Multimodal Technologies and Interaction

MDPI

Article

Enhancing Object Detection for VIPs Using YOLOv4_Resnet101 and Text-to-Speech Conversion Model

Tahani Jaser Alahmadi [1,*], Atta Ur Rahman [2], Hend Khalid Alkahtani [1] and Hisham Kholidy [3]

[1] Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University (PNU), P.O. Box 84428, Riyadh 11671, Saudi Arabia; hkalqahtani@pnu.edu.sa
[2] Faculty of Computer Science and Engineering, GIK Institute of Engineering Sciences and Technology, Swabi 23640, Pakistan; atta.rahman@giki.edu.pk
[3] Department of Networks and Computer Security, SUNY Polytechnic Institute, College of Engineering, Utica, NY 13502, USA; kholidh@sunypoly.edu
* Correspondence: tjalahmadi@pnu.edu.sa

Abstract: Vision impairment affects an individual's quality of life, posing challenges for visually impaired people (VIPs) in various aspects such as object recognition and daily tasks. Previous research has focused on developing visual navigation systems to assist VIPs, but there is a need for further improvements in accuracy, speed, and inclusion of a wider range of object categories that may obstruct VIPs' daily lives. This study presents a modified version of YOLOv4_Resnet101 as backbone networks trained on multiple object classes to assist VIPs in navigating their surroundings. In comparison to the Darknet, with a backbone utilized in YOLOv4, the ResNet-101 backbone in YOLOv4_Resnet101 offers a deeper and more powerful feature extraction network. The ResNet-101's greater capacity enables better representation of complex visual patterns, which increases the accuracy of object detection. The proposed model is validated using the Microsoft Common Objects in Context (MS COCO) dataset. Image pre-processing techniques are employed to enhance the training process, and manual annotation ensures accurate labeling of all images. The module incorporates text-to-speech conversion, providing VIPs with auditory information to assist in obstacle recognition. The model achieves an accuracy of 96.34% on the test images obtained from the dataset after 4000 iterations of training, with a loss error rate of 0.073%.

Keywords: object detection; recognition; tracking; text-to-speech conversion; YOLOv4_Resnet101; visual impairments; disabilities

Citation: Alahmadi, T.J.; Rahman, A.U.; Alkahtani, H.K.; Kholidy, H. Enhancing Object Detection for VIPs Using YOLOv4_Resnet101 and Text-to-Speech Conversion Model. *Multimodal Technol. Interact.* **2023**, *7*, 77. https://doi.org/10.3390/mti7080077

Academic Editors: Christos Troussas, Cleo Sgouropoulou, Akrivi Krouska, Ioannis Voyiatzis and Athanasios Voulodimos

Received: 7 July 2023
Revised: 17 July 2023
Accepted: 26 July 2023
Published: 2 August 2023

1. Introduction

People with visual impairment disabilities, referring to person-centered disorders, have some degree of visual impairment or complete loss of vision. This can have significant impacts on their daily lives, affecting their mobility, ability to perform tasks, and overall quality of life. According to the World Health Organization (WHO), an estimated 246 million people globally have a visual impairment, of which 36 million are blind and 210 million have low vision [1]. The prevalence of person-centered disorder is higher in low-income countries, and the majority of people with such disabilities are over 50 years of age. There are many reasons for visual disorders. Certain genetic disorders, such as retinitis pigmentosa and muscular degeneration, cause loss of vision. Injuries to the eye, head, and other kinds of trauma can lead to an eye disease which substantially leads to person-centered disorder. Some other diseases that can damage the eyes and lead to visual disorders include diabetic retinopathy, cataracts, glaucoma, and the degeneration of the age-related macular. Pink eye and conjunctivitis infections can result in permanent vision loss. Certain types of multiple sclerosis and stroke can result in vision loss. Exposure to pollutants and toxins, long-term viewing of bright screens and glare, and long-term use of contact lenses can all result in total vision loss. As we experience the phenomenon of aging,

the eyes naturally become impaired, which can lead to a loss of vision in certain people. Visually impaired individuals face a range of difficulties on a daily basis, some of which include mobility, information accessibility, education, and social activities [2]. Without the ability to see, it can be difficult to navigate new or unfamiliar environments, leading to difficulties in traveling, crossing the street, or simply getting around [3]. VIPs cannot access information such as maps, graphs, and images. Many technologies and devices are not designed to be accessible to visually impaired individuals, making it difficult for them to use computers, smartphones, or other devices. They may face challenges in obtaining an education, especially if their school or learning environment is not equipped to accommodate their needs. Without the ability to see, VIPs may find it difficult to engage in social activities and build relationships with others and may face an increased risk of health problems, such as depression, due to the difficulties they face in their daily lives. These difficulties highlight the importance of developing smart devices that may be used by VIPs to move without manual assistance and perform their daily activities easily. Researchers adopted various methods and techniques to ease the lives of VIPs [4].

Automatic object detection and recognition is a hot research area for researchers and continuously improving results are being seen. After the invention of deep convolutional neural networks (DCNN), which brought huge success in the field of digital image processing [5], it is now used in almost every field of life, including face recognition, smart transportation, surveillance cameras, medical imaging, etc. DCNN-based systems are 30–40% more accurate than traditional image processing-based systems. There are fewer efforts needed to perform feature engineering and train the model as compared to traditional and machine learning (ML)-based models. Smart navigators used to help VIPs are also an example of automatic object detection and recognition. Today, almost every previous research performed for smart navigation systems for VIPs based on image processing utilizes DCNN-based models [6].

In the proposed system, a modified version of YOLOv4_Resnet101 is trained to detect and recognize the obstacles in the way of VIPs indoors and outdoors. Yolov4 is a single-shot detector famous for both speed and accuracy. Most of the previous research shows that it performs well compared to other state-of-the-art models, including SSD and Faster-RCNN [7]. We made significant modifications to the original model by incorporating channel pruning and fine-tuning techniques. Channel pruning, a key component of our approach, allowed us to streamline the model by identifying and removing redundant or unnecessary channels in the convolutional layers. This not only reduced computational complexity but also optimized memory requirements [8]. To further enhance the training process, we used additional augmentation techniques. These techniques involved applying various transformations and modifications to the training images, thereby expanding the dataset artificially. By doing so, we aimed to improve the model's ability to generalize and reduce the risk of overfitting. Augmentation techniques play a crucial role in training deep neural networks, as they enable the model to learn from a more diverse range of examples. The model has trained for 300 epochs over 4000 iterations. Preprocessing techniques, including image resizing and normalization, are performed. Mix images (an image that contains multiple categories) were used to generalize the model. It is tested on another set of unseen images taken from the MS COCO dataset. The proposed system transforms the recognized items into audio translation utilizing the text-to-speech conversion process. For example, if the camera identifies a chair, the system will announce "chair" loudly. To assist VIPs, the proposed model can be implemented in smart glasses or a smart cane using Arduino, Raspberry Pi devices, or any microcontroller device. The rest of the paper is organized as follows: Section 2 contains related work; Section 3 contains a proposed method; Section 4 presents results and discussion; Section 5 concludes the proposed work.

2. Related Studies

Smart navigators based on object detection and recognition are now widely used to guide VIPs to their destinations. These systems could be implemented as smart glasses or

smart canes using microcontroller devices such as Arduino and Raspberry Pi. Researchers adopt different techniques to enhance the lives of VIPs. DCNN-based models are widely used with satisfactory results for obstacle detection equipped with text-to-speech modules to generate an audio message to inform the blind person regarding the detected objects. The study conducted in [9] used a custom dataset of six categories and trained a YOLOv3 model for 3000 iterations and obtained 99% accuracy. They used the OpenCV library for frame capture, a Python module for Text-to-speech, and the Raspberry Pi as a hardware device for smart eyeglasses. The study conducted in [10] used a modified version of RCNN. Edge boxes were used instead of selective search for region proposals. SSD-MobileNet was used as the backbone of the model, and a softmax classifier was used instead of a support vector machine (SVM) to classify the object. All this development was conducted using a DL framework called Café. An Arabic text-to-speech generator was used to facilitate the VIPs. Edge boxes and selective search produce the same accuracy, but edge boxes are ten times faster than selective search. The research conducted in [11] used SSD-RNN to provide smart navigators to VIPs. They trained an SSD model for object detection, and an RNN was trained for text-to-speech conversion. They used the COCO dataset for SSD model training and obtained 95.06% indoor accuracy and 87.68% outdoor accuracy. The ImageNet dataset for training a CNN model, which consists of 1000 classes/categories, was used in [12]. A transfer learning technique based on MobileNet with fine-tuning was used and achieved 83.3% accuracy. They suggested the Raspberry Pi as a microcontroller device to execute the model. The drawback of public datasets such as ImageNet and COCO is that they contain many categories that may not pose an obstacle for VIPs but greatly affect the training process of the CNN model.

The Mask-RCNN model was employed in [13] to detect and monitor the progress of work in the construction industry. They used a custom dataset of 1143 images. Image preprocessing and data augmentation were used to enhance the model's accuracy with generalized detection results and achieve 90.6% accuracy. The study conducted in [14] trained YOLOv3 for the development of a smartphone-based smart navigator with Text-to-speech conversion for VIPs. They used the PASCAL VOC 2007 and 2012 datasets, which consist of 20 classes, and combined them into a single dataset. In their work, 50% of the images were used for training and validation, and 50% were used for testing. YOLOv4 for the facilitation of VIPs in outdoor environments was employed in [15]. They used a custom dataset of two classes: gutter and bollard. They archived 80% and 72% accuracy, respectively, for gutters and bollards. The study conducted in [16] used pre-processing techniques to enhance the quality of the input image. The CCTV image was converted to grayscale and reshaped to 64 $\times$ 64. Each image was normalized, and a CNN model was developed and trained with one input layer, several hidden layers, and one output layer. They implemented the model to ensure the use of face masks. They achieved 98% accuracy. The study performed in [17] compared various AI-based classifiers for face mask detection in crowded areas, and InceptionV3 was declared the most successful model for face mask classification with 99% training accuracy. The study conducted by [18] trained and compared three models named Retina-Net, Faster RCNN, and YOLOv3 for commodity detection and recognition. They used ResNet101 as the backbone for the first two models, and Darkent-53 for YOLOv3. They concluded that Yolo outperforms Faster RCNN and Retina-Net in terms of speed and accuracy. A hybrid model for road obstacle detection was employed in [19]. They used the Dlib library for drowsy face recognition and driver distraction, and a pre-trained YOLOv3 model was used for object detection. The object detection module is responsible for obstacle detection, and the second module based on image segmentation and edge detection was used to analyze the object as an obstacle or any other routine object such as roadside fences. If the object is analyzed as an obstacle, the driver is issued an alarm, and an Arduino microcontroller device attached to the vehicle will take control of the vehicle to overcome the chance of an accident. They achieved recall and precision of 92.2%, 97.4%, 95.8%, and 87.5%, respectively. The research done in [20] trained a model for multi-food detection and calorie counting. They manually encircled

the region of interest (ROI) and removed the background. Selective search was used to create region proposals, and SVM was used for the classification of these regions. A module named "region mining" was used to discard the less important regions, and the remaining regions were passed to a CNN model to classify the food type with its rate of calories. A public dataset containing 7000 images of 30 classes called FooDD was used for training and testing the model.

Faster RCNN was used for object detection, and the ROI was cropped and passed to ResNet for classification. The hill-climbing algorithm was used for ensemble predictions. Determining accessibility, including usability assessments, is an essential equitable step in evaluating and enhancing the efficacy and efficacy of online learning and general resources for learners who have disabilities [21]. In order to help blind and visually disabled people in new big public structures with complicated horizontal and vertical connectedness, a method for creating an interior guidance system that is conscious of its surroundings was developed [22]. The suggested building data model seeks to address the shortcomings of current Indoor Navigation Systems (INS). The study conducted in [23] used deep learning based on CNN using Keras with Tensor-flow for face mask detection to classify people as wearing or not wearing masks. More than 2000 images are collected from various sources, including Kaggle.com, with 95% accuracy. Pre-processing techniques were used to de-noise the images, and reshaping was used for image normalization before feeding the model for training.

2.1. Limitations of Previous Works

Several efforts have been made to create visual navigators for VIPs utilizing CNN-like methods. These methods still have some flaws, which restrict their abilities to assist VIPs in their everyday tasks. Some of these constraints are as follows:

- In a variety of real-world settings, current object identification methods may not reliably identify and localize objects [24]. VIPs who significantly rely on precise object identification for navigation and routine tasks face difficulties as a result of this constraint [25].
- Many of the earlier models are trained on limited datasets, therefore they are not adaptable well to different settings or circumstances. This restricts the models' capacity to help VIPs navigate new settings or environments [10].
- The previously created models are frequently susceptible to changes in illumination, camera perspectives, and other external variables. Because of such limitations, those models are not always able to supply VIPs with trustworthy help in different real-world circumstances [5].
- Many earlier methods have poor processing rates, which make them unsuitable for real-time applications [26].
- Prior research could not have effectively considered the speech features of providing VIPs with object information. VIPs must obtain brief and clear audible input in order to fully comprehend the qualities and context of the object [27].

2.2. Novelty and Contributions

This work improves object detection accuracy by utilizing a novel model YOLOv4_Resnet101, which results in more precise and reliable identification of objects in an image or video frame. The suggested system makes a significant contribution to the field of assistive technologies for VIPs with several novel characteristics and advancements. The following are the key contributions of this work:

- This work focuses on ensuring real-time performance for object identification, enabling the quick and efficient processing of images. This is essential for VIPs because they need immediate information about their environment.
- To recognize objects in the user's surroundings immediately, the system makes use of the extremely accurate and quick real-time object detection technology known as YOLOv4_Resnet101. This makes it possible for the system to give the user up-to-

date information about their surroundings, which is essential for safe and independent travel.

- The proposed system uses state-of-the-art object detection and natural language processing methods to deliver instantaneous object detection and auditory feedback, enabling visually disabled people to freely navigate their environments.
- The incorporation of text-to-speech conversion technology within the system makes a substantial contribution. It enables visually impaired people to observe and comprehend the visual world through audible methods by converting the identified items into audible speech. This improves VIP accessibility and makes it possible for them to interact with their surroundings more successfully.
- The proposed model effectively detected and categorized objects in real-world contexts, reaching a remarkable accuracy of 96.34% on test images taken from the MS COCO dataset.

3. Proposed Work

In the proposed method, a YOLOv4_Resnet101 model with custom parameters is trained for obstacle detection and recognition to facilitate VIPs. The MS COCO dataset is used to validate the model performance. Image preprocessing techniques are used to remove noise and normalize the images for further processing. Data augmentation techniques are used to improve the training process. Text-to-speech conversion is used to alert the VIPs regarding the obstacle through audio responses. The model can be installed in the form of smart eyeglasses or a smart cane on a microcontroller such as Arduino or Raspberry Pi. The block diagram of the proposed system is shown in Figure 1.

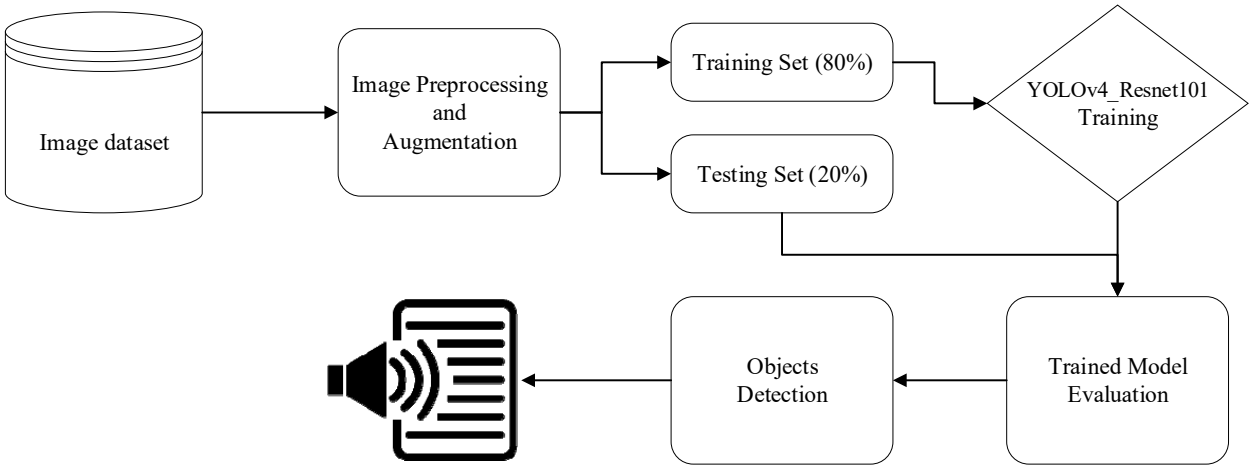

Figure 1. Block diagram of the proposed work.

3.1. Dataset and Pre-Processing

In this work, the MS COCO dataset [28] was used to train, test, and validate the model. This dataset contains a vast number of images depicting numerous everyday objects in various scenarios. To assist in accurate object localization, objects are labeled using per-instance segmentations. There are 328 k images in this dataset, representing 91 different object kinds, with a total of 2.5 million labeled instances. In contrast to the well-known ImageNet dataset, MS COCO contains fewer categories but more instances per category. This can help in learning intricate object models that are capable of exact 2D localization. In this study, the training images were resized to 416 × 416 pixels. During training, the images are normalized using a mean of [0.485, 0.456, 0.406] and a standard deviation of [0.229, 0.224, 0.225]. These normalization settings assist in the preprocessing of images and putting them into a standard range for efficient model training.

3.2. Data Augmentation

In this study, data augmentation techniques are applied to expand the dataset and introduce variations to the existing images. This was used to overcome overfitting and improve the model's generalization. We applied zooming to enlarge or shrink the objects of interest within the images. Random rotations were applied to the images to simulate different viewpoints. We used rotate = (−10, 10) to rotate the images by −10 to +10 degrees. We used shear = (−5, 5) to shear the images by −5 to +5 degrees. Image shifting was used to simulate changes in object positions. To detect objects irrespective of their orientation, image flipping was utilized.

3.3. Object Detection Using YOLOv4

Identifying and detecting objects of interest inside an image or a video is a computer vision (CV) problem known as object detection. YOLO is a famous object detection technique because it can identify objects in real-time and with remarkable accuracy. YOLO adopts an alternative strategy by carrying out object detection in a single run of the neural network, as opposed to conventional object detection techniques, which need numerous phases and complex computations [29]. Recent methods, such as R-CNN, employ region proposal techniques to create feasible bounding boxes in an image before applying a classifier to the suggested boxes. After classification, bounding boxes are improved, duplicate detections are removed, and the boxes are given new scores depending on additional items in the scene [30]. Due to the need to train each component independently, these complex pipelines are sluggish and challenging to optimize. The study conducted in [24] developed YOLO in 2015. They reframe object identification as a single regression problem, shifting from image pixels to determining bounding box coordinates and object class probabilities. Later, improvements were made step by step, and five variants were developed for object detection in real-world applications. It is a one-stage object detection system, meaning it processes an entire image in one forward pass through the neural network, rather than requiring multiple passes or separate regions of interest to be identified first [31]. This allows YOLO to perform object detection in real-time on video, with a processing speed of up to 45 frames per second. The architecture of YOLO is based on a single convolutional neural network that divides an image into a grid and uses each cell in the grid to make predictions about the presence and location of objects in the image.

3.3.1. Workflow of YOLO

The workflow of the YOLO framework is broadly divided into four steps:

1. Input processing: The input image is resized to a fixed size, such as 416 × 416 pixels, to ensure that the aspect ratio of objects in the image is preserved.
2. Neural Network: In this step, the DCNN is deployed to make predictions about the presence and location of objects in an image. The DCNN is trained on a large dataset of annotated images to learn to recognize objects and predict their bounding boxes.
3. Grid Cell Assignment: In this step, the image is divided into a grid of cells, where each cell is responsible for predicting the presence and location of objects in a specific region of the image. For each grid cell, YOLO predicts multiple bounding boxes, along with the class probabilities of the objects present in each box.
4. Object Detection: These predictions are then filtered using non-maximal suppression to remove overlapping bounding boxes and obtain the final detection result. Finally, the resulting bounding boxes are drawn on the image, along with the class labels of the detected objects. The whole process is repeated for each frame in a video stream, allowing YOLO to perform real-time object detection.

3.3.2. Workflow of YOLOv4

In YOLOv4, the architecture is organized into three main blocks: the backbone, the neck, and the head.

1. Backbone: The backbone of the Yolo is typically composed of a series of convolutional layers, which are used to extract high-level features from the input images. The convolutional layers apply a set of filters to the input, sliding them over the input to detect patterns and features at different spatial locations. In YOLOv4, the backbone of YOLOv3, called Darket-53, is replaced by CSPDarknet-53. Cross-stage partial connections (CSP) involve connecting different stages or blocks of the network through a shortcut connection, where a portion of the output from one stage is combined with the output from another stage. This technique improves information flow between the different stages of the network and enables better reuse of features learned at different levels of the network. This can help to improve the overall performance of the network, especially in tasks such as object detection and segmentation, where precise localization and feature reuse are important.
2. Neck: Following the backbone, the neck module acts as a link between the high-resolution characteristics of the backbone and the next detection layers. The backbone extracts feature from the input image, while the detection head predicts bounding boxes, abjectness scores, and class probabilities based on those features. The neck in YOLOv4 consists of three different components: the SPP (Spatial Pyramid Pooling) module, the PAN (Path Aggregation Network) module, and the SAM (Spatial Attention Module) module. The SPP module performs max-pooling at different levels of the feature map to capture context information at different scales. The PAN module is designed to aggregate features from different scales and resolutions, which can help improve the accuracy of object detection. The SAM module is a form of attention mechanism that helps the network focus on important features while suppressing irrelevant information.
3. Head: The head module is responsible for producing the final predictions for object detection. It uses the feature maps from the neck module and adds additional convolutional layers, commonly followed by fully connected layers, to predict the bounding box coordinates, class probabilities, and other pertinent details for each grid cell. The head module often uses anchor boxes or previous boxes to help in predicting bounding box locations and sizes.

3.3.3. Modifications

The existing YOLOv4 showed effective performance for the task of object detection, however, it has many parameters and a complicated network structure, which makes it computationally expensive. In the proposed study, we modify the architecture and training procedure of the YOLOv4 model for object detection.

1. Architecture modifications:

Backbone Network: YOLOv4 commonly uses the Darknet-53 backbone, however, we explored using ResNet-101 as the backbone to extract features. The input tensor X was processed through a convolutional layer with filters of size 7×7 and stride 2, followed by batch normalization and ReLU activation. To decrease the spatial dimensions, max pooling with a pool size of 3×3 and stride 2 was used. The residual blocks are stacked to generate feature extraction.

Spatial resolution: In order to identify objects of various sizes, we changed the size of the grid. We also modified the network's downsampling rate and the size of the input image.

Anchor boxes: Anchor boxes were used by YOLOv4 to predict the sizes and positions of objects. Based on the attributes of our dataset, we modified the aspect ratios and scales of anchor boxes.

Model pruning: Model pruning was used to reduce the size and complexity of a trained model without significantly reducing its performance. We applied various pruning approaches, such as weight pruning, neuron pruning, channel pruning, iterative pruning, and layer-wise structure pruning. Using the channel pruning techniques, we located and removed unnecessary channels from the network along with the input and output interac-

tions. Channels with high contributions are kept, while channels with low contributions are eliminated, according to the distribution of the Gamma coefficient and the desired pruning rate. Neurons from channels with lower contributions are not included in the connections during the connection step. Figure 2 represents the results of the proposed model.

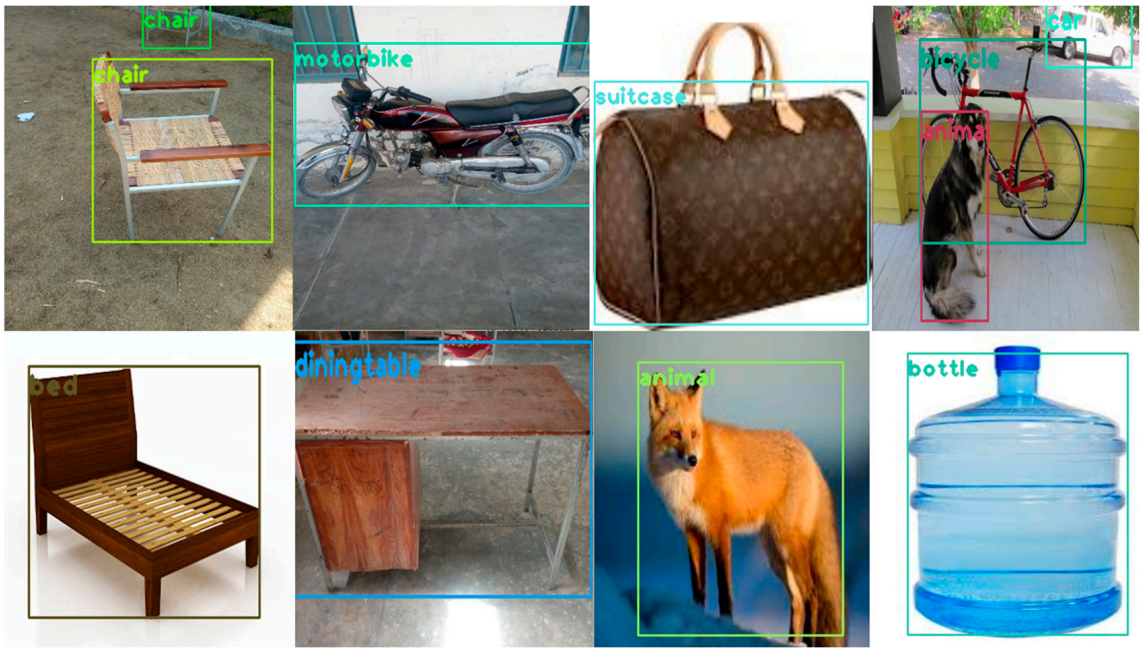

Figure 2. Object detection by the proposed model.

2. Training modifications:

Data augmentation: Use a variety of data augmentation methods to enhance the diversity of our training data, such as random cropping, flipping, rotation, and scaling.

Hyperparameter tuning: To improve the performance of the model, experiment with various learning rates, batch sizes, optimizer settings (such as Adam, SGD), and regularization methods (such as weight decay and dropout).

Transfer learning: Starting with pre-trained weights on a big dataset (such as MS COCO), we are able to fine-tune the model using our target dataset. This strategy aids in the model's faster convergence and improved performance.

3.4. Model Training

The training images and the associated annotations were loaded and preprocessed using a data loader. Set the learning rate, optimizer, batch size, and number of epochs for the training procedure. Pass the training data through the model, calculate the loss, then update the weights using backpropagation. Analyze the loss and other indicators as the training progresses. Iterate through the training data and repeat the process for several epochs.

Evaluation: Periodically assess the model's performance on a validation set using measures such as accuracy, average precision, recall, and F1 score. To enhance the performance of the model, modify the hyperparameters and training scheme following the evaluation results.

Model saving: Save the trained updated YOLOv4 model weights and settings for future usage and deployment.

The model was trained for 4000 iterations with a loss error rate of 0.0730 and a mean average precision (mAP) of 0.97. The experiments of the proposed work's experiments are implemented using Python 3.6 and PyTorch 2.0. The training was conducted on a computer equipped with an Intel Core i9-7900X processor, 128 GB of RAM, 2 Nvidia Titan V graphics

cards, and CUDA 9.0. The dataset was randomly divided into training (80%) and testing sets (20%). Table 1 outlines some of the parameters used in the experiment for training YOLOv4 with ResNet-101.

Table 1. Parameters of YOLOv4_Resnet101 for object detection.

Parameter	Description
YOLOv4_Resnet101	YOLOv4 with ResNet-101 as the backbone
Input-image (416, 416)	The size of input images provided for training
[(10, 13), (16, 30), (33, 23)]	Anchor Boxes are used for predicting bounding box coordinates.
kernel_size = (7, 7)	Filters size
Pool-size = (3 × 3), stride (2, 2)	Max pooling with a pool size of 3 × 3 and stride 2 is used
train_ratio = 0.8	Training set
test_ratio = 0.2	Testing set
num_epoch = 300	Number of epochs
num_iterations = 4000	Number of iterations
lr = 1×10^{-4}	Learning rate
lr_decay_epoch = 50	Every 50 epochs, the learning rate decays
batch_size = 32, 64	Samples processed in a single forward and backward pass
Adam Optimizer	To optimize the parameters of the network

3.5. Text-to-Speech Conversion

The combination of text-to-speech conversion technology with object detection attempts to convert observed objects into audio-based messages that visually impaired users may perceive. This combination improves accessibility and comprehension of objects in the environment. The method of combining text-to-speech conversion with object identification is described in the steps that follow.

Text Generation: Once objects have been discovered, the following step is to generate text descriptions for each detected object. This is accomplished by employing predetermined textual descriptions linked to each class label.

Speech Conversion: Using speech synthesis techniques, the resulting written descriptions of the observed items are translated into speech. The synthesized speech, which represents the written descriptions of the items, is heard through headphones. Using the pyttsx3 library, we converted text to voice in Python by setting the speech rate ('rate', 150).

4. Results and Discussions

The proposed model, YOLOv4_Resnet101, was tested using the MS COCO dataset. This dataset comprises a wide range of images with labeled object categories, allowing the model to learn from a large number of examples. Training the model on a big and diverse dataset, such as MS COCO, allows it to generalize better to real-world settings, resulting in improved accuracy. After 4000 iterations of training, the model obtained an average precision of 95.6%, recall of 97.10%, and F1 score of 96.34%, indicating a strong performance in detecting and localizing objects in the images. The achieved accuracy demonstrates the model's ability to accurately recognize and categorize objects found in images. Additionally, the model had a low loss error rate of 0.073%, demonstrating that it was successful in learning to capture the intricate visual patterns required for precise object recognition. The proposed YOLOv4_Resnet101 model achieved the maximum training accuracy of 99.03% for person detection, followed by trucks with an accuracy of 99.01% and automobile detection with an accuracy of 98.17%. The model consistently demonstrated outstanding accuracy for the various categories of objects listed in Table 2. The proposed model achieves an overall mean average precision (mAP) of 96.025%, emphasizing its outstanding performance in reliably recognizing and classifying objects in the dataset. The proposed scheme benefits from the ResNet-101 backbone network's deeper design and increased feature extraction capabilities. This improves the encoding and comprehension of complicated visual patterns, leading to increased accuracy in object recognition tasks.

Table 2. Category-wise accuracy of the proposed model.

S. No.	Obstacle Category	Average Accuracy
1	Person	99.03
2	Car	98.17
3	Bus	95.81
4	Truck	99.01
5	M-Cycle	93.00
6	Cycle	97.80
7	Table	95.80
8	Desk	98.90
9	Chair	93.17
10	Bed	92.76
11	Tree	97.55
12	Animal	96.87
13	Pillar	97.00
14	Gutter	87.00
15	Fan	96.73
16	Refrigerator	97.00
17	mAP	96.025

4.1. Comparison with the Related Frameworks

The performance of the proposed model for object detection and accessibility for people with visual impairments was compared with that of existing approaches. It was compared with other state-of-the-art models called YOLOv5, SSD, and Faster RCNN. YOLOv5 is the unofficial copy of YOLOv3 with some modifications that have been implemented in PyTorch. This model achieves a high F1 score of 93.99%. It is a very light model and needs less time to train for a custom dataset, but in most cases, it could not compete with the YOLOv4 in accuracy. SSD is a popular single-shot object detector which is trained with backbones such as VGG-16, Inception, ResNet, etc. This model achieves a higher F1 score of 89.54%. Faster RCNN is the third version of RCNN released in 2015. It is a two-stage detector. Two-stage detectors are slower than one-stage detectors. Before the release of YOLOv4, it was the most famous model regarding accuracy. Using Faster RCNN, the higher F1 score of 93.82% was achieved, as shown in Table 3 and Figure 3. The outcomes showed that the suggested method produced more accurate object identification with an F1 score of 96.34% and more thorough and accurate descriptions using text-to-speech conversion. For input-image sizes of (416, 416), the YOLOv4 reaches real-time inference speeds ranging from 20 to 40 frames per second (FPS). Similar to this, the YOLOv5 reaches inference speeds between 30 and 50 FPS for the given images. MobileNetV2-SSD obtained faster inference times of 50 to 70 FPS at the expense of significantly worse accuracy. Compared to the other models, the faster R-CNN is a region-based object identification model that achieves excellent accuracy but has somewhat longer inference times (5 to 10 FPS). The proposed model, which has real-time object identification capabilities, reaches inference speeds of 60 to 80 FPS for 416 $\times$ 416 input images.

Table 3. Comparison with other closely related frameworks.

Model	Inference Speed (FPS)	Precision (%)	Recall (%)	F1-Score (%)
Proposed Model	60–80	95.6	97.1	96.34
YOLOv4	20–40	93.41	94.07	93.74
YOLOv5	30–50	95	93	93.99
MobileV2-SSD	50–70	88.31	90.8	89.54
Faster RCNN	5–10	95	92.67	93.82

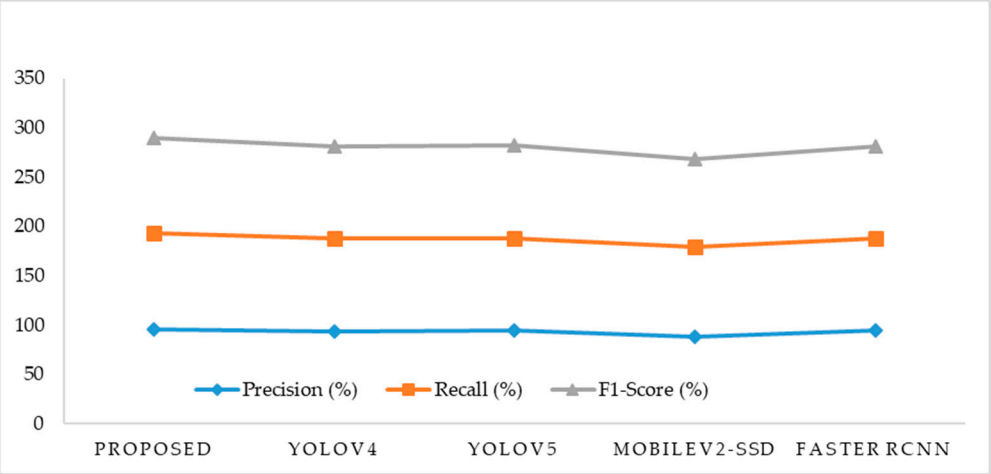

Figure 3. Comparison with closely related frameworks.

4.2. Comparison with State-of-the-Art Works

We compared the suggested strategy with cutting-edge efforts in the area of object detection. We compared the performance of several models based on their technique, dataset used, and obtained accuracy. Table 4 represents the comparison of the proposed work with state-of-the-art works.

Table 4. Comparison with state-of-the-art works.

Study	Year	Method	Dataset	Accuracy
Shadi et al. [5]	2019	DeepLabV3+	Custom	79%
Wong et al. [10]	2019	Edge Box- SSD	CIFAR	73.7
Ashiq, F. et al. [12]	2022	MobileNet	ImageNet	83.3%
Suman, S. et al. [11]	2022	SSD-RNN	MS COCO	95.06
Adeyanju et al. [15]	2022	Custom CNN	Custom	84%
Kuriakose et al. [1]	2023	Efficient Net	Custom	87.8%
Proposed method	2023	YOLOv4_Resnet101	MS COCO	96.34%

The study conducted by Shadi et al. [5] deployed the DeepLabV3+ for object detection on a custom dataset consisting of 15 classes. They achieved a high accuracy of 79%. Wong et al. [10] utilized Edge Box-SSD for object detection on the CIFAR dataset containing 10 classes. They achieved a high accuracy of 73.7%. Compared to real-world situations, the CIFAR dataset is renowned for its images that are small and low resolution. The Edge Box-SSD technique may not perform well in increasingly complicated and high-resolution situations seen in real-world object identification applications. Adeyanju et al. [15] applied CNN for object detection on a custom dataset consisting of two classes. They obtained a high accuracy of 84%. Their model's capacity to generalize and recognize items in other datasets or in real-world situations is hampered by its narrow scope. When presented with distinct classes or variances in object appearance and context, their model's performance suffers. Kuriakose et al. [1] utilized Efficient Net for object detection on a custom dataset containing 20 classes. Their approach achieved an accuracy of 87.8% for object detection. The study conducted in [11] used Single Shot Multibox Detector (SSD-RNN) on the MS COCO dataset and attained an accuracy of 95.06%. SSD-RNN analyzes images sequentially, which introduces delay and limits real-time performance, particularly when working with a high number of objects or in circumstances that require fast processing rates. In comparison to existing methods, our suggested technique, which employs a modified YOLOv4_Resnet101 architecture on the MS COCO dataset, obtains a maximum accuracy of 96.34%. This highlights the better performance and efficiency of our technique for identifying objects across multiple scenes and classifications. YOLOv4_Resnet101 detects

objects in real-time by scanning the image in a single pass. This makes it quicker than SSD-RNN, which analyses images sequentially and creates delays. YOLOv4_Resnet101 benefits from parallel processing since it predicts bounding boxes and class probabilities across many sizes and aspect ratios at the same time. Compared to SSD-RNN, it also offers a deeper and more effective feature extraction network.

4.3. Discussion

Imagine Sarah, a person with vision impairment, trying to find her way alone across a crowded downtown street. Sarah travels with a little gadget that includes a camera, a speaker, and the YOLOv4_Resnet101 model. The machine's camera records live video of Sarah walking along the street. Object recognition and localization on the video frames are carried out constantly in the background by the YOLOv4_Resnet101 model. It recognizes a variety of things around Sarah, including individuals, vehicles, traffic signals, and obstructions such as poles or garbage cans. After detecting and localizing the objects, the model employs the text-to-speech conversion function to turn the object labels into audio information. Sarah is then given real-time auditory feedback about the objects in her vicinity through the device's speaker. For example, the device then states, "Pedestrian on your left", "Car approaching from the right" or "Obstacle ahead, please proceed with caution". With the help of the device's audio feedback, Sarah can explore her surroundings and her path to make a smart journey. The model assists her in detecting and avoiding possible risks, making her street crossings and general movement more efficient and secure.

5. Conclusions

In this study, we introduced a novel approach for improving object detection which particularly works for VIPs. We developed a system that offers real-time auditory feedback on identified objects by fusing the powerful YOLOv4_Resnet101 object detection model with text-to-speech conversion methods. The usefulness of the suggested approach in supporting VIPs with object identification and localization tasks has been demonstrated through our thorough experimental assessment and user studies. The YOLOv4_Resnet101 integration enables precise and effective object detection, enabling the system to function in real-time. The text-to-speech translation module transforms item names into sounds, enabling VIPs to perceive and comprehend their environment. The user experience is improved by the system's customization features, which provide users the freedom to adjust it to their own preferences. ResNet-101 makes use of residual connections to avoid and reuse the characteristics of earlier layers, allowing the network to learn more effectively. This promotes the training of deeper networks and alleviates the vanishing gradient issue. We were able to represent complicated visual patterns more effectively using the ResNet-101's enhanced depth and capacity, which increases the precision of object recognition. Additionally, YOLOv4_Resnet101 strikes a nice compromise between inference speed and detection accuracy.

In order to help VIPs to see and comprehend their surroundings better and engage in richer interactions with the identified items, aural signals and touch-based feedback can be incorporated in the future.

Author Contributions: Conceptualization, T.J.A.; methodology, A.U.R. and H.K.; software, H.K.A.; validation, T.J.A., A.U.R. and H.K.; formal analysis, A.U.R.; investigation, T.J.A. and H.K.A.; resources, H.K.; writing—original draft preparation, T.J.A. and A.U.R.; writing—review and editing, A.U.R. and H.K.; visualization, A.U.R. and H.K.A.; funding acquisition, T.J.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the King Salman center For Disability Research for funding this work through Research Group no KSRG-2023-021.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kuriakose, B.; Shrestha, R.; Sandnes, F.E. DeepNAVI: A deep learning based smartphone navigation assistant for people with visual impairments. *Expert Syst. Appl.* **2023**, *212*, 118720. [CrossRef]
2. Khan, G.; Tariq, Z.; Khan, M.U.G. *Multi-Person Tracking Based on Faster R-CNN and Deep Appearance Features*; Intechopen: London, UK, 2019. [CrossRef]
3. Tambe, P.M.; Khade, R.; Shinde, A.; Ahire, A.; Murkute, N. Third eye: Object recognition and tracking system to assist visually impaired people. *Int. Res. J. Mod. Eng. Technol. Sci.* **2022**, *218*, 1–5.
4. Rathi, M.; Sahu, S.; Goel, A.; Gupta, P. Personalized Health Framework for Visually Impaired. *Informatica* **2022**, *46*. [CrossRef]
5. Tapu, R.; Mocanu, B.; Zaharia, T. DEEP-SEE: Joint Object Detection, Tracking and Recognition with Application to Visually Impaired Navigational Assistance. *Sensors* **2017**, *17*, 2473. [CrossRef] [PubMed]
6. Shadi, S.; Hadi, S.; Nazari, M.; Hardt, W. Outdoor Navigation for Visually Impaired Based on Deep Learning. 2019. Volume 2514, pp. 97–406. Available online: https://ceur-ws.org/Vol-2514/paper102.pdf (accessed on 2 June 2023).
7. Deepa, R.; Tamilselvan, E.; Abrar, E.; Sampath, S. Comparison of yolo, ssd, faster rcnn for real time tennis ball tracking for action decision networks. In Proceedings of the International Conference on Advances in Computing and Communication Engineering (ICACCE), IEEE, Sathyamangalam, India, 4–6 April 2019; pp. 1–4.
8. Kim, J.; Sung, J.Y.; Park, S. Comparison of Faster-RCNN, YOLO, and SSD for real-time vehicle type recognition. In Proceedings of the IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Seoul, Republic of Korea, 1–3 November 2020; pp. 1–4.
9. Abdul-Ameer, H.S.; Hassan, H.J.; Abdullah, S.H. Development smart eyeglasses for visually impaired people based on you only look once. *Telkomnika Telecommun. Comput. Electron. Control* **2022**, *20*, 109–117. [CrossRef]
10. Wong, Y.; Lai, J.; Ranjit, S.; Syafeeza, A.; Hamid, N. Convolutional neural network for object detection system for blind people. *J. Telecommun. Electron. Comput. Eng.* **2019**, *11*, 1–6.
11. Suman, S.; Mishra, S.; Sahoo, K.S.; Nayyar, A. Vision Navigator: A Smart and Intelligent Obstacle Recognition Model for Visually Impaired Users. *Mob. Inf. Syst.* **2022**, *2022*, 9715891. [CrossRef]
12. Ashiq, F.; Asif, M.; Bin Ahmad, M.; Zafar, S.; Masood, K.; Mahmood, T.; Mahmood, M.T.; Lee, I.H. CNN-Based Object Recognition and Tracking System to Assist Visually Impaired People. *IEEE Access* **2022**, *10*, 14819–14834. [CrossRef]
13. Shamsollahi, D.; Moselhi, O.; Khorasani, K. A Timely Object Recognition Method for Construction using the Mask R-CNN Architecture. In Proceedings of the International Symposium on Automation and Robotics in Construction, Dubai, United Arab Emirates, 2–4 November 2021; pp. 372–378. [CrossRef]
14. Rachburee, N.; Punlumjeak, W. An assistive model of obstacle detection based on deep learning: YOLOv3 for visually impaired people. *Int. J. Electr. Comput. Eng.* **2021**, *11*, 3434–3442. [CrossRef]
15. Adeyanju, I.A.; Azeez, M.A.; Bello, O.; Badmus, T.A.; Oyediran, M. Development of a Convolutional Neural Network-Based Object Recognition System for Uncovered Gutters and Bollards. *ABUAD J. Eng. Res. Dev.* **2022**, *5*, 147–154.
16. Rahman, M.M.; Manik, M.M.H.; Islam, M.M.; Mahmud, S.; Kim, J.-H. An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network. In Proceedings of the 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, Canada, 9–12 September 2020; pp. 1–5. [CrossRef]
17. Mbunge, E.; Simelane, S.; Fashoto, S.G.; Akinnuwesi, B.; Metfula, A.S. Application of deep learning and machine learning models to detect COVID-19 face masks—A review. *Sustain. Oper. Comput.* **2021**, *2*, 235–245. [CrossRef]
18. Xie, L. Analysis of Commodity image recognition based on deep learning. In Proceedings of the 6th International Conference on Multimedia and Image Processing, Zhuhai, China, 8–10 January 2021; pp. 50–55.
19. Wang, T.; Zheng, N.; Xin, J.; Ma, Z. Integrating Millimeter Wave Radar with a Monocular Vision Sensor for On-Road Obstacle Detection Applications. *Sensors* **2011**, *11*, 8992–9008. [CrossRef] [PubMed]
20. Pouladzadeh, P.; Shirmohammadi, S. Mobile Multi-Food Recognition Using Deep Learning. *ACM Trans. Multimedia Comput. Commun. Appl.* **2017**, *13*, 1–21. [CrossRef]
21. Alahmadi, T.; Drew, S. Subjective evaluation of website accessibility and usability: A survey for people with sensory disabilities. In Proceedings of the 14th International Web for All Conference, Perth, Australia, 28 May–1 June 2017; pp. 1–4.
22. Ivanov, R. An approach for developing indoor navigation systems for visually impaired people using Building Information Modeling. *J. Ambient. Intell. Smart Environ.* **2017**, *9*, 449–467. [CrossRef]
23. Bhadani, A.K.; Sinha, A.J. A facemask detector using machine learning and image processing techniques. *Eng. Sci. Technol. Int. J.* **2020**, 1–8.
24. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
25. Arora, A.; Grover, A.; Chugh, R.; Reka, S.S. Real Time Multi Object Detection for Blind Using Single Shot Multibox Detector. *Wirel. Pers. Commun.* **2019**, *107*, 651–661. [CrossRef]

26. Afif, M.; Ayachi, R.; Said, Y.; Pissaloux, E.; Atri, M. An Evaluation of RetinaNet on Indoor Object Detection for Blind and Visually Impaired Persons Assistance Navigation. *Neural Process Lett.* **2020**, *51*, 2265–2279. [CrossRef]
27. Alzahrani, N.; Al-Baity, H.H. Object Recognition System for the Visually Impaired: A Deep Learning Approach using Arabic Annotation. *Electronics* **2023**, *12*, 541. [CrossRef]
28. Lin, Y.T.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Part V 13; pp. 740–755.
29. Huang, R.; Pedoeem, J.; Chen, C. YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers. In Proceedings of the 2018 IEEE International Conference on Big Data, Seattle, WA, USA, 10–13 December 2018; pp. 2503–2510.
30. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
31. Ahmad, T.; Ma, Y.; Yahya, M.; Ahmad, B.; Nazir, S.; Haq, A.U. Object Detection through Modified YOLO Neural Network. *Sci. Program.* **2020**, *2020*, 8403262. [CrossRef]