Automating Machinery with Object Detection using YOLO and Servo Controllers

Riya Peter
B.E Computer Engg.
Trinity Academy Of Engineering
Pune,India
riya.peter47@gmail.com

Gillian Pereira
B.E Computer Engg.
Trinity Academy Of Engineering
Pune,India
pergill007@gmail.com

Yash Kamble
B.E Computer Engg.
Trinity Academy Of Engineering
Pune,India
291yashkamble@gmail.com

Dr. M. B. Wagh
B.E Computer Engg.
Trinity Academy Of Engineering
Pune,India
mukund.wagh.81@gmail.com

***Abstract*—Now-a-days Computer Vision and Machine Learning algorithms play an important role in automation. With the help of Computer Vision and deep learning algorithms, data like images and videos are being used for classification and prediction.**

This paper proposes a real time object detector using computer vision and deep learning algorithms. YOLO (You Only Look Once) which is a deep learning algorithm is a state-of-the-art algorithm used for object detection. A binary classifier using CNN (Convolutional Neural Network) can be used to detect whether a class is present or absent indicating the presence or absence of a particular object.

The two classes will be A) desired object present and B) the desired object absent. The input to the model will be from a live camera. The classifier detects the object by creating a bounding box around the object and then predicting the class. If class A is predicted the model will calculate the distance from the object. After calculating the distance, the model will instruct the machine to pick up the object and place it on the required position. If class 'B' is present the model will just display the message stating the class is absent.

After the detection of the correct object, this paper proposes using the Position Based Visual Servoing technique to pick and place the object to the desired location.

Keywords—component; formatting; style; styling; insert (key words)

I. INTRODUCTION

Today's world is progressing rapidly in order to automate nearly all systems and devices as much as possible. One of the most important fields which would be benefited by automation is Defence. Defence sector can make use of Machine Learning to reduce manpower, increase efficiency and reduce the time required to perform various tedious and time-consuming tasks.

The machines that are used by the armed forces all require an operator to handle and operate the machines. This can be completely avoided with the use of automation. The repetitive tasks of picking up and placing the object can be sped up after the object is detected. Hence, reducing manforce and increasing work rate while using the strength of officers elsewhere.

Classification is the process of being able to differentiate an input image on the basis of different features that were extracted and then compared to the training image features. Binary Classifications mean that our classifier is built to classify the images into two specified classes.

Deep learning, which is a subset of Machine Learning, aims to mimic the human brain. Using a neural network, we aim to classify objects in the image using a neural network. Neural networks work similarly to how neurons function in the human brain. A neural network is connected using nodes just like how neurons are connected in the brain, collect features then process the information, and recognize patterns in the data. The whole process helps to train our model by understanding and learning the patterns from the images which helps the machine to classify the input.

In this paper, we are proposing to build an object detection model which will state whether the object is present and if it is it will give a positive output. This positive output signifies that the object is present and hence be the input to the servo control model. The object detection model is a supervised learning model. Labelled data on the Positive class and Negative class will be provided.

The dataset will be created using our own images and some from various resources. The data will be labelled using a process called Image Annotation. It is a process of labelling the particular object in the image. Our model contains 2 classes, hence labels have to be provided for the objects in our dataset. The training data set will have the objects that belong to the two classes i.e Object Present and Object Absent. Each object will be labelled in the training data that is expected to be detected.

Based on the data given the model will be trained, and the model will learn its features and learn how to classify.

The output layer of the classification network will contain only two nodes. Depending on the node activated the class will be predicted.We propose implementing the YOLO (You Only Look Once) model for object detection. Object detection has two parts viz classification and localization. As discussed above, during classification the class of the object is predicted. Localization is the process of finding out the position of the object in the input.

First the object is localised i.e in which part of the frame is the image present, a rectangular bounding box is drawn around the object. Then the object classification is done by extracting features from the object.

YOLO is a state-of-the-art model that uses a backbone like DarkNet19, DarkNet53, as a classifier and YOLO for object detection. When the positive class is detected, the model will calculate the distance and trigger the pick-up of the object and place it in the desired location.

There are numerous pick up and place systems that use robot arm tools, actuators to make the movement, end effectors to grab the object, sensors and controllers to carry out the pick up and place operation. The pick up and place robot's basic components include a controller, manipulator, grippers and power source.[25]

The method this paper proposes required cameras instead of sensors to take the input. The camera will provide a visual stream that works like an eye of our system. The visual servoing method is used to detect, classify and hence calculate the trajectory of the arm for the pick and place.

We propose to use PBVS(Position Based Visual Servoing) for the latter part of the work viz the pick up and place system. PBVS is a part of visual servoing. Visual servoing consists of Image based and Position based visual servoing[23]. It comprises of image processing, robotics and control theory to generate the pick up and place movement.[23]

The position-based visual servo (PBVS) is used to control the error between the desired and actual poses and the end-effector's motion directly in a 3D space. [24]Hence using inverse kinematics to pick up the object from the desired location.

II. LITERATURE SURVEY

The field of object detection has advanced significantly. There are no boundaries to what computer vision and machine learning can accomplish paired with the new cameras and quicker machines.

Joseph Redmon, et al. introduced a unique methodology for seeing in their 2016 paper you simply Look Once: Unified, period Object Detection by Joseph Redmon et al. from the USA. Classifiers were employed in the past to try to object detection. As another, we tend to create mental object detection as a regression issue to spatially distinct bounding boxes and associated category chances. Bounding boxes and sophistication chances are directly foreseen by one neural network from complete pictures during a single assessment. Since the whole detection pipeline consists of one network, detection performance will be tuned from getting down to the finish.[1]

In 2017, Chao LI, et al. from Shanghai Jiao Tong University suggested an object select and place manipulation system based on visual servoing, where the entire system makes use of eye-in-hand visual servoing. The proposed paper adds the following three findings to the body of knowledge: 1) This work suggests a workable system for object pick and place manipulation. 2) For contour extraction and object recognition, an effectively constructed edge detector is employed. 3) Use the force sensor's feedback to determine whether the pick operation was successful.[2]

Wei Chen et al. conducted research and analysis on the subject of Picking Robot Visual Servo Control Based on Modified Fuzzy Neural Network Sliding Mode Algorithms in 2019. To do this, an apple-picking robot's manipulator joint angles and target positions are analysed kinematically and dynamically, and the sliding-mode control (SMC) approach is introduced into robot servo control in accordance with the properties of servo control. The variable structure of the sliding mode causes chattering, which may be effectively addressed by the fuzzy neural network control technique, which also improves the dynamic and static performances of the control system.[3]

In their 2019 study "A Comparative Study of State-of-the-Art Deep Learning Systems for Vehicle Detection," Chinese researchers Hai Wang et al. analysed the outcomes of various vehicle detection algorithms. On KITTI data, they compared the performance of R-CNN, R-FCN, SSD, RetinaNet, and YOLOv3. The recall rate and precision rate on the KITTI test set, the average precision on the KITTI test set, the fps, and other metrics were all utilised to compare the overall performance of the algorithms.[4]

In February 2022, Zicong Jiang et al changed the networking of YOLOv4 and also decreased the parameters for embedded devices' ease of use. First, the complexity of the computation is reduced by using two ResBlock-D modules in the ResNet-D network rather than two CSPBlock modules in Yolov4-tiny. In order to extract additional feature information about objects and decrease detection mistakes, it creates an auxiliary residual network block. Two consecutive 3x3 convolutions are utilised to create 5x5 receptive fields in the architecture of an auxiliary network in order to extract global features. Channel attention and spatial attention are also used to extract more useful information. Finally, it combines the backbone network and auxiliary network to create the upgraded YOLOv4-tiny's entire network structure.[5]

Faster R-CNN is significantly better than fast R-CNN, while both are still better than RCNN. Additionally, YOLO v3 is superior to single shot detector, whereas Faster R-CNN is superior to single shot detector. Prior to the creation of YOLO v3, SSD was the greatest. However, the most recent and effective method is YOLO v3, which is faster and better than SSD. YOLOv3 is incredibly quick and precise. As a result, we can recognise many objects more quickly using the YOLO v3 model and add custom images and labels to the datasets. This Yolo version 3 model is advantageous because it can detect items immediately and only detects them once.[18]

Each layer in the YOLO structure had features predicted using FPN(Feature Pyramid Network) in YOLO version 4. As a result of YOLO's detection of high resolution features, the issue of not catching small objects was resolved. Version 4 was created with the added goal of being able to recognise things accurately with just one GPU. Yolov4's learning produced the best result in our experiment because we also measured performance utilising one GPU.[19]

In this study, multiple street-level object detection techniques were examined. A modified version of the Udacity Self Driving Car Dataset was used to train and test five algorithms, including SSD MobileNetv2 FPN-lite 320x320, YOLOv3, YOLOv4, YOLOv5l, and YOLOv5s. The dataset was also enhanced by rescaling, hue shifting, and

adding noise to the image. In comparison to the other algorithms, the results showed that YOLOv5l is the most accurate method. Further research reveals that YOLOv5s is the best algorithm for real-time applications, such as self-driving cars, because it delivers reasonably accurate findings in a fair amount of time.[20]

The delta robot is used to pick up and place moving parts on the conveyor using vision-servo control. The vision servo programme was made in C++ to handle image processing and picture identification. Research is being done on edge detection methods like Canny and Sobel for application in image processing algorithms. The experimental results show that Canny edge detection outperforms the Sobel technique in this case. Finally, employing vision-servo technology, a single system that integrates image processing, image registration, and motion controller actions can automatically pick up and position objects on a conveyor. [26]

Reliable visual input (pallet pose) must be delivered at relatively wide distances because of the limited degrees of freedom, differential constraint mobility, and massive machine size. According to the research, a control architecture should have three main sub-systems: feedback motion control, path planning from the current pose to the pallet frame's origin, and pose estimation for estimating the posture of the body and fork. The pallet acts as the local ground fixed frame in this design, where decisions about plans and poses are determined. There is a logical structure for combining the wheel odometry/inertial sensor data with vision once the pallet has been recognised for the first time, therefore planning is only essential at that point. [27]

III. METHODOLOGY

The current mechanism of the system is shown below in Fig.1. The said mechanism is completely managed and operated by trained operators.

Various operations like giving the instruction to the machine to pick up, then to load, instructing the machine to come back to the position where the object is supposed to be placed and finally unloading the object requires the operators. This process is proposed to be automated using our methodology.

The system proposes to use the YOLO model, the architecture of the YOLO algorithm is studied and has to be modified to solve our problem statement. YOLO is a generalised object detector which is trained over millions of images in 20 classes in the PASCAL dataset and over 80 classes in COCO dataset. We propose this methodology for real time application through direct cameras. The model will be intensively trained on a custom dataset to achieve the results.

The work will be completed in two stages i.e the study and dataset gathering set and the implementation of the customised YOLO model.Since this work aims to build a model for binary object detection to check if the object we require is present or not and based on the output of the detection model our servo controller will pick up and place the object.

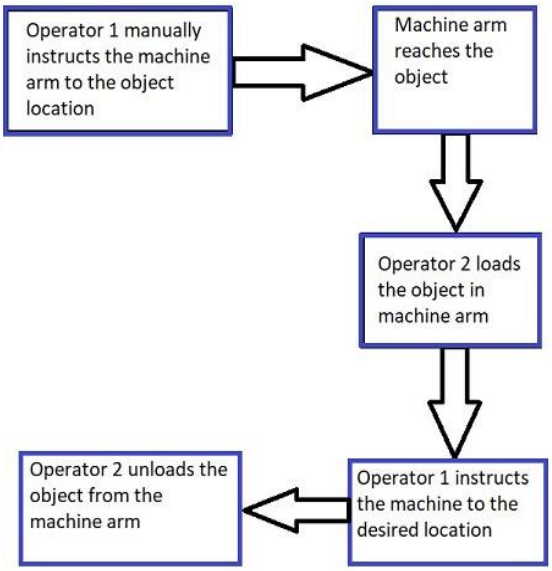

Fig. 1. Current Operation Mechanism

Our work proposes binary classification, hence it aims to have only two nodes in the output layer out of which one will get activated to predict the class based on probability.

According to the design of the model, the codes will be written and implemented. The necessary libraries will be installed. Then the code will be written in Python language. The dataset will be divided into training and testing.

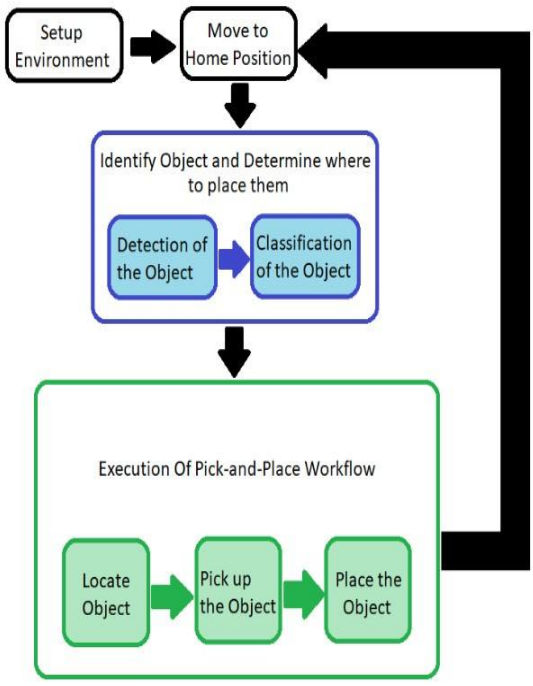

Fig. 2. Working of the proposed system

The work will be implemented in six steps:

1. Data collection.
2. Data annotation.
3. Implementation of YOLO on a customised dataset.
4. Metrics and visualization
5. Optimization and Retraining
6. Pick up and Place system

A. Data collection

For personalised object detection, it is important to collect data that is more suitable to the user's needs. Data will be collected from various sources such as the internet, databases and even taking pictures/ videos of our object.

Dataset will be divided into two sets i.e training and testing and validation. Each one is individually important for choosing the best model for our system. The dataset will comprise of both images and videos for training.

B. Data Annotation

Data Annotation stage consists of labelling objects in the dataset as Object present or object absent. YOLO has a special format for labelling. Along with the label specifying the class of the object, for the YOLO model we also have to provide the coordinates of the object, object class as well as height and width of the anchor box.

C. Implementation of YOLO on custom dataset

All the versions of YOLO are already pre trained on Pascal VOC and COCO datasets which have their own classes which may or may not be of the object that we want our model to detect. Hence, we have to provide our dataset with their labels and retrain the model to make it ready for our use.

YOLO will be implemented in the python language. Using YOLO we train the dataset for classification and detection. The ground truth for calculating the IOU is provided as a text file with the dataset to our YOLO model.

D. Metrics, Visualization and Evaluation

Once we make the necessary changes and retrain our model, we have to validate and test our model. We will then calculate the metrics and also visualise the results.

The necessary metrics that we need are precision, recall, time taken by the model, mAP (Mean average precision), F1 score graph. The testing dataset will be used for testing after the first implementation. Once we have trained and tested we use metrics to decide if our model needs changes.

The metrics will help us understand and record our results, and also understand how to optimize the model. The recorded results and visualization techniques will help decide the parameter necessary for better results.

E. Optimize and Retrain

This is an iterative step. After evaluation of the model, it becomes necessary to optimize the model. Based on what we learnt from the results we will make changes in our current model and retrain it.

This step will be repeated till we observe significant results and are satisfied. Object detection can be extremely crucial in many application systems. Hence, in order to make the system reliable we have to check conditions like overfitting and underfitting that may hamper the results. These conditions can be resolved with changes in the dataset.

F. Pick up and Place System

We propose to make use of a visual servoing system to pick up and place the object. PBVS (Positional Based Visual Servoing) is a model based technique that makes use of a single camera. The camera attached to the system will be used to detect the object, classify and if the object is the desired object then the pose of the object is calculated with respect to the camera.

Since we are aware of the locations of the joints, we can utilise the geometric relationship to determine the position of the end-effector, which is known as forward kinematics. The input command solution can be derived using backward kinematics if the position of the end-effector is known; this method is known as the backward solution.

This estimated distance is then sent forward to the robot controller which then controls the robot in our case the system. This part of the system extracts features as well to estimate the pose of the object in 3D space.

IV. ALGORITHMS

A. Convolutional Neural Network

Convolutional neural networks (CNN/ConvNet) are a class of deep neural networks used most frequently to interpret visual data in deep learning. For our model CNN will act as a backbone. Backbone of any object detection model is used for classification.

1) Input Layer:

The input layer accepts the image for classification in the form of a matrix. Our input will be in a form of live stream, hence a frame of the stream will be taken as input.

2) Hidden Layers:

The hidden layers in the CNN are feedforward networks which take the input. The hidden layers further consist of 4 layers:

a) The convolutional layer:

The convolutional layer is the first layer and the most important layer as its function is to extract the varied features from the input images. A matrix filter of a specific size n*n is used to perform mathematical operation of convolution between the input image and the filter. By sliding the filter over the input image, we take the scalar product between the filter and therefore the parts of the input image with reference to the dimensions of the filter (MxM).

The output is called the Feature map which provides us information about the features of the image like the corners and edges. After it passes through the convolutional layer, it is passed on to the pooling layer.

b) The Pooling Layer

This layer is specifically used to reduce the dimension of the image by removing the portions of the image which are not needed while preserving the important characteristics. It is mostly placed between two consecutive convolutional layers. By performing the pooling operation we reduce the parameter and hence the calculation.

c) ReLU (Rectified Linear Unit)

ReLU alludes to the real non-linear function defined by ReLU(x)=max(0,x). The ReLU correction layer substitutes all negative values received as inputs as zeros. It behaves like an activation function.

d) The Fully Connected Layer:

Fully connected layers form the last few layers of CNN. It consists of weights and biases and connects the neurons using an activation function to predict the final output. First, the input image received is flattened before feeding it to the Fully Connected Layer where it produces a new output vector. The classification happens at this stage, the output vector is of size K where each K has a probability value of each class that the object can belong to.

3) Output Layer:

The output is given using the highest probability of the class that the object can belong to.

B. YOLO (You Only Look Once)

Yolo is an object detection algorithm that is a one-stage detector and hence in one step itself detects and hence localises the objects in the frame. To classify an object first you need to localise the object by constructing a correct bounding box around it. YOLO treats the object detection problem as a regression problem and follows the following steps:

a) **Dividing the image into grids:** The first step is to divide the frame into S*S grids. These grids each individually predict if an object is present or not. They give the probability of an object being present in a grid.

b) **Bounding Box Regression:** all the grids with a probability greater than zero are used to further construct a bounding box.

c) **Intersection Over union:** In many cases, there can be multiple bounding boxes that can cover the object but those boxes often are not the correct ones. IOU helps us choose a correct bounding box by calculating the ratio of the area that overlaps with the area of the union. A threshold is set and bounding boxes with IOU greater than equal to are considered.

d) **Non-maximal suppression:** While displaying the output only one box should be displayed, hence non-maximal suppression rules out all other boxes except the box with the highest IOU.

e) Different versions of YOLO like YOLOv2, YOLOv3 use backbones, which is the classification model like Darknet-19[1], Darknet-53[13],FPN i.e Feature Pyramid Network has been used as the neck in YOLOv3[15].

i) YOLO consists of a backbone, a neck and a head. The backbone is a pretrained model that is used to extract features of the input. It forms a feature map using the backbone network.

ii) The neck of a system is used to connect the backbone and the head and hence used as a medium to aggregate the feature maps that were obtained from the backbone.

iii) The head of the model is used to process the feature maps and hence predict the bounding box, object score, etc.

1) YOLOv4

YOLOv4 is the fourth member of the YOLO family model that uses CSPDarkent-53[12] as the backbone. It takes a single-scale object as input and provides a proportionally sized output. Spatial Pyramid Pooling(SPP) and Path Aggregation Network(PANet) are used as the neck[15] in YOLOv4.

PaNet(Path Aggregation Network) [16] helps to boost the path through which the information regarding the features travel in the neural network, it enhances the feature hierarchy and shortens the information path between the lower layers and topmost layers using bottom-up path augmentation.

For the head YOLOv4 makes use of YOLOv3 i.e. single shot object detectors. YOLOv4 is different to other YOLO models because of Bag of Freebies[21] [12] and Bag of Specials[12] . These two techniques enable the algorithm to improve training strategy and training cost so as to receive better accuracy. Some of the methods that are used in these two techniques are data augmentation, semantic distribution bias in datasets.

New data augmentation techniques, such as Mosaic and Self-Adversarial Training, were introduced in YOLOv4 (SAT). mosaic combines four practice pictures. Self-Adversarial Training has two steps, one forward and one backward. The network just modifies the image in the first step, not the weights. The network is trained to recognise an object on the changed image in the second stage.

Bag of freebies helps to change the training strategy thereby might increase the training cost whereas in bag of specials it might increase training cost but also results in significant increase in accuracy of model.

In order to increase accuracy and improvement of mode BoF and BoS in the CNN makes changes in the activation layer, bounding box regression loss, data augmentation, regularisation method, Normalisation of the network activations by their mean and variance and skip connection.[12]

2) YOLOv5

Yolov5 almost resembles Yolov4 with some of the following differences:Yolov4 is released in the Darknet framework, which is written in C. Yolov5 is based on the PyTorch framework. Yolov4 uses .cfg for configuration whereas Yolov5 uses .yaml file for configuration.

The CSP—Cross Stage Partial Networks [22] are used as a backbone to extract rich in informative features from an input image. CSPNet has shown significant improvement in processing time with deeper networks.

PANet[15] is utilised in YOLO v5 as a neck to obtain feature pyramids. Pyramids' features may be found out more

about. The head of the YOLO v5 model is identical to the heads of the YOLO V3 and V4 models.

Any deep neural network's selection of activation functions is extremely important. Many activation functions, such Leaky ReLU, mish, and swish, have recently been introduced. The Leaky ReLU and Sigmoid activation function was chosen in YOLO v5.

In YOLO v5, the final detection layer uses the sigmoid activation function while the middle/hidden layers use the Leaky ReLU activation function. Binary Cross-Entropy with Logits Loss function from PyTorch is used to calculate the loss calculation of class probability and object score.

C. Servo Controllers

With the help of YOLO, we will be able to successfully detect and classify the object. Once it is also classified, it can be further used to track/locate the object in the environment and pick it up and place it in the desired location.

We propose position based visual servo control, which is one of the two classes of vision based robot control. Here, the target's relative pose and orientation with respect to the camera will be taken into consideration to calculate the *pose*[17]. This will directly give us the trajectories that will be used for the end effector frame. The *pose* of the object with respect to the camera is determined by three position parameters and three orientation parameters. In this, first we map object feature points onto the image plane. As a result of which, we get classical photogrammetric equations which are a function of *pose parameters*. Next, we apply the kalman filter which provides an implicit recursive solution of pose parameters over time. These kalman filters also facilitate the use of redundant feature points and provide temporal filtering of the solutions. A *pose vector* is in terms of roll, pitch and yaw angles.

Furthermore, it involves taking into account how the mistake was made(error calculation) . After establishing a connection to the joint actuation variables, the vector is placed in the robot's end-point frame. This creates the control system architecture, which may then be utilised as the foundation for a variety of control design methodologies, such as traditional PID designs.

With this, we can locate the position of the object and calculate the trajectory for our arm to move there and pick up the object, place it in the desired position and come back to the home position.

V. CONCLUSION

The CNN family, YOLO versions and SSD are the modern object identification algorithms briefly covered in this work. YOLO versions have more sophisticated uses in real life compared to CNNs and SSD. It is easy to build and can be trained on complete images. All YOLO versions are trained on a loss function that directly relates to detection performance, in contrast to classifier-based techniques, and the entire model is trained concurrently. In various detection datasets, YOLO offers state-of-the-art object detection performance in terms of real-time speed and excellent accuracy. Furthermore, YOLO is the best model for applications that require quick, accurate object recognition because it generalises objects better than other models. YOLO has the ability to work on extensive datasets which is of great importance for the detection of objects. With YOLO bettering its performance and overcoming the limitations of the previous versions, it is safe to say that the versions of YOLO are the best suited for real-time object detection.

REFERENCES

[1] You Only Look Once: Unified, Real-Time Object Detection Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788

[2] Li, Chao Cao, Chu-qing Gao, Yun-feng. (2017). Visual Servoing based object pick and place manipulation system: Selected Papers from CSMA2016. 10.1515/9783110584998-036.

[3] Chen, W.; Xu, T.; Liu, J.; Wang, M.; Zhao, D. Picking Robot Visual Servo Control Based on Modified Fuzzy Neural Network Sliding Mode Algorithms. Electronics 2019, 8, 605. https://doi.org/10.3390/electronics8060605

[4] H. Wang, Y. Yu, Y. Cai, X. Chen, L. Chen and Q. Liu, "A Comparative Study of State-of-the-Art Deep Learning Algorithms for Vehicle Detection," in IEEE Intelligent Transportation Systems Magazine, vol. 11, no. 2, pp. 82-95, Summer 2019, doi: 10.1109/MITS.2019.2903518.

[5] Jiang, Z., Zhao, L., Li, S., & Jia, Y. (2020). Real-time object detection method based on improved YOLOv4-tiny. *ArXiv*, *abs/2011.04244*.

[6] Y. Lu, L. Zhang and W. Xie, "YOLO-compact: An Efficient YOLO Network for Single Category Real-time Object Detection," 2020 Chinese Control And Decision Conference (CCDC), 2020, pp. 1931-1936, doi:10.1109/CCDC49329.2020.9164580.

[7] T. -H. Wu, T. -W. Wang and Y. -Q. Liu, "Real-Time Vehicle and Distance Detection Based on Improved Yolo v5 Network," 2021 3rd World Symposium on Artificial Intelligence (WSAI), 2021, pp. 24-28, doi: 10.1109/WSAI51899.2021.9486316.

[8] Xu, J., Li, Z., Du, B., Zhang, M., Liu, J. (2020). Reluplex made more practical: Leaky ReLU. 2020 IEEE Symposium on Computers and Communications(ISCC)doi:10.1109/iscc50000.2020.9219587 10.1109/ISCC50000.2020.9219587

[9] Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., . . . Summers, R.M. (2016). Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. IEEE Transactions on Medical Imaging, 35(5), 1285–1298. doi:10.1109/tmi.2016.2528162

[10] Jan Hosang, Rodrigo Benenson, Bernt Schiele; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 45074515

[11] J. Yang, X. Fu, Y. Hu, Y. Huang, X. Ding and J. Paisley, "PanNet: A Deep Network Architecture for Pan-Sharpening," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 1753-1761, doi: 10.1109/ICCV.2017.193.

[12] Bochkovskiy, Alexey & Wang, Chien-Yao & Liao, Hong-yuan. (2020). "YOLOv4: Optimal Speed and Accuracy ofObjectDetection".doi:https://doi.org/10.48550/arXiv.2004.1093

[13] Redmon, Joseph and Farhadi, Ali, "YOLOv3: An Incremental Improvement", arXiv, 2018, 10.48550/ARXIV.1804.02767

[14] Lin, Tsung-Yi and Dollár, Piotr and Girshick, Ross and He, Kaiming and Hariharan, Bharath and Belongie, Serge, " Feature Pyramid Networks for Object Detection", arXiv, 2016, doi : 10.48550/ARXIV.1612.03144

[15] Nepal, Upesh & Eslamiat, Hossein. (2022). Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. Sensors. 22. 10.3390/s22020464.

[16] Liu, Shu and Qi, Lu and Qin, Haifang and Shi, Jianping and Jia, Jiaya, "Path Aggregation Network for Instance Segmentation",arXiv, 2018, 10.48550/ARXIV.1803.01534

[17] W. J. Wilson, C. C. Williams Hulls and G. S. Bell, "Relative end-effector control using Cartesian position based visual servoing," in IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pp. 684-696, Oct. 1996, doi: 10.1109/70.538974.

[18] John, Anand & Meva, Dr. Divyakant. (2020). A Comparative Study of Various Object Detection Algorithms and Performance Analysis. INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING. 8. 158-163. 10.26438/ijcse/v8i10.158163.

[19] J. -a. Kim, J. -Y. Sung and S. -h. Park, "Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition," 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), 2020, pp. 1-4, doi: 10.1109/ICCE-Asia49877.2020.9277040.

[20] Naftali, M. G., Sulistyawan, J. S., & Julian, K. (2022). Comparison of Object Detection Algorithms for Street-level Objects. *arXiv*. https://doi.org/10.48550/arXiv.2208.11315

[21] Zhang, Zhi and He, Tong and Zhang, Hang and Zhang, Zhongyue and Xie, Junyuan and Li, Mu, Bag of Freebies for Training Object Detection Neural Networks, arXiv, 2019, 10.48550/ARXIV.1902.04103

[22] Wang, Chien-Yao and Liao, Hong-Yuan Mark and Yeh, I-Hau and Wu, Yueh-Hua and Chen, Ping-Yang and Hsieh, Jun-Wei, CSPNet: A New Backbone that can Enhance Learning Capability of CNN, arXiv, 2019, 10.48550/ARXIV.1911.11929

[23] Pomares, Jorge. (2019). Visual Servoing in Robotics. Electronics. 8. 1298. 10.3390/electronics8111298.

[24] Dong, Gangqi & Zhu, Z. H.. (2015). Position-based visual servo control of autonomous robotic manipulators. Acta Astronautica. 115. 10.1016/j.actaastro.2015.05.036.

[25] Premkumar[1], S & Varman[2], K & Rajendren, Balamurugan. (2016). Design and Implementation of multi handling Pick and Place Robotic Arm. International Journal of Environment and Sustainable Development.

[26] C. -J. Lin, J. Shaw, P. -C. Tsou and C. -C. Liu, "Vision servo based Delta robot to pick-and-place moving parts," *2016 IEEE International Conference on Industrial Technology (ICIT)*, 2016, pp. 1626-1631, doi: 10.1109/ICIT.2016.7475005.

[27] M. M. Aref, R. Ghabcheloo, A. Kolu, M. Hyvönen, K. Huhtala and J. Mattila, "Position-based visual servoing for pallet picking by an articulated-frame-steering hydraulic mobile machine," *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 2013, pp. 218-224, doi: 10.1109/RAM.2013.6758587.