Computers, Materials & Continua
DOI:10.32604/cmc.2020.013231
Article

Tech Science Press

Autonomous Parking-Lots Detection with Multi-Sensor Data Fusion Using Machine Deep Learning Techniques

Kashif Iqbal[1,2], Sagheer Abbas[1], Muhammad Adnan Khan[3,*], Atifa Ather[4], Muhammad Saleem Khan[1], Areej Fatima[3] and Gulzar Ahmad[1]

[1]School of Computer Science, National College of Business Administration & Economics, Lahore, 54000, Pakistan
[2]Department of Computer Sciences, GC University, Lahore, 54000, Pakistan
[3]Department of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan
[4]Department of Computer Science, Comsats University Islamabad, Lahore Campus, Lahore, 54000, Pakistan
[*]Corresponding Author: Muhammad Adnan Khan. Email: madnankhan@lgu.edu.pk

Received: 30 July 2020; Accepted: 11 September 2020

Abstract: The rapid development and progress in deep machine-learning techniques have become a key factor in solving the future challenges of humanity. Vision-based target detection and object classification have been improved due to the development of deep learning algorithms. Data fusion in autonomous driving is a fact and a prerequisite task of data preprocessing from multi-sensors that provide a precise, well-engineered, and complete detection of objects, scene or events. The target of the current study is to develop an in-vehicle information system to prevent or at least mitigate traffic issues related to parking detection and traffic congestion detection. In this study we examined to solve these problems described by (1) extracting region-of-interest in the images (2) vehicle detection based on instance segmentation, and (3) building deep learning model based on the key features obtained from input parking images. We build a deep machine learning algorithm that enables collecting real video-camera feeds from vision sensors and predicting free parking spaces. Image augmentation techniques were performed using edge detection, cropping, refined by rotating, thresholding, resizing, or color augment to predict the region of bounding boxes. A deep convolutional neural network F-MTCNN model is proposed that simultaneously capable for compiling, training, validating and testing on parking video frames through video-camera. The results of proposed model employing on publicly available PK-Lot parking dataset and the optimized model achieved a relatively higher accuracy 97.6% than previous reported methodologies. Moreover, this article presents mathematical and simulation results using state-of-the-art deep learning technologies for smart parking space detection. The results are verified using Python, TensorFlow, OpenCV computer simulation frameworks.

Keywords: Smart parking-lot detection; deep convolutional neural network; data augmentation; region-of-interest; object detection

1 Introduction

The Internet has generally changed our lives, from the way we connect, the way we conduct business locally or globally, and how we move or travel. The Internet-of-Things (IoT) is a multi-dimensional expanding service of interconnected devices, networks, peoples, and valuable things that are provided with radio frequency identifications (RFIDs) and the ability to publish data over a smart-cloud without requiring a human-to-human interaction. Artificial intelligence, internet of things and big data analytics technologies are at hype nowadays and will stay to solve future transportations and other challenges. These ecosystem technologies are in an escalating period of growth in both the military, government, commercial sphere, and part of leading jobs to monitor and leverage those advanced developments. According to the world health organization (WHO) [1] reported in 2020, vehicular accidents damaging resources and killing approximately 1.35 million peoples worldwide annually. Therefore, an urgent need for the development of autonomous driving assistant solutions for reducing these vital accidental deaths ratio by improving roadways safety and surveillance. Companies are striving to develop technological solutions to ensure, guarantees reduction by automating driving tasks because almost more than 90% of these accidents are caused by human error [2]. The objective of this research is to improve people's lives by enabling adaptive transportation systems and integrate such services. Self-driving cars innovations are critical to that mission: it can make our streets, roadways safer, cities greener and reducing traffic issues. There are many roadside smart infrastructures, installed multi-sensor ecosystem everywhere in the smart-city for collecting data through video-cameras, global positioning systems (GPS), inertial measurement units (IMUs), variable messaging signs (VMSs), light detection & ranging (LiDAR), and microminiaturized electromechanical sensors (MEMS) Technologies within the vehicles suggested by [3–5]. These devises collecting live information and transmitted to the collecting spatial centralized edge-gateway computers or the autonomous cloud-gateway servers. Big-data analytics utilizing fusion of data, fusion of applications and advantages of transfer learning modules would stream-line most of the autonomous data processing tasks to work together and play a pivotal role in data-driven services worldwide proposed by Shivappa et al. [6].

The ecosystems of connected devices do most of the data preprocessing tasks without human intervention, although humans may interact with these devices as well, simultaneously set them up and give instructions to actuators. The data-linking, networking, and communication protocols are used with these IoT-enabled devices, mostly dependent on Open-IoT APIs deployed on mobile devices by [7].

The recent results of self-driving vehicle systems, object identification showing improve results. Autonomous vehicle modalities and innovative ideas are one of the blazing application areas of the AI-ML research community recently, it can be benefited greatly from advanced technologies such as image augmentation, virtual reality, augmented reality, semantic segmentation, and explainable AI-techniques. However, with the advent of XR-AI, it might just be one step closer to make machines accountable, reasonable for their actions in the same manner as that humans may do [8].

The rest of the paper is organized as in Section 2, we discuss the general concept of vision-based technologies. Also, we look into the different groups of neural networks such as support vector machine (SVM), filter-based object reorganization, classification as well as optimized neural network model. In Section 3 we will discuss different machine learning techniques that provide the solution for many other object detection problems. Approaches such as deep convolutional neural network (D-CNN), recurrent convolutional neural network (R-CNN), inception-vs, and mask-RCNN will be discussed here. In Section 4, the comparison of the techniques in terms of its benefits and tradeoff are described based on statistical metrics. Also, we see how to improve the connected vehicles systems utilizing the proposed machine deep learning F-MTCNN model for the autonomous parking-lot detection. Finally, in Section 5, we will conclude this paper by looking back into all these methods and discuss whether these methods have somehow managed to formulate the solutions to the subject area intelligent transportation system (ITS).

2 Preliminaries

2.1 Vision-Based Modalities

Video-cameras have been installed all-around the city on poles, lamppost, and fixed on walls which are always vertical to the ground level. These mounted cameras are enabling the surveillance along with smart parking-lots detection along roadsides that how many vehicles are present in the parking areas. Smart parking solutions with multi-sensor data fusion is work together by senses of vision, hearing and touching systems are used to help us to navigate and understand its local or global positions. These sensor-suites are performing reasoning by using multimodal data collection utilization in AI-based ML-systems resulted in perfections as describe in following Fig. 1. To design a system to leverage the sensor's complementary strengths, for example, a lidar provides very accurate depth environment mapping, localizing and tracking but does not provide color sensing information (so it cannot tell when a signal light is green or red). Smart-cameras can fill this gap by providing color information but don't capture an object's depth (so it can be challenging to localization and mapping of objects). Radar provides mobility measurement velocity of the objects, complementing both lidar and cameras. The multi-sensor fusion applications required proper machine learning techniques with intelligent system design module as describe in Fig. 1 below.

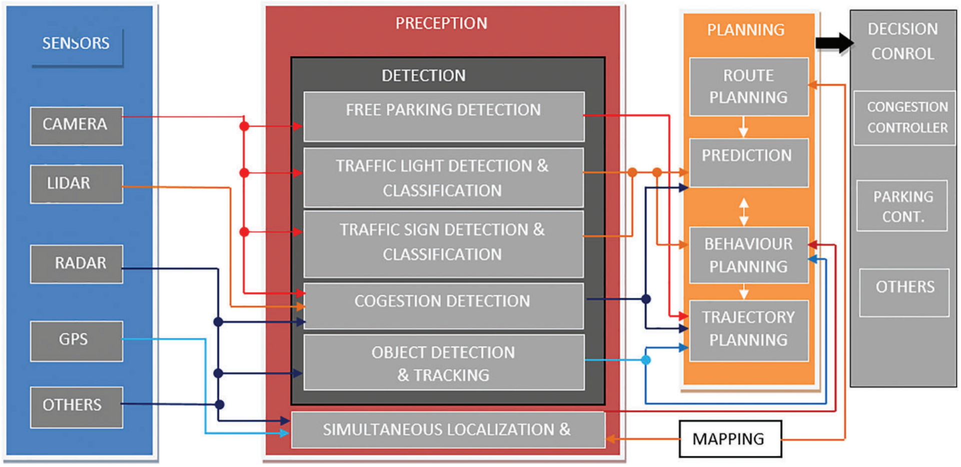


Figure 1: Multi-sensors fusion of data required proper machine learning techniques [9]

2.1.1 Sensors

A comprehensive autonomous driver assistant system (ADAS) gathers data with multiple-sensor that decides whether to plane, take the control decision to move based on a data-driven approach that could result in possible new solutions of current potential challenges. Multi-sensor data collection besides cameras ADASs further use sensors such as light detection and range finders (LIDAR), GPS, RADAR, inertial measurement units (IMUs) and more recently employing video cameras for data collections in ego-vehicles. ADAS can communicate with external aerial wireless network devices, satellites, or global position systems to help the driver with alternative routes planning and real-time information sharing proposed by Kubler et al. [10].

2.1.2 Stereo-Camera

To detect the presence of vehicles in parking 360°-cameras recording covers approximately 200 m range or above surroundings is used to detect the availability of free parking spaces and possibly guide the vehicle autonomously the availability of parking space. Embedded cameras also support multi-streaming to expand its functionally to monitoring surveillance security solutions as well as outdoor, indoor parking solutions. These cameras play an important role in applications as they are inexpensive, easy to install, and easy to maintain. Using close-circuit television (CCTV) cameras make it possible to monitor general open-areas without the need for other expensive sensors. In-vehicle smart-cameras provide blind spot detection, 3D object mapping, localizing, and other proactive safety measurements provide by end-to-end connectivity autonomously.

2.1.3 Sensor-Suites

LiDAR technology involves collecting and ranging the surrounding environment obstacles that can be fixed (i.e., mounted on a pole, or in vehicles), or it is mounted in moving vehicles is used to visualize the concept of time-of-flight. Lidar sends infrared lights beam that can detect smaller objects better (like obstacles, storms, bicyclists, and nearby objects) in nearer displacement based on the notion of time-of-flight. The interpretation of LiDAR generally involves perception, localization of the planned routing area. A radio-wave detection and ranging (RADAR) with a front-mounted camera provided enough information to analyze the road space ahead of the car, detecting road signs, traffic lights, other objects for information processing as that a human eye perceives with back-mounted radar as well. In each vehicle, a GPS-tractor is located onboard sensors to support vehicle tracking using google-maps, apple-maps, location identification.

2.2 Connected-Sensors Data Fusion

The preceding sensor-suites described their strengths and limitations so the researchers take multi-sensor data integration approaches in synchronization, configuration, and calibration of key features that have becomes more important for accurate, reliable performance for the autonomous driving system. These sensors are used to augment the geographic information system (GIS) and global position system (GPS) sensing technologies to track the vehicle in accurate and precise manner.

Shivappa et al. described numerous multimodal fusion data techniques and an excellent survey is presented very well in the article. An example of late fusion how a camera and lidar detection separately and combine to produce effective outputs as illustrated in Fig. 2b. Sensors of distant sensing technologies such as radar, lidar to others IMUs sensors, near-infrared radiation, and far frequency infrared devices have also been employed in all ego cars. Broadly categorize information integration into the five categories are described hereby: Early-fusion, parameter-level fusion, classifier-level fusion, final-level fusion, semantic-level fusion.

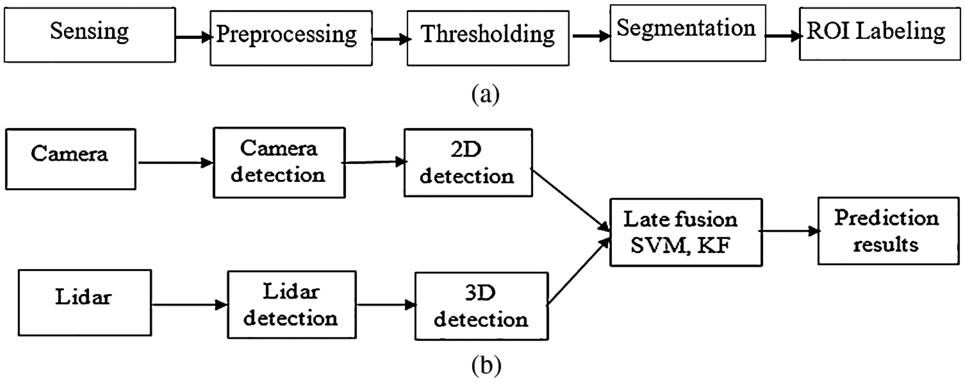


Figure 2: (a) All-purpose flowchart of machine learning modules, (b) Over-all multi-sensor fusion data preprocessing system

3 Background Review

Today, all leading automobile companies are developing algorithms for self-driving vehicles. Autonomous-cars (driverless cars, drones, autonomous-robots what so ever) are happenings, and appreciations all around the world currently. Autonomous-robots can perform the major human-like abilities in decision making just as a conventional car driver. Autonomous-cars are equipped with smart-cameras, GPS, LiDAR, LADAR, and advanced sensor technologies. The software empowering by tesla motors, general motors (GM), waymo formally google's self-driving vehicle is known as google chauffeur and uber recently allowed testing of self-driving cars without a steering wheel and pedals on public roads describe by Krompier [11].

Seo et al. [12] described synchronization between local smart devices and built-in sensor-network using a middleware universal plug-and-play (UPnP) control area network (CAN) gateway system. The proposed gateway comprises a CAN communication device, a UPnP communication device, and a translator device. Real-time vehicle data transmitted and received by CAN communication device along with the carla simulator and smart infrastructure through the UPnP an open-IoT. They proposed an internal gateway scheduler that supports reliable real-time data communication and transmission to solve the delay problem.

The Chu et al. [13] proposed a fully-convolutional neural networks (F-CNNs) and field-of-view (FoV) based voting results for vehicle detection. This article augments the supervised info with region overlap, a category for each training region-of-interest (ROI), and draw bounding-box as a trained model.

Liu et al. [14] proposed a visual traffic surveillance framework that integrates deep neural networks with balanced image augmentation data set to solve the perception problem. The proposed method enhances the precision-recall measures of all the object classes which improve the overall system model accuracy.

Luo et al. [15] proposed a deep convolution neural network using a vehicle dataset taken from multiple viewpoints to construct a robust algorithmic model. This model comprises less than nine layers to percept the vehicle location trained on the machine learning algorithm has low accuracy as compared to the proposed deep convolutional neural network.

In this paper, the authors described a high capability of classification using deep learning stacked network framework to encode the key features of input data streams, the implementation requirement is on graphical processing unit (GPU) devices. The pre-trained models extracted as If-THEN rules on the network input signal flow result in high classification capability. The generated test of the neural network model is designed using a deep belief neural network (DBNN) with effective computational speed [16].

Open-source programs developed by ROVIS research group that integrates the applications of vision-based perception to sense roadside objects that integrate pre-trained models developed using machine learning algorithms. The proposed system provides comprehensive development steps of object detection, mapping, localization using a smart 360°-camera setting, data streams preprocessing noise filtering, labeling, and semantic segmentation, object recognition along with 3D scene reconstruction [17].

An integrated self-diagnosing system (ISDS) for an autonomous agent-based on IoT-gateways and model transfer learning techniques. Connected vehicles detecting traffic patterns and find autonomous vehicle parking-lots available and assist the driver through SMS-alert, variable messaging signs boards, or display on dashboard describe by Frost et al. [18]. Nowadays everyone wants every-things to be actual, real, and have access to everything or what is happening all-round the world instantly. For this reason, a live camera feeds along with real-time information-sharing on how much traffic is moving on roads and how many empty parking-lots are available displayed on mobiles screen, proposed by [19].

In this article, the researchers voluntarily work and screen their working framework's from somewhere on the planet through connectivity using GPS sensor density with other existing frameworks. Smart automation system utilizes big-scale computing hardware and software resources that people integrate of remote correspondence, to give the remote-control differences how mobile apps provide proactive accident warnings, early parking solutions reduce overall traffic congestion, and increase the awareness of emergency detection [20].

4 Proposed Multi-Task Fusion Model Using Machine Learning Methodology

In this study the proposed model employed an autonomous vehicle classification, parking space detection and count no. of vehicles with sensory input data by bringing their readings into a mutually synchronized framework. Precise calibration of key features with dimensionality reduction techniques is critical for the optimum performance. The design model serves as the prerequisite for data preprocessing, fusion of data with the deep neural network and enabling transfer-learning pretrained models. Fig. 3 below provides a detailed overview of deep extreme machine learning and layered fusion data working system. It provides an optimized solution for the problem that occurs at each time step in a smart parking environment. Outdoor parking solution mostly uses 360° video-cameras installed on the poles in open car parking area to guide and monitor vehicles in parking area. These cameras detect cars in parking lots then transmit the information of real-time available parking spaces displaying panel output units with 24/7 parking monitoring system. Each camera monitors parking lots as overlapped which significantly enhances detection rates with minimizing blind spots area.

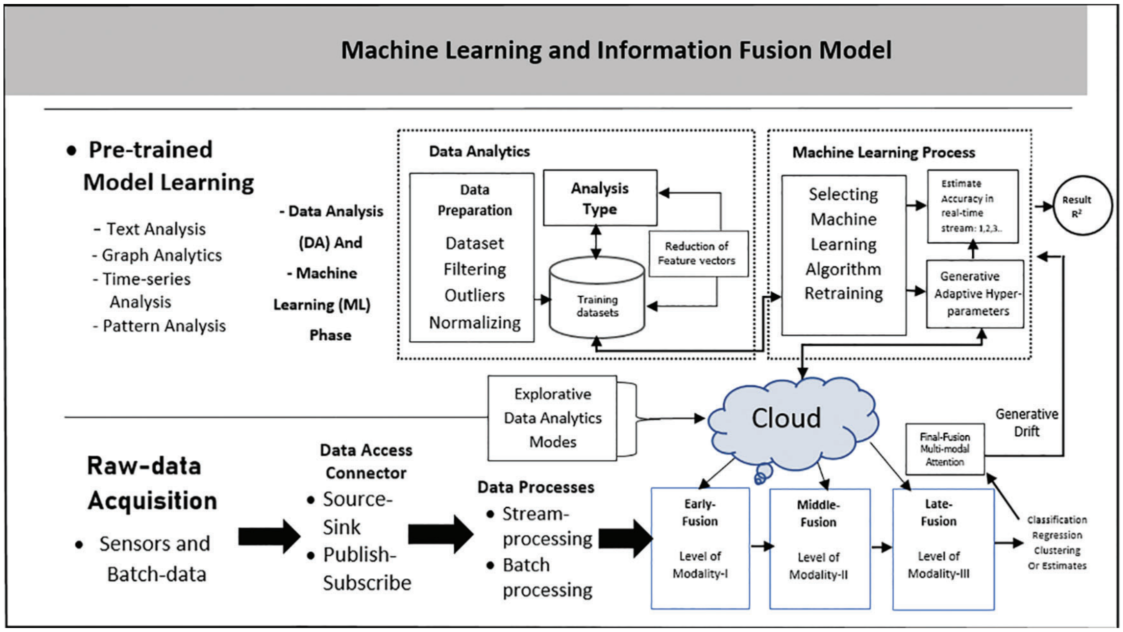

Machine Learning and Information Fusion Model
• Pre-trained Model Learning
- Text Analysis
- Graph Analytics
- Time-series Analysis
- Pattern Analysis
- Data Analysis (DA) And - Machine Learning (ML) Phase
Data Analytics
Data Preparation
Dataset Filtering Outliers Normalizing
Analysis Type
Reduction of Feature vectors
Training datasets
Machine Learning Process
Selecting Machine Learning Algorithm Retraining
Estimate Accuracy in real-time stream: 1,2,3..
Generative Adaptive Hyper-parameters
Result R^2
Explorative Data Analytics Modes
Cloud
Generative Drift
Final-Fusion Multi-modal Attention
Raw-data Acquisition
• Sensors and Batch-data
Data Access Connector
• Source-Sink
• Publish-Subscribe
Data Processes
• Stream-processing
• Batch processing
Early-Fusion Level of Modality-I
Middle-Fusion Level of Modality-II
Late-Fusion Level of Modality-III
Classification Regression Clustering Or Estimates

Figure 3: Proposed machine learning based data fusion architecture

4.1 Parking-Lots Detection with Data Augmentation

Data or image augmentation is a technique for semantically generating more training data for image classification. The growing large training dataset, it may remove the issue of model overfitting on the observation of the perceive results. For a particular image class, scene images that can be easily created, refined by rotating, cropping, resizing, or augmenting the original images. To mitigate the issue of model overfitting by false injection, it can add information to network weights considering as hyper-parameter added with image augmentation [21].

All the operations on the input images, that can be generated a lot more training dataset from our original input data frames, which makes our final trained model much more accurate, precise by applying these functions operations randomly. The pk-lot dataset aims to democratize access to thousands of parking vehicle images, and foster innovation in higher-level autonomy functions for everyone, everywhere. It looks like a computer talks to you and you talk to the computer.

4.2 Systems Parameters and Support Vector Mathematical Model

Support vector machine (SVM) algorithm does not work optimally with the datasets that are not linearly separable, firstly, it transforms features into a high dimensional space so the margin between two classes is maximized, SVM is a standard classifier works optimally with the linearly separable image datasets. This problem can be reduced by using "Kernel Trick" a method that returns the dot product of the parameters in the feature space, so that, each data point is mapped into a higher dimensional vector using some transformational techniques [22]. SVM is memory efficient too and effective in transforming high dimensional space image classification. It provides a higher precision-recall ratio and making it applicable to a large number of features in the image datasets.

Support vector machine mathematical model description:

$$x_2 \ = ax_1 + b$$

where a is the slope of line and b is any constant, therefore

The form above equation we get

$$ax_1 - \ x_2 + b = 0 \tag{1}$$

Vector Notation of the Eq. (1) maybe written as:

Let $\bar{x} = (\mathrm{x}_1,\ \mathrm{x}_2)^{\mathrm{T}}$ *and* $\ \bar{w} = (\ a - \ 1)$
Vector Form of Eq. (1) is given by:

$$\bar{w}.\bar{x} + b = 0 \tag{2}$$

The magnitude of vector w and x is given by

$$w = \frac{x_1}{\| x \|} + \frac{x_2}{\| x \|} \tag{3}$$

The cartesian form of the magnitude vector is given by the norm.

$$\| x \| = \ \sqrt{x_1^2 + x_2^2 + \ x_3^2 \ldots \ \ldots \ \ldots .x_n^2}$$

As we know the inner product returns the cosine of the angle between 2 vectors of unit length.

$$\cos(\theta) = \frac{x_1}{\| x \|} \text{ and}$$

$$\cos(\alpha) = \frac{x_2}{\| x \|}$$

From Eq. (3) we have

$$\begin{aligned} & w = (\cos\theta,\ \cos\alpha) \\ & \bar{w}.\bar{x} \ = \| w \| \| x \| \, cos\theta \\ & \cos\theta = \cos\beta - \cos\alpha \\ & \cos\theta = \cos(\beta - \alpha) \\ & \theta = (\beta - \alpha) \\ & \cos(\theta) = \cos(\beta)\cos(\alpha) + \sin(\beta)\sin(\alpha) \end{aligned} \tag{4}$$

We know that x (x_1, x_2) and w (w_1, w_2) are two points on the xy-plane.

Form the above equation we may get:

$$cos\theta = \frac{w_1}{\parallel w \parallel} \cdot \frac{x_1}{\parallel x \parallel} + \frac{w_2}{\parallel w \parallel} \cdot \frac{x_2}{\parallel x \parallel}$$

$$cos\theta = \frac{w_1x_1 + w_2x_2}{\parallel w \parallel\parallel x \parallel}$$

Putting the value of cos (θ) in Eq. (4) and the evaluated term is given by

$$\bar{w}.\bar{x} = \parallel w \parallel\parallel x \parallel . \frac{(w_1x_1 + w_2x_2)}{\parallel w \parallel\parallel x \parallel}$$

$$\bar{w}.\bar{x} = (w_1x_1 + w_2x_2 + \cdots w_nx_n)$$

$$\bar{w}.\bar{x} = \sum\nolimits_{i=1}^{n} w_ix_i$$

Let the fitness function of slop can be computed for n-dimensional vectors is given by:

$$f_{(i)} = y\ (\bar{w}.\bar{x} + b)$$

The minimum value of classification is either 0 or 1, only two possibilities are there, i.e., *if signature* $f_{(i)} > 0$, for correct classification otherwise incorrectly classified.

For training the whole dataset D we have to compute (multiple inputs, labels) for training a dataset, such that it is given:

$$f_{(i)} = y_{(i)}(w.x + b)$$

$$F = min_{(i=1,2,3...m)} f_{(i)}$$

To compute the functional optimal margin (F) value of dataset it is evaluated by the Langrangian Multiplier Method for weight optimization. Our objective is to find an optimal hyperplane that we can get after optimizing the weight vector $\bar{w}$ and the bias vector $\bar{b}$ as well given by in our case.

$$\mathcal{L}(w,b,\alpha) = \frac{1}{2}w.w - \sum_{i=1}^{m} \alpha_i \left[y_{(i)}\left(w.x_{(i)} + b\right) - 1\right]$$

We expand the last equation w.r.t. of $\overrightarrow{w}$ and b to the following form.

$$\nabla_w\mathcal{L}(w,b,\alpha) = w - \sum\nolimits_{i=1}^{m} \alpha_{(i)}y_{(i)}x_{(i)} = 0 \tag{5}$$

$$\nabla_b\mathcal{L}(w,b,\alpha) = -\sum\nolimits_{i=1}^{m} \alpha_{(i)}y_{(i)}x_{(i)} = 0 \tag{6}$$

$$w = \sum\nolimits_{i=1}^{m} \alpha_{(i)}y_{(i)}x_{(i)} \quad \& \quad b = \sum\nolimits_{i=1}^{m} \alpha_{(i)}y_{(i)} \tag{7}$$

where

$$\alpha_{(i)}y_{(i)} = \alpha_{(1)}y_{(1)} + \alpha_{(2)}y_{(2)} + \alpha_{(3)}y_{(3)} \ldots\ \ldots\ldots\ldots\ldots .\alpha_{(m)}y_{(m)}$$

After substitute the Langrangian Function $\mathcal{L}$ we get

$$w(\alpha, b) = \sum\nolimits_{i=1}^{m} \alpha_{(i)} - \frac{1}{2}\sum_{i=1}^{m}\sum\nolimits_{i=1}^{m} \alpha_{(i)}\alpha_{(j)}y_{(i)}y_{(j)}x_{(i)}x_{(j)} \tag{8}$$

Thus

$$max_{\alpha} \sum\nolimits_{i=1}^{m} \alpha_{(i)} - \frac{1}{2}\sum_{i=1}^{m}\sum\nolimits_{i=1}^{m} \alpha_{(i)}\alpha_{(j)}y_{(i)}y_{(j)}x_{(i)}x_{(j)}$$

Subject to

$$\alpha_i \geq 0,\ where\ i = 1, 2, 3 \ldots..m,\ \sum\nolimits_{i=1}^{m} x_{(i)}y_{(i)} = 0$$

Because of the constraints, we have inequalities, by putting for b and $\overrightarrow{w}$ back in the new equation we may get rid of the dependence on b and $\overrightarrow{w}$. The non-zero of the α's we extend the Langrangian Multipliers by Karush–Kuhn–Tucker (KKT) theorem, this theorem is applying to find the inner products of $x_{\ (i)}x_{\ (j)}$. The complementary condition of KKT results is as under.

$$\alpha_i[y_i(w_i x^{\star} + b] - 1 = 0 \tag{9}$$

However, input and output weights are updated at the optimal point $x^{\star}$. Most of the weights w_i will be zeros and will be nonzero only the support vectors (on the gutters or margin).

$$w = \sum_{i=1}^{m} \alpha_{(i)}y_{(i)}x_{(i)} = 0 \tag{10}$$

$$y_i\left(w_{(i)}.x^{\star} + b\right) - 1 = 0 \tag{11}$$

Multiple Eq. (11) by 'y' on both sides we get.

$$y_i^2\left(\left(w_{(i)}.x^{\star} + b\right)\right) - y_{(i)}\right) = 0 \tag{12}$$

$$y_i^2 = 0, \quad and\ \left(w_{(i)}.x^{\star} + b\right) - y_{(i)} = 0$$

$$y_{(i)} - w_{(i)}.x^{\star} = b \tag{13}$$

$$b = \frac{1}{S}\sum_{i=1}^{S}(y_{(i)} - w_{(i)}.x) \tag{14}$$

where, it is known as a support vector, which is the closest point to the hyperplane given by Eq. (14). The hypothesis function is given by formula

$$h_{\left(w_{(i)}\right)} = \left\{\begin{array}{ll} C_{(1)}(+1) & if\ w_{(i)}.x + b\ \geq 0 \\ & \quad . \\ C_{(2)}(-1) & if\ w_{(i)}.x + b\ < 0 \end{array}\right\}$$

The above point on the hyperplane will be classified as class +1 (blank space found) and the point below the hyperplane will be classified as −1 (no space available). This function will exploit if we give nonzero values of a's that correspond to the support vectors i.e., in fixing the maximum margin width on those that make all the a's positive.

4.3 Backbone Inception Network Model

As in the current problem we have a small number of classes that are being recognized from the input video stream. We don't need an extensively large neural network architecture instead of having a pretrained transfer-learning inception model describe by Raj [23]. In this article we used RestNet-50 as our backbone network which provide good results on the publicly available Pk-lot parking dataset. RestNet-50 is an architecture in which there is a residual connection between the layers. These are known as short connections. In the inception v-2 module, there is a problem, that parallel feature extraction requires a lot of computational cost power. In the inception v-3 module applied the factorization on the large size kernels. They factorize the n × n convolution into the 1 × n and n × 1 convolution.

The mathematical form for residual connections is illustrated in Eq. (15). Where W is weight matrix while G represents convolution function and X is the input feature map.

$$F = G(X, \{Wi\}) + X \tag{15}$$

The visual depiction for the RestNet block diagram is shown in Fig. 4 below.

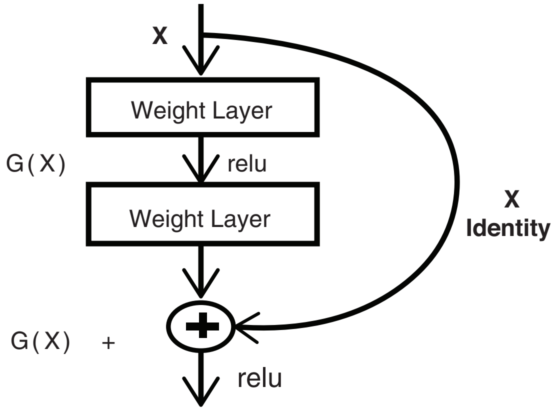


Figure 4: Residual RestNet block diagram taken from internet

The descendant features of the instant image frame are locally represented by X and the processed computed features develop global features G(x). To check the localization of the vehicle concerning the landmark location within the frame on each motion and measurement-updates, a new posterior distribution can be established which provides its calculation. The convergence factor helps in establishing the correct loop closure that will function as the localization of the vehicle towards the correct prediction by Ravankar et al. [24].

5 Simulation and Experimental Analysis

In our proposed architecture, we have utilized the transfer-learning techniques on inception v-3 module which is pertained model to ImageNet 1000 classes [25]. After the inception modules, we get the 2048 features. After that, we have built our linear classifier, which contains the three fully connected layers. The first fully connected layer maps the 2048 features to 1024 neurons. In the second fully connected layer, 1024 features are mapped to the 512 features. Whereas in the last fully connected layers these features are mapped to the 256 neurons by giving the probability of our two classes which is free space and allocated space in our case. For the training process, the model used 25000 vehicles and 20000 non-vehicles manually labeled sample images using Pk-lot open dataset [26].

In this proposed study, we utilized mask-RCNN along with the inception deep learning module for vehicle count and vacant parking space detection. Initially, the visual frame is captured from the input parking video feed. Once the frame is cropped, they passed to the inception network for counting the total number of vehicles present in frame of parking space with occupancy detection. The inception module is responsible for binary class prediction whether the slot is occupied or empty. Once the inception module returns result in the color of the rectangle represents the occupancy status (green means empty and red means occupied) as shown in video frames below. Other than occupancy detection, the proposed system is capable of detecting, labeling and instance segmenting out the vehicles on-road or anywhere in the video stream. The splitting of a dataset is 80% for training data and 20% validation data of sample dataset. The detection rate of the classifier was about 97.6% for input video feed of parking vehicles initially. On noisy images, Fusek et al. achieved 78% for positive samples prediction and 94% for negative ones proposed by Fusek et al. [27] while the SVM detector algorithm achieved 97% and respectively 97.6% to our proposed model accuracy.

5.1 System Setup

The artificially produced dataset which contains frames from dynamic video-streams. All training and testing were carried out on NVIDIA 1080 Ti GPU with 11 GB of memory, Intel Core i7 with a 64-bit operating system (CPU i7 (64)). The computational cost was measured for the projected model and sequential methods for the parking video dataset.

5.2 Convolutional Neural Network (CNN) Technologies

Firstly, based on neural network architecture such as CNNs that extract the dimensional points from an image or video feed, from low-level features, such as lines, edges, ROI, data segments, or circles to higher-level features of vehicles parts, persons, motorbike, etc. A few well-known base-neural networks models are LeNet, InceptionNet (aka. GoogleNet), RestNet, VGG-Net, AlexNet, and MobileNet, etc. Then secondly, pretrained perception, planning neural network model is attached to the end of a base neural network model that used to concurrently identify multi-class objects from a single frame or image with the help of the base high-level extracted features. After selecting the ROIs, it does the classification and regression task on them. Regression for precisely bound the ROI on the object and classification for prediction if it is an object. The detail of mask-RCNN is given below.

5.3 Mask-RCNN (Region-Based Convolutional Neural Network)

Deep learning algorithms are being used widely from classification task to, object detection and instance segmentation task due to their high reported accuracy. Detection of objects present in an image or sequence of images can be done using R-CNN, Fast-RCNN, or Faster-RCNN algorithms. The problem of segmentation is more advanced than the object detection, which cannot be alone done using only detection technique. It requires a pixel-level classification to make a segmentation mask around the detected object, which is very tough to classify individual objects and localize each using a bounding box. But the beauty of deep convolutional neural network is that they learn to perform this task. The problem of instance segmentation can be performed using mask-RCNN, which is a deep learning-based approach. The backbone architecture of mask-RCNN is similar to faster R-CNN. It performs region proposal task using region proposal network (RPN). After that, each region is classified as an object or instance background. The background regions are discarded and an object containing regions is further passed to a classifier network for recognizing the particular object class. After that, the detected regions are passed to a fully convolution neural network which draws the segmentation mask around the objects.

5.4 Proposed Fully-Multitask Convolutional Neural Network (F-MTCNN) Model

After implementing multi-task mask-RCNN the extracted masked regions are passed to the fully multi-task convolutional network model (F-MTCNN). Nevertheless, before passing to the F-MTCNN, the bounding boxes are drawn with the multi-parameters using dimensionality reduction values by using the ROI-alignment augment method. F-MTCNN is a simple deep convolutional neural network of classification layers. It outperforms the segmentation task on extracted ROIs on the selected parking frames. It draws the segmentation mask around the predicted vehicles present in the ROIs area as shown in images given below. We see in Tab. 1 multi-layer convolution network with nine inception layers that are proposed in this architecture which is based on the inception-v3 and mask-RCNN module. The first two convolution layers followed by the max-pooling hidden layers. On the input the frame size is 299 × 299 × 3, we have a batch size of 128, that have 64 filters of 112 × 112, and in the second layer 64 filters of 56 × 56 size have been applied followed by the max-pooling layers respectively as shown in Tab. 1 below.

Table 1: Deep convolutional neural network and pooling layers description

Layer-name	Input size	Output size
Conv	299 × 299 × 3	112 × 112 × 64
MaxPool	112 × 112 × 64	56 × 56 × 64
Conv	56 × 56 × 64	56 × 56 × 192
MaxPool	56 × 56 × 192	28 × 28 × 192
Inception-3A	28 × 28 × 192	28 × 28 × 256
Inception-3B	28 × 28 × 256	28 × 28 × 480
MaxPool	28 × 28 × 480	14 × 14 × 256
Inception-4A	14 × 14 × 256	14 × 14 × 512
Inception-4B	14 × 14 × 512	14 × 14 × 512
Inception-4C	14 × 14 × 512	14 × 14 × 512
Inception-4D	14 × 14 × 512	14 × 14 × 528
Inception-4E	14 × 14 × 528	14 × 14 × 832
MaxPool	14 × 14 × 832	7 × 7 × 832
Inception-5A	7 × 7 × 832	7 × 7 × 832
Inception-5B	7 × 7 × 832	7 × 7 × 1024
AvgPool	7 × 7 × 1024	1 × 1 × 1024
Dropout (0.5)		
Dense-1 (Fully connected)	1 × 1 × 124	1024
Dense-2 (Fully connected)	1024	512
Dense-3 (Soft-max)	512	2

After in the first inception module, there are two layers in which we applied 256 and 480 filters that have been applied on the 28 × 28 image size followed by the max-pooling layer. Whereas in the second inception module there are 4 inception modules of 512, and 528 filters with the 14 × 14 kernel size. The second inception module also followed by the max-pooling layer. Whereas, in the last inception module 832 and 1024 filters have been applied with the kernel size of 7 × 7. After that, we have built the linear network which contains three fully connected layers. In the first fully connected layer, we obtained 1024 features. These features are mapped to the 512 features in the second fully connected layer, whereas in the last fully connected layer these 512 features are mapped to the second layers which give the probability against our defined classes in the algorithm.

5.5 Evaluation Criterion

The goal of this article is to introduce a video stream of open parking area with more frame per seconds for perfect classification scheme on the frames received from configured multi-sensor parking surveillance videos-cameras. The evaluation parameters are, let TP represent true positive, FP denotes false positive and FN represents false negative. To multi-sensory feeds the evaluation performance of the new large dataset CNN baseline model. The evaluation of our approach for vehicle parking segmentation and classification using the following 6 statistical metrics are given below.

$$\text{Precision of each category} \quad Pre_i = \frac{TP_i}{TP_i + FP_i} \tag{16}$$

$$\text{Recall of each category} \quad Rec_i = \frac{TP_i}{TP_i + FN_i} \tag{17}$$

$$\text{Accuracy of each category} \quad \mathrm{ACC} = \frac{TP}{\#\ of\ Testing\ Images} \tag{18}$$

$$\text{Mean Recall of each category} \quad mPre = mean(Rec_i) \tag{19}$$

$$\text{Mean Precision of each category} \quad mPre = mean\ (Pre_i) \tag{20}$$

$$\text{F-measure of each category} \quad F_{\beta=1} = \frac{2(\ Pre *Rec\ )}{Pre + Rec} \tag{21}$$

Here the model employed dynamic learning rate with Adam-optimizer and dropout the neurons that leading to over-fitting of training data. The starting learning rate was 0.001. After 100 epochs it changed into 0.0001 and after 250 epochs that rate become 0.00001 to converge optimally close to the target output lastly. We train our model up to 4000 epochs to optimize the system until the desired accuracy is achieved.

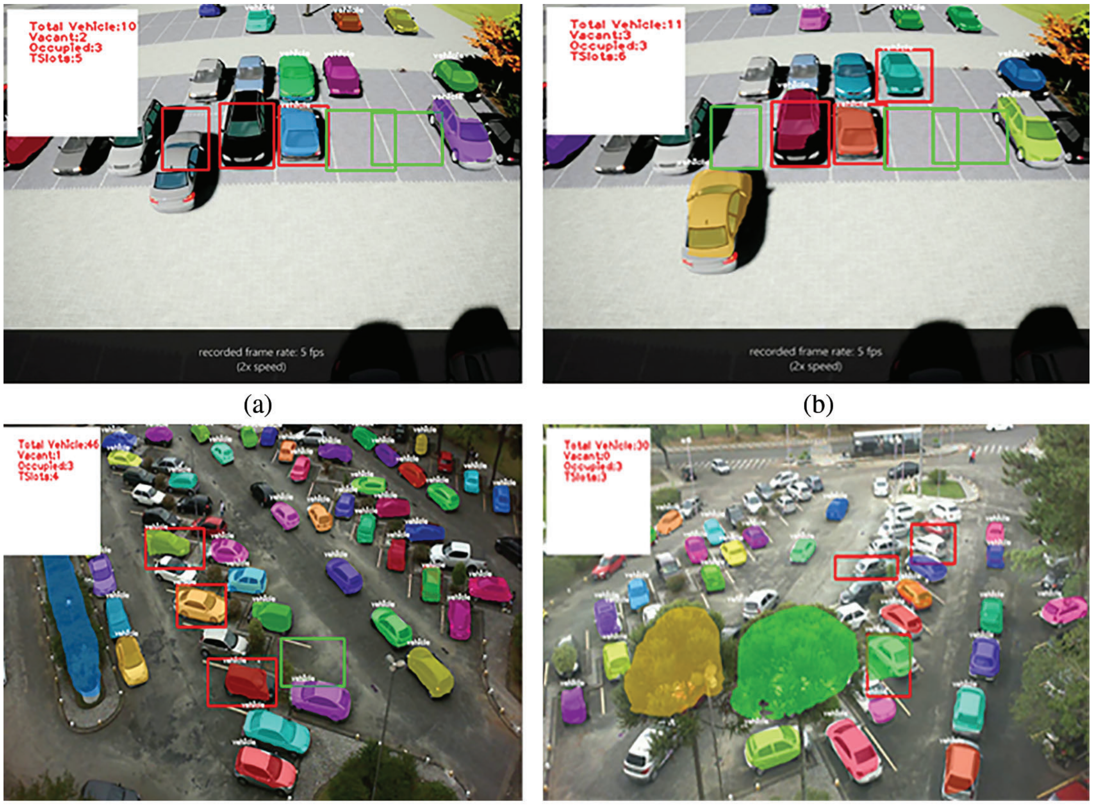
6 Experimental Results and Discussion

It could be better that the performance depends on the utilization of highly powerful GPUs, TPUs, and high-resolution graphic card systems. The experimental results show that the powerful computational resources looking for distribution of processes, refining the proposed model described by Böhm et al. [28].

There also need to add efficient transfer-learning models with the proposed working model where integration of key parameters is using articulate-set of data structure for a multi-core GPUs to significantly increased computational performance using powerful devices.

The comparison of experimental results with earlier related reported work are presented here as follows.

Tab. 2 shows the evaluating of autonomous parking-lots and vehicle detection in parking vehicles video frames as shown in Fig. 5 for our proposed F-MTCNN model with previous methods proposed by Fabian [29] also with Amato et al. [30]. The number of epochs is the number of complete passes through the training dataset, by seeing the factors of training and validation accuracy and miss rate metrics used to measure and compare with the proposed model. At the start of testing the error rate is 0.7% at initial epochs gradually error rate decreases to 0.04% at final epochs as shown in Fig. 6b below. Accuracy & miss rate metrics are compared to the previous work evaluation with the proposed F-MTCNN system model. As the proposed F-MTCNN system model gives the best results as compared with others 97.6%, 96.6% accuracy, 2.40%, 3.40% miss rate during training & validation respectively.



(a) (b)

(c) (d)

Figure 5: (a) and (b) Example of video frames with description parked vehicles and vacant spaces. (c) and (d) Sample video frames with description of parked vehicle's and vacant spaces count

Table 2: Comparison the performance of proposed F-MTCNN model with approaches in the literature

Literature	Training		Validation	
	Accuracy (%)	Miss rate (%)	Accuracy (%)	Miss rate (%)
Fabian (2013) [29]	96.40	3.60	96.2	3.80
Amato et al. (2018) [30]	96.36	3.64	96.1	3.90
Proposed system model	97.60	2.40	96.6	3.40

The above Fig. 6 representing the learning curve on graph 6(a) and 6(b) which illustrate the training and validation accuracy and losses on the inception model. We have trained our model until 4000 epochs. We have set the learning rate for every 100 epochs. For the first 100 epochs, the learning rate was 10^{-1}, and it was decreased with −1 power after 100 epochs. The weights have been updated using the stochastic gradient descent function and mean square error function. Initially, the training accuracy was 6.8% and its validation accuracy was 3.4 and after an increasing number of epochs, its accuracy gradually increased.

Finally, the number of epochs increases up to 4000 numbers training accuracy reaches to 97.6% and validation accuracy is 96.6%. Our training and validation accuracy increase gradually which shows that our model does not go towards the overfitting.

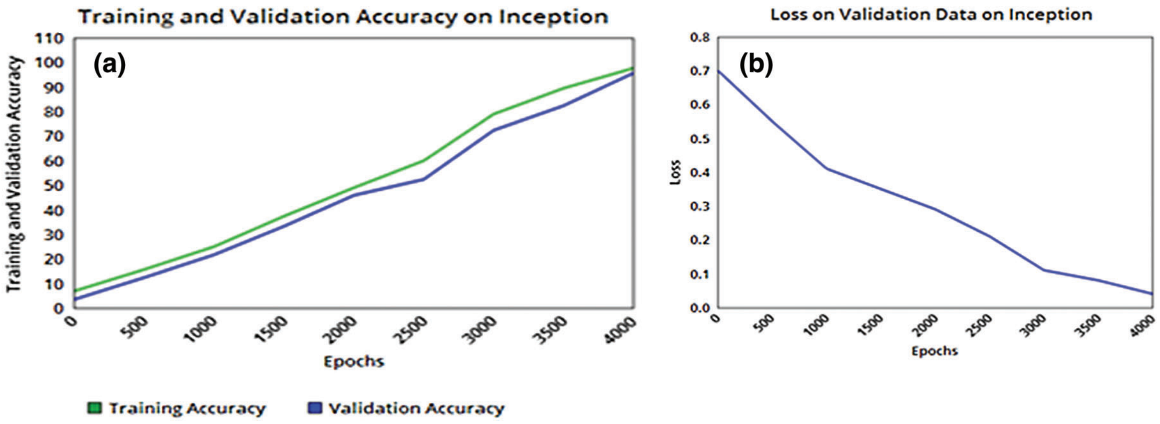


Figure 6: Learning curve (a) Training and validation accuracy, (b) validation loss on inception model

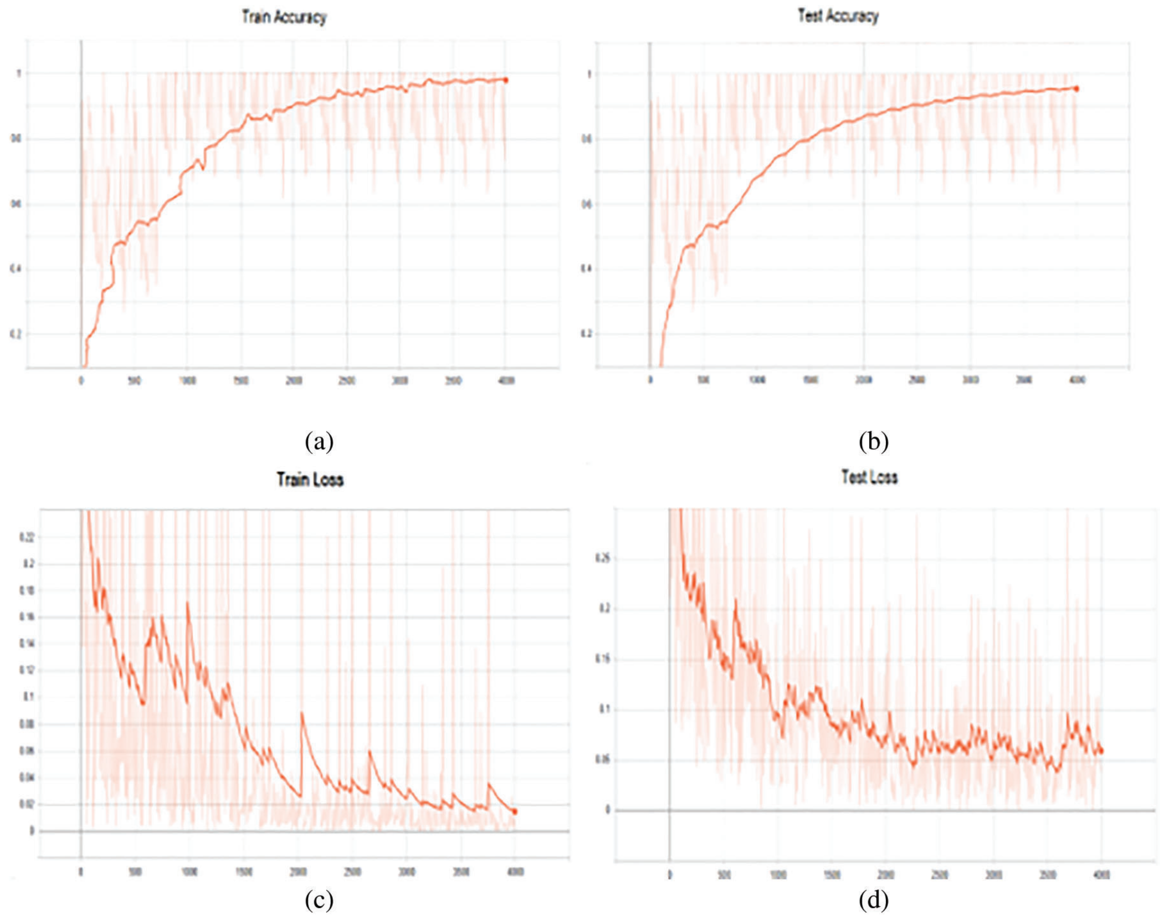
Fig. 6b depicts the loss value on the validation data. At the start of training, the loss was 0.7, but as the training continues, it decreased with time due to our good selection of training hyperparameter standardization rules and utilizing dropout functions. We have computed the loss error rate which was based on taking the difference by the predicted and the actual value.

The reason of stopping the training at 4000 epochs is due to the loss value becomes constant after 4000 epochs, so the best possible accuracy which is achieved on the training and validation dataset is shown in Fig. 7a–7d below respectively.

The Fig. 7 below illustrates learning curve in 7(a) & graph 7(c) illustrate the performance on training, and error rate or training loss, further graph 7(b), and graph 7(d) shows testing accuracy and testing loss during training and testing time on mask-RCNN model. There was a total of 4000th epochs to train the system to its best performance. As we analyze, there was no improvement or relatively less improvement in loss and accuracy, so, we stopped our training at 4000th epochs for obtaining optimum results.

In training accuracy as depicted in Fig. 7a there is an abrupt increase in accuracy learning curve which gradually becomes relatively flatten with smaller changes or we can say with no change. The top training accuracy achieved is 97.60% at the 4000th epoch. The test accuracy curve results as shown in Fig. 7b as in graph 7(a) follows different trends. There is a less change up to the 700th epoch with several fluctuations. After that, it started to increase by following some trend with lower slope value and finally become almost flatten from the 3700th epoch. The test accuracy achieved in the proposed model is 96.60% at 4000 epochs as illustrated in above graphs.

The loss rate of the trained model should be as minimized as much as possible. The loss curves in both testing loss and training loss graph as shown in Fig. 7 of graph 7(c) and graph 7(d) respectively are not as smooth as accuracy graphs' curves. These loss curves trend is quite ambiguous with a lot of curvy ups and downs. In both loss graphs, the loss is decreasing as a whole but there are untrendy ups in both graphs as you can see in the training loss graph there is an abrupt increase in the loss curve at 500, 1000 and 2000 epoch but finally, it converges to 0.04% at 4000th epochs. The test loss finally converges to 0.07% at final 4000th epochs.



(a) (b)

(c) (d)

Figure 7: (a) Training-accuracy on mask-RCNN, (b) Testing-accuracy on mask RCNN, (c) Training-loss on mask-RCNN model, (d) Testing-loss on mask-RCNN

7 Conclusion and Future Work

This article proposed an ordered autonomous parking space detection system by providing visual input data to count empty vehicles parking spots and parked. Deep learning algorithms are showing increasing attention to own the growth of connected traffic data. Automated vehicles new functionality is advancing at a rapid pace virtually by all major auto concerns. The sheer number of sensors, the complexity of onboard diagnostic systems and decision-making systems are integrated with real traffic data analytics to disseminate information to solve user everyday needs. In this article, we proposed a deep convolutional neural network model F-MTCNN for parking spot detection, but not the last at least. The analysis results showed that the proposed multi-model system performs relatively well and attained accuracy 97.6%. Overall, the system is investigated a lot about how the mask-RCNN and inception CNN model in different video feeds to attain reasonable results and minimized error losses. Furthermore, we are developing more multi-model key features extracting algorithms for high training and testing accuracy performance. The possibilities are endless in terms of how the CNN technologies that would be applied and exciting to think about how to give our machines the “ability to see & talk” and help us to make the world better. Our future work includes incorporating deep knowledge about human behaviors, mobility, and connected vehicular technologies in multi-model object classification and detection.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] WHO, "Injury facts, motor vehicles safety issues," 2020. [Online]. Available: https://injuryfacts.nsc.org/motor-vehicle/motorvehicle-safety-issues/ (accessed 29 June 2020).

[2] S. Singh, 2018. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries (accessed 29 June 2020).

[3] N. E. Faouzi, H. Leung and A. Kurian, "Data fusion in intelligent transportation systems: Progress and challenges —A survey," *Information Fusion*, vol. 12, no. 1, pp. 4–10, 2011.

[4] K. Iqbal, M. A. Khan, S. Abbas, Z. Hasan and A. Fatima, "Intelligent transportation system (ITS) for smart-cities using mamdani fuzzy inference system," *International Journal of Advance Computer Science and Applications*, vol. 9, no. 2, pp. 94–105, 2018.

[5] H. Geng, *Internet of Things and data analytics, handbook*. 1st Edition. Palo Alto, CA, USA: John Wiley & Sons Amica Research, pp. 409–425, 2017.

[6] S. T. Shivappa, M. M. Trivedi and B. D. Rao, "Audiovisual information fusion in human-computer interfaces and intelligent environments: A survey," *Proceedings of the IEEE*, vol. 98, no. 10, pp. 1692–1715, 2010.

[7] R. Kitchin, "Big data new epistemologies and paradigm shifts," *Big Data & Society*, vol. 1, no. 1, pp. 2053951714528481, 2014.

[8] D. Acharya, W. Yan and K. Khoshelham, "Real-time image-based parking occupancy detection using deep learning," in *Proc. of the 5th Annual Research@Locate Conf.*, Adelaide, Australia, pp. 33–40, 2018.

[9] 2020. [Online]. Available: https://medium.com/@giacaglia/self-driving-cars-f921d75f46c7.

[10] S. Kubler, J. Robert, A. Hefnawy, C. Cherifi, A. Bouras *et al.,* "IoT-based smart parking system for sporting event management," in *Proc. of the 13th Int. Conf. on Mobile and Ubiquitous Systems, Computing, Networking and Services*, Hiroshima, Japan, pp. 104–114, 2016.

[11] J. Krompier, "Safety first: The case for mandatory data sharing as federal safety standard for self-driving cars," *University of Illinois Journal of Law, Technology & Policy*, vol. 2017, pp. 439, 2017.

[12] H. S. Seo, B. C. Kim, P. S. Park, C. D. Lee and S. S. Lee, "Design and implementation of a UPnP-can gateway for automotive environments," *International Journal of Automotive Technology*, vol. 14, no. 1, pp. 91–99, 2013.

[13] W. Chu, Y. Liu, C. Shen, D. Cai and X. S. Hua, "Multi-task vehicle detection with region-of-interest voting," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 432–441, 2017.

[14] W. Liu, M. Zhang, Z. Luo and Y. Cai, "An ensemble deep learning method for vehicle type classification on visual traffic surveillance sensors," *IEEE Access*, vol. 5, pp. 24417–24425, 2017.

[15] X. Luo, R. Shen, J. Hu, J. Deng, L. Hu *et al.,* "A deep convolution neural network model for vehicle recognition and face recognition," *Procedia Computer Science*, vol. 107, pp. 715–720, 2017.

[16] S. Kamada and T. Ichimura, "Knowledge extracted from recurrent deep belief network for real time deterministic control," in *2017 IEEE Int. Conf. on Systems, Man, and Cybernetics, IEEE*, North America, pp. 825–830, 2017.

[17] S. M. Grigorescu, D. R. Durrant and A. Gräser, "ROVIS: Robust machine vision for service robotic system," in *2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IEEE*, pp. 3574–3581, 2009.

[18] Frost and Sullivan, "A safety-first approach to developing and marketing driver assistance technology," 2019. [Online]. Available: https://lifesaversconference.org/wp-content/uploads/2019/03/Frykman-ID-06.pdf, (accessed 29 June 2020).

[19] C. Thompson, J. White, B. Dougherty, A. Albright and D. C. Schmidt, "Using smartphones to detect car accidents and provide situational awareness to emergency responders," in *Int. Conf. on Mobile Wireless Middleware, Operating Systems, and Applications*, Berlin, Heidelberg. Springer, pp. 29–42, 2010.

[20] T. N. Pham, M. F. Tsai, D. B. Nguyen, C. R. Dow and D. J. Deng, "A cloud-based smart-parking system based on Internet-of-Things technologies," *IEEE Access*, vol. 3, pp. 1581–1591, 2015.

[21] L. Cao, Q. Jiang, M. Cheng and C. Wang, "Robust vehicle detection by combining deep features with exemplar classification," *Neurocomputing*, vol. 215, pp. 225–231, 2016.

[22] Y. Ma, G. Gu, B. Yin, S. Qi, K. Chen *et al.,* "Support vector machines for the identification of real-time driving distraction using in-vehicle information systems," *Journal of Transportation Safety & Security*, pp. 1–24, 2020.

[23] B. Raj, "A simple Guide to the versions of the Inception network," 2018. [Online]. Available: https://towardsdatascience.

[24] A. A. Ravankar, Y. Kobayashi and T. Emaru, "Clustering based loop closure technique for 2D robot mapping based on ekf-slam," in *2013 7th Asia Modelling Sym.,* IEEE, NW Washington, DC, USA, pp. 72–77, 2013.

[25] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li *et al.,* "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conf. on Computer Vision and Pattern Recognition,* IEEE, Miami, Florida, pp. 248–255, 2009.

[26] P. R. Almeida, L. S. Oliveira, A. S. Britto Jr, E. J. Silva Jr and A. L. Koerich, "PKLot: A robust dataset for parking lot classification," *Expert Systems with Applications*, vol. 42, no. 11, pp. 4937–4949, 2015.

[27] R. Fusek, K. Mozdřeň, M. Šurkala and E. Sojka, "Adaboost for parking lot occupation detection," in *Proc. of the 8th Int. Conf. on Computer Recognition Systems CORES 2013*, Heidelberg: Springer, pp. 681–690, 2013.

[28] C. Böhm, R. Noll, C. Plant, B. Wackersreuther and A. Zherdin, "Data mining using graphics processing units, " in *Transactions on Large-Scale Data-and Knowledge-Centered Systems I.* Berlin, Heidelberg: Springer, pp. 63–90, 2009.

[29] T. Fabian, "Parking lot occupancy detection using computational fluid dynamics," in *Proc. of the 8th Int. Conf. on Computer Recognition Systems CORES*, Heidelberg: Springer, pp. 733–742, 2013.

[30] G. Amato, P. Bolettieri, D. Moroni, F. Carrara, L. Ciampi *et al.,* "A wireless smart camera network for parking monitoring," in *IEEE Globecom Workshops*, IEEE, Abu Dhabi, UAE, pp. 1–6, 2018.