine visual and spatial information from traffic light photographs, resulting in more accurate and robust traffic light identification and recognition, even in challenging environments.

Liu et al. [34] presented a multi-task learning technique that use a convolutional neural network (CNN) to learn both traffic light detection and state recognition simultaneously. Their multi-task CNN performed better in traffic light detection and right state recognition because it used similar features and correlations between the two objectives.

III. MATHALOGY

This work falls under applied research and focuses on applying machine learning techniques to develop and deploy a Traffic Light Recognition (TLR) system.

This study emphasizes how important it is to detect and recognize traffic lights as a prerequisite for the smooth integration of driverless vehicles into the broader context of smart cities. The smart city model envisions a unified urban landscape in which technological advancements are used to improve many aspects of urban life, including transportation systems, energy usage, public services, and infrastructure oversight. In this light, the successful implementation of self-driving vehicles represents a significant step toward achieving the goals of smart city development. Figure 1 shows a block diagram explaining the proposed methodology.

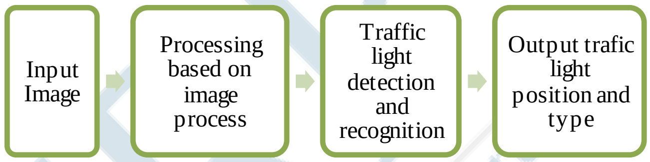

Fig 1. Proposed Methodology.

The flowchart outlined the process of detecting and recognizing traffic lights in an input image.

1. Input Image: The process begins with an input image that contains a scene captured by a camera, for traffic lights

2. Image Processing: The input image undergoes various image processing techniques to enhance the quality of the image and extract relevant features that can help in detecting traffic lights. Image processing techniques include operations such as resizing, noise reduction, color correction, edge detection, and other preprocessing steps to prepare the image for traffic light detection.

3. The LISA (Laboratory for Intelligent and Safe Automobiles) dataset, developed by the University of California, San Diego, collects video sequences from moving automobiles to promote studies in computer vision and autonomous driving. The simulator offers realistic driving scenarios on highways and city streets, with varying weather, illumination, traffic patterns, and road kinds. The following are some key characteristics:

4. Output Traffic Light Position and Type: Finally, the output of the system includes the position information (coordinates) of each detected traffic light within the image and the corresponding type or state of each traffic light.

A. Dataset Collection

The LISA (Laboratory for Intelligent and Safe Automobiles) dataset, created by the University of California, San Diego, is a comprehensive collection of video sequences captured from moving vehicles to advance research in computer vision and autonomous driving. It features a wide variety of realistic driving scenarios on highways and city streets, under different weather and lighting conditions, and with diverse traffic patterns and road types. Key attributes include:

1. Diverse Driving Scenarios: Covers urban and highway

environments with varying weather (rain, fog, sunshine) and lighting (daytime, nighttime) conditions.
2. Features extensive annotations for tasks like object detection, lane detection, pedestrian detection, and traffic sign recognition.
3. Vast and Varying Data: A left-sided stereo camera recorded more than 43,000 frames and 113,888 traffic light annotations in JPG format.
4. Open Access: Promoting cooperation and innovation in autonomous driving technology, this approach is accessible to students, developers, and academics.
5. Data Augmentation: To improve dataset diversity and model adaptation, adjust image brightness, flipping, and rotation.

The LISA dataset is used by researchers to train and evaluate computer vision algorithms, allowing for the creation of robust and reliable autonomous driving systems by using its vast and diverse data.

1) Data Preprocessing

Data Data preprocessing for the LISA dataset includes:

1. Image Resizing: Standardizing all images to a uniform size suitable for machine learning models.
2. Noise Reduction: Applying techniques to enhance image quality and eliminate irrelevant information that may obstruct traffic light detection.
3. Color and Brightness Normalization: Adjusting colors and brightness levels to ensure consistency across different lighting conditions.

2) Resampling

Resampling is a machine learning technique that addresses imbalances, overfitting, and underfitting. To balance the dataset, it either oversamples or undersamples. We created a function called "resample_dataset" to achieve a more balanced and representative dataset for better model performance.

3) Feature Extraction

Feature extraction aims to identify and extract meaningful information from preprocessed images for machine learning algorithms, reducing dimensionality and computational resources. Common features for traffic light detection include color histograms, edge information, shape characteristics, and texture descriptors. Techniques used involve cropping images from 640×480 to 640×240 to focus on areas where traffic lights appear and converting colors from RGB to HSV. Additionally, binarization is applied to convert images into a binary format based on threshold values, enhancing contrast between objects and the background. These techniques improve the accuracy of traffic light detection and recognition models by enabling efficient and effective feature identification.

4) Model Selection and Training:

In This work we used an appropriate machine learning algorithm model to detect and recognize traffic lights. The output of the method is a computational modeling model based on data. Divide the dataset into training and validation sets. Train the model on a subset of the data and evaluate its performance on new data.

Optimize model performance by training on tagged images from the LISA dataset and adjusting hyperparameters. Logistic Regression, Support Vector Machines (SVM), and Decision Trees are commonly used machine learning algorithms for opting for models in the LISA dataset, which may include geographical data.

5) Decision Trees:

Decision Trees is a tree-like structure where nodes represent features, branches represent decisions, and leaves represent outcomes. The tree is built by recursively partitioning data based on feature values to create pure subsets. Training involves selecting the best feature to split data at each node and growing the tree until stopping criteria are met. The tree is traversed from root to leaf based on feature values to predict outcomes during prediction [34]. Decision Trees are easy to understand, handle both numerical and categorical data, and implicitly perform feature selection. However, they can overfit, be sensitive to data variations, and may only generalize well with proper pruning or regularization [35]. Overall, Decision Trees are versatile and effective for various machine-learning tasks.

6) Logistic Regression:

One supervised machine learning technique used for binary classification is logistic regression. Using one or more predictor variables in addition to observed data, it models the probability of a binary result using a logistic function. Interestingly, despite what its name might imply, logistic regression is classified as a linear model. In order to provide an output that represents the likelihood of the positive class, the logistic (sigmoid) function is essential to the linear combination of input features and their weights. The simplicity, effectiveness, and interpretability of this method are widely recognized, and it works especially well when dealing with nonlinear interactions between independent and dependent variables [36].

7) Support Vector Machine (SVM):

Support vector machines. This robust supervised machine learning approach handles problems which includes outlier identification, regression, and classification. SVM classifies data points into discrete classes by identifying the optimal hyperplane in feature space. This hyperplane optimizes the margin, which is the distance between the hyperplane and the nearest data points or support vectors [38]. SVM can perform both linear and nonlinear classification tasks utilizing various kernel functions, including linear, polynomial, radial basis function (RBF), and sigmoid kernels. SVM is suitable for a variety of applications, including pattern recognition, image classification, text categorization, and bioinformatics,

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 12, Issue 1, January 2025

because to its high-dimensional performance and resistance to overfitting [39].

8) Testing:

Testing In machine learning, testing is the process of evaluating a trained model's performance and generalizability using previously unknown data. It is an important stage in the machine learning process that ensures the model can make accurate predictions on fresh, previously unseen samples. Testing entails providing the model with a distinct dataset, known as the test set, that was not used during the training phase. After training a machine learning algorithm on an initial training data set, it is tested on a secondary (or tertiary) data set. Predictive models are designed always to have some undisclosed potential that needs to be evaluated rather than merely viewed from a programming standpoint.

9) Predicting

In machine learning, prediction is the use of a trained model to forecast a target variable based on input data. Once trained on a dataset, a machine learning model may predict fresh data. The model predicts the target variable based on input attributes from new data and training patterns. Prediction accuracy can be measured using many metrics, including accuracy, precision, and recall [37].

Machine learning relies heavily on prediction for different applications, including classification and regression. The accuracy of the predictions can be evaluated using

various metrics, such as accuracy, precision, recall, etc. Predicting is a key aspect of machine learning and is used in various applications such as classification, regression, clustering, and more. They are using a trained machine learning model to predict new, unseen data points. After training on labelled data, the model learns patterns and relationships between input features and output labels. the prediction method of the trained model object is typically used for generating predictions.

10) Evaluating:

Evaluate a machine learning model's performance on a distinct dataset to determine its ability to generalize to new data. To evaluate a model's efficacy, metrics like as accuracy, precision, recall, and F1-score are used. Sci-kit-learn offers Table 1: displaying the precisions assessment metrics and approaches, including classification accuracy, precision, recall, F1-score, confusion matrix, and ROC curves, to measure model performance.

IV. RESULT

The database comprises 43,007 frames and 113,888 annotated traffic lights from continuous test and training video sequences. These scenes were recorded with a stereo camera installed on the top of a car, which traveled through a variety of settings with different lighting and weather patterns both during the day and at night. But for traffic light analysis, just the left camera view was used, essentially depending on monocular vision. This approach mirrors real-world scenarios where stereo information may not be available. Additionally, Fig1 displays the initial rows and columns of the dataset

This dataset includes 43,007 frames of continuous test and training video sequences with 113,888 traffic light labels. Using only the left view, the recordings were made with a stereo camera and were taken in various weather and lighting conditions. Afunction image_traffic_light_crop(df) processes

	filename	target	x1	y1	x2	y2	image_path
0	dayClip13--00000.jpg	go	463	177	490	222	../input/lisa-traffic-light-dataset/dayTrain/d...
1	dayClip13--00000.jpg	go	616	185	637	235	../input/lisa-traffic-light-dataset/dayTrain/d...
2	dayClip13--00000.jpg	go	888	300	912	345	../input/lisa-traffic-light-dataset/dayTrain/d...
3	dayClip13--00001.jpg	go	461	176	488	216	../input/lisa-traffic-light-dataset/dayTrain/d...
4	dayClip13--00001.jpg	go	617	197	641	227	../input/lisa-traffic-light-dataset/dayTrain/d...
...	...	...	...	...	...	...	...
70580	nightClip1--00492.jpg	go	872	17	958	143	../input/lisa-traffic-light-dataset/sample-nig...
70581	nightClip1--00493.jpg	go	938	5	1028	131	../input/lisa-traffic-light-dataset/sample-nig...
70582	nightClip1--00494.jpg	go	1006	6	1094	117	../input/lisa-traffic-light-dataset/sample-nig...
70583	nightClip1--00495.jpg	go	1081	0	1171	110	../input/lisa-traffic-light-dataset/sample-nig...
70584	nightClip1--00496.jpg	go	1158	0	1246	101	../input/lisa-traffic-light-dataset/sample-nig...

70585 rows × 7 columns

Fig 2. dataset description

the annotation DataFrame by extracting image paths, filenames, target classes, and bounding box coordinates. It uses OpenCV to read and crop each image to the region of interest, converts the color space from BGR to RGB, and stores the cropped images in a dictionary img_values. These cropped images are then visualized in a grid of subplots (Figure 5) with target class names, with axis labels hidden for clarity.

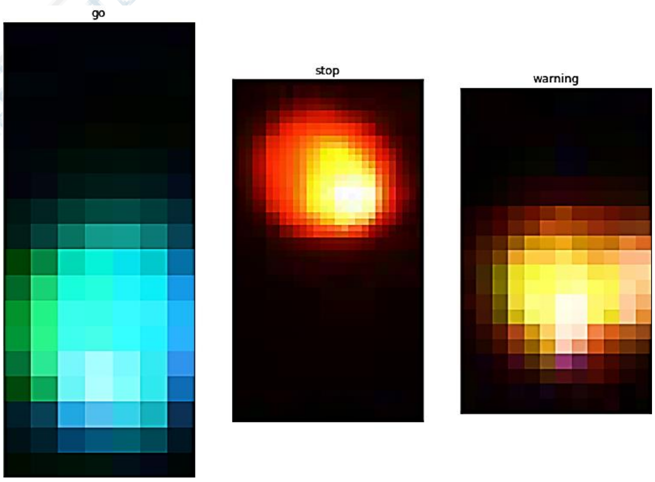

Fig 3. sample of images from different classes

Additionally, the image_binarization (img_values) function converts the colored cropped images into binary images by resizing them to 30x50 pixels, converting to grayscale, applying Otsu's thresholding, inverting the binary values, and storing the results in binary_img_values (Figure 6). This preprocessing enhances feature extraction for traffic light detection.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 12, Issue 1, January 2025

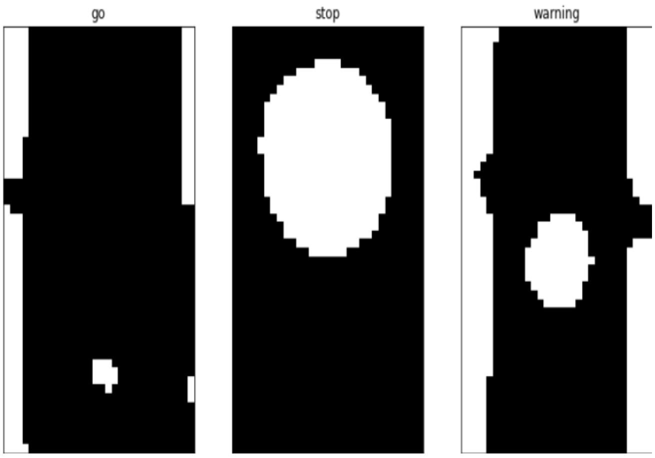

Fig 4. Binarization Images

Based on the results presented in Table 3, the Support Vector Machine (SVM) algorithm outperforms both Decision Tree and Logistic Regression in classifying "Go," "Stop," and "Warning" signals. Specifically, SVM achieves an overall accuracy of 98%, indicating its high reliability in correctly classifying traffic light states.

Table 1. Performance Comparison of Machine Learning Algorithms with accuracy

Algorithm	Accuracy	Precision		
		Go	***Stop***	***Warning***
Decision Tree	97%	0.97	0.98	0.97
Logistic Regression	98%	0.97	0.99	0.96
SVM	98%	0.98	0.99	0.98

Accuracy: Both Logistic Regression and SVM achieve the highest accuracy of 98%, slightly outperforming the Decision Tree at 97%.

Precision and Recall: SVM demonstrates superior precision and recall across all classes ("Go," "Stop," and "Warning"), ensuring accurate identification of both positive and negative instances.

F1-score: SVM maintains the highest F1-scores, particularly excelling in the "Warning" class, indicating a balanced performance between precision and recall.

Detecting a blue light labeled as "Go" indicates that the traffic light is currently displaying a green light. Green lights typically signify that it is safe for vehicles and pedestrians to proceed. As shown in Figure 3, the images visually depict recognized objects and their predicted labels, demonstrating the model's ability to accurately identify and classify traffic light states.

Table 2. Performance Comparison of Machine Learning for F1-Score

Algorithm	F1-score		
	Go	***Stop***	***Warning***
Decision Tree	0.97	0.98	0.98
Logistic Regression	0.96	0.99	0.98
SVM	0.98	0.99	0.98

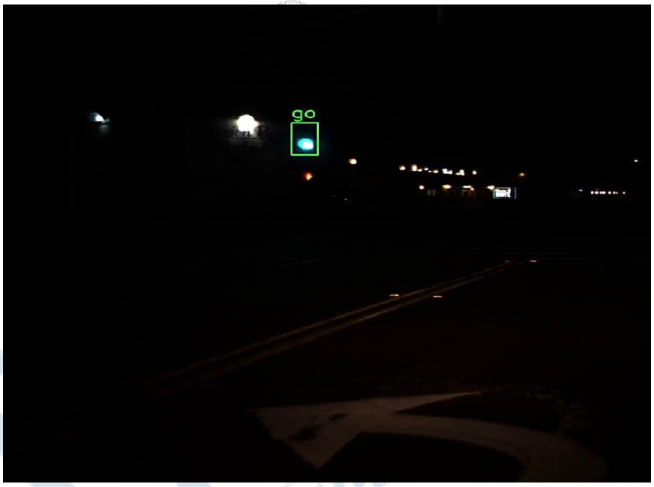

Fig 5. Detection and recognizing correct result

Overall, SVM emerges as the most reliable algorithm for classifying traffic light signals due to its superior performance metrics, making it highly suitable for autonomous driving applications.

V. CONCLUSION AND FUTURE WORK

Traffic Light Recognition (TLR) has significantly advanced through Machine Learning techniques, achieving high accuracies of 97% with Decision Trees, 98% with Logistic Regression, and 98.62% with SVM. The systematic methodology—including region identification, feature extraction, and image processing—proved effective across diverse lighting and weather conditions, demonstrating the reliability of ML approaches for autonomous driving systems. Future work aims to enhance TLR by leveraging transfer learning to improve algorithm performance, utilizing Generative Adversarial Networks (GANs) to generate synthetic traffic light images, and developing multi-view recognition algorithms to accurately detect traffic lights from various angles. These initiatives seek to boost the accuracy and robustness of TLR systems in real-world applications, contributing to more intelligent and dependable autonomous driving technologies.

REFERENCES

[1] N. Fairfield and C. Urmson, "Traffic Light Mapping and Detection," in *Proc. Int. Conf. Robotics and Automation (ICRA)*, Shanghai, China, 2011, pp. 1-6.

[2] D. Barnes, W. Maddern, and I. Posner, "Exploiting 3D Semantic Scene Priors for Online Traffic Light

Interpretation," in *Intelligent Vehicles Symposium (IV)*, Seoul, South Korea, 2015, pp. 573–578.

[3] M. B. Jensen et al., "Vision for Looking at Traffic Lights: Issues, Survey, and Perspectives," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1800–1815, 2016.

[4] R. D. Charette and F. Nashashibi, "Traffic Light Recognition using Image Processing Compared to Learning Processes," in *Proc. Int. Conf. Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, 2009, pp. 333–338.

[5] R. D. Charette and F. Nashashibi, "Real-Time Visual Traffic Lights Recognition Based on Spot Light Detection and Adaptive Traffic Lights Templates," in *Intelligent Vehicles Symposium (IV)*, Xi'an, China, 2009, pp. 358–363.

[6] G. Siogkas, E. Skodras, and E. Dermatas, "Traffic Lights Detection in Adverse Conditions using Color, Symmetry and Spatiotemporal Information," in *Proc. Int. Conf. Computer Vision Theory and Applications (VISAPP)*, vol. 1, 2012, pp. 1-10.

[7] M. P. Philipsen et al., "Traffic Light Detection: A Learning Algorithm and Evaluations on Challenging Dataset," in *Intelligent Transportation Systems Conference (ITSC)*, Las Palmas, Spain, 2015, pp. 2341–2345.

[8] R. F. Berriel et al., "Deep Learning Based Large-Scale Automatic Satellite Crosswalk Classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 9, pp. 1513–1517, 2017.

[9] R. F. Berriel et al., "Automatic Large-Scale Data Acquisition via Crowdsourcing for Crosswalk Classification: A Deep Learning Approach," *Comput. Graph.*, vol. 68, pp. 32–42, 2017.

[10] R. Guidolini et al., "Handling Pedestrians in Crosswalks Using Deep Neural Networks in the IARA Autonomous Car," in *Int. Joint Conf. Neural Networks (IJCNN)*, Rio de Janeiro, 2018, pp. 1–8.

[11] J. Redmon et al., "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788.

[12] J. Levinson, J. Askeland, J. Dolson, and S. Thrun, "Traffic light mapping, localization, and state detection for autonomous vehicles," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011.

[13] U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein, and R. G. Herrtwich, "Making Bertha See," in *Proc. Int. Conf. Computer Vision Workshop (ICCVW)*, 2013.

[14] Z. Ouyang, J. Niu, Y. Liu, and M. Guizani, "Deep CNN-Based Real-Time Traffic Light Detector for Self-Driving Vehicles," *IEEE Trans. Mob. Comput.*, vol. 19, pp. 300–313, 2020.

[15] C. I. Kim, J. Park, Y. Park, W. Jung, and Y. S. Lim, "Deep Learning-Based Real-Time Traffic Sign Recognition System for Urban Environments," *Infrastructures*, vol. 8, p. 20, 2023.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Lecture Notes in Computer Science*, 2015.

[17] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014.

[18] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv*, 2018.

[19] R. Wang, "Optimal method of intelligent traffic signal light timing based on genetic neural network," *Adv. Transp. Stud.*, vol. 1, pp. 3–12, 2021.

[20] Al Ahdal, A., Chawla, P. (2023). A Review on Machine Learning-Based Patient Scanning, Visualization, and Monitoring. In: Gupta, D., Khanna, A., Bhattacharyya, S., Hassanien, A.E., Anand, S., Jaiswal, A. (eds) International Conference on Innovative Computing and Communications. Lecture Notes in Networks and Systems, vol 473. Springer, Singapore. https://doi.org/10.1007/978-981-19-2821-5_40

[21] Q. Chen, Y. Song, and J. Zhao, "Short-term traffic flow prediction based on improved wavelet neural network," *Neural Comput. Appl.*, vol. 33, pp. 8181–8190, 2021.

[22] J. D. Gadze, A. A. Bamfo-Asante, J. O. Agyemang, H. Nunoo-Mensah, and K. A. B. Opare, "An Investigation into the Application of Deep Learning in the Detection and Mitigation of DDoS Attack on SDN Controllers," *Technologies*, vol. 9, p. 14, 2021.

[23] Y. Zhang and D. R. Xin, "Short-term traffic flow prediction model based on deep learning regression algorithm," *Int. J. Comput. Sci. Math.*, vol. 14, pp. 155–166, 2021.

[24] A. Newell, K. Yang, and J. Deng, "Stacked Hourglass Networks for Human Pose Estimation," in *Lecture Notes in Computer Science*, Cham, Switzerland: Springer, 2016.

[25] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv*, vol. abs/1704.04861, 2017.

[26] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

[27] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

[28] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," *arXiv*, vol. abs/2107.08430, 2021.

[29] H. Law and J. Deng, "CornerNet: Detecting Objects as Paired Keypoints," *Int. J. Comput. Vis.*, vol. 128, pp. 642–656, 2020.

[30] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint Triplets for Object Detection," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, Piscataway, NJ, USA, 2019.

[31] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully Convolutional One-Stage Object Detection," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, Piscataway, NJ, USA, 2019.

[32] G. Song, Y. Liu, and X. Wang, "Revisiting the Sibling Head in Object Detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Piscataway, NJ, USA, 2020.

[33] Y. Wu, Y. Chen, L. Yuan, Z. Liu, L. Wang, H. Li, and Y. Fu, "Rethinking Classification and Localization for Object Detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Piscataway, NJ, USA, 2020.

[34] H. Zhang, Y. Wang, F. Dayoub, and N. Sünderhauf, "VarifocalNet: An IoU-Aware Dense Object Detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021.

[35] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. D. Reid, and S. Savarese, "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019.

[36] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection," in

Proc. Adv. Neural Inf. Process. Syst. 33 (NeurIPS 2020), 2020.

[37] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Proc. Comput. Vis.-ECCV 2014: 13th Eu*

[38] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition. Miami, FL, USA.

[39] Electronics and Telecommunications Research Institute (ETRI). (2019). ETRI Traffic Light Dataset. Available online: https://nanum.etri.re.kr/share/kimjy/etri_traffic_light?lang=ko_KR (accessed on 30 January 2024.)