SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ DERGİSİ

Sakarya University Journal of Science
SAUJS

e-ISSN 2147-835X | Period Bimonthly | Founded: 1997 | Publisher Sakarya University |
http://www.saujs.sakarya.edu.tr/en/

Title: Comparison of Object Detection and Classification Methods For Mobile Robots

Authors: Önder ALPARSLAN, Ömer ÇETİN

Recieved: 2020-11-20 17:56:44

Accepted: 2021-04-26 13:29:55

Article Type: Research Article

Volume: 25
Issue: 3
Month: June
Year: 2021
Pages: 751-765

How to cite
Önder ALPARSLAN, Ömer ÇETİN; (2021), Comparison of Object Detection and Classification Methods For Mobile Robots. Sakarya University Journal of Science, 25(3), 751-765, DOI: https://doi.org/10.16984/saufenbilder.828841
Access link
http://www.saujs.sakarya.edu.tr/en/pub/issue/62736/828841

Sakarya University Journal of Science 25(3), 751-765, 2021

Comparison of Object Detection and Classification Methods For Mobile Robots

Önder ALPARSLAN*[1], Ömer ÇETİN[2]

Abstract

As one of today's popular research field, mobile robots, are widely used in entertainment, search and rescue, health, military, agriculture and many other fields with the advantages of technological developments. Object detection is one of the methods used for mobile robots to gather and report information about its environment during these tasks. With the ability to detect and classify objects, a robot can determine the type and number of objects around it and use this knowledge in its movement and path planning or reporting the objects with the desired features. Considering the dimensions of mobile robots and weight constraints of flying robots, the use of these algorithms is more limited. While the size and weight of mobile devices should be kept relatively small, successful object classification algorithms require processors with high computational power. In this study, to be able to use object detection information for mapping and path planning, object detection and classification methods were examined, and for the usage in low weight and low energy consuming platforms through developer boards, detection algorithms were compared to each other.

Keywords: Object Detection and Classification, Mobile Robots, Convolutional Neural Network, Deep Learning.

1. INTRODUCTION

An autonomous mobile robot is a kind of robot which is not bounded to a physical location and has the ability to move around. The travel of the vehicle can be provided by the help of guidance devices which allow the robots to go through pre-defined trajectory or the robot itself has the capability to understand its environment and move around the obstacles and plan a route to the target. This route can be updated in every step by controlling the robot's position and surrounding objects.

Creating a robust mobile robot depends on a reliable and effective path planning strategy [1]. This problem was solved by the algorithms which take the robot to the target in secure with the help

*Corresponding author: oalparslan@hho.edu.tr
[1] National Defence University, Hezârfen Aeronautics and Space Technologies Institute, 34049, Istanbul.
ORCID: https://orcid.org/0000-0001-8803-1597
[2]National Defence University, Air NCO Higher Vocational School, 35410, Izmir.
E-Mail: oalparslan@hho.edu.tr; omer_cetin@outlook.com.tr
ORCID: https://orcid.org/0000-0001-5176-6338

of a well-known pre-defined map [2-4]. A wheeled robot can follow a trajectory with an odometer and internal/external sensors [5]. Nevertheless, these methods are only available when there is a pre-defined map. If the robot is to go into an obscure vicinity, it needs to determine its position and be aware of its surroundings by itself. For these kinds of cases various mapping and path planning algorithms have been proposed also [6].

The problem of understanding the surrounding environment and robot's current position is figured out with SLAM (Simultaneous Localization and Mapping). SLAM is a method that builds up a consistent map and locates the robot on the created map [7,8]. While SLAM or the other known methods are used effectively for robot navigation, these methods are not interested in what are the surrounding objects and obstacles. An object can be classified as an obstacle or as a crossing way and both classifications may be used in path planning. If a robot is to use some specific crossing points or some objects as a temporary target, there is a need for another mapping algorithm including the capability of defining objects. This competence can be provided by object detection algorithms with the help of computer vision technologies.

Object detection and classification is a technology consisting of image processing and computer vision. The main purpose is to identify the objects in an image by associating them with a dataset. The recent methods have additionally started to look for semantic information to understand the scene. One of the biggest challenges for computer vision was processing time. Developers had to whether work on archived data and wait for the results for a while or use supercomputers for real-time application. For this problem, researchers have started to use data preprocessing and some useful methods to shorten processing time. The most notable contributions came from the convolutional neural networks (CNN) and GPU based computing power. Since convolutional neural networks produce very accurate results for detecting and classifying objects, it has become reliable. Moreover, parallel graphic processors work together to run neural network faster, so it is possible to use applications in real world even on developing boards. Besides, researchers have started to seek new mechanisms which speed up the CNN to work with less memory and fewer computational resources, such as compression and quantization of the networks [9].

A mobile robot which requires to determine its route in an unknown environment by using the particular objects nearby needs to have an object detection integrated SLAM method. This has not been used in any research according to the recent inquires. But it would be very useful for certain tasks such as creating a path using doors, windows, or stairs.

In this study, for mobile robot navigation in an unknown environment, to be able to use the object classification with SLAM, the usage of object detection algorithms has been inquired. As it can be seen from various studies, today's object detection and classification methods can be implemented in robots for real-time tasks. It is considered by detecting the crossing points indoor environment, a better understanding of the environment can be provided and robot's navigation can be planned with this information. By means of using object classification in navigation, a new contribution to path planning algorithms is targeted. However, running both algorithms simultaneously demands huge processing power which is not so easy to have in an indoor aircraft or a small mobile device. For this, it is crucial to find an accurate and fast method to perform in this limited capacity. Having knowledge of the advantages, drawbacks and limits of the detection algorithms, one can choose a proper method for new research. The solutions for robot positioning indoor, development of object detection and convolutional neural networks are reviewed in section two. In section three, the criteria to measure object detection algorithms' accuracy is explained and the best-known detection methods are compared to each other for their speed, processing power needs and detection accuracy considering implementing them on small developer boards. Detection algorithms' accuracy with the well-known datasets, their advantages and drawbacks are evaluated in the following

section. It is explained what has been learned, acquired and what is to be done in future works in the last section.

2. LITERATURE REVIEW

The appreciation of robots across the world and the latest research in robotics have started to make robots more apparent in daily life. Among the different types of modern robots, industrial robots and service robots promise a brighter future for now. One of the most basic and important skills for robots that we will see frequently in our near future, self-driving cars [10] and cleaning robots [11], is path planning and autonomous routing. The way the moving robots move from their current position to the target position with their own sensors and decision-making mechanisms has been tried in many different areas with different studies. It has been investigated how the robot can move on different types of surfaces like ground vehicles [12,13], underwater robots [14,15], wall-climbing robot [16] and aircrafts [17,18].

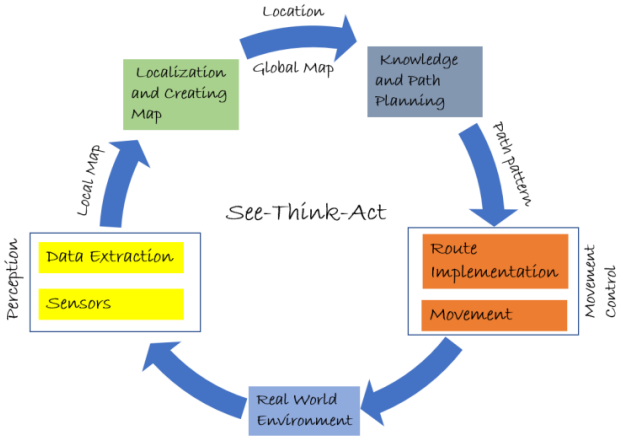

Figure 1 Autonomous robot's life cycle

Mobile robots usually have a similar life cycle which starts with taking data from the sensors (LIDAR, camera, IMU, etc.), extracting meaningful information from these data which results in having a local map. Next, it starts to model the unknown environment fostered with more sensor data which makes it possible to locate itself on the global map. Subsequently, it starts to draw a pathway to the target and sends commands to the moving parts of the robot to go further. This cycle keeps going until the robot's ultimate position. When the robot realizes that there is another obstacle or crossing point, it requires to update its route.

For robot navigation, many various methods have been proposed until. Patle et al. classified path planning algorithms into two categories [6]. The traditional methods comprised of Cell Decomposition [19], Roadmap Approach [20] and Artificial Potential Fields [21] and on the other hand, reactive methods consist of Genetic Algorithms, Fuzzy Logic [22], Neural Networks [23], Particle Swarm Optimization [24], Ant Colony algorithm [25] and the other biological modeling algorithms. In recent years, due to the capacity of exploring the environment, efficient calculation, rapid reaction, resiliency in the operation and capability to decide by itself, reactive methods are generally preferred [6].

While it is very possible to locate a robot by Global Positioning System, it is not robust in the buildings. An autonomous mobile robot in a building needs to obtain its position, map the environment, update it simultaneously to ensure loop closures and abstain from wasting time. In an unknown environment, the problem of building up a map and localization simultaneously with the help of sensors is known as SLAM [7,8]. Modern SLAM approaches' architecture can be seen in Figure 2. Following the sensor data was fused and processed, the graph structure is constructed. After the controlling of data connections and loop closures, the graph optimization is made, and metric and topologic maps are built up. In the closed areas, LIDAR has been the primary sensor for SLAM and robot navigation [26]. Besides, low-cost mono and stereo optic cameras have also been preferred for SLAM under the name of VSLAM [27,28].

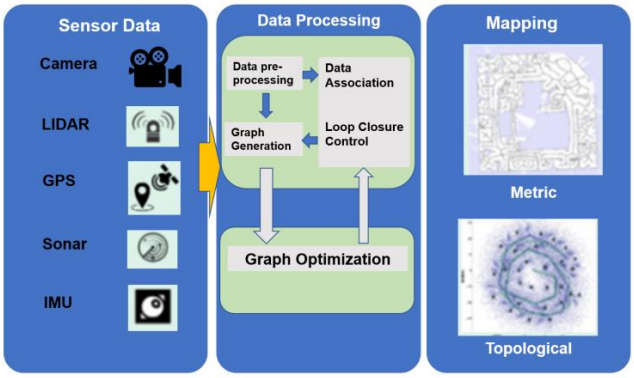

Figure 2 A typical SLAM architecture for generating metric or topologic maps

Apart from SLAM, detection, and classification of objects through images, videos and live cameras have been attracting so much interest in recent years thanks to its increasing success rate and practical applications. The methods and algorithms are in the scope of research fields in academic communities and companies working with computer vision related technologies. There have been many technological developments in the last decade for this common interest and making it very effectively usable in real world applications. Without a doubt, robotic researchers benefit from the advantages of computer vision and object detection. The latest developments made it possible to use them for security applications, robotic vision, analysis of territories, medical diagnoses and many independent categories including even for finding the rotten potato in a factory.

The first steps of computer vision came up with the projects in the 1960s which tried to mimic human visual using artificial intelligence [29]. Studies in the 1970s included current methods such as extraction of edges, motion prediction and optical flow [29]. As a major successful work, Fischler [30] achieved to detect roughly certain shapes such as faces with template matching in the 80s. Following researches generally had used geometric representations for object detection until the 1990s [31], which later evolved into statistical methods (Artificial Neural Networks [32], SVM [33], Adaboost [34]). This period is to be considered until 2012 when convolutional deep neural networks were successfully implemented [35].

A Convolutional Neural Network (CNN or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery [36]. Convolutional Neural Network (CNN) was first proposed for image recognition in computer vision by LeCun et al. [37] in 1989 to recognize handwritten postal codes and later used widely in image recognition and classification tasks. There is also an important study of Kunihiko who used a similar model with a different name in 1980 [38]. Later, CNN has been used in classifying handwritten numbers [39], recognizing address numbers [40], determining traffic signs [41] and in various other studies. However, the high computational cost and memory consumption of deep neural networks prevented its use in small memory devices and latency sensitive applications. Especially the emergence of SVM and Bayesian models and the fact that they can work with smaller data sets (MNIST, Caltech-101) with fewer parameters has reduced the use of CNN. With the use of much larger data sets after the 2000s, the use of deep networks has become feasible.

The great success of Krizhevsky et al. [35] for object classification in ILSVRC 2012, has been another milestone. The success of Krizhevsky's classification by labeling 1.2 million images in the Imagenet dataset with the benefit of data augmentation techniques led to the wide use of CNN. Realizing the potential of CNN in image classification, many researchers have studied to understand CNN and apply it to traditional computer vision tasks. The achievements and lessons learned in these studies have contributed to CNN and computer vision science.

Many models have been proposed with the use of convolutional neural networks for object recognition and classification. R-CNN [42-44], one of the two-level algorithms, divides the image into regions by means of the features in the image and performs the object classification and bounding process on these regions. Mask R-CNN [45] is a convolutional network model reinforcing R-CNN. There are other studies to enhance the performance of R-CNN such as the Faster R-CNN [44] which increases the number of frames per second (fps) to be processed. However, although

these two-level models have a good detection success, the detection speed is relatively low and the memory consumption is high. This fact makes two-level detection algorithms almost impossible to use in mobile robots considering the existing processor technologies especially on small platforms that can carry very limited load.

To have a lower computation cost and faster results, one level models have been proposed. Compared to the two-level models, YOLO [46-48], SSD [49], DSSD [50] and RetinaNet [51] algorithms appear to be able to detect objects much faster. This relatively high speed comes from the simpler and shorter algorithms.

While conventional convolutional networks have N connections for each N layer, the network of DenseNet has N(N+1)/2 direct connections. The former layers' feature maps are used as inputs for subsequent layers. In this way, it considerably reduces the number of parameters and provides higher performance results. [52]. SSD model is quite simple compared to methods that need object proposals, since it completely eliminates the proposal stage, feature resampling phase and covers all computations in a single network. This model ensures to be easy for training and compatible with detection systems [49]. RetinaNet is comprised of two task-specific subnetworks and one backbone network which are united in a single network. Backbone network computes feature maps for input image while subnet networks are responsible for classifying objects and bounding with a box. It provides a simple, one-stage model detection [51]. YOLO ignores the wide pipeline which optimizes individual components. It focuses on directly being faster without giving up detection accuracy and it succeeds to be fast by its structure [53]. The simple network provides 45 frames per second, while the fast version can reach more than 150 fps. YOLO takes the input image as a whole both in training and test and thereby it figures out contextual information about categories. As it is said highly generalizable [53], when applied to new domains or unexpected inputs, it is still successful.

These new methods use CNN architectures as the network backbone which determines the depth of the network, the number of layers and parametric values of the network structure. Although the depth of the network directly affects the calculation performance, it seems the trend in recent years is on deepening the network. While AlexNet [35] has 8 layers, GoogleNet [54] announced in 2015, has 22 layers, ResNet and DenseNet [57,52] are over 100 layers. Another popular network Darknet, which is frequently engaged with YOLO, has two versions with 19 and 53 layers [47,48].

Open source deep learning tools such as Caffe [58], TensorFlow (from Google) [59], CNTK (from Microsoft) [60], Torch [61] have also been offered for CNN, which are successfully used today in topics such as object recognition [54], image classification [35,56], motion detection [55] and natural language processing [62,63].

Looking to the usage of computer vision for mobile robots' navigation, in numerous works, visionary data has been used for path planning. Moghadam et al. [64], combined the 2-dimensional laser data with stereo camera data. In this way, they succeeded to perceive 3-dimensional buildings. Sabe et al. [65], benefited from the stereo camera to avoid obstacles and move around on different surfaces. However, to be successful, it must have sufficient texture knowledge. The biggest disadvantage of using the camera as a sensor is there may be deviations in distance data on homogeneous environments such as a flat wall and in low light environments. In such cases, texture validation and surface validation techniques are applied [66]. Pomerleau handled path tracking as a classification problem and succeeded in detecting the deviation of the vehicle from the road line with the system he trained with artificial neural networks [67]. Ran et al. trained the images using CNN which are gathered with a spherical camera and they ensured that the robot turned to the correct route by determining how many degrees it was traveled from the desired direction [68]. Hadsell and his friends labeled the environment by dividing it into 5 classes in order to provide a pathway on off roads [69]. Surfaces such as trees and buildings are classified as obstacle or super obstacle, surfaces such as soil and asphalt as ground or

super-ground, and navigation is planned on the land where the robot can go between obstacles. In [70], a digital surface model is created using the images taken by an unmanned aerial vehicle to identify the vine rows and inter-row terrain. This model is then used to generate a path plan for unmanned ground vehicle Many other methods also successfully classified the environment as obstacles, targets or corridors, yet they didn't look for what exactly is the surrounding objects. Understanding what the crossing point is and moving to that point in accordance with this information may be helpful for the robot's navigation.

3. COMPARISION OF OBJECT DETECTION METHODS

In the context of the study, the most popular object detection methods are inquired and compared for their success, performance and working principles. To be able to compare successfully, there is a need to use standard datasets and measuring tools which are also defined below.

3.1. How to measure accuracy and error

One of the most important steps in designing artificial neural networks is to evaluate the performance of the system or in other words to measure errors to be able to minimize them. The final goal of all detection algorithms is reducing the loss. The loss value has to be calculated before various strategies are implemented to diminish it. A basic loss function (Mean Absolute Error) in a classification or regression operation can be calculated using Eq.1 when the loss function of a single training data is *l* and the dataset is x.

$$L(x, W) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{l}(x_i, \mathrm{W}) \qquad (1)$$

Mean squared error as one of the most known methods to measure the error calculates the sum of the differences between expected output and actual output as seen in Eq.2.

$$L = \frac{1}{N}\sum_{i=1}^{N} ||o_i - y_i||^2 \qquad (2)$$

Softmax. is a method that is usually used for multiclass problems and frequently preferred in image classification. It receives input data from the preceding fully connected layer and uses it to classify. It takes probabilistic input data and determines belonging value to a certain class [71]. Below loss function calculates the cross- entropy of softmax with the output of the neural network where y_j is output and p_j is the estimated probability vector:

$$L = -\sum_j \; y_j - logp_j \qquad (3)$$

The success of object recognition and classification algorithms is measured with their accuracy and speed. However, there is more than one metric used for the concept of accuracy. These metrics are important for a proper understanding of the success to be considered separately. TP (True Positive), truely located targets; FP (False Positive), mistakenly detected objects; FN (False Negative), undetected objects; β (threshold), the probability of the prediction; Recall, the proportion of correctly found objects to all real objects; Precision refers to the sensitivity of the true detections in total predictions. To calculate Precision and Recall Eq.4 can be used as shown below.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP+FP}},\ Recall = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \qquad (4)$$

The proportion of overlapping predicted bounding box with the actual minimum bounding box gives IOU (Intersection Over Unit) value and can be obtained for the predicted bounding box b and the expected box b_g by the Eq.5:

$$\mathrm{IOU}(b, b^{\mathrm{g}}) = \frac{\mathrm{area}(b \cap b^{\mathrm{g}})}{\mathrm{area}(b \cup b^{\mathrm{g}})} \qquad (5)$$

For object recognition algorithms, TP, FP and FN values are determined according to the IOU value. According to the threshold value of the datasets or the one determined by the researcher, when the IOU value is higher than the given threshold the case is accepted as a TP. Average Precision (AP), calls for the use of Precision and Recall together for a category, whereas mean Average Precision (mAP) refers to the accuracy value for all categories. FPS, which is the number of frames that can be processed per second, is also used as

an important metric for real time detection algorithms.

3.2. The Role of Datasets

In research of object detection, datasets played a vital role. They helped to solve both the complex problems and measuring and comparing the different detection algorithms' performances. Looking the datasets used for detection methods, 4 significant datasets outshine: PASCAL VOC (proposed by Everingham et al. in 2010 and upgraded in 2015 [72,73]), MS COCO [74], ImageNet [75] and Open Images [76]. In Table 1, it is possible to see the main features of these datasets and highlights. Moreover, many other studies have played an important role in the object detection field, such as Caltech [77] and KITTI [78].

PASCAL VOC dataset has the most faced 20 categories of daily life. The number of image counts for every class is high and possible to use it in real-life applications. It contains more than one object with different features in an image and also uncommon examples. ImageNet outshines with the high number of categories and images. MS COCO is developed for real life applications and has more categories and images than VOC. In every image, there are many images and it is available for segmentation which is not comprised in ImageNet. Lastly, Open Images dataset released in 2017 and it supports big scale object detection, object extrication and visual correlation.

Table 1
The comparison of well-known datasets

Dataset	Number of Input Data	Number of Category	Object number in each image	Image Size	Release Date
PASCAL VOC (2012)	11540	20	2.4	470×380	2005
ImageNet	More than 14 million	21841	1.5	500×400	2009
MS COCO	328000+	91	7.3	640x480	2014
Open Images	More than 9 million	6000++	8.3	Various	2017

3.3. The Comparison of Methods for Using in Robot Navigation

Considering mentioned criteria for mobile robots and calculations of the object detection algorithms, when the most known object detection algorithms have been compared, the results in Table 2 have been acquired.

From numerous detection algorithms, R-CNN[42] is the first one in which CNN was integrated into RP (Region Proposal) methods and it came up with considerably better results. However, calculation cost is high, training and test time are long. SPPNet [79], is the first usage of SPP (Spatial Pyramid Pooling) in CNN and it provided speeding up of R-CNN. Nevertheless, it has similar drawbacks with R-CNN. The first method training the network end-to-end without region proposal is Fast R-CNN [43] in which one pooling layer is suggested. By the way like its name it is much faster than SPPNET and quite successful. However, external RP calculation creates a bottleneck and it is still too slow for real-time applications.

With Faster R-CNN [44], the RPN (Region Proposal Network) was suggested instead of selection sort to create high quality and zero cost RPs. Convolution layers were shared and RPN and Fast R-CNN were merged in a single network. It showed a processing speed of 5 FPS with VGG16 network model. Speaking of disadvantages, the training process is quite difficult and it doesn't have enough speed for real time applications.

Authors use an insignificant region generation scheme, constant for each image in R-CNN-R [80] method. Combined with SPP, this provides a fast detector that does not require processing an image with algorithms other than the CNN itself. It showed the community that it is possible to use simple and fast algorithms with CNN. However, its detection speed is lower than 5 FPS and it is still inadequate for real-time applications and it sometimes has unsatisfactory detection results owing to the failure of detecting regions. RFCN [81] used a fully connected convolutional neural network. Without sacrificing detection accuracy,

detection speed is raised (~10 FPS). Its shortcomings are training process is long and difficult and yet not satisfactory for real-time applications. Mask RCNN [45] is announced as a simple, resilient and efficient object segmentation. The common usage of bounding box was attached with object mask by upgrading Faster RCNN. Anyway, its detection speed was around 5 FPS which is not suitable for real-time applications.

YOLO [46] is the first efficient single layer object detection algorithm. It totally abolishes the RP process which results in high speed and makes it possible to use it in real-time applications. Its biggest disadvantages are the detection accuracy which is lower than modern detection methods and being unsuccessful for small object detection. With YOLOv2 [47], A faster background structure, DarkNet19, was suggested which provided a more accurate and fast detection. Nonetheless similar to its preceding it is unsuccessful for detecting small objects.

Table 2
The comparison of popular detection algorithms

Detection Method	Backbone Structure	Size of Input Image	Results with Different Datasets		FPS
RCNN (2014) [42]	AlexNet	Constant	58.5 (PASCAL VOC07)	53.3 (PASCAL VOC12)	<0.1
SPPNet (2014) [79]	ZFNet	Optional	60.9 (PASCAL VOC07)		<1
Fast RCNN (2015) [43]	AlexNet VGGM VGG16	Optional	70.0 (PASCAL VOC07)	68.4(PASCAL VOC12)	<1
Faster RCNN (2015) [44]	ZFnet VGG	Optional	73.2 (PASCAL VOC07)	70.4 (PASCAL VOC12)	<5
RCNN⊖R (2015) [80]	ZFNet +SPP	Optional	59.7(PASCAL VOC07)		<5
RFCN (2016) [81]	ResNet101	Optional	80.5 (07+12) 83.6 (07+12+CO)	77.6 (07++12) 82.0 (07++12+CO)	<10
Mask RCNN (2017) [45]	ResNet101 ResNeXt101	Optional	50.3 (ResNeXt101) (COCO Dataset)		<5
YOLO (2016) [46]	GoogleNet	Constant	66.4 (PASCAL VOC07)	57.9 (PASCAL VOC12)	<25
YOLOv2 (2017) [47]	DarkNet	Constant	78.6 (PASCAL VOC07)	73.5 (PASCAL VOC12)	<50
SSD (2016) [49]	VGG16	Constant	76.8 (PASCAL VOC07)	74.9 (PASCAL VOC12)	<60
YOLOv3(2018) [48]	DarkNet	Variable	79.26 (PASCAL VOC07)	57,9 (MS COCO)	<155 (Fast YOLO)
YOLOv4(2020) [82]	CSPDarknet 53	Variable	65,7 (MS COCO)		<120

SSD [49] is a single layer successful detection method. It is both benefited from the ideas of YOLO and region proposal methods. By the way, multi-scale convolutional layers are extracted. It is faster (around 60 FPS, while it is lower than 50 FPS in YOLOv2) and more accurate than YOLOv2. It is used in many studies and researches and obtained successful results. Yet, it is not proper to say it is successful for small objects.

YOLOv3 [48], is quite faster than SSD (claimed to be 155 FPS) and lessened the weakness of YOLOv2 for detecting small objects. Even though approaching the claimed speed means sacrificing detection accuracy, it is quite accurate and fast compared to other single level models. The tests with the same datasets in the same processors produce highly superior results.

In 2020, YOLOv4 [82] has been announced. The usage of CSPDarknet53 improved the learning

capability of the network. A spatial pyramid pooling block was added to CSPDarknet53 to enhance the perceiving of the area and extract the best context features. Instead of the Feature Pyramid Networks (FPN) used in YOLOv3 for detecting objects, PANet is used in various detection levels as adding parameters method. It is possible to see that YOLOv4, drives YOLOv3's success up especially in the field of detection rate.

4. EVALUATION

The object detection methods are scrutinized to be able to understand which one is more suitable to use in a small mobile robot particularly a flying one. Since one of the constraints of mobile robots is carriage limits, it is substantially important to use a smaller developer board for determining surrounding objects. Although parallel graphic processors help to have a better computing ability, it is crucial to limit the demands from the developer board. Therefore, when looking over to methods one of the biggest criteria is the speed of the algorithm which slows down with the limited computing capabilities. The other criterion, of course, is how accurate it is over well-known datasets. Lastly, another important figure is input size which is to be bigger if it is needed to detect small objects. While it costs more computing power, for a route searching robot the far distant objects might be crucial.

For object classification, region proposal methods have offered the most accurate results and two-stage models have shown promising outcomes. However, in small mobile robots such as UAVs and rovers, it seems quite impossible to use them due to their detection speed. In a mobile robot, the algorithm needs to work in real-time with successful classifications. The very first attempts of one stage models were promising with detection speed but not satisfactory for their detection accuracy. However, in recent years it seems one stage models like YOLO, SSD and RetinaNet have made the detection and classification possible to use in mobile robots. These methods and variations did have nearly %80 detection accuracy with MS COCO and PASCAL VOC datasets. Even if this is not satisfying for critical decisions for a single image, it could be plausible for using real-time vision on a mobile robot.

A mobile robot equipped with a camera sensor displays the environment lively. When the robot is navigating, it approaches the objects from different angles and sometimes under various lightings. This is helpful to object detection if the method can't classify or falsely detects the object in the first frame. One of the disadvantages of recent fast algorithms, they have difficulties in detecting small-sized objects. Whereas it is quite advantageous making the neural network simpler and smaller, it is difficult to cope with small sized objects in the images. It is recommended to use a slower method or increase the input image size manually if sacrificing detection speed is possible.

Having a good detection rate by using only one convolutional network, YOLO appears to be the fastest algorithm among the current detection algorithms. As stated by Bochkovskiy et al. [82], when YOLOv4 is compared to its peers, it does not fall short in detection success while showing a clear better performance in object detection speed.

5. CONCLUSION

In this study, it is aimed to research object detection algorithms to be able to use for mobile robot navigation. Even if various computer vision methods have been used for robot navigation, there is no certain work on using the classification data in path planning as a crossing point. This is not so easy particularly with the limited computing capacity of small developer boards on mobile robots. To find a robust solution, it is needed to dig into detection methods for their computing power requirements, real-time detection speed and detection accuracy as well.

CNN and modified network models are popularly used by robotic researchers so as to determine what is surrounding the robot. These methods have been examined and their accuracy, speed and computational requirements have been compared in the context of the study. It can be inferred from the results that the latest algorithms show

dominance over older methods by detection speed. Among the contemporary methods, there are many superiorities, but the best method mostly depends on users' expectations.

For mobile robot navigation in an unknown environment, an algorithm running real-time on a developer board, though, it does not have the most precise detection rate, classifying various categories with a limited accuracy is satisfactory for using object detection in path planning. In the study, the results of methods have been obtained with standard datasets. It is a good way to compare algorithms, yet it cannot guarantee that one can take the exact same results in different environments. For this research, it was tried to specialize in the comparison of their usage in small mobile robots. Consequently, it is thought that the most appropriate method is YOLOv4 which demands small computational power but easily classifies crossing points like windows, doors and ladders.

It is probable that newer algorithms have already been proposed for object detection. Hopefully, they will produce more accurate results with less computational needs. While this superfast development is still in progress, it is planned to experience the latest algorithms in our flying robot during its journey for navigation according to detection information and it is quite possible to compare the methods after having experimental results.

Funding

The authors has no received any financial support for the research, authorship or publication of this study.

The Declaration of Conflict of Interest/ Common Interest

No conflict of interest or common interest has been declared by the authors.

Authors' Contribution

The authors (a PhD candidate and thesis advisor) contributed equally to the study in the context of thesis- "Mapping and Path Planning of an autonomous flying robot in unknown closed environment with object classification".

The Declaration of Ethics Committee Approval

This study does not require ethics committee permission or any special permission.

The Declaration of Research and Publication Ethics

The authors of the paper declare that they comply with the scientific, ethical and quotation rules of SAUJS in all processes of the paper and that they do not make any falsification on the data collected. In addition, they declare that Sakarya University Journal of Science and its editorial board have no responsibility for any ethical violations that may be encountered, and that this study has not been evaluated in any academic publication environment other than Sakarya University Journal of Science.

REFERENCES

[1] Hassani, Imen, Imen Maalej, and Chokri Rekik. "Robot path planning with avoiding obstacles in known environment using free segments and turning points algorithm." Mathematical Problems in Engineering 2018 (2018).

[2] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", International Journal of Robotics Research, 5(1):90-98, 1986.

[3] J.F.Canny and J.H. Reif, "New Lower Bound Techniques for Robot Motion Planning Problems", Proceedings of the 28th IEEE Symposium on Foundations of Computer Science, pp. 49-60, Los Angeles, CA, 1987.

[4] R. A. Jarvis, "Distance Transform Based Collision-Free Path Planning for Robot", Advanced Mobile Robots, World Scientific Publishing, pp.3-31, 1994.

[5] Ganganath, Nuwan, and Henry Leung. "Mobile robot localization using odometry and kinect sensor." 2012 IEEE International Conference on Emerging Signal Processing Applications. IEEE, 2012.

[6] Patle, B. K., et al. "A review: On path planning strategies for navigation of mobile robot." Defence Technology (2019).

[7] J. Leonard and H. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91, no. 91, pp. 1442–1447, 1991.

[8] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," IEEE Transactions on Robotics, vol. 32, no. 6, pp. 1309–1332, 2016.

[9] Cruz, Nicolás, Kenzo Lobos-Tsunekawa, and Javier Ruiz-del-Solar. "Using convolutional neural networks in robots with limited computational resources: detecting NAO robots while playing soccer." Robot World Cup. Springer, Cham, 2017.

[10] Badue, Claudine, et al. "Self-driving cars: A survey." Expert Systems with Applications (2020): 113816.

[11] Seo, T., Jeon, Y., Park, C., & Kim, J. (2019). Survey on Glass And Façade-Cleaning Robots: Climbing Mechanisms, Cleaning Methods, and Applications. International Journal of Precision Engineering and Manufacturing-Green Technology, 6(2), 367-376.

[12] Jaradat, Mohammad Abdel Kareem, Mohammad H. Garibeh, and Eyad A. Feilat. "Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field." Soft Computing 16.1 (2012): 153-164.

[13] Rus, Daniela, Bruce Donald, and Jim Jennings. "Moving furniture with teams of autonomous robots." Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots. Vol. 1. IEEE, 1995.

[14] Yao, P., Zhao, Z., & Zhu, Q. (2019). Path planning for autonomous underwater vehicles with simultaneous arrival in ocean environment. IEEE Systems Journal, 14(3), 3185-3193.

[15] M. P. Aghababa, "3D path planning for underwater vehicles using five evolutionary optimization algorithms avoiding static and energetic obstacles," Applied Ocean Research, vol. 38, pp. 48–62, 2012.

[16] R. Yue, J. Xiao, S. L. Joseph, and S. Wang, "Modeling and path planning of the city-climber robot part II: 3D path planning using mixed integer linear programming," in Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '09), vol. 6, pp. 2391–2396, Guilin, China, December 2009.

[17] Shiri, H., Park, J., & Bennis, M. (2019, December). Massive autonomous UAV path planning: A neural network based mean-field game theoretic approach. In 2019 IEEE Global Communications Conference (GLOBECOM) (pp. 1-6). IEEE.

[18] Ropero, F., Muñoz, P., & R-Moreno, M. D. (2019). TERRA: A path planning algorithm for cooperative UGV–UAV exploration. Engineering Applications of Artificial Intelligence, 78, 260-272.

[19] Tunggal, Tatiya Padang, et al. "Pursuit algorithm for robot trash can based on fuzzy-cell decomposition."International Journal of Electrical and Computer Engineering 6.6 (2016): 2863.

[20] Ma, Xiaozhi, et al. "Conceptual framework and roadmap approach for integrating BIM into lifecycle project management."Journal of Management in Engineering 34.6 (2018): 05018011.

[21] J. Sun, J. Tang and S. Lao, "Collision Avoidance for Cooperative UAVs With Optimized Artificial Potential Field Algorithm," *IEEE Access*, vol. 5, pp. 18382-18390, 2017.

[22] Azzeddine Bakdi, Abdelfetah Hentout, Hakim Boutami, Abderraouf Maoudj, Ouarda Hachour, Brahim Bouzouia, Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control, Robotics and Autonomous Systems, Volume 89, pp. 95-109, 2017.

[23] Aleksandr I. Panov, Konstantin S. Yakovlev, Roman Suvorov, Grid Path Planning with Deep Reinforcement Learning: Preliminary Results, Procedia Computer Science, Volume 123, pp. 347-353, 2018.

[24] Manh Duong Phung, Cong Hoang Quach, Tran Hiep Dinh, Quang Ha, Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection, Automation in Construction, Volume 81, Pages 25-33, 2017.

[25] Liu, J., Yang, J., Liu, H. et al. An improved ant colony algorithm for robot path planning. Soft Comput 21, 5829–5839 (2017).

[26] Filipenko, Maksim, and Ilya Afanasyev. "Comparison of various slam systems for mobile robot in an indoor environment." 2018 International Conference on Intelligent Systems (IS). IEEE, 2018.

[27] C. Tao, Z. Gao, J. Yan, C. Li and G. Cui, "Indoor 3D Semantic Robot VSLAM Based on Mask Regional Convolutional Neural Network," in IEEE Access, vol. 8, pp. 52906-52916, 2020.

[28] Chen, L.; Jin, S.; Xia, Z. Towards a Robust Visual Place Recognition in Large-Scale vSLAM Scenarios Based on a Deep Distance Learning. Sensors, 2021.

[29] Szeliski, Richard. Computer vision: algorithms and applications. Springer Science & Business Media, 2010.

[30] Fischler, M., & Elschlager, R. (1973). The representation and matching of pictorial structures. IEEE Transactions on Computers, 100(1), 67–92.

[31] Mundy, J. (2006). Object recognition in the geometric era: A retrospective. In J. Ponce, M. Hebert, C. Schmid, & A. Zisserman (Eds.), Book toward category level object recognition (pp. 3–28). Berlin: Springer.

[32] Rowley, H., Baluja, S., & Kanade, T. (1998). Neural network based face detection. IEEE TPAMI, 20(1), 23–38.

[33] Osuna, E., Freund, R., & Girosit, F. (1997). Training support vector machines: An application to face detection. In CVPR (pp. 130–136).

[34] Viola, P., Jones, M. Rapid, "Object detection using a boosted cascade of simple features." CVPR, 1, 1–8, 2001.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, pp. 1097–1105, 2012.

[36] Valueva, M.V.; Nagornov, N.N.; Lyakhov, P.A.; Valuev, G.V.; Chervyakov, N.I. (2020). "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation". Mathematics and Computers in Simulation. Elsevier BV. 177: 232–243.

[37] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E.Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to

handwritten zip code recognition. Neural computation, 1(4):541–551, 1989.

[38] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4): 93-202, 1980.

[39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradientbased learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[40] P. Sermanet, S. Chintala, and Y. LeCun. Convolutional neural networks applied to house numbers digit classification. In Pattern Recognition (ICPR), 2012 21st International Conference on, pages 3288–3291. IEEE, 2012.

[41] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In Neural Networks (IJCNN), The 2011 International Joint Conference on, pages 2809–2813. IEEE, 2011.

[42] R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587, 2014.

[43] R. Girshick, "Fast R-CNN IEEE International Conference on Computer Vision", IEEE, vol. 2015, pp. 1440-1448.

[44] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, June 2017.

[45] K. He, G. Gkioxari, P. Dollár, R. Girshick, "Mask R-CNN", 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980-2988, 2017.

[46] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified Real-Time Object Detection", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, 2016.

[47] J. Redmon, A. Farhadi, "YOLO9000: Better Faster Stronger", 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517-6525, 2017.

[48] J Redmon, A. Farhadi, YOLOv3: An Incremental Improvement, 2018.

[49] W Liu, D Anguelov, D Erhan et al., SSD: Single Shot MultiBox Detector, pp. 21-37, 2015.

[50] C Y Fu, W Liu, A Ranga et al., DSSD: Deconvolutional Single Shot Detector, 2017.

[51] T Y Lin, P Goyal, R Girshick et al., "Focal Loss for Dense Object Detection", IEEE Transactions on Pattern Analysis & Machine Intelligence, no. 99, pp. 2999-3007, 2017.

[52] Huang, G., Liu, Z., Weinberger, K. Q., & van der Maaten, L. (2017a). Densely connected convolutional networks. In CVPR.

[53] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real time object detection. In CVPR (pp. 779–788).

[54] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229, 2013.

[55] Salman, Ahmad, et al. "Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system." ICES Journal of Marine Science, 2020, pp. 1295-1307.

[56] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In CVPR (pp. 1–9).

[57] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In CVPR (pp. 770–778).

[58] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014.

[59] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin et al., "Tensorflow: Largescale machine learning on heterogeneous systems, 2015," Software available from tensorflow. org, vol. 1, 2015.

[60] D. Yu, A. Eversole, M. Seltzer, K. Yao, Z. Huang, B. Guenter, O. Kuchaiev, Y. Zhang, F. Seide, H. Wang et al., "An introduction to computational networks and the computational network toolkit," Technical report, Tech. Rep. MSR, Microsoft Research, 2014, 2014. research. microsoft. com/apps/pubs, Tech. Rep., 2014.

[61] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlablike environment for machine learning," in BigLearn, NIPS Workshop, no. EPFL-CONF-192376, 2011.

[62] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. arXiv preprint arXiv:1411.4555, 2014.

[63] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. arXiv preprint arXiv:1412.2306, 2014.

[64] Moghadam, Peyman, Wijerupage Sardha Wijesoma, and Dong Jun Feng. "Improving path planning and mapping based on stereo vision and lidar." 2008 10th International Conference on Control, Automation, Robotics and Vision. IEEE, 2008.

[65] Sabe, Kohtaro, et al. "Obstacle avoidance and path planning for humanoid robots using stereo vision." IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004. Vol. 1. IEEE, 2004.

[66] PTGrey, 2004. Point grey research inc. http://www.ptgrey.com/

[67] Pomerleau, Dean A. "Efficient training of artificial neural networks for autonomous navigation." Neural computation 3.1 (1991): 88-97.

[68] Ran, Lingyan, et al. "Convolutional neural network-based robot navigation using uncalibrated spherical images." Sensors 17.6 (2017): 1341.

[69] Hadsell, R.; Sermanet, P.; Ben, J.; Erkan, A.; Scoffier, M.; Kavukcuoglu, K.; Muller, U.; LeCun, Y. Learning long-range vision for autonomous off-road driving. J. Field Robot. 2009, 26, 120–144.

[70] Zoto, J., Musci, M. A., Khaliq, A., Chiaberge, M., & Aicardi, I. (2019, June). Automatic path planning for unmanned ground vehicle using uav imagery. In International Conference on Robotics in Alpe-Adria Danube Region (pp. 223-230). Springer, Cham.

[71] Tang, Y. "Deep Learning using Linear Support Vector Machines." arXiv: Learning (2013):

[72] Everingham, M., Gool, L. V., Williams, C., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. IJCV, 88(2), 303–338.

[73] Everingham, M., Eslami, S., Gool, L. V., Williams, C., Winn, J., & Zisserman, A. (2015). The pascal visual object classes

challenge: A retrospective. IJCV, 111(1), 98–136.

[74] Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, L. (2014). Microsoft COCO: Common objects in context. In ECCV (pp. 740–755).

[75] Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Li, F. (2009). ImageNet: A large scale hierarchical image database. In CVPR (pp. 248–255).

[76] Kuznetsova, Alina, et al. "The open images dataset v4." International Journal of Computer Vision (2020): 1-26.

[77] Fei-Fei, Li, Rob Fergus, and Pietro Perona. "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories." 2004 conference on computer vision and pattern recognition workshop. IEEE, 2004.

[78] Geiger, Andreas, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012.

[79] He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV (pp. 346–361).

[80] Lenc, K., & Vedaldi, A. (2015). R-CNN minus R. In BMVC15.

[81] Dai, J., Li, Y., He, K., & Sun, J. (2016c). RFCN: Object detection via region based fully convolutional networks. In NIPS (pp. 379–387).

[82] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection." arXiv preprint arXiv:2004.10934 (2020).