DEPTH-BASED REGION PROPOSAL : MULTI-STAGE REAL-TIME OBJECT DETECTION

Shehab Eldeen Ayman[1], Walid Hussein[2], Omar H. Karam[2]

[1]Department of Software Engineering, The British University in Egypt, Cairo, Egypt
[2]Department of Computer Science, The British University in Egypt, Cairo, Egypt

ABSTRACT

Many real-time object recognition systems operate on two-dimensional images, degrading the influence of the involved objects' third-dimensional (i.e., depth) information. The depth information of a captured scene provides a thorough understanding of an object in full-dimensional space. During the last decade, several region proposal techniques have been integrated into object detection. scenes' objects are then localized and classified but only in a two-dimensional space. Such techniques exist under the umbrella of two-dimensional object detection models such as YOLO and SSD. However, these techniques have the issue of being uncertain that an object's boundaries are properly specified in the scene. This paper proposes a unique region proposal and object detection strategy based on retrieving depth information for localization and segmentation of the scenes' objects in a real-time manner. The obtained results on different datasets show superior accuracy in comparison to the commonly implemented techniques with regards to not only detection but also a pixel-by-pixel accurate localization of objects.

KEYWORDS

Real-time object detection, region proposal, computer vision, RGBD object detection, and two-stage object detection.

1. INTRODUCTION

Object detection techniques have made significant strides in reconciling detection accuracy in real-time performance. The score of these techniques was usually measured through the number of correctly identified objects and their associated detection accuracy. Recently, more efficient neural network-based techniques for real-time object detection were developed, such as YOLO and its variants. Moreover, numerous scene optimization approaches were applied to minimize the size of the input frames, as Region Proposal by Guided Anchoring [1], to achieve a reasonably faster detection. Hence, these approaches resulted in the development of a new generation of object detection models that are deployed in a variety of real-time applications such as autonomous driving[2].

Convolutional neural networks (CNN) have been enjoying booming success in object detection applications in recent years due to the accessibility of high-performance computing and large-scale distributed clusters. CNN has replaced many computer vision algorithms to increase accuracy and streamline the computer vision process [3]. The constant pursuit to improve the obtained CNN accuracy and speed has made them viable for deployment in real-life applications.

David C. Wyld et al. (Eds): ICDIPV, CBIoT, ICAIT, WIMO, NC, CRYPIS, ITCSE, NLCA, CAIML -2023
pp. 51-65, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131305

The widely implemented object detection techniques rely on either one-stage or two-stage models. The two-stage models separate the object classification and the object localization processes. Many classification and localization approaches have been utilized to formulate the two-stage models.

2. RELATED WORK

2.1. Real-time object detection

The common approaches are R-CNN, Fast R-CNN, and revised Fast R-CNN. On the other hand, the one-stage models, such as SSD and YOLO, combine both the object classification and the localization processes into a single pipeline[4].

R-CNN was one of the earliest effective models that detected objects in images by using a massive convolutional neural network[3][5]. Fast R-CNN was designed to mitigate some of the drawbacks of R-CNN. Some of the R-CNN drawbacks are the very slow test time detection, the high computation and time expense of training, and the multi-stage pipeline architecture[5]. A Fast R-CNN network accepts an entire image as input along with a collection of object recommendations. An ROI pooling layer is used to identify region proposals that are later fed to a SoftMax layer to predict the class of the bounded region[6]. Faster R-CNN is another update to R-CNN which aims to further improve upon R-CNN and Fast R-CNN in terms of speed [7]. The aim was to move two-stage object detection models towards real-time performance while maintaining high accuracy compared to one-stage models [8][9].

You only look once famously known as “YOLO”which is a novel method for detecting objects. Prior work on object detection has repurposed classifiers for detection purposes. A single neural network predicts bounding boxes and class probabilities directly from entire pictures as shown in figure 1. Because the whole detection pipeline is a single network, its performance can be tuned end-to-end[10].

To speed up object detection, the suggested two-stage approach uses depth information to compress the input frame. This approach will need not just a stream of input frames, but also depth information for each frame. Depth information may be used to decrease the amount of redundant data in a picture, eliminating the necessity for an image detection neural network. Because the bulk of the frame may, in certain situations, comprise features that aren't essential for the application, such as sky, ground, or backdrop buildings. It will also help to eliminate some of the more prevalent issues with traditional object detection approaches, such as overlapping objects, incomplete objects, and objects that are too near or far for detection. Furthermore, the suggested approach may rank objects depending on their closeness. This demonstrates that items that are closer to the camera are given greater priority during the object detection process than ones that are further away. This might have a big effect in certain applications since it's important not just to accurately identify objects, but also to make sure that nearby items are detected quickly. Autonomous driving is an excellent example of this capability. An autonomous driving automobile must precisely and quickly assess oncoming traffic to traverse the roadways effectively. However, the device's accuracy and reliability are critical safety concerns. It's reason for worry if a system correctly detects nine out of ten vehicles approaching a traffic bottleneck but misses the one vehicle that may cause an accident. Even if distant objects are not identified precisely at first, a better degree of safety may be achieved by prioritizing nearer items based on their depth information. As they get closer, their chances of being detected accurately increase.

[11] BlitzNet conducts object detection and semantic segmentation in a single forward pass, enabling calculations in real time. Using image data labeled at the global object level utilizing bounding boxes and at the pixel level utilizing segmentation maps, they hope to efficiently handle both challenges at the same time.

[12] Convolutional neural networks are used for object detection R-CNN CNN [13]and semantic segmentation [14]. A convolutional neural network initially processes the input image to generate a map with high-level features. The resolution of the feature map is then progressively decreased to undertake a multi-scale search of bounding boxes [15]. The feature maps are then up-scaled through deconvolutional layers to predict more accurate segmentation maps. Prediction is accomplished using single convolutional layers, one for detection and one for segmentation, in a single forward pass.

Utilized is a technique with skip connections that combines downscale and upscale feature maps. Using an approach called ResSkip, maps from the downscale and upscale streams are blended. Through bilinear interpolation, incoming feature maps are up-sampled to the size of the appropriate skip link. Both skip connection feature maps and up-sampled maps are then concatenated, passed through a block of 1 X 1 convolution, 3 X 3 convolution, and 1 X 1 convolution, and then added to the up-sampled input via a residual connection.

[16] presents a unified network to combine implicit and explicit information, enabling the learned model to include a generic representation, and provide sub representations suited for specific purposes.

Explicit deep learning may be accomplished primarily using a query, key, or value to elicit self-attention. Non-local networks are an additional method for obtaining attention since they extract attention pairwise in time and space. Implicit neural representations and deep equilibrium models constitute the majority of implicit deep learning approaches [17]. The primary objective of the former is to generate the parameterized continuous mapping representation of discrete inputs for performing various tasks, while the objective of the latter is to turn implicit learning into a residual form neural network and calculate its equilibrium point.

[18] created a strategy for scaling models based on YOLOv4 and offered scaled YOLOv4. After conducting model scaling for the suggested object detector, the following step is to address the changing quantitative elements, such as the number of qualitative parameters. Among these parameters are model inference time and average accuracy. Depending on the equipment or database utilized, the qualitative parameters will have varying gain effects. The study also offers a redesign for YOLOv4 termed YOLOv4-CSP, which scales dependent on the GPU power available to balance the speed against the accuracy tradeoff.

Their method combines the characteristics originating from several feature pyramids and then goes via two sets of Darknet residual layers in reverse without using shortcut connections. In addition, they placed the SPP module amid the first calculation list group of the CSPPAN. This significantly lowered the computational burden compared to YOLOv4

2.2. Semantic Segmentation

The objective of semantic image segmentation is to associate each pixel in an image with the class of the object being represented as shown in figure 4 below. Because values are predicted for each pixel in the image, this activity is referred to as "dense prediction." Notably, instances of the same class are not segregated; the only concern is with the category of each pixel. In other words, if an input picture contains two items belonging to the same category, the segmentation map does

not fundamentally differentiate them as distinct things [25]. The proposed method not only provides separation of pixels based on object localization. It also provides millimeter-accurate ground truth distance information for each pixel.

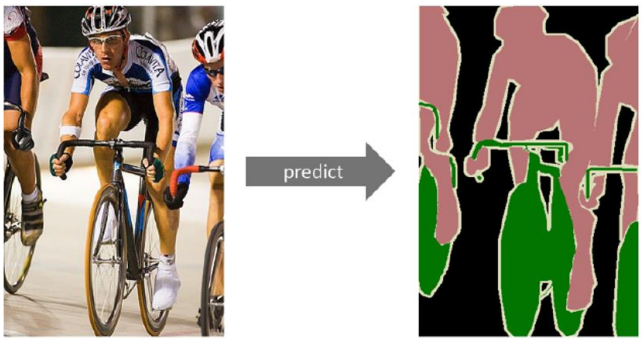

Figure 1. semantic segmentation is applied to determine which pixels in the scene belong to a certain object[19].

3. MATERIAL AND METHODS

3.1. Scene Analysis

Traditional techniques for scene analysis and comprehension use input frames as discrete 2d frames without regard for the scene's real depth. In-depth knowledge may provide a great deal of context to a scenario. Depth may assist in determining the relatively insignificant things in a scene that may consume most of its area (e.g., floor, wall, sky, asphalt). Understanding the value of each thing in a frame may assist in refocusing attention and computing resources on the scene's most significant objects. We need a firm grasp of the size (how near or distant they are from the camera and how large they truly are) and the number of significant things (how many important objects are there in the scene). What the 2D input frames lack is an accurate representation of the location of the items in the scene. Additionally, some items (whether from the same class or other classes) overlap, since real-world situations are often complicated, with several items overlapping. Additionally, differentiating three-dimensional items in a three-dimensional picture using just two-dimensional information extracted from standard two-dimensional photographs might generate inconsistent results. Figure 2 below is a still frame captured from a dashcam shows how most of the scene is occupied by asphalt and sky which are not significant for an autonomous driving model. So, the focus should be shifted towards vehicles, pedestrians

Figure 2. Car dashcam footage of resolution 640 X 480 shows the sky occupying 57% of the image. While the road covers 17% of the image. While the car -indicated in red- covers only 0.59%.

3.2. Object Sorting

Multiple objects of the same or different classes may exist in a single scenario. Not only is it necessary to comprehend the kinds of items in a picture, but also to classify them according to their distance, might be highly critical. Sorting objects in a scene may assist in determining which things are in the forefront and which are in the background. Background objects are practically irrelevant to scene analysis in certain applications. Alternatively, in other circumstances, foreground items should be disregarded since the primary objective is to evaluate items at a great distance.

3.3. Distance-Based Object Prioritization

Accuracy improvements in real-time machine learning-based object identification have enabled a flurry of new applications in a range of sectors. And there seems to be a continual attempt to improve these machine learning systems' accuracy and efficiency. These developments have lifted object detection from a theoretical concept to a set of useful applications (e.g., autonomous driving, security, etc.). However, as no machine learning system has ever reached 100 percent accuracy, there is always potential for improvement. Numerous studies exist that, although theoretically exhaustive, fall short of being exhaustive, and there is also no hardware capable of assessing these methods. While existing machine learning algorithms perform well, many applications need near-perfect accuracy, especially when speed is critical. Rather than pursuing excessive accuracy, which may require very complicated technology that is not suited for the situation, priorities high-priority things above low-priority items. Certain applications may need a greater priority for identifying the furthest visible objects, depending on the conditions. Figure 3 shows thatin other use circumstances, the nearest objects are prioritized even if not all of the components of the items are visible in the scene.

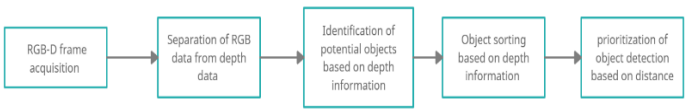

Figure 3. The proposed object detection pipeline

3.4. Region Proposal Elimination

Region proposal techniques vary from one implementation to another and from one-stage to two-stage models. Two-stage models rely on a separate independent process to localize objects. While one-stage models rely on a single stage to localize and classify objects in a given scene. Regardless of the used techniques, relying on acquired depth data would eliminate the need to predict where objects rely in a given scene. This could greatly impact compute performance and accuracy. compute performance could be saved as no computations are required to propose the location and existence of objects in a given scene. It could also resolve scenes that are considered challenging for some region proposal techniques due to scale, lighting,and occlusion.

3.5. Implemented Neural Networks

Different networks were explored and tested to find a suitable one that can fit the pipeline. The main two criteria that were focused on during the evaluation were accuracy and speed. The potential networks had to achieve a good mean average precision value. while at the same time, it has to output a classification in a relatively timely manner. The compromise between speed and

accuracy was needed to achieve real-time or near-real-time performance. Originally, the networks chosen were designed solely for classification. Meaning that they could only classify one object per frame. The model implemented for this project is a two-stage detection model. Meaning that in any given scene, the model may need to detect multiple objects of multiple classes and still be applicable for real-time application. The chosen networks for the object detection implementation are ResNet-50[20]and VGG-16 [21] demonstrated in Table 1 below.

Table 1 The table displaysthe VGG-16 network layout (Karen & Andrew, 2015). The model consists of 13 convoluted layers, 5 pooling layers, and 3 dense layers.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

4. THE SYSTEM ARCHITECTURE

This system is composed of four primary components. The division of the system's pipeline into numerous distinct modules guarantees the system's resilience and flexibility. That is, a module may be tweaked or even removed entirely without impacting the rest of the system. As a result, existing modules may be adapted to new modules with little code changes.

The modules are structured in such a way that each module feeds information to the subsequent one. The components include depth acquisition and analysis, preparation of training data, neural network construction and training, and an object identification system.

4.1. Depth Stream Histogram

After successfully obtaining and displaying a depth stream, the system proceeds to analyze each frame. A histogram of depth analysis was implemented. Figure 2 below shows that the histogram would enable a better object segmentation and understanding of the scene because it represents potential objects as spikes in the represented graph.

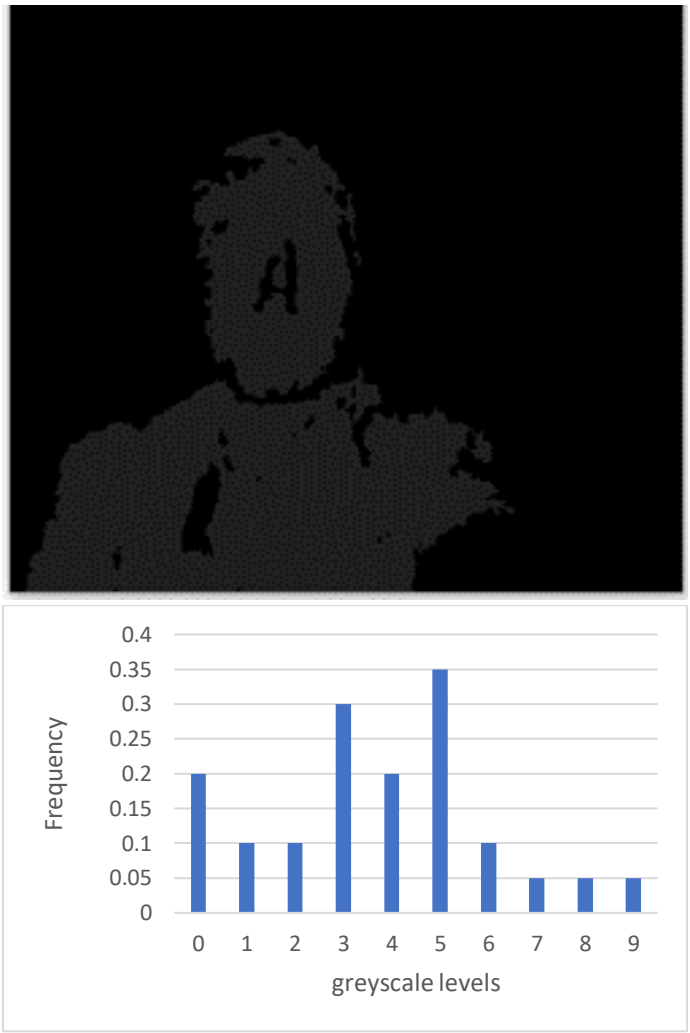

Figure 4. Histogram of oriented gradients. greyscale image intensity is divided into 10 bins where objects are indicated on the graph by a spike in value

In addition, another approach was implemented relying on transferring the depth bits into pixels. This was achieved by mapping depth information to an 8-bit pixel intensity value.

4.2. Neural Network Structure and Training

The neural networks implemented in this project are discussed extensively along with their performance results in the results section. for the code, we relied on the TensorFlow framework for implementing the neural networks. The framework enabled a flexible use of training data and the fast production of results[22]. some of the most crucial features without which training a model would not have been possible are batch normalization and image data generator. Image data generator enables the division of training data into small chunks. Each of these is capable of passing training on its own with no dependency on the other chunks. Another feature of the data generator function is the rescale option. The rescale function enables the fly resizing of images before being packaged into a batch. This enables for a streamlined process to normalize the size of the images to have a value that ranges between zero and one regardless of their original size[23]. All the different neural network models were trained on the same dataset to ensure fair comparison and results between the different neural network models. All of this means that the

training pipeline is the same for every model. Each neural network model is enclosed within a function. So different models can slot in easily without needing to modify any part of the code. Each model is saved by its name along with its associated weights so that they can be used in the detection part.

4.3. All the models ran on an 'Adam' optimizer with a steady learning rate of 0.001. The 'Adam' optimizer is a method for optimizing stochastic objective functions using first-order gradients and adaptive estimates of lower-order moments. The approach is computationally efficient, requires minimal memory, is invariant to gradient diagonal rescaling, and is ideally suited for issues with a large amount of data and/or parameters[24]. Additionally, the approach applies to gradients that are noisy or sparse. Adam performs well in practice and is competitive with other stochastic optimization approaches.

4.4. Detection System

The detection module is the final part of the system. And is designed to output the final decisions by the system where each object in a given scene is classified and given a proper class label. And each object is localized, and bounding boxes are drawn to indicate where the object resides in a given scene.

The system starts with accepting the depth capture input frames from the depth capture module. It then proceeds to grab the color frame capture as a frame of reference to the depth one. The system then proceeds to parse each frame to its correct color space. The depth frame is parsed to an RGB color space. while the color frame is parsed to an RGB color space. the module then calls among the sub-modules that are necessary to output a correct classification, labeling, and localization. The sub-modules are as follows:

4.5. Layer Extraction

The layer extraction submodule proceeds to break down the depth map into sublayers. We use a threshold to break down the depth map into multiple layers. Each layer represents an operating distance. Then, the extracted depth layers are enhanced using some image processing techniques. The goal of image enhancement here is not to sharpen the image but rather blur it. The original depth image comes with a lot of noise. Applying a Gaussian blur with a large kernel would significantly reduce the noise in the depth frame [25]. since depth frames can maintain information even after high compression. Therefore, the Gaussian blur does not negatively affect the detection quality. Figure 7 below shows the difference between blurred and original (non-blurred) images. The blur aids in streamlining edges and contours which aids in identifying potential objects.

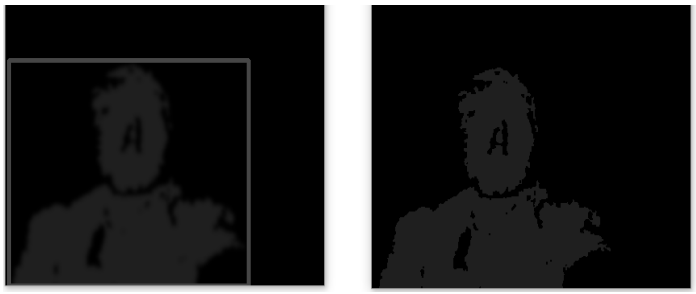

Figure 5. blurred image (left) vs original image (right). Gaussian blur is applied to the frame to iron out all the jagged edges and maintain the uniformity of the object

The next step is to find the contours of the object/objects in the scene. Once contours are found, the positions of these contours are recorded as the boxes surrounding them represent a potential object. The position parameters are $(,y,w,h)$. Were $xand$ that represents the position of the topmost left corner of the object.And w, represent the width and height of the object where the top left corner position(x,y)e bottom right corner would be $(x+w,y+h)$. This box is then taken as a reference and its positions are remapped onto the RGB color frame for detection.

4.6. Object Classification

The last sub-module is the object classification sub-module. This module loads the pre-trained model and uses it for classification. The way the system was designed is to enable the hot swapping of pre-trained neural networks without needing to modify the code. The only requirement is that all the pre-trained neural network models have to be trained on the same class labels so that the output label corresponds correctly to the class of the classified image. We found that the best practice is to train all the different neural network models on the same dataset. This establishes an equal baseline where all the models can be trained. And then, we can determine the accuracy and speed of each model. Remember that accuracy only is not a sufficient metric for the system as the model has to also conform to real-time performance to be applicable for use. Figure 8 below shows a sample of the final operation of the pipeline which is object classification. The object was correctly classified, labeled, and localized with a surrounding bounding box in a challenging scenario with multiple overlapping objects.

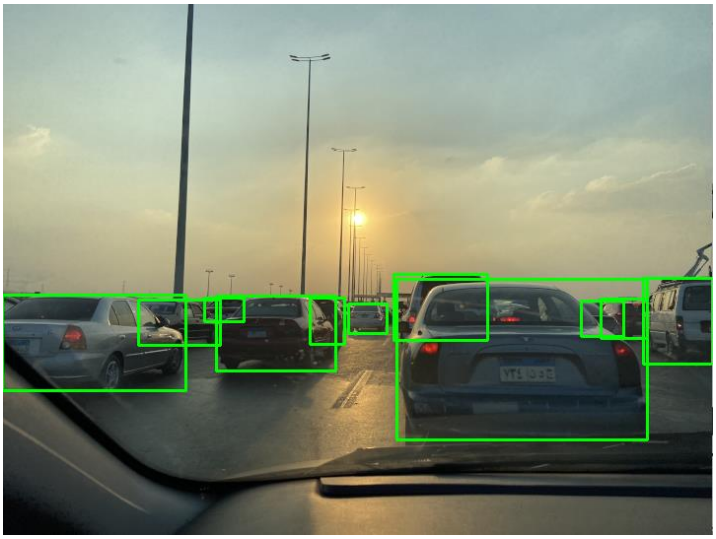

Figure 6. the application of the proposed method on the detection of objects in a challenging environment where many objects suffer from minor or major occlusions

Figure 7. the application of the proposed method was successful in detecting objects in the scene while prioritizing objects in the foreground for sooner detection compared to further objects

5. EXPERIMENTAL RESULTS

5.1. The Dataset

A modified version of the Pascal VOC 07 [26] dataset was chosen for evaluation. Instead of using the entire dataset, random selection has been applied to produce a reduced version. The decision to utilize a reduced version of the dataset is due to the resources required to augment the dataset for our model. For 2D-based object detection models, it was a straightforward operation to train and evaluate the performance results. but since our model relies on depth information, the dataset had to be augmented with artificial depth information. The added depth information was introduced as a gray scale mask where each pixel intensity value determines the distance of said pixel from the capturing source.

5.2. Balancing Performance

The hardware used for the development, training, and performance evaluation of the pipeline had an Intel 4th gen i5-4460 processor along with an Nvidia GTX 1060 graphics processing unit capable of 4.4 teraflops. 16.0 GB DDR3 @ 1600 MHz of Ram and 250 GB NVME PCI gen 3 SSD for storage. Models ran on Python (3.7) in Anaconda. Many Crucial packages were utilized and most importantly was TensorFlow version 2.0.

The pipeline aims to be applicable for training and testing on regularly accessible hardware. To achieve this performance goal, the pipeline had to be divided into multiple parts. Each part of the pipeline would have to be examined carefully to find the optimal hardware for it. The pipeline is divided into three main modules: obtaining video stream along with depth information, depth layer separation for object segmentation, and object detection. For video processing and depth capturing, the infrared projector and camera module used had an onboard computer which was utilized. Calculating the depth of information of each point projected on board freed up a lot of valuable resources. The second part of the pipeline focused on analyzing the depth information and segmenting different layers of a given scene in a way that would simplify object detection. This part was done on the CPU. Lastly, the object detection part utilizing the neural network was run on the GPU.

5.3. Neural Network Performance

The accuracy metric used is the mean average precision "mAP" at 0.5 intersections over the union. Figure 10 below compares the implemented pipeline to other two-stage models regarding performance, speed, and output. Multiple versions of these two-stage pipelines exist with different resolutions. But this study only focused on the models with resolutions similar to our proposed pipeline. As shown in the table, implementing the pipeline showed comparable performance in terms of speed and accuracy to other existing methods. The VGG-16 model showed better accuracy but a much longer processing time per frame which was not sufficient for real-time performance. On the contrary ResNet implementation showed a considerate performance but with less accuracy compared to VGG-16.

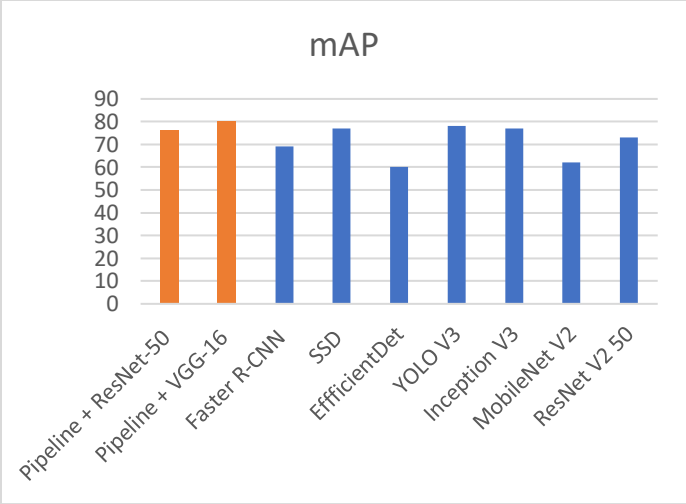

Figure 1

For ResNet-50, the model was trained with a stochastic gradient descent and a categorical cross-entropy loss function. The model was set to a constant learning rate of 0.001.The model was able to achieve an accuracy of 76%.

For VGG-16, the model was trained with the ADAM optimizer and a categorical cross-entropy loss function. This model was also trained with a constant learning rate set to 0.001. the model had more than 134 million training parameters. The model was able to achieve an accuracy of 80%.

5.4. Depth Retrieval Results

For the depth retrieval, the system was able to pull 30 frames per second. This was achieved using a maximum RGB resolution of 640 X 480 pixels. And 240 X 320 pixels for the depth stream[27]. The system relies on the depth stream to understand the distance of objects from the vision source. The depth values are represented as greyscale intensities where the nearest objects tend to have a higher intensity value "whites" while the further objects are represented with lower intensities "greys and blacks". The depth map was divided into separate layers where each layer represents a distance operating range. For a single extracted layer, the system was able to successfully perform object localization and classification utilizing one of the two implemented networks.

5.5. Real-Timeperformance

The real-time object detection system was able to find and localize objects successfully while maintaining near real-time performance. The speed was evaluated by how much time does it take to output a frame. The time unit used is milliseconds “ms”.

Table 2. Compares the results of the proposed method to the state-of-the-art methods in terms of speed and precision

Model name	Inf. ms
Pipeline +ResNet-50	86
Pipeline + VGG-16	200
Faster R-CNN	588
SSD	135
EffficientDet	66
YOLO V3	88
Inception V3	128
MobileNet V2	87
ResNet V2 50	101

6. CONCLUSION

This paper introduced an integrated pipeline that proposes a two-stage model that relies on an RGB color space input in addition to ground truth depth information RGB-D input feed. Using this ground truth data, we were able to eliminate the need for region proposals to localize potential objects in the scene. Instead, we utilize the ground truth depth information through a depth inference device to understand the scene. This allows for deconstructing any given scene into foreground, background, and multiple layers in between. These layers are then sorted according to a given use case. We aim to sort these layers and detect objects in them from nearer to further to prioritize the detection of the nearest objects first. This saves on computational power as depth data is obtained directly from the depth retrieval device without any need for further augmentation. The pipeline consists of three modules which are the neural network model, the depth retrieval system, and the object detection and labeling system. The system has a modular design meaning that any module can be easily modified or replaced independently from the other modules. ResNet 50 was able to successfully classify objects with a high average rate and comparable mean average precision while VGG-16 was able to perform classification with a considerable mean average precision but at a lower frame rate. The networks themselves were not designed with localization in mind as they only perform classification. The system extracts and crops the potential objects and passes that information to the classification networks with no need to alter or modify their original behavior. the neural network was examined carefully during training and testing. And it was discovered that utilizing GPU instead of the CPU made the training phase an order of magnitude faster. were designed for doing matrices operations at relatively high speeds. Meaning that batches of large data can be trained on a neural network at a relatively smaller time compared to if they were trained on the CPU. The GPU uplift aided in achieving real-time performance.

Simplifying the process of identifying objects in the scene, cropping them from the background, sorting and prioritizing them, and finally detecting the appropriate class for each object would result in a significant reduction in the amount of computing power required, allowing the pipeline to be used in real-time even at high resolutions. One of the most critical issues that have been addressed is how to speed the detection process to allow for real-time detection. Rather than

examining each frame in detail, the idea was to traverse the critical portions of each frame. Numerous techniques are applied in this scene optimization. Some use machine learning to discover areas of interest, while others use image processing to identify significant changes in color intensity to distinguish between items. Rather than that, we offer a novel strategy that uses depth data to further differentiate foreground items from background items to entirely isolate and crop them. On current systems, this strategy is quite undemanding. Additionally, this technique demonstrates the importance of calculating intersection over union since no multiple bounding boxes will overlap on a single item since the item can be readily cropped using depth information.

Prioritizing objects based on their depth information may result in efficient use of available hardware for real-time object identification (in some scenarios). Camera and lidar sensor advancements have enabled the acquisition of an extremely high-resolution scene. High-resolution sceneries with larger frames result in a considerably clearer and more detailed picture. However, better-resolution data has one significant disadvantage: they need far more processing power. For instance, a full HD video feed broadcasts at a rate of 30 frames per second and includes around two million pixels in each frame. In comparison, a 4K video feed that transmits at 30 frames per second includes around eight million frames per second. This is a fourfold increase in resolution, which would need the use of more powerful technology to process in a timely way. Thus, by using depth data, we may concentrate on the critical things in an image rather than on the whole picture. While the objects will retain their greater resolution, the time saved processing superfluous background items will outweigh the expense of the greater resolution.

The principle of object prioritization is scalable and may be used independently of the item identification technique used or the depth data collected from the scene. This implies that individual pipeline components (hardware or software) may be upgraded without impacting the whole system. For depth information, new and upgraded technology capable of capturing more precise data and a wider field of view might assist applications that depend on distance. For detection algorithms, new and better algorithms may significantly enhance detection accuracy or discover more classes, benefiting the whole pipeline.

References

[1] J. Wang, K. Chen, S. Yang, C. Change L, and D. Lin, "Region Proposal by Guided Anchoring," in CVPR 2019, California, 2019.
[2] C. Michaelis, B. Mitzkus, R. Geirhos and E. Rusak, Benchmarking Robustness in Object Detection: Autonomous Driving when Winter is Coming, arXiv, 2020.
[3] S. Ansari, Building Computer Vision Applications Using Artificial Neural Networks, Apress.
[4] J. Jordan, "An overview of object detection: one-stage methods," 2018. [Online]. Available: https://www.jeremyjordan.me/object-detection-one-stage/.
[5] G. Ross, D. Jeff, D. Trevor and M. Jitendra, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014.
[6] R. Girshick, "Fast R-CNN," 2015.
[7] R. Shaoqing, H. Kaiming, G. Ross and S. Jian, "Faster R-CNN: Towards Real-Time Object," 2016.
[8] L. Wei, A. Dragomir, E. Dumitru, S. Christian, R. Scott, F. Cheng-Yang and C. B. Alexander, "SSD: Single Shot MultiBox Detector," 2016.
[9] G. LAITILA, "Evaluation of pre-trained object detection models for the use in the SURE project," Tampere University of Applied Sciences, Tampere, Finland, 2021.
[10] R. Joseph, D. Santosh, G. Ross and F. Ali, "You Only Look Once: Unified, Real-Time Object Detection," 2016.
[11] N. Dvornik, K. Shmelkov, J. Mairal and C. Schmid, "BlitzNet: A Real-Time Deep Network for Scene Understanding," arXiv, 2017.

[12] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural computation, 1989.
[13] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in CVPR, 2014.
[14] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," in CVPR, 2015.
[15] H. Noh, S. Hong and B. Han, "Learning deconvolution network for semantic segmentation," in ICCV, 2015.
[16] C.-Y. Wang, I.-H. Yeh and H.-Y. Liao, "You Only Learn One Representation: Unified Network for Multiple Tasks," arXiv, 2021.
[17] V. Sitzmann, J. Martel, A. Bergman, D. Lindell and G. Wetzstein, "Implicit neural representations with periodic activation functions," In Advances in Neural Information Processing Systems, 2020.
[18] C.-Y. Wang, A. Bochkovskiy and H.-Y. Liao, "Scaled-YOLOv4: Scaling Cross Stage Partial Network," arXiv, 2021.
[19] J. JORDAN, "An overview of semantic image segmentation," JEREMY JORDAN, 21 5 2018. [Online].
[20] K. He, X. Zhang, S. Ren and J. Sun, Deep Residual Learning for Image Recognition, arXiv, 2015.
[21] S. Karen and Z. Andrew, "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION," 2015.
[22] P. Benjamin and A. Eliot, Hands-On Computer Vision with TensorFlow 2, Packt.
[23] A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'REILLY.
[24] P. K. Diederik and L. B. Jimmy, "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION," in ICLR 2015 , 2015.
[25] S. G. Estevão and H. Murielle, "Investigation on the Effect of a Gaussian Blur in Image Filtering and Segmentation," in ELMAR, 2011.
[26] "Common Objects in Context," [Online]. Available: https://cocodataset.org/#home.
[27] H. Fairhead, "Introduction to kinect," I Programmer, 1 2 2012. [Online].
[28] K. Alex, S. Ilya and E. H. Geoffrey, "ImageNet Classification with Deep Convolutional," 2012.
[29] A. James and W. Jarrett, Beginning Kinect Programming with the Microsoft Kinect SDK, Apress, 2012.
[30] H. Fairhead, "Using the Kinect Depth Sensor," I Programmer, 20 2 2012. [Online].
[31] H. Fairhead, "Using the Kinect Depth Sensor," I Programmer, 20 2 2012. [Online].
[32] J. Xiaoyue, J. Xiaoyue, P. Yanwei, P. Yanwei and F. Xiaoyi, Deep Learning in object detection and recognition, Springer.
[33] S. Connor and M. K. Taghi, "A survey on Image Data Augmentation," 2019.
[34] M. Everingham, A. Eslami, L. Van Gool, C. Williams, J. Winn and A. Zisserman, "The PASCAL Visual Object Classes Challenge: A Retrospective," International Journal of Computer Vision volume, vol. 111, 2015.

AUTHORS

Shehab Eldeen Ayman is an assistant lecturer at The British University In Egypt. Shehab received his bachelors from the software engineering department and graduated with a first-class honorary degree. Currently, he is pursuing Ph.D.in artificial intelligence. His main research interests are machine learning, data science,deep learning, signal processing, and computer vision.

Dr. Walid Hussein is an associate professor at the Faculty of Informatics and Computer Science since 2014, and the faculty research officer. Before joining BUE, Dr. Hussein was a postdoctoral researcher at the Department of Electrical Engineering, Faculty of Physical and Mathematical Sciences, Universidad de Chile, Santiago, Chile, where he conducted Multidisciplinary Research in Astronomy, Mining, Radar, Volcanology & Speech using Signal Processing Schemes (Funded by Chilean National Commission for Scientific and Technological Research [CONICYT], grant: 850,000 USD).

Professor Omar H. Karam is the Dean of the Faculty of Informatics and Computer Science at The British University in Egypt (BUE). Dr. Karam obtained his B.Sc. (1980) in Electrical Engineering (Communications and Electro-physics) from Alexandria University, Egypt, and his M.Sc. (1987) and Ph.D. (1993) in Computer Engineering from North Carolina State University, USA. He is on secondment since 2007 from Ain Shams University, Egypt where he is a Professor of Information Systems at the Faculty of Computer and Information Sciences.