

Indoor Localization and Navigation based on Deep Learning using a Monocular Visual System

Rodrigo Eduardo Arevalo Ancona¹, Leonel Germán Corona Ramírez², Oscar Octavio Gutiérrez Frías³

UPIITA-IPN SEPI Section, Instituto Politécnico Nacional
Mexico City, Mexico

Abstract—Now-a-days, computer systems are important for artificial vision systems to analyze the acquired data to realize crucial tasks, such as localization and navigation. For successful navigation, the robot must interpret the acquired data and determine its position to decide how to move through the environment. This paper proposes an indoor mobile robot visual-localization and navigation approach for autonomous navigation. A convolutional neural network and background modeling are used to locate the system in the environment. Object detection is based on copy-move detection, an image forensic technique, extracting features from the image to identify similar regions. An adaptive threshold is proposed due to the illumination changes. The detected object is classified to evade it using a control deep neural network. A U-Net model is implemented to track the path trajectory. The experiment results were obtained from real data, proving the efficiency of the proposed algorithm. The adaptive threshold solves illumination variation issues for object detection.

Keywords—Visual localization; visual navigation; autonomous navigation; feature extractor; object detection

I. INTRODUCTION

Computer systems are important for autonomous mobile robots to perform crucial tasks using sensors to extract useful information from the environment [1]. Vision-based sensors are powerful tools, providing the capability to interact with the environment [2], [3]. Cameras are used as a sensing equipment for environment perception in artificial vision systems, extracting relevant features from the image, allowing autonomous navigation, 3D mapping, visual robot localization, and object recognition [4], [5], [6].

Visual perception describes the environment extracting key-points around the system. The extracted key-points can achieve full autonomy to a mobile robot [6], [7], [8].

The robot localization problem implies the absolute current position estimation of the system, but also includes the relative position with respect to the object [2], [9], [10]. Localization is mandatory for a successful indoor navigation, and situates the system and the goal on a map.

Many visual localization methods are based on deep learning extractors. In [11] Perez, Caballero and Merino proposed an extending Monte Carlo localization method, extracting information from the images to generate a visual recognition using a map-based localization approach. Fu *et al* [12] applied a 3D mapping with an RGB-D camera, using an improved ORB algorithm as a feature extractor and measure the similarity of the descriptors with Hamming distance. Chen and Cheng proposed a cloud-based visual SLAM in [13], using

the oriented Fast and ORB features to build the dense map and estimate the position. Chiang, Hsia and Hsu [14] proposed a stereo vision self-localization system applying a trigonometric function finding coarse distances from the robot to the landmark; therefore it is applied to a neural network to increase the precision of the distance measurement. Xu, Chou and Dong [15] presented a multi sensor based indoor algorithm for a localization system, integrating a convolutional neural network (CNN) based on image retrieval, to determine the location of the robot. Another example is presented by Xie, Wang, Li and Tang [16] use a CNN robot localization to recognize the scene, reducing the error of relocating the robot.

Object detection is necessary to predict the position of the obstacles to avoid them, and move through the environment using extracted landmarks from the images to detect objects and predict the next state of the system. Many approaches for object detection use image feature extractors based on edge features, color features and textures features. Edge features methods aim on the identification of key-points from brightness changes on images. Color features detectors compare nearest regions from key-points. Texture features used regions of interest to classify pixel intensity levels in a neighborhood, such copy-move detection algorithms that search similar areas in images [17], [18], [19], [20], [21]. Ahn and Chung in [22] combined the advantages from the multi-scale Harris corner and SIFT descriptors to make an object detection using depth range features for metric information. Zheng, Barahimi, Aoun and Amar [23] present a boosted convolutional neural network for object recognition, using a boosted blocks in a succession of convolutional layers. Pertusa, Gallego and Bernabeu [24] propose an application for smartphones that allows object recognition using CNN.

Robot navigation use images to determine the system position using key-points as representations of the environment and detect obstacles [4], [5], [6], [11], [25]. Indoor navigation incorporates some knowledge of the environment, in particular a space representation, such as a map. The environment map representation is divided into three groups: Map-based navigation, the system depends on a created model or topological map. Map building-based navigation, a topological representation is contracted while the robot navigates on the environment. Mapless, the system does not have a spatial representation of the environment [26].

Visual Navigation describes the environment using landmarks on the image to build the map, locate the robot and detect obstacles. Sineglazov and Ischenko [27] used a SURF and RANSAC algorithms to locate landmarks and a neural

network to determine the object coordinates. Yilmaz and Gupta focus on combine signals from visual and inertial sensors in [28] for an indoor environment, this approach is based on visual odometry and a building information model. Shao *et al* [29] fusion the visual and inertial navigation system and applied a Kalman Filter to decline the errors of visual navigation. Sineglazov and Ischenko [30] analyze the characteristic features of different methods to develop n algorithmic maintenance for intellectual visual navigation based on neural networks.

The present approach addresses indoor map-based localization and navigation problems. A deep convolutional neural network is used to locate the system on the map and the pose estimation is based on a background modeling for feature extraction. A copy-move detection algorithm is applied to detect obstacles and a neural network for its evasion. Furthermore, a U-Net model is used for path trajectory tracking. This paper is divided in: Section II presents the proposed method for visual localization and navigation; Section III consists on the experiment results and analysis; Section IV contains the conclusions of the proposed algorithm.

II. PROPOSED METHOD

The present paper provides a solution for indoor navigation problem, including a map-based robot localization on the environment, and planning the trajectory using visual information from the mobile robot. Fig. 1 presents a general diagram from the proposed algorithm.

A. Robot Localization

Robot visual localization interprets the image information to estimate the current position of the system. Artificial neural networks are computational models capable of adapting their behavior in response to the stimulus actions from the environment [31]. The proposed visual localization stage is based on a deep neural network (Fig. 2) as a feature extractor. The convolutional deep neural network works as a feature extractor and classifies the image to locate the system in the environment; afterward it measures the robot position using the extracted interest points. The architecture of the deep neural network consists on three blocks, the first block has two CNN blocks with 64 filters, a (3,3) kernel size, a stride (1,1), a relu activation function, and an average pooling; the second block has two CNN blocks with 128 filters, a (5,5) kernel size, a stride (2,2), a relu activation function, and an average pooling; the first block has two CNN blocks with 256 filters, a (5,5) kernel size, a stride (2,2), a relu activation function, and an average pooling, therefore is used a Dense block after the Flatten stage, and classify eight classes. An Adam optimizer and a 0.0005 learning rate are used to get more accurate results.

Illumination changes influence the image scene on indoor visual localization. The neural network training process uses an image processing (gamma transformation, logarithmic transformation, blurring, rotation and translation) to reduce the error on the classification and increase the accuracy of each area of the environment, such as, rooms or corridors. The localization stage is completed by estimating the position of the robot.

Subsequently, the background model is used as a reference image. The diagram of the background subtraction is in Fig. 3, the main idea is to extract points from the image to generate a background model and analyze the image frames using an adaptive threshold to detect object when the system is static. On the other hand, a copy move detection is used to detect objects while the system is on movement [32]. One of the most frequent image forgery attacks is duplicating a fragment and place it on an image. Copy-move detection is a common technique for image forensic for detection of the tamper zones with a duplicated region from the digital image and verifies its originality [33], [34], [35], [36], [37], [38].

Key-points on the background model are used for the identification of possible objects.

$$B = \frac{\sum I_r}{N} \tag{1}$$

B is background model, I is the r^{th} frame, and N is the number of frames. A subtraction between the current frame and the reference image is applied to estimate the position of the system. This process identifies pixel intensity variations to detect possible objects.

$$S = |f_s - B_s| \tag{2}$$

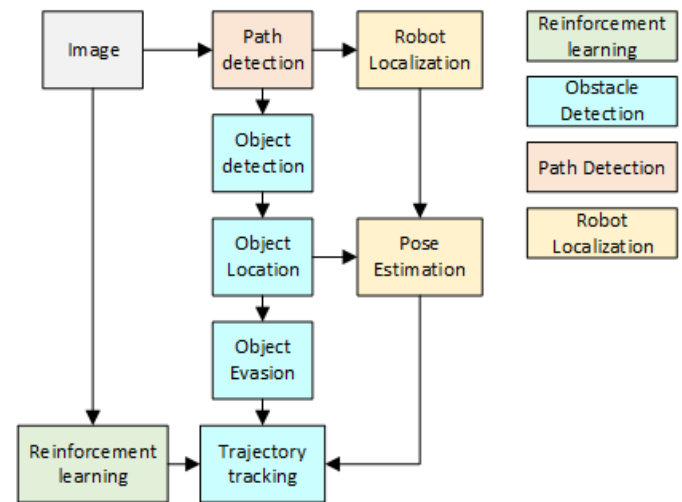


Fig. 1. Overview of the Proposed Method.

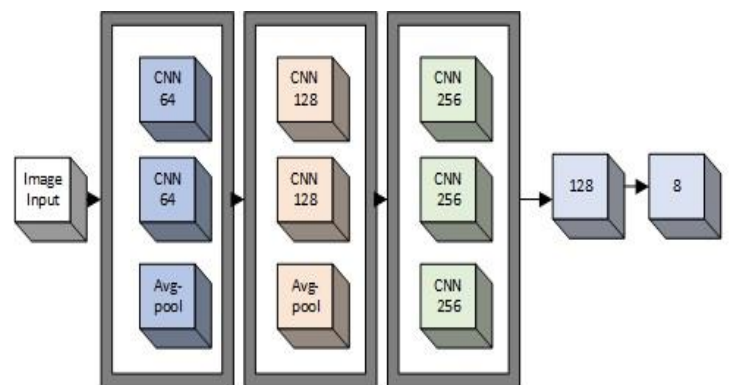


Fig. 2. Localization and Object Classification Deep Neural Network.

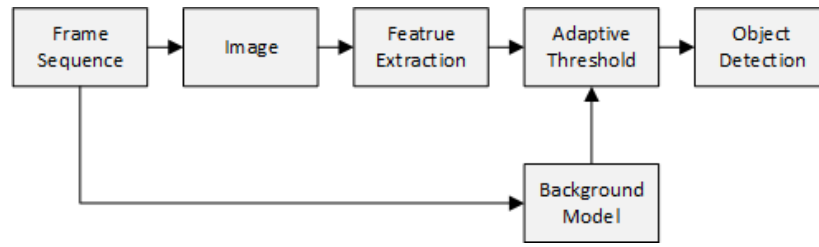


Fig. 3. Motion Detection Diagram.

where f_s is the current frame and B_s is the background. Therefore, it is extracted stable and invariant image features to get the position of the robot and obstacle detection, using an adaptive threshold based on copy move detection, such as in [35], [39]. The reference image S from the current image sequence and the last current image sequence are segmented the Discrete Cosine Transform is apply (DCT) [40].

$$C_{i,j}^K = \begin{pmatrix} C_{11}^k & \dots & C_{18}^k \\ \vdots & \ddots & \vdots \\ C_{81}^k & \dots & C_{88}^k \end{pmatrix} \quad (3)$$

The first stage of thresholding is obtained from C:

$$\lambda_1 = \sigma^2(\sqrt{\log(N)}) \quad (4)$$

$$\lambda_2 = \frac{\sum c_r}{r} \quad (5)$$

$$\lambda_3 = (\lambda_2 + \sigma^2) \sigma^2(\sqrt{\log(N)}) \quad (6)$$

where σ^2 is the variance, N are the number of frames, c_r are the values of each block C .

The Discrete Wavelet Transform (DWT) is applied to each block B and divides the image into signal components as level filters, approximations (low frequency), and details (high frequency) using the approximation sub-band as a noise filter and obtain characteristic features from the image.

$$D_{i,j}^K = \begin{pmatrix} d_{11}^k & \dots & d_{18}^k \\ \vdots & \ddots & \vdots \\ d_{81}^k & \dots & d_{88}^k \end{pmatrix} \quad (7)$$

The feature extraction allows the detection and recognition of the object, through the correlation between the images.

$$EV^k = (C^k \cdot D^k) \quad (8)$$

$$ev = (ev_{k2}, ev_{k3}, ev_{k4}, ev_{k5}, ev_{k6}, ev_{k7}, ev_{k8}) \quad (9)$$

The descriptors are computed using the block statics.

$$mean = \frac{\sum ev_1 - ev_2}{8} \quad (10)$$

ev_1 is the feature vector extracted from the current reference image and ev_2 is the feature vector extracted from the last reference image. The second stage from the adaptive threshold avoids the error in object detection and its localization. Each matrix D_{ij}^k of the current image are a shift to the right the column, shift down rows as in [39] and the descriptors are computed generating $\beta = \kappa = \varsigma = ME$.

$$T_1 = \frac{\beta + \kappa + \varsigma}{3} \quad (11)$$

Therefore, the thresholds are generated as in [39], then is proceed to compare the descriptors and obtain the interest points.

$$a = |mean| < \lambda_2 \text{ and } |mean| < T_1$$

$$b = \lambda_2 < T_1 \text{ and } mean < \lambda_3 \quad (12)$$

$$obj = \begin{cases} 0, a \text{ or } b \\ 1, other \end{cases} \quad (13)$$

The detection and localization of the object is owed to the pixel values changes on the image sequence. Therefore, a neural control network is applied for object classification and evasion. The neural control network allows the recognition of the detected objects, and the control tasks to evade the objects.

Once the interest points are detected, the robot position is estimated. The camera position is on the center and bottom of the image O as in Fig. 4.

The distance between the camera h and the object is given in centimeters:

$$h = mean(h_s, h_i) \quad (14)$$

where h_s is acquired by ultrasonic sensor, h_i is the calculated distance with the camera.

$$h_s = f_s - f_f \quad (15)$$

$$h_i = \sqrt{(\sin(b) * (\overline{OQ}))^2} \quad (16)$$

$$b = \frac{P * Q}{|P| * |Q|} \quad (17)$$

$$\overline{OQ} = \sqrt{(O(x) - Q(x))^2 + (O(y) - Q(y))^2} \quad (18)$$

f_s is the start time from the pulse and f_f is the final time from the pulse. The width from the detected obstacle is:

$$w = (\overline{OQ} * h) / 100 \quad (19)$$

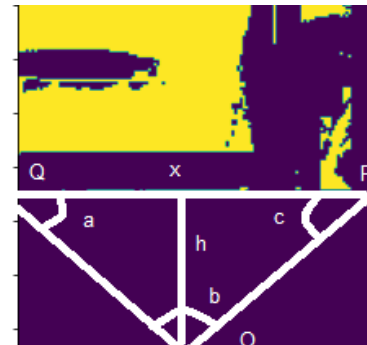


Fig. 4. Obstacle Detection and Pose Estimation.

where:

$$\overline{PQ} = \sqrt{(Q(x) - P(x))^2 + (Q(y) - P(y))^2} \quad (20)$$

B. Robot Navigation

The neural control network classifies the detected objects and the control tasks to evade them. A path planning is implemented a control system using the neural network approach described in using predefined controls to evaded the recognized object; considering a system with multiple inputs and multiple outputs.

$$v = F(\dot{s}, s) \quad (21)$$

The system considered involve both non-moving and moving rigid objects and define the robot position (p), rotation matrix (q), velocity (v), angular velocity (w).

$$p = [p_x, p_y]^T \quad (22)$$

$$q = [q_\theta, q_x, q_y]^T \quad (23)$$

$$v = [v_x, v_y]^T \quad (24)$$

$$w = [w_x, w_y]^T \quad (25)$$

Each control v defines a force-torque pair (F, τ) acting on the mass center B , therefore the motion equations can be defined as:

$$\dot{x}(t) = \begin{bmatrix} \dot{p} & v(t) \\ \dot{q} & 1/2((w(t) - q(t))) \\ \dot{v} & F/M \\ \dot{w} & R(t)I^{-1}R(t)^T \end{bmatrix} \quad (26)$$

The system used a set of predefined controls U evaluating the current system state.

$$U = \begin{bmatrix} U_1 & 0 & 1 \\ U_2 & 1 & 1 \\ U_3 & 0 & 0 \\ \vdots & \vdots & \vdots \\ U_4 & 1 & 0 \end{bmatrix} \quad (27)$$

Evaluating the states choose the control to evade the object and track the trajectory. The trajectory is predefined using reinforcement learning, using a cost function, based on the Euclidean distance, to assign priorities for the movements.

Then the trajectory is tracked by the cloning behavioral network. The system learns the trajectory and reproduces the user's control using the knowledge of a human pilot for training a convolutional neural network (CNN) that receives as input the current camera frame and outputs the next state. The idea is to reinforce the knowledge of the trajectory using a UNET model [41], [42].

A successful path tracking depends on the path and specific object recognition. To perform path recognition in an image is required a semantic segmentation. Path detection describes the systems movements in the environment [43]. The predicted class $P(class)$ represents the coincidence between the input image $O_{detected}$ and the predicted class $O_{predicted}$.

$$P(class) = \frac{A \cap B}{A \cup B} \quad (28)$$

where:

$$A \cap B = O_{detected} * O_{predicted}$$

$$A \cup B = O_{detected} + O_{predicted}$$

The existing probability of a specific class and the object detected (in this case the trajectory) are taken in consideration. During the training, the following loss function is used the categorical cross-entropy:

$$L(y, \bar{y}) = \sum \sum (y_{ij} * \log(\bar{y}_{ij})) \quad (29)$$

y is the training image and \bar{y}_{ij} is the label. The use of the categorical cross-entropy will compare the distribution of the predictions, where the probability of the true class is 1 and 0 for other class.

III. EXPERIMENTAL RESULTS AND ANALYSIS

In this section the results are presented for the visual robot localization, object detection and trajectory tracking using a mobile robot with a Raspberry Pi 4 and a Raspberry V2 module (Fig. 5).

The proposed neural networks are trained under Windows 10 operating system using TensorFlow and Pytorch deep learning frameworks, the CPU model is 8GB Intel® Core i7-6700 @ 3.40 GHz, with a graphics card model NVIDIA GeForce GTX 960.

A. Robot Localization

The algorithm for robot localization uses extracted features from images as landmarks to determine where the robot is, by comparing this feature with the environment image.

The class coincidence in Table I has a good performance an 88% of accuracy, obtaining a high efficiency for robot localization. To increase the performance of the localization neural network more images for each class have to be added to the dataset.

The train dataset uses 1532 images from Room1, 1665 images from Room2, 1869 from Hall and 2052 from Room3, the validation dataset uses 1002 images from Room1, 1105 images from Room2, 1157 from Hall and 2500 from Room3, the test dataset uses 600 images, in addition, a pre-processing image process was applied to increase the classification accuracy (translation, rotation, and blurring). The neural network for localization presents in Fig. 6. An accuracy of 95%, reducing the classification error, then the position of the system is estimated by extracting key points from the background model and the object detection.

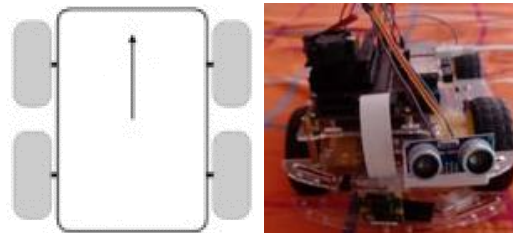


Fig. 5. Robot.

TABLE I. LOCALIZATION

Localization	Room 1	Room 2	Hall	Room 3
Coincidence	90 %	91 %	93 %	85 %

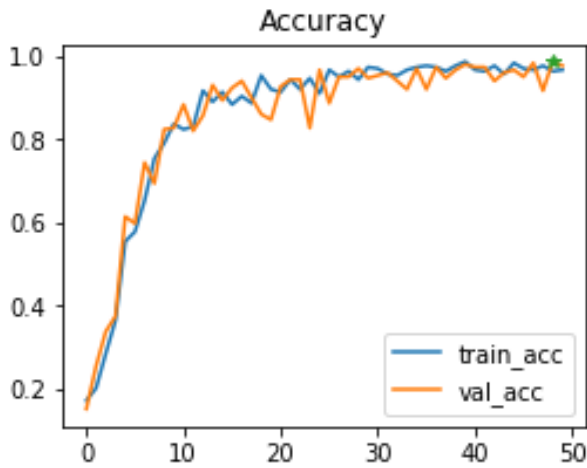


Fig. 6. Neural Network for Localization Accuracy.

The proposed approach tracks all the detected key points on the frame sequence, obtaining the robot position and locates the obstacle as in Fig. 7. The main advantage is the detection and points matching despite the illumination changes or the system position. In order to evaluate the proposed algorithm is used the accuracy and feature ratio sequence rs .

$$rs = \frac{np}{np_i} \quad (30)$$

where np are the detected points on the current image and np_i are the detected points on the previous image frame. In order to analyze the effectiveness of the object detection algorithm is compare in Table II with popular algorithms.

Clearly, the proposed method has a good performance tracking 97% of the detected features. In Fig. 8 is observed the behavior of the precision in a sequence, the proposed method has a 90%. The extracted features can be tracked without losing them on the frame sequence; the ratio sequence has a variation because the robot position changes.

The experiment was realizes using the robot on different conditions of illumination on the indoor environment. An advantage of the proposed algorithm is the feature detection and tracking on frame sequences, this advantage is useful for the detection and tracking of moving objects. In Fig. 8, shows that the extracted features can be tracked without losing information regardless of changes on the system position and illumination changes, being an advantage over the conventional methods. Once the obstacle is located a neural network classifies the object for its evasion. The effectiveness of the object classification is evaluated using the confusion matrix in Table III.

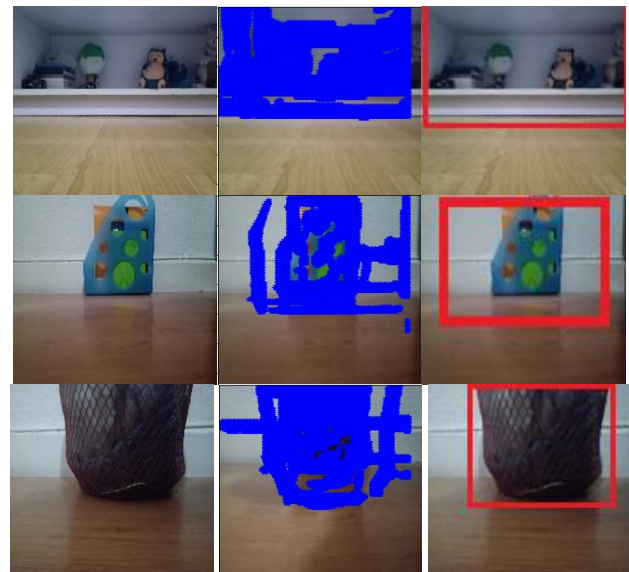
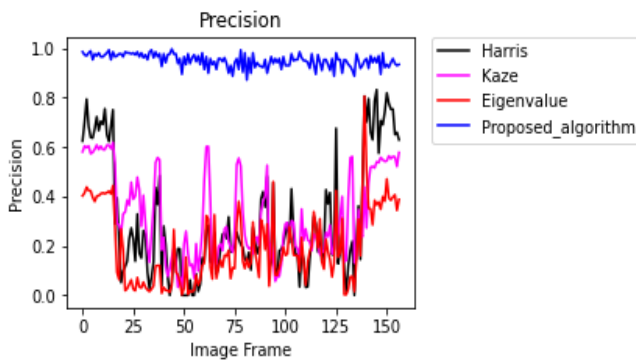


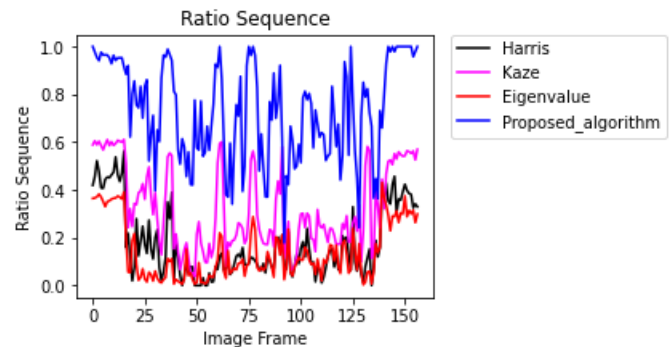
Fig. 7. Object Detection.

TABLE II. COMPARISON OF FEATURES EXTRACTORS

Method	Kaze [44], [45]	Harris [40], [46]	Eigen-Features [47]	Proposed Method
Accuracy	0.8322	0.6821	0.5051	0.9035
Ratio sequence	0.7418	0.544	0.3608	0.8011



(a)



(b)

Fig. 8. Object Detection, (a) Precision, (b) Ratio Sequence.

TABLE III. CONFUSION MATRIX

	False	True	Total
Negative	93	7	100
Positive	6	474	480
total	99	481	580

The accuracy and the recall from the control neural network are described using statics values from the confusion matrix. Where T_p are the true positives, F_p are the false positives and F_n the false negatives.

$$precision = \frac{T_p}{T_p+F_p} \quad (31)$$

$$recall = \frac{T_p}{T_p+F_n} \quad (32)$$

The efficiency of the object classification neural network has high accuracy. Clearly, in Table IV, the proposed algorithm has a good performance with a 93% of accuracy and a 98% recall for object classification.

The trajectory is generated using reinforcement learning, and it is tracked by the U-Net model. In order to verify the accuracy performance of the U-Net model trained for the trajectory detection, the used data was acquired using the proposed system. The dataset contains real 1500 images with the path, in addition, the pre-processing image was applied to increase the accuracy (translation, rotation, and blurring). Furthermore, a segmentation on the image detects the path to decide the control actions to track a trajectory.

The accuracy from the U-Net model in Fig. 9 is 90% for path detection. The semantic segmentation recognizes the predicted class on the image, moreover is used to track the trajectory and evade a collision. The U-Net was trained using 1500 images and labels on different conditions on the environment to get a better performance.

To verify the applicability and accuracy of the algorithm in Fig. 10 is presented the image segmentation. The red region recognizes the path.

TABLE IV. STATISTICS VALUES

Precision	93 %
Recall	98 %

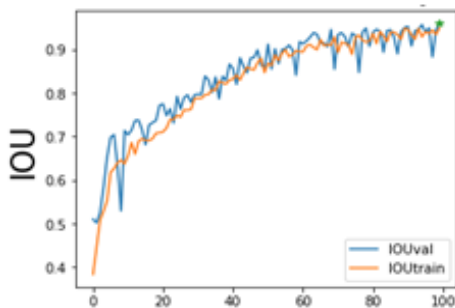


Fig. 9. U-Net Model Accuracy.

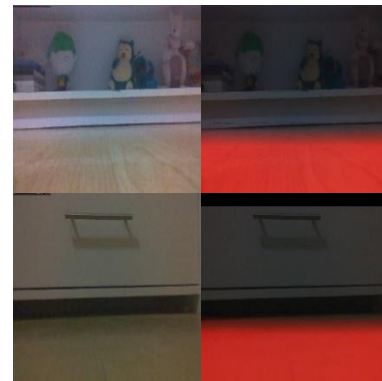


Fig. 10. Trajectory Tracking.

IV. CONCLUSIONS

This paper proposes an indoor navigation method, applying a self-localization solution and novel object detection. This approach has a high precision for feature tracking, this increases object detection and evasion.

A map-based is used in the present approach to self-locate the system on the map using a neural network. The accuracy for the localization neural network is >90%. In addition, the use of a deep learning neural network reduces the error in the system localization.

Furthermore, object detection is used to estimate the position of the system. The object detection has a >90% accuracy owing to the feature extractor and an adaptive threshold solving problems with illumination changes. The proposed solution tracks the object in the image frames; therefore, the system classifies the obstacle and reinforcement learning to evade it.

The path detection and track a trajectory a U-Net model is applied, providing the detection of the path, also the image semantic segmentation reduces the possibility of a collision. Semantic segmentation is an accurate process for the identification of the path.

To get an efficient object detection the system has to be on a dynamic environment for the detection of the objects, such as, an object has to move or the system is moving through the goal. The system navigates on unknown environments, but has to be trained to get the localization of system, because is used a supervised learning, also for the path detection is necessary to train the system with on the environment.

The proposed algorithm provides a novel solution for the navigation problem. In addition, the detection of similar zones on the frames increases the accuracy and precision for the object detection stage, also the detected features are used as landmarks to estimate the position of the system. This algorithm can be used to build a map representation of the environment. Landmarks can be set on the map to train the system with a reinforcement learning to set the locations.

The proposed algorithm has many applications, such as, car detection and pedestrian detection for autonomous driving, a surveillance robot, face tracking for recognition, car tracking for speed infractions, explorer robot, a background subtraction for a surveillance system.

ACKNOWLEDGMENT

This article was supported by the Investigation and Postgraduate Secretary of the Instituto Politécnico Nacional. Rodrigo Eduardo Arevalo-Ancona is a scholarship holder of the CONACYT.

REFERENCES

- [1] J. Rantanen, M. Mäkelä, L. Ruotsalainen and M. Jaakkola, "Motion Context Adaptive Fusion of Inertial and Visual Pedestrian Navigation," in International Conference on Indoor Positioning and Indoor Navigation (IPIN), Nantes, France, 2018.
- [2] S. Tarao, Y. Fujiwara, N. Tsuda and S. Takata, "Development of Autonomous Mobile Robot Platform Equipped with a Drive Unit Consisting of Low-End In-Wheel Motors," in 5th International Conference on Control and Robotics Engineering (ICCRE), Osaka, Japan, 2020.
- [3] N. Alfiany, G. Jati, N. Hamid, R. A. Ramadhani, M. W. D. Santika and W. Jatmiko, "Kinematics and Simulation Model of Autonomous Indonesian "Becak" Robot," in 2020 IEEE Region 10 Symposium (TENSymp), Dhaka, Bangladesh, 2020.
- [4] M. Pan, Y. Liu, J. Cao, Y. Li, C. Li and C.-H. Chen, "Visual Recognition Based on Deep Learning for Navigation Mark Classification," IEEE Access, vol. 8, pp. 32767 - 32775, 2020.
- [5] R. Druon, Y. Yoshiyasu, A. Kanezaki and A. Watt, "Visual Object Search by Learning Spatial Context," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 1279 - 1286, 2020.
- [6] T. Fäulhammer, R. Ambruş, C. Burbridge, M. Zillich, J. Folkesson, N. Hawes, P. Jensfelt and M. Vincze, "Autonomous Learning of Object Models on a Mobile Robot," IEEE Robotics and Automation Letters, vol. 2, no. 1, pp. 26 - 33, 2017.
- [7] C. Häne, Heng, Lee, Fraundorfer, Frugale, Sattler and Pollefeys, "3D Visual perception for self driving cars using a multicamera system: Calibration, mapping, localization and obstacle detection," Image and Vision Computing, pp. 1-35, 2017.
- [8] Q. Hu, S. Paisitkriangkrai, C. Shen, A. v. d. Hengel and F. Porikli, "Fast Detection of Multiple Objects in Traffic Scenes With a Common Detection Framework," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 4, pp. 1002 - 1014, 2015.
- [9] Y. Yasuda, L. E. Martins and F. Cappabianco, "Autonomous Visual Navigation for Mobile Robots," ACM Computing Surveys, vol. 53, no. 1, pp. 1-34, 2020.
- [10] S. Huang and G. Dissanayake, "Robot Localization: An Introduction," in Wiley Encyclopedia of Electrical and Electronics Engineering, Wiley, 2016.
- [11] J. Perez, F. Caballero and L. Merino, "Enhanced Monte Carlo Localization with Visual Place Recognition for Robust Robot Localization," Journal of Intelligent and Robotic Systems, vol. 80, no. 3-4, pp. 641-656, 2015.
- [12] Q. Fu, H. Yu, L. Lai, J. Weng, X. Peng, W. Sun and M. Sun, "A Robust RGB-D SLAM System With Points and Lines for Low Texture Indoor Environments," IEEE Sensors Journal, vol. 19, pp. 9908-9921, 2019.
- [13] Y. Zheng, S. Chen and H. Cheng, "Real-Time Cloud Visual Simultaneous Localization and Mapping for Indoor Service Robots," IEEE Access, pp. 16816-16829, 2020.
- [14] J. Chiang, C. Hsia and H. Hsu, "A Stereo Vision Based Self Localization System," IEEE Sensors Journal, vol. 13, no. 5, 2013.
- [15] S. Xu, W. Chou and H. Dong, "A Robust Indoor Localization System Integrating Visual Localization Aided by CNN Based Image Retrieval with Monte Carlo Localization," MPDI Senosrs, no. 249, pp. 1-19, 2019.
- [16] T. Xie, K. Wang, R. Li and X. Tang, "Visual Robot Relocalization Based on Multi-Task CNN and Image-Similarity Strategy," MDPI Sensors, vol. 20, pp. 2-20, 2020.
- [17] M. Ikhlal, M. Hariadi and I. K. E. Pumama, "Modified Multi-scale Feature Extraction for Copy-Move Forgery Detection Based on CMD-SIFT," in 2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 2019.
- [18] A. Mumcu and I. Savran, "Copy move forgery detection with using FAST key points and SIFT description vectors," in 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2018.
- [19] M. M. Al-Hammadi and S. Emmanuel, "Improving SURF Based Copy-Move Forgery Detection Using Super Resolution," in 2016 IEEE International Symposium on Multimedia (ISM), San Jose, CA, USA, 2017.
- [20] W. C. N. Kaura and S. Dhavale, "Analysis of SIFT and SURF features for copy-move image forgery detection," in International Conference on Innovations in Information, Embedded and Communication Systems (ICIECS), Coimbatore, India, 2017.
- [21] Y. Aydin, G. Muzaffer and G. Uluş, "Detection of copy move forgery technique based on SIFT and SURF," in 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2018.
- [22] S. Ahn and W. Chung, "Efficient SLAM algorithm with hybrid visual map in an indoor environment," in 2007 International Conference on Control, Automation and Systems, Seoul, Korea, 2007.
- [23] S. Barahimi, N. Aoun and C. Amar, "Boosted Convolutional Neural Network for object recognition at large scale," Neurocomputing, vol. 330, pp. 337-354, 2019.
- [24] A. Pertusa, A. Gallego and M. Bernabeu, "MirBot: A collaborative object recognition system for smartphones using convolutional neural networks," Neurocomputing, vol. 293, pp. 87-99, 2018.
- [25] M. Rysuke, M. Adachi, H. Ishida, T. Watanabe, K. Matsutani, H. Komatsuzaki, S. Sakata, R. Yokota and S. Kobayashi, "Visual Navigation Based on Semantic Segmentation Using Only a Monocular Camera as an External Senso," Journal of Robotics and Mechatronics, vol. 32, no. 6, pp. 1137-1155, 2020.
- [26] D. Souza and Kak, "Vision for Mobile Robot Navigation: A Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 2, pp. 237-268, 2002.
- [27] V. M. Sineglazov and V. S. Ischenko, "Intelligent Visual Navigation System of High Accuracy," in IEEE 5th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD), Kiev, Ukraine, 2019.
- [28] A. Yilmaz and A. Gupta, "Indoor positioning using visual and inertial sensors," in 2016 IEEE SENSORS, 2016.
- [29] W. Shao, L. Dou, H. Zhao, B. Wang, H. Xi and W. Yao, "A Visual/Inertial Relative Navigation Method for UAV Formation," in 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 2020.
- [30] V. Sineglazov and V. Ischenko, "Intelligent system for visual navigation," in 4th International Conference on Methods and Systems of Navigation and Motion Control (MSNMC), Kiev, Ukraine, 2016.
- [31] A. Alanis, N. Arana, C. Lopez and E. Guevara Reyes, "Integration of an inverse Optimal Neural Controller with Reinforced SLAM for Path Planning and Mapping in Dynamic Environments," Computación y Sistemas, vol. 19, no. 3, pp. 445-456, 2015.
- [32] Kermani and Asemani, "A Robust Adaptive Algorithm of Moving Object Detection for Video Surveillance," Journal on Image and Video Processing, vol. 1, no. 127, pp. 2-9, 2014.
- [33] Mushtaq and Mir, "Copy-Move Detection Using Gray Level Run Length Matrix Features," Optical and Wireless Technologies, vol. 472, pp. 411-420, 2018.
- [34] Kuznetsov and Myasnikov, "A Copy-Move Detection Algorithm Using Binary Gradient Contours," International Conference on Image Analysis and Recognition, vol. 9730, pp. 349-357, 2016.
- [35] Ghorbani, Firouzmand and Faraahi, "DWT-DCT (QCD) Based Copy-move Image Forgery," in 2011 18th International Conference on Systems, Signals and Image Processing, Sarajevo, Bosnia and Herzegovina, 2011.
- [36] J. N. Yuan Rao, "A Deep Learning Approach to Detection of Splicing and Copy move Forgeries in Images," IEEE International Workshop on Information Forensics and Security, 2016.
- [37] Y. Wu, W. Abd-Almageed and P. Natarajan, "Deep Matching and Validation Network An end to end to constrained Image Splicing Localization and Detection," IEEE Computer Vision and Pattern Recognition, pp. 1480-1489, 2017.

- [38] S. L. Ashwini Malviya, "Pixel based Image Forensic Technique for copy-move forgery," 7th International Conference on Communication Computing and Virtualization 2016, vol. 79, pp. 383-390, 2016.
- [39] X. Wang, X. Zhang, Z. Li and S. Wang, "A DWT-DCT Based Passive Forensics Method for Copy-move Attacks," in Third Conference on Multimedia Information Networking and Security, 2011.
- [40] Harris and Pike, "3D Positional integration from image sequences," Image and Vision Computing, vol. 6, no. 2, pp. 97-90, 1988.
- [41] O. Ronneberger, P. Fischer and T. Brox, "U-Net Convolutional Networks for Biomedical Image Segmentation," Medical Image Computing and Computer Assisted, pp. 234 - 241, 2015.
- [42] X. Hu, M. Naiel, A. wong, M.-. Lamm and P. Fieguth, "RUNET: A Robust UNET Architecture for Image Super Resolution," in IEEE Conference on Computer Vision, 2019.
- [43] S. LaValle, Planning Algorithms, Cambridge University Press, 2006.
- [44] P. F. Alcantarilla, "KAZE features," in European Conference on Computer Vision, Berlin, Berlin, Germany, 2012.
- [45] M. Pourfard, T. Hosseinian, R. Saeidi, S. A. Motamedi, M. J. Abdollahifard, R. Mansoori and R. Safabakhsh, "KAZE-SAR: SAR Image Registration Using KAZE Detector and Modified SURF Descriptor for Tackling Speckle Noise," IEEE Transactions on Geoscience and Remote Sensing, vol. press, pp. 1-12, 2021.
- [46] B. A. Jasani, S.-K. Lam, P. K. Meher and M. Wu, "Threshold-Guided Design and Optimization for Harris Corner Detector Architecture," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 12, pp. 3516 - 3526, 2018.
- [47] J. Shi and C. Tomasi, "Good Features to Track," IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8, 1994.