RES MILITARIS
REVUE EUROPEENNE D ETUDES EUROPEAN JOURNAL OF MILITARY STUDIES
Social Science Journal

Building An Intelligent Floor Mapping Using 2-D Hand-Held Lidar System Based On Proposed Deep Learning Network

By

Manar Mohammed Raghib Ali
Iraqi Commission for Computers and Informatics Institute for Post Graduate Studies, Baghdad, Iraq
Email: manarmra248@gmail.com

Assist. Prof. Dr. Saad Hameed Abid
Head of computer engineering department, Al-Mansour University College, Baghdad, Iraq
Email: saad.hameed@muc.edu.iq

Abstract

Thanks to advancements in sensor technology, it is now possible to create wireless sensor networks that can be used for a variety of purposes, including locating objects, industry, and healthcare. These applications must continue as long as people want them to. One of the most important types of location navigation services is indoor mapping. Simultaneous localization and mapping (SLAM) is a crucial technique in an indoor navigation system using a low-cost 2-D LIDAR (Light Detection and Ranging) system from Slamtech. SLAM seeks to simultaneously produce a uniform, global map of the environment. In this paper, we propose the design of a system that executes two models concurrently and in real-time, which are considered crucial in recent years, especially in computer vision and artificial intelligence. The first one is to build a floor mapping based on simultaneous localization and mapping (SLAM). SLAM is often used to map indoor areas and localize handheld systems. The second one is for object detection based on the COCO dataset and using a convolutional neural network (CNN). These two models are implemented in the Raspberry Pi 4, which has less memory and processing capacity but is capable of handling massive quantities of data. Our system made the working of the RPLIDAR A1 smart because, in addition to its main work of mapping, it also added object detection.

Keywords: indoor mapping, PCA, CNN, RPLIDAR A1, MS COCO-dataset.

Introduction

In recent years, LIDAR scanning systems based on the concepts of simultaneous localization and mapping (SLAM) have gained acceptance as a way for collecting interior environment data and then building a map. The 2-D or 3-D LIDAR scanner sensors are developed into handheld or wearable systems. When employing handheld lidar, the process of generating point clouds is made to be simple to operate and provides feedback on the scan in real-time as it is being taken [1]. In this work, and in addition to build map, one of the most significant fields in computer vision which is object detection is added using a convolution neural network (CNN) algorithm. The use of CNN is widely, because, it is distinct from the other traditional neural networks [2]. The method of the two models working in parallel means the RPLIDAR have the responsibility of building an indoor mapping in real-time, and simultaneously the camera which was trained by CNN is used for object detection. There are two kinds of objects: static and dynamic. The static object is considered as a better, the reason beyond that RPLIDAR will do not need to scan the position more than once, and this will reduce from the power consumption. Following the presentation of this section, we will go on to the presentation of the remaining sections that will be covered in this article. In section II, we will talk about the studies done before, and in section III, we'll talk about how the proposed system works. Then, in the IV section, the two models of the proposed system will be shown in detail. After that, the devices that were used in the proposed system will be mentioned in the V section, and the CNN assessments will be discussed with in the VI section. Finally, the results will be discussed in the VII section.

Literature Review

In the literature survey of previous studies, we will highlight two groups of studies: the first one is related to object detection using the deep learning algorithm especially (CNN) that based on COCO-DATASET data. The second one is related to studies related to using the RPLIDAR A1 in the construction of floorplan mapping or for obstacle detection, which we will mention its description and architectures in the subsequent parts.

Object-Detection: in this paper [3], It was designed to identify basic, little things in small picture areas. CNN's propose area attributes have low resolution, less discrimination, and noise. Hence, improving the extraction feature from tiny items is necessary to improve their detection. Based on Faster R-CNN algorithm, context information may be merged without using the picture to frame recommended area windows and context information windows. The RPN generates region suggestions. Non-maximal suppression (NMS) reduces suggestions.

In object detection [4], the two tasks classification and localization of objects is performed. The research proposes a new object detection network based on the distinction between the two tasks (classification of and localization). The proposed deep learning network is divided into two sections. 1) Attention grid section, in which two interest maps are created for the two tasks.; 2) Layer separation section, with two feature maps, object classification and object localization are separately conducted. The suggested object detection network beat the most recent approaches, according to extensive experimental results based on the PASCAL VOC dataset and MS COCO dataset. In this paper, a topic-guided neural image captioning model is described that integrates a topic model into the CNN model. Each image is represented by the model as a set of topics, and each topic as a collection of words with relevant frequency distributions.

(Q.Wu et al.) [4] Are performed experiments using the Microsoft COCO dataset? Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) were merged. It is demonstrated that an explicit representation of image content enhances V2L performance in all circumstances, and that the RNN-based VQA technique can be extended to accommodate the huge quantities of information necessary to answer broad, open-ended inquiries about images.In the study (H.Caesar) [5], a wide - ranging CoCo-stuff dataset is presented, The study showed that the important stuff cover most of the surface of the image, and more than a third of the names in the human descriptions of the image, is not easy to divide the stuff in general from the things. The coco-stuff training process led to an improvement in the semantic segmentation performance.

Floorplan-mapping: (Z He et al.,) [6] are introduced an efficient method for generating furniture-free maps of interior buildings in real time (SLAM) Utilizing Mobile laser scanning based on simultaneous localization and mapping, The robot utilises a perpendicular pair of horizontal and vertical LIDAR sensors. The horizontal LIDAR forecasts the robot's location, while the vertical LIDAR creates a map devoid of objects. Using the RANSAC method, the largest plane with an orthogonal surface normal to the ground is determined. Three phases comprise the process: point cloud rearrangement, wall plane identification, and semantic classification. This investigation achieved 99.60% accuracy in maintaining the wall in the 3-Dimenssion output, demonstrating that the algorithm is able to eliminate the bulk of the environment's furnishings. In addition, the use of 2-Dimenssion furniture-free mapping for room segmentation is presented.

While (T R Madhavan, M Adharsh) [7] are introduced an approach for collecting data using RPLIDAR A1 with two algorithms which are: obstacle detection algorithm which using filtering , preprocessing, and clustering to the data. The other algorithm is obstacle-avoiding algorithm, if there are no obstacles, the optimal direction is the reference point-destination line, and the controller receives the destination point from the raspberry pi.

The study of (Mustafa ZEYBEK) [1] a semi-automatic approach for constructing indoor mapping processing using point clouds collected with GeoSLAM ZEB-REVO has been developed. The primary purpose is to extract the walls precisely. Consequently, the error rates of the areas estimated using only the building facade planes were assessed throughout the performance evaluation. In order to test the accuracy potential of the handheld LiDAR using manual reference data, the noise components were eliminated from the point clouds. False points originating from different surfaces must be deleted from the ZEB-REVO point cloud using a statistical method.

A method for producing indoor mapping by merging point clouds from several LIDARs that scan the identical area from different points. was proposed by (Hikaru Yoshisada* et al.,) [8]. Sets of "line segments" and "edge points" were utilised as the features of LIDAR point clouds in the method because to their spatial shape characteristics. In addition, a new method for detecting a set of line segments based on random sample consensus (RANSAC) and the Hough transform was developed.

The objective of the paper of (M. Royhan Iqbal et al.) [9] is to provide a detailed interior mapping and establish localization approach. The purpose of this study is to enhance the robot's capability by employing corner detection and the triangle's Cosine Law to precisely detect furniture so that the robot is aware of its position. On the basis of the LIDAR data, calculations are done to derive the corner features, which comprise (convex, concave, and flat surfaces). Experiments suggest that the robot can detect furniture at a distance of up to 5 metres with an inaccuracy of around 12.5% for the measured angle and 6.12% for the measured distance. The robot's rate of escape from the furnished room is 94.7%.

Methodology

The proposed method involves building a system composite of Software and Hardware. Software part has two models: The first model is an object detection model that has been applied to COCO dataset, Based on the CNN algorithm. The second one is a model for building a smart mapping based on SLAM.

The Hardware part consists of (RPLIDAR A1, and camera), all of these components connected to raspberry pi 4. The RPLIDAR’s mission is to scan the position horizontally and generate point clouds. The points generated by LIDAR will include the obstacle's distance, the angle at which the LIDAR scanned the location. As for the distance sensor, it is used for measuring the distance vertically. The camera will have the function of detecting objects, and it works in parallel with the laser, meaning if a specific object is scanned, the camera determines what the object is. The two models are implemented in microprocessor (Raspberry pi 4).

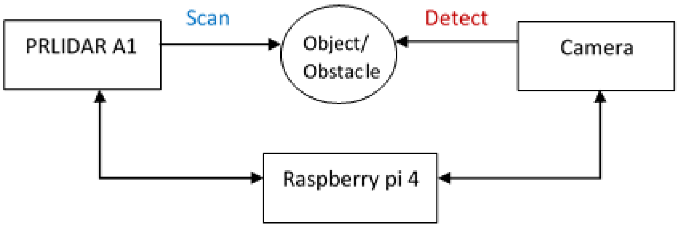

Figure 1 *the work of LIDAR and camera in parallel*

Software

Object Detection Model

In the below flow-chart which illustrate in figure (2), each phase will be covered briefly starting from acquisition of the dataset type used. Next, the data set will be entered into stages of image preprocessing. Then, the feature extraction will be explained and the PCA algorithm used, after that, the extracted features will be entered to the CNN classifier to be trained. Finally, the model will be ready.

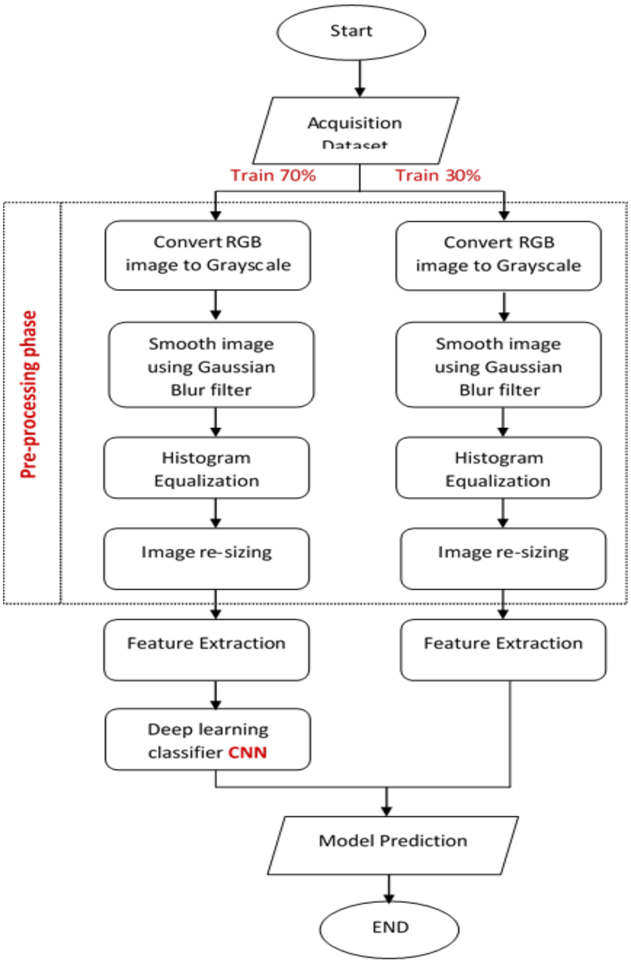

Figure 2 *proposed project flow-chart*

Acquisition Dataset

The MS COCO 2017 dataset contains 80 object classes with a total of 123K labeled images splitting to 118k for training and 5k for validation and 20k for testing without provided annotations. On the COCO training set, detectors were trained, and on the validation set, they were tested.

Image Preprocessing

Preprocessing is the stages that are taken to prepare the images into formats to be ready used by the model [10]. The stages are: convert image to grayscale, histogram equalization, image blurring using Gaussian blur, and image resizing.

Image conversion to gray scale

This operation is simply convert the colored image into image of balck and white and in-between (grayscale). Grayscale image contains only 8 bits/pixel data, means each pixel in an image has a value ranging from (0 - Black) to (255 -white) and the in-between values have the grayscale colors. The importance of this stage in pre-processing is get rid of unwanted information from the color image (RGB), and also uses one-matrix instead of three-matrix [11][12]. The figure (3-B) illustrates the original image to Grayscale. The equation (1) [13] is used to convert the colored image to gray scale:

$$\text{grayscale\_img} = ((0.3 \times \text{R}) + (0.59 \times \text{G}) + (0.11 \times \text{B})) \qquad (1)$$

Gaussian Blur

Blurring is the process of simply blurring an image to make it smooth with no visible edges, as well as eliminating image noise. The method that is using to blur the image, each pixel of the original image will be changed by using blurring techniques and convolution. Kernel will be applied to each pixel in an image, and the resulting value will be considered as the new value of the pixel. The choosing how much we should smooth the image is based on the size of kernel, it is usually take the sizes (3×3) [14]. The figure (3-C) illustrates the grayscale image to Gaussian Blur with its histogram. When apply Gaussian function in 1D, the equation (2) is:

$$\text{G}(\text{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \text{e}^{-}\frac{\text{x}^2}{2\sigma^2} \qquad (2)$$

Where σ represents the standard deviation of distribution. The 2-D Gaussian function need to be utilised while working with images.This is simply the product of two 1D Gaussian functions and is given by equation (3):

$$\text{G}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \qquad (3)$$

Histogram Equalization

The histogram equalization is another stage in image pre-processing. The function of it is to enhance the contrast of the blurry image. Histogram equalization is done by re-distribution of the most frequently pixels, means stretching out the intensity range of the image, this allows to the regions that have less local contrast to get a higher contrast [15][16]. To achieve the histogram equalization, firstly we calculate the cumulative histogram distribution (CDF) values (x_i) as shown in equation (4) below:

$$y_i = T(x_i) = \sum_{i=0}^{i} P(x_i) = cdf(x_i) \qquad (4)$$

Next, normalize the values, by dividing the resulted values from the previous step by the total pixels as shown in the equation (5) below:

$$P(Xi) = \frac{n_i}{N}, N = \sum_{k=0}^{L-1} n_k \quad (5)$$

Where, n_i represents the frequently number of each pixel, N indicates the overall pixel count, and L represents the levels number of brightness.

After that, multiply the previous results with the gray level (L-1) and round it using the equation (6) below. This step applies to each value separately.

$$T_i = Round\left(\left(\frac{cdf(x) - cdf(x)_{min}}{C \times R - cdf(x)_{min}}\right) * L - 1\right) \quad (6)$$

Where, C×R represent the column and row respectively. The figure (3-D) illustrates the histogram equalization and its histogram.

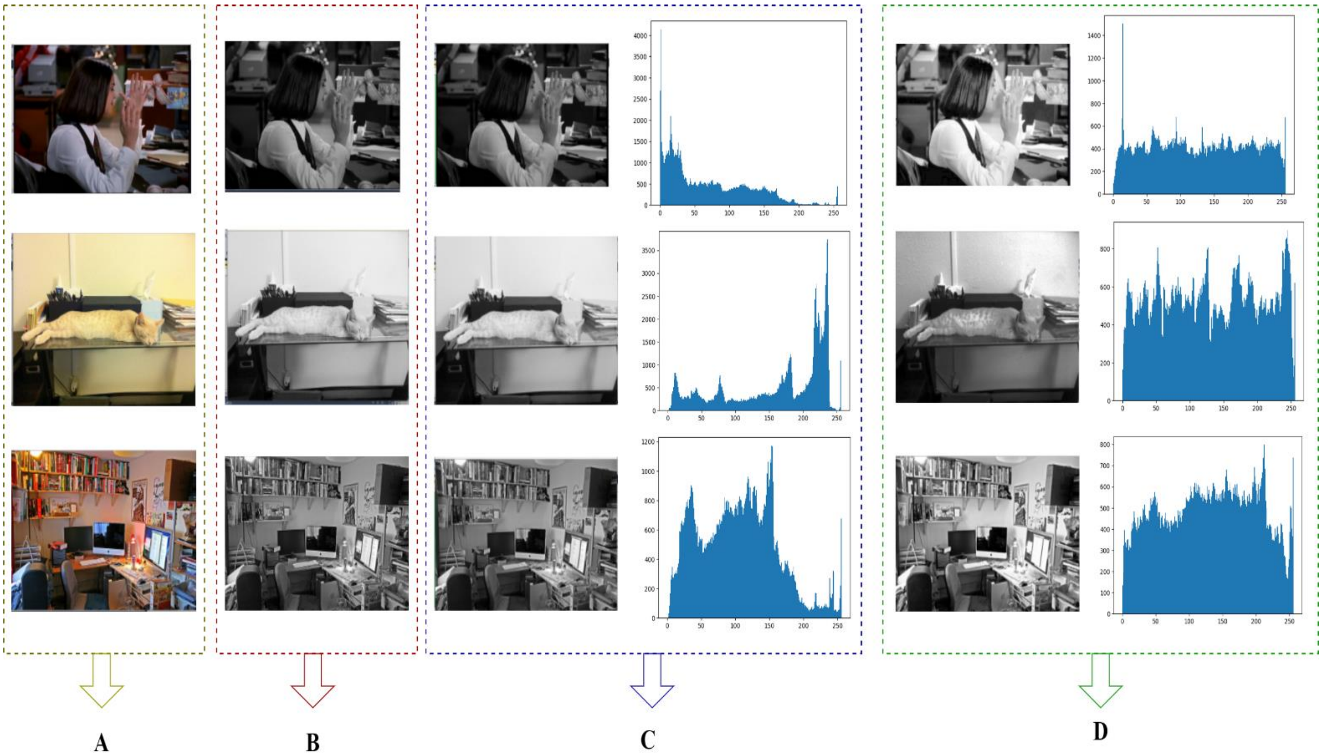

Figure 3 *the preprocessing stages. A is the original images, B is grayscale of original images, C is the Gaussian blur of grayscale image with its histogram, and D is the histogram equalization with there's histogram*

Image re-sizing

Image resizing is an essential preliminary stage in computer vision. In general, deep learning models train faster on tiny images. A bigger input image necessitates that the neural network learn from four times as many pixels, hence increasing the training duration of the architecture. The method that we used is bilinear interpolation [16]. Bilinear interpolation is a technique for interpolating functions with two variables that use repeated linear interpolation. The initial step of bilinear interpolation is linear interpolation in one direction, followed by a repetition in the other direction. While each step is linear in terms of sampled values and location, the overall interpolation is quadratic in terms of sample location. This is the equation for bilinear interpolation (7,8):

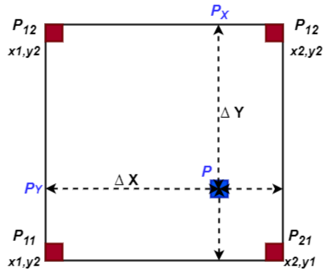

Figure 4 *The four red squares represent the data points, while the blue square represent the desired point that shoud be interpolated*

$$f(x, y_1) = \frac{x_2 - P_x}{x_2 - x_1} P_{11} + \frac{P_x - x_1}{x_2 - x_1} P_{12}\,, \tag{7}$$

$$f(x, y_2) = \frac{x_2 - P_x}{x_2 - x_1} P_{12} + \frac{P_x - x_1}{x_2 - x_1} P_{22} \tag{8}$$

Feature Extraction

Feature extraction is the third step in proposed model. Extraction of features is the transformation of input data into a collection of features. The fundamental goal of feature extraction is to decrease the redundant data and extract the most relevant information by assessing the unique traits or attributes that differentiate one input pattern from another, means converting raw data into numerical features that may be handled while keeping the original contents of data set [17].

Principle Component Analysis (PCA)

PCA is an unsupervised machine learning algorithm which aims at minimizing the dimensions of the features of dataset, while maintaining as much necessary data as possible. It turns the variables into a new set of variables known as principle components, It is a statistical procedure that uses orthogonal transformation to turn observations of correlated features into a collection of linearly uncorrelated data.[18], [19].

Deep Learning

The majority of deep learning techniques employ neural network designs, which is why deep learning models are commonly referred to as deep neural networks. Deep learning is a subset of machine learning that teaches computers to do and think like, as if computer has a brain to think and conclude. In deep learning, a computer model learns to classify images, text, or sounds directly from them. Deep learning models may attain high accuracy, often beyond human comprehension [21]. A sizable amount of labelled data and multi-layered neural network architectures are used to train the models.[20]. Convolutional neural networks are among the most frequent types of deep neural networks (CNN or ConvNet). This architecture is suitable for processing 2-Dimenssion data, such as photos, as it combines learnt features with input data and employs convolutional layers. CNNs classify images without manual feature extraction. CNNs extract features from images directly. The network learns relevant features while training on photos. Deep learning algorithms are accurate for object classification due to automated feature extraction [21].

Classification based on CNN

CNNs classify images without manual feature extraction. CNNs extract features from images directly. The network learns relevant features while training on photos. Deep learning algorithms are accurate for object classification due to automated feature extraction. A mathematical procedure on two functions yields a third function that shows how one function is affected by the other. Most CNNs have two parts: the feature extraction module, which includes the convolution layer and pooling layer. The classifier module is the scond part and includes “the fully-connected layer”. CNNs in one, two, or three dimensions work the same. The acquired CNN layers and parameters in our project are one-dimensional .

Convolution Layer

The convolutional layer is an arithmetic operation. The convolutional layer of the input image is calculated utilizing kernel filters to extract features. The kernel filter matrix is usually of the same dimension and with fixed and smaller parameters compared to the input images. The depth of the kernel filter is the same as the depth of the input image. Example: If the image size is 8 × 8 × 2 (two dimensions), then the filter size will be f*f*2, where f is usually an odd number (3,5,7, .. etc.). The filter mask incrementally covers the whole input image, estimating the dot product between the weights of the kernel filters and the value of the input image, so creating a 2D activation map as shown in figure (5) [22].

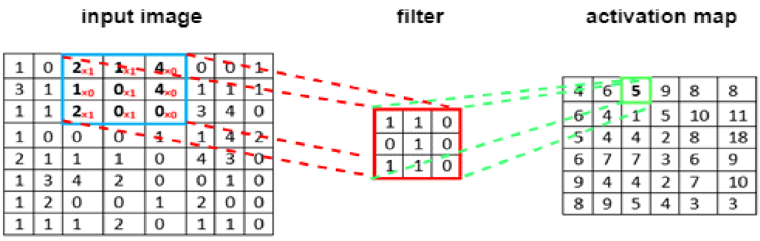

Figure 5 *convlution layer*

Pooling Layer

The pooling process includes passing a two-dimensional filter over each channel of the feature map and summing the features included within the filter's coverage zone. Max pooling is extensively used because it is more efficient in selecting the maximum number of objects from the feature map region covered by the filter., as shown in figure (6).

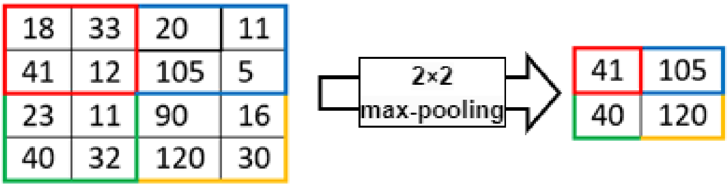

Figure 6 *pooling layer*

Fully Connected Layer

The output of the previous layer is connected to the fully connected layer as an input. By integrating the attributes of the preceding levels, this layer executes a high-level logical operation. This leads to a conclusive determination.

The proposed CNN model is described in full in figure (7), which describes its 27 layers distributed as follows:

- 1-D convolutional layers for feature extraction (9).
- Maxpooling 1-D layers (6).
- Leaky ReLU layer (8)
- Fully connected layer (dense) (3).
- Flatten layer (1).

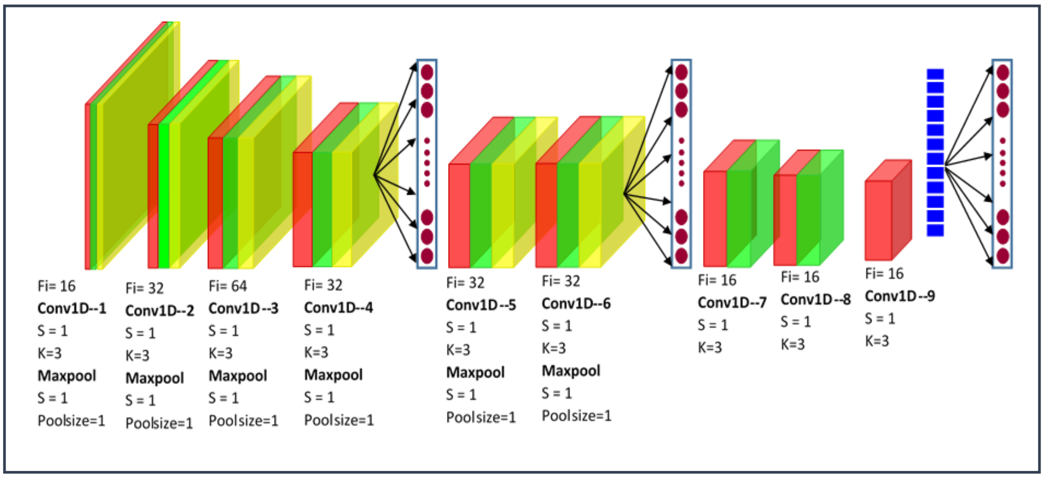

Figure 7 *Figure 5 the proposed CNN layers*

Build indoor mapping model

One of the popular techniques used to build maps is Simultaneous Localization and Mapping (SLAM). SLAM is a fundamental topic in the fields of Artificial Intelligence and mobile robotics that addresses the issue of localization and mapping in the absence of a prior map of the workspace [23]. SLAM is not an algorithm in and of itself; rather, it is a concept that may be implemented in a variety of ways. SLAM is a technical mapping method adopted by several sensors (ultrasonic, LIDAR, rangefinder, RGB camera, etc.) to simultaneously generate a map and locate themselves on it. The SLAM software system enables these sensors to plan a path across an unknown environment while simultaneously determining their own location inside that environment. SLAM technology makes use of many mathematical and statistical algorithms, and from these techniques is Kalman-filter and particle-filter. Given a series of observations, the Kalman and Particle filters are algorithms that iteratively update a state estimate and determine the innovations that drive a stochastic process. The Kalman filter achieves this purpose by linear projections, whereas the Particle filter employs a sequential Monte Carlo approach.

Kalman Filter

KF is a simplification of the Bayesian estimate for the linear model, which was originally proposed by Kalman (1960). Kalman filters are utilised to estimate the states of linear dynamical systems in state space format. As seen in the equation (9), the process model specifies the evolution of the state through time. t-1 to time t.

$$x_t = Fx_{t-1} + Bu_{t-1} + n_{t-1} \tag{9}$$

Where F represents the transition matrix of the state that applied to the previous state vector xt-1

The process model is paired with the measurement model, which describes the relationship between the current state and the measurement at time step t.

$$z_t = Ex_t + v_t \tag{10}$$

Where zt represents the measurement vector, E represents the measurement matrix, and vt is the vector of the noise measurement.

Particle Filter

Particle filter is a method for creating recursive Bayesian filter using Monte Carlo sampling. The concept is to depict the posterior density using random particles with weights.

Next, based on these samples and weights, construct estimates. The equation (11) represent the Bay's theorem:

$$P(X_K|Z_K) = \frac{P(Z_K|X_K)P(X_K)}{P(Z_K)} \quad (11)$$

The Particle issue may be summed up as follows: tracking the state of the system as it changes over time, having (noisy or ambiguous) observations arrive sequentially, and knowing the best feasible estimate of the hidden variable. The posterior distribution is determined by two iterations of prediction and updating. The prediction stage relies on a probabilistic model of the transitional density transition, equation (12):

$$P(x_k|z_{k-1}) = \int (x_k|x_{k-1})\,(x_{k-1}|z_{k-1})dx_{k-1} \quad (12)$$

Where: $(x_{k-1}|z_{k-1})$ is the probability density function (pdf). In the update stage, the priority of pdf is updated, equation (13).

$$P(x_k|z_k) = \frac{P(z_k|x_k)P(x_k|z_{k-1})}{(z_k|z_{k-1})} \quad (13)$$

Hardware Architecture

The figure (8) shows the how the components of the system that are used in our project are connected with each other.

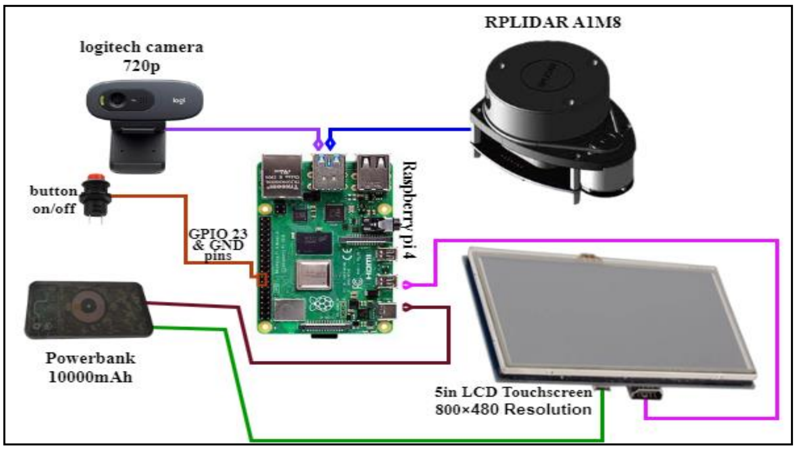

Figure 8 *Hardware components architecture*

Figure 9 *Design hand-held LIDAR*

2d-Rplidar A1 8m

RPLIDAR A1 is fundamentally a triangulation laser measurement system. LiDAR will be damaged if it direct sunlight exposure. RPLIDAR produces a modified infrared laser beam, which is then reflected by the target object for detection. The vision acquisition system in RPLIDAR A1 samples the returned signal, and the DSP built into RPLIDAR A1 begins processing the sample data to determine the distance and angle between the object and RPLIDAR A1. The figure (10) illustrates the algorithm of RPLIDAR A1 working. The distance of LIDAR is measured using (Triangulation ranging method) which illustrated in figure (11), and using the equation (14):

$$\text{Distance} = \frac{A \times B}{C} \tag{14}$$

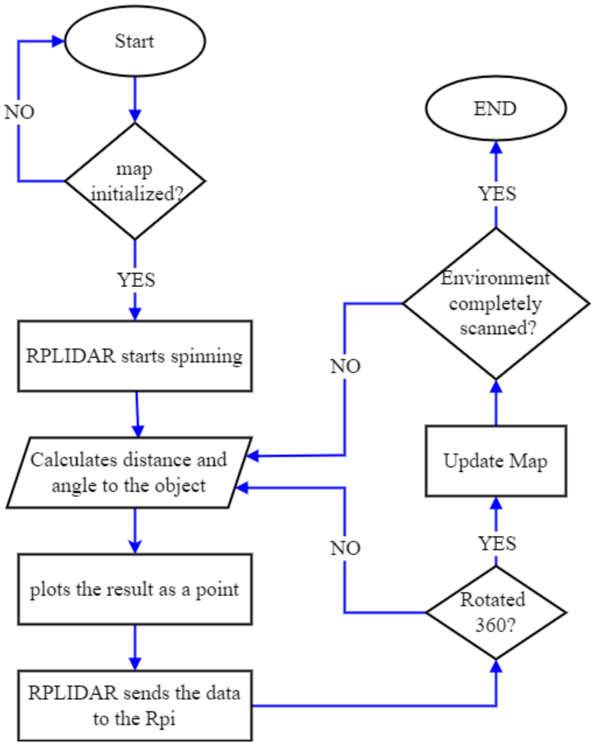

Figure 10 *RPLIDAR A1 algorithm*

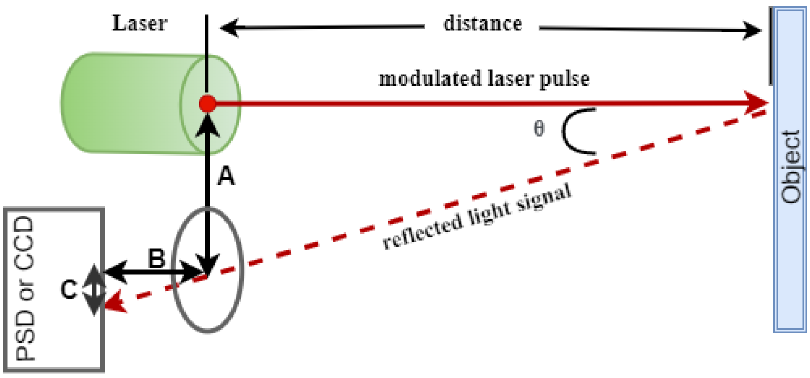

Figure 11 *the Distance using triangulation ranging method*

As the distances A and B are known, and the G value is measured using CCD or PSD, the distance is then calculated. Since the emitted and reflected laser light form a triangle, they are referred to as laser triangles [7].

Camera

Camera is a device that programmed to detect any object in real time and through RPLIDAR'S scan. The camera has resolution 720p, operating voltage range (2.8V — 5V), detection distance range (30mm-2000mm), ranging time less than 30ms, and the consumption power is 20mW. The camera is connected to the raspberry pi through GPIO pins (3V3 power, GPIO2 (SDA), GPIO 3 (SCL), GND).

Raspberry pi

Raspberry Pi as shown in figure (12) is a single board computer (SBC) with microprocessor(s), memory, input/output (I/O) and other features required of a functional computer, weighing only 50g. The electricity consumption of a raspberry is rated at (5V, 700mA). It is a comprehensive-computer, as the peripheral devices can be linked through USB2.0 ports. In addition, the board includes an Ethernet port for network connectivity and 40-GPIO pins for interfacing and controlling switches, sensors, LEDs, and other devices. The operating system used in raspberry pi is (raspbian) which is downloaded into micro SD card. There are multiple types of board models (A, B). The board that we used is Raspberry pi model-B, with RAM 8G, and the its power consumption 5V DC (via USB-C or via GPIO header).

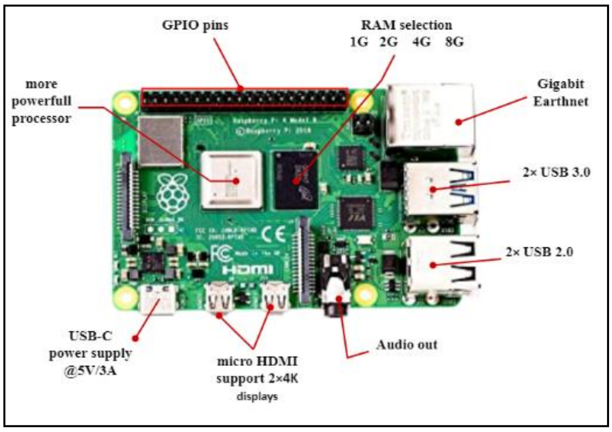

Figure 12 *the board of raspberry pi 4 model-B*

Liquid Crystal Display (LCD)

LCD's as indicated by its name, is used for display the output of our project. LCD comes with 5 inch Resistive Touch Screen LCD, HDMI interface, Designed for RPi. The hardware resolution is 800 × 480.

Model assessment of CNN algorithm

After completing the application of our model, it must be evaluated to see its efficiencyModel evaluation is the process of determining a model's performance, as well as its strengths and weakness, using a variety of measures. Model assessment is essential for assessing the model's efficacy at the earliest phases of research, and it also plays a part in model monitoring.The most common metrics to measure classification performance include accuracy, precision, F1 score, and recall [24]. These metrics are calculated by confusion matrix, which is a table that displays the classification model's performance to predict samples belonging to different classes. The confusion matrix consists of the terms: TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative).

Accuracy: it calculates the division of the total number of correctly classified samples by the total number of classified samples. The equation (11) of accuracy is:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (11)$$

Precision: it calculates the division of the number of actual correct positive prediction by total positive (true and false) predictions made by model. The equation (12) of precision is:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (12)$$

Recall: it is calculated by dividing the number of actual correct positive prediction by the total number of true positive and false negative. The equation (13) of recall is:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (13)$$

F1-Score: it calculates the weights average of precision and Recall. The F1 score improves as precision and recall improve. F1-score varies between 0 and 1. The closer the model is to 1, the better it is. The equation (14) of F1-score is:

$$\text{F}_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \quad (14)$$

Results and discussion

In this work we used CNN classification methods, and it is compared it with other works using MS COCO dataset and the same classifier, and based on accuracy as shown in table (1) .The result obtained from the our model during training/validation was the best if it is compared with other working or without making comparison. In addition, we used others performance evaluation measurements to see how efficient the model is and obtained highest evaluation based on (F1-Score, Recall, and Precision), as shown in table (2). As for the results of the system as a whole, which are related to building ground maps and object detection, Figure (13) shows the how efficiency of the system is.

Table 1 *Performance evaluation with other woking depends on accuracy*

Authors	model used	Dataset kind	Accuracy
G.ÖZTÜRK et al. **[25]**	Faster R-CNN Resnet 101	COCO dataset	85.1%
M.Shehab et al. **[26]**	CNN	Pascal VOC and Microsoft COCO datasets	98%
N.Nife, M. Chtourou **[27]**	CNN	COCO dataset	95.52 divided as follows: 98.97% people detection 99.12% Cars 100% Bicycle and clock
Proposed system	CNN	COCO dataset	100%

Table 2 *the performance evaluation depends on CNN classifier*

	F1-Score	Recall	Precision	Support
Macro avg.	1.00	1.00	1.00	9284
Weighted avg.	1.00	1.00	1.00	9284

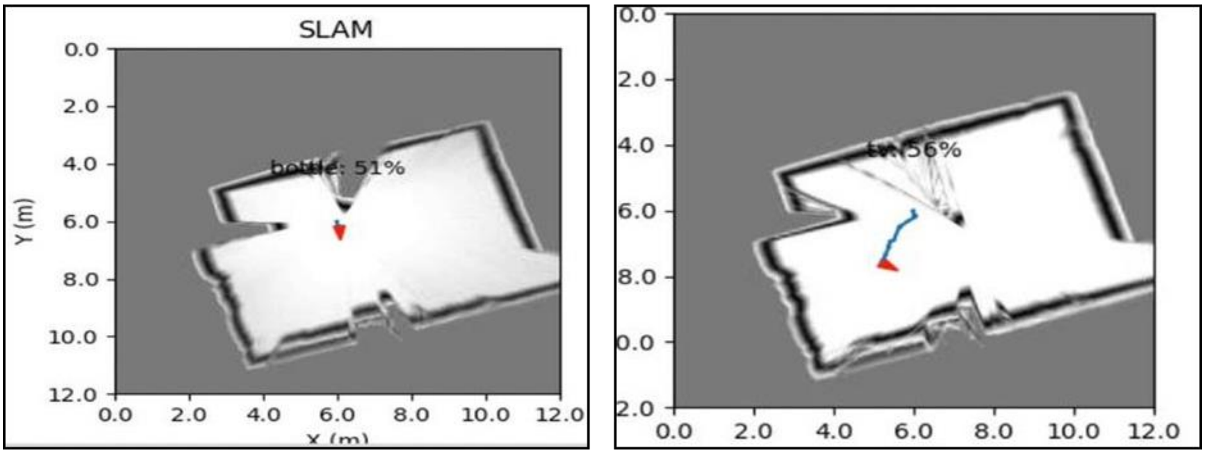

Figure 13 *screenshot of SLAM map, the red triangle represent the site of LIDAR, and the The blue squiggly line is the site's altering*

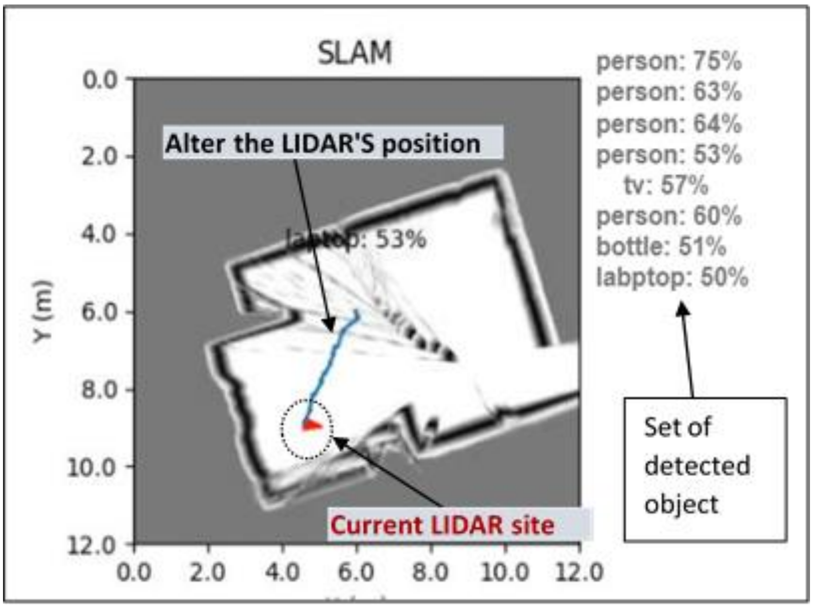

Figure 14 *list of objects detection appears on the right*

Conclusion

Floor mapping is one of the things that has become a reliable method in recent years for obtaining indoor environment data. In this project, we introduced an addition to SLAM main function, which is detecting objects, and the result was to build an intelligent system because it can detect any object at the same time as scanning the object with the LIDAR. Using performance evaluation metrics to see the efficiency of the system in terms of detecting objects using the CNN algorithm, introducing features in layers, and training the network for 100 epoch, 100% was achieved for each of the metrics: accuracy, F1-score, recall, and precision. In the future work, we aspire to add another sensor to measure the height between the LIDAR and the floor, and convert the map into 3-Dimenssion rather than 2-Dimenssion.

References

M. ZEYBEK, "Indoor mapping and positioning applications of hand-held LiDAR Simultaneous localization and mapping (SLAM) systems," Turkey Lidar J., vol. 3, no. 1, pp. 7–16, 2021, doi: 10.51946/melid.927004.

J. You, "Deep Neural Networks for Object Detection," vol. 17, pp. 159–165, 2022.

P. Fang and Y. Shi, "Small Object Detection Using Context Information Fusion in Faster R-CNN," 2018 IEEE 4th Int. Conf. Comput. Commun., vol. 1, no. 978, pp. 1537–1540, 2018.

J. U. Kim and Y. M. Ro, "ATTENTIVE LAYER SEPARATION FOR OBJECT CLASSIFICATION AND OBJECT LOCALIZATION IN OBJECT DETECTION Jung Uk Kim and Yong Man Ro," 2019 IEEE Int. Conf. Image Process., vol. 1, no. 978, pp. 3995–3999, 2019.

H. Caesar, J. Uijlings, and V. Ferrari, "COCO-Stuff : Thing and Stuff Classes in Context," in CVPR paper, 2018, pp. 1209–1218. doi: https://doi.org/10.1109/CVPR.2018.00132.

Z. He, J. Hou, and S. Schwertfeger, "Furniture Free Mapping using 3D Lidars," in Furniture Free Mapping using 3D Lidars, 2019, no. December, pp. 583–589. doi: 10.1109/ROBIO49542.2019.8961806.

T. R. Madhavan and M. Adharsh, "Obstacle Detection and Obstacle Avoidance Algorithm based on 2-D RPLiDAR," in 2019 International Conference on Computer Communication and Informatics, ICCCI 2019, 2019, pp. 1–4. doi: 10.1109/ICCCI.2019.8821803.

M. S. Aslam et al., .

"https://bucket-download.slamtec.com/60ef2c5a82129bf7213f0ff5d51e732d1c124c3a/LD108_SLAMTEC_rplidar_datasheet_A1M8_v2.3_en.pdf," Procedia Comput. Sci., vol. 17, no. 2, p. 18, 2016, doi: 10.1007/978-981-10-0830-6.

M. Royhan Iqbal, A. Husein Alasiry, H. Oktavianto, A. Darmawan, A. Teguh Budi Antok, and I. Kresno Wibowo, "Corner Detection with 2-D RPLIDAR to Detect Furniture on TCIFFRC Tracks," in IES 2020 - International Electronics Symposium: The Role of Autonomous and Intelligent Systems for Human Life and Comfort, Sep. 2020, pp. 297–301. doi: 10.1109/IES50839.2020.9231648.

A. J. Mohammed, M. M. Hassan, and D. H. Kadir, "Improving Classification Performance for a Novel Imbalanced Medical Dataset using SMOTE Method," Int. J. Adv. Trends Comput. Sci. Eng., vol. 9, no. 3, pp. 3161–3172, 2020, doi: 10.30534/ijatcse/2020/104932020.

A. Abualola and T. S. Gunawan, "Development of Colorization of Grayscale Images using CNN-SVM Proposed Colorization using CNN-SVM," pp. 1–10.

I. Žeger, S. Grgic, and J. Vuković, "Grayscale Image Colorization Methods : Overview and Evaluation," IEEE Access, vol. 9, pp. 113326–113346, 2021, doi: 10.1109/ACCESS.2021.3104515.

R. Bala, K. M. Braun, X. I. Group, and P. Rd, "Color-to-grayscale conversion to maintain discriminability," in Color Imaging IX: Processing, Hardcopy, and Applications, 2003, vol. 5293, no. December 2003, pp. 196–202. doi: 10.1117/12.532192.

J. J. Hwang, "Gaussian Forensic Detection using Blur Quantity of Forgery Image," 2019 Int. Conf. Green Hum. Inf. Technol., pp. 86–88, 2019, doi: 10.1109/ICGHIT.2019.00027.

Rasool.D. Haamied1, Bushra.Q. Al-Abudi1, and Raaid.N. Hassan1, "Classification Coco Dataset Using Machine Learning Algorithms," Turkish J. Physiother. Rehabil., vol. 32, no. 3, pp. 29686–29693, 2022, [Online]. Available: https://www.researchgate.net/publication/358229248%0ACLASSIFICATION

B. Oktavianto and T. W. Purboyo, "A Study of Histogram Equalization Techniques for Image Enhancement," Int. J. Appl. Eng. Res., vol. 13, no. 2, pp. 1165–1170, 2018, doi:

10.1088/1742-6596/1700/1/012076.

C. Networks and P. Analysis, “Feature Extraction for Dimensionality Reduction in Cellular Networks Performance Analysis,” pp. 1–10, 2020, doi: 10.3390/s20236944.

F. Kherif and A. Latypova, “Principal component analysis,” in Machine Learning: Methods and Applications to Brain Disorders, vol. 1, no. C, Elsevier Inc., 2019, pp. 209–225. doi: 10.1016/B978-0-12-815739-8.00012-2.

L. Malago, “Object detection using image reconstruction with PCA Object detection using image reconstruction with PCA,” no. October, 2019, doi: 10.1016/j.imavis.2007.03.004.

“Short PCG classification based.pdf.”

C. Janiesch, P. Zschech, and K. Heinrich, “Machine learning and deep learning,” Electron. Mark., vol. 31, no. 3, pp. 685–695, 2021, doi: 10.1007/s12525-021-00475-2.

“Diving deep in Deep Convolutional Neural Network-----layers.pdf.”

H. Taheri and Z. C. Xia, “SLAM; definition and evolution,” Eng. Appl. Artif. Intell., vol. 97, no. April 2020, p. 104032, 2021, doi: 10.1016/j.engappai.2020.104032.

N. Architecture, “MULTI-LABEL CLASSIFIER PERFORMANCE EVALUATION WITH CONFUSION MATRIX Damir,” in Computer Science & Information Technology (CS & IT), 2020, pp. 1–14. doi: 10.5121/csit.2020.100801.

D. K. Gülyeter ÖZTÜRK, Raşit KÖKER, Osman ELDOğAN, “Recognition of Vehicles, Pedestrians and Traffic Signs Using Convolutional Neural Networks.pdf,” in Recognition of Vehicles, Pedestrians and Traffic Signs Using Convolutional Neural Networks, 2020, pp. 1–8. doi: https://doi.org/10.1109/ISMSIT50672.2020.9255148.

M. Shehab, A.-G. Ammar, and S. M. Swadi, “Efficient Real-Time Object Detection based on Convolutional Neural Network,” Aug. 2021. doi: https://doi.org/10.1109/ICATE49685.2021.9465015.

N. I. Nife and M. Chtourou, “Video Objects Detection Using Deep Convolutional Neural Networks,” in 2022 6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 2022, pp. 1–6. doi: 10.1109/ATSIP55956.2022.9805931.