Detection and segmentation of robot hands and objects selected for picking

Muraleekrishna Harikumar Sreeja
Student Number: 210696370
m.harikumarsreeja@se21.qmul.ac.uk
Project Supervisor: Lorenzo Jamone
MSc. Artificial Intelligence, QMUL

***Abstract*— The relentless advancement of machine learning in various spectrums of life from discernable technology, multimedia, and healthcare technology to agriculture, supermarkets, and house interior improvements has become the mainstream discussion as well as the area of research in the scientific community. Computer vision, being an inevitable component of artificial intelligence utilizes machine learning and a deep-learning framework. The present level of computer vision assists in detecting and tracking of individual objects (faces, people walking, automobiles) in an unconstrained environment. Object detection and segmentation is the process of accurately identifying and classifying every item visible in the image frame and assigning a class label to each pixel. By performing instance segmentation at the pixel level, we attempt to develop automated object detection in this study. This is performed by picking the 420 images at random and separating them into a training set and a test set. After that, all these photos were labelled using the LabelImg toolbox. Then, using transfer learning with "coco" weights, these images were trained using a mask R-CNN. The proposed Mask RCNN recognizes both robot hands and food objects, marking them with bounding boxes, class labels, and masks. For each identified item, the confidence scores generated by the Mask-RCNN method are shown. Most of the pictures demonstrate that the recognized items have high confidence levels.**

***Keywords*— mask rcnn, object detection, segmentation, robot hands**

I. Introduction

Computer vision is an interdisciplinary scientific topic and critical domain in the field of artificial intelligence where the computer is trained to interpret the visual world. It aims to comprehend and automate activities that the human visual system is capable of performing from an engineering standpoint. Computer vision tasks include techniques for gathering, processing, analysing, and comprehending digital pictures as well as methods for extracting high-dimensional data from the actual environment to create numerical or symbolic information or judgments [17]. In this context, understanding refers to the conversion of visual representations into descriptions of the real world that might trigger appropriate behaviour. This image comprehension may be thought of as the decoupling of symbolic information from visual data utilising models created with the help of geometry, physics, statistics, and learning theory. The theory behind artificial systems that extract data from images is the focus of the scientific field of computer vision. The visual data might be in the form of video sequences, numerous camera views, multi-dimensional data from a 3D scanner, or data from medical scanning equipment, among other formats. Computer vision is a technical field that aims to build computer vision systems using its ideas and concepts. The sub-domains of computer vision include scene reconstruction, object identification, event detection, video tracking, object recognition, 3D pose estimation, learning, indexing, motion estimates, visual servoing, 3D scene modelling, and image restoration[1].

In computer vision, object detection is a never-ending field of research where machine learning methods are used to support the rapid development of deep learning or machine learning. Object detection is the process of identifying and labelling the objects accurately in a given image frame. It is accomplished through two main steps namely object localization and image classification where a bounding box detects the accurate position of the object in the former and located objects are labelled in the latter step. Object detection methods are essentially subdivided into traditional and deep-learning-based methods. While traditional methods bank on visualisation characteristics, deep-learning-based methods utilise artificial neural networks and stratified computation to produce required features from the input given [2], [12].

Traditional object detection is normally regarded as those methods widely used before the advent of deep learning. Viola-Jones Detector set in motion the initial works of object detection in machine learning followed by HOG Detector, which implemented the prominent image processing and object detection in computer vision by introducing feature description. Later, the DPM detector initially introduced the feature of bounding box regression. An example of a traditional object detection method that works by using visual features is detecting a car. The shape of a car is identified by a rectangular body and circular wheels which are considered the features of the car. Therefore, detecting a structure that aligns with these features becomes an easy endeavour. Similarly, texture and colour are other differentiating features used to detect various objects[2].

Deep-learning methods comprise two-stage and one-stage object detection algorithms which are RCNN, SPPNet, Fast RCNN, Faster RCNN, Mask-RCNN, Pyramid Networks/FPN, GRCNN and YOLO, SSD, RetinaNet, YOLOv3, YOLOv4, YOLOR respectively. Object detection in deep-learning collect features from the image or video input and find a discretionary number of objects (sometimes including zero) and categorises each object and uses a bounding box to estimate its size.

In two-stage object detectors, bounding box regression is utilised for the object and deep features are used to suggest approximate object areas before these features are employed for classification. Region convolutional neural network (RCNN), including evolutions Faster R-CNN or Mask R-CNN, is one of several two-stage detectors and the granulated RCNN is the most recent development (G-RCNN). Without performing a step of region proposal, one-stage detectors forecast bounding boxes over the input images. Because this procedure takes less time, it may be applied in real-time scenarios. The YOLO (You Only Look Once), SSD (Single

Stage Detectors), and RetinaNet are the most widely used one-stage detectors. The most recent real-time detectors are YOLOR and YOLOv4-Scaled. Today, the majority of vision-based AI software and systems are built on object recognition and some of the real-time applications are object detection in retail where an AI-based customer analysis is used to detect customer interaction, animal detection in the agriculture industry to estimate and evaluate the quality of products, self-driving or autonomous cars to detect traffic signs, pedestrians and other vehicles, detecting people in security to prevent any dangerous encounters such as suicide attempt, vehicle detection using AI in the transportation sector, diagnosing diseases in healthcare. In RCNN [3], the Region Proposal Network is the backbone of its functioning and employs a bottom-up strategy to localise class segments in any given picture. It comprises three stages: creating region proposals, utilising CNN to extract characteristics, and classifying and localising items [12]. However, RCNN only offers a cursory localization of ideas, and to enhance these proposals, the regions' speed and accuracy are sacrificed [4].

The process of segmentation, which comes after object detection, involves creating bounding boxes for each item. It creates a border around each object and is aware of the pixel-level features rather than producing a box around the objects. Therefore, in semantic segmentation, each pixel in the picture is given a name and its appropriate class is determined and each class can have a different colour. Instance segmentation is a step more advanced than semantic segmentation. Instead of giving all objects of the same class identical pixel values, it tries to segment and display various instances of the same class. It does this by giving each instance of a class an instance ID and the result is an image with pixel borders separating each item.

This paper uses Mask R-CNN for object detection and instance segmentation on Crisp Teleoperated Fruit Picking Dataset. A team of Facebook AI researchers created Mask R-CNN, an object detection model based on deep convolutional neural networks (CNN), in 2017. Mask RCNN, being a deep neural network target to solve problems with instance segmentation in computer vision and machine learning, is often an extension of Faster RCNN with an extra branch for predicting segmentation masks for each Region of Interest (ROI). For each item that is spotted in an image, the model is capable of returning both the bounding box and a mask. Further, Mask RCNN is capable of differentiating images and videos and has high detection accuracy compared to other conventional target detection methods [8], [11].

II. Related Work

The three works that are mentioned below are some instances of works that are generally connected. This section discusses both the robotics field's usage of mask R-CNN-based object identification and related robotics-related applications.

A. Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot

The interminable growth of AI has contributed to the gradual automation in the agriculture industry. A model of a fruit-detecting robot vision detector based on Mask Region Convolutional Neural Network (Mask R-CNN) is suggested to more effectively utilise the strong feature extraction and target detection capabilities employed in deep learning to fruit identification in orchards. The region of interest (RoI) is created using feature maps as input to the Region Proposal Network (RPN), which then creates the mask to identify the apple-containing region. A random test set of 120 photos is used to evaluate the approach. Additionally, the identification speed is quicker and can satisfy the demands of the vision system on the apple harvesting robot. It demonstrates that the algorithm suggested in this study has a high identification accuracy and, if used in practice, may increase the overall effectiveness of the picking robot. This issue was effectively resolved by the case segmentation suggested by Mask R-CNN and according to the entity class in the image, it may categorise the picture [5].

B. Fruit detection for strawberry harvesting robot in a non-structural environment based on Mask-R-CNN

In this, Mask Region Convolutional Neural Network (Mask-RCNN) was used to boost machine vision performance in fruit recognition for a strawberry harvesting robot. To extract features, the Feature Pyramid Network (FPN) architecture was paired with Resnet50 as the backbone network. For producing region suggestions for each feature map, the Region Proposal Network (RPN) was completely trained. After Mask R-CNN produced mask pictures of ripe fruits, a visual localization technique for strawberry harvesting locations was used. Fruit identification findings from 100 test photos revealed high average detection precision rates and high recall rates. Mask R-CNN was able to extract object areas from the backdrop at the pixel level in an unstructured environment in addition to correctly classifying the categories (ripe or unripe fruit) and delineating object regions with bounding boxes. By examining the form and edge characteristics of the mask pictures produced by Mask R-CNN, it is also possible to visually locate strawberry harvesting places [6].

C. Target Detection Based on Improved Mask Rcnn in Service Robot

Here, a large selection of supermarket goods that were captured from both long and short is used as the test set with the intention of accelerating detection speed without sacrificing detection accuracy. Consider that the Mask R-CNN network's mask branch and excessive full connection layer will use a lot of network detection time, and the convolutional neural network's derived feature map's high dimension will require a lot of computational memory. This project, therefore, enhances the Mask R-CNN network by removing the mask branch, adding Light-Head RCNs to the Mask RCN network, increasing the R-CNN subnet and RoI warping, and adjusting the fraction of Anchors in the RPN network. The enhanced model is then used using the TensorFlow framework. These techniques can increase detection speed and conserve computer memory. Finally, a service robot platform using Kinect II has confirmed the enhanced Mask R-CNN network. The test results demonstrate that, in comparison to Faster R-CNN, the improved Mask R-CNN network can have a high level of detection accuracy; in comparison to the original Mask R-CNN network, the enhanced Mask R-CNN network can significantly increase algorithmic speed while maintaining detection accuracy. The target capture task of the service

robot is made more effective by a more than 2-fold reduction in detection time [7].

III. Proposed Method

Fig. 1 depicts the project workflow. To build the dataset required for this project, the primary dataset is sorted and annotated. Then, the data is divided into a training set and a test set. This is then trained using the proposed approach to provide the outcomes.

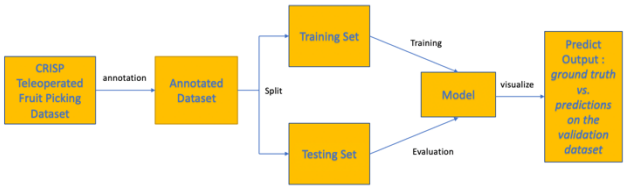

Fig. 1: Proposed workflow

A. Dataset Background

The CRISP (Cognitive Robotics and Intelligent Systems for the People) Teleoperated Fruit Picking Dataset is obtained from RoboPuppeteer, an economical bidirectional teleoperation system that uses leap motion-based hand tracking and a custom-designed vibrotactile haptic glove. The dataset sourced from RoboPuppeteer comprises images originating from a fixed RGB-D sensor, Kinect v2 (which is positioned on top of the boxes which accommodate the grocery items), the motion of the robot (that is, the 3D position of the hand and angles of the finger joints over time), motion of the human hand training the robot, tactile feedback from the robot fingertips (concerning the environment it interacts with).

B. Image Collection

Only the images obtained from the Kinect v2 are used for this study. Three grocery items, such as avocado, banana, and blueberry in a single and as well as cluttered manner are the major subjects of these images. For training and testing purposes, we have chosen 420 pictures at random from both the cluttered and the individual images of these grocery items. Out of which, 300 images for training and 120 images will be utilised for testing.

C. Data Annotations

Using LabelImg (shown in fig. 2), the selected images are then graphically labelled. LabelImg is a Python-based tool for graphical image annotation that makes use of Qt for its graphical user interface. The PASCAL VOC format is used to save the annotations as .xml files.

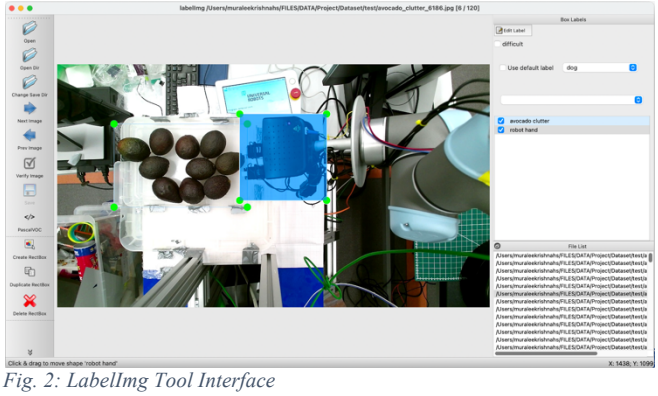

Fig. 2: LabelImg Tool Interface

D. Generating initial regions

Fig. 3 shows the training set's randomly chosen images from which the pixel masks have been extracted and shown using the mrcnn visualise function.

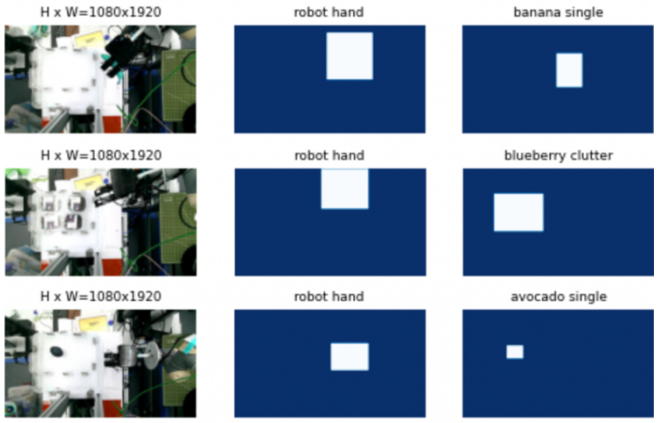

Fig. 3: visualizing pixel mask

E. Model Training

All the images together with their corresponding.xml files are made into a folder and uploaded to Google Drive when the annotations are completed. Then an array made up of all these images and annotations is parsed.

Mask RCNN, a model with instance segmentation capabilities, is the main model utilized for training. With a base learning rate of 0.002, the model first trained heads with higher learning rates to speed up learning. Each epoch took 25 steps. Following that, the model was trained with a base learning rate of 0.001 until the 10th epoch. The weights were imported from the "COCO dataset" model and the model was trained using transfer learning methods [10]. The MS COCO dataset is a large object identification, segmentation, and captioning dataset made public by Microsoft. Engineers in machine learning and computer vision frequently utilise the COCO dataset for a variety of computer vision applications. Since the image dataset was created with the intention of enhancing image recognition, it is known as COCO, or Common Objects in Context. The COCO dataset offers difficult, high-quality visual datasets for computer vision, primarily using cutting-edge neural networks. The COCO is frequently used to test algorithms and assess the effectiveness of real-time object detection. Modern neural network packages automatically comprehend the COCO dataset's format [14].

Transfer learning allows you to use feature representations from a model that has already been trained rather than having to create a new model from start. The pre-trained models are often developed using massive datasets, which serve as a common benchmark in the field of computer vision. Other computer vision tasks can use the weights derived from the models. These models can be included in the process of training a new model or utilised directly to make predictions on novel problems. The training time and generalisation error of a new model is reduced by incorporating previously learned models. When you only have a limited training dataset, transfer learning is highly helpful since you can utilise the weights from previously trained models to establish the weights of the new model.

The loss values recorded during training for each sequential epoch make up the training history. When evaluating our model, these loss numbers are crucial. Fig. 4 represents the overall loss metric. which is followed by,

- rpn_class_loss = RPN anchor classifier loss (Fig. 5).
- rpn_bbox_loss = RPN bounding box loss graph (Fig. 6).
- mrcnn_class_loss = loss for the classifier head of Mask R-CNN (Fig. 7).
- mrcnn_bbox_loss = loss for Mask R-CNN bounding box refinement (Fig. 8).
- mrcnn_mask_loss = mask binary cross-entropy loss for the masks head (Fig. 9) [15].

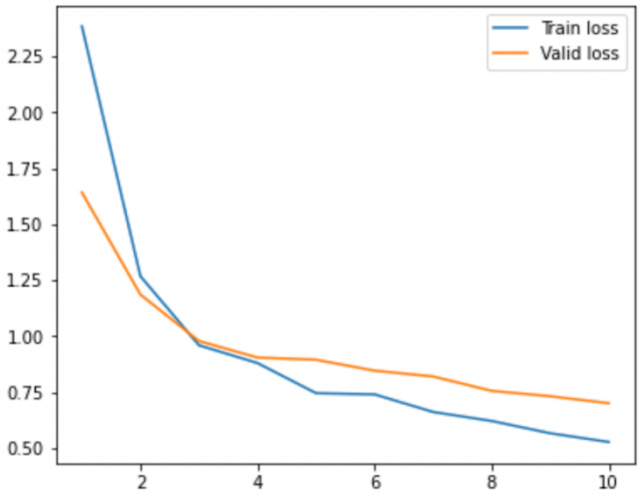

Fig. 4: overall loss metric

These loss metrics are the total of all the loss values that have been separately determined for each of the regions of interest. The total of these five losses is used to calculate the overall loss metric [8].

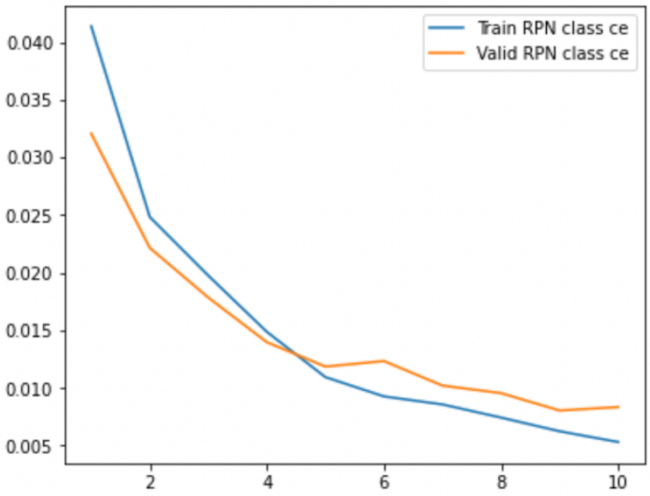

Fig. 5: RPN anchor classifier loss

Since the classification loss values are based primarily on the real class's confidence score, they serve as a gauge of how confident the model is in its ability to predict class labels or how near it is to it. All the object classes are considered in the case of mrcnn class loss, but only the anchor boxes are classified as foreground or background in the case of rpn class loss, which explains why this loss tends to have lower values because conceptually there are only "two classes" than can be predicted.

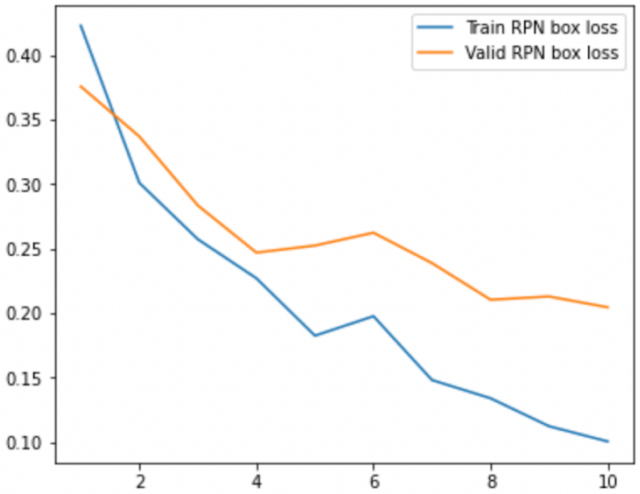

Fig. 6: RPN bounding box loss graph

The bounding box loss values represent the difference between the actual and predicted box parameters, which include the position of the box in (x, y) coordinates, as well as its width and height. It is a regression loss by definition and penalises larger absolute differences. Thus, in the case of rpn_bbox_loss, it demonstrates how effective the model is at locating objects within the image, and in the case of mrcnn bbox loss, it demonstrates how effective the model is at precisely predicting the area(s) within an image corresponding to the different objects that are present.

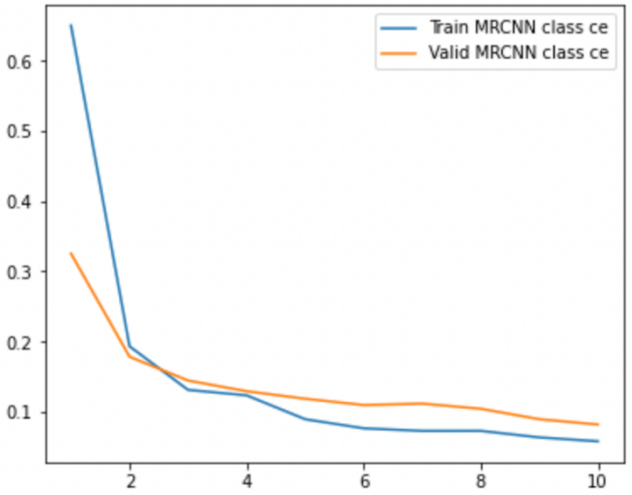

Fig. 7: loss for the classifier head of Mask R-CNN

The mask loss penalises incorrect per-pixel binary classifications (foreground/background, in relation to the true class label), much like the classification loss does. For each of the interest regions, it is determined differently: The mask loss for a particular RoI is computed using only the mask corresponding to its real class, preventing the mask loss

from being impacted by class predictions. Mask R-CNN encodes a binary mask per class for each of the RoIs.

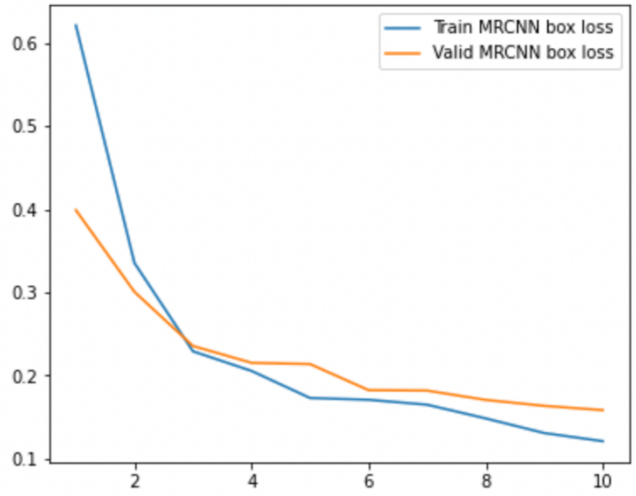

Fig. 8: loss for Mask R-CNN bounding box refinement

It might be challenging to make an accurate assessment based simply on the charts as fluctuations in the validation loss can happen for a variety of different causes. They may be brought on by either an insufficient validation set that provides unreliable loss values since small changes in the output can result in large changes in loss values or a learning rate that is too high, causing the stochastic gradient descent to overshoot when attempting to find a minimum [8].

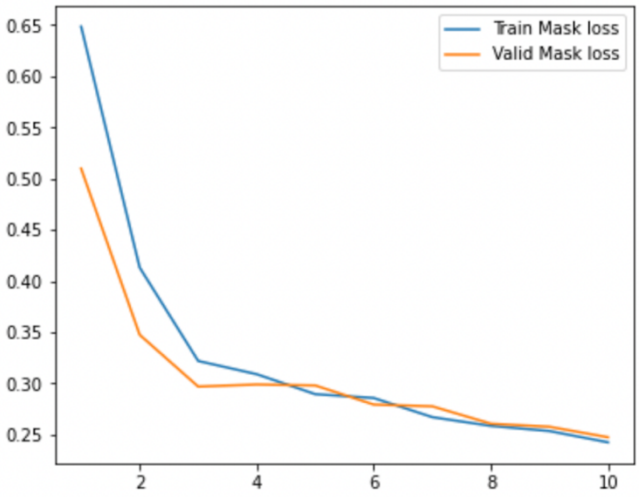

Fig. 9: mask binary cross-entropy loss for the masks head

IV. Models

A. *MASK RCNN*

The object shape masks are superior to bounding boxes for contours of objects, while instance segmentation is superior for illustrating regions of interest. They were inspired by a general Mask R-CNN for object identification and segmentation. In this study, robotic hands and the objects that were chosen for picking are detected and segmented using Mask R-CNN. The overall framework, as shown in Fig. 10, is made up of several components, including a Region Proposal Network (RPN), a Convolutional Neural Network (CNN) backbone with a Feature Pyramid Network (FPN), RoI (Region of Interest) features extraction using RoI align, bounding box regression, label classification, and mask prediction [8], [16].

Mask RCNN is a deep neural network that was created to handle instance segmentation, meaning it can recognise objects at the pixel level. Mask RCNN is capable of doing both instance segmentation and object detection. Since the goal of the study is to recognise objects at the pixel level in real-time, the Mask RCNN Algorithm is utilised. There are two training phases for Mask RCNN. First, regions, where objects could be present in an input picture, are proposed. These ideas are now utilised in the second step to predict the object class and create a bounding box around the object identified. The bounding box is improved and a mask is generated at the pixel level for the proposal in the initial stage. The backbone architecture is tied to both of these levels [10].

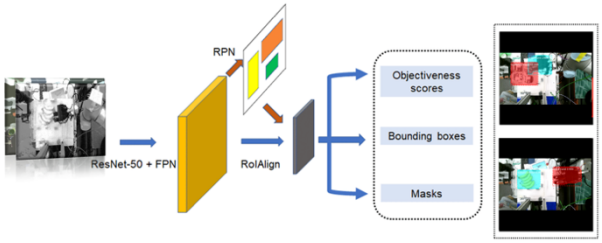

Fig. 10: The framework of object detection network based on Mask R-CNN.

The "RESNET50" backbone architecture (Fig.11) is what this project is built around. A feature pyramid network-style deep neural network is called a backbone architecture. A bottom-up method called "RESNET50" architecture is used to extract features from the input raw photos [10].

Using a region proposal network to suggest the regions where objects are expected is the first step. The ideas to link the features to their positions in the raw images are now used to create a feature map. Utilize anchors, a collection of boxes with predetermined locations and scaled dimensions relative to the images. The region proposal network utilises these anchors to determine the region of the feature map that should identify an object and what the bounding box dimension should be. Use anchors of varying sizes to link the various tiers of the feature map. The relative locations of items in the original image would be preserved during downsampling, upsampling, and convolving of the feature map [8], [10].

Consider another neural network in the second stage that receives suggested areas as input that are produced by the region proposal network in the first stage. A fully connected network (FCN) performs pixel-level categorization. A new neural network is currently using these proposals and assigning them to various specific areas of a feature map level. These regions are scanned, and bounding boxes, as well as masks, are utilized to create object classes. The "ROI align" (Region of interest) technique, which is used to pinpoint the pertinent regions of the feature map, is the major trick here.

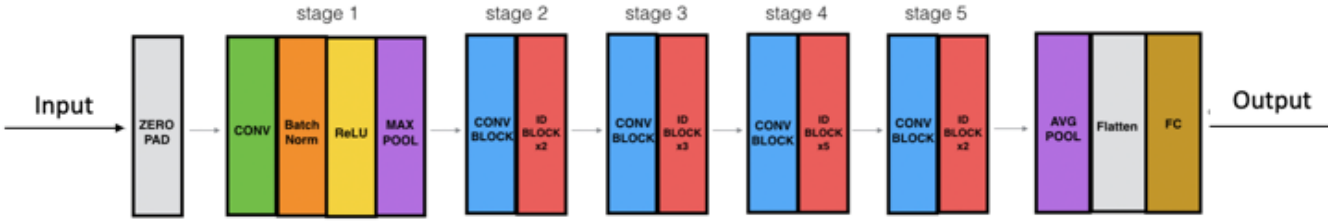

Fig. 11: RESNET50 Architecture

To create pixel-level masks for a picture, a branch of "ROI align" is utilised [10].

For the case quantization in "Faster RCNN," where the bounding box dimensions are converted to the appropriate dimensions, "ROI pooling" is used. There is no need for quantization while using "ROI align" for data pooling. The dimensionality must be altered throughout the mapping or pooling process, necessitating the usage of bilinear interpolation [10].

B. RESNET50

The vanishing gradient problem was the motivation behind the creation of the Resnet50 algorithm. In neural networks, the disappearing gradient problem is a situation where the loss is sent back to update node weights, but the calculated differential value is so minimal that the weights are no longer updated. In this situation, the model's loss does not diminish, and it loses its capacity to learn. The input and processed input are both transferred to the subsequent layer in RESNET models, and the propagation of loss also occurs at this point. "Skip connection" is the name for this idea (shown in fig. 12) [9], [10].

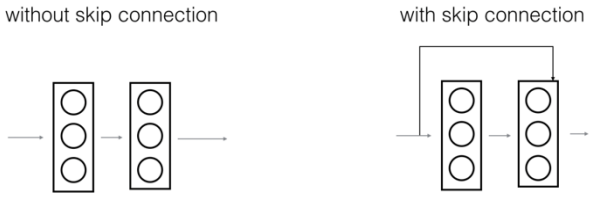

Fig. 12: Skip Connection

If f(x) is the function output and "x" is the input. Given that neural networks are effective to function approximators, they ought to be able to recognise a function where the output acts as the input [10]..

$$f(x) = x$$

Assuming this, the network should be able to forecast the function it was learning previously with the input added to it if the input of the initial layer of the model is skipped to be the output of the final layer of the model [10].

$$f(x) + x = h(x)$$

Through skip connections to a model's subsequent levels as well as from the final layers to the first layers, RESNET allows for the transfer of gradients. In addition to 1 "MaxPool" and 1 "Average Pool," Resnet50 uses 48 convolutional layers. A convolutional layer with a kernel size of 7*7*3 and a dimension of 64 precedes stride 2 in the order of operations. Then, "MaxPooling" is carried out, in which the maximum number of feature maps is extracted to reduce the size of the image, which also reduces noise. After that, the main Resnet layers are entered, where convolutions with a kernel size of 3*3*3 and a dimension set of 64, 128, 256, and 512 are performed. Whenever there are two convolutional blocks, there is a skip connection where the input and output values are constant [9], [10].

Dimensional changes, such as those from 64 to 128 during stride 2 of the skip connection, cause issues at that specific moment. To ensure that dimension change occurs without mismatches and mistakes, one convolutional block is in any instance added to the skip connection procedure. Following the traversal of all RESNET layers, "Average pooling" is carried out, which extracts the output as the average value from a certain feature map. This assists in image smoothing, however, it cannot be utilised for crisp images. The generated feature map is then flattened and sent to an artificial neural network after that. Forward propagation is the next step in the process when specific weights are applied to the inputs and conveyed to the next layers. Loss is computed at the output layer and transmitted back to the previous levels to adjust weights for the model's improvement [9], [10].

V. EXPERIMENTAL RESULT

For simple classifiers, precision is simple to calculate because it just considers True positive, False positive, True negative, and False negative. But even with classifiers, accuracy isn't necessarily the statistic that will result in the optimal model when it comes to additional operations like detection or segmentation. The average precision scores are used to gauge the outcomes for the instance segmentation issues. Calculate the "True Positive" (TP), "True Negative" (TN), "False Positive" (FP), and "False Negative" before measuring any categorization object (FN). Use the Intersection over Union (IoU) score to determine them during instance segmentation. Try to locate the area of intersection of the two masks and compute the area union of the two masks as a mask is being formed around the objects in the input picture and you have the ground truth of the object. To calculate the intersection over union score, these numbers are split.

The Intersection over Union score is now used in conjunction with a threshold, with the IoU score being regarded as a "True Positive" if it exceeds the threshold and a "False Positive" if it falls short of it. A "False Negative" is when an object is not identified, while a "False Positive" is when an object is incorrectly classified. The region covered by the bounding boxes and the ground truth used to get the IoU score for classification are shown in Fig. 13 [10]..

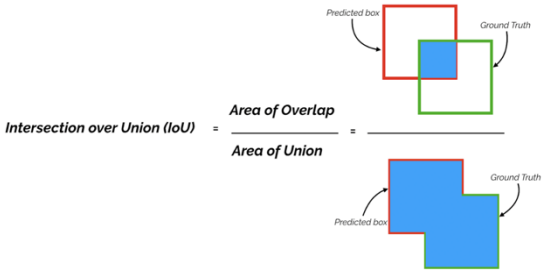

Fig. 13: Calculation of Intersection over Union

Precision measures how accurate the predictions are. It can otherwise be said that precision is the measure of true positive detections.

$$Precision = \frac{TP}{TP + FP}$$

TP = True Positives (Predicted as positive as was correct)
FP = False Positives (Predicted as positive but was incorrect)

The IoU serves as the initial statistic test to determine how accurate a model is. For instance, if a forecast's IoU value is 0.7 and the IoU threshold is 0.5, the prediction is classified as True Positive (TF). However, if IoU is more than 0.3, we label it as a False Positive (FP). Changing the IoU threshold allows us to obtain distinct binary TRUE or FALSE positives for a prediction.

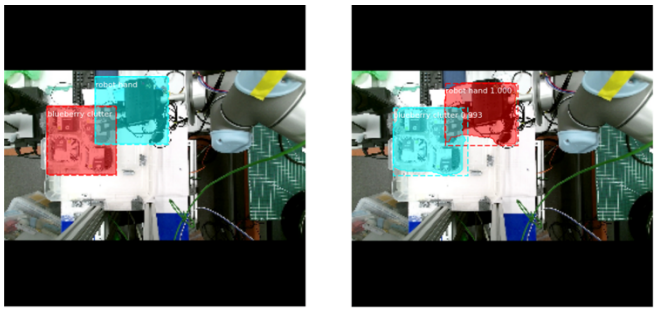

Fig. 14: Sample 1 of object detection result with the proposed approach

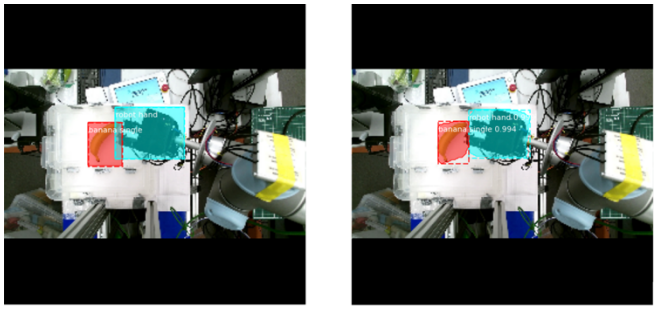

Fig. 15: Sample 2 of object detection result with the proposed approach

Fig. 14, 15, 16 and 17 compare the validation dataset's predictions to the ground truth. The confidence scores of classes discovered in various samples are displayed on the prediction image to the right. The confidence scores provided by the Mask-RCNN algorithm are displayed for each object that is discovered. Most of the images show that the identified objects have a confidence score of more than 0.8.

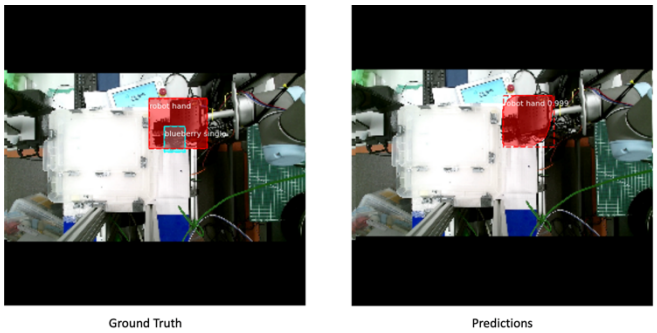

Fig. 16: Sample 3 of object detection result with the proposed approach

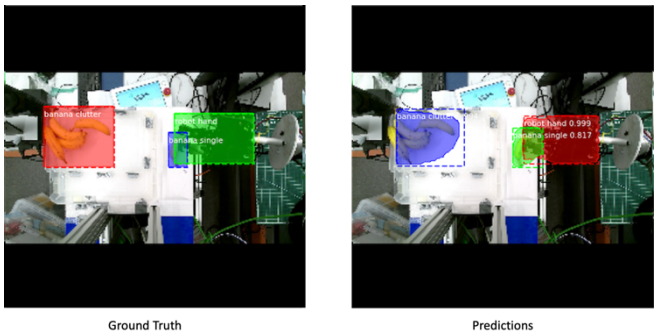

Fig. 17: Sample 4 of object detection result with the proposed approach

TABLE I. CONFIDENCE SCORE OF DETECTED CLASSES

Frame No.	***Classes***	***Confidence Scores***
1	Robot hand	1.000
	Blueberry clutter	0.993
2	Robot hand	0.999
	Banana single	0.994
3	Robot hand	0.999
	Blueberry single	Not detected
4	Robot hand	0.932
	Avocado Clutter	0.887
5	Robot hand	1.000
	Banana clutter	0.997
6	Robot hand	0.991
	Avocado single	0.987
7	Robot hand	0.999
	Banana clutter	0.994
	Banana single	0.817
8	Robot hand	1.000
	Blueberry single	0.982
9	Robot hand	0.977
	Blueberry clutter	0.897
10	Robot hand	0.991
	Avocado clutter	0.921

As the robot hand is above the blueberry in fig. 16 and as only a small portion of it is visible, we can see that the blueberry is not predicted in the outcome. Additionally, it was observed that in some samples when the model was trained with lower

epochs, the robot hand and some of the objects were found more than once. It was later resolved when the model was trained for a longer amount of time. Table I illustrates the various confidence scores we were able to get from the result obtained after the final training [13].

VI. Conclusion

In this study, the dataset generated from the camera mounted on top of the robotic hand RoboPuppetteer is used for object recognition and dataset segmentation. First, we divided the 420 photos into a training set and a test set after picking them at random. Then, using the LabelImg toolkit, we labelled all these images. These pictures were then trained with mask R-CNN by making use of transfer learning with "coco" weights. Mask R-CNN is implemented using Python 3, Keras, and TensorFlow. Each time an item appears in the picture, the model creates segmentation masks and bounding boxes for that occurrence. It has a ResNet50 backbone and Feature Pyramid Network (FPN) as its foundation. The entire Google Colab training took 8 hours. More epochs could have been performed and better results could have been produced if there was no runtime restriction. In any case, the result obtained was satisfactory.

VII. Future work

The dataset may be labelled using image pixel-by-pixel or polygonal annotation methods like V7, label or comparable sophisticated image annotation technologies. This will make it possible for the R-CNN model to produce segmentation masks and bounding boxes more precisely for each occurrence of an item in the images. Furthermore, using this study as the base it will be possible to take real-time images of the robot hand and pinpoint its precise coordinates for the things it is supposed to pick up. Depending on the items that have been detected, the robot hand could then be configured using the mask R-CNN inference result and action segmentation so that it can be operated autonomously using visual sensing.

Acknowledgement

First and foremost, I would like to extend my sincere gratitude to Professor Lorenzo Jamone, my supervisor and Senior Lecturer in Robotics at the School of Electronic Engineering and Computer Science (EECS) of the Queen Mary University of London, as well as the Director of the CRISP group. With his expert guidance and unwavering support, he has helped me gain a great deal of knowledge about my project. He has patiently and clearly explained each step of the project to me as he has guided me through it. Additionally, I want to express my gratitude to the CRISP team for allowing me to work on their Teleoperated Fruit Picking Dataset.

References

[1] Wikipedia Contributors (2019). *Computer vision*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Computer_Vision.

[2] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X. and Pietikäinen, M., 2020. Deep learning for generic object detection: A survey. *International journal of computer vision*, *128*(2), pp.261-318.

[3] Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).

[4] Girshick, R., 2015. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[5] Jia, W., Tian, Y., Luo, R., Zhang, Z., Lian, J. and Zheng, Y., 2020. Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot. *Computers and Electronics in Agriculture*, *172*, p.105380.

[6] Yu, Y., Zhang, K., Yang, L. and Zhang, D., 2019. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Computers and Electronics in Agriculture*, *163*, p.104846.

[7] Shi, J., Zhou, Y. and Zhang, W.X.Q., 2019, July. Target detection based on improved mask rcnn in service robot. In *2019 Chinese Control Conference (CCC)* (pp. 8519-8524). IEEE.

[8] He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

[9] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[10] Aparna, S., Muppavaram, K., Ramayanam, C.C. and Ramani, K.S.S., 2021. Mask RCNN with RESNET50 for Dental Filling Detection. *International Journal of Advanced Computer Science and Applications*, *12*(10).

[11] T. D. D and K. V, "Deep Learning based Object Detection using Mask RCNN," 2021 6th International Conference on Communication and Electronics Systems (ICCES), 2021, pp. 1684-1690, doi: 10.1109/ICCES51350.2021.9489152.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[12] Namjoshi, M. and Khurana, K., 2021. A Mask-RCNN based object detection and captioning framework for industrial videos. *International Journal of Advanced Technology and Engineering Exploration*, *8*(84), p.1466.

[13] Tahir, H., Khan, M.S. and Tariq, M.O., 2021, February. Performance analysis and comparison of faster R-CNN, mask R-CNN and ResNet50 for the detection and counting of vehicles. In *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)* (pp. 587-594). IEEE.

[14] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L., 2014, September. Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740-755). Springer, Cham.

[15] Kankane, A. and Kang, D., 2021, November. Detection of Seashore Debris with Fixed Camera Images using Computer Vision and Deep learning. In *2021 6th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)* (Vol. 6, pp. 34-38). IEEE.

[16] Xu, C., Shi, C., Bi, H., Liu, C., Yuan, Y., Guo, H. and Chen, Y., 2021. A page object detection method based on mask R-CNN. *IEEE Access*, *9*, pp.143448-143457.

[17] Ugwu, E.M., Taylor, O.E. and Nwiabu, N.D., 2022. An Improved Visual Attention Model for Automated Vehicle License Plate Number Recognition Using Computer Vision. *European Journal of Artificial Intelligence and Machine Learning*, *1*(3), pp.15-21.