Journal of Network and Information Security
12 (2) 2024, 07-23
http://www.publishingindia.com/jnis

Image Segmentation and Object Detection for Automobile using OpenCV and CNN

Precious Ochofie Adaji[1] and Jesse Ismaila Mazadu[2*]
[1]Computer Science Department, Faculty of Computing and Information Systems, Federal University Wukari, Nigeria. Email: adajiprecious@gmail.com
[2]Computer Science Department, Faculty of Computing and Information Systems, Federal University Wukari, Nigeria. Email: jesse@fuwukari.edu.ng
[*]Corresponding Author

Abstract: Image segmentation and object detection using CNN (Convolutional Neural Network) and OpenCV (Open-Source Computer Vision) is a popular research area in the field of computer vision and autonomous driving. This method employs deep learning techniques and image processing algorithms to detect and track objects in real-time from a video stream captured by a camera mounted on a vehicle. The main aim of this project is to develop an accurate and robust object detection system that can detect various types of objects such as vehicles, pedestrians, and bicycles on the road. The proposed system uses a pre-trained CNN model to detect objects and OpenCV for further image processing and filtering. The system is evaluated on a publicly available dataset and achieves high accuracy and detection rates for various objects. The results of this study show the potential of using deep learning and image processing algorithms for real-time object detection in autonomous vehicles and traffic control systems. This study examines the use of Convolutional Neural Network techniques that have been used for image segmentation and object detection in road traffic. The study explored the use of Gaussian filters in image pre-processing. The study also trained the model to detect road traffic objects and return output/ feedback. The experimental result of the model was an accuracy of 96% across 26 classes and a recall of 92%. The study, therefore, recommends the use of object detection models in road traffic systems and autonomous vehicles.

Keywords: Artificial Intelligence (AI), Computer vision, Image segmentation, Object detection.

I. Introduction

The recent advancement in artificial intelligence and machine learning has contributed to the growth of computer vision and image recognition concepts (a concept targeted at understanding visual content) while empowering its popularity and continuous application of autonomous unmanned aerial vehicles [1]. Hence, artificial intelligence is one of the most fascinating and controversial technologies in today's era. Artificial intelligence technology in specific computer vision has been applied in more and more industries and fields, such as unmanned driving to bring changes to the transportation industry, the use of identification algorithms to help police arrest suspects, and intelligent robots to solve the problem of resource allocation in the medical industry. In the process of artificial intelligence and visual interface/ object detection, the optimization of automatic recognition algorithm technology is always the core and key link. The so-called automatic recognition algorithm optimization refers to the realization that the new algorithm can be added to the algorithm framework of artificial intelligence timely through the system optimization of the problem in concern [2].

Despite its evolution, artificial intelligence seems to struggle when it comes to rendering images making machine learning, image detection, recognition, and classification a hot topic in the world of technology [3]. Image detection or object detection is a computer technology that processes an image or a video and detects it, whereas, image classification classifies the object when dealing with classification problems [4]. The activity of identifying a specific object among others can be simple for a human brain. However, computers have obvious challenges with this seemingly easy task [3]. These challenges have prompted engineers around the globe in finding the best approach to train machine learning and deep learning model to detect objects in pictures or video. This is no small task for developers as to train a model to detect certain objects one has to show these objects first to the model to learn. In other words, a computer engineer has to feed the model with labeled data images containing the targeted objects, item coordinates, location, and class labels. A more challenging question to this approach becomes how many images are needed to better train a model while taking into cognizance noise in the images, blurriness, low resolution, and small target sizes.

An approach proposed by [3] to better train a model is to choose images with different locations of the target objects so that the items change their coordinates and size during machine learning helping the model to better learn and understand even if the objects are located in different places on the images either be it small or big. Hence, it can be seen that this process can be time-consuming and requires lots of resources and effort. But a promising aspect of artificial intelligence is the progress in GPUs (Graphic processing units) making deep learning much faster and easier. GPU is an electronic circuit that allows the manipulation of computer memory with a target to accelerate graphic processing [5].

Traditionally hand-tuned features were used for object recognition and detection by [2]. With the breakthrough of deep learning using Convolutional Neural Networks, there was a striking performance increase in dealing with these computer vision tasks. The key idea is to learn object models and features from the raw pixel data of the image. Therefore, in an attempt to complement computer vision tasks while conducting image segmentation and object detection for cars, from video footage with the motive of notifying the presence of objects ahead. This study proposed the application of the Convolutional Neural Network in the classification of extracted objects. In the real world, entities within an image are aligned in a different direction and when fed to a model, the model predicate inaccurately due to its inability to understand the alignment of the object in an image, making it challenging to recognize objects. Furthermore, the variation in the size of an object makes the classification of the object challenging. In some scenarios, some of the objects are deformed or distorted making it more complex for the model to effectively learn. Hence, to provide a feasible solution to the stated problem, this study proposed the adaptation of the Convolutional Neural Network in the classification of extracted objects for automobile cars.

This study aims to segment images and detect objects in an automobile environment. Hence, the specific objectives are to:

- Perform image preprocessing using Gaussian filters.
- Apply the Convolutional Neural Network in developing the proposed model and receive feedback about the detected object.
- Evaluate the performance of the model.

II. Literature Review

A. Theoretical Review

The field of image processing is continually evolving. During the past five years, there has been a significant increase in the level of interest in image morphology, neural networks, full-color image processing, image data compression, image recognition, and knowledge-based image analysis systems [6]. Before 2000, several methods were used in digital image processing which includes: threshold segmentation, region segmentation, edge segmentation, texture features, clustering, and a lot more. From 2000 to 2010, there are four main methods: graph theory,

clustering, classification, and a combination of clustering and classification [7].

Since 2010, the rise of neural network models and the development of deep learning mainly involves several models. By 2010, CNN became a very efficient visual processing tool because it can learn hierarchical features [7]. Researchers replace the full join layer with the convolution layer to output a spatial domain mapping (deconvolution) rather than a simple output class probability, thus transforming the image segmentation problem into an end-to-end image processing problem.

B. Image Segmentation

Image segmentation is a method of dividing a digital image into subgroups called image segments, reducing the complexity of the image and enabling further processing or analysis of each image segment. Technically, segmentation is the assignment of labels to pixels to identify objects, people, or other important elements in the image. The result of image segmentation is a set of segments that collectively cover the entire image or a set of contours extracted from the image. Each of the pixels in a region is similar concerning some characteristic or computed property by [8] such as color, intensity, or texture. Adjacent regions are significantly different in color for the same characteristic(s) [9]. When applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like marching cubes [10].

C. How Image Recognition Technology Works

Facebook can now perform face recognition at 98% accuracy which is comparable to the ability of humans. Facebook can identify a friend's face with only a few tagged pictures. The efficacy of this technology depends on the ability to classify images. Classification is pattern matching with data. Images are data in the form of 2-dimensional matrices. Image recognition is classifying data into one category out of many. One common and important example is optical character recognition (OCR). OCR converts images of typed or handwritten text into machine-encoded text. The major steps in the image recognition process are gathering and organizing data, building a predictive model, and using it to recognize images.

Considering that Image Segmentation, Detection, Recognition, and Classification technologies are only in their early stages, it can be expected that great things will be happening shortly. Imagine a world where computers can process visual content better than humans. How easy lives would be when AI could find rooms, cars, machine keys, and solutions for us and the user would not need to spend precious minutes on a distressing search.

D. Review of Related Literature

[11] proposed two methods for vehicle detection using color and texture features. They used L U V colour space and the Dual-Tree Complex Wavelet Transform (DTCWT) for texture and background modelling in the first method. The other method consists of a change detection process that combines the variations in intensities and texture information among the current frame and formerly reconstructed background. They have also used L U V colour space and difference of texture measures which depends on the relation between gradient vectors. To estimate the background image, [12] used an autoregression algorithm. Due to this, it is possible to directly separate dynamic images from the foregrounds by frame difference. They have used Fast Wavelet Transform (FWT) algorithm to extract the features from the detected dynamic regions and the Grey Level Co-Occurrence Matrix (GLCM) to calculate and evaluate the extracted textures for segmenting of images based on texture is proposed by [11] using co-occurrence matrices.

[13] developed a system for the detection and classification of moving vehicles termed Improved Spatio-Temporal Sample Consensus. Firstly, the moving vehicles are identified using the Spatio Temporal Sample Consensus algorithm, from the intrusion of brightness variation and the shadow of the vehicle. Furthermore, utilizing feature fusion techniques the objects are classified according to the area, face, number plate, and vehicle symmetry features. [14] considered research on insect pest

image detection and recognition based on bio-inspired methods. The study investigated a collection of ten categories of insect pests (mainly affecting tea plants). To extract the invariant features for representing the pest appearance, the authors extended the bio-inspired Hierarchical Model and X (HMAX) model in the following ways. Scale Invariant Feature Transform (SIFT) was integrated into the HMAX model to increase the invariance to rotational changes. Meanwhile, Non-negative Sparse Coding (NNSC) was used to simulate the simple cell responses. Moreover, invariant texture features were extracted based on the Local Configuration Pattern (LCP) algorithm. Finally, the extracted features were fed to Support Vector Machines (SVM) for recognition. The experimental results demonstrated that the proposed method had an advantage over the compared methods: HMAX, Sparse Coding, and Natural Input Memory with Bayesian Likelihood Estimation (NIMBLE), and was comparable to the Deep Convolutional Network. The proposed method has achieved a good result with a recognition rate of 85.5% and could effectively recognize insect pests under complex environments and also provides a new idea and method for rapid detection and recognition of insect pests.

[15] researched semantic regions segmentation using a spatio-temporal model from an UAV image sequence with an optimal configuration for data acquisition. The study proposed a hierarchical region-based approach to joint object detection and image segmentation. Their approach simultaneously reasons about pixels, regions, and objects in a coherent probabilistic model. Pixel appearance features allow us to perform well on classifying amorphous background classes, while the explicit representation of regions facilitates the computation of more sophisticated features necessary for object detection. Importantly, the adopted model gives a single unified description of the scene and explains *every* pixel in the image, and enforces global consistency between all random variables in our model. Experiments were run on the challenging Street Scene dataset and the results show a significant improvement over state-of-the-art results for object detection accuracy. The multi-class image segmentation component of the model achieves an overall pixel-level accuracy of 84.2% across the eight semantic classes compared to 83.0% for the pixel-based baseline method described in the works of [16].

[17] presented a Convolutional Neural Network (CNN) for image detection and recognition. The study considers Convolutional Neural Network models that are implemented for image recognition on the MNIST dataset and object detection on the CIFAR-10 dataset. The model was trained on only a CPU unit and real-time data augmentation was used on the CIFAR-10 dataset. Along with that, Dropout was used to reduce Overfitting on the datasets. The algorithm was implemented on MNIST and CIFAR-10 datasets and its performance is evaluated. The experimental result shows that the accuracy of models on MNIST was 99.6%, and CIFAR-10 was using real-time data augmentation and dropout on CPU units. The accuracy of MNIST was good but the accuracy of CIFAR-10 can be improved by training with larger epochs and on a GPU unit. The calculated accuracy on MNIST is 99.6% and on CIFAR-10 is 80.17%.

[18] introduced a system that automatically calculated the number of vehicles, classifies vehicles by type, measure the speed, and decides lane usage. To achieve this, they have implemented and compared two systems: one system comprising background modeling using a Mixture of Gaussians (MoG) for foreground detection and an SVM classifier for the classification of vehicles and another system based on the Faster RCNN. However, they reported that the Faster RCNN algorithm performs better in dynamic traffic scenes. [19] proposed an optimized Convolutional Neural Network architecture based on deep learning algorithms for vehicle detection and classification system used for intelligent transportation applications. PVANET by [20] as the base network, is selected and improved by fine-tuning to get better accuracy. It consists of eight Concatenated ReLU convolution layers and eight inception layers for the base network. The hypernet architecture is used to associate dissimilar features, thereby making it better to achieve the desired bounding boxes for the Region Proposal Net layer. [21] developed an Adaptive Neuro Fuzzy Inference

System classifier for the classification of moving vehicles on the roads. It includes six main phases pre-processing, feature extraction, detection, structural matching, tracking, and classification of vehicles. Background subtraction and the Otsu threshold algorithm are used for vehicular detection. The characteristics of the vehicles detected are obtained by the log Gabor filter and Harris corner detector, which are used to classify the vehicles.

[22] conferred a segment before detect approach using deep learning techniques. Segmentation and followed by detection and classification of multiple wheeled vehicle variants are tested for high-resolution remote sensing pictures. The detection and classification of vehicles depending on a virtual detection zone were suggested by [23] which comprises foreground extraction, detection, feature extraction, and classification. The GMM is used in the detection of vehicles and also some operations are performed to get the foreground objects and classification is done, using the k-nearest neighbor classifier. [24] recommended a semi-supervised Convolutional Neural Network technique for vehicle classification based on the front view of the vehicle. Yet, the features trained by CNN are too biased to work in raster images. [25] presented a 3D fully convolutional network for vehicle detection in the point cloud. The research aimed to design a 3D Fully Convolutional Network (FCN) to detect and localize objects as 3D boxes from point cloud data. The proposed approach extends FCN to 3D and was applied to 3D vehicle detection for an autonomous driving system, using a Velodyne 64E lidar. Meanwhile, the approach can be generalized to other object detection tasks on point clouds captured by Kinect, stereo, or a monocular structure from motion. The KITTI training dataset was employed which contains 7500+ frames of data, of which 6000 frames are randomly selected for training in the experiments. The rest 1500 frames are used for offline validation, which evaluates the detection bounding box by its overlap with ground truth on the image plane and the ground plane. The experiment results mainly focus on the detection of the *Car* category for simplicity. Regions within the 3D center sphere of a *Car* are labeled as positive samples, i.e. in *Van* and *Truck* are labeled to be ignored. *Pedestrians*, *Bicycles,* and the rest of the environment are labeled as negative backgrounds, i.e. $P - V$. The performance improvement of the adopted method was significant compared to previous point cloud-based detection approaches.

[26] worked on vehicle detection and tracking based on corner and lines adjacent detection features. The study used the corner detection process and the line adjacent detection features by creating black-and-white images through thresholding. The corners in the image are vibrant and strong in black-and-white mode with the use of the thresholding function. The threshold value is set to 30 for the linking threshold and for upper thresholding they have to consider up to 50. Then after thresholding, the result is assigned to the Kanade-Lucas-Tomasi algorithm for the detection of corner spots and masked with the original image to observe the angle of detection [27]. Later the XY positions of individual corner spots are recorded and each corner spot is connected with straight lines. A filtering process is done to remove unnecessary lines by limiting the length of the line not exceeding the vehicle's length and leaving alone those lines which fall under the limit.

[28] considered food detection and recognition using a Convolutional Neural Network. The authors proposed a Convolutional Neural Network (CNN) for the tasks of detecting and recognizing food images due to the diversity of types of food. The researchers find out that food recording is usually a manual exercise using textual description, but manual recording is tedious and time-consuming. Therefore, despite the attempts at food item recognition, recognition performance is not yet satisfactory. So that leads to the application of CNN to the tasks of food detection and recognition through parameter optimization. They constructed a dataset of the most frequent food items in a publicly available food-logging system and used it to evaluate recognition performance. The result obtained from the study showed that CNN possesses significantly higher accuracy than the traditional support-vector machine-based methods with handcrafted features. In addition, it was discovered that the convolution kernels show that color dominates the feature extraction process. In regards to food image

detection, CNN also showed significantly higher accuracy than a conventional method.

[29] proposed a unified framework for multi-oriented text detection and recognition. The goal of this study is to build a general system for scene text detection and recognition. Text detection and recognition are accomplished concurrently with the same features and classification scheme. A dictionary search-based method for recognition error correction was also proposed. The proposed system is capable of detecting and recognizing texts of different scales, colors, fonts, and orientations, in diverse real-world scenes. They trained a unique model using the training data. III-G. 600 trees are used for training the component level classifier and 300 trees for the chain level classifier. About 2 M component-level examples and 560 k chain-level examples are bootstrapped from the training data for learning this model. They later applied a unique model with the same parameter setting to all the datasets which include ICDAR 2011, and MSRA-TD500. To further assess the detection functionality of the proposed system, the MSRA-TD500 dataset was applied in which texts in the images may be with different directions, fonts, colors, and scales. The extensive experiments demonstrate that compared to existing methods in the literature the proposed algorithm achieves state-of-the-art or very competitive performance on various challenging benchmarks as well as on a novel database we propose. The proposed algorithm can assist numerous applications that require text information extraction from images or videos, such as video search, target geolocation, and automatic navigation.

[30] considered image-based face detection and recognition: "state of the art". The goal of the study was to evaluate various face detection and recognition methods and provide a complete solution for image-based face detection and recognition with higher accuracy, and better response rate as an initial step for video surveillance. The system evaluates the face detection and recognition methods which are considered to be a benchmark. mFive datasets were used for the experiments which include; Face 94, Face 95, Face 96, Grimace, and Pain Expressions. Solutions were proposed based on performed tests on various face-rich databases in terms of subjects, pose, emotions, race, and light. Some methods performed consistently over different datasets whereas other methods behave very randomly however based on average experimental results performance as it was evaluated. [31] have decided on the presence of vehicles based on the present and previous frame. They used the method of association to define the relation between consecutive frames. This method exploits the displacement of edges in the frames. The Adaboost classifier is used to assess whether an obstacle is a vehicle.

[32] uses regions for object detection instead of the traditional sliding window approach. However, unlike our method, the study uses a single over-segmentation of the image and make the strong assumption that each segment represents a (probabilistically) recognizable object part. Our method, on the other hand, assembles objects (and background regions) using segments from multiple different over-segmentations. The multiple over-segmentations avoid errors made by anyone segmentation. Furthermore, we incorporate background regions which allow us to eliminate large portions of the image thereby reducing the number of component regions that need to be considered for each object.

[33] use a non-parametric approach to image labeling by warping a given image onto a large set of labeled images and then combining the results. This is a very effective approach since it scales easily to a large number of classes. However, the method does not attempt to understand the scene semantics. In particular, the method is unable to break the scene into separate objects (e.g., a row of cars will be parsed as a single region) and cannot capture combinations of classes not present in the training set. As a result, the approach performs poorly on most foreground object classes.

[34] considered image parsing: Unifying segmentation, detection, and recognition. The authors resented a Bayesian framework for parsing images into their constituent visual patterns. Their study introduces a computational framework for passing images into basic visual patterns. The problem was formulated using Bayesian probability theory

and designed a stochastic DDMCMC algorithm to perform inference and give a rigorous way to combine segmentation with object detection and recognition. The study gives proof of concept by implementing a model whose visual patterns include generic regions (texture and shading) and objects (text and faces). Their approach enables these different visual patterns to compete and cooperate to explain the input images. The goal of the adopted algorithm was to construct a parse graph representing the image. The algorithm proceeds by constructing Markov Chain dynamics, implemented by sub-kernels, for different moves to configure the passing graph such as creating or deleting notes. These types of patterns compete and cooperate to explain the image and so image parsing unifiers image segmentation, object detection, and recognition. The algorithm illustrated natural images of complex city scenes and shows examples where image segmentation can be improved by allowing object-specific knowledge to disambiguate low-level segmentation cues, and concisely where object detection can be improved by using generic visual patterns to explain away shadows and occlusions. The result shows that the method can be applied to any other inference problem that can be formulated as probabilistic inference on graphs.

[35] researched the detection and recognition of license plate characters with different appearances. The study proposed an automatic license plate detection and rotation-free character recognition system. In conventional license plate detection and recognition methods, it is difficult to determine the license plate with large pan and tilt angles and is hard to recognize the rotation-free character. Car images are taken from various positions outdoors. Because of the variations of angles from the camera to the *car,* license plates have various locations and rotation angles in an image. In the license plate detection phase, the magnitude of the vertical gradients is used to detect candidate license plate regions. These candidate regions are then evaluated based on three geometrical features: the ratio of width and height, the size, and the orientation. The last feature is defined by the major axis. In the character recognition phase, character features that are non-sensitive to the rotation variations are detected. The various rotated character images of the specific character can be normalized to the same orientation based on the major axis of the character image. The crossing counts and peripheral background area of an input character image are selected as the features for rotation-free character recognition. The authors implemented the proposed system on a Pentium II 300 MHz PC with C++ language under a Windows environment and used Nikon 5700 digital camera as an input device. The experimental results show that the license plates detection method can correctly extract all license plates from *102 car* images taken outdoors and the rotation-free character recognition method achieved an accuracy rate of 98.6%.

E. Knowledge Gap

One potential knowledge gap in the topic of "Road Traffic Object Detection using CNN and OpenCV" is the lack of research on the effectiveness and efficiency of this approach in real-world scenarios with varying weather and lighting conditions. While there have been studies on the use of CNN and OpenCV for road traffic object detection, many of these studies have been conducted under controlled laboratory conditions or with limited variations in environmental factors. Therefore, it would be valuable to conduct research on the performance of this approach in more diverse and challenging real-world conditions, such as fog, rain, and low-light conditions. This could involve collecting and analyzing data from a variety of locations and weather conditions, and comparing the results with other object detection techniques to evaluate the relative strengths and weaknesses of the CNN and OpenCV approach. Such research could provide valuable insights into the practical applications of this approach for real-world road traffic object detection.

III. Research Methodology

For the effective and efficient implementation of any artificial intelligence research project particularly computer vision, it is crucial to devise a research methodology that specifies the procedures or techniques to detect and recognize objects in video, segment and classify these objects to identify the targeted object. Hence, the methodological approach

proposed by this study takes into consideration three phases. The first step encompasses reading the dataset via the open-CV library and extracting objects of concern from the dataset. The second phase encompasses the conductance of image preprocessing on the extracted objects from the sourced video. The image preprocessing techniques adopted by this study are the Low-Pass filters and the High-Pass filters. Both the low-pass and high-pass filter techniques were utilized to filter out high-frequency content such as noise, edges in the extracted object, and also low-intensity edges. The low-pass filter is used in removing noise and blurring images whereas the high-pass filter is used in finding edges. The last stage involves feeding the filtered images to a Convolutional Neural Network and lastly conducting a performance evaluation analysis from the output of The Convolutional Neural Network. The stepwise approach for the implementation of the proposed system is shown in Fig. 1.

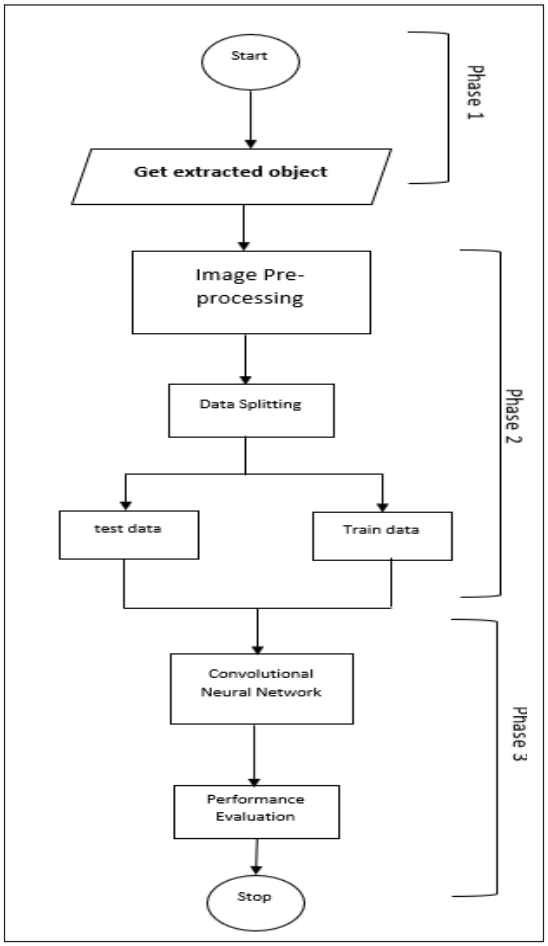

Fig. 1: Research Methodology

A. Experimental Setup

To experiment and analyze video footage on the proposed image segmentation and object detection for automobile cars, this study proposed the utilization of 64-bit Windows OS on an Intel(R) Core (TM) i5-6200U CPU @ 2.30 GHz-2.40 GHz with 8.00 GB of RAM. Furthermore, the proposed implementation utilized the viability of the below packages for its effective implementation:

- Python Software Development Tool (SDK).
- NumPy, Sklearn, Open-CV, and Kera's from TensorFlow a high-level API (Application Programming Interface).
- Jupiter Notebook as the programming environment.

B. Dataset Description

Car_TrafficLight_Pedestrian Dataset for Computer Vision Project. The dataset includes 25951 images. Traffic Objects are annotated in Tensorflow Object Detection format. The following pre-processing was applied to each image: Auto-orientation of pixel data (with EXIF-orientation stripping) and Resize to 416x416 (Stretch).

The following augmentation was applied to create 2 versions of each source image:

- Randomly crop between 0 and 54 percent of the image
- Salt and pepper noise was applied to 10 percent of pixels

Author: Alexandra Lasic

Project Type: Object Detection

Subject: trafficObjects

Classes: biker, car, pedestrian, trafficLight, trafficLight-Green, trafficLight-GreenLeft, trafficLight-Red, trafficLight-RedLeft, trafficLight-Yellow, trafficLight-YellowLeft, truck.

C. Image Preprocessing

Data cleansing is a crucial step in deep learning before feeding a model with input. Hence, the aim

of image preprocessing is to improve the image data that suppress undesired distortions or enhance some of the image features relevant to the analysis task. The concept of image pre-processing doesn't increase image formation content but decreases it if the entropy is an information measure. In essence, the idea behind image pre-processing is to process image data at its lowest level of abstraction. Thus, to filter-out high-frequency content such as noise, edges in the extracted object, and also low-intensity edges, the blurring techniques were applied to the extracted objects using the Low-Pass filters and the high-pass filters. The low-pass filter is used in removing noise and blurring images whereas the high-pass filter was in finding edges. In detail the image preprocessing stages include:

- *Noise Reduction:* Edge detection is susceptible to noise in an image. So, to remove noise this study applied a 5 X 5 Gaussian filter.
- *Finding the Intensity Gradient of the Image:* This phase involves filtering the image in both horizontal and vertical directions.
- *Non-Maximum Suppression:* Involve scanning object to remove unwanted pixels which may not have any edge. This encapsulates checking every pixel to validate if it is a local maximum in its neighborhood in the direction of the gradient.
- *Hysteresis Thresholding:* Deciding which edges are edges and which are not using threshold values called minimum value and maximum value. Then, edges with an intensity gradient more than the maximum are retained whereas those with a value less than the minimum value are discarded.

D. Model Description (Convolutional Neural Network)

Convolutional Neural Network (CNN) is a variation of an artificial neural network that requires a convolutional layer but can have other types of layers, such as nonlinear, pooling, and fully connected layers, to create a deep Convolutional Neural Network. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field and a collection of such fields overlaps to cover the entire visual area [36].

In the proposed application of CNN, convolutional filters are trained using the backpropagation method. The shapes of the filter structure depend on the sample dimensionality. In the case of object detection as in the case of this study, a filter can be used to perform edge extraction, while another filter for image extraction. A suitable determination of filter value is during the model learning phase. Hence, in the proposed convolutional layer, multiple filters slide over the layer for the given input data. A summation of an element-by-element multiplication of the filters and receptive field of the input is then calculated as the output of this layer. The weighted summation is placed as an element of the next layer.

It is important to note that the filter size (receptive field) must be fixed across all filters used in the same convolutional operation. Hence, to control the size of the output feature map, zero padding techniques were applied. Zero padding adds zero rows and columns to the original input matrix to control the size of the output feature map. The aim of applying zero padding is to also include the data at the edge of the input matrix. Without zero padding, the convolution output is smaller in size than the input. Therefore, the network size shrinks by having multiple layers of convolutions, which limits the number of convolutional layers in a network as zero padding prevents the shrinking of networks and provides unlimited deep layers in our network architecture. The first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High-Level features as well, providing a network that has a wholesome understanding of images in the dataset.

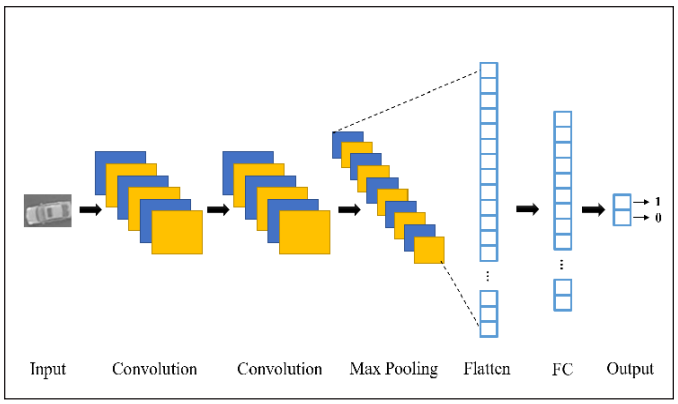

Fig. 2: Convolutional Neural Network

E. Nonlinearity

The major reason for using nonlinearity is to adjust or cut off the generated output. Several nonlinear functions can be utilized in CNN. However, the rectified linear unit (ReLU) is one of the most common nonlinearities applied in various fields, such as image processing. The ReLU can be represented as:

ReLU={0,x, if x<0, if x ≥ 0}.

F. Pooling Layer

The Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training the model. Hence, to reduce the dimensionality of the inputs while reducing the computational power required by the proposed Convolutional Neural Network model, the pooling layer is proposed. The survey conducted revealed various types of pooling methods such as averaging and summation, etc. but the max pooling best represents the maximum value inside the pooling filter as the output. The max pooling based on the pieces of the literature surveyed provides significant results by down sampling input size by 75%. Furthermore, max pooling performs a noise suppressant as it discards noisy activation and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise-suppressing mechanism [37].

G. SoftMax Layer

SoftMax layer is considered an excellent method to demonstrate categorical distribution. The SoftMax function, which is mostly used in the output layer, is a normalized exponent of the output values. This function is differentiable and represents a certain probability of the output. Moreover, the exponential element increases the maximum value probability. The SoftMax equation is given as follows:

$$o_i = \frac{e^{z_i}}{\sum_{i=1}^{M} e^{z_i}} \quad (1)$$

Where, o_i is the softmax output number i, z_i is the output i before the softmax, and M is the total number of output nodes.

H. Justification for Convolutional Neural Network

A ConvNet can successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and the reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

Furthermore, considering that the objective of the Convolution Operation is to extract the high-level features such as edges, objects, etc, from the input image which suits this problem of detecting objects for a car.

I. Performance Evaluation Metrics

To evaluate the performance of the model. This study utilized the precision, and recall, accuracy evaluation parameters.

Precision measures the classifier's accuracy. It is the percentage of the number of correctly predicted positive reviews divided by the total number of predicted positive reviews:

$$precision = \frac{TP}{TP + FP} \quad (2)$$

Accuracy is the most important metric of a model performance evaluation. Accuracy is measured as a percentage of the number of correctly predicted reviews to the total number of reviews present in the dataset. Thus, the accuracy calculates the ratio of inputs in the test set correctly labeled by the classifier:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

Loss Function

In most deep learning tasks, the evaluation of loss function is essential in determining the accuracy of a model. Therefore, the loss function utilized by this study is the binary cross-entropy which in other terms is referred to as the log loss function. The Binary Cross Entropy in operation compares the predicted probabilities of the adopted model to the actual class output which can be either 0 or 1 corresponding to a detected and non-detected object image.

$$Log\ loss = \frac{1}{N}\sum_{i=1}^{N} -(y_i * log(p_i) + (1 - y_i) * log(1 - p_i)) \quad (4)$$

Here, pi is the probability of class 1 (detected object), and (1-pi) is the probability of class 0 which corresponds to an undetected image.

IV. Result and Discussions

A. Experimental Setup

To experiment and analyze the traffic video footage for the proposed image segmentation and object detection for automobile cars. A 64-bit Windows Operating System, with an Intel(R) Corel (TM) i5-3630QM CPU @ 2.40 GHZ with 4.00 GB of RAM was used. The programming environment utilized for implementing the program code was the Anaconda environment using the Python 3.8 software development kit. The application programming interface utilized was Kera's TensorFlow API and some other Python dependencies such as the NumPy for vector operations, pandas for reading files, and the Python Open-CV library for computer vision operation.

B. Parameter Setting

The meta-parameters of the proposed CNN architecture are presented in Table I. The batch size defines the number of the dataset input instance passed to the model per unit layer. Max epochs define the maximum number of epochs conducted which was set to 20 with a learning approach of 0.0001, with Adam set as the optimizer and drop out of 0.25 for each of the layers. An epoch refers to a complete iteration through the entire training dataset during the training of a neural network. The learning rate is a hyperparameter that determines the step size at which the model updates its parameters during training. In other words, it controls the rate at which the model learns from the data. The optimizer is responsible for adjusting the weights and biases in the model to improve its accuracy and performance and the optimizer utilized in this study is Adam. Furthermore, the activation function utilized was the SoftMax activation considering that the dataset categorization was based on forty-three classes. The meta-parameters of our CNN architecture are presented in Table I.

Table I: Parameter Setting

Parameter	Value
Batch Size	64
Max epochs	20
Initial learn rate	0.001
Optimizer	SoftMax
Momentum	0.01

C. Data Visualization

Considering that the dataset consists of images, the dataset was initially backed with a CSV file to specify the attribute of each respective image file. Fig. 4 shows the width, height, label, and path information of each image file.

	Width	Height	label	path
0	53	54	16	Test/00000.png
1	42	45	1	Test/00001.png
2	48	52	38	Test/00002.png
3	27	29	33	Test/00003.png
4	60	57	11	Test/00004.png

Fig. 3: Dataset Attribute Information

D. Model Architectural Discussion

The approach utilized by this study to detect and classify objects for automated cars uses CNN to classify objects in a traffic environment. The main challenge was finding an optimal architecture for the CNN. Therefore, the study first defined the input and output structures of the network before presenting the optimal architecture based on the results of various experiments. Each recorded sample is a $128 \times 128 \times 3$ matrix. However, the frame length (N) is variable for each sample because each subject uniquely carries out object detection and classification. For an effective implementation, the input size should be fixed. One process assumes the input size according to the sample with the maximum frame length and utilizes zero padding (at the end of recorded samples) for those with short frames. However, this process provides CNN with a huge input size (i.e., $512 \times 8 \times 8$) and is computationally expensive. The study cascaded the convolutional layers together to build the classifier. Each convolutional layer consists of convolution, nonlinearity, and pooling. Furthermore, for each convolutional layer, a non-linearity is added to the output. Typically, this was done by the Rectified Linear Unit (Relu) Activation function. In total, three convolutional layers with one fully connected layer. The fully connected layer has a SoftMax Activation function at the end which ensures that the sum of output probabilities from the Fully Connected Layer is 1. The SoftMax function with 4 classes is the output shape of the developed CNN method. Although the study employed the peak value in the output node as the calculated class the success of the study relied on the SoftMax values to consider other highly probable hypotheses.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	2432
conv2d_1 (Conv2D)	(None, 22, 22, 32)	25632
max_pooling2d (MaxPooling2D)	(None, 11, 11, 32)	0
dropout (Dropout)	(None, 11, 11, 32)	0
conv2d_2 (Conv2D)	(None, 9, 9, 64)	18496
conv2d_3 (Conv2D)	(None, 7, 7, 64)	36928
max_pooling2d_1 (MaxPooling2	(None, 3, 3, 64)	0
dropout_1 (Dropout)	(None, 3, 3, 64)	0
flatten (Flatten)	(None, 576)	0
dense (Dense)	(None, 256)	147712
dropout_2 (Dropout)	(None, 256)	0

...
Total params: 242,251
Trainable params: 242,251

Fig. 4: Model Architecture

E. Data Scaling

Normalization is an essential aspect of the machine learning task after all necessary data purification. Thus, it is essential to scale the dataset into a feature range, enhancing the model's capabilities to detect and classifies object in an automobile driving environment. To ensure feature scaling this study optimized the normalizer function from the TensorFlow API library as shown in Fig. 5. The essence was to bind the data records to a feature range between 0 and 1 and hence empower the model with great magnitude distance control among the datasets and thus enable it to converge faster with good results.

```
#Converting the labels into one hot encoding
y_train = to_categorical(y_train, 43)
y_test = to_categorical(y_test, 43)
```

Fig. 5: Data Scaling

F. Percentage Split Technique

In training the Convolutional Neural Network, this study divided the dataset into some training and test proportions, where the training set was used to train the models and the test set to validate the workings of the model. The train-to-split proportion utilized for this experiment was a 70:30 train-to-

test proportion, where 70% of the dataset was used to train the model and 30% was used to validate the learning accuracy of the Convolutional Neural Network.

G. Result Presentation

Considering the proposed evaluation metrics, accuracy, recall, log-loss, and precision. For accuracy and precision, the higher the values are, the better the classifier performs. It is important to note that precision and recall are exploited to overcome the shortcoming of Accuracy. To be clearer, the precision reflects how many samples that are exactly positive among samples indicated as positive, whilst recall highlights the ratio of samples predicted as positive inside definite positive samples. Hence, the statistical result of the above-discussed Convolutional Neural Network algorithm is shown in Table 4. Hence, prior to the result discussion, the model was compiled using 20 epochs, and 20 iterations for a time period of 1537 ms with a batch size of 64 and a learning rate of 0.002. The result of the Convolutional Neural Network from the compilation parameters in Table II achieves an accuracy score of 96% as shown in Table III, with a precision score of 95%, a recall of 92%, and a log-loss of 0.068%. It is important to note that the lower the log-loss evaluation metric the better the model performance.

Table II: Model Compilation

Epoch	Iteration	Time Elapsed	Batch	Learning Rate
20	20	1537	64	0.002

Table III: CNN Result

Metrics	Scores
Accuracy	0.96
Precision	0.95
Recall	0.92
Log-Loss	0.068

H. Results

Fig. 6

Fig. 7

Fig. 8

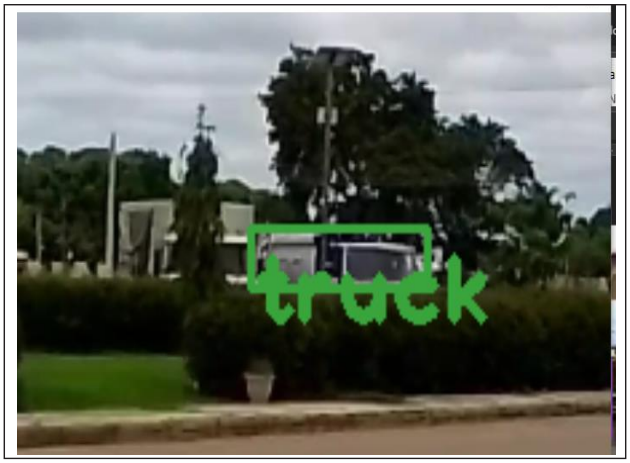

Fig. 9

V. Summary and Conclusion

A. Summary

The study has investigated the performance of the Convolutional Neural Network on an automated traffic dataset sourced from the Kaggle machine learning repository. The architecture of the Convolutional Neural Network integrates three convolutional layers with one fully connected layer bounded by a SoftMax activation function. To enhance the performance of the CNN model, a three-phase methodological approach was utilized. The first step encompasses reading the dataset via the openCV library and extracting objects of concern from the dataset. The second phase encompasses the conductance of image preprocessing on the extracted objects from the sourced video using the Low-Pass and High-Pass filters to filter-out high-frequency content such as noise, edges in the extracted object, and also low-intensity edges. The last stage involves feeding the filtered images to a Convolutional Neural Network and lastly conducting a performance evaluation analysis from the output of the Convolutional Neural Network after being fed with training data of 70%, with 30% used for testing. To facilitate the conductance of this analysis, it is worth notable that the implementation of the models was carried out using Python programming language and other third-party libraries such as NumPy, Keras TensorFlow, Python Open-CV, Matplotlib, Pandas, and sklearn.

B. Conclusion

In this study, a system that classifies extracted objects for automobile cars identification of traffic control and human passage. The CNN is presented, which is considered a good feature extractor algorithm. The automobile traffic detection dataset sourced from the Kaggle machine learning repository was used to train the CNN for various classes. The Convolutional Neural Network (CNN) was able to achieve a score of 98% which was validated using some evaluation metrics such as the recall, precision, and log-loss function. The precision score achieved was a 100%, with a recall of 98%, and a log-loss of 0.068%.

VI. Recommendation

The recommendation module is structured to proffer the possible future application areas for the detection and classification of objects for automobiles and likewise, a future research advancement area to enhance the performance of the utilized models in the detection and classification of objects in an automated mobile environment.

A. Application Areas

- The key application area of this implementation is in the automobile industry with self-driving cars. Hence, the implementation can be utilized in self-driving cars to visualize objects and classify them into their respective classes while enhancing and transforming the transportation industry into a safe and automated driving paradigm.
- The methodology implemented by this study can be optimized to equip drones for surveillance so as to detect unusual activities around the borderland while preventing infiltration in keeping a nation safe.
- The computation concept and algorithm of the designed model implementation can be utilized in various institutions to expose students to the

coherent complexities of computational models and their respective efficiency and effectiveness.

B. Suggestions for Further Research

After an extensive study and review of several kinds of literature relating to the detection and classification of objects, this study suggests the future research areas as follows:

- The application of Res-Net and ImageNet neural networks considering that these algorithms over the past decades have proven successful from the literature reviewed.
- The adaptation of the utilized convolutional neural network architecture for the identification and classification of objects for drones.

References

[1] I. Mademlis, N. Nikolaidis, A. Tefas, I. Pitas, T. Wagner, and A. Messina, "Autonomous unmanned aerial vehicles filming in dynamic unstructured outdoor environments [applications corner]," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 147-153, 2018.

[2] Y. Xiao, Z. Tian, J. Yu, Y. Zhang, S. Liu, S. Du, and X. Lan, "A review of object detection based on deep learning," *Multimedia Tools and Applications*, vol. 79, no. 33, pp. 23729-23791, 2020.

[3] Azati, "Image detection recognition and classification with machine learning," 2022. [Online]. Available: https://azati.ai/image-detection-recognition-and-classification-with-machine-learning/

[4] L. Liu, O. Wanli, W. Xiaogang, F. Paul, C. Jie, L. Xinwang, and P. Matti, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261-318, 2020.

[5] T. D. Ngo, T. T. Bui, T. M. Pham, H. T. Thai, G. L. Nguyen, and T. N. Nguyen, "Image deconvolution for an optical small satellite with deep learning and real-time GPU acceleration," *Journal of Real-Time Image Processing*, vol. 18, no. 5, pp. 1697-1710, 2021.

[6] H. Lei, T. Lei, and T. Yuenian, "Sports image detection based on particle swarm optimization algorithm," *Microprocessors and Microsystems*, vol. 80, p. 103345, 2021.

[7] S. Srivastava, A. V. Divekar, C. Anilkumar, I. Naik, V. Kulkarni, and V. Pattabiraman, "Comparative analysis of deep learning image detection algorithms," *Journal of Big Data*, vol. 8, no. 1, pp. 1-27, 2021.

[8] S. Hore, S. Chakraborty, S. Chatterjee, N. Dey, A. S. Ashour, L. Van Chung, and D. N. Le, "An integrated interactive technique for image segmentation using stack-based seeded region growing and thresholding," *International Journal of Electrical & Computer Engineering,* vol. 6, no. 6, pp. 2088-8708, 2016.

[9] J. Shi, Q. Yan, L. Xu, and J. Jia, "Hierarchical image saliency detection on extended CSSD," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 717-729, 2015.

[10] D. Mun, and B. C. Kim, "Three-dimensional solid reconstruction of a human bone from CT images using interpolation with triangular Bézier patches," *Journal of Mechanical Science and Technology*, vol. 31, no. 8, pp. 3875-3886, 2017.

[11] K. V. Keerthi, P. Parida, and D. Sonali, "Vehicle detection and classification: A review," *Journal of Information Assurance and Security,* 2020. [Online]. Available: www.mirlabs.net/jias/index.html

[12] P. Lin, X. Jianmin, and B. Jianyong, "Robust vehicle detection in vision systems based on fast wavelet transform and texture analysis," in *Proceedings of the IEEE International Conference on Automation and Logistics*, IEEE, 2007, pp. 2958-2963.

[13] Y. Wang, B. Xiaojuan, W. Huan, W. Di, W. Hao, Y. Shouqing, L. Sinuo, and L. Jinhui, "Detection and classification of moving vehicle from video using multiple spatio-

temporal features," *IEEE Access,* vol. 7, pp. 80287-80299, 2019.

[14] L. Deng, Y. Wang, Z. Han, and R. Yu, "Research on insect pest image detection and recognition based on bio-inspired methods," *Biosystems Engineering*, vol. 169, pp. 139-148, 2018.

[15] H. Vu, T. L. Le, V. G. Nguyen, and T. H. Dinh, "Semantic region segmentation using a spatio-temporal model from a UAV image sequence with an optimal configuration for data acquisition," *Journal of Information and Telecommunication*, vol. 2, no. 2, pp. 126-146, 2018.

[16] G. Stephen, F. Rick, and K. Daphne, "Decomposing a scene into geometric and semantically consistent regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 2, pp. 517-529, 2009.

[17] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, "Convolutional neural network (CNN) for image detection and recognition," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 2018, pp. 278-282.

[18] A. Arinaldi, A. P. Jaka, and A. G. Arlan, "Detection and classification of vehicles for traffic video analytics," *Procedia Computer Science*, vol. 144, pp. 259-268, 2018.

[19] C. Tsai, T. Ching-Kan, T. Ho-Chia, and G. Jiun-In, "Vehicle detection and classification based on deep neural network for intelligent transportation applications," in Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPAASC), *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 1605-1608, 2018.

[20] K. H. Kim, H. Sanghoon, R. Byungseok, C. Yeongjae, and P. Minje, "Deep but lightweight neural networks for real-time object detection," 2016. arXiv preprint arXiv:1608.08021.

[21] V. Murugan, and V. R. Vijaykumar, "Automatic moving vehicle detection and classification based on artificial neural fuzzy inference system," *Wireless Personal Communications,* vol. 100, no. 3, pp. 745-766, 2018.

[22] N. Audebert, L. S. Bertrand, and L. Sébastien, "Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images," *Remote Sensing,* vol. 9, no. 4, p. 368, 2017.

[23] N. Seenouvong, W. Ukrit, N. Chaiwat, K. Khamphong, and O. Noboru, "Vehicle detection and classification system based on virtual detection zone," *in Proceedings of the International Joint Conference on Computer Science and Software Engineering (JCSSE),* IEEE, 2016, pp. 1-5.

[24] Z. Dong, W. Yuwei, P. Mingtao, and J. Yunde, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems,* vol. 16, no. 4, pp. 2247-2256, 2015.

[25] B. Li, "3D fully convolutional network for vehicle detection in the point cloud," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1513-1518.

[26] M. D. Munajat, Enjat, D. H. Widyantoro, and R. Munir, "Vehicle detection and tracking based on corner and lines adjacent detection features," in *Proceedings of the International Conference on Science in Information Technology (ICSITech)*, IEEE, 2016, pp. 244-249.

[27] L. Miyashita, Y. Watanabe, and M. Ishikawa, "Midas projection: Markerless and modelless dynamic projection mapping for material representation," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1-12, 2018.

[28] H. Kagaya, K. Aizawa, and M. Ogawa, "Food detection and recognition using convolutional neural network," in *Proceedings of the 22nd ACM International Conference on Multimedia*, 2014, pp. 1085-1088.

[29] C. Yao, X. Bai, and W. Liu, "A unified framework for multi-oriented text detection

and recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4737-4749, 2014.

[30] F. Ahmad, A. Najam, and Z. Ahmed, "Image-based face detection and recognition: State of the art," 2013. arXiv preprint arXiv:1302.6379.

[31] Z. Khalid, M. Abdenbi, and E. A. Mohamed, "A new vehicle detection method," *International Journal of Advanced Computer Science and Applications*, vol. 1, no. 3, 2011.

[32] S. Gould, T. Gao, and D. Koller, "Region-based segmentation and object detection," *Advances in Neural Information Processing Systems*, vol. 22, 2009.

[33] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing: Label transfer via dense scene alignment," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, vol. 2, no. 3, pp. 45-47, 2009.

[34] Z. Tu, X. Chen, A. L. Yuille, and S. C. Zhu, "Image parsing: Unifying segmentation, detection, and recognition," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 113-140, 2005.

[35] S. Z. Wang, and H. J. Lee, "Detection and recognition of license plate characters with different appearances," in *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, vol. 2, pp. 979-984, 2003.

[36] A. Jain, and A. Bhardwaj, "Convolutional neural networks: A comprehensive survey," *IEEE Access,* vol. 9, pp. 31552-31577, 2021, doi: https://doi.org/10.1109/ACCESS.2021.3055801.

[37] K. K. Singh, P. Mishra, and K. Singh, "A comprehensive review on convolutional neural network," in *Proceedings of the 3rd International Conference on Intelligent Computing and Control Systems (ICICCS 2018),* IEEE, 2018, pp. 722-728, doi: https://doi.org/10.1109/ICCONS.2018.8663213.