POLARIS GLOBAL JOURNAL OF SCHOLARLY RESEARCH AND TRENDS

Object Detection using the ImageAI Library in Python

Renas Rajab Asaad[1], Rasan Ismael Ali[2], Awaz Ahmad Shaban[3], Merdin Shamal Salih[4]
[1]*Department of Computer Science, Nawroz University, Duhok, Kurdistan Region of Iraq*
[2,3,4]*Reseacher*
Corresponding Email: renas.beda89@Gmail.com

ARTICLEINFO

KEYWORDS
Python, OpenCV, Computer Vision, Image Processing.

ABSTRACT

Recent progress in deep learning methods has shown that key steps in object detection and recognition, including feature extraction, region proposals, and classification, can be done using ImageAi libraries. Object detection is a computer vision technique that works to identify and locate objects within an image or video. Specifically, object detection draws bounding boxes around these detected objects, which allow us to locate where said objects are in a given scene. Object detection is commonly confused with image recognition, so before we proceed, it's important that we clarify the distinctions between them. In that it aids in our comprehension and analysis of scenes in images or videos, object detection is intrinsically tied to other related computer vision techniques like image recognition and image segmentation. Significant variations. Image segmentation develops a pixel-level comprehension of a scene's elements while image recognition just produces a class label for an identified object. Object detection differs from these other jobs in that it has the capacity to specifically find objects inside an image or video. This enables us to count such things and later track them.

E-ISSN: 2961-3809

How to cite:
Rajab Asaad, R., Ismael Ali, R., Ahmad Shaban, A., & Shamal Salih, M. (2023). Object Detection using the ImageAI Library in Python. Polaris Global Journal of Scholarly Research and Trends, 2(2), 1–9. https://doi.org/10.58429/pgjsrt.v2n2a143

1. INTRODUCTION

The field of computer vision has developed rapidly in recent years, and brought us science fiction results a few years ago. Here we are witnessing a real revolution in the development of this field, starting from the analysis of X-ray images and diagnosing patients to self-driving cars, and these developments and results have many reasons; The most important of which is to build better and more intuitive computing resources.

The task of object detection in computer vision involves identifying the presence of one or more objects in an image and specifying their location and type. It is one of the difficult problems that involves taking advantage of the methods of identifying objects (where they are), their location (what is their location) and their classification (what they are). Detection algorithms in recent years have achieved new results, including the YOLO algorithm developed by Joseph Redmon and others in 2015. It is a Convolution Neural Network CNN that is able to detect multiple objects in real time with just one pass, so it is named after you only look once. "You Look Only Once".

It is a computer vision technology that allows us to determine the type and location of objects in an image or video. This technology can count the total number of objects in a given scene and pinpoint their exact locations.

Object detection is sometimes confused with image classification, so I will briefly explain the difference between these two processes.

In image classification, a specific image is classified into pre-scheduled categories. Where the image represents the input of the process, and the output represents the name of the element in the image. For example, if the input is an image containing a cat, the output will be the word “cat.” If the input is an image containing two cats, the output will also be “a cat.” From this we can conclude that the process of classifying images, despite its importance, is limited to the results it gives us. The opposite of this is in the object detection process where each of the detected elements is defined within a frame, in addition to that the location of this element is determined in the given scene. From this we conclude that object recognition is a more complex process than image classification and gives us more information about the image and thus we can benefit from it more.

1.1 The importance of object detection technology and its applications

The importance of object detection is represented in the ability of the applied algorithm to determine the type and location of a specific object in an image or video, the speed required to do so, as well as the number of objects that this algorithm can detect. The importance of this we see when applying the detection of objects in practice. Here I will talk about some of the most important use cases for detecting purposes.

1.2 Self-driving cars

Real-time object detection is one of the most important keys to the success of intelligent driving systems. These systems need to know and determine the locations of the surrounding objects and also recognize the traffic lights, all in order to be able to drive safely and effectively.

1.3 Anomaly detection

Object detection can be used in the agricultural field. As the object detection model can detect possible threats to the agricultural crop that the human eye cannot follow in the normal case. It can also be used in the medical field to detect abnormalities within medical images or, for example, in skin care by identifying and detecting acne spots.

The beauty of this is that these techniques enable us to immediately obtain information or identify specific diseases that, in the normal case, can only be detected by specialists, and are now within reach after a little training. Object detection can also be applied to video surveillance and crowd counting, which has benefits for organization and security.

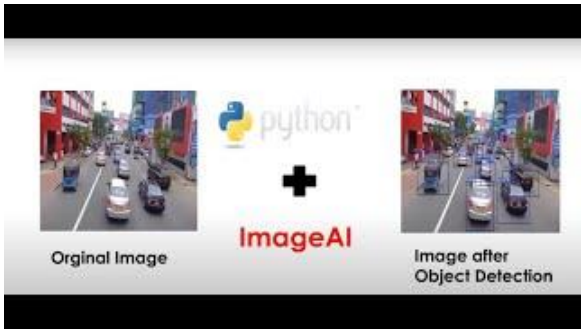

Figure 1: ImageAI test

2. Literature Review

Object detection models based on deep learning typically contain two components. An encoder receives an image as input and processes it through a number of layers and blocks that teach them to extract statistical features that are used to identify and locate things. A decoder receives the encoder's outputs and determines the bounding boxes and labels for each item.

A pure regressor serves as the simplest decoder. The regressor directly predicts the location and size of each bounding box by connecting to the encoder's output. The model's output is the object's and its area in the image's X, Y coordinate pair. This kind of model has limitations despite being straightforward. In advance, you must indicate how many boxes there will be. If there are two dogs in your image but your model can only identify one, one will be left unidentified. However, pure regressor-based models might be a viable choice if you know in advance how many things you need to forecast in each image.

A region proposal network is an expansion of the regressor method. The model in this decoder suggests areas of an image where it thinks an object might be present. A classification subnetwork is then used to assign a label to the pixels in these locations (or reject the proposal). The pixels that contain those regions are subsequently sent through a classification network. This approach has the advantage of providing a more precise, adaptable model that can suggest any number of locations that might include a bounding box. But the reduced computing efficiency comes at the expense of the increased precision.

Custom object detection models can be trained using ImageAI's straightforward and effective method using the YOLOv3 architecture. You can use any set of photos that correspond to any kind of object of interest to train your own model in this way.

This class, Recognition Model Training, enables you to train object detection models using the YOLOv3 and TinyYOLOv3 model on image datasets that are annotated in the YOLO format. The training process provides a JSON file that links the names of the items in your image dataset to the anchors used for detection as well as several models.

For performing Video Object Detection and Tracking as well as Video Analysis, ImageAI offered incredibly powerful yet simple to use classes and functions. All of these tasks are possible with ImageAI thanks to cutting-edge deep learning algorithms like RetinaNet, YOLOv3, and TinyYOLOv3. You can use ImageAI to perform detection tasks and analyze recorded movies as well as live video streams from IP and device cameras. You can utilize the classes listed below along with their corresponding functions. These classes can be incorporated into any conventional Python program that you are creating, whether it be a website, a Windows/Linux/MacOS application, a system that supports or is a part of a Local-Area Network.

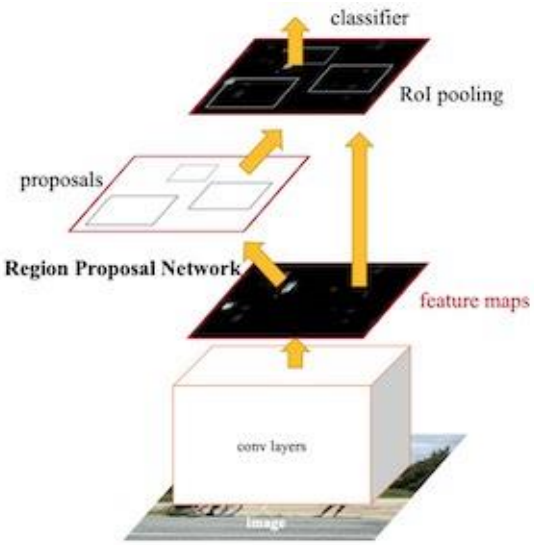

Figure 2: Object detector based on ML

2.1 Object Detection using Machine Learning algorithms

In this article, we will focus on deep learning-based object detection approaches that are the current state of the science. There are currently different architectures for detecting objects such as R-CNN, Fast R-CNN, Faster R-CNN, YOLO, you only look once. Only once, as well as the SSD one-step detector.

Two major challenges can be distinguished for most of the algorithms used. First we need to divide the image into sections so that each section contains one of the shapes we want to identify. To achieve this purpose, many approaches have been developed, which vary in their effectiveness and speed. Through it, the image enters the encoder, which passes the image through many layers and templates, which in turn extract the statistical features for scheduling and locating the objects. The output is taken and put into a decoder which predicts the possible frames for each object. The second challenge is categorizing this part of the picture. For this purpose, these parts of the image are fed into a pre-trained Convolution Neural Network CNN, and the process will be transformed into image classification. There are many details that we have overlooked for the time being, as they are not the subject of our article.

2.2 Yolo algorithm for object detection

Yolo's algorithm uses a convolution neural network to divide the input into a network of cells that each directly predict the bounding box and object classification. The result will be a large number of candidate squares that are combined into the final prediction at the post-processing step.

There are three versions of this algorithm. YOLOv1 proposed the general structure, July YOLOv2 improved the design and made use of predefined anchor boxes to improve the proposed bounding box, and July YOLOv3 improved the model structure and training process.

Although the results of this algorithm are accurate, convolutional neural network region R-CNNs give more accurate results. But Yolo is popular for its speed, as it displays results in real time. In this blog we will focus on the third version of it.

2.3 Keras Yolo Code Repository v3

The Yolo model is difficult to implement from scratch, especially for beginners, as it requires the development of many elements of the models for training and forecasting. For example, when using a pre-trained model that requires writing a complex block of code to filter and interpret the bounding boxes generated from the model. Therefore, DarkNet provided a repository on GitHub called DarkNet GitHub that includes the code for all versions of Yolo written in C, with the necessary references on how to use these codes to detect objects. Therefore, one of the following projects in stock can be relied upon, which greatly facilitated the use of Yolo.

- The YAD2K Yet Another Darknet 2 Keras project: The certified benchmark for Yolo v2 provided scripts to convert pre-trained weights into Keras format and used the pre-trained model to make predictions. It also provided the code required to filter bounding boxes. Many third-party applications took this code as a starting point and developed it to support Yolo v3.
- keras-yolo2 project: which provides code similar to YOLOv2 as well as a detailed explanation of how to use it.
- keras-yolo3 project: The most popular project for training and detecting objects using pre-trained yolo models. The code in the project is provided under an open-source license from the Massachusetts Institute of Technology (MIT). Which is what we will adopt in this blog.

3. Object detection implementation through ImageAI's computer vision AI application library

3.1 ImageAI Applications Library for Artificial Intelligence in Computer Vision

It is a library built into the Python programming language to encourage developers and students to write applications in the field of deep learning, especially computer vision, through a few lines of code. This library provides us with a set of easy-to-use classes and functions to perform object detection. These classes can detect objects in a video or image as well as display an analysis of the results and can be integrated into any Python program in a web application or Windows/Linux/MacOS operating system. This library supports templates such as Yolo and RetinaNEt, through which it can detect approximately 80 different shapes.

3.2 Practical application of this process to an image

To implement object recognition using imageai on the CoLab platform, we first need to perform the following steps.
Run pip install imageai -upgrade to install imageai on Colab. In the normal case, we also need to install other libraries such as Google Tensorflow, Keras, and OpenCv, but they are already installed on the Colab platform.
Download Retna-Net through this following link, then upload it to the Colab platform.
We upload the image on which we want to execute the code to the work environment in the Colab platform

```
from imageai.Detection import ObjectDetection

import os

execution_path = os.getcwd()

detector = ObjectDetection()

detector.setModelTypeAsRetinaNet()

detector.setModelPath( os.path.join(execution_path , "resnet50_coco_best_v2.0.1.h5"))

detector.loadModel()

detections = detector.detectObjectsFromImage(input_image=os.path.join(execution_path, "Dog&Cat.jpg"), output_image_path=os.path.join(execution_path , "imagenew.jpg"))

for eachObject in detections:

    print(eachObject["name"] , " : " , eachObject["percentage_probability"] )
```

In the first line we have imported the ObjectDetction class from the imageai computer vision AI application library.
In the second line we have imported the os class in Python. In line 4 we define the execution-path variable to define the path to the Python file that is currently on the Colab platform and to which we have attached the RetinaNet model and the image to be exposed.
In line 6 we define a detector variable which represents the ObjectDetction class. On the next line we define the path to the uploaded form. On line 8 the form is uploaded to the program. Then we come to the most important function in the class, which is detectObjectsFromImage, through which we know the input image that we want to detect the objects in and we know the output image. It is also possible to specify some other properties such as extracting objects and placing them in separate images, all of which are found in the library documentation on the link below. Of course, all these functions are for the library of artificial intelligence applications in computer vision imageai, and they were written by developers to facilitate dealing with these techniques and obtain results with minimal effort. In the normal case one needs to deal with more complex libraries. As shown in figure (1).

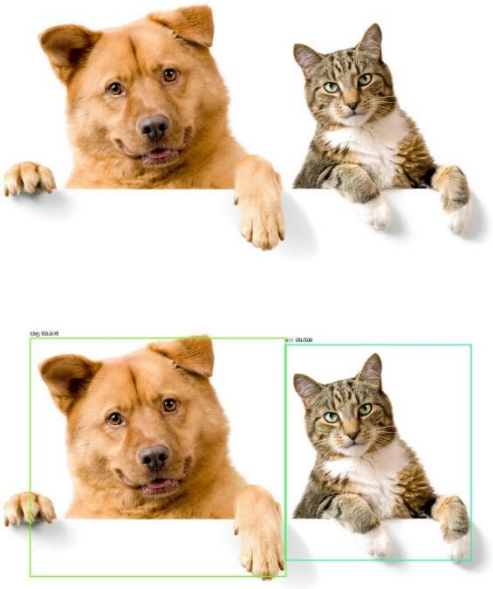

Figure 3: Object Detector

The result which reflects the accuracy of the detection:
cat: 99.27995800971985
dog: 95.54893374443054

3.3 Practical application of this process on video

To apply object detection to a video, we will follow almost the same steps. We just have to use a different class than the one we used on the images which is VideoObjectDetection. There also another model has been used, Yolov3, which can be downloaded from this following link.
The following code on the Colab platform, uploaded the Yolov3 version 3 form and the video to be exposed to the Colab platform.

```
from imageai.Detection import VideoObjectDetection

import os

execution_path = os.getcwd()

detector = VideoObjectDetection()
```

```
detector.setModelTypeAsYOLOv3()

detector.setModelPath( os.path.join(execution_path , "yolo.h5"))

detector.loadModel()

 video_path =
detector.detectObjectsFromVideo(input_file_path=os.path.join(execution_path,
"traffic.mp4"), output_file_path=os.path.join(execution_path,
"traffic_detected"), frames_per_second=20, log_progress=True)

print(video_path)
```

To detect objects in a video, we used different categories and functions, but in general the same method was used when detecting an image.

4. Conclusion

In conclusion, object detection is an essential task in computer vision that enables us to identify and locate objects in images and videos. The ImageAI library in Python provides a simple and efficient way to implement object detection models with pre-trained models such as RetinaNet, YOLOv3, and TinyYOLOv3. The library also offers powerful tools for fine-tuning these models on custom datasets, making it suitable for a wide range of applications. With ImageAI, developers can easily build intelligent applications that can recognize and locate objects in images and videos, ranging from surveillance systems to autonomous vehicles. The library's simple API and extensive documentation make it accessible even to developers with limited experience in computer vision. Overall, ImageAI is a powerful tool for object detection in Python, and it is worth exploring for anyone interested in computer vision or machine learning.

REFERENCES

Mohammed, M.A., et al.: Adaptive secure malware efficient machine learning algorithm for healthcare data. CAAI Trans. Intell. Technol. 1– 12 (2023). https://doi.org/10.1049/cit2.12200

Asaad, Renas Rajab. (2014). An Investigation of the Neuronal Dynamics Under Noisy Rate Functions. Thesis (M.S.), Eastern Mediterranean University, Institute of Graduate Studies and Research, Dept. of Computer Engineering, Famagusta: North Cyprus.

Ur Rahman, A.; Saeed, M.; Saeed, M.H.; Zebari, D.A.; Albahar, M.; Abdulkareem, K.H.; Al-Waisy, A.S.; Mohammed, M.A. A Framework for Susceptibility Analysis of Brain Tumours Based on Uncertain Analytical Cum Algorithmic Modeling. *Bioengineering* 2023, *10*, 147. https://doi.org/10.3390/bioengineering10020147

Asaad, R. R., Abdurahman, S. M., & Hani, A. A. (2017). Partial Image Encryption using RC4 Stream Cipher Approach and Embedded in an Image. Academic Journal of Nawroz University, 6(3), 40–45. https://doi.org/10.25007/ajnu.v6n3a76

D. A. Zebari, H. Haron, D. M. Sulaiman, Y. Yusoff and M. N. Mohd Othman, "CNN-based Deep Transfer Learning Approach for Detecting Breast Cancer in Mammogram Images," 2022 IEEE 10th Conference on Systems, Process & Control (ICSPC), Malacca, Malaysia, 2022, pp. 256-261, doi: 10.1109/ICSPC55597.2022.10001781.

Almufti, S., Marqas, R., & Asaad, R. (2019). Comparative study between elephant herding optimization (EHO) and U-turning ant colony optimization (U-TACO) in solving symmetric traveling salesman problem (STSP). Journal Of Advanced Computer Science & Technology, 8(2), 32.

Ibrahim, D. A., Zebari, D. A., Mohammed, H. J., & Mohammed, M. A. (2022). Effective hybrid deep learning model for COVID-19 patterns identification using CT images. Expert Systems, 39(10), e13010. https://doi.org/10.1111/exsy.13010

Asaad, R. R., & Abdulnabi, N. L. (2018). Using Local Searches Algorithms with Ant Colony Optimization for the Solution of TSP Problems. Academic Journal of Nawroz University, 7(3), 1–6. https://doi.org/10.25007/ajnu.v7n3a193

M. A. Jubair et al., "A QoS Aware Cluster Head Selection and Hybrid Cryptography Routing Protocol for Enhancing Efficiency and Security of VANETs," in IEEE Access, vol. 10, pp. 124792-124804, 2022, doi: 10.1109/ACCESS.2022.3224466.

Asaad, R. R., & Ali, R. I. (2019). Back Propagation Neural Network(BPNN) and Sigmoid Activation Function in Multi-Layer Networks. Academic Journal of Nawroz University, 8(4), 216–221. https://doi.org/10.25007/ajnu.v8n4a464

Arshad, M.; Saeed, M.; Rahman, A.U.; Zebari, D.A.; Mohammed, M.A.; Al-Waisy, A.S.; Albahar, M.; Thanoon, M. The Assessment of Medication Effects in Omicron Patients through MADM Approach Based on Distance Measures of Interval-Valued Fuzzy Hypersoft Set. *Bioengineering* 2022, *9*, 706. https://doi.org/10.3390/bioengineering9110706

Almufti, S., Asaad, R., & Salim, B. (2018). Review on elephant herding optimization algorithm performance in solving optimization problems. International Journal of Engineering & Technology, 7, 6109-6114.

Mohammed, H.J.; Al-Fahdawi, S.; Al-Waisy, A.S.; Zebari, D.A.; Ibrahim, D.A.; Mohammed, M.A.; Kadry, S.; Kim, J. ReID-DeePNet: A Hybrid Deep Learning System for Person Re-Identification. *Mathematics* 2022, *10*, 3530. https://doi.org/10.3390/math10193530

Asaad, R. R. (2021). Penetration Testing: Wireless Network Attacks Method on Kali Linux OS. Academic Journal of Nawroz University, 10(1), 7–12. https://doi.org/10.25007/ajnu.v10n1a998

D. A. Zebari, D. M. Sulaiman, S. S. Sadiq, N. A. Zebari and M. S. Salih, "Automated Detection of Covid-19 from X-ray Using SVM," 2022 4th International Conference on Advanced Science and Engineering (ICOASE), Zakho, Iraq, 2022, pp. 130-135, doi: 10.1109/ICOASE56293.2022.10075586.

Asaad, R. R. (2019). Güler and Linaro et al Model in an Investigation of the Neuronal Dynamics using noise Comparative Study. Academic Journal of Nawroz University, 8(3), 10–16. https://doi.org/10.25007/ajnu.v8n3a360

Al-Waisy AS, Ibrahim DA, Zebari DA, Hammadi S, Mohammed H, Mohammed MA, Damaševičius R. 2022. Identifying defective solar cells in electroluminescence images using deep feature representations. PeerJ Computer Science 8:e992 https://doi.org/10.7717/peerj-cs.992

Rajab Asaad, R., & Masoud Abdulhakim, R. (2021). The Concept of Data Mining and Knowledge Extraction Techniques. Qubahan Academic Journal, 1(2), 17–20. https://doi.org/10.48161/qaj.v1n2a43

Asaad, R. R., Ahmad, H. B., & Ali, R. I. (2020). A Review: Big Data Technologies with Hadoop Distributed Filesystem and Implementing M/R. Academic Journal of Nawroz University, 9(1), 25–33. https://doi.org/10.25007/ajnu.v9n1a530

D. A. Zebari, S. S. Sadiq and D. M. Sulaiman, "Knee Osteoarthritis Detection Using Deep Feature Based on Convolutional Neural Network," 2022 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Iraq, 2022, pp. 259-264, doi: 10.1109/CSASE51777.2022.9759799.

Rajab Asaad, R. (2021). Review on Deep Learning and Neural Network Implementation for Emotions Recognition . Qubahan Academic Journal, 1(1), 1–4. https://doi.org/10.48161/qaj.v1n1a25

Ferinia, R., Kumar, D.L.S., Kumar, B.S. et al. Factors determining customers desire to analyse supply chain management in intelligent IoT. J Comb Optim 45, 72 (2023). https://doi.org/10.1007/s10878-023-01007-8

Asaad, R. R., Abdulrahman, S. M., & Hani, A. A. (2017). Advanced Encryption Standard Enhancement with Output Feedback Block Mode Operation. Academic Journal of Nawroz University, 6(3), 1–10. https://doi.org/10.25007/ajnu.v6n3a70

Poornima, E., Muthu, B., Agrawal, R. et al. Fog robotics-based intelligence transportation system using line-of-sight intelligent transportation. Multimed Tools Appl (2023). https://doi.org/10.1007/s11042-023-15086-6

Abdulfattah, G. M., Ahmad, M. N., & Asaad, R. R. (2018). A reliable binarization method for offline signature system based on unique signer's profile. INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING INFORMATION AND CONTROL, 14(2), 573-586.

Almufti, S. M. (2022). Vibrating Particles System Algorithm performance in solving Constrained Optimization Problem. Academic Journal of Nawroz University, 11(3), 231–242. https://doi.org/10.25007/ajnu.v11n3a1499

Almufti, S. M., Ahmad, H. B., Marqas, R. B., & Asaad, R. R. (2021). Grey wolf optimizer: Overview, modifications and applications. International Research Journal of Science, Technology, Education,and Management, 1(1),1-1.

Poornima, E., Muthu, B., Agrawal, R. et al. Fog robotics-based intelligence transportation system using line-of-sight intelligent transportation. Multimed Tools Appl (2023). https://doi.org/10.1007/s11042-023-15086-6

Asaad, R. R., Sulaiman, Z. A., & Abdulmajeed, S. S. (2019). Proposed System for Education Augmented Reality Self English Learning. Academic Journal of Nawroz University, 8(3), 27–32. https://doi.org/10.25007/ajnu.v8n3a366

Saman M. Almufti. (2022). Artificial Bee Colony Algorithm performances in solving Welded Beam Design problem. Computer Integrated Manufacturing Systems, 28(12), 225–237. Retrieved from http://cims-journal.com/index.php/CN/article/view/405

Asaad, R. R. (2020). Implementation of a Virus with Treatment and Protection Methods. ICONTECH INTERNATIONAL JOURNAL, 4(2), 28-34. https://doi.org/10.46291/ICONTECHvol4iss2pp28-34

M. Almufti, S. (2022). Hybridizing Ant Colony Optimization Algorithm for Optimizing Edge-Detector Techniques. Academic Journal of Nawroz University, 11(2), 135–145. https://doi.org/10.25007/ajnu.v11n2a1320

Boya Marqas, R., M. Almufti, S., & Rajab Asaad, R. (2022). FIREBASE EFFICIENCY IN CSV DATA EXCHANGE THROUGH PHP-BASED WEBSITES. Academic Journal of Nawroz University, 11(3), 410–414. https://doi.org/10.25007/ajnu.v11n3a1480

Almufti, S. (2022). Vibrating Particles System Algorithm: Overview, Modifications and Applications. ICONTECH INTERNATIONAL JOURNAL, 6(3), 1–11. https://doi.org/10.46291/ICONTECHvol6iss3pp1-11

Rajab Asaad, R., & Luqman Abdulnabi, N. (2022). A Review on Big Data Analytics between Security and Privacy Issue. Academic Journal of Nawroz University, 11(3), 178–184. https://doi.org/10.25007/ajnu.v11n3a1446

Yahya Hussien , A., & Rajab Asaad, R. (2022). Review on Social Media and Digital Security. Qubahan Academic Journal, 2(2), 1–4. https://doi.org/10.48161/qaj.v2n2a119

Asaad, R. R. (2022). Keras Deep Learning for Pupil Detection Method . Academic Journal of Nawroz University, 10(4), 240–250. https://doi.org/10.25007/ajnu.v10n4a1328

Asaad, R. R., & Segerey, R. I. (2018). School Management Application Using iOS. Academic Journal of Nawroz University, 7(4), 38–44. https://doi.org/10.25007/ajnu.v7n4a269

Asaad, R. R., Mustafa, R. F., & Hussien, S. I. (2020). Mortality Statistics and Cause of Death at Duhok City from The Period (2014-2019) Using R Language Data Analytics. Academic Journal of Nawroz University, 9(3), 1–7. https://doi.org/10.25007/ajnu.v9n3a699

Asaad, R. R. (2021). A Study on Instruction Formats on Computer Organization and Architecture. ICONTECH INTERNATIONAL JOURNAL, 5(2), 18-24. https://doi.org/10.46291/ICONTECHvol5iss2pp18-24

Asaad, R. R. (2021). Virtual reality and augmented reality technologies: A closer look. Virtual reality, 1(2).

Asaad, R. R. A Review: Emotion Detection and Recognition with Implementation on Deep Learning/Neural Network.

Asaad, R. R., Saeed, V. A., & Abdulhakim, R. M. (2021). Smart Agent and it's effect on Artificial Intelligence: A Review Study. ICONTECH INTERNATIONAL JOURNAL, 5(4), 1-9.

Asaad, R. R. A Asaad, R. R. A Review: Emotion Detection and Recognition with Implementation on Deep Learning/Neural Network.

Asaad, R. R., & Saeed, V. A. (2022). A Cyber Security Threats, Vulnerability, Challenges and Proposed Solution. Applied Computing Journal, 2(4), 227-244. https://doi.org/10.52098/acj.202260

Renas Rajab Asaad. (2022). Support vector machine classification learning algorithm for diabetes prediction. International Research Journal of Science, Technology, Education, and Management, 2(2), 26–34. https://doi.org/10.5281/zenodo.6975670