International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 12 | Dec 2022 www.irjet.net p-ISSN: 2395-0072

Sanjaya: A Blind Assistance System

Hitanshu Parekh[1], Akshat Shah[2], Grinal Tuscano[3]

[1]*Student, Dept. of Information Technology, St. Francis Institute of Technology, Mumbai, Maharashtra, India*
[2]*Student, Dept. of Computer Engineering, Thadomal Shahani Engineering College, Mumbai, Maharashtra, India*
[3] *Professor, Dept. of Information Technology, St. Francis Institute of Technology, Mumbai, Maharashtra, India*

Abstract - *According to the World Health Organization (WHO), there are approximately 284 million people worldwide who have limited vision, and approximately 39 million who are completely blind [10]. Visually impaired people confront multiple challenges in their daily lives, particularly when navigating from one place to another on their own. They frequently rely on others to meet their daily needs. The proposed system is intended to help the visually impaired by identifying and classifying common everyday objects in real-time. This system provides voice feedback for improved comprehension and navigation. In addition, we have depth estimation, which calculates the safe distance between the object and the person, allowing them to be more self-sufficient and less reliant on others. We were able to create this model using TensorFlow and pre-trained models. The recommended strategy is dependable, affordable, realistic, and feasible.*

Key Words: **Object Detection, SSD Mobilenet, TensorFlow Object Detection API, COCO Dataset, Depth Estimation, Voice Alerts, Visually Impaired People, Machine Learning.**

1. INTRODUCTION

With the recent rapid growth of information technology (IT), a great deal of research has been conducted to solve everyday difficulties, and as a result, many conveniences for people have been supplied. Nonetheless, there are still many inconveniences for the visually impaired.

The most significant inconveniences that a blind person faces on a daily basis include finding information about objects and having difficulties with indoor mobility. Currently, the majority of vision-impaired people in India rely on a stick to identify potential obstacles in their path. Previous research included object analysis using ultrasonic sensors. However, it is difficult to tell exactly where an object is located using these approaches, especially in the presence of obstacles.

As a result, our contribution to solving this problem statement is to create a blind assistance system capable of object detection and real-time speech conversion so that they may hear it as and when they need it, as well as depth estimation for safe navigation. The SSD algorithm was combined with machine learning using the TensorFlow API packages to produce this system. This is encapsulated in a single system, an app, for a simple and better user experience.

2. Related Work

Balakrishnan et al.[1] described an image processing system for the visually impaired that uses wearable stereo cameras and stereo headphones. These devices are attached to helmets. The stereo cameras capture the scene in front of the visually impaired. The images are then analyzed, and significant features are retrieved to assist navigation by describing the scene. The distance parameters are captured using stereo cameras. The system is both complex and cost-ineffective due to the usage of stereo cameras.

Mahmud et al.[2] described a method that uses a cane and includes an ultrasonic sensor mounted to the cane and a camshaft position sensor (CMP) compass sensor (511), which gives information about various impediments and potholes. There are a handful of drawbacks to this strategy. Because the solution relies on a cane as its principal answer, only impediments up to knee height can be recognized. Furthermore, any iron item in the surrounding area might impact the CMP compass sensor. This strategy assists the blind in the outdoor world but has limitations in the indoor environment. The solution just detects the obstruction and does not attempt to recognize it, which might be a significant disadvantage.

Jiang et al.[3] described a real-time visual recognition system with results converted to 3D audio that is a multi-module system. A portable camera device, such as a GoPro, captures video, and the collected stream is processed on a server for real-time image identification using an existing object detection module. This system has a processing delay, making it difficult to use in real-time.

Jifeng Dai et al.[4] describe a fully convolutional network that is region-based. R-FCNN is used to recognize objects precisely and efficiently. As a result, this work can simply use ResNets as fully convolutional image classifier backbones to recognize the object. This article presents a straightforward yet effective RFCNN framework for object detection. This approach provides the same accuracy as the faster R-FCNN approach. As a result, adopting the state-of-the-art image classification framework was facilitated.

Choi D. et al.[5] describe current methods for detection models and also provide standard datasets. This paper discussed different detectors, such as one-stage and two-stage detectors, which assisted in the analysis of various object detection methods and gathered some traditional as well as new applications. It also mentioned various object detection branches.

A.A. Diaz Toro et al.[6] described the methods for developing a vision-based wearable device that will assist visually impaired people in navigating a new indoor environment. The suggested system assists the user in "purposeful navigation." The system detects obstacles, walking areas, and other items such as computers, doors, staircases, chairs, and so on.

D.P.Khairnar et al.[7] suggested a system that integrates smart gloves with a smartphone app. The smart glove identifies and avoids obstacles while also enabling visually impaired people to understand their environment. Various objects and obstacles in the surrounding area are recognized using smartphone-based obstacle and object detection.

V. Kunta et al.[8] presented a system that uses the Internet of Things (IoT) to connect the environment and the blind. Among other things, sensors are used to detect obstacles, damp floors, and staircases. The described prototype is a simple and low-cost smart blind stick. It is outfitted with several IoT modules and sensors.

Rajeshvaree Ravindra Karmarkar[9] proposes a deep learning-based item recognition system for the blind. Voice assistance can also help visually impaired people determine the location of objects. The deep learning model for object recognition employs the “You Only Look Once” (YOLO) approach. To assist the blind in learning information about items, a voice announcement is synthesized using text-to-speech (TTS). As a result, it utilizes an efficient object-detection technology that assists the visually impaired in finding things inside a particular environment.

Table I compares multiple TenserFlow mobile object detection models in terms of inference time and performance. The models' mAP performance varies substantially, with the more complex Faster RCNN Inception model achieving near-optimal performance but taking much longer to infer than the less complex Tiny Yolo model.

Existing solutions include limitations such as limited scope and functionality, inefficiency in cost, systems that are not portable, multiple sensor needs, and the inability to manage visually impaired persons in both indoor and outdoor situations in real time. We have made sincere efforts to combine all of the excellent aspects of the preceding solutions in order to create a full, portable, cost-effective system capable of efficiently solving real-world challenges.

Table -1: Inference time and mAP performance of trained models [8]

Model	Inference(ms)	mAP
Tiny Yolo V2	VOC 2007+2012	87.57
SSD MobileNet V1	COCO trainval	91.16
SSD MobileNet V2	COCO trainval	91.90
SSD Inception V2	COCO trainval	96.82
Faster RCNN Inception V2	COCO trainval	96.69

3.WORKING

The proposed design of our system (Fig. 1) is based on the detection of objects in the environment of a visually impaired individual. The suggested object detection technology needs a number of phases, from frame extraction to output recognition. To detect objects in each frame, a comparison between the query frame and database objects is performed. We present a real-time object recognition and positioning system. An audio file containing information about each identified object is activated. As a result, object detection and identification are addressed concurrently.

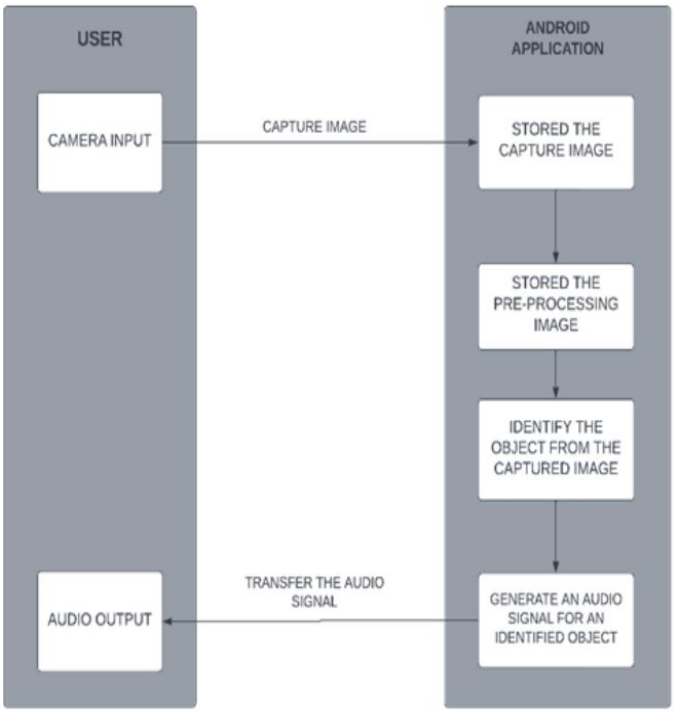

Fig -1: System Flowchart

The user must initially activate the program before using a camera to capture real-time, live images to feed into a model. Now that the image has been captured, the model must store it and evaluate it to determine how far it is from the subject. The output parameter of this information is then transformed into an audio signal.

For feature extraction, this system uses a Residual Network (ResNet) architecture, a sort of artificial neural network that allows the model to skip layers without affecting performance. [11]

3.1 Methodology

This system mainly comprises of

1. The system is configured in such a manner that an Android application captures real-time frames and sends them to a networked server, where all calculations are performed.

2. The server will use a pre-trained SSD detection model that has been trained using COCO datasets. It will then test and detect the output class using an accuracy metric.

3. Following testing with voice modules, the object's class will be translated into default voice notes, which will then be sent to the blind victims for assistance.

4. Along with object detection, we have an alert system that will calculate the approximate distance of the object. If the blind person is close to the frame or far away in a safer location, it will produce voice-based outputs in addition to distance units.

3.2 COCO Dataset

The Common Objects in Context (COCO) dataset is one of the most widely used open-source object identification databases for training deep learning algorithms. This database contains hundreds of thousands of images along with millions of pre-labeled objects for training [12].

3.3 TensorFlow Object Detection API

We used TensorFlow APIs to put it into action. The advantage of using APIs is that they provide us with access to a variety of standard actions. The TensorFlow object detection API is essentially a framework for developing a deep learning network that handles the object detection problem. There are trained models in their framework, which they term the "Model Zoo." This package contains the Open Images Dataset, the KITTI dataset, and the COCO dataset. COCO datasets are our primary interest here.

3.4 Models

Many pre-trained models now employ Tensorflow. There are several models for addressing this task. We are using SSD Mobile Net detection in this project since the majority of the system performs well, but you can also use MASK RCNN for more accuracy or SSD detection for faster accuracy. So let's take a closer look at the SSD algorithm.

3.5 SSD Architecture

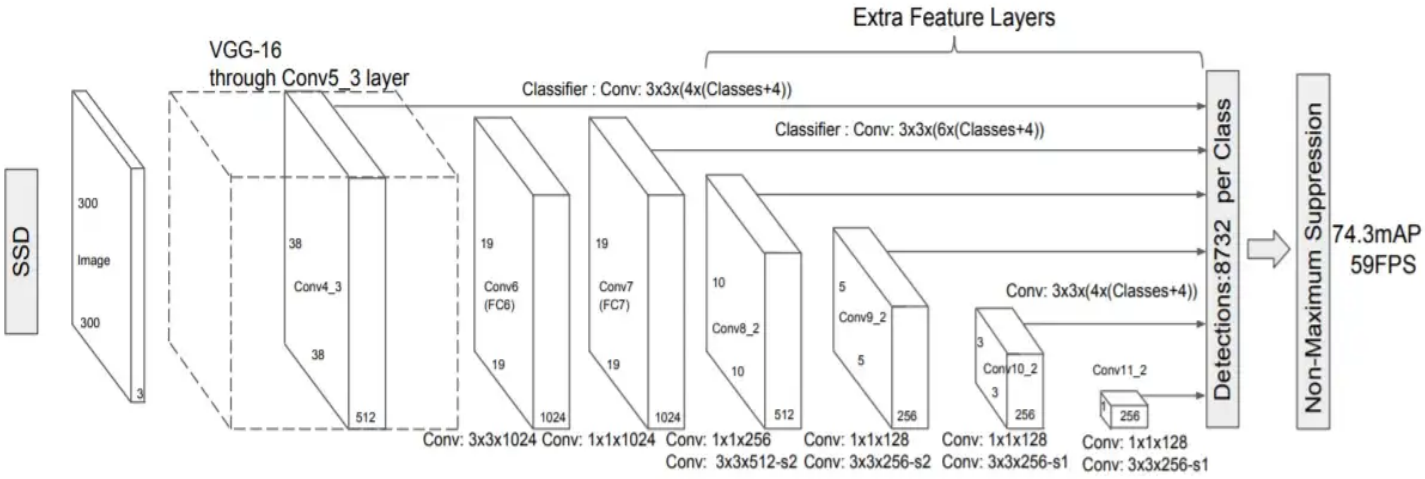

Fig -2: SSD Model Architecture [13]

SSD is made up of two parts: an SSD head and a backbone model.

The backbone model is essentially a trained image classification network that is used to extract features. Similar to ResNet, this is frequently a network trained on ImageNet that has had the last fully connected classification layer removed.

The SSD head is just one or more convolutional layers added to this backbone, and the outputs are interpreted as the bounding boxes and classifications of objects in the final layer activations' spatial position. As a result, we have a deep neural network that can extract semantic meaning from an input image while keeping its spatial structure, although at a lesser resolution.

The backbone produces 256 7x7 feature maps in ResNet 34 for an input image. SSD splits the picture into grid cells, with each grid cell is charge of identifying objects in that portion of the image. Detecting objects involves predicting the class and location of an object inside a given region.

3.6 Anchor Box

In SSD, each grid cell can have many anchor/prior boxes associated with it. Within a grid cell, these pre-defined anchor boxes are individually responsible for size and form. During training, SSD employs the matching phase to ensure that the anchor box matches the bounding boxes of each ground truth object inside an image. The anchor box with the most overlap with the item is responsible for predicting the object's class and location. After the network has been trained, this property is used to train it and predict the detected objects and their locations. In practice, each anchor box is assigned an aspect ratio and a zoom level. We all know that not all objects are square in form. Some are somewhat shorter, slightly longer, and slightly broader. To cater for this, the SSD architecture provides for pre-defined aspect ratios of the anchor boxes. The different aspect ratios may be configured using the ratios parameter of the anchor boxes associated with each grid cell at each zoom or scale level.

3.7 Zoom Level

The anchor boxes do not have to be the same size as the grid cells. The user may be looking for both smaller and bigger things within a grid cell. The zooms option is used to determine how much the anchor boxes should be scaled up or down in relation to each grid cell.

3.8 MobileNet

The MobileNet model, as its name implies, is intended for usage in mobile applications and is TensorFlow's first mobile computer vision model [14]. This model is built on the MobileNet model's idea of depthwise separable convolutions and forms factorized convolutions. These are functions that convert basic conventional convolutions into depthwise convolutions. Pointwise convolutions are another name for these 1 x 1 convolutions. These depthwise convolutions enable MobileNets to function by applying a generic, single filter-based concept to each of the input channels. These pointwise convolutions combine the outputs of depthwise convolutions with a 1 x 1 convolution. Like a typical convolution, both filters combine the inputs into a new set of outputs in a single step. The depth-wise identifiable convolutions divide this into two layers: one for filtering and another for combining. This factorization process has the effect of significantly reducing computation and model size.

3.9 Depth Estimation

The depth estimation or extraction feature refers to the techniques and algorithms used to produce a representation of a scene's spatial structure. To put it another way, it is used to compute the distance between two objects. Our prototype is used to assist the blind, and it seeks to alert the blind about impending obstacles. To do so, we must first determine the distance between the obstacle and the individual in any real-time circumstance. After detecting the object, a rectangular box is formed around it. If the object occupies the majority of the frame, the estimated distance of the object from the specific individual is calculated, subject to some constraints.

3.10 Voice Generation

Following the detection of an object, it is critical to notify the individual of the existence of that thing on his/her path. PYTTSX3 is essential for the voice creation module. Pyttsx3 is a Python conversion module that converts text to voice. To obtain a reference to a pyttsx. Engine instance, a factory function called as pyttsx.init() is invoked by an application. Pyttsx3 is a simple tool for converting text to voice.

The way this algorithm operates is that every time an object is detected, an approximation of its distance is generated. Texts are then displayed on the screen using the cv2 library's cv2.putText() function. We use Python-tesseract for character recognition to find hidden text in a picture. OCR recognizes text content on images and encodes it in a format that a computer can understand. This text detection is accomplished by image scanning and analyzing. Thus, text contained in images is identified and "read" using Python-tesseract.

As output, audio commands are created. When the object is too close, it displays the message "Warning: The object (class of object) is very close to you." "Watch out!" Otherwise, if the item is at a safer distance, a voice is generated that states, "The object is at a safer distance." This is achieved through the use of libraries such as pytorch, pyttsx3, pytesseract, and engine.io.

Pytorch's primary function is as a machine learning library. Pytorch is mostly used in the audio domain. Pytorch assists in loading the voice file in mp3 format. It also controls the audio dimension rate. As a result, it is used to manipulate sound qualities such as frequency, wavelength, and waveform.

4. Output

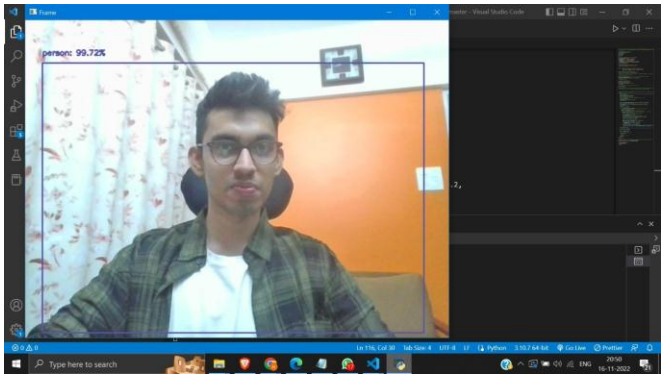

Fig -3: Object Detection of a person

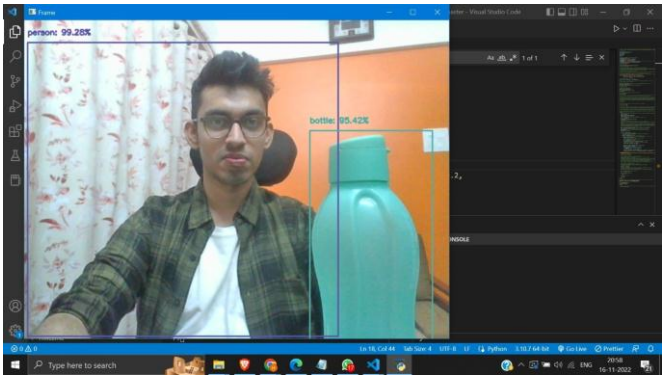

Fig -4: Object Detection of a person and bottle

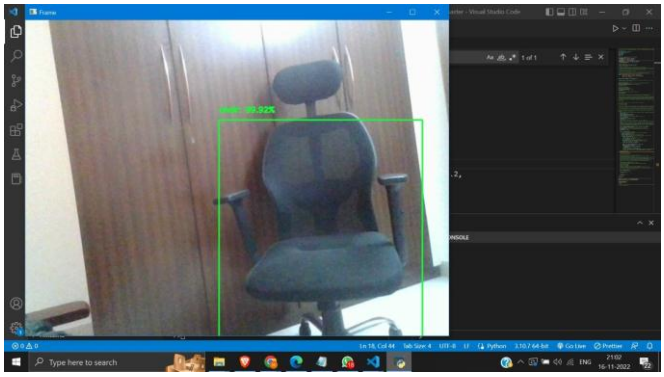

Fig -5: Object Detection of a chair

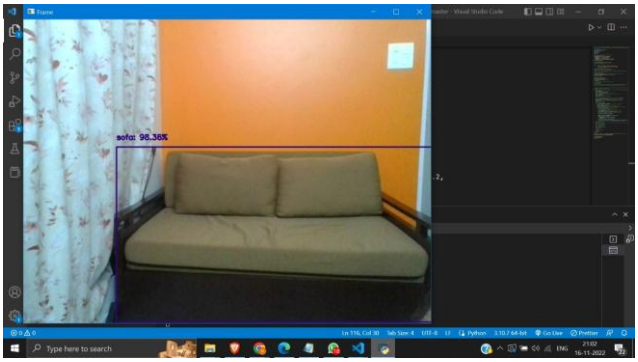

Fig -6: Object Detection of a sofa

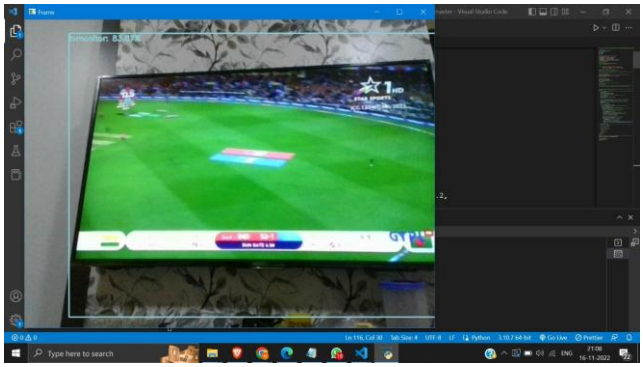

Fig -7: Object Detection of a television(TV)

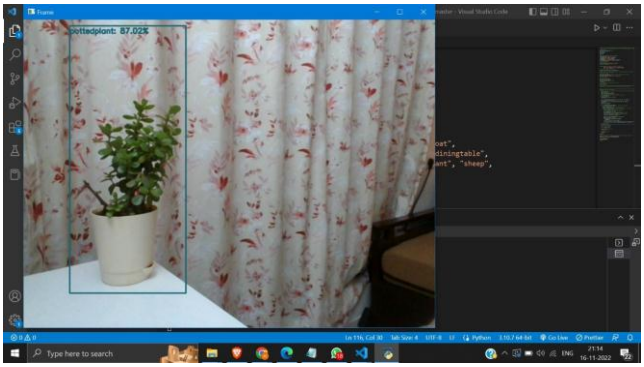

Fig -8: Object Detection of a plant

5. CONCLUSION

The suggested system successfully detects and labels 90 objects while also demonstrating its accuracy. When the person with the camera approaches the object, the model estimates the distance between the object and the camera and provides voice feedback. The dataset was tested on an SSD Mobilenet V1 model. The model showed less latency and was faster at detecting objects.

6. FUTURE SCOPE

We can improve the system's accuracy. Furthermore, the current system is based on the Android operating system and may be modified to work with any convenient device.

Future applications of the technology might include language translation, currency checking, reading literary works, a chatbot that allows the user to communicate and interact, smart shopping, email reading, real-time location sharing, sign board identification, such as the "Washroom" sign, etc.

7. ACKNOWLEDGEMENT

I would like to express my deep gratitude to Ms. Grinal Tuscano, Associate Professor at the St. Francis Institute of Technology and my research supervisor, for their patient guidance, enthusiastic encouragement, and useful critiques of this research work.

8. REFERENCES

[1] Balakrishnan, G. & Sainarayanan, G.. (2008). Stereo Image Processing Procedure for Vision Rehabilitation.. Applied Artificial Intelligence.

[2] N. Mahmud, R. K. Saha, R. B. Zafar, M. B. H. Bhuian and S. S. Sarwar, "Vibration and voice operated navigation system for visually impaired person," 2014 International Conference on Informatics, Electronics & Vision (ICIEV), 2014, pp. 1-5.

[3] Jiang, Rui & Lin, Qian & Qu, Shuhui. (2016). Let Blind People See: Real-Time Visual Recognition with Results Converted to 3D Audio.

[4] Jifeng Dai, Yi Li, Kaiming He and Jian Sun, "R-FCN: Object Detection via Regionbased Fully Con- volutional Networks." Conf. Neural Inform. Process. Syst., Barcelona, Spain,Dec. 4-6, 2016, pp. 379-387

[5] Choi D., and Kim M. ', "Trends on Object Detection Techniques Based on Deep Learning," Electronics and Telecommunications Trends, Vol. 33, No. 4, pp. 23- 32, Aug. 2018.

[6] A. A. Diaz Toro, S. E. Campaña Bastidas and E. F. Caicedo Bravo, "Methodology to Build a Wearable System for Assisting Blind People in Purposeful Navigation," 2020 3rd International Conference on Information and Computer Technologies (ICICT), 2020, pp. 205-212.

[7] D. P. Khairnar, R. B. Karad, A. Kapse, G. Kale and P. Jadhav, "PARTHA: A Visually Impaired Assistance System," 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA), 2020, pp. 32-37.

[8] V. Kunta, C. Tuniki and U. Sairam, "Multi-Functional Blind Stick for Visually Impaired People," 2020 5th International Conference on Communication and Electronics Systems (ICCES), 2020, pp. 895-899.

[9] Rajeshvaree Ravindra Karmarkar, "Object Detection System for the Blind with Voice Guidance", International Journal of Engineering Applied Sciences and Technology(IJEAST), 2021, pp. 67-70.

[10] https://www.indiatoday.in/education-today/gk-current-affairs/story/world-sight-day-2017-facts-and-figures-1063009-2017-10-12

[11] https://builtin.com/artificial-intelligence/resnet-architecture

[12] https://deepai.org/machine-learning-glossary-and-terms/CoCo%20Dataset

[13] https://iq.opengenus.org/ssd-model-architecture/

[14] https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470

[15] https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470