Received: 8 October 2023 | Revised: 1 January 2024 | Accepted: 8 January 2024
DOI: 10.37917/ijeee.21.2.18 Early View | December 2025

Open Access

Iraqi Journal for Electrical and Electronic Engineering
Original Article

IJEEE University of Basrah College of Engineering

A Novel Deep Learning Object Detection Based on PCA Features for Self Driving Cars

Namareq Odey*[1], Ali Marhoon[2]
[1]Department of Electrical Engineering, University of Basrah, Basrah, Iraq
[2]Department of Cyber Security Engineering, Almaaqal University, Basrah, Iraq

Correspondance
*Namareq Odey
Department of Electrical Engineering, University of Basrah,
Basrah, Iraq
Email: namareqalmotter22@gmail.com

Abstract
In recent years, self-driving cars and reducing the number of accident casualties have drawn a lot of attention. Although it is crucial to increase driver awareness on the road, autonomous vehicles can emulate human driving and guarantee improved levels of road safety. Artificial intelligence (AI) technologies are often employed for this purpose. However, deep learning, a subset of AI, is prone to numerous errors, a wide range of threats, and needs to handle vast amounts of data, which imposes high-performance hardware requirements. This study suggests a deep learning model for object recognition that employs characteristics to describe data rather than images. Our model employs the COCO dataset as the training foundation, and it was suggested that the features be retrieved using the principal component analysis PCA extraction method. The current results demonstrate the efficacy and precision of our model, with an accuracy of 99.96 %. Furthermore, the performance indices, i.e., recall, precision, and F1-score, achieved about 1 for most of the COCO classes in training phase and promising results in testing phase.

Keywords
Artificial Intelligence, COCO Dataset, Deep Learning, Self Driving Cars.

I. Introduction

Intelligent systems for constructing complex machine aggregating abilities are autonomous vehicle systems. The ultimate objective of these systems is to create a hybrid system inspired by the brain and capable of thinking and acting independently. The future of transportation is autonomous vehicles. Artificial intelligence is being employed in almost everything from trains to airplanes. An autonomous vehicle's main purpose is to perceive its surroundings and offer the optimal route without human intervention. This eliminates common human driving errors such as driving under the intoxicated, using a cell phone while driving, and speeding, which are the main causes behind accidents and traffic congestion [1]. Whether they are trains or airplanes, autonomous vehicles are the way of the future for all modes of transportation. They rely on artificial intelligence to implement autonomous sensing, decision-making, sensor modeling, and control in dynamic situations. All environmental factors should be dealt with in the shortest amount of time possible, and the reactions and judgments of autonomous cars must be more trustworthy than those of human drivers [2]. Machine learning, robotics, pattern recognition, and intelligent control have all been used in the past few decades to develop theories and applications for autonomous vehicles with varying degrees of autonomy, from fully human involvement to none at all [3, 4]. The learning skills of autonomous cars must thus be improved, including online learning, object identification, sensor planning, learning from ambiguous data, and motion control. Different Deep Learning DL techniques are crucial in achieving these skills since they help to extract knowledge from seen data [5].

Deep learning approaches enable performing activities that would be difficult to be performed with standard software; nonetheless, the choice of algorithm depends on the task's domain. The CNN technique for convolutional neural networks is often utilized in fields like object identification and image categorization. CNN is developed as a form of biologically inspired by deep learning that learns via a sequence of convolutional filters together with straightforward non-linearities [6–8]

To design a DL system that offers an effective improvement, a number of data sources must be found to provide full coverage of the expected input. Feature extraction is a crucial technique in several areas, including classification, diagnosis, detection, and identification. To decrease dimensionality, it removes the complexity of time, space, and machine learning [9–11].

Analysis by Principal Components Popular feature extraction techniques, like principal component analysis PCA, may pick out certain key features from all the feature components. PCA has been effectively used in deep learning applications. Most PCA applications involve rearranging samples into a new lower space [12].

This study presents a method for building a deep learning model by integrating feature extraction methods with our suggested hybrid model, which learns based on features retrieved from the PCA feature extraction method. In order to feed a suggested hybrid model that is constructed based on the principles of CNN and Dense methods, our model employs the PCA extraction approach to extract features. The main goal of our suggested deep model is to implement an object detector model that takes less than an hour to train because it uses learning features rather than large image datasets, consumes fewer resources, and improves performance because it can detect ambiguous objects as well as distant objects.

The paper is structured as follows: Section II presents the literature review. Section III represents the study methodology. Section IV illustrates the deep model's technique. Section V is reserved for the findings of the current study. Finally, Section VI includes the conclusions and future studies.

II. Related Works

In this section, the main previous works that address the object detection process using deep learning models are covered.

Tao Chen et. al. [13] presented a subcategory–aware CNN technique for object detection to solve the object intra- class variation problem, S-CNN employs a new loss function to capture the multiple sub-categories information. The training samples are first grouped into multiple sub-categories automatically through a novel instance sharing maximum margin clustering MMC algorithm; a multi-component aggregated channel feature ACF detector is then trained to produce more latent training samples. Here, each ACF component corresponds to one clustered sub-category. Experiments on INRIA person dataset, PASCAL VOC 2007 dataset, and MS_COCO dataset demonstrate superior object detection performance of the S_CNN as compared with state of art techniques, such as fast/ faster R_CNN and SSD with an accuracy of 49.5%.

Yousong Zhu and colleagues [14] introduced a novel convolutional network named CoupleNet, which captures global, local, and contextual signals for accurate object detection. The effectiveness of their approach was demonstrated through extensive experiments, and their algorithm achieved state-of-the-art results on three challenging datasets: 82.7% mAP on VOC07, 80.4% on VOC12, and 34.4% on COCO. CoupleNet performs at the cutting edge on the difficult PASCAL VOC and COCO datasets without the need of any additional testing methods, demonstrating the efficacy of our approach.

Xu et al [15] introduced a novel object detection framework named "Deep Ragionlets", comprising a region selection network and a deep regional learning module. The proposed framework is capable of identifying non-rectangular regions within the detection bounding box simultaneously, resulting in improved detection accuracy. To evaluate the effectiveness of their approach, the authors conducted experiments on two detection scales using the PASCAL VOC and Microsoft COCO datasets. The experimental results demonstrated competitive performance compared to state-of-the-art methods, exhibiting high accuracy in object detection, with an accuracy of 49.5% in COCO dataset.

Kumar et al [16] introduced an object detection model designed for assisting individuals with visual impairments, which employs deep learning neural networks. Their approach utilized a multi-box one-shot detection algorithm, coupled with a faster-area convolutional neural network architecture. To evaluate the model's effectiveness, the authors employed standard VOC and COCO datasets. Their results showed accuracy exceeding 75%. Moreover, the model includes a sound device, which can aid the visually impaired in object detection tasks.

Zhang et al. [17] displayed a deep learning-based object detection algorithm that is built on M2Det. Their work improved the feature pyramid generation in M2Det, resulting in more accurate object detection. The authors focused on applying their algorithm to fruit spoilage detection, which is a critical step in the pre-sale process. They evaluated the effectiveness of their approach using the COCO dataset. Furthermore, the proposed algorithm exhibited several favorable characteristics, including high accuracy, fast prediction speed, and real-time performance, highlighting its potential practical value, 44.2%.

Manoj Acharya et al. [18] have introduced a novel online object detection method called Replay Online Streaming

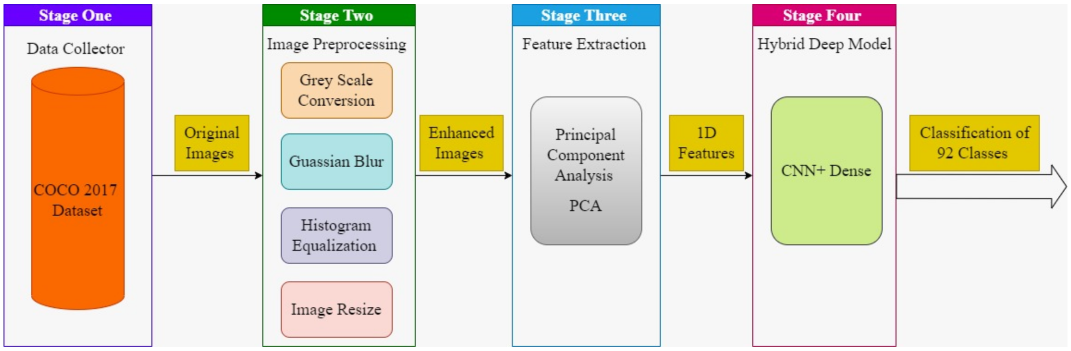

Fig. 1. Block diagram of the proposed method.

Object Detection (RODEO), which enables the detection of objects in an online streaming learning setting, where an agent learns one instance at a time with limited computational and memory resources. The proposed method uses replay learning and mid-level CNN features to address the problem of catastrophic forgetting in a fixed memory budget. RODEO has demonstrated state-of-the-art performance on two popular datasets, PASCAL VOC 2007 and MS COCO, achieving a high accuracy of 82.9% on COCO dataset.

Ehtesham Hassan et al. [19] proposed an object detection framework that incorporates handcrafted features into a convolutional neural network for object detection. Their approach is based on the latest version of YOLO's object detection architecture. The framework was evaluated on the PASCAL-VOC and MS-COCO datasets, resulting in a significant improvement in detection rates. Compared to YOLO version 3 detector by Redmon & Farhadi (2018), the proposed framework achieved an increase in detection rates of 11.4% and 1.9% on the mAP scale in PASCAL-VOC and MS-COCO datasets, respectively with an accuracy of 37.7% on COCO dataset.

In their study, Sethi et al. [20] suggested a deep learning-based methodology for detecting facial masks in public settings, aiming to mitigate the transmission of the Coronavirus within communities. The experiment was carried out by making adjustments to three widely used baseline models, namely ResNet50, AlexNet, and MobileNet. The suggested approach demonstrated a significant increase in accuracy, achieving a high accuracy rate of 98.2%, specifically when applied with ResNet50. In addition, the suggested model exhibits an accuracy and recall improvement of 11.07% and 6.44%, respectively, in the task of mask recognition when compared to the RetinaFaceMask detector, which is a recent public baseline model.

Mihir M. et al. [21] have used the COCO data set to detect real-time objects in self-driving car systems. The study relied on neural network algorithms from deep learning algorithms to help self-driving systems detect the path. The results showed that most algorithms do not give accurate results, but the proposed algorithm (NEAT algorithm) achieved high accuracy in detecting objects quickly.

K. Vaishnavi et al. [22] aimed to develop a deep learning-based model that can effectively recognize and classify items inside images. The research employs an enhanced solid-state drive (SSD) methodology in conjunction with a multilayer convolutional network. More than 80% of the predictions provided by the suggested model are valid. Convolutional neural networks use feature mapping to get the class label after the removal of feature data from the image.

The current study employs principal component analysis (PCA) to extract features from the modified MS_COCO 2017 images dataset in order to assist deep classification. The objective of the proposed deep model is to accurately identify all 92 classes in the COCO dataset while requiring minimal training time and hardware requirements. By attaining state-of-the-art deep model performance, this research contributes to the advancement of autonomous driving car technologies.

III. Methodology

In this work, a brand-new deep learning algorithm is created. This algorithm learns based on features and applies them to the prediction of 92 classes from the COCO2017 dataset in order to build object detector aims to enhance autonomous driving technologies. As explained in Fig. 1, the proposed method is based on four stages. The first stage is the data collector, were we use data from the COCO dataset. The second stage is the preprocessing of our datasets, which is a common technique in computer vision to expand the existing data to produce new data. This stage involves a number of processes: converting images to grey scale, removing image noise's to improve the size and lighting of the images. Then the third stage is the features extraction, features are retrieved using principal component analysis, which is more practical feature extraction method for indoor and outdoor data, as

Fig. 2. Sample images from ms- coco 2017 dataset.

in this situation. Convolutional neural networks and dense concepts are used to construct the deep model. Its creation is based on the excellent traits retrieved from the preceding stage. This stage will be explained in detail in the next section. The dataset utilized in our deep model, the methods employed for preprocessing and feature extraction, and the performance indicators used for assessing our model are all covered in detail in the subsections that follow.

A. The Dataset

The Common Objects in COntext (COCO) [23] dataset, a sizable dataset of images with high COCO complexity, is the only one included in the training dataset. This includes images of complicated scenes with numerous little items that are tagged with extremely precise outlines in order to facilitate the study of thing-thing interactions. According to certain research articles [21], COCO is superior to other well-known datasets that are often utilized. These datasets include IMAGE NET, SUN, PATTERN ANALYSIS, STATISTICAL MODELING, and computational learning of visual object classes. The size and classifications of the aforementioned dataset varied greatly, although they were finely grained. It made use of a variety of images from the outside world and from nature. COCO was created for the purpose of detecting and classifying items in their traditional natural setting. The MS COCO 2017 dataset covers a considerable number of dense objects, tiny objects, and objects with large scale changes, including 92 categories, with each category having its own category information and local label information. Sample images from the MS- COCO 2017 dataset are shown in Fig. 2.

B. Pre-processing Stage

Image preprocessing is a multidimensional process where the main goal of image enhancement is to change a image's characteristics so that they are more suited to a certain purpose and viewer. One or more image characteristics are changed throughout this operation. In this study, preprocess operations like Grey conversion, Gaussian Blur, Image Resize, as well as Histogram Equalization [23–25] are used. The sequence of these operations is an important aspect in preparing images before PCA feature extraction step. The concept of these processes is explained briefly below.

1) Grey Scale Conversion

A procedure of mapping from a higher-level vector space into a lower dimensional space is used to transform an image from RGB (Red, Green, and Blue) space to greyscale space. Grayscale images have a single channel (one dimension), 256 shades of gray, which are represented in 8-bit. The essential feature of grayscale images is that they have equal levels of red, green, and blue color. In some applications, it is necessary to convert color images to grayscale, as some display and image capture devices can only handle 8-bit images. The conversion equation from RGB mode to grayscale is given by (1).

$$Gray\ Scale\ Image = ((0.3 \times R) + (0.59 \times G) + (0.11 \times D)) \tag{1}$$

2) Gaussian blur

A linear low-pass filter called Gaussian Blur is used to blur, smooth, and remove noise from photographs. The Gaussian Blur may be calculated using the following function.

$$G(x,y) = \frac{1}{(2\pi\partial^2)\, e^{\left(\frac{x^2+y^2}{2\sigma^2}\right)}} \tag{2}$$

In this context, the variables x and y denote the distances from the reference point or origin in the horizontal and vertical directions, respectively. Additionally, the σ variable represents the standard deviation of the Gaussian distribution, which is a measure of the spread of the probability density function. Together, these parameters define the characteristics of a two-dimensional Gaussian distribution, which can be used in various applications such as image processing, signal filtering, and statistical modeling. The value determines the extent of the distribution and the relative influence of neighboring points on the overall shape. Generally, as increases, the distribution becomes wider and flatter, indicating a greater degree of uncertainty or variation in the data.

3) Image Resize

The resizing process plays a key role in both reducing and enhancing the size of the image. There are two ways to interpolate a image: by down- or up-sampling the original image. In many situations, selecting an accurate interpolation technique is crucial. In this study, we reduce the images resize to (20×20); the image size determines the number of PCA features.

4) Histogram Equalization

Histogram processing is a widely used method to enhance images, where histograms are typically normalized by the total number of pixels in the image. A digital image's histogram, which has intensity levels between [0, L-1], is a discrete function. The histogram also serves as a cumulative distribution function, which is crucial in computing histogram equalization, as shown in equation (3). Histogram equalization is a popular image enhancement method used in digital photography to increase image intensity.

$$\mathrm{C_{df}} = \sum_{\mathrm{i}=1}^{\mathrm{X}} \mathrm{h(i)} \tag{3}$$

where h shows the image's histogram and X stands for the grey value.

$$\mathrm{T[\,pixel\,]} = \mathrm{round}\left(\left(\frac{\mathrm{cdf(x)} - \mathrm{cdf(x)_{min}}}{\mathrm{E} * \mathrm{f} - \mathrm{cdf(x)_{min}}}\right) * (\,\mathrm{L}-1)\right. \tag{4}$$

C. Principal Component Analysis (PCA)

A mathematical approach called principal component analysis (PCA) [26], also known as Karhunen-Loeve expansion, employs orthogonal transformations to examine and describe the data. In order to create a set of values for a set of linearly uncorrelated variables, a collection of characteristics of potentially correlated variables must first be converted. PCA has been applied in many areas, including signal processing, image processing, and machine learning, as it enables the representation of a function in terms of its key properties. In order to discriminate between objects, PCA employs the Eigenvectors technique, which involves extracting the distinctive characteristics from the image and representing the target objects as a linear combination of the so-called "eigenvectors" produced from the feature extraction process. The number of features extracted in our study is 400, which depends on the images size, which in our case is 20* as determined by the subsequent procedure.

D. Indices for Model Evaluation

The model's performance in multi-class image classification was evaluated using four metrics, namely Accuracy, Recall, Precision, and F1-Score, based on the results obtained from the testing set classification. These metrics provide a comprehensive evaluation of the model's performance across all classes, with the Accuracy that is describing the overall performance of the model (7), Recall that is measuring the fraction of relevant instances retrieved (8), Precision that is indicating the fraction of retrieved instances that are relevant (9), and F1-score that is providing a balanced measure of recall and precision (10). These four indices [27] were chosen as the primary evaluation metrics.

The analysis of our proposed object detection technique for self-driving cars is presented in this section. TP represents the number of correctly identified positive samples, FP represents the number of negative samples incorrectly identified as positive, and FN represents the number of positive samples predicted as negative, and TN represents the number of accurately predicted samples belonging to the negative class by a given model.

Algorithm 1: PCA Algorithm

Input: Processed Images.
Output: feature vector.
M The training set of all the images.
μ reflects the median Mean.
Sub reflect the average image μ after being subtracted..

Begin:

Step1: Make a training set of total M images to use in the in computing the Average Mean as shown in the equation (5)

$$Avarage = \frac{1}{M}\sum_{n=1}^{M} Trainingimages(n) \tag{5}$$

Step2: Subtract the original image from the Average Mean as shown in the equation (Sub= Training images- Average).

Step3: Calculate the Covariance Matrix's as shown in equation (6).

$$\text{covariance} \ = \ \sum_{n=1}^{M} \mathrm{sub}(n)\,\mathrm{sub}^{T}(n) \tag{6}$$

Step4: Compute the eigenvectors and corresponding eigenvalues for the matrices.

Step5: Select the top Eigenvalues after sorting. We select the M Eigenvectors that best characterize the set of greatest eigenvalues that make up the group of eigenvectors. The Eigenvectors can be adjusted or computed as needed to take into account the appearance of fresh faces.

Step6: Project the training sample data onto Eigenvectors.

End

$$\text{Accuracy} \ = \frac{\mathrm{Tp}+\mathrm{TN}}{\mathrm{TP}+\mathrm{Fp}+\mathrm{TN}+\mathrm{FN}'} \tag{7}$$

$$\text{Recall} \ = \frac{\mathrm{Tp}}{\mathrm{TP}+\mathrm{FN}'} \tag{8}$$

$$\text{precision} = \frac{\mathrm{Tp}}{\mathrm{Tp}+\mathrm{Fp}} \tag{9}$$

$$\mathrm{F1-\ score} = \frac{2 * \text{ Precision } * \text{ Recall}}{\text{Precision } + \text{ Recall}} \tag{10}$$

IV. Deep Learning Model

The deep model utilized in our study, as illustrated in Fig. 3, is a hybrid model that incorporates a sequence of convolution layers featuring filters (kernels), pooling, and fully connected layers. The model is designed to classify objects and generate probabilistic values ranging from 0 to 1. Specifically, the proposed deep model consists of a one-dimensional network layer comprising 26 layers. The utilization of 1D CNNs offers the benefit of minimal computational demands, rendering it suitable for real-time applications. Additionally, the straightforward and condensed structure of 1D CNNs permits low-cost hardware implementation.

The present study investigates the viability of a low-cost hardware implementation strategy for a neural network architecture consisting of eight deep layers that are formed by one-directional convolutional layers. The utilization of a 1D convolutional layer involves the creation of a convolution kernel that traverses a singular spatial or temporal dimension, resulting in the generation of a tensor of outputs. Additionally, the implementation of fully connected layers is achieved through the utilization of Dense layers, which are the most commonly employed and frequently utilized layer type. The Dense layer executes a specific operation on the input and then subsequently returns the output.

The remaining layers comprise of six Maxpooling layers. The process of downsampling the input representation of 1D temporal data through the use of a maxpooling operation involves selecting the maximum value within a defined window size. The inclusion of a Max pooling layer between convolutional layers results in an increase in spatial abstractness as feature abstractness increases. Maxpooling is a process that involves computing the maximum value within each patch of a given feature map.

The neural network architecture comprises of a single normalization layer, which is represented by a flattened layer. This layer takes the output of the preceding layers, flattens them, and converts them into a singular vector that can be utilized as an input for the subsequent stage. Furthermore, there are seven leaky rectified linear units (leaky RELU) activation layers. The RELU activation function was observed to enhance the training efficiency of deep neural networks in contrast to conventional activation functions, owing to its derivative being equal to 1 for positive inputs.

The features (3,713,600) are partitioned into receptive fields which are then utilized as inputs for convolutional layers. The neural network comprises a set of layers that are sequentially stacked on top of each other, with an input size of 400 features. The convolutional layers in this study utilize filters of varying sizes, specifically 16, 32, 64, 128, 256, 512, and 1024, as well as 485; each with a kernel size of 3, a stride of 1, and identical padding. The Maxpooling layers exhibit a stride of 1, a pool size of 2, and employ the same padding technique. Conversely, the linear collectors or Dense layers employ varying kernel sizes of 512, 256, 128, and 92, respectively, and are associated with distinct activation functions. Specifically, the first three layers utilize the linear activation function, while the last layer employs the softmax activation function.

V. Results

The model was implemented using the open-source deep learning framework, Keras, with Python programming language. All experiments were conducted on a Windows 10 Laptop with a 2.60GHz Intel(R), Core(TM) i7-10750H CPU (16 GB memory) and a NVIDIA GeForce GTX1660Ti GPU (6 GB memory). For our detection experiment, we used a modified version of the Ms-COCO 2017 dataset, which consists of 92 classes. In this paper, we fed 3,713,600 PCA features extracted from our dataset, which consists of 400 features extracted from one image multiplied by 9284 (the total number of dataset images) as inputs to the network. PCA features were divided into 80 % training samples and 20% test samples.

Accuracy, Precision, Recall, and F1-score were the four performance indicators chosen to assess the model. Fig. 4 shows the performance indices parameters that were attained in our model for the various 92 classes. Moreover, the proposed model shows promising results in accuracy, where it achieves 99.96% in both training and validation phases as explained in Fig. 5. The proposed deep model contains a total of 3,162,741 parameters and is trained across 100 epochs. The key benefit of the proposed model is that it reduces computing time since one epoch's training takes about 46 seconds, as shown in Fig. 6. The proposed model is also conducted in real-time mode, hence collected data from various Iraq locations are employed, and samples of the precise findings are displayed in the image. We have demonstrated a variety of items that the suggested

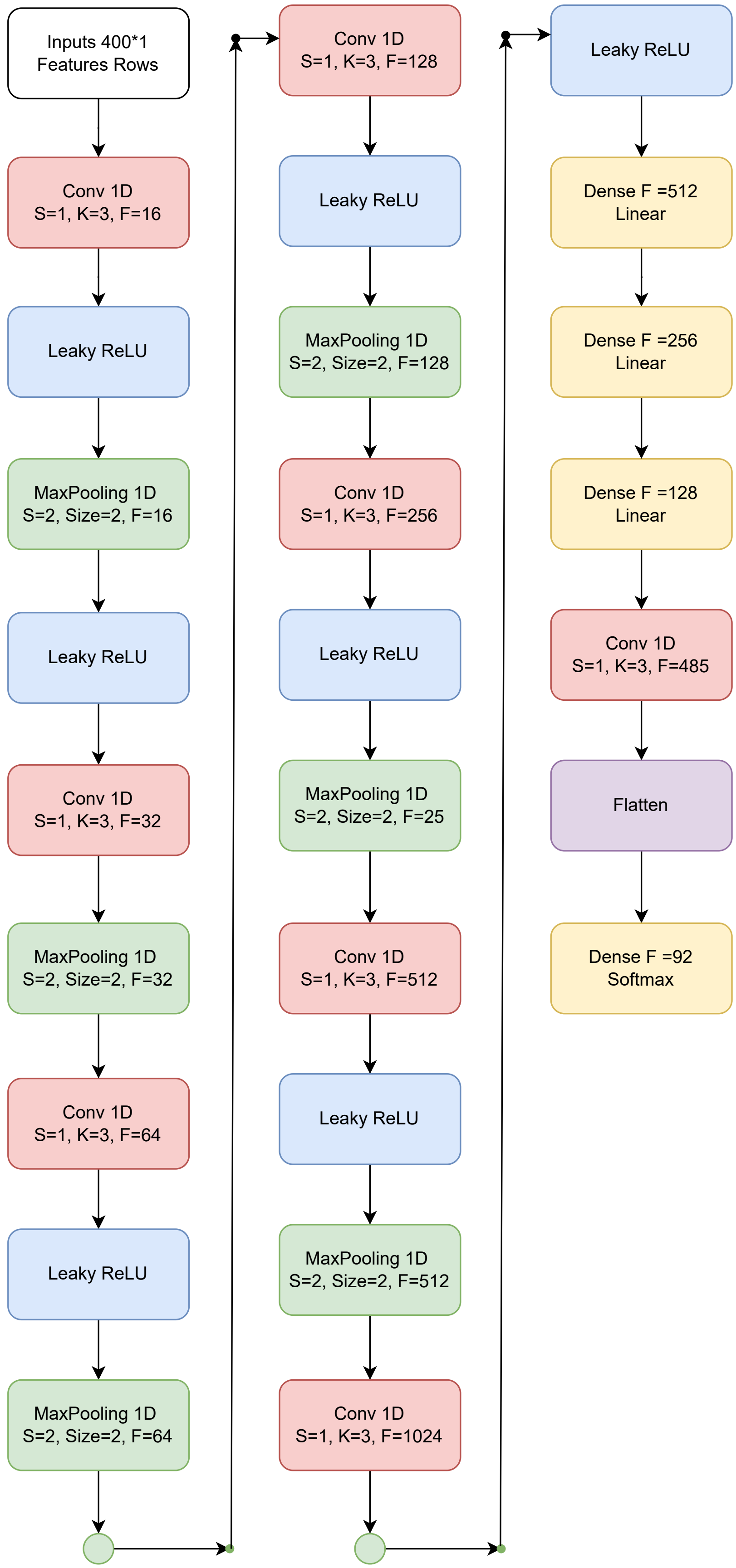

Fig. 3. Block diagram of the proposed deep model construction.

TABLE I.
COMPARISON SUMMARIZATION

Tao Chen et al [13]	2017	S-CNN	49.5
Yousong Zhu [14]	2017	S-CNN	49.5
Hongyu Xu et al [15]	2018	Deep regionlets,	59.5
Ashwan Kumar et al. [16]	2019	Proposed approach modified faster r sun+ ssd with additional layers	78.68
Tao Zhang [17]	2020	Modified M2DET	44.2
Manoj Acharya [18]	2020	RODEO	82.9
Ehtesham, Hassan et al [19]	2020	Novel object detection approach which fuses hand crafted features with the latest version of yolo detectors	The best experiments resulted at 37.7
Shilpa Sethi et al. [20]	2021	Resdet 50 Alextet Mobile Net	98.2
K. Vaishnavi et al. [22]	2023	Modified SSD	97.50
Our Proposed Model	2023	Hybrid model	99.96

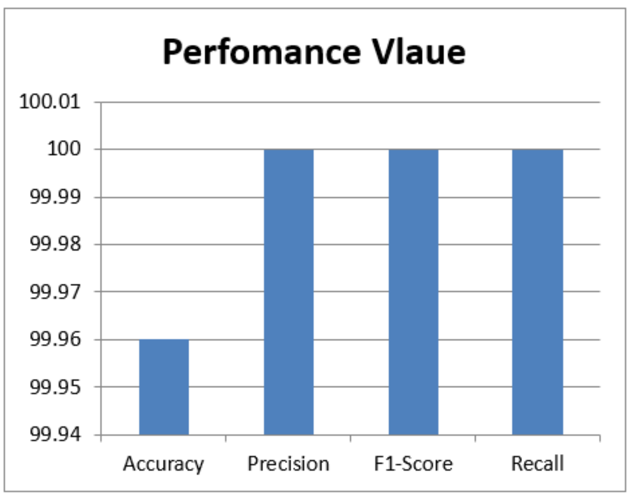

Fig. 4. The performance indices of the proposed model.

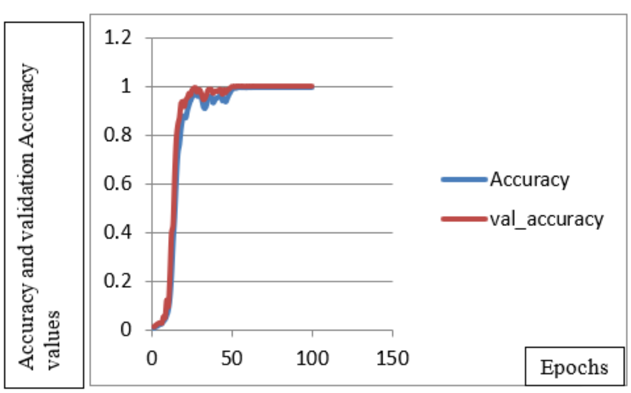

Fig. 5. Resulting accuracy.

```
- 43s 5ms/step - loss: 4.4076 - acc: 0
- 43s 5ms/step - loss: 4.3589 - acc: 0
- 44s 5ms/step - loss: 4.3064 - acc: 0
- 43s 5ms/step - loss: 4.1656 - acc: 0
- 43s 5ms/step - loss: 4.0411 - acc: 0
- 43s 5ms/step - loss: 3.6749 - acc: 0
- 43s 5ms/step - loss: 3.1639 - acc: 0
- 43s 5ms/step - loss: 2.5224 - acc: 0
- 43s 5ms/step - loss: 2.0055 - acc: 0
```

Fig. 6. Computational time for epoch training.

Fig. 7. object detection samples of the proposed deep model.

Fig. 8. (a-f) Object Detection samples of our deep model (cont.)

method discovered, along with their corresponding class labels as shown in Fig. 7.

VI. Conclusions

Self-driving cars require robust object detection techniques to operate safely and efficiently. In this paper, we propose a novel approach for object detection that leverages PCA feature extraction and a hybrid deep learning model to achieve state-of-the-art performance. We describe the detailed methodology for fusing PCA features with our detector model and evaluate its performance on the COCO dataset. To enhance the model's capabilities, we employ a MS- COCO2017 dataset and perform a series of image augmentation techniques to generate millions of images. Our model is trained on 92 different object classes. We conducted extensive experiments on our dataset and evaluated the performance of our model using several metrics: Accuracy, Precision, Recall, and F1-score. This analysis allowed us to identify the strengths and weaknesses of our approach. To further improve accuracy, we employed rigorous training and verification procedures, resulting in a high-performing model with an accuracy of 99.96% in both the training phase and the testing phase. Our proposed approach represents a significant advancement in object detection for self-driving cars, with the potential to enhance the safety and reliability of autonomous vehicles, with minor additions can be applied with hardware equipment's.

Conflict of Interest

The authors have declared no conflict of interest.

References

[1] L. C. Possatti, R. Guidolini, V. B. Cardoso, R. F. Berriel, T. M. Paixão, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "Traffic light recognition using deep learning and prior maps for autonomous cars," in *2019 international joint conference on neural networks (IJCNN)*, pp. 1–8, IEEE, 2019.

[2] H. Liang, X. Sun, Y. Sun, and Y. Gao, "Text feature extraction based on deep learning: a review," *EURASIP journal on wireless communications and networking*, vol. 2017, pp. 1–12, 2017.

[3] M. Masmoudi, H. Friji, H. Ghazzai, and Y. Massoud, "A reinforcement learning framework for video frame-based autonomous car-following," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 2, pp. 111–127, 2021.

[4] B. Alvey, D. T. Anderson, A. Buck, M. Deardorff, G. Scott, and J. M. Keller, "Simulated photorealistic deep learning framework and workflows to accelerate computer vision and unmanned aerial vehicle research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3889–3898, 2021.

[5] V. Thanh Dat, P. Quang Huy, N. Dinh Tra, T. Hai Anh, B. Ngoc Anh, and N. Tung Son, "A deep learning based implementation for self-driving car," in *Proceedings of the 2021 10th International Conference on Software and Computer Applications*, pp. 279–287, 2021.

[6] X. Ye and Q. Zhu, "Class-incremental learning based on feature extraction of cnn with optimized softmax and one-class classifiers," *IEEE Access*, vol. 7, pp. 42024–42031, 2019.

[7] M. Maggipinto, A. Beghi, S. McLoone, and G. A. Susto, "Deepvm: A deep learning-based approach with automatic feature extraction for 2d input data virtual metrology," *Journal of Process Control*, vol. 84, pp. 24–34, 2019.

[8] X. Li, W. Zhang, and Q. Ding, "Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction," *Reliability engineering & system safety*, vol. 182, pp. 208–218, 2019.

[9] Y. Xianjia, S. Salimpour, J. P. Queralta, and T. Westerlund, "Analyzing general-purpose deep-learning detection and segmentation models with images from a lidar

as a camera sensor," *arXiv preprint arXiv:2203.04064*, 2022.

[10] M. Abdulla and A. Marhoon, "Agriculture based on internet of things and deep learning," *Iraqi Journal for Electrical and Electronic Engineering*, vol. 18, no. 2, pp. 1–8, 2022.

[11] H. Hakim and A. Fadhil, "Survey: Convolution neural networks in object detection," in *Journal of Physics: Conference Series*, vol. 1804, p. 012095, IOP Publishing, 2021.

[12] S. P. Mishra, U. Sarkar, S. Taraphder, S. Datta, D. Swain, R. Saikhom, S. Panda, and M. Laishram, "Multivariate statistical data analysis-principal component analysis (pca)," *International Journal of Livestock Research*, vol. 7, no. 5, pp. 60–78, 2017.

[13] T. Chen, S. Lu, and J. Fan, "S-cnn: Subcategory-aware convolutional networks for object detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 10, pp. 2522–2528, 2017.

[14] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, and H. Lu, "Couplenet: Coupling global structure with local parts for object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 4126–4134, 2017.

[15] H. Xu, X. Lv, X. Wang, Z. Ren, N. Bodla, and R. Chellappa, "Deep regionlets for object detection," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 798–814, 2018.

[16] A. Kumar, S. S. S. Reddy, and V. Kulkarni, "An object detection technique for blind people in real-time using deep neural network," in *2019 Fifth International Conference on Image Information Processing (ICIIP)*, pp. 292–297, IEEE, 2019.

[17] T. Zhang and L. Li, "An improved object detection algorithm based on m2det," in *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pp. 582–585, IEEE, 2020.

[18] M. Acharya, T. L. Hayes, and C. Kanan, "Rodeo: Replay for online object detection," *arXiv preprint arXiv:2008.06439*, 2020.

[19] E. Hassan, Y. Khalil, and I. Ahmad, "Learning feature fusion in deep learning-based object detector," *Journal of Engineering*, vol. 2020, no. 1, p. 7286187, 2020.

[20] S. Sethi, M. Kathuria, and T. Kaushik, "Face mask detection using deep learning: An approach to reduce risk of coronavirus spread," *Journal of biomedical informatics*, vol. 120, p. 103848, 2021.

[21] M. M. Parmar, R. R. Rawlo, J. L. Shirke, and S. Sangam, "Self-driving car," *Int. J. Res. Appl. Sci. Eng. Technol.(IJRASET)*, vol. 10, pp. 2305–2309, 2022.

[22] K. Vaishnavi, G. P. Reddy, T. B. Reddy, N. C. S. Iyengar, and S. Shaik, "Real-time object detection using deep learning," *Journal of Advances in Mathematics and Computer Science*, vol. 38, no. 8, pp. 24–32, 2023.

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.

[24] V. M. Mohan, R. Kanaka Durga, S. Devathi, and K. Srujan Raju, "Image processing representation using binary image; grayscale, color image, and histogram," in *Proceedings of the Second International Conference on Computer and Communication Technologies: IC3T 2015, Volume 3*, pp. 353–361, Springer, 2016.

[25] L. M. Francis and N. Sreenath, "Pre-processing techniques for detection of blurred images," in *Proceedings of International Conference on Computational Intelligence and Data Engineering: Proceedings of ICCIDE 2018*, pp. 59–66, Springer, 2019.

[26] A. S. M. M. A. A. Z. M. H. A. Karamizadeh, S., "An overview of principal component analysis," *Journal of Signal and Information Processing, 4(3B), 173*, 2013.

[27] H. Luo, P. Wang, H. Chen, and V. P. Kowelo, "Small object detection network based on feature information enhancement," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, p. 6394823, 2022.