electronics

MDPI

Article

Object Recognition System for the Visually Impaired: A Deep Learning Approach using Arabic Annotation

Nada Alzahrani and Heyam H. Al-Baity *

Information Technology Department, Collage of Computer and Information Sciences, King Saud University, Riyadh P.O. Box 145111, Saudi Arabia

* Correspondence: halbaity@ksu.edu.sa

Abstract: Object detection is an important computer vision technique that has increasingly attracted the attention of researchers in recent years. The literature to date in the field has introduced a range of object detection models. However, these models have largely been English-language-based, and there is only a limited number of published studies that have addressed how object detection can be implemented for the Arabic language. As far as we are aware, the generation of an Arabic text-to-speech engine to utter objects' names and their positions in images to help Arabic-speaking visually impaired people has not been investigated previously. Therefore, in this study, we propose an object detection and segmentation model based on the Mask R-CNN algorithm that is capable of identifying and locating different objects in images, then uttering their names and positions in Arabic. The proposed model was trained on the Pascal VOC 2007 and 2012 datasets and evaluated on the Pascal VOC 2007 testing set. We believe that this is one of a few studies that uses these datasets to train and test the Mask R-CNN model. The performance of the proposed object detection model was evaluated and compared with previous object detection models in the literature, and the results demonstrated its superiority and ability to achieve an accuracy of 83.9%. Moreover, experiments were conducted to evaluate the performance of the incorporated translator and TTS engines, and the results showed that the proposed model could be effective in helping Arabic-speaking visually impaired people understand the content of digital images.

Keywords: artificial intelligence; object detection; visually impaired; deep learning; object recognition; Mask R-CNN; assistive technologies

Citation: Alzahrani, N.; Al-Baity, H.H. Object Recognition System for the Visually Impaired: A Deep Learning Approach using Arabic Annotation. *Electronics* **2023**, *12*, 541. https://doi.org/10.3390/electronics12030541

Academic Editors: Daniel Hládek, Matúš Pleva, Piotr Szczuko and Andrej Zgank

Received: 15 November 2022
Revised: 15 January 2023
Accepted: 16 January 2023
Published: 20 January 2023

1. Introduction

The development of computer vision systems has been necessitated by the requirement to make sense of the massive number of digital images from every domain. Computer vision involves the scientific study of developing ways that can enable computers to see, understand, and interpret the content of digital images, including videos and photographs.

To obtain a high-level understanding of digital images or videos, it is not sufficient to focus on classifying different images; instead, the categories and locations of objects contained within every image should be specified [1]. This task is referred to as object detection, which is a major area of interest within the field of computer vision. Object detection is the process of finding various individual objects in an image and identifying their location through bounding boxes with their class labels [2]. Additionally, object detection methods can be implemented either by employing traditional machine-learning techniques or using deep learning algorithms.

In recent years, many researchers have leveraged deep learning algorithms such as convolutional neural networks (CNNs) to solve object detection problems and proposed different state-of-the-art object detection models. These object detection models can be classified into two major categories: two-stage detectors such as R-CNN [3], Fast R-CNN [4], Faster R-CNN [5], Mask R-CNN [6], and G-R-CNN [7], and one-stage detectors such

as YOLO [8], SSD [9], YOLOv4 [10], and YOLOR [11]. A two-stage detector produces a set of region proposals of the image in the first stage; then, it passes these regions to the second stage for object classification. On the other hand, a one-stage detector takes the entire image and feeds it to a CNN within a single pass to generate the region proposals and identify and classify all objects in the image. These models have obtained highly accurate results in less computational time and have generally outperformed traditional machine learning algorithms [1]. This is due to the availability and capabilities of high-performance graphical processing units (GPUs) in modern computers, as well as the availability of large benchmark datasets for training models [1]. Object detection has been incorporated into diverse domains, from personal security to productivity in the workplace, including autonomous driving, smart video surveillance, face detection, and several people-counting applications.

Moreover, object detection methods can be employed to assist people with special needs in understanding the content of images, including those with visual disabilities. Moreover, a study conducted by Bourne et al. [12] estimated that, worldwide, there will be 38.5 million completely blind people by 2020, a number that will increase to 115 million by 2050. In 2017, an analysis by the General Authority of Statistics in Saudi Arabia [13] revealed that 7.1% of the Saudi population has disabilities, while 4% of them suffer from some form of visual disability. However, most object detection research dedicated to the visually impaired has been focused on the English language, while a limited number of related studies exist for Arabic. Therefore, Arabic language object detection has emerged as a promising research topic. In one study proposed by Al-muzaini et al. [14], an Arabic image dataset that consisted of images and captions was constructed by means of crowdsourcing, a professional translator, and Google Translate. This dataset was primarily built for testing different image captioning models. Another study [15] proposed an Arabic image captioning model using root-word-based RNN and CNN for describing the contents of images. However, as far as we are aware, the possibility of helping Arabic-speaking visually impaired people through the development of an Arabic text-to-speech engine that can utter objects' names and their positions in images has not been previously investigated.

In this paper, an object detection model based on the Mask R-CNN algorithm for identifying and locating objects in images was developed. Furthermore, a text-to-speech engine was incorporated into the proposed model to convert the predicted objects' names and their positions within the image into speech to help Arabic-speaking people with vision impairment recognize the objects in images. The contribution of our study to the existing literature is the incorporation of a translator and text-to-speech engines into the model to translate and utter the names and positions of the detected objects in images in Arabic. We believe that this is the first approach that utilizes these services with an object detection model to assist Arabic-speaking visually impaired people.

2. Related Work

Leaf disease detection for various fruits and vegetables is an agricultural application of object detection. One such application based on Faster R-CNN was proposed by Zhou et al. [16] and was used to detect rapid rice disease, which appears on the rice leaves. To manage the large number of bounding boxes produced by Faster R-CNN, the authors integrated R-CNN with the Fuzzy C-means (FCM) and K-means (KM) clustering algorithms to reset the size of the bounding boxes. The performance of the designed algorithm was verified against datasets of different rice diseases: rice blast, bacterial blight, and sheath blight. The results showed that FCM-KM-based Faster R-CNN performed better than the original Faster R-CNN. Moreover, the proposed algorithm successfully eliminated most of the background interference, including rice ears and disease-free rice leaves. However, it could not eliminate rice ears completely; therefore, further investigation was needed to apply threshold segmentation theory in the segmentation of plant leaf diseases to make the diseased rice leaves completely segmented.

Similarly, Lakshmi et al. [17] proposed an effective deep learning model called DPD-DS for the automatic detection of diseases in plants and segmentation, which is based on an improved pixelwise Mask R-CNN. The authors introduced a light head region convolution neural network (R-CNN) by adjusting the proportions of the anchor in the RPN network and modifying the backbone structure to save memory space and computational cost, thus increasing detection accuracy and computational speed. The proposed model comprised three parts: first, the classification of the plant's health, either healthy or diseased; second, the detection of the infected portion of the plant by using a bounding box; finally, generating a mask for each infected region of the plant. The authors used two benchmark datasets to train the proposed model, which contained leaf images of crop species, namely the PlantVillage dataset and the Database of Leaf Images dataset. Furthermore, the model was tested and achieved a 0.7810 mAP, outperforming other state-of-the-art models, including SSD, Faster R-CNN, YOLOv2, and YOLOv3. However, the authors highlighted some of the proposed model's limitations, which could be addressed in the future, such as the fact that chinar-infected regions were not segmented properly on some images and that successful detection depends on exact ground-truth data (annotations).

Additionally, another agricultural application of object detection is fruit detection. Fruit detection and segmentation using the Mask R-CNN can be used by modern fruit-picking robots to effectively pick fruit without damaging the plant. For example, Liu et al. [18] utilized and improved the Mask R-CNN to detect cucumber fruit by integrating a logical green (LG) operator into the region proposal network (RPN). This is because the region of interest (RoI) only includes green regions since the algorithm's purpose is to detect cucumber fruit alone. The backbone of the network consisted of the Resnet-101 architecture, which was further consolidated using feature pyramid network (FPN) to extract features of different sizes from the input image. The performance of the developed model was tested and achieved 90.68% precision and 88.29% recall, exceeding the YOLO, the original Mask R-CNN, and Faster R-CNN algorithms in terms of fruit detection. However, due to the Faster R-CNN structure, the developed model's average elapsed time was slightly slow and could not perform real-time detection.

Another study conducted by Zhan et al. [19] proposed an improved version of Mask R-CNN to detect and segment the damaged areas of an automobile in an accident. This can provide speedy solutions for insurance companies to process damage claims. To cope with the challenges in the detection of damaged areas of vehicles, the authors proposed some modifications in Mask R-CNN, such as lowering the number of layers in the original ResNet-101, providing better regularization to solve overfitting and adjusting anchor box dimensions according to the requirements. The performance of the proposed model was compared with the original Mask R-CNN with promising results, where it attained an AP with a value of 0.83 as compared to the original Mask R-CNN's 0.75. However, the authors addressed the limitations of their approach, including situations where, in some cases, the segmentation was not completely accurate and the difficulty in segmenting some areas in which damage was not visible enough.

3. Proposed Method

The framework of the proposed model was developed through several steps: data acquisition, data pre-processing, building the object detection model, specifying the positions of the detected objects in the image, generating the Arabic annotations and positions, text-to-speech (TTS) conversion, and finally, the final model evaluation. Figure 1 shows the general framework of the proposed model.

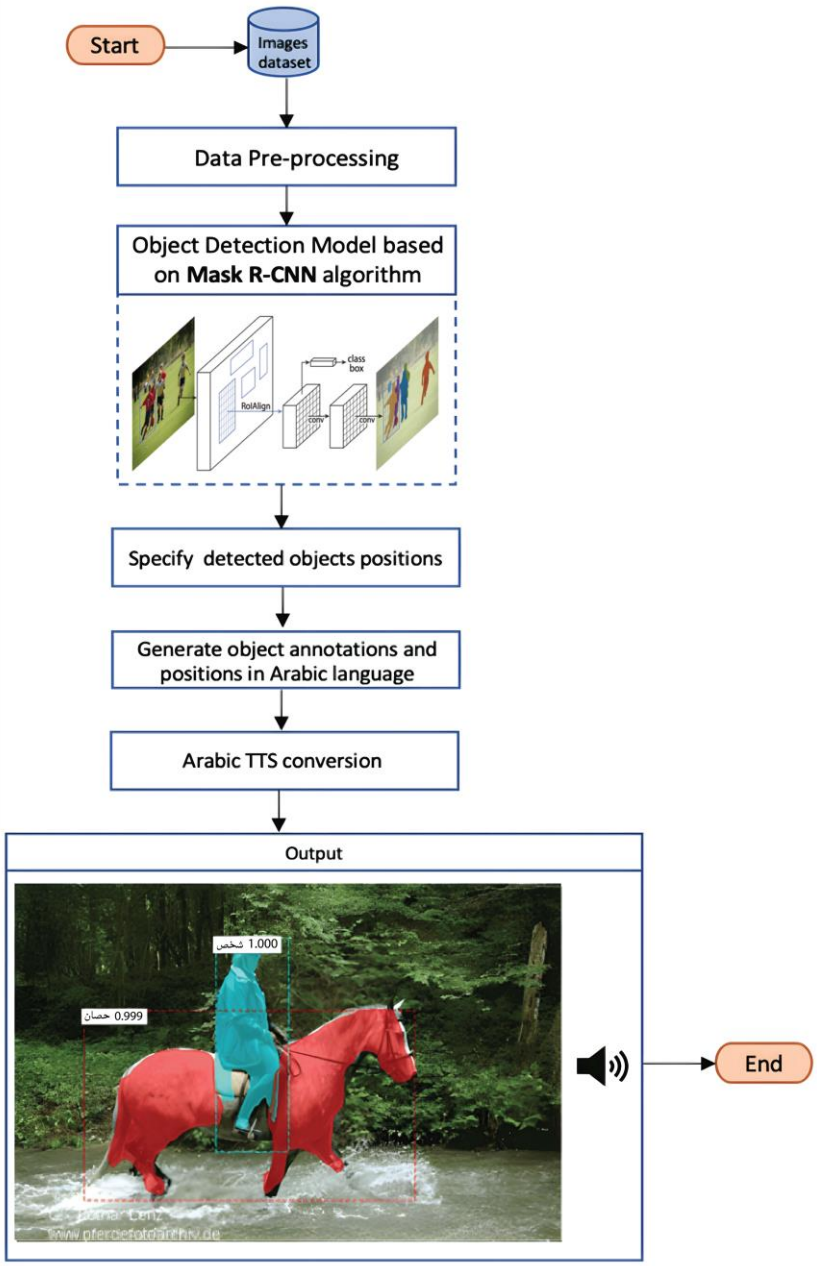


Figure 1. The proposed model framework, (شخص means person, حصان means horse).

3.1. Mask R-CNN Algorithm

Mask R-CNN [5] is an object detection and segmentation algorithm. It is built on top of the Faster R-CNN algorithm [6], which is an object detection algorithm that uses a region proposal network (RPN) with the CNN model. The essential difference between them is that Mask R-CNN operates one step ahead of the object detection mechanism provided by Faster R-CNN and provides pixel-level segmentation of the object, i.e., it not only detects given objects in an image, but also decides which pixel belongs to which object. In this study, Mask R-CNN was employed because of its flexibility, simplicity, and ability to provide flexible architecture designs. Figure 2 shows the Mask R-CNN architecture.

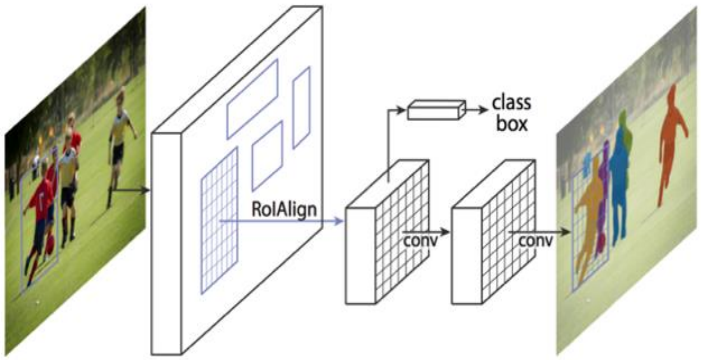


Figure 2. The Mask R-CNN architecture [6].

Mask R-CNN generally consists of the following modules:

- The backbone: This is a standard CNN, which is responsible for feature extraction. The initial layers of the network detect low-level features such as edges and corners, while subsequent layers detect high-level features such as car or person. The backbone network converts the input image into a feature map and passes it to the next module.
- Region proposal network (RPN): This is a lightweight neural network that uses a sliding window mechanism to scan the feature map and extracts regions that contain objects, known as regions of interest (ROIs). ROIs are fed to the RoIAlign method to generate fixed-sized ROIs and precisely locate the regions. The RPN produces two outputs for each ROI: the class (i.e., if the region is background or foreground) and the bounding box refinement.
- ROI classifier and bounding box regressor: ROIs are passed through two different processes, the ROI classifier and the bounding box regressor. The ROI classifier is responsible for classifying the object in the ROI to a specific class, such as person or chair, while the bounding box regressor is responsible for predicting the bounding box for the ROI.
- Segmentation masks: The mask branch is a fully convolutional network (FCN), which is mainly responsible for assigning object masks to each detected object, i.e., the positive regions selected by the ROI classifier.
- Mask R-CNN has several advantages, especially in terms of training simplicity. First, it enhances Faster R-CNN by increasing its speed while adding only minor overhead. Furthermore, the mask branch allows for faster systems and more rapid testing with only minor computational overhead. Moreover, Mask R-CNN can be easily adapted to various tasks. Human pose estimation, for instance, can be easily accomplished with Mask R-CNN using the same framework. Such techniques are key to assisting people with visual impairment in understanding image content. Mask R-CNN has also been widely used in different domains to produce segmented masks; possible uses in the computer vision field are video surveillance, autonomous car systems, and tumor detection. On the other hand, a prevalent issue in Mask R-CNN is that it ignores part of the background and marks it as foreground, resulting in target segmentation inaccuracy.

3.2. Specifying Positions of the Detected Objects

In this paper, we propose a new method for specifying the detected objects' positions regarding the image. This method determines the position of a predicted object by utilizing its bounding box, where each bounding box is defined in terms of its bottom-left and top-right coordinates in the image. These coordinates are retrieved to identify the center of the detected object. Using this defined center and based on the height and width of the image, the model specifies the nine possible positions of the predicted object, as shown in Figure 3.

Top left side	Top of center	Top right side
Mid left side	Mid of center	mid right side
Bottom left side	Bottom of center	Bottom right side

Figure 3. Map of the possible detected objects' positions regarding the image.

3.3. Generating the Annotations and Positions in the Arabic Language

Since there is no study in the literature that proposes object detection with Arabic annotations, this paper fills a gap by incorporating a translator API, which will translate the English labels of the detected objects and their positions relating to the image into the Arabic language. The Google Translation API (Googletrans) was used to translate the class labels for the predicted objects and the positions of the detected objects into Arabic. The Googletrans library was chosen due to its speed and reliability [20]. This library uses the Google Translate Ajax API, which is responsible for detecting the language of the text and translating it into Arabic. However, since the Matplotlib library does not support the Arabic language, it displays the names of the objects to the user with the Arabic text written from left to right, whereas Arabic writing usually flows from right to left. Moreover, it renders the Arabic characters in an isolated form; thus, every character is rendered, regardless of its surroundings, as in Figure 4.

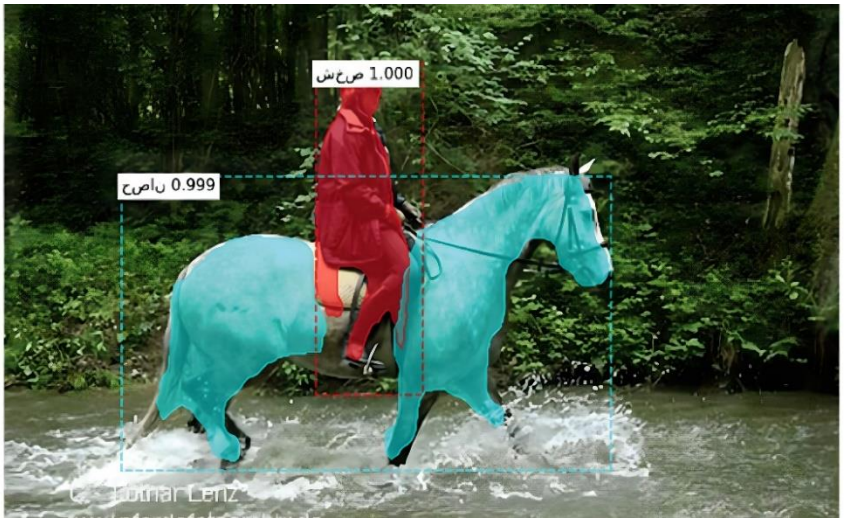


Figure 4. The problem of rendering Arabic characters in isolated forms, شخص means person, حصان means horse).

To solve these issues, two main libraries were imported; a bidi algorithm, which enables the text to be written from right to left, and an Arabic reshaper, which reshapes the Arabic characters and replaces them with their correct shapes according to their surroundings.

3.4. Text-to-Speech Conversion

Text-to-speech (TTS) has emerged as a promising new area of research that bridges the gap between humans and machines. In this study, TTS is incorporated as a later step, where the names of the detected objects and their positions are uttered. As promising as it sounds, this process is not straightforward since every language has diverse and unique features including punctuation and accent. Existing TTS engines vary in terms of their

performance, clarity, naturalness, punctuation consideration, and flexibility, such as Google text-to-speech (gTTS) API [21], Google Cloud TTS API [21], and AI Mawdoo3 TTS API [22]. For our model, we used the gTTS API, a widely adopted TTS engine developed by Google. One of the remarkable facts about the gTTS API is its ease of use, where text input is easily transformed into spoken words, in the form of an MP3 audio file. Additionally, it supports several languages including Arabic and offers two audio speeds for speech, slow and fast.

4. Materials and Methods

4.1. Experimental Environment

Google Colaboratory (Colab) [23] was selected as the platform for developing the proposed system. It is a free cloud service introduced by Google to promote machine learning and artificial intelligence research by providing hosted CPUs and GPUs. One of the main features of Colab is that the essential Python libraries such as TensorFlow, Scikit-Learn, and Matplotlib are already installed and can be imported when needed. Colab was built on a Jupyter Notebook [24], which was used in our study.

4.2. Dataset Acquisition

The Pascal Visual Object Classes (Pascal VOC) dataset [25] was used in this study. Pascal VOC is widely used in object detection as it encapsulates a large number of standardized images. The dataset was first introduced in the PASCAL Visual Object Classes challenge, which took place between 2005 and 2012. This dataset's primary goal is to capture realistic scenes that are full of objects to be recognized. Each image is associated with an annotation file in XML format that specifies the label, as well as the x and y coordinates of the bounding box for each object contained within the image, which makes this dataset suitable for supervised learning problems. This dataset has two variations, Pascal VOC 2007 and Pascal VOC 2012, both of which identify objects as belonging to one of 20 classes, including person, animal, or other objects such as sofa.

The Pascal VOC 2007 dataset includes a total of 9936 images that contain 24,640 annotated objects, while the Pascal VOC 2012 dataset has a total of 11,530 images with 27,450 annotated objects. In our study, each dataset was split into two halves, the first of which was used for training the model and the second for testing the model's performance. The designed model was trained and validated on the Pascal VOC 2007 and 2012 training datasets and tested on the Pascal VOC 2007 testing dataset. The Pascal VOC 2007 testing dataset was used since the VOC 2012 annotation files of the testing dataset are not publicly available.

Additionally, image resizing was applied to the dataset as an additional data preprocessing technique; each image was resized to 1024×1024 pixels and padded with zeros to make it a square, allowing several images to be batched together. The image resizing was applied to reduce the training and inference time of the proposed model. Moreover, the training dataset was expanded by using the image data augmentation technique, which creates modified versions of images in the dataset to improve the performance and ability of the model to generalize. There are various types of image augmentation, such as perspective transformations, flipping, affine transformations, Gaussian noise, contrast changes, hue/saturation changes, and blurring [26]. However, a horizontal flipping augmentation technique was applied to the training dataset so that 50% of all images were flipped horizontally, as in Figure 5. The horizontal flipping was selected because it is suitable to the nature of the classes in the dataset where the model would be able to recognize the object regardless of its left–right orientation.

(**a**) (**b**)

Figure 5. (**a**) Before and (**b**) after the application of the horizontal flipping augmentation technique.

4.3. Transfer Learning

Usually, a vast amount of labeled image data are needed to train a neural network from scratch to detect all the features in images, starting with the most generic before detecting more specific elements. Transfer learning is the process of gaining the advantage of a pre-trained model, and load weights that have been trained on a vast labeled dataset then use comparatively less training data to customize the model. Therefore, transfer learning was implemented in this work using the Mask R-CNN pre-trained on the MS COCO dataset [27] to reduce the training time. First, the Mask R-CNN model was created in the training mode, and then, the MS COCO weights were loaded into the model. However, all the backbone layers were frozen, and all the output layers of the loaded model were removed, including the output layers for the classification label, bounding boxes, and masks. Thus, new output layers were defined and trained on the Pascal VOC dataset, where the pre-trained weights from MS COCO were not used.

4.4. Parameter Settings

The designed object detection model is based on the Mask R-CNN algorithm, which uses Resnet101 as its backbone network for feature extraction. Resnet101 is a CNN that is 101 layers deep, and it is pretrained on more than a million images from the ImageNet database, which is a well-known benchmark dataset for visual recognition models research [28].

The first step when designing the model was to define its configurations. These configurations determine the hyperparameters, such as the learning rate for training the model. The process of tuning the appropriate hyperparameters is crucial when defining a deep learning model since they impact the performance of the model under training. Table 1 shows some of the designed object detection model's parameters, as well as the hyperparameters that were tuned during the design phase of the object detection model.

Table 1. Some of the designed object detection model's parameters.

Parameter	Value
LEARNING_RATE	0.001
LEARNING_MOMENTUM	0.9
DETECTION_MIN_CONFIDENCE	0.7
NUM_CLASSES	21
VALIDATION_STEPS	50
STEPS_PER_EPOCH	1405

4.5. Evaluation Index

In order to measure the performance of the designed object detection model, the average precision (AP) and mean average precision (mAP) evaluation metrics were used:

- Average precision (AP):

The AP was computed for each class by using the 11-point interpolation method, where the precision values are interpolated against 11 equally spaced recall values. The interpolated precision is the highest precision against the recall value larger than the present recall value. This is given by the formula below:

$$AP = \frac{1}{11} \sum_{r \in R} p\,(r) \tag{1}$$

where R indicates the 11 equally spaced recall values, and p represents the interpolated precision.

- Mean average precision (mAP):

As mentioned above, the calculation of AP comprises only one class. However, in object detection, there are usually $N > 1$ classes. The mAP is defined as the mean of the AP across all N classes, as follows:

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{2}$$

5. Experimental Results and Analysis

The experiments with the developed object detection and segmentation model were conducted in two stages. The first of these consisted of experiments designed to facilitate the creation of the object detection and segmentation model based on the Mask R-CNN algorithm using the Pascal VOC 2007 and 2012 datasets and evaluate its performance using the Pascal VOC 2007 testing dataset. In the second stage, the experiments were conducted to evaluate the accuracy of the designed method that is responsible for specifying objects' positions within an image, as well as to evaluate the accuracy of the translator and TTS services.

5.1. The Object Detection and Segmentation Model Experiments

Following the tuning of the hyperparameters, a learning rate of 0.001 was selected where the model gave the best performance. Furthermore, when the learning rate was set to a high value, such as 0.2, the weights might be exploded [29]. Furthermore, when a learning rate of 0.0001 was tried, the performance of the model on the testing dataset deteriorated. With regard to the number of epochs, 80 was selected, where the training loss began to increase.

5.1.1. Training and Evaluating the Model

The Tensorboard [30] was employed to visualize the losses after each epoch. This was intended to facilitate the monitoring of the training process and the subsequent modification of the hyperparameters. Figure 6 presents the performance of the developed Mask R-CNN model during the training process on the Pascal VOC 2007 and 2012 training sets. It shows the curve of the general training loss in relation to the number of epochs. This general training loss comprises the sum of five loss values, as calculated automatically for each of the RoIs. The five losses are as follows:

- rpn_class_loss: the extent to which the region proposal network (RPN) succeeds in separating the background with objects.
- rpn_bbox_loss: the degree to which the RPN succeeds in localizing objects.
- mrcnn_bbox_loss: the degree to which the Mask RPN succeeds in localizing objects.
- mrcnn_class_loss: the extent to which Mask R-CNN succeeds in identifying the class for each object.

- mrcnn_mask_loss: the extent to which Mask R-CNN succeeds in attempting to segment objects.
- As is evident in Figure 6, the general training loss decreased dramatically for the first 50 epochs, after which it changed very slowly. Each epoch took 14 min approximately; thus, the overall training process took 17 h, 30 min, and 28 s to complete 80 epochs, resulting in a final training loss of 0.2402.

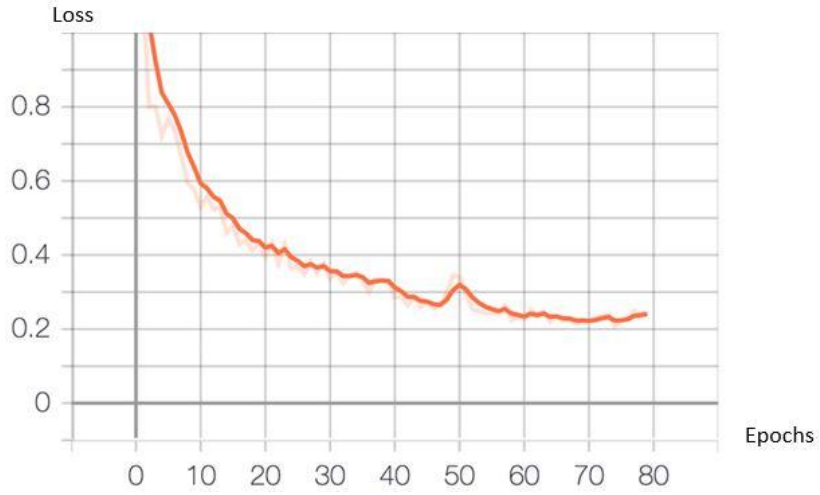


Figure 6. The general training loss of the Mask R-CNN model per epoch.

As a part of the parameter tuning for the model, multiple epochs were used to train the model and produced different weights, which were used to compute the mAP on the validation set to select the model weights that achieved the highest mAP.

Further experiments were performed to achieve the learning rate that produced the model's best performance. The results are illustrated in Figure 7, which compares the performance in different epochs for learning rate values of 0.001 and 0.0001. As Figure 7 shows, the model performed better with a learning rate of 0.001, with an mAP of 83% achieved in the 80th epoch.

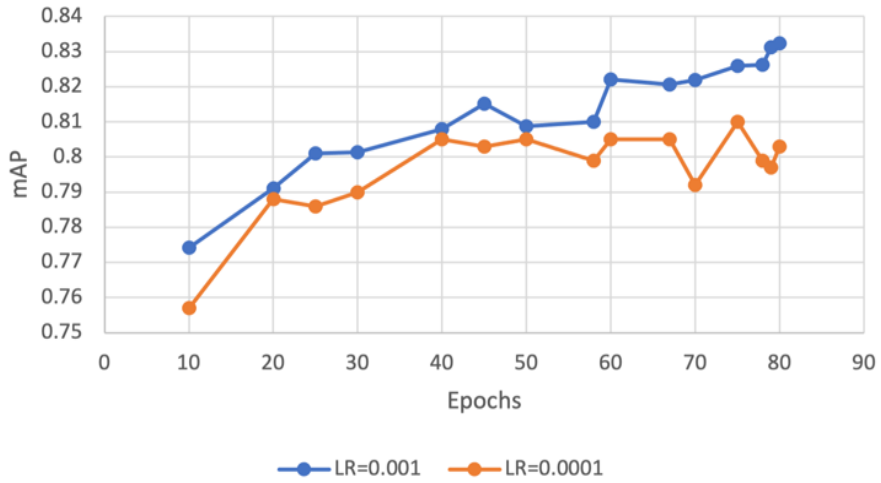


Figure 7. The mAP of the Mask R-CNN model at different epochs.

5.1.2. Testing the Model

As a result of the training process, the model was selected in the 80th epoch. The performance of the designed model was evaluated on the testing dataset using the AP and mAP metrics. The AP was measured separately for each class in the dataset, and the results are shown in Figure 8. Consequently, the mean average precision was calculated, showing that the model achieved good performance with an 83.9% mAP.

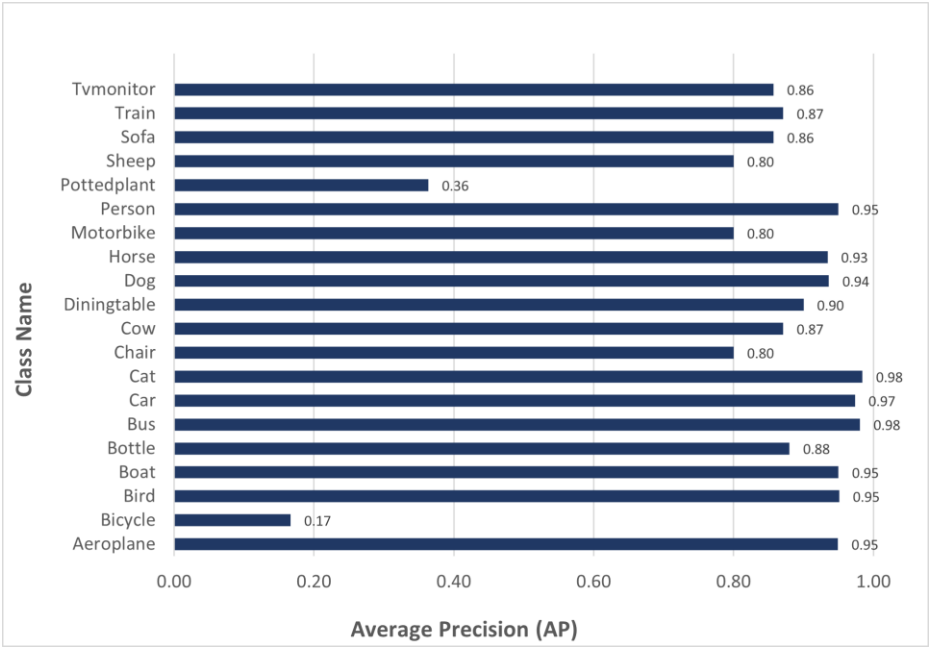


Figure 8. Average precision (AP) per class.

5.1.3. Ablation Experiments

To analyze the effectiveness and contributions of different techniques used in the proposed method, we conducted an ablation experiment on the proposed model as listed in Table 2. We trained three Mask R-CNN variants on the Pascal VOC 2007 + 12 training set and present the evaluation results for the Pascal VOC 2007 testing set. In the first model, we reduced the number of layers in the Resnet backbone to 50 to examine the effect on the performance. In the second model, we removed the fourth convolutional layer in the computation graph of the mask head of the feature pyramid network (FPN). The FPN was used for feature extraction; it takes a single-scale image of any size as an input and produces proportionally sized feature maps with several levels in a fully convolutional form, referred to as CN-Mask R-CNN. Finally, in the third model, we removed the second conventional layer in the convolution block in the Resnet graph, which is referred to as CC-Mask R-CNN. Then, the object detection performance of the trained models was compared with our proposed model.

Table 2. Ablation experiments for the proposed model.

Method	Backbone	mAP	Aero	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	TV
Mask R-CNN	Resnet-50	68.9%	0.69	0.79	0.82	0.67	0.59	0.65	0.75	0.89	0.65	0.76	0.65	0.73	0.78	0.59	0.57	0.66	0.65	0.65	0.62	0.62
CN-Mask	Resnet101	82.2%	0.94	0.34	0.96	0.92	0.88	0.8	0.97	0.98	0.9	0.87	0.9	0.93	0.94	0.4	0.62	0.45	0.88	0.92	0.97	0.87
CC-Mask	Resnet101	80.7%	0.92	0.11	0.87	0.85	0.8	0.98	0.97	0.97	0.6	0.77	0.98	0.83	0.88	0.99	0.98	0.34	0.76	0.82	0.94	0.73
Proposed Model	Resnet101	83.9%	0.94	0.16	0.95	0.95	0.88	0.98	0.97	0.98	0.8	0.87	0.9	0.93	0.93	0.8	0.95	0.36	0.8	0.85	0.87	0.85

By analyzing the values of the mAP for the proposed model in this paper, we found that removing the fourth convolutional layer in the mask head affected the detection accuracy, leading to a 1.7% reduction in the mAP. It is worth noting that, after the backbone layers were changed from 50 to 101, the detection accuracy of the proposed method in this paper gradually improved by 15%.

5.1.4. Comparison with Previous State-of-the-Art Object Detection Models

A comparison between the performance of the developed Mask R-CNN model and previous object detection models was conducted, including Faster R-CNN [5], SSD300 (the input image size was 300 × 300) [9], SSD512 (the input image size was 512 × 512) [9], and YOLO v2 [31], and more recent models, including EEEA-Net-C2 [32], HSD [33] (the

input image size was 320 × 320) and Localize [34], as illustrated in Table 3. Faster R-CNN, SSD300, and SSD512 were first trained on MS COCO and subsequently fine-tuned on the Pascal VOC 2007 and 2012 datasets. Other models were trained on the Pascal VOC 2007 and 2012 datasets. As Table 3 clarifies, the developed Mask R-CNN outperformed the base variant (Faster R-CNN) and the remaining state-of-the-art models on the Pascal VOC 2007 testing dataset.

Table 3. Comparative results on the Pascal VOC 2007 testing dataset.

Detection Models	Trained on	mAP
Faster R-CNN [5]	07 + 12 + COCO	78.8%
SSD 300 [9]	07 + 12 + COCO	79.6%
SSD 512 [9]	07 + 12 + COCO	81.6%
YOLOv2 [31]	07 + 12	78.6%
EEEA-Net-C2 [32]	07 + 12	81.8%
HSD [33]	07 + 12	81.7%
Localize [34]	07 + 12	81.5%
The proposed model	07 + 12	83.9%

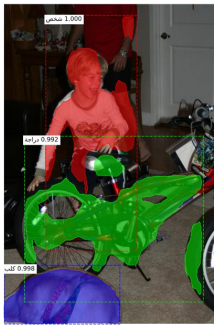
5.2. The Arabic Annotations, Objects Positions, and TTS Experiments

To evaluate the proposed method for specifying the objects' positions within an image, as well as the translator and TTS services, several experiments were conducted based on the evaluation method used in [35] with a sample of 100 images containing 340 objects that were chosen randomly from the testing dataset. The performance evaluation was completed by a human judge fluent in both Arabic and English. The judge examined four outputs for each object in each image passed to the model and was asked to complete an online datasheet with the results; those outputs were as follows:

(1) The accuracy of the predicted object's position within an image, where the judge was asked to use the map illustrated in Figure 3 as a reference and indicate if the detected object's position was in the expected position within the map.
(2) The accuracy of the translated label of the predicted object, where the judge was responsible for indicating whether the translated Arabic label was "accurate" or "inaccurate".
(3) The accuracy of the translated position of the predicted object, where the judge was also responsible for indicating if the translated Arabic position was "accurate" or "inaccurate".
(4) The Arabic speech of the predicted object label with its position, where the judge was responsible for deciding whether the spoken object's name and its position were "clear" or "unclear".

The results of the experiments were calculated for each output. The results of the first output, which relates to the accuracy of the proposed method, were high: accurate positions were detected for 300 out of a total of 340 objects. Regarding the translation service for the Arabic labels and positions, the results were accurate for all the detected objects' labels and their positions. Finally, all the Arabic speech in the TTS service was categorized as "clear". Table 4 presents two samples of the resulting images with their correct object detections and translations. The fourth column in Table 4 presents the names and positions of the detected objects for Samples A and B in Arabic. For instance, the result of Sample A is: "person in the top of the right side, person in the mid of the left side, chair in the mid of the left side, dining table in the bottom of the center". The result of Sample B is: "person in the mid of the center, dog in the bottom of the left side, bicycle in the bottom of the center".

Table 4. Samples of detection results. (شخص means person, كرسي means chair, طاولة الطعام means dining table, دراجة means bicycle).

Sample	Original Image	Image with the Detected Objects with Arabic Annotations	Names and Positions of Detected Objects
A			شخص في الجزء العلوي من الجانب الأيمن ، شخص في منتصف الجانب الأيسر ، كرسي في منتصف الجانب الأيسر ، طاولة الطعام في الجزء السفلي من المركز
B			شخص في منتصف المركز ، الكلب في الجزء السفلي من الجانب الأيسر ، دراجة في الجزء السفلي من مركز

5.3. Discussion of Results

This section comprises overall remarks regarding the evaluation of the generated model. Based on the analysis of our experimental results, we came to the following conclusions. Firstly, the training time is defined not only by the quantity of data involved, but also by the complexity of the architecture of a deep learning model. As Mask R-CNN is a large model, the training time was lengthy, with a completion time for each epoch of approximately 14 min. Secondly, the Mask R-CNN model proved its ability to provide highly accurate results when trained on the Pascal VOC 2007 and 2012 datasets, as compared with previous object detection models that employed Faster R-CNN and SSD300. Hence, we contend that our model could be beneficial for supporting Arabic-speaking visually impaired people with the recognition of objects in images.

6. Conclusions

To conclude, there has been a dramatic interest in using a deep learning approach for object detection in recent years, while various state-of-the-art models have been proposed to enhance detection performance in terms of accuracy and speed. However, rising numbers of Arabic-speaking visually impaired people mean that an Arabic language object detection model is needed. Therefore, we successfully developed and tested a prototype system that could help Arabic-speaking individuals with impaired vision recognize objects in digital images.

The proposed model is based on the Mask R-CNN algorithm to classify and locate different objects in an image. The model was trained and tested using the Pascal VOC dataset. In addition, extensive experiments were performed to measure its effectiveness by reducing the number of backbone layers in the mask R-CNN to 50 and removing different layers, including the fourth convolutional layer in the computation graph of the mask head of the FPN and the second conventional layer in the convolution block in the Resnet graph. The results obtained were detailed and compared with the accuracy of the proposed model.

The performance of the developed object detection model was evaluated and compared with other state-of-the-art object detection models using the Pascal VOC 2007 dataset, and the results revealed that the proposed Mask R-CNN achieved the highest accuracy, at

83.9%. Moreover, a new method for specifying the position of detected objects within an image was introduced, and an Arabic-speaking text-to-speech engine was incorporated to utter the detected objects' names and their positions within an image for the visually impaired. We believe that this is the first approach to incorporate these services with an object detection model to assist the Arabic-speaking visually impaired.

Despite its efficacy and reliability, the proposed model has several limitations that indicate potential directions for future research. First, the proposed model lacks Arabic language, real-time video stream detection, which is vital to developing the means for Arabic-speaking visually impaired people to recognize their surroundings. Second, another noticeable limitation is found with certain object classes: although the model performs well at detecting different objects within images, some classes such as *bicycle* are hard to segment, given their complex shapes and curved design, as shown in Table 4 Sample B. Third, the model lacks interactivity; to further enhance the user experience, an Arabic language application should be developed that enables users to interact with the system.

The complexity of certain shapes such as *bicycles*, which are characterized by their varying shapes and curved designs, were proven to have a low average precision rate of identification within our model. Therefore, as part of its future development, the model should be trained on more examples of such images to enhance its detection performance. Moreover, the proposed object detection model is intended to be deployed as a mobile-based application, designed to not only detect objects in images, but also within videos in real-time. Moreover, it will be designed to predict the Arabic caption of an image that describes its contents for Arabic-speaking visually impaired people. The Web Content Accessibility Guidelines 2.1 (WCAG 2.1) produced by the World Wide Web Consortium (W3C) [36] will be followed to ensure the accessibility of the application.

Author Contributions: Conceptualization, H.H.A.-B.; Methodology, H.H.A.-B.; Software, N.A.; Validation, N.A. and H.H.A.-B.; Formal analysis, N.A.; Investigation, N.A.; Resources, N.A.; Writing—original draft, N.A.; Writing—review & editing, H.H.A.-B.; Supervision, H.H.A.-B.; Project administration, H.H.A.-B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The two analyzed datasets during the current study are publicly published. The Pascal VOC 2007 and The Pascal VOC 2012 datasets are available in the PASCAL visual object classes challenge 2007 and 2012 repository, http://host.robots.ox.ac.uk/pascal/VOC/index.html (accessed on 15 July 2022).

Acknowledgments: The authors would like to acknowledge the Researchers Supporting Project Number RSP-2021/287, King Saud University, Riyadh, Saudi Arabia, for their support in this work.

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

References

1. Zhao, Z.; Zheng, P.; Xu, S.; Wu, X. Object Detection With Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]
2. Pathak, A.R.; Pandey, M.; Rautaray, S. Application of Deep Learning for Object Detection. *Procedia Comput. Sci.* **2018**, *132*, 1706–1717. [CrossRef]
3. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
4. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
5. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]

6. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988. [CrossRef]
7. Wang, J.; Hu, X. Convolutional Neural Networks with Gated Recurrent Connections. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3421–3436. [CrossRef] [PubMed]
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
9. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision-ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
10. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
11. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. You Only Learn One Representation: Unified Network for Multiple Tasks. *arXiv* **2021**, arXiv:2105.04206.
12. Bourne, R.R.A.; Flaxman, S.R.; Braithwaite, T. Magnitude, Temporal Trends, and Projections of the Global Prevalence of Blindness and Distance and near Vision Impairment: A Systematic Review and Meta-Analysis. *Lancet Glob. Health* **2017**, *5*, e888–e897. [CrossRef]
13. Zeried, F.M.; Alshalan, F.A.; Simmons, D.; Osuagwu, U.L. Visual Impairment among Adults in Saudi Arabia. *Clin. Exp. Optom.* **2020**, *103*, 858–864. [CrossRef] [PubMed]
14. Al-muzaini, H.A.; Al-yahya, T.N.; Benhidour, H. Automatic Arabic Image Captioning Using RNN-LSTM-Based Language Model and CNN. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 67–73. [CrossRef]
15. Jindal, V. Generating Image Captions in Arabic Using Root-Word Based Recurrent Neural Networks and Deep Neural Networks. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, New Orleans, LA, USA, 2–4 June 2018; pp. 144–151. [CrossRef]
16. Zhou, G.; Zhang, W.; Chen, A.; He, M.; Ma, X. Rapid Detection of Rice Disease Based on FCM-KM and Faster R-CNN Fusion. *IEEE Access* **2019**, *7*, 143190–143206. [CrossRef]
17. Kavitha Lakshmi, R.; Savarimuthu, N. DPD-DS for Plant Disease Detection Based on Instance Segmentation. *J. Ambient. Intell. Humaniz. Comput.* **2021**. [CrossRef]
18. Liu, X.; Zhao, D.; Jia, W.; Ji, W.; Ruan, C.; Sun, Y. Cucumber Fruits Detection in Greenhouses Based on Instance Segmentation. *IEEE Access* **2019**, *7*, 139635–139642. [CrossRef]
19. Zhang, Q.; Chang, X.; Bian, S.B. Vehicle-Damage-Detection Segmentation Algorithm Based on Improved Mask RCNN. *IEEE Access* **2020**, *8*, 6997–7004. [CrossRef]
20. de Vries, E.; Schoonvelde, M.; Schumacher, G. No Longer Lost in Translation: Evidence That Google Translate Works for Comparative Bag-of-Words Text Applications. *Polit. Anal.* **2018**, *26*, 417–430. [CrossRef]
21. Cambre, J.; Colnago, J.; Maddock, J.; Tsai, J.; Kaye, J. Choice of Voices: A Large-Scale Evaluation of Text-to-Speech Voice Quality for Long-Form Content. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 25–30 April 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1–13.
22. Mawdoo3 AI. Available online: https://ai.mawdoo3.com/ (accessed on 10 October 2022).
23. Google Colaboratory. Available online: https://colab.research.google.com/notebooks/intro.ipynb (accessed on 15 September 2022).
24. Project Jupyter. Available online: https://www.jupyter.org (accessed on 18 October 2022).
25. The PASCAL Visual Object Classes Homepage. Available online: http://host.robots.ox.ac.uk/pascal/VOC/index.html (accessed on 15 July 2022).
26. Shorten, C.; Khoshgoftaar, T.M. A Survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
27. Abdulla, W. Mask R-CNN for Object Detection and Instance Segmentation on Keras and TensorFlow. In *GitHub Repository*; Github: San Francisco, CA, USA, 2017.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
29. Kelleher, J.D.; Namee, B.M.; D'Arcy, A. *Fundamentals of Machine Learning for Predictive Data Analytics, Second Edition: Algorithms, Worked Examples, and Case Studies*; MIT Press: Cambridge, MA, USA, 2020.
30. TensorBoard. TensorFlow. Available online: https://www.tensorflow.org/tensorboard (accessed on 25 July 2022).
31. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [CrossRef]
32. Termritthikun, C.; Jamtsho, Y.; Ieamsaard, J.; Muneesawang, P.; Lee, I. EEEA-Net: An Early Exit Evolutionary Neural Architecture Search. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104397. [CrossRef]

33. Cao, J.; Pang, Y.; Han, J.; Li, X. Hierarchical Shot Detector. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9704–9713. [CrossRef]
34. Zhang, H.; Fromont, E.; Lefevre, S.; Avignon, B. Localize to Classify and Classify to Localize: Mutual Guidance in Object Detection. *arXiv* **2020**, arXiv:2009.14085.
35. Alsudais, A. Image Classification in Arabic: Exploring Direct English to Arabic Translations. *IEEE Access* **2019**, *7*, 122730–122739. [CrossRef]
36. Spina, C. WCAG 2.1 and the Current State of Web Accessibility in Libraries. *Weav. J. Libr. User Exp.* **2019**, *2*. [CrossRef]