



## RESEARCH ARTICLE - ENGINEERING

### Custom YOLO Object Detection Model for COVID-19 Diagnosis

Noor Najah Ali<sup>1\*</sup>, Aseel Hameed<sup>1</sup>, Asanka G. Perera<sup>2</sup>, Ali Al-Naji<sup>3</sup>

<sup>1</sup> Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq

<sup>2</sup> School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2610, Australia

<sup>3</sup> School of Engineering, University of South Australia, Mawson Lakes, SA 5095, Australia

\* Corresponding author E-mail: [noornajah268@gmail.com](mailto:noornajah268@gmail.com)

Article Info.	Abstract
<p><i>Article history:</i></p> <p>Received 11 January 2023</p> <p>Accepted 27 March 2023</p> <p>Publishing 30 September 2023</p>	<p>The emergence and spread of the new coronavirus (COVID-19) poses a new public health threat to the entire world (SARS-CoV-2). This new virus is highly contagious and pathogenetically different from other mainstream respiratory viruses. Clinical staff can benefit from Computer Aided Diagnostics (CAD) systems that combine deep learning algorithms and image processing technologies as diagnostic tools for COVID-19. These tools also help to better understand the course of the disease. In most cases, medical staff and healthcare facilities would be more equipped to promptly diagnose COVID-19 for patients with improved flexibility. To examine the training performance of the contemporary YOLOv4 model, this work presents the development of a computer-assisted automatic detection system that focuses specifically on identifying viral cells in blood samples from patients using electron microscopy images to detect the infected blood cell. The mean average precision of the proposed custom model is 86.5% mAP, making it suitable for the upcoming COVID-19 monitoring systems.</p>
<p>This is an open-access article under the CC BY 4.0 license (<a href="http://creativecommons.org/licenses/by/4.0/">http://creativecommons.org/licenses/by/4.0/</a>)</p>	
<p>Publisher: Middle Technical University</p>	

**Keywords:** Yolo; Convolutional Neural Network; Coronavirus; Object Detection.

## 1. Introduction

The epidemic was reported to the WHO by China on December 31, 2019, and the Huanan seafood market was shut down on January 1 of 2020. A coronavirus was discovered to be the virus on January 7. Positive test results from samples of the surrounding environment taken from the Huanan seafood market further supported this theory that the virus originated there [1]. The exponential spike in cases, some of which were not connected to the live animal industry, was a red flag that the disease was being spread from person to person [2]. The first fatality was reported on January 11, 2020. The current coronavirus epidemic was declared a Public Health Emergency of International Concern by the World Health Organization on January 30, 2020. According to the COVID-19 Open Datasets compiled by the Johns Hopkins University Center for Systems Science and Engineering, there have been 4.57 million deaths and 221.1 million cases of COVID-19 as of September 6, 2021 [3]. Researchers and others have recently created customized ways to overcome the challenges of employing automated Computer-Aided Diagnosis to diagnose several life-threatening disorders (CAD). Through this technique, medical equipment can offer quick and precise detection to support medical professionals in extending or even improving the quality of a patient's life. Automated CAD is used to identify life-threatening disorders based on artificial intelligence techniques. These techniques are cutting-edge models that have resolved challenging CAD recognition, classification, segmentation, and even detection challenges [4].

Because diagnostics allow healthcare personnel to focus on patients with COVID-19, it is an important tool in the toolkit for controlling outbreaks. Transmission electron microscopy was used to determine the shape of the virus [5] and is a useful tool that provides an understanding of SARS-CoV-2 invasion, replication, and spread in body cells and allows direct imaging of SARS-CoV-2 in tissues of COVID-19 patients and provides a good understanding of coronavirus replication in cells. Hence the importance of the proposed model in this study using these images with deep learning fields to perform automatic detection of SARS-CoV-2 cells to help medical experts in detecting and monitoring the development of this virus inside the infected body.

Recent developments in machine learning—particularly deep learning—help to recognize, categorize, and quantify patterns in medical images. These advancements rely on the use of hierarchical feature representations learned only from data rather than features that are manually created using domain-specific expertise. As a result, computers are now responsible for feature engineering, enabling machine learning novices to employ deep learning for their applications and/or research. Deep learning technically allows for the creation of networks with several (more than two) layers, which can be viewed as an enhancement over traditional artificial neural networks.

Nomenclature & Symbols			
YOLO	You Only Look Once	NMS	Non-Maximal Suppression
CNN	Convolutional Neural Network	WHO	World Health Organization
DNN	Deep Neural Network	CAD	Computer-Aided Diagnosis
MRI	Magnetic Resonance Imaging	CT	Computed Tomography
SPP	Spatial Pyramid Pooling	CSPNet	Cross-Stage Partial Network
RIO	Region of Interest	FPN	Feature Pyramid Network
mAP	Mean Average Precision		

In light of this, a different variant known as You-Only-Look-Once (YOLO), which is deep learning–based approach developed to create a one-step process involving detection and classification, distinguished by minimal memory reliance, has a reputation for being a rapid object detection model that works well with CAD systems. Strong feature extraction skills of the Convolutional Neural Networks (CNN) are crucial for deep learning [6], which are frequently employed in medical detecting fields. To detect objects, Redmon et al. suggested using the You Only Look Once (YOLO) model [7, 8]. With a single feed-forward network, the YOLO models directly anticipate the bounding boxes for creating probable bounding boxes for a picture, then apply a classifier to the resulting boxes. Post-processing was used after categorization to enhance the bounding boxes, eliminate redundant detections, and rescore the boxes in light of additional objects in the scene [9]. Due to the need to train each component separately, these complex pipelines were slow and challenging to optimize. Deep learning models are the main component of most commonplace object detection systems nowadays. The DNN automatically learns a set of visual features from the training data. YOLO is an example of how it immediately generates the category probability and coordinate position values of objects [10].

What follows are the remaining contents of this paper: the literature review is presented in section 2. The methods and materials, including the experimental setup and overall system architecture, are described in section 3. The experimental findings are in Section 4, followed by the performance evaluation in section 5. Finally, the conclusions remarks are presented in section 6.

## 2. Related Works

The subject of computer-aided diagnosis has made tremendous advancements because of deep learning techniques (CAD), including medical image analysis [11, 12], medical image detection [13-15], medical image classification [16-18] and medical image segmentation [19, 20].

Darknet-19, a convolutional neural network (CNN)-based model, is used in this research to lay the groundwork for a real-time object detection system. This system's architecture was developed to enable real-time detection. Based on the Darknet architecture's minimal layer count and filter set, this research created a DarkCovidNet model. Using X-ray tomography, it demonstrates a technique for automating the detection of COVID-19 in images of exposed chests. The proposed DarkCovidNet model aims to appropriately classify both binary diagnoses (COVID vs. no detection) and multiclass diagnoses (COVID vs. no outcome vs. pneumonia). The average accuracy for the implemented model was calculated to be 98.46% for binary classification and 98.97% for multiclass classification, with an additional 87.868% accuracy for the latter[21]. To quickly and accurately identify, this paper[22] proposes a Deep Convolutional Neural Network approach. Patient chest X-ray images were used with the InceptionV3, Inception-ResNetV2, and ResNet50 models to make predictions about the presence of COVID-19 infection. Experiments are conducted using the chest X-ray images of over 150 confirmed COVID-19 patients obtained from the Kaggle data repository. According to the findings, the proposed system successfully identifies the cases 93% of the time. In this paper[23], the deep feature plus support vector machine (SVM) based methodology is suggested for the detection of coronavirus(COVID-19) using X-ray images which makes use of deep features and a supported vector machine (SVM) rather than relying on a deep learning-based classifier, an SVM is used for classification. The CNN model's fully connected layer provides the basis for extracting the deep features that are then fed into the SVM for classification. Solar corona X-ray images are sorted by the support vector machine. There are three types of X-rays used in this method: those for COVID-19, for pneumonia, and for a healthy individual. 13 different CNN models' depth features are used to assess SVM's efficacy at detecting COVID-19. To detect COVID-19, the classification model (ResNet50 plus SVM) attained percentages of 95.33 % accuracy, 95.33 % sensitivity, 2.33 % FPR score, and 95.34 % F1 score, respectively. The X-ray pictures hosted on GitHub and Kaggle were used to determine the final score. Images were normalized to extract better features [24], and those features were fed into deep learning-based image classification algorithms. VGG19, MobileNetV2, Inception, Xception, and InceptionResNetV2 are the five state-of-the-art CNN systems that were put through their paces in a transfer-learning scenario to detect COVID-19 in images of both controls and pneumonia. Two sets of experiments were carried out, the first using 504 control images, 700 images of bacterial pneumonia, and 224 images of COVID-19, and the second using the same normal and COVID-19 data but 714 images of bacterial and viral pneumonia. The MobileNetV2 network performed best in both the two- and three-class classifications, with an accuracy of 96.78% and 94.72%, respectively.

The purpose of this research was to use deep learning methods to develop a model for early screening that could use pulmonary CT images to differentiate between COVID-19, IAVP, and normal conditions. There were a total of 618 CT samples taken, 219 from 110 COVID-19 patients, 224 from 224 IAVP patients, and 175 from 175 healthy individuals. To begin, a 3D deep learning model was used to segment potential injury regions from a pulmonary CT scan. Afterward, an attention-to-site classification model was used to categorize these individual images into the COVID-19, IAVP, and ITI categories, along with confidence scores. Finally, the Noisy-OR Bayesian function was used to determine the infection type and overall confidence score for each CT case. A total accuracy rate of 86.7% was found in all CT cases when using the standard data set in the experiments [25]. Another work [26] used computed tomography and chest X-ray images based on a new CNN model to identify COVID-19. Two classes (COVID and Normal), three categories (COVID, Normal, and Pneumonia), and four classifications (COVID, Normal, Non-COVID Viral Pneumonia, and Non-COVID Bacterial Pneumonia) might all be accurately diagnosed using the suggested model, CoroDet. The proposed model's class classification precision for two classes, three issues, and four categories was 99.1%, 94.2%, and 91.2%, respectively. A 22-layer network that includes convolutional, max pooling, dense, and flattened layers, as well as three activation functions, including sigmoid, RELU, and leaky RELU, forms the foundation of our proposed CNN model. In [27], a new method for the automatic detection of COVID-19 from raw chest X-ray images was proposed. The proposed approach was established to provide accurate diagnoses of multiclass (COVID VS no outcome) and binary (COVID VS outcome) (COVID no outcome VS pneumonia) classification. Concerning binary categories,

this model produced classification accuracy results of 98.08% and 87.02%, respectively. Also, a YOLO real-time object detection system classifier was created using the DarkNet model with 17 convolutional layers and different filtering applied to each layer.

Several medical imaging modalities, such as MRI, CT, or microscopy, allow quantitative analysis of Covid-19 control. It is very necessary to know its structure and behavior, and this requires imaging and seeing it more clearly and accurately, and some evidence indicates that SARS-CoV-2 infection may not be limited to the respiratory system but can spread.

For other organs, including the heart, liver, kidneys, intestines, skin, and brain. For this purpose, the light used for imaging must be suitable for the dimensions of the virus. This is why the use of transmission electron microscopy rather than light microscopes, where one can go deeper into the details of cellular and molecular structures — about five thousand times deeper, to be exact. Since the average size of this virus is about 100 nanometers (0.1  $\mu\text{m}$ ), it is smaller than what an optical microscope can see (if the magnification exceeds 2000x, the image of the sample becomes blurry or blurry). So it was examined using transmission electron microscopy. This transmission microscope magnifies objects up to 2.5 million times. After obtaining images of the virus under an electron microscope, the current study proposed a customized model based on YOLOv4 and electron microscopy images to detect COVID-19 in the cells and tissues of the infected person. YOLOv4 is released by Alexey Bochkovskiy, Chien Yao Wang, and Hong Yan Mark Liao. As a convolutional neural network, YOLOv4 outperforms the latest detection methods in terms of accuracy, speed, and efficiency. To our knowledge, this is the first report of the detection of SARS-CoV-2 using TEM images.

### 3. Methods and Material

#### 3.1. Experimental setup

Downloading and installing the Anaconda package is the first step in configuring our computer to code in Python. Open CV, the open-source computer vision Python library, was installed. In phase 1st of the proposed model, the data needed to train the model was collected. All coronavirus images (microscopic output only) were collected from the internet and arranged in folders. Internet not having that many of the original coronavirus images, so the proposed system will be able to learn from different magnifications and also if it is a little darker gray or more brightly shaded. It can be used as separate images called 'augmentation'. The same image represented at a different magnification contrast and different color saturation help our system learn how to handle this kind of change. It will then be 'marked' or 'annotated' on the coronavirus object using an open-source annotation tool called 'labellmg', according to experimental requirements. For the model (algorithm) to make decisions with high accuracy, it must be provided with sufficient training data so that there are no problems with the reliability of this model divide the collected dataset 80 % for training and 20 % for testing then update and prepare the files with the chosen training and test datasets. Now that we have all the collected files ready. In the second phase, they were zipped and uploaded to Google Drive. Additionally, Google Colab Notebook and Colab runtime tuning are included to use Google's fast, powerful, and free GPU service. Next, connect Google Drive to Google Colab Runtime and unzip the downloaded darknet zip folder. Our Colab runtime is a humble computer, and we have to convert all editable files from "Windows Dos" to "Unix". Next, the darknet framework source code is compiled, and the framework is tested using a sample image. Google Collab's free GPU-based runtime is "fluctuating," meaning it is not permanent. It will reset every 12 hours or when the internet connection is lost. So it is a good idea to head to Google Drive regularly while working out to keep our weight in check. Finally, we are ready to start training with the proposed coronavirus model using YOLOv4.

#### 3.2. Dataset

Unfortunately, databases (electron microscopy images) of COVID-19 from most countries were few. The severe acute respiratory syndrome-coronavirus 19 strain is a subgroup of the Coronavirus family. The viral particle has an oval or spherical shape that resembles a solar corona. Different coronaviruses have quite distinct spinous processes. The diameter of coronavirus particles ranges from 60 to 140 nm; therefore, they are not all the same size [28]. An example of the microscopic image of SARS-CoV-19 is shown in Fig. 1.

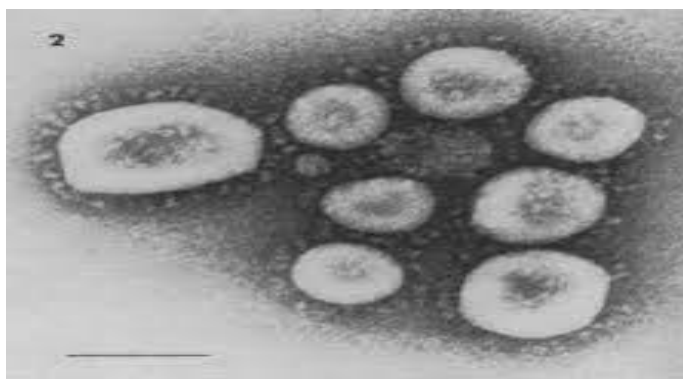


Fig. 1. SARS-CoV-19 image from an electron microscope

The data of this research were collected as electronic images of SARS-CoV-2 from the Internet. The powerful CNN extraction allowed us to obtain deep-learning properties that could be used for detection. The data set's structure includes a training set, a validation set, and a test set. Training data was utilized to create the gradient, train the detection model, and modify the weights. The validation data was used to adjust individual hyperparameters and avoid overfitting (epoch size, learning rate). Test data was used to e The input port of the object detection network accepts only data of a fixed, standardized size for each image. However, ResNet forces the input image to be 224 x 224, whereas the sizes of the images in various data sets vary. YOLO-v4 Detection Network Input Port was pre-configured with two standard sizes (416 x 416) and 608 x 608. YOLO-v4 then directly image sizes have been standardized via the source of information before feeding it into the network to complete the full end-to-end learning process. The input images used in YOLO-training v4's and testing phases had dimensions of 416 by 416

pixels and 608 by 608. The original images were scaled, and then the gray images with dimensions of 416 416, or 608 608 were created to standardize images of varying sizes. To obtain uniformly sized image data, the scaled image was finally superimposed on the grayscale original. Valuate the model's performance. In this study, 20% of the data was used as a test set, and 80% of the data was used as a training set.

3.3. YOLO

YOLOV4 is a real-time, high-precision stage object detection method based on regression that was first suggested in 2020. It combined the strengths of earlier versions, including YOLOV1, YOLOV2, and YOLOV3 by balancing speed and accuracy in detection to achieve the best possible result. The backbone, neck, and prediction are the three components that make up the structural components of the model, which is depicted in Fig. 2.

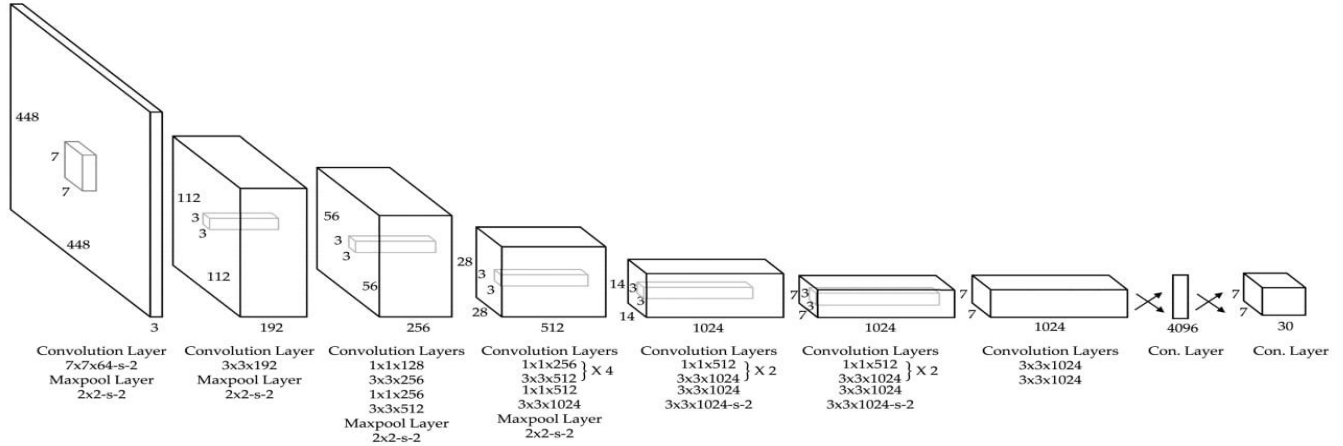


Fig. 2. YOLOV4 architecture [29]

It is important to note that YOLOv4 still uses YOLOv3 as its head detection system. However, YOLOv4 included a more advanced backbone known as CSPDarknet53 that used the Cross-Stage Partial Network (CSPNet). By splitting the input into two halves and sending it separately to a dense block, the CSPNet managed the model's input. Consequently, it decreased the previously high computational needs to train bespoke models, making YOLOv4 competently operate on low-performance devices [30]. At present, the neck consists of the Spatial Pyramid Pooling (SPP) and the Path Aggregation Network (PAN). SPP increased the flexibility of a CNN model like YOLOv4 by disregarding predefined image dimensions. This lets the model train and recognize a broader range of scaled images. The detector is trained with a single feature calculation across the entire image, this advanced SPP approach randomly pools regions that yield fixed-length representations. The Spatial Pyramid Pooling (SPP) and Path Aggregation Network currently make up the neck section (PAN). SPP increased the flexibility of a CNN model like YOLOv4 by disregarding predefined image dimensions. In addition, PAN replaced the earlier Feature Pyramid Network (FPN) of YOLOv3 and enhanced the neural network propagation of YOLOv4, which increased the flow of information throughout the entire network. improvement in the transmission of network features from lower to higher layers as a result of this strategy increased model performance efficiency [31].

What made YOLO unique was that it could take in a whole image at once, divide it into a S x S grid as in Figure 3, and use regression to make a map of the likelihood for a Region Of Interest (ROI).

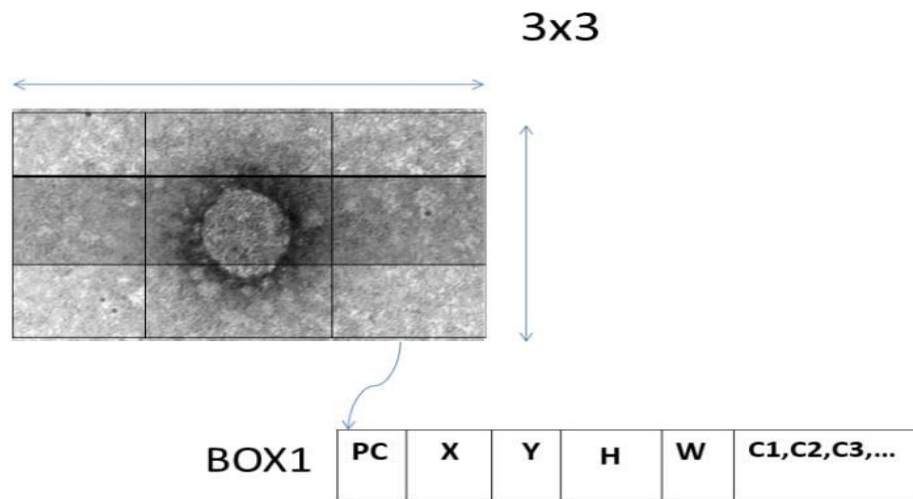


Fig. 3. The grid cells were created from the image, and each grid cell has predictions in it

This area of interest tells CNN where the object was probably located in each frame. In the fourth version of YOLO, called YOLOv4, these attributes have been improved, making it possible to identify objects with high precision in real time. The parameters (x, y) in Fig. 4 represent the coordinates of the center of the bounding box that was predicted, whereas (h, w) represent the expected dimensions of the bounding box, and (pc) reflects the probability that a given object was contained inside the grid. Each bounding box on the grid, regardless of its size, was anticipated to have its own set of parameters specific to that box. For a given grid cell, the predicted values are (B + n), where B is the number

of bounding boxes in the cell. The more layers a convolutional neural network has, the deeper the network is, and the better it is at extracting features from images. To build the YOLO (v4) architecture, it is decided to use a network with 29 convolutional layers, 3 filters, and around 27.6 million parameters. Three assumptions are made for each bounding box:

- The coordinates of the object's centroid concerning the grid cell.
- About the entire image, the bounding box's width and height.
- Confidence score (ranging from 0% to 100%) that the object matches what the algorithm anticipated.

In this work, YOLO V4 is adopted, which is the most advanced detector and is faster and more accurate than any other option. The feasibility of the proposed detector is determined based on a single-stage anchor. YOLO's low computational cost is a result of the fact that it does not rely on extracting features on sliding windows. To directly detect each bounding box and its class label probability, it only employs features from the entire image. The many grid designs in YOLO V5 are more flexible, have a small sample size, and are as accurate as the YOLO V4 standard. Since YOLO V5 is not much different from YOLO V4, many people are still wary of it. YOLO V4 has undergone significant revisions, with an increased focus on data comparison and other improvements. YOLO's main network improvement measures from V1 to V5:

- YOLOV1: Loss of trust and detection is caused by grid division.
- YOLO V2: Two-stage training, a fully convolutional network, and an anchor that includes K-means.
- YOLO V3: which employs multi-scale detection that is FPN-based.
- YOLO V4: The GIOU (Generalized Intersection over Union) loss function, SPP, MISH activation function, mosaic/ mix-up data augmentation.
- With YOLO V5, the Hardswish activation algorithm, flexible model size management, and enhanced information are used.

### 3.4. Image labeling and annotation

Initially, a square box of the image sample was labeled using the object detection and labeling tool. As shown, the application created a label file that corresponded to the image naming., as shown in Figure 4. A text file was produced as the callout. which mostly remembered the type of label and its position relative to the box's upper left and right corners., as shown in Table 1.

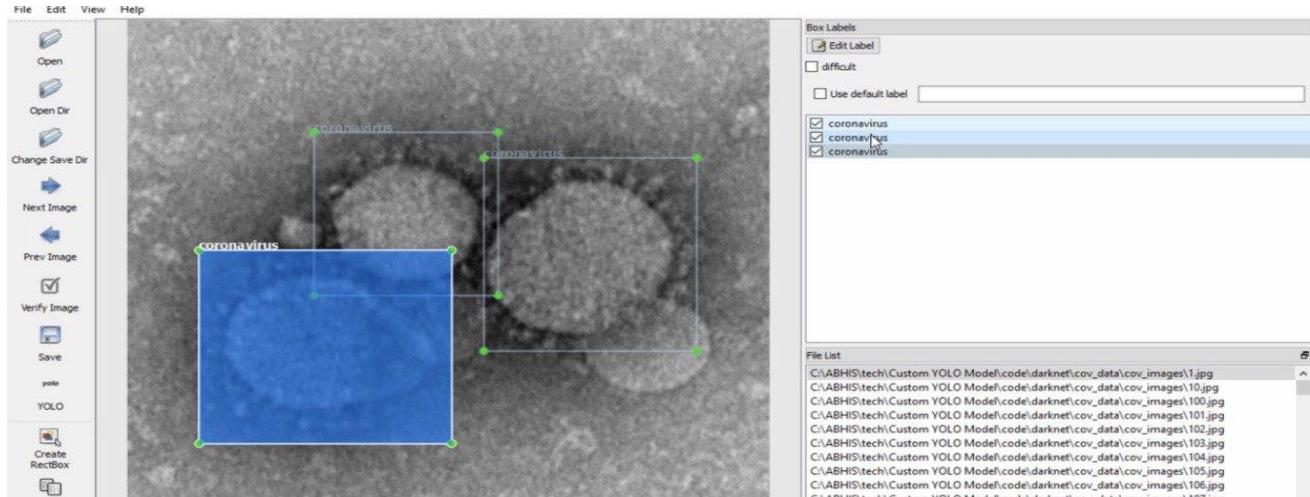


Fig. 4. The schematic diagram of the rectangular box label

Table 1. Annotated text file diagram in the form of a diagram

Category	X	Y	W	H
1	0.720670	0.424581	0.3314771	0.350093
2	0.433892	0.352886	0.286778	0.296089
3	0.308194	0.592179	0.392924	0.350093

In the YOLO format, annotating each line of txt indicated the class and index of an item. The first column represented the object's category. The following four columns represented the object's position information, which was x, y, w, and h. Each image corresponded to a txt file, and each file contained multiple categories of objects. The target's central coordinates were X and Y, and its width and height were w and h, respectively. Standardization of the coordinates has been performed. For this normalization, the width of the original diagram was used for X and W, and the height was used for Y and H.

To properly train machine learning models, features must be extracted from radiative modalities because the quality of the extracted features directly affects the model's effectiveness. This study describes a deep learning strategy based on abstraction and classifier machine learning for COVID-19 pneumonia. Several machine learning algorithms were trained on the features extracted by known CNN structures to find the best combination of learner and features. Deep feature extraction is frequently regarded as a critical step in developing deep CNN models due to the enormous visual complexity of image data. If this cell contains no objects, the confidence score should be 0. If not, the confidence score must be equal to the intersection of the union (IOU) of the projected box and the actual data. During the creation of bounding boxes in Eq. 1, By comparing the ground truth image from the dataset to the prediction, the Intersection Over Union (IOU) measures how well the two images match up.

$$IOU \text{ scores} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \tag{1}$$



The detected object is further categorized as a coronavirus cell according to its category, its class probability, and its confidence score for that specific class label, as explained below in Eq. 2.

$$C \text{ class probability} = Prob (Classi | object) \times IoU \text{ score} \tag{2}$$

However, Eq. (3) determines that a decrease in confidence score is warranted when the predicted object bears only a passing resemblance to the true object, resulting in an unsatisfied prediction. The Intersection Over Union (IOU) score is multiplied by the probability of an existing class object to derive the confidence score, as shown in Eq. 3.

$$Confidence = Prob (object) \times IoU \text{ score} \tag{3}$$

#### 4. Results and Discussions

YOLOv4 could effectively evaluate the input image by dividing it into a grid of B bounding boxes, and their confidence scores were predicted for each grid cell, as shown in Fig. 5.

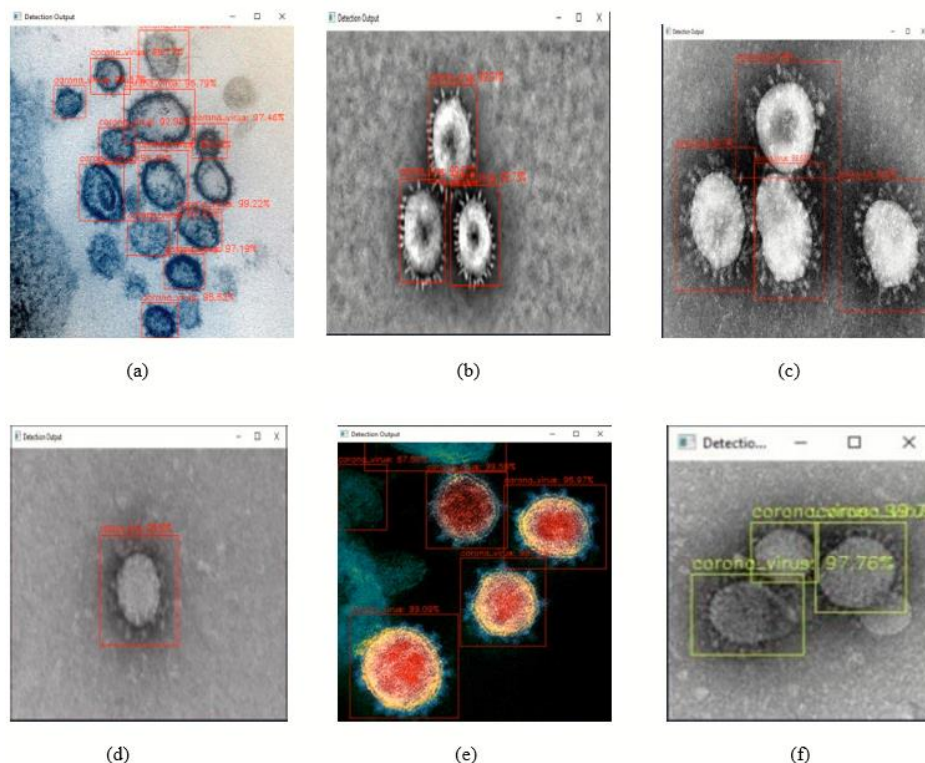


Fig. 5. Examples of microscopy datasets of coronavirus cells and their confidence score detected by our custom model, where the yellow and red squares indicate the presence of the viral cell in the blood of a sick person

These confidence scores represent the model's assessment of the accuracy of the box as well as the degree of certainty that an object is in the box. Squares with a low probability of items and bounding squares with a large common area are removed. Non-maximal suppression (NMS) is the term for this condition. Figure 6 shows that the experimental discovery results on the microscopic data set available for viral cells in infected blood achieved different confidence scores ( 99.9%, 98.82%, 97.46% ). This degree shifts depending on the model's configuration, the input data, and the classification procedure. The internal procedure was conducted by YOLO to determine the probability score for class naming (Coronavirus). The proposed custom model achieved an average mean accuracy of 86.5% mAP in terms of classification accuracy from the small CNN-based data. Thus, the grid cell detects an object if its center is located within it with a learning rate of 0.001 and a total detection time of 1 second.

#### 5. Performance Evaluation

There are many ways to evaluate the performance of the YOLO model, but to evaluate the proposed system, this study uses the most popular measurement technique known as mAP. This technique measures the average accuracy (accuracy) of object classification, the plotted square of the detected object, and the level of confidence of the model in making the prediction. The image to be trained and the image to be tested for performance evaluation must be built independently for this evaluation. The mAP value of the YOLOv4 model used in this study is shown in Fig. 6, where the vertical axis of the graph represents the loss value and the horizontal axis represents the number of times the YOLOv4 model is trained. The loss value is represented by the blue line, and it decreases with the increasing number of training sessions (the lower the value, the better). The mAP value is shown by the red line, which rises with increasing training intensity until it approaches a constant value and becomes flat (the higher the value, the better).

In the period that this graph was drawn, the average accuracy was 85.1% mAP. for every 200 epochs. The average accuracy was recorded in the graph, and the loss was minimized. The loss continued to decrease to around zero, but that was not the point of extreme precision. The maximum accuracy point was where we had the best average accuracy, which was 89.40% mAP. Whenever it reaches a maximum mean average point, it records that, and whenever it drops, it immediately saves the model at this 89.40% mAP point. After reaching the average accuracy, it is now decided which weights to use, either using the "best" weights which gave an average accuracy of 89.40% at 1000 epochs or using the latter weights of 86.52% mA at 2000 epochs. According to experts, the best mean average precision model will give us the best result, so the weights and configuration for YOLOV4 at the point (89.40% mAP) were downloaded time to use it. The confusion matrix is used as a second evaluative value to establish the accuracy of the classifications made in this study. Following is an explanation of the various metrics (Precision, Accuracy, Sensitivity (Recall), Specificity, and F1-score) that the matrix provides for assessing the classification model. To see how distinct the classifier is between the two classes, the confusion matrix can be looked at in Fig. 7. Common statistical measures used to assess bounding box condition outcomes include Intersection Over Union (IOU), recall, accuracy, and F1 score. When analyzing classifier effectiveness, the confusion matrix is the general analytics tool for describing classifier performance. The IOU process for model discovery involves estimating the distance of the model's observed and predicted bounding boxes and using it to determine whether the bounding box is reliable.

The intercept ratio (IOU) is defined as the ratio of the intercept and the union of the actual box and the projected box. If the value of the IOU is higher than 0.5, the categorized results for object detection are defined as a True Positive (TP) if the IOU value is less than 0.5, it is classified as a false positive (FP). For object detection, a false negative (FN) indicates that the predicted results should be positive, but the models instead detected nothing. Accuracy, recall, specificity, Precision, and the F1 score are all defined above Precision is a measure of the ability to recognize from negative data sets. Higher precision scores indicate that models are better able to separate negative data. When dealing with positive datasets, recall is the indicator of recognition success. When the recall score is high, the model's ability to capture positive data is also high. The F1 score is a metric used to compare different models, and it is calculated by averaging the model's precision and recall. As the F1 score rises, it indicates that the model is more robust. TP=93, TN=49, FP=7, FN=1.



Fig. 6. Loss and accuracy graph of the YOLOv4

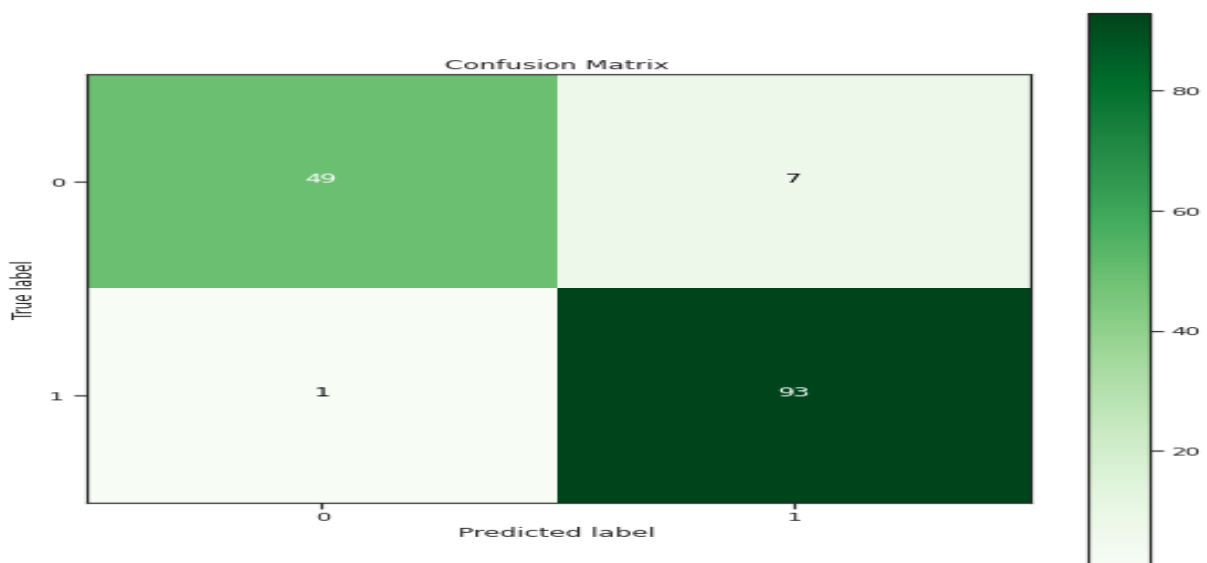


Fig. 7. Confusion Matrix of the model

- Precision = True Positive (TP) / True Positive (TP) + False Positive (FP).
- Recall = True Positive (TP) / True Positive (FP) + False Negative (FN).

- $\text{Accuracy} = \frac{\text{True Positive (TP)} + \text{True Negative (TN)}}{\text{True Positive (TP)} + \text{False Positive (FP)} + \text{False Negative (FN)} + \text{True Negative (TN)}}$ .
- $\text{Specificity} = \frac{\text{True Negative (TN)}}{\text{True Negative (TN)} + \text{False Positive (FP)}}$ .
- The F1 score will be medium if one of the values (either Precision or Recall) is low and the other has a high value.

For the quantitative evaluation of the YoloV4 model, this paper used 150 images as a test data set to examine the experimental results. Statistical indices of accuracy, recall, specificity, precision, and F1 score were implemented to assess the robustness of the model for detecting virus cells. As mentioned above, the F1 score is a comprehensive indicator for evaluating the durability of a model. The result reveals that the YoloV4 model detected cells with an F1 score of 95%, accuracy 94%, recall 98%, specificity 87% and precision 93%.

## 6. Conclusion

This paper's main originality lies in applying the State-of-the-art YOLO network to detect COVID-19 from microscopic coronavirus images. The results were encouraging even with a limited data set. Principal Inference why YoloV4 performed better for the automatic detection of COVID-19 the larger grid size and DarkNet feature extraction were better suited to the images in this dataset. The real-time YOLO object detection system was formulated to produce a straightforward discovery and sorting procedure. Image classification was the process of sorting an image, or an object in an image, into one of several categories. Usually, after training the model on a data set with a large classification, supervised machine learning or deep learning techniques were used to overcome this problem. This study proposed a detection model for CAD that can be run automatically and concentrates on detecting viral cells in the blood samples of infected patients through electron microscopy images. For COVID-19 cases from CNN-based small data, the proposed model enhanced the detection performance by using the average IOU as an input and using it to estimate the number of fit squares. The proposed model was trained with YOLOv4 and then used a publicly available dataset to build a deep learning technique, YoloV4-arch neural network, which detected COVID-19 with 86.5% mAP with the threshold set to 0.5. In the future, if more electron microscopy images are obtained to expand the data set for COVID-19 so that they are used to extract more visual features for image classification, and if laboratories equipped with transmission electron microscopy are available, they are used to examine virus samples and a uniform standard is used in the magnification process for all the samples under examination, and thus through which the size of each cell is measured, can be used to distinguish between Covid-19 and seasonal influenza, in addition to counting the number of these cells in specific areas, and thus they are used to monitor the development and spread of the virus within the infected body.

## Acknowledgments

In particular, the authors wish to express their gratitude to the Department of Computer Engineering Techniques, Electrical Engineering Technical College, Middle Technical University-Baghdad, Iraq for their support in conducting this work.

## References

- [1] Guo, H., et al., The impact of the COVID-19 epidemic on the utilization of emergency dental services. *Journal of dental sciences*, 2020. 15(4): p. 564-567.
- [2] Lundervold, A.S. and A. Lundervold, An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für Medizinische Physik*, 2019. 29(2): p. 102-127.
- [3] Chen, Y., et al., New-onset autoimmune phenomena post-COVID-19 vaccination. *Immunology*, 2022. 165(4): p. 386-401.
- [4] Redmon, J. and A. Farhadi. YOLO9000: better, faster, stronger. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
- [5] Martin-Cardona, A., et al., SARS-CoV-2 identified by transmission electron microscopy in lymphoproliferative and ischaemic intestinal lesions of COVID-19 patients with acute abdominal pain: two case reports. *BMC gastroenterology*, 2021. 21(1): p. 1-10.
- [6] Redmon, J., et al. You only look once: Unified, real-time object detection. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- [7] Redmon, J. and A. Farhadi, Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
- [8] Girshick, R., et al. Rich feature hierarchies for accurate object detection and semantic segmentation. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.
- [9] LeCun, Y., Y. Bengio, and G. Hinton, Deep learning. *nature*, 2015. 521(7553): p. 436-444.
- [10] Liu, W., et al. Ssd: Single shot multibox detector. in *European conference on computer vision*. 2016. Springer.
- [11] Anwar, S.M., et al., Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, 2018. 42(11): p. 1-13.
- [12] Ker, J., et al., Deep learning applications in medical image analysis. *Ieee Access*, 2017. 6: p. 9375-9389.
- [13] Singh, S.P., et al., 3D deep learning on medical images: a review. *Sensors*, 2020. 20(18): p. 5097.
- [14] Ker, J., et al., Image thresholding improves 3-dimensional convolutional neural network diagnosis of different acute brain hemorrhages on computed tomography scans. *Sensors*, 2019. 19(9): p. 2167.
- [15] Singh, S.P., et al., Shallow 3D CNN for detecting acute brain hemorrhage from medical imaging sensors. *IEEE Sensors Journal*, 2020. 21(13): p. 14290-14299.
- [16] Yadav, S.S. and S.M. Jadhav, Deep convolutional neural network based medical image classification for disease diagnosis. *Journal of Big Data*, 2019. 6(1): p. 1-18.
- [17] Milica, M. and M.C. Badža, BarjaktaroviC. Classification of Brain Tumors from MRI Images Using a Convolutional Neural Network. *MDPI*. 2020, MultiDisciplinary Digital Publishing institute.
- [18] Ker, J., et al., Automated brain histology classification using machine learning. *Journal of Clinical Neuroscience*, 2019. 66: p. 239-245.
- [19] Zhou, T., S. Ruan, and S. Canu, A review: Deep learning for medical image segmentation using multi-modality fusion. *Array*, 2019. 3: p. 100004.
- [20] Ramesh, K., et al., A review of medical image segmentation algorithms. *EAI Endorsed Transactions on Pervasive Health and Technology*,



2021. 7(27): p. e6-e6.
- [21] Nair, R., et al., Detection of COVID-19 cases through X-ray images using hybrid deep neural network. World Journal of Engineering, 2022. 19(1): p. 33-39.
- [22] Medhi, K., M. Jamil, and M.I. Hussain, Automatic detection of COVID-19 infection from chest X-ray using deep learning. medrxiv, 2020: p. 2020.05. 10.20097063.
- [23] Sethy, P.K. and S.K. Behera, Detection of coronavirus disease (covid-19) based on deep features. 2020.
- [24] Apostolopoulos, I.D. and T.A. Mpesiana, Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. Physical and engineering sciences in medicine, 2020. 43: p. 635-640.
- [25] Xu, X., et al., A deep learning system to screen novel coronavirus disease 2019 pneumonia. Engineering, 2020. 6(10): p. 1122-1129.
- [26] Hussain, E., et al., CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images. Chaos, Solitons & Fractals, 2021. 142: p. 110495.
- [27] Ozturk, T., et al., Automated detection of COVID-19 cases using deep neural networks with X-ray images. Computers in biology and medicine, 2020. 121: p. 103792.
- [28] Bullock, H.A., C.S. Goldsmith, and S.E. Miller, Best practices for correctly identifying coronavirus by transmission electron microscopy. Kidney International, 2021. 99(4): p. 824-827.
- [29] Koylu, C., C. Zhao, and W. Shao, Deep neural networks and kernel density estimation for detecting human activity patterns from geo-tagged images: A case study of birdwatching on flickr. ISPRS international journal of geo-information, 2019. 8(1): p. 45.
- [30] Wang, C.-Y., et al. CSPNet: A new backbone that can enhance learning capability of CNN. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020.
- [31] Liu, S., et al. Path aggregation network for instance segmentation. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.