Real-Time Deep Learning-Based Object Detection Framework

Ph.D. William Tarimo
Assistant Professor of Computer Science
Connecticut college
wtarimo@conncoll.edu

Moustafa M.Sabra
Physics and Computer Science Student
Connecticut college
msabra@conncoll.edu

Shonan Hendre
Computer Science Student
Connecticut college
shendre@conncoll.edu

***Abstract*— Recently real-time detection, and recognition of an object of interest are becoming vital tasks in visual data processing and computer vision. Various models have been deployed to implement object detection and tracking in multiple fields. However, conventional classifiers are often faced with challenging tasks that visual frames come distorted due to overlapping, camera motion blur, changing subject appearances, and environmental variations. Models using OpenCV-based HAAR feature-based cascade classifiers, without integrating any error minimizing object detection algorithm, were unable to accurately detect an object and track it in a changing environment. Therefore, developing an embedded powerful framework for real-time object detection and recognition becomes more of a vital need for future implementation in various fields. This study presents a powerful technique for a real-time detector that utilizes integrated Deep Learning Neural Networks (DNN) for optimal computational accuracy. Deploying such a framework will ensure the flexibility and reliability of the detector by eliminating the sources of distortion previously mentioned. The model relies on integrating the ImageAI deep learning libraries and You Only Look Once (YOLO-v3) object detection method with a DarkNet-53 architecture. The algorithm was trained using the TensorFlow framework to ensure accurate data processing. This paper targets one vital component of our long-term project of developing a multi-agent system, as the proposed model is to be implemented in autonomous agents for the detection of landmines, ocean debris, and wildlife beside environmental scanning missions. In this study the performance of the model has been assessed through detecting and collecting tennis balls as a preliminary test for real-world applications. The model was able to approach the desirable result of surpassing the accuracy of conventional detectors.**

Keywords—Deep Learning, Neural Networks, Object Detection, YOLOv3, Residual Networks.

I. INTRODUCTION

Witnessing a huge advancement in the field of machine learning especially in the fast-growing domain of deep learning is encouraging scientists around the world to implement such instruments for solving problems in computer vision. Such a transition holds the potential of adopting new technologies for detecting and recognizing objects in autonomous exploration and environmental scanning applications.

Similar to the human neural network, Artificial Neural Networks (ANN) are primarily composed of interconnected layers of neurons. The first layer, being the input layer, contains a set of neurons corresponding to an input feature. Each neuron takes a weighted sum of inputs that is calculated using a set of activating functions. Training an ANN simply means optimizing the weight for all neuron connections, which are decided using an algorithm known as backpropagation [2]. Deep learning aims to facilitate end-to-end optimization to increase the computational power exponentially. With the ability to learn multiple levels of representation corresponding to hierarchies of concept abstraction, deep learning allows the automation of tasks that requires computational power far more demanding than a human equivalent would deliver. One of the implementations of deep learning are object recognition and detection based on video streaming. Convolutional Neural Networks (CNN) [3] holds the ability to intuitively learn local features by processing input visual data through a series of convolutional and pooling layers. The convolutional layer aims to connect the output with the input as the visual data is processed in the pooling layer using a series of extraction algorithms and activation functions.

In this paper, ImageAI python libraries were utilized to perform object detection and recognition. ImageAI provides the simplest and powerful approach to training custom object detection models using the YOLOv3 [6] object detection models alongside Residual Network (ResNet) architecture. The final merged model (YOLOv3-ResNet) was trained using a TensorFlow framework that is generically inherited by the ImageAI python libraries. There are three main leading detection methods in the field of deep learning-based object detection and recognition; Single Shot Detector (SSD) [5], Faster Region CNN (F-RCNN) and You Only Look Once (YOLO) [4] with its two old versions. In the proposed framework, we choose the YOLOv3 algorithm because it is fast and compatible with the chosen architecture.

YOLOv3 algorithm matches objects based on applying default kernels of various aspects ratios. It utilizes multi-scale feature maps through generating output by applying a 1 x 1 detection kernel on a feature map of three different sizes at three different places in the network; therefore, offering a better precision with diverse scales while maintaining its incomparable processing speed. YOLOv3 algorithm mainly uses the new network structure of Darknet-53 [7], adopts logistics to replace the softmax function for object classification, and uses multi-scale features for object detection, all of which improve the performance under the premise of maintaining its confident detection speed. DarkNet-53 is organically composed of 53 layers, however YOLOv3 is supplied with 53 more layers stacked onto it, giving us a 106 layer fully convolutional underlying architecture enabling boasts of residual skip connections, and upsampling. YOLOv3 compatibility has been challenged through surpassing previous limitations in previous

978-1-7281-2547-3/20/$31.00 ©2020 IEEE

versions. Challenges such as decreasing the computational power along the processing speed of the network as it deepens and classifying features maps for multi-object detection prevented further progress in achieving high levels of accuracy. However, Increasing the complexity of the network through integrating the use of DarkNet-53 resulted in achieving relatively high precision on various datasets with maintaining a processing speed of 30 frames per second (FPS) [8].

ImageAI mainly relies on using Keras classes to extract feature maps for objects of interest. Keras is built upon a Residual Neural Networks (ResNet) CNN architecture that stacks up identity mappings, layers that initially do not do anything, and skips over them, reusing the activations from previous layers thus innately compressing the network into few layers enabling faster learning. By combining both the YOLOv3 detection algorithms and the ResNet NN architecture, we reach a fast and efficient object detection and recognition system based on ImageAI python libraries.

To analyze and assess the proposed module, the framework is to be implemented in a coordinated multi-robot system [15] for modular exploration and mapping. The multi-robot system mainly relies on using distinguished Robot Operating System (ROS) packages to facilitate effective data transmission between the system's components. The model will serve as a vital component for data collection and processing once deployed in multiple terrestrial and aerial agents. The framework's ability to carry on the task of tracking and recognizing objects of interest such as landmines and human individuals will be examined carefully through comparing the obtained results with the previously used HAAR feature-based cascade classifier. Parameters such as The Intersection Over Union matching (IoU) and the Average precision (AP) values will be used to evaluate the model's mean average precision (mAP) for future improvements.

II. Literature Review

Traditional object detection methods operate on the bases of three main stages. First region proposals are selected on a specific image. Then features are extracted from the region using feature descriptors such as histogram of oriented gradient (HOG), scale-invariant feature transform (SIFT) and HAAR. Finally, a series of trained classifiers utilizes these features in distinguishing objects of interest. Despite the simplicity of implementing these stages, the characteristics of manual design are vulnerable to change in diversity, which will directly affect the accuracy of the object detector. Subsequently, further methods of object detection have been proposed, but most were dependent on the structure of the deformable part model (DPM) [9], which features relatively complex structure with a slow processing speed. Therefore finding a tool to surprise these computational limitations became a vital task in object detection and recognition.

With the rapid improvement and innovation of research in the field of machine learning, deep learning algorithms provide significant improvement to object recognition and classification problems discussed earlier. Deep learning allows the assigned architecture to extract and learn features fastest to accurately identify the object from a ton of visual data. Various deep learning methods have been implemented in solving problems ranging from scene recognition problems [7], where researchers reached the accuracy of 94.42 ± 0.76 % with more than 7 million datasets, to object recognition and classification problems. Scientists have trained millions of datasets to recognize hundreds of object classes where most reached more than 80% of accuracy in both 3D and RGB descriptors with their proposed deep learning algorithm [8].

Some of the popular deep learning-based object detection methods are R-CNNs and Single Shot Detectors (SSD), and You Only Look Once (YOLO) with its various versions. The region-based Convolutional neural network (R-CNNs) proved to be efficient in detecting objects of interest. However, the training and testing on this method takes a long time and requires large space. Subsequently, Faster R-CNN [10] algorithms have been proposed to solve the problem of having insufficient space and slow processing speed. Faster R-CNNs is composed of two modules. The first module is a deep fully convolutional network that proposes regions, and the second module is the Fast R-CNN detector. Even with the faster implementation of R-CNNs, the algorithm could only process 7 frames per second, failing to achieve the desired processing speed. Shortly afterwards, the Single Shot MultiBox detector (SSD) [11] was introduced with a higher processing speed and a more accurate processing algorithm. SSD algorithm matches objects with default boxes of diverse aspect ratios. So, it uses multi-scale feature maps to detect objects that offer a better precision with diverse scales. The SSD object detection algorithm consists of two sections: extraction of feature maps and applying small convolutional filters to detect objects. The filters performed on feature maps to predict class scores and boxes for a static set of default boxes. Later, the final detection is produced by using a non-maximum suppression algorithm.

Object recognition tasks are becoming more essential to various areas of research, mainly autonomous investigation and environmental scanning, relief operations [13], landmine detection [12], and more. The proposed model aims to effectively increase the computational capacity of object detection modules to reach the desired level of compatibility for real-world applications. As a part of our long-term multi-robot system [15], the framework would enable further communication between the system agents for tracking objects of interest. The existing object recognition and classification algorithms have achieved good results. However, the existing algorithm will soon fail to cope with the rising demand for faster and more accurate detection tasks; therefore, there is always room for improvement both in theory and in practice. In view of these highlighted problems, this study aims to utilize YOLOv3 models supplied by the ImageAI python libraries alongside a deep residual network of the same number of layers known as the Darknet-53 network structure for more efficient feature extraction process.

III. Methodology

A. Deep Learning

In this paper, we aim to implement deep learning techniques for object detection and recognition to be used for detection of landmines, ocean debris, and wildlife beside environmental scanning missions. Recently, deep convolutional neural networks proved to surpass humans at object detection and

recognition. Similar to the human body, a Neural Network (NN) is mainly composed of layers of artificial neurons that contribute to designing various models. The connections of these layers relative to one another give raise to numerous deep learning architectures suitable for various problems. Some of the mainly utilized models are multi-layer perceptron (MLP)[14] architecture with two or more hidden layers , in addition to the widely used Convolutional Neural Network (CNN).

Deep learning models are generally composed of input layers, hidden layers and output layers. Hidden layers can have multi forms depending on the architecture used. Convolutional layers, dropout layers, pooling layers and Relu layers are some of the examples of hidden layers that are fundamental in creating a deep learning model for object detection tasks. Moreover, deep learning models utilize various procedures in connecting those layers efficiently to avoid overfitting for faster processing speed. The most common procedures are data augmentation, dropout and regularization. Fundamentally, Deep neural networks use a procedure known as backpropagation algorithm for effectively training a neural network through a method called chain rule. Typically deep neural networks depend on using a series of activation functions to define various essential parameters between layers such as: the weight, the output value, and the baise value of each neuron. As we move forward in the network, the final vector is evaluated depending on the type of data supplied and the set of output and activation function predefined. The network automatically assesses the evaluated result through comparing the predicted output vector against the calculated vector for a particular set of inputs using what is known as the cost function. Using the cost function's output, the network utilizes the backpropagation algorithm to adjust the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. These adjustments are calculated using the chain rule through defining the gradient of each parameter used in the activation and the input function in each neuron.

B. Convolutional Neural Networks

Convolutional Neural network (CNN) architecture in deep learning is becoming fundamental in computer vision and object detection and recognition tasks. In this paper, we propose utilizing the ImageAI library's models for computer vision and visual analysis to perform object detection. The YOLOv3 object detection and recognition model embedded inside ImageAI library depends on using a unique neural network architecture known as DarkNet-53, which is an advanced processing network that depends on using the Residual Neural Networks (ResNet) scheme. Residual Neural Networks are mainly convolutional neural networks that are primarily composed of series of convolutional layers, input layers, activation functions and pooling layers. The complexity and diversity of layers involved provide the network with an effective method of managing its input effectively to avoid overfitting issues. Real-time visual data, such as videos, are mainly processed through applying a grid to each frame - image- creating small squares that are often referred to as windows. Convolutional layers contain specific color filters that can be applied spatially to each window's primary components - pixels. After each pixel is processed the values are passed to a series of activation and summation functions to define the input for each individual layer. In case the predicted value exceeds the layer's threshold then it is passed to the proceeding pooling layer before going to the next convolutional filter. Pooling layers server as a reduction station that allows regression function to reduce the size of the window for further processing. ImageAI libraries mainly utilize YOLOv3 detection modules that rely on using non-linear regression function allowing further representation of complex visual patterns. A convolutional layer connects each output to only a few close inputs, intuitively learning local features, which is essential in object detection and recognition. Stacking various levels of convolution and pooling layers increases the complexity of the network and subsequently its ability to distinguish features in the object of interest for better results. The CNNs inner connection enables layers to pass local features more efficiently across various classes for multiple object. Various CNN architectures hold complimentary processing features depending on their scheme; however they all share the same prime structure of a convolutional model.

C. Object Detection Method

To acquire a full understanding of the procedure taken in training and building the nominate model, the main strategy of object detection using YOLOv3 architecture should be analyzed. YOLOv3 is majorly composed of a multi-scale feature extractor and an object detector. Real-time visual data from images to live videos are directed to go through the feature extractor first to obtain the needed features embeddings at three different default scales. Then, these features are fed into the detector's corresponding branches to obtain bounding boxes and class information for detection. YOLOv3 utilizes the DrakNet-53 NN that is majorly composed of a chain of multiple units containing a 1x1 and 3x3 convolutional layers followed by a skip connection. These major units are separated with some strides of 2 convolutional layers in between to reduce dimensions as the network gets deeper. Finally features are exported from the final three units to be used later in the detector.

1. Multi-Scale Feature Maps and Aspect Ratio for Default Boxes

In order to analyze various object scales, YOLOv3 methods process images in diverse sizes to be combined later in the detector. Once features are supplied to the detector, it goes through a series of 1x1 and 3x3 convolutional kernels before passing by a final 1x1 filter for final output. Such configuration concatenates features from previous layers enabling small object detection to benefit from large object features. Subsequently, inputting a picture of dimensions SxS would output three matrices, representing the number of scales detected. These matrices are later transformed to bounding boxes that bounds the area of interest according to its class. Bouncing boxes are usually normalized in a ymin, xmin, ymax, xmax format to avoid facing regression problems that emerge from seeking direct numerical values. Using network prediction based on the calculated mean square value would solve the problem; however such a method makes it hard for the network to converge considering the huge variation in scales and aspect ratios of each bounding box. To avoid such a problem YOLOv3 creates anchor boxes using aspect ratios predefined by running k-means across the whole database when training. Convolutional layers at the detector produce output as square matrices that are later

transformed into a grid of various windows or cells. Anchor boxes are defined for each window of the grid. Once the anchor boxes are defined for the whole grid, the model calculates the overlap between the anchor box area and the ground truth box predicted for coupling pairs that have the highest intersection Over Union (IOU). A number of anchor boxes are created per window corresponding to the number of scales to be detected. Three main parameters are defined for each anchor box: the location offset against the anchor box (tx, ty, tz, th), the abjectness score, in dictating if the box contains the object or not, and finally the class probability, indicating what class would this object most likely belong to. Through merging expectations for every anchor box with diverse aspect ratios and scales from every location of various feature maps, we have a various collection of shapes, expectations and covering numerous object sizes. When the expectations comprise more shapes, the model can discover more object categories. Multi-scale feature maps make training more stable and much easier. Furthermore, it enhances accuracy proficiently.

2. Loss functions:

It is a widely used technique to measure the poor performance of the model through computing the loss against the ground truth values. It is basically a weighted summation of four main parameters: centroid (x,y) loss, width and height (w,h) loss, abjectness (obj and noobj) loss and classification loss as presented by Formula (1):

$$\lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B}1_{ij}^{obj}\left[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right]$$

$$+\lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B}1_{ij}^{obj}\left[\left(\sqrt{\omega_i}-\sqrt{\hat{\omega}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right]$$

$$+\sum_{i=0}^{s^2}\sum_{j=0}^{B}1_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2$$

$$+\lambda_{noobj}\sum_{i=0}^{s^2}\sum_{j=0}^{B}1_{ij}^{noobj}\left(C_i-\hat{C}_i\right)^2$$

$$+\sum_{j=0}^{s^2}1_{ij}^{obj}\sum_{c\in classes}(p_i(c)-\hat{p}_i(c))^2 \quad (1)$$

Initially the loss of the bounding boxes is computed first through the relative centroid location from the ground truth and the centroid prediction from the detector directly. Consequently, The smaller this loss is, the closer the centroids of prediction and ground truth are. Due to the fact that some windows might not include data, the abjectness scores are used in distinguishing empty windows to be eliminated from the loss function later. YOLOv3 uses binary cross-entropy instead of mean square error to avoid regression problems. In the ground truth, abjectness is always 1 for the cell that contains an object, and 0 for the cell that does not contain any object. By measuring this parameter, we can gradually teach the network to detect a region of interest. As the network deepens, penalizing false proposals becomes a necessity. Having a threshold value to calculate the abjectness of the windows ensures that the network won't be filled with false proposals while maintaining a realistic assessment for the window's content. Next the loss in width and height are computed for the whole image and the anchor box respectively. Finally the classification loss is computed through YOLOv3 hierarchical classification categories. YOLOv3 uses multi-label classification instead of multi-class classification due to the presence of related classes. Such categorization method enables an object to have multiple classes which ensure more accuracy in defining objects of various aspect ratios

3.YOLOv3 mechanism

Either for YOLOv3's detector mains mechanism, a deeper look seems necessary in understanding the underlying principle of multi-scale object detection. The YOLO algorithm applies a single neural network to a full image through dividing the entire image into grids and inputting it, which are directly related to the position of the predicted bounding boxes, their probabilities, and their associated categories in the output layer. These bounding boxes are weighted by the predicted probabilities. First, the image is divided into SxS grids; each grid predicts B bounding boxes, and each bounding box is accompanied by a confidence value in addition to its own position, indicating the probability of the object in the prediction box, presented by the Formula (2), where IoU is the Intersection Over Union that each bounding box exhibits with the ground truth:

$$\Pr(Object)*IOU \quad (2)$$

A value of 1 is assigned in case the object falls in the grid, while zero is assigned to the opposite case. The second most important element of the algorithm is the Intersection-over-Union (IoU) of the bounding box to the truth box. Each bounding box predicts a set of four values of (x ,y ,w ,h) in addition to their corresponding confidence values, with each grid predicting a category, which is recorded C. Thus, the result is S*S*(5* B +C) dimensions for an SxS image.

When testing, the category that each grid predicts is multiplied by a confidence value, which is predicted by the bounding box corresponding to the final category score of the bounding box, represented by Formula (3):

$$\Pr(Class(i) \mid object) * \Pr(object) * IOU \quad (3)$$

Once the final category score is obtained, the threshold is set, the lower score area is filtered, and the remaining frames are processed by the non-maximum suppression (NMS) algorithm to obtain the final detection result, ensuring object detection algorithm only detects each object once

The output layer, being a fully connected layer, only allows images of the same size to be imputed during detection, knowing that each grid can only detect one object. On this basis, the YOLOv2 algorithm increases the normalization layer, suppressing the need to learn the data distribution of each layer. Such adjustments speed up the convergence for faster processing capabilities. Although being supplied with five anchor boxes for each feature map, which is more than the original algorithm, YOLOV2 algorithm faces problems with overlapping classifications.

Considering such a limitation, YOLOv3 utilizes a multi-label task with a sigmoid classifier, so that the output of each class is always between 0 and 1. The anchor is issued as a detection frame to out as long as the confidence value exceeds the predefined threshold value. Thus, the multi-label model is very effective in solving the detection effect of high-coverage images.

composition guarantees that the activation function is distributed in the linear interval through standardization, and the model conducts gradient descent more boldly. A number of residual network modules are used in the DrakNet-53 model to solve the gradient disappearance problem introduced earlier and thus deepening the network structure. This speeds up convergence, jumps out of the local minimum, and alleviates overfitting.

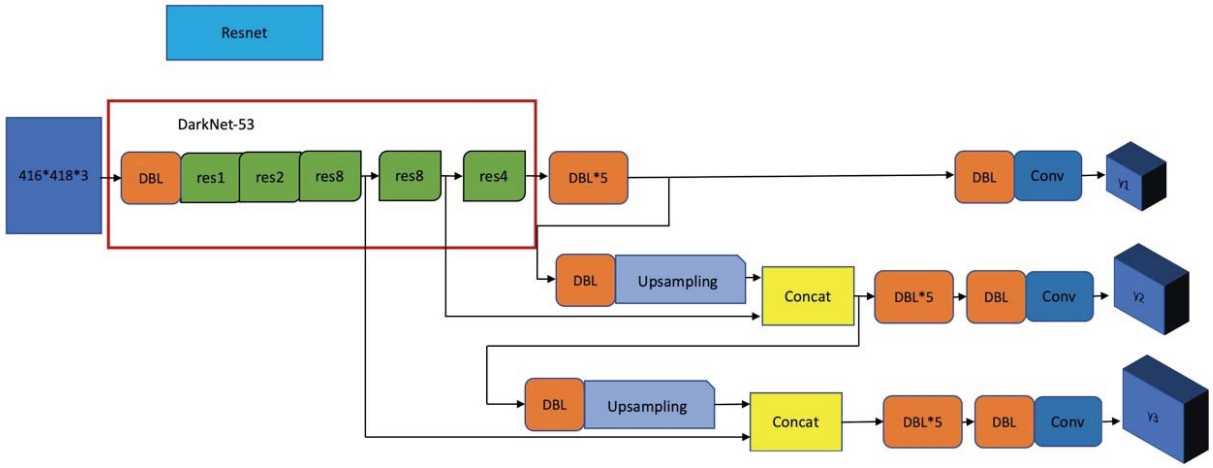


Fig. 1. YOLOv3-Resnet network structure

Images normally contain various objects of different sizes. Since YOLO aims on ideally detecting all objects at the same time, the feature map becomes progressively smaller as the network deepens with smaller sized objects. Therefore, it becomes harder to detect them. Essentially, in the feature map, the semantic information of the low-level layer is relatively less, but the object position is accurate, while the semantic information of the high-level layer is rich, but the object position is relatively inaccurate. Therefore, YOLOv3 is based on the idea of a feature pyramid network (FPN) and extracts features of different scales.

D. The DarkNet-53 structure

Although The YOLOv3 algorithm proved to be efficient in multi-scale detection, its accuracy recall rate is low. Therefore, deepening the network design is essential in improving the accuracy of the model even further. However, gradient disappearance appears as an inevitable problem when choosing to deepen the network as much as possible. Residual Neural Network (ResNet) schemes provide a compatible solution for gradient disappearance while deepening the network, thus improving the accuracy of the model. Since we are utilizing YOLOv3 models from the ImageAI library, the object detection framework is generically based on Darknet-53 network structure for feature extraction. ImageAI models combine ResNet schemes and the existing Darknet-53 network structure to solve the problem of poor accuracy in object detection.

The Darknet-53 is mainly composed of a series of 3 x 3 and 1 x 1 convolutional layers, with a total of 53 layers including fully connected layers and without counting the residual layers. Each convolutional layer is followed by a batch normalization layer and a LeakyReLU activation layer, and together they provide the smallest component of the network. This particular

The structure of the improved algorithm is presented in Fig (1) while the main structure of DarknNet-53 is presented in Fig (2). An Image with 416 x 416 pixels and 3 channels object position identification is poor, and the then the features are extracted by the Darknet-53 Neural network and the ResNet. Next, local feature interaction is realized by means of a convolution kernel, then multi-scale prediction is performed.

	Type	Filters	Size	Output
	Convolutional	32	3 x 3	256 x 256
	Convolutional	64	3 x 3 /2	128 x 128
1x	Convolutional	32	1 x 1	
	Convolutional	64	3 x 3	
	Residual			128 x 128
	Convolutional	128	3 x 3 /2	64 x 64
2x	Convolutional	64	1 x 1	
	Convolutional	128	3 x 3	
	Residual			64 x 64
	Convolutional	256	3 x 3 /2	32 x 32
8x	Convolutional	128	1 x 1	
	Convolutional	256	3 x 3	
	Residual			32 x 32
	Convolutional	512	3 x 3 /2	16 x 16
8x	Convolutional	256	1 x 1	
	Convolutional	512	3 x 3	
	Residual			16 x 16
	Convolutional	1024	3 x 3 /2	8 x 8
4x	Convolutional	512	1 x 1	
	Convolutional	1024	3 x 3	
	Residual			8 x 8
	Avgpool		Global	
	Connected		1000	
	SoftMax			

Fig. 2. DarkNet-53 Structure

E. Model Training and Testing

There are many libraries for deep learning techniques such as Caffe, TensorFlow, Torch and ImageAI. In this paper, we used ImageAI python library because it is one of the best libraries to implement deep learning techniques. ImageAI genetically inherits TensorFlow training framework as it uses Keras modules for training, which makes it easy to train using the YOLOv3 modules.

Prior to testing the nominated model, we need to train the model using a compatible database. The new framework has been trained using our own database, containing 500 annotated images of the object of interest. For experimental purposes, the object of interest would be tennis balls of various shapes and sizes. The DarkNet-53's CNN architecture was trained using bounding boxes and loss functions pre-defined inside the Keras module. Generally, their loss functions are more complex since they have to manage numerous objectives like bounding box, check if there is an object or non-object, and classification.

The Intersection over Union (IoU) and Average Precision (AP) values will be used to assist the nominated model during training time to relate the prediction to the ground truth and calculate the model Mean Average Precision (mAP). TensorFlow2 is considered to be a milestone in the training framework in machine learning, the new version comes with the ability to leverage native Python code to run the graph in a dynamic mode rather than using specific APIs to calculate graphs, which makes debugging and controlling flow easier. For the purposes of this study, we are utilizing our own database for training the model. The model shall be incorporated into a multi-robot system for environmental scanning missions. The model is expected to revolutionize the detection of hazardous objects of interest in unknown environments. Objects of interest have been replaced with tennis balls for safety purposes in our testing experiments.

IV. Results analysis

Experiments were carried out to measure essential performance parameters such as the average precision, the and the total run time of the model to asset further enhancements.

A. Experimental database

The dataset used in our experiment was generated through selecting as series of high quality images from various resources, containing the different configurations of the object of interest. Consequently, chosen images have been annotated using LabelImg annotation tool to obtain a Pascal VOC format that is compatible with ImageAI. For testing purposes, our designated object of interest would be tennis balls, simulating hazardous objects in the model's real world implantation. Our own dataset provides a complete set of excellent, standardized collections of images suitable for image recognition and classification, in addition to image segmentation. Among them, the balls' collection contains 10,000 images for 180 different types of balls used in numerous sports. We have chosen images containing tennis balls with different backgrounds and angles to eventually select 500 images as a validation custom set for our model to train on sample chosen later.

B. Experimental Layout and Parameters

Either for the experiment's layout, experimental equipment were configured as follows: the graphics card INVIDIA GTX16504GB was utilized in action to ubuntu as the operating system with imageAI framework Keras. In maximizing our mahcine's computational power, certain adjustments hold a considerable effect on the processing speed of the utilized GPU and computer memory. Parameters such as the GPU learning rate and the batch size are two of the hyperparameters that tremendously affects the performance of the model. The batch size, being the number of examples the training algorithm analyzes before making a step, decides the number and the nature of the features a model extracts from a certain dataset thus affecting the detection process. On the other hand, the GPU's learning rate is the size of the step taken, controlling the effective capacity of the model in a more complicated way. Together they adjust how the weight of the network is distributed through a gradient of loss functions. Consequently when both parameters are optimal, the effective capacity of the model is maximized. In the experiment, various learning rates were selected to be tested with batch sizes of (16, 32 , 64). Testing results indicated that a batch size of 8 forces the loss function to oscillate at either a learning rate of 0.01 or 0.001. However, with a batch size of 64, we encountered an out of memory error. Eventually a batch size of 32 was proven to be the optimal choice with slightly better results with the 0.01 learning rate than the 0.001.

It is essential to evaluate the nominate model in light of standardized parameters to verify its functionality in processing object detection tasks. Images may have different objects of different categories. Therefore, the area under the Precision Recall (PR) curve is used as an index to evaluate the quality of a model, namely, Average Precision (AP). The higher the AP value, the better the model will be, as expressed by Formula (4).

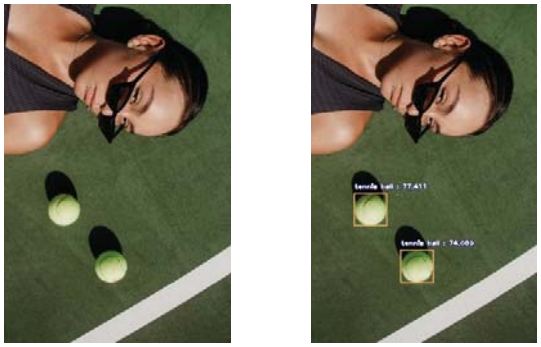
$$Average\ Precision = \frac{\sum Precision_c}{N\ (Total\ Images)_c} \tag{4}$$

Images might have different objects of different sizes. Since the nominated model aims at recognising objects of interest, objects are classified into categories depending on what they resemble. Considering that our focus is on the object being the tennis ball then calculation will be conducted across one class in defining the efficiency of the model. The Insertion Over Union (IoU) is calculated for each prediction box through measuring the intersection between the predicted box and actual box divided by their union. Such processes create True Positive and False Positive values that are later used in defining the recall and precision parameters for each prediction. ImagAI utilizes datasets of the Pascal VOC format therefore models are normally evaluated through calculating Average Precision (AP). The calculated recall and precision value of each prediction is then used in defining the confidence scores, which are used in sorting predictions. After sorting prediction, recall and precision values are calculated using a pre-defined threshold value. In this experiment the threshold value was set to 0.3 and the IoU to 0.5 for optimal results. Eventually the precision and recall values are plotted to generate a PR curve. The area under the PR curve is used as an index to evaluate the quality of a model, namely, Average Precision (AP).

In the multi-object detection task, there is more than one image label, so the evaluation cannot use the standard of the ordinary single-label image classification. In this paper, a new evaluation index, mean Average Precision (mAP), is proposed, which is the average value of all AP values. Although we are relying on single object detection, the nominated model is capable of carrying multi-object detection tasks. Due to having more than one class, evaluations should rely on more comprehensive parameters that reflect how data is related across all classes. In this paper, the mean Average Precision (mAP), is proposed, which is the average value of all AP values. Such a model will give us the flexibility of evaluating the performance of our model with any multi-object detection tasks in the future. For now, we will assess our evaluation on the calculating mAP for our tennis ball class.

C. Analyzing Experimental Results

The nominated model is yet to be implemented in a Solar-Powered UAV for mine detection as a part of a multi-robot system for detection and retrieval, environmental scanning, and relief operations. Autonomous investigation requires a deliberate level of precision and flexibility to achieve the lowest latency and the needed responsiveness. Therefore, achieving the highest level of accuracy when detecting objects of interest with the optimal processing speed is a major design requirement for the improved algorithm.



(a) (b)

Fig 3. Comparsion of image detection results between different algorithms. (a) the orginal image (b) the YOLOv3-ResNet algorithm

In proving the experiment effect of the nominated algorithm, some representative images have been chosen for testing. Fig 3 (a) represents the original image. Since there are more than one object to be detected, the identification process can be described as being demanding. Faster R-CNN algorithm fails in identifying the tennis ball out of all of the objects in the picture as illustrated by the inaccurate default box. The SSD algorithm, succeeded in distinguishing the tennis balls but with low accuracy. Finally, Fig 3(b) represents the detection results of the improved YOLOv3-ResNet Algorithm. The model succeeded in detecting the tennis ball with higher accuracy as the deception effect was further improved in the premise of guaranteeing the detection on only the tennis ball, not the other object in the picture.

The new algorithm should demonstrate the highest level of object localization to effectively detect all the objects in a frame with their accurate corresponding position in a map. This criteria is essential to ensure the effectiveness of the model in investigating hazardous objects using real time data. Fig 4 (a) represents the original picture containing 3 tennis balls. The image contains other objects and sources of distraction for the model such as uneven shadow and light distribution and color schemes. Faster R-CNN algorithm detects two of the three balls and fails to detect the other. Although the SSD algorithm manages to detect all balls, it fails to accurately report their corresponding positions. Finally, Fig 4(b) represents the YOLOv3- ResNet algorithm, which not only successfully detects all the tennis balls in the picture, but also improves the accuracy of reporting the balls accurate positions.

(a) (b)

Fig 4. Comparsion of image detection localization results between different algorithms. (a) the orginal image (b) the YOLOv3-ResNet algorithm

Table 1 represents the AP values of the tennis ball sample under different models. Comparison shows that the AP values of the YOLOv3-ResNet algorithm is the highest compared to other methods such as SSD, which comes second. Although the differences are not significant, The YOLOv3-ResNet algorithm has been improved based on the former two algorithms' prediction function.

TABLE I. THE AP VALUES(%) OF THE SAME SAMPLE (10 IMAGES) UNDER DIFFERENT MODELS

	Model Used		
	Faster R-CNN	***SSD***	***YOLOv3-ResNet***
AP (%)	60%	66.5%	68%

Table II shows the mAP value and the total running time of the four algorithms. Considering that the number of sample remains constant, the mAP value of the faster R-CNN is 71% with the lowest detection rate and longest running time. On the other hand, the mAP values of the YOLOv3-ResNet algorithm is considered to be the highest with 78%, maintaining a considerable difference from the SSD and a significant difference compared to the Faster R-CNN.

TABLE II. THE MAP VALUES(%) AND THE RUNNING TIME (S) OF THE SAME SAMPLE (10 IMAGES) UNDER DIFFERENT MODELS.

	Model Used		
	Faster R-CNN	***SSD***	***YOLOv3-ResNet***
mAP	71%	76%	78%
Time	290	275	270

D. Experimental Conclusion

The nominate model improves its feasibility and increases the AP value to 68%. Expectedly, the time spent on training increases as the network deepens. Therefore, due to access to

limited computational power, the new architecture effect on the total run time was not visible from experimental data. Excluding the time as factor, the YOLOv3-ResNet algorithm, featured in this study, succeeds in approaching the desired accuracy.

V. Conclusion

Recently, object detection methods based on deep learning has attracted a lot of attention in multiple research fields. Although we managed to improve the algorithms related to object detection tasks, there is still room for development when it comes to multiple and single object detection with larger accuracy. Therefore, in this paper, a nominated model based on the YOLOv3-ResNet detection module in the ImageAI deep learning library based on deep learning is proposed. Darknet-53 was successfully integrated with ResNet architecture to produce an algorithm capable of extracting features efficiently. the experimental results illustrate that our model achieved an accuracy of 68%, therefore compared to conventional object detection methods, the accuracy has been improved.

References:

[1] Xia, Z. (2019). An Overview of Deep Learning. *Deep Learning in Object Detection and Recognition,* 1-18. doi:10.1007/978-981-10-5152-4_1

[2] Leung, H., & Haykin, S. (1991). The complex backpropagation algorithm. *IEEE Transactions on Signal Processing, 39*(9), 2101-2104. doi:10.1109/78.134446

[3] Real-Time Object Detection for Aiding Visually Impaired using Deep Learning. (2020). *International Journal of Engineering and Advanced Technology Regular Issue, 9*(4), 1600-1605. doi:10.35940/ijeat.d8374.049420

[4] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2016.91

[5] Budiharto, W., Gunawan, A. A., Suroso, J. S., Chowanda, A., Patrik, A., & Utama, G. (2018). Fast Object Detection for Quadcopter Drone Using Deep Learning. *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*. doi:10.1109/ccoms.2018.8463284

[6] Li, X., Wang, J., Xu, F., & Song, J. (2019). Improvement of YOLOv3 Algorithm in Workpiece Detection. *2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. doi:10.1109/cyber46603.2019.9066490

[7] Zhao, L., & Wan, Y. (2019). A New Deep Learning Architecture for Person Detection. *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. doi:10.1109/iccc47050.2019.9064172

[8] Lu, Z., Lu, J., Ge, Q., & Zhan, T. (2019). Multi-object Detection Method based on YOLO and ResNet Hybrid Networks. *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*. doi:10.1109/icarm.2019.8833671

[9] Zhang, D. (2018). Vehicle target detection methods based on color fusion deformable part model. *EURASIP Journal on Wireless Communications and Networking, 2018*(1). doi:10.1186/s13638-018-1111-8

[10] Kang, M., Leng, X., Lin, Z., & Ji, K. (2017). A modified faster R-CNN based on CFAR algorithm for SAR ship detection. *2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP)*. doi:10.1109/rsip.2017.7958815

[11] Hung, P. D., & Kien, N. N. (2019). SSD-Mobilenet Implementation for Classifying Fish Species. *Advances in Intelligent Systems and Computing Intelligent Computing and Optimization,* 399-408. doi:10.1007/978-3-030-33585-4_40

[12] Castiblanco, C., Rodriguez, J., Mondragon, I., Parra, C., & Colorado, J. (2014). Air Drones for Explosive Landmines Detection. *ROBOT2013: First Iberian Robotics Conference Advances in Intelligent Systems and Computing,* 107-114. doi:10.1007/978-3-319-03653-3_9

[13] Ma, H., Liu, Y., Ren, Y., & Yu, J. (2019). Detection of Collapsed Buildings in Post-Earthquake Remote Sensing Images Based on the Improved YOLOv3. *Remote Sensing, 12*(1), 44. doi:10.3390/rs12010044

[14] Khong, L. M., Gale, T. J., Jiang, D., Olivier, J. C., & Ortiz-Catalan, M. (2013). Multi-layer perceptron training algorithms for pattern recognition of myoelectric signals. *The 6th 2013 Biomedical Engineering International Conference*. doi:10.1109/bmeicon.2013.6687665

[15] Andre, T., Neuhold, D., & Bettstetter, C. (2014). Coordinated multi-robot exploration: Out of the box packages for ROS. *2014 IEEE Globecom Workshops (GC Wkshps)*. doi:10.1109/glocomw.2014.7063639