iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** V **Month of publication:** May 2022

DOI: https://doi.org/10.22214/ijraset.2022.43303

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

Custom Object Detection and Analysis in Real Time - YOLOv4

Indira Adak[1], Aayush Kumar[2], Amit Kumar[3], Avanish Chandra Dubey[4]
[1, 2, 3, 4]*Department of Information Technology, Inderprastha Engineering College, Ghaziabad, UP, India*

Abstract: ***Object recognition is one of the most basic and complex problems in computer vision, which seeks to locate object instances from the enormous categories of already defined and readily available natural images. The object detection method aims to recognize all the objects or entities in the given picture and determine the categories and position information to achieve machine vision understanding. Several tactics have been put forward to solve this problem, which is more or less inspired by the principles based on Open Source Computer Vision Library (OpenCV) and Deep Learning. Some are relatively good, while others fail to detect objects with random geometric transformations. This paper proposes demonstrating the " HAWKEYE " application, a small initiative to build an application working on the principle of EEE i.e. (Explore→Experience→Evolve).***
Keywords: Convolution Neural Network, Object detection, Image classification, Deep learning, Open CV, Yolov4.

I. INTRODUCTION

We humans easily identify the objects from the images in real-time, but in the case of algorithms, it's a bit complicated before CNN. CNN was brought by Alex net in 2012, and after that, most things have changed from that time. Many enhancements have happened, and thus we can say that today's machines are faster and more accurate than humans. And its preciseness is improving day by day as millions of pictures are uploaded on the internet, whether it is on social media or the cloud etc.; with the help of the vast data, these algorithms are making them fast and accurate with deep learning. This technology of classification and localization of the object in an image or live video feed using the YOLOv4 is instrumental in surveillance, security, and traffic control using AI, CCTV, and numerous other applications. HAWKEYE comprehensively includes a variety of essential techniques, such as image processing, pattern recognition, artificial intelligence, and machine learning. It is broadly divided into 3 phases. Phase 1 consists of an end-to-end solution using Tensor Flow to train a custom object-recognition model in Python, putting it into production, and running real-time illustrations in the browser through TensorFlow.js, which will be able to deeply analyze and detect objects in images. Phase 2 consists of an end-to-end result using TensorFlow to train a custom object-recognition model in Python, putting it into production, and running real-time illustrations in the browser through TensorFlow.js, which will be able to deeply analyze and detect objects in video format. Phase 3 demonstrates a method to train a convolutional neural network (CNN) based on multi-class object recognition classifiers and then importing the model to a device based on Android, increasing speed and accuracy. The major problem with the traditional extracting feature models is that they can only determine low-level feature information, such as contour information and texture information, and have limits on obtaining multiple targets under complex scenarios due to poor generalization performance. The second major problem is that even if some of the Deep learning CNN models can extract the detailed texture features from CNN, they lack to deal with the trade-off between accuracy and speed in object detection. Yolov4, on the other hand, helps to counter exactly that.

II. LITERATURE REVIEW

A. Convolutional Neural Network (CNN)

A convolutional neural network (CNN or ConvNet) kind of networking construct for deep science tends to learn straight from a dossier, removing the need for manual feature distillation. CNNs are specifically beneficial for finding patterns in snaps or similar data to understand objects, faces, and complex entities. They can further be pretty potent for classifying non-image dossiers like audio, time series, and signal dossier. Applications that call for object identification and computer vision - like self-propulsive vehicles and face identification techniques - entrust densely to CNNs. A convolutional neural network can have hundreds or thousands of layers that each grasp to discover distinct visage from a figure. Filters are then enforced to each training image at different outcomes, and the yield of each convolved picture is used as the recommendation to the next layer. The filters can start as very easy visage, like brightness and edges, and increase intricacy to visage that exclusively outlines the object. CNN supports an optimal construct for disclosing and learning key physiognomy in illustration and time-sequence dossier. CNN's play a crucial component in applications like Medical Imaging, Audio Processing, Stop Sign Detection, Synthetic Data Generation, etc.

©IJRASET: All Rights are Reserved

1) The Fundamental Structure Of CNN

A typical CNN structure comprises an info layer, the convolutional layer, a pooling layer pursued by a completely associated layer, and a yield layer. The figure underneath demonstrates the typical structure of a CNN.

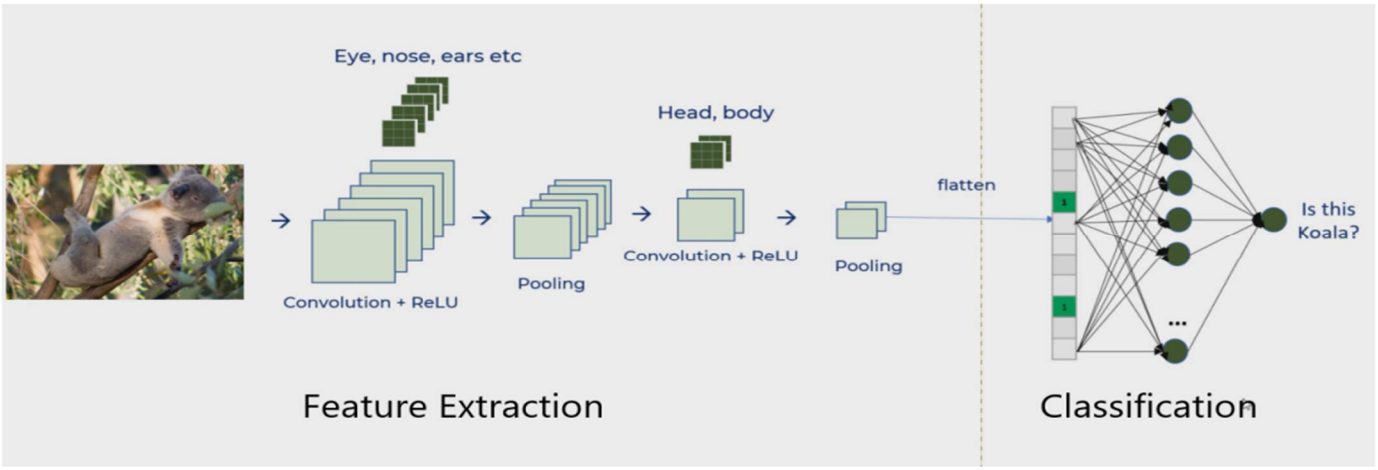


Fig 1. Typical CNN Architecture

a) *Convolutional Layer:* The convolutional layer is the centerpiece of the Convolutional Neural Network. It comprises nearby associations inside the convolutional layer and loads of mutual attributes. The essential point of the convolutional layer is to find out about the component portrayal of the pictures. The convolutional layer comprises a few element maps, including an exact number of neurons. Every neuron of an element map is utilized to extricate nearby qualities of various positions in the previous layer. In request to get another element, the info, including maps, is first convolved with a scholarly part, and afterward, the outcomes are passed into a nonlinear enactment function. Different element maps are gotten by applying distinctive pieces. Sigmoid, tanh, and Relu are the run-of-the-mill actuation capacities.

b) *Pooling Layer:* Convolutional Neural Networks comprise a pooling layer that joins the neurons' yields at one layer into a solitary neuron in the following layer. We can say that it diminishes the elements of the component maps and increment the strength of highlight extraction. The pooling layer is usually situated between two Convolutional layers. The parts resolve the span of the component maps in the pooling layer. Activities of a pooling layer incorporate max pooling and regular pooling. In max pooling, the most potent incentive from every one of the groups of neurons from the previous layer is utilized. While in standard pooling, the regular encouragement from the group of neurons of the past layer is used. Abnormal state attributes of data sources can be acquired by stacking a few Convolutional and pooling layers.

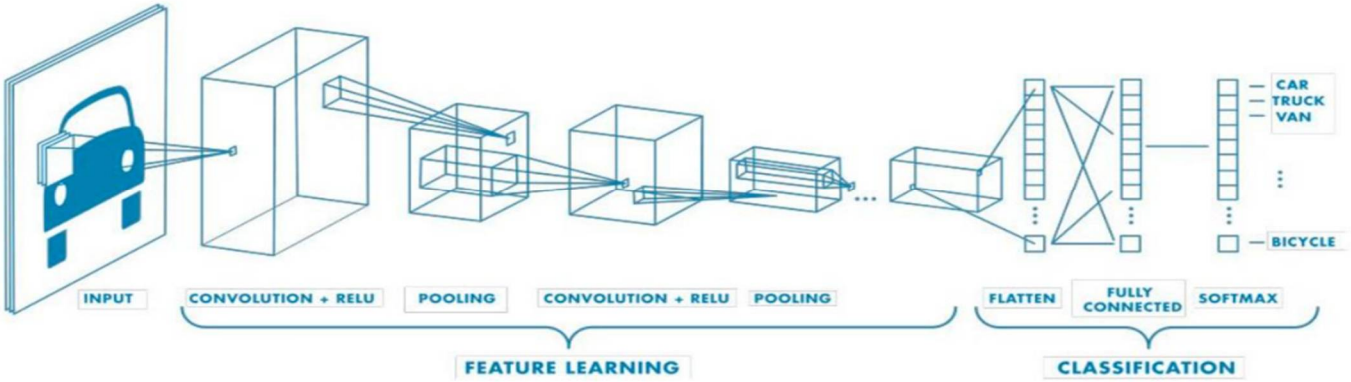


Fig 2. Example of a network with many convolutional layers

c) *Fully Connected Layer:* The completely associated layer in the Convolutional Neural Network takes every one of the neurons from the past layer. It interfaces them to every neuron of the present layer. No spatial data is saved in these completely associated layers. A yield layer constantly trails the last wholly associated layer.

2) *Feature Learning, Layers And Classification:* A CNN derivative of an input layer, an output layer, and many hidden layers like other neural networks. A deep learning model following the connections between the input data, multiple layers, and outputs. All the layers, as mentioned earlier, perform operations capable of altering the data with the intent of learning features specific to the dossier. The most common three layers are convolution, activation or ReLU, and pooling. Convolution puts forward the input images through convolutional filters; each activates certain features from the pictures. Rectified linear unit (ReLU) allows faster and more persuasive training by mapping negative values to zero and upholding positive values. This is occasionally called activation because only the activated features are carried forward into the next layer. Pooling simplifies the output by performing non-linear down sampling, reducing the number of arguments that the network needs to learn. These operations are repeated over hundreds or thousands of layers, with every layer learning to identify different features.
3) *Shared Weights And Biases:* A CNN has neurons accompanying weights and biases like a traditional neural network. The model learns these values amid the training process, and it continually refurbishes them with each new training instance. Although, in the case of CNN's, the weights and bias values are invariable or constant for all unseen neurons in that given layer. This means that all unseen neurons spot the same characteristics, like an edge or a blob, indifferent picture segments. This enacts the network tolerant to the translation of objects in an image. For example, a network trained to perceive jeeps will be able to do so wherever the jeep is present in the picture.
4) *Classification Layers:* The CNN architecture shifts to classification after learning features in many layers, involving the extraction of elements from the image to observe some patterns in the dossier. Using an ANN for image classification would be very costly in terms of computation since the trainable parameters become extremely large. The next-to-last layer is an entirely connected layer with K-dimensions, where K refers to the number of classes forecasted by the given network. The vector, as mentioned, holds the probabilities for every class of any picture being classified. A classification layer like softmax is used to provide the classification output for the final layer of the CNN architecture.
5) *Designing and Training CNNs:* It can be done using MATLAB with Deep Learning Toolbox, enabling you to design, train, and deploy CNNs. MATLAB provides a large set of pre-trained models from the deep learning community that can be used to learn and identify features from a new data set. This transfer learning method is a convenient way to apply deep learning without starting from scratch. Models like GoogLeNet, AlexNet, and Inception provide a starting point to explore the science involved in deep learning, using proven architectures built by experts.
6) *Using Pre-Trained Models For Transfer Learning:* Fine-tuning a pre-trained network with transfer learning is typically faster and more accessible than training the same from scratch. The least possible amount of data and computational resources are required. Transfer learning retrieves the knowledge from one type of problem to resolve similar issues. You start with a pre-trained network and use it to grasp a new task. One of the benefits of transfer learning is that the pre-trained network has already learned a diversified set of characteristics. These characteristics can then be applied to a whole new diversified range of other analogous tasks. For example, you can take a pre-trained network on millions of pictures and re-train it for the next fresh batch of object identification using only hundreds of images.
7) *Hardware Acceleration with GPUs:* A convolutional neural network, or CNN as we all know, is trained on hundreds, thousands, or even millions of images. GPUs can significantly speed up the entire processing time to train a model when working with the enormous amount of dossiers and complex network architects. Thus, the GPU-based CNN accelerator is dominant due to its improved throughput over CPUs.

B. Structure Of The Yolo-V4 Network

1) Introduction—What's YOLO v4?

YOLOv4 is an acronym for You Only Look Once version 4. It's an object detection model used in deep learning use cases. There are mainly two prominent families:

- Two-Stage Detectors
- One-Stage Detectors

YOLO belongs to the family of One-Stage Detectors (You only look once — one-stage detection). The idea of one-stage detection (also referred to as one-shot detection) is that you only look at the image once. In a sliding window + classification approach, you look at the picture and classify it for every window. In a region proposal network, you look at the image in two steps—the first to identify regions where there might be objects and the next to specify them.

YOLOv3 was introduced as an "Incremental Improvement," stating that it was better than YOLOv2, but nothing changed. YOLOv4 was recently introduced as the "Optimal Speed and Accuracy of Object Detection." Let's dig into the YOLOv4. We have four apparent blocks after the input image:

- ✓ Backbone
- ✓ Neck
- ✓ Dense Prediction - used in one-stage-detection algorithms like YOLO, SSD, etc.
- ✓ Sparse Prediction - used in two-stage-detection algorithms like Faster-R-CNN, etc.

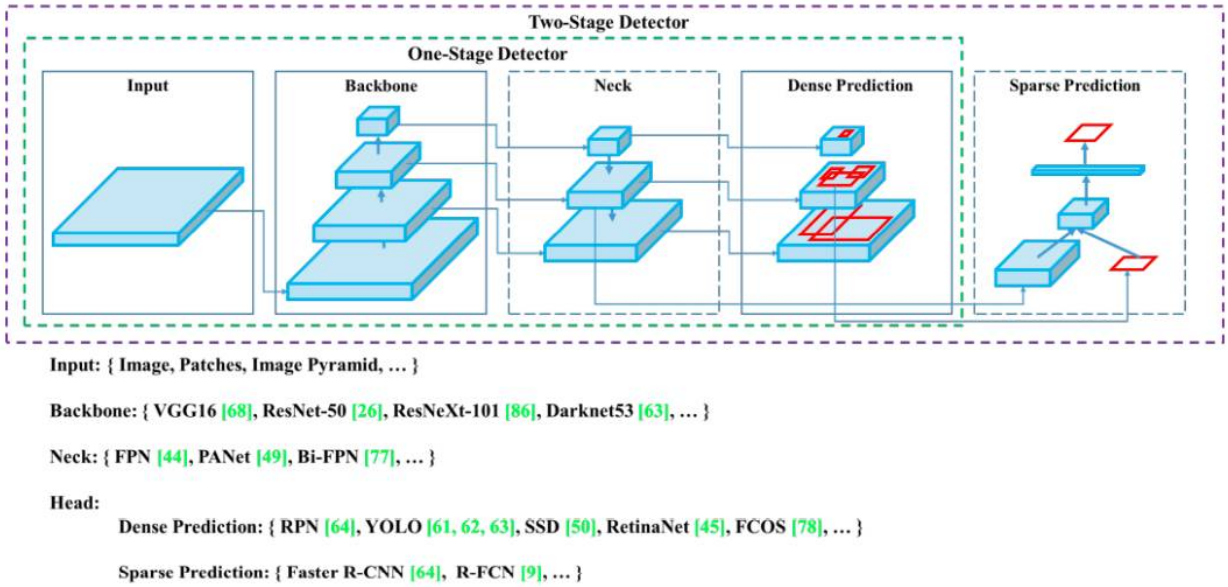


Fig 3. Object Detector

a) Backbone

Backbone here refers to the feature-extraction architecture. You might know it by different names if you're used to YOLO, such as Tiny YOLO or Darknet53. The difference between these is the backbone.

- Tiny YOLO has nine convolutional layers, so it's less precise but much faster and better fitted for mobile and so-called embedded projects.
- Darknet53 has 53 convolutional layers, making it more precise but slower.

Exactly like there are numerous versions of ResNet, there are innumerable versions of YOLO, depending on the backbone. YOLOv4 can have backbones like VGG, ResNet, SpineNet, EfficientNet, ResNeXt, or Darknet53 (the backbone used in YOLOv3). Looking at the paper for YOLOv4, we'll find that the backbone used is not Darknet53 but CSPDarknet53. These letters embossed here are not here to look nice - they mean something crucial. CSP is an acronym for Cross-Stage-Partial connections. The central theme here is to separate the current layer into two parts so that one part will pass through a block of convolutions and another part won't. Then, we aggregate the results. Here's an example with DenseNet:

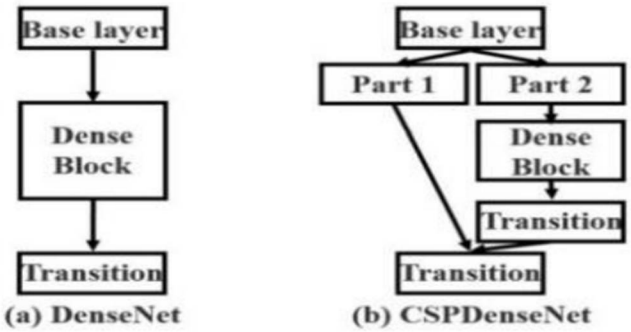


Fig 4. An example with DenseNet

Backbone refers to the extraction of features. You can study them but always keep in mind that YOLO is and will always remain the general architecture and not just the backbone. The authors use CSPDarknet53 for the GPU version and lighter networks for the VPU (Visual Processing Unit) variety—MobileNet, for example.

Table 1. Darknet-53.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

b) Neck

This neck block provides extra layers between the backbone and the head (dense prediction block). We might see different feature maps from the various layers used. Everything was very linear in the early days of convolutional neural networks (CNNs). In more current versions, we have plenty of middle blocks, skip connections, and aggregations of dossiers between these layers. This is a family of techniques known as "parameter or argument aggregation methods."

- We'll utilize a modified interpretation of the PANet (Path Aggregation Network) [Fig 5]. To aggregate information to get higher accuracy is the main objective hidden behind.
- Another technique used is the Spatial Attention Module (SAM) [Fig 6]. Attention mechanisms have been widely used in deep learning, especially in recurrent neural networks. It primarily focuses on a specified part of the input.

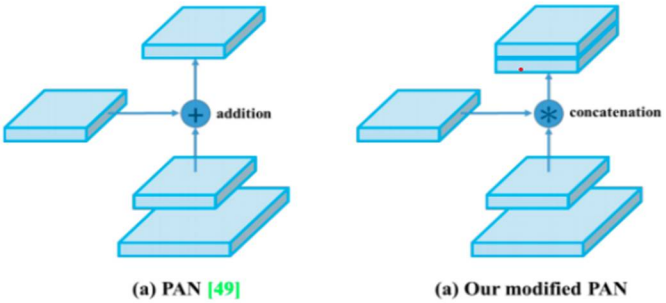


Fig 5. Modified PAN

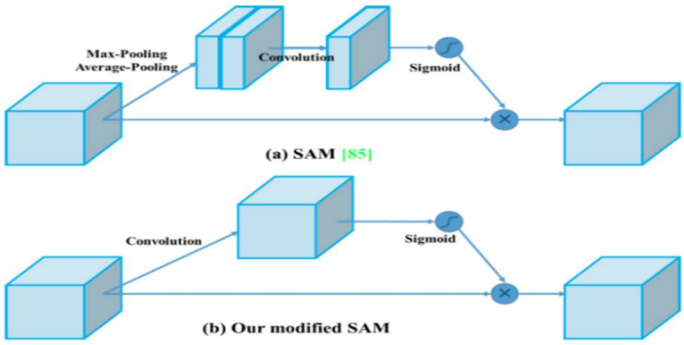


Fig 6. Modified PAN

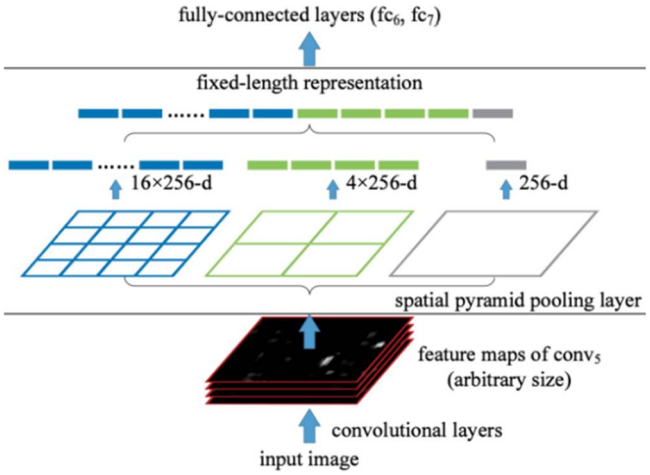


Fig 7. SPP

- Finally, Spatial Pyramid Pooling (SPP) [Fig 7], used in R-CNN networks and numerous other algorithms, is used here.
- Some techniques exist for adding information in a layer, a bit like a ResNet would do. YOLOv4 uses a modified Path Aggregation Network, a modified Spatial Attention Module, and Spatial Pyramid Pooling.

c) Head

The head block is the part used to:

- Locate bounding boxes
- Classify what's inside each box

Here, we have the look-alike process as was in YOLOv3. The network identifies the bounding box coordinates (x,y,w,h) and the confidence score for a given class. This technique is anchor-based. YOLO aims to divide the picture into a grid of numerous cells and then, for each cell, forecast the probability or feasibility of having an object/element using the anchor boxes. The yield is a vector with bounding box coordinates and probability classes. Post-processing techniques such as non-maxima suppression (NMS) are also used.

d) Bag-Of-Freebies (BoF)

Bag-Of-Freebies (BoF) are methods that help amid training without increasing much inference time. Popular methods include dossier augmentation, random cropping, shadowing, dropout, etc. Here's the list of techniques used, in relation with the paper:

- *BoF for the Backbone:* CutMix and Mosaic data augmentation, DropBlock regularization, and Class label smoothing.
- *BoF for the Detector:* CIoU-loss, CmBN, DropBlock regularization, Mosaic data augmentation, self-adversarial training, eliminating grid sensitivity, using multiple anchors for a single ground-truth sample, cosine annealing scheduler, optimizing hyperparameters, random training shapes.

e) Bag-of-Specials (BoS)

Bag-of-Specials (BoS) is another family of methods. Unlike BoF, they change the architect of the given network and augment the inference cost a bit. We have seen the behavior of SAM, PAN, and SPP, which all belong to this family. Here is a complete list from the mentioned paper:

- *BoS for Backbone:* Mish activation, cross-stage partial connections (CSP), multi-input weighted residual connections (MiWRC)
- *BoS for Detector:* Mish activation, SPP-block, SAM-block, PAN path-aggregation block, DIoU-NMS

C. Mish Activation

If we take a look at the YOLOv4 backbone, we may observe that it uses the Mish as an activation function in the backbone. Mish is another activation function with proximity to Rectified Linear Unit. Mish can surpass them in most of the deep networks across different dossiers:

Table 2. Statistical Test Summary

Activation Function	Mean Accuracy	Mean Loss	Standard Deviation (Accuracy)	P-value
Mish	**87.48%**	**4.13%**	**0.3967**	-
Swish	87.32%	4.22%	0.414	0.197
GELU	87.37%	4.339%	0.472	0.4
ReLU	86.66%	4.398%	0.584	<1e-4
ELU	86.41%	4.211%	0.3371	<1e-4
Leaky ReLU	86.85%	4.112%	0.4569	<1e-4
SELU	83.91%	4.831%	0.5995	<1e-4
SoftPlus	83.004%	5.546%	1.4015	<1e-4
ReLU6	86.75%	4.355%	0.4501	<1e-4
SReLU	85.05%	4.541%	0.5826	<1e-4
ISRU	86.85%	4.669%	0.1106	<1e-4
LeCun's Tanh	82.72%	5.322%	0.58256	<1e-4
RReLU	86.87%	4.138%	0.4478	<1e-4
ELisH	87.38%	4.288%	0.47731	0.428

Mish is a novel smooth and non-monotonic neural activation function that can be defined as:

$$f(x) = x \cdot tanh(\varsigma(x))$$

where, $\varsigma(x) = \ln(1 + e^x)$ is the softplus activation function.

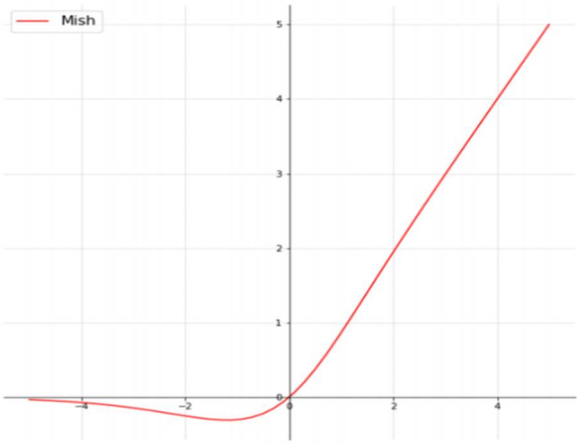


Fig 8. Mish

It turns out that this activation function shows encouraging results. For example, using a Squeeze Excite Network with Mish (on the CIFAR-100 dataset) increased Top-1 test accuracy by 0.494% and 1.671% compared to the same network with Swish and ReLU respectively.

III. METHODOLOGY - USING GOOGLE COLLAB

A. Open CV

Open CV, which means Open source Computer Vision, is a library used for image processing. Image processing is a type of signal processing in which an image is an input, and the output is also an image or set of characteristics related to the picture. Open CV was started as a research project by Intel. It combines different tools to solve the problems of computer vision.

B. Determining the Locations of Objects

After applying the NMS method, there is one bounding box for each detected object in the image. Each bounding box has five values x, y, w, h, and confidence. Coordinates (x,y) is the bounding box's center point. The 416x416 image has a width of 416. Depending on the width of the image and the center point of the bounding box, the object's location can be determined if it is Right, Center, or Left.

C. Custom Dataset (Images Dataset)

For training the deep learning model, a lot of image data is required. The proposed system's prepared dataset consists of 180 labeled images for six objects (Table, Person, TV, Bottle, Chair, and Laptop). There are 30 images for each object. These images have different sizes, and they are in the .jpg format.

D. The Training Process

The Google Collab's GPU was used to train the proposed neural network. The following are the steps for the training process.

1) *Step 1:* A set of high-resolution and colored images of different sizes is collected.
2) *Step 2:* Labelling was used to label each object in the picture. Labeling is an application used for marking the elements in the image.
3) *Step 3:* The prepared dataset is ready to train the neural network. The dataset, consisting of 180 color images, was split into two groups. The first group comprises 85% of the total images as the training images, and the second group includes the remainder, which is 15% of the pictures left as the testing images. The training process was executed on Google Collab, and it took nearly three hours. The neural network was then trained for 3000 iterations.
4) *Step 4:* the weight file is created at the end of the training process.

IV. CONCLUSION AND FUTURE SCOPE

Our YOLOv4 is located in the Pareto optimality curve and is superior to the fastest and most accurate detectors in both speed and accuracy. The paper concentrated on CNNs, the fundamental structure of Convolutional Neural networks, object location dependent on YOLOv4, and the library utilized in executing this task. YOLOv4 is performed with the assistance of the open-source OpenCV library utilizing CNN. We have verified a few features and selected them for use, such as improving the accuracy of both the classifier and the detector. These features can be used as the best course for future studies and development.

The article acknowledges innovation has an assortment of uses; for example, recognizing the deformities in the mechanical production system can likewise be utilized for creating instructive and learning applications. We infer that the offline pattern acknowledgment and classification field has a bright future ahead and trust it keeps building at an expanded pace. The future extension of it can be made available with the downloadable modules thereof representing a particular category/class of images (e.g., Flora and Fauna, Astronomical Images, Sea Creatures, Insectopedia, etc.) and will be able to distinguish them in real-time offline and thereby creating the self-fed loop of constant evolution from newly captured images by the communities across the globe.

V. RESULT

The result of this paper is that we can detect the object in the frame, whether it is a still image or a live video feed using the YOLOv4. It does not affect the detection quality, even if multiple objects are in the frame. It can work as flawlessly as there is one object in the frame. As in today's world, safety is a more critical aspect and this can play an important role in security with some modification as well as it can play a crucial role in n-number of applications like gesture recognition, pose estimation, etc. Below is the speed v/s accuracy table and snapshots for the same.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

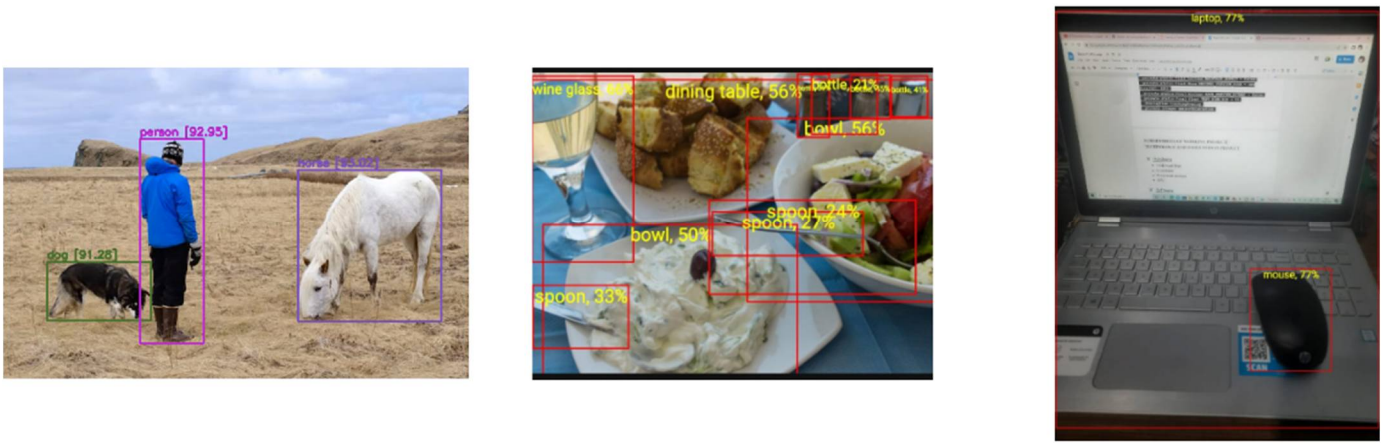


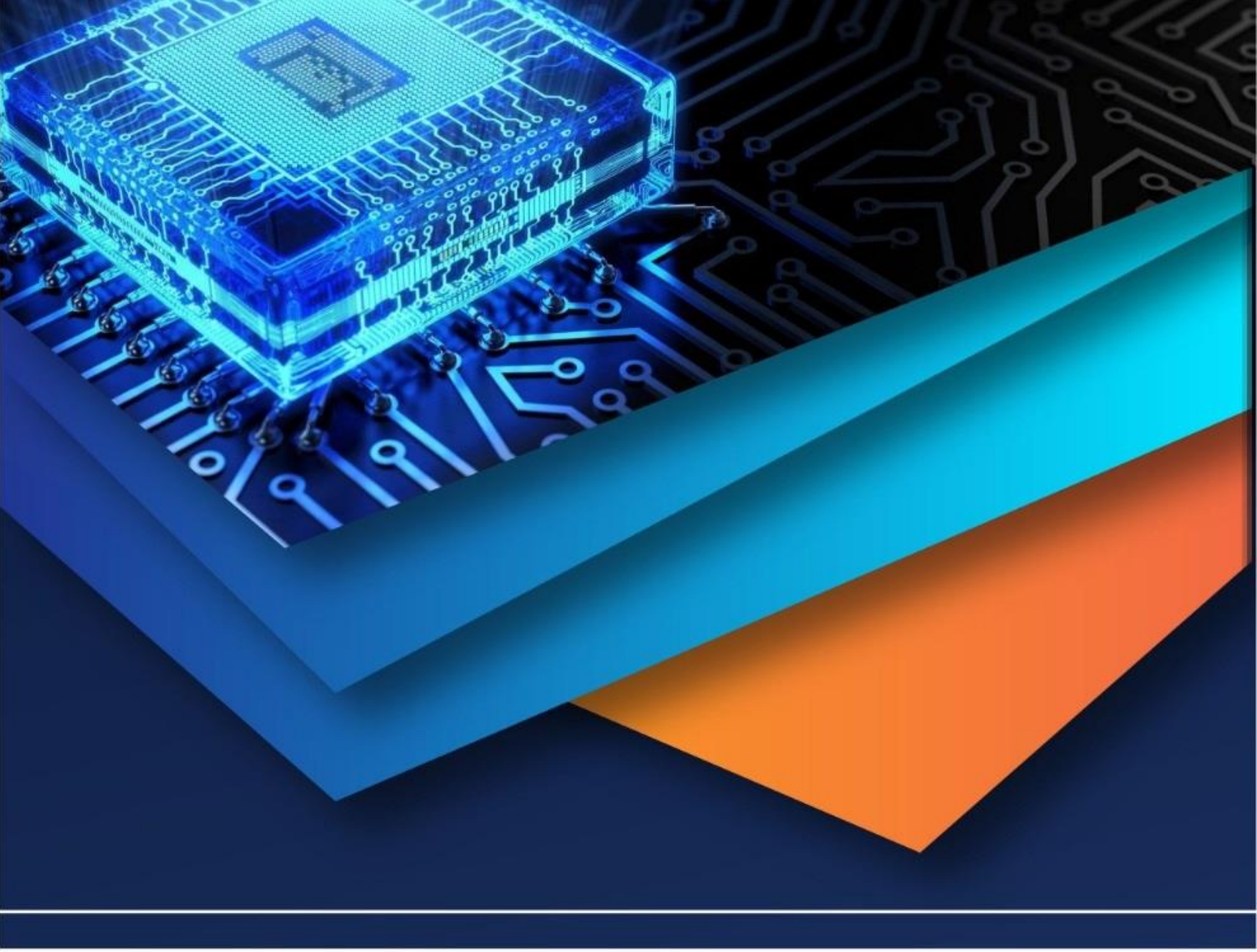
Fig 9. Object Detection using Yolov4

Table 3. Speed and Accuracy in Object Detection.

Method	Backbone	Size	FPS	AP	AP50	AP75	APS	APM	APL
YOLOv4: Optimal Speed and Accuracy of Object Detection									
YOLOv4	CSPDarknet-53	416	38 (M)	41.20%	62.80%	44.30%	20.40%	44.40%	56.00%
YOLOv4	CSPDarknet-53	512	31 (M)	43.00%	64.90%	46.50%	24.30%	46.10%	55.20%
YOLOv4	CSPDarknet-53	608	23 (M)	43.50%	65.70%	47.30%	26.70%	46.70%	53.30%

REFERENCES

[1] Jimin Yu and Wei Zhang. Face Mask Wearing Detection Algorithm Based on Improved YOLO-v4 (2021), Sensors 2021, 21(9) ; https://doi.org/10.3390/s21093263
[2] Chien-Yao Wang, Alexey Bochkovskiy, HongYuan Mark Liao. Scaled-YOLOv4: Scaling Cross Stage Partial Network (2021), Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13029-13038
[3] A Bochkovskiy, CY Wang, HYM Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection (2020), arXiv:2004.10934v1 [cs.CV]; https://doi.org/10.48550/arXiv.2004.10934
[4] R. Phadnis, J. Mishra and S. Bendale.
[5] R. Phadnis, J. Mishra and S. Bendale. Objects Talk - Object Detection and Pattern Tracking Using TensorFlow (2018) - Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 1216-1219, doi: 10.1109/ICICCT.2018.8473331.
[6] Wang Zhiqiang and Liu Jun. A Review of Object Detection Based on Convolutional Neural Network, Proceedings of the 36th Chinese Control Conference July 26-28,(2017)
[7] Nadia Jmour, SehlaZayen, AfefAbdelkrim. Convolutional Neural Networks for image classification (2017)
[8] FatihErtam, Galip Aydin. Data Classification with Deep Learning using Tensorflow (2017), (UBMK'17)2nd International Conference on CSE.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)