International Journal of Scientific Research and Engineering Development-– Volume 5 Issue 6, Nov- Dec 2022

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Real Time Detection of Fruits Health Using Yolo V5 Based on Convolution Neural Network

Rahul Chopra*, Shubhrangi Shandilya**
*(EXTC, NMIMS/Mukesh Patel, and Mumbai
Email: rahulchopra1998@gmail.com)
**(EXTC, NMIMS/Mukesh Patel, and Mumbai
Email: shubhrangi2002@gmail.com)

----------**********----------

Abstract:

Mithara is a Sanskrit term linked to healthy eating habits as it believes that awareness about food affects one's body and mind. The term was coined in the early medieval era but still has its roots deep enough. As is quite eminent from the current situation wherein eating nutritious and maintaining healthy life choices is not an option anymore, our project is a sincere effort to induce these changes into the common man's life in the easiest way possible. Our project not only explains the importance of consuming hygienic fruits and vegetables but also how they would affect one's health over a period of time. To accomplish this we have tried to amalgamate technology with the food sciences and explore the results. We have used an algorithm that goes by the name of Yolo version 5.

This algorithm is highly renowned for image detection as it is very efficient and quick. The main aim of this paper is to identify and determine the health of fruit in real-time and then accurately determine the shelf life of the fruit**.**

Keywords **—Yolo, Yolo V5, object detection, Image processing, Fruit Health, Fruit Damage detection, Real-time, CNN.**

----------**********----------

I. INTRODUCTION

The Covid-19 pandemic has impacted in some way or the other impacted all of our lives, sparing not even the rich and the high classes. It has in some cases done irreversible damage to lives. Even those who were not affected by the disease directly have been negatively impacted in some form or the other. However, it is observed that the impact

of the pandemic is felt differently amongst different sections of society. While some are constantly trying to adapt to working from home, home-schooling their children and ordering food via various online food delivery apps, others have duties that expose them to the virus and help society function smoothly, these people are generally referred to as the front-line workers. However, some are forced

to be exposed to the virus as they do not have access to basic necessities like clean food, water, and shelter. According to a recent article printed by the American news channel, NPR Covid-19 has killed more people than WW-II and the death count continues to pile up with the ever-changing and mutating virus affecting more people by the day. The Covid-19 pandemic has posed a serious challenge to the health infrastructures and cleanliness of countries around the globe. The

world is spending and investing large amounts of money on developing and improving the vaccine's efficiency against the virus, health infrastructures, sanitation, and isolation booths to fight the ever-evolving virus.

In such desperate times, it has become very important to not only consume hygienic foods and fruits but also how they would affect one's health over a period of time. Since the pandemic has restricted the majority of the population to their houses, eating high-calorie foods can lead to a lot of problems like obesity, high blood pressure, and diabetes to name a few. These diseases make one's immune system weak and they become an easy target of the deadly covid virus.

With the ever-evolving technology and increasing processing speeds, Artificial Intelligence (the simulation of human intelligence by machines.) has become the

fastest-growing science in the World of technology and innovation. Many experts today believe that AI could be used to solve major challenges and in crisis situations to solve highly complex problems. Coming back to the present moment, AI and deep learning have already achieved various milestones in fields varying from self-driving cars to landing booster rockets on platforms in oceans and performing highly complex mathematical calculations in the snap of a finger.

In all the applications mentioned above we see that image processing and object recognition play a tremendously important role as this forms the bases on which various algorithms are applied and various calculations are done.

The most important part of deep learning is neurons. A neuron can be understood as a point of interest through which data and computations flow, neurons receive one or more input signals which come from raw datasets or neurons from the previous layers. To make right decisions and accurate predictions one needs to have sufficient data to process and draw conclusions from. This is one of the major challenges that data scientists face while dealing with data that is unfiltered or unclassified as understanding and working on raw data takes a huge amount of time further after data filtering is done one needs to be sure that all the important features have been taken into consideration that can affect the prediction or the outcome. After the data is filtered and labelled it is fed to the algorithm that makes decisions or predictions based on the mathematical logic in the code. Another major challenge that arises in this phase of execution is, pre-processing of the data furthermore, the Laptop/ PC must have the sufficient computing power and powerful processors and in some cases, an additional GPU may be required for complex computations. In this paper, we shall be using various image recognition techniques to identify and determine the health of fruit in real time using nothing but the computer's hardware. In today's time this is done by poking the fruit with a needle and then testing it for the sucrose and sugar content left in the fruit, with these values one can accurately determine the shelf life of the fruit. However, in this paper, we shall be using a few basic features like texture, pattern, color, shape, etc. to determine the fruit and further its shelf life and its current condition using the YOLO algorithm.

The YOLO algorithm is a very powerful image detection algorithm. Its specialty is that it is one of a kind and beats most algorithms in terms of speed. This is major because it detects the entire image in a single forward pass.

II. RELATED WORK

The original idea of YOLO was to develop an algorithm that would divide an input image into a grid. Here each grid would be responsible to detect the object within itself. Hence it is possible that multiple grids could give a positive result of detecting the object within itself [1]. All the Yolo algorithms work on bounding boxes. This bounding box represents five elements: X-coordinate, Y- y-coordinate, Width, Height and the Confidence score. Here the x and y coordinates are the coordinates of the object detected inside the image. These represent the center point of the image. The 'w' represents the width of the object with respect to the x coordinate and 'h' represents the height of the object with respect to the y coordinate. The fifth element that we obtain from the bounding box is the confidence score, the confidence score of the

bounding box tells us the probability of the object present in the box. The probability here shows us the accuracy of the bounding box.

There have been various versions of the YOLO algorithm released till date, starting with the first version that was released in 2016, also known as YoloV1. This network was trained on the ImageNet-1000 dataset. It was the first algorithm to use a single neural network to predict bounding boxes and associate class probabilities. It also demonstrated that image recognition can be viewed as a regression problem rather than a classification problem. YoloV1 was extremely fast and processed real-time images at 45 frames per second. Yolo was the first algorithm to detect an entire image in a single run, instead of using the sliding window technique. By using this technique yolo not only became faster than other models but also highly accurate. This version of yolo had 24 convolutional layers followed by 2 fully connected layers. Joseph Redmon's proposed model could train the network end-to-end when an image was passed through it only once getting the information of the category and the bounding boxes.[2]

The YoloV2 also known as Yolo9000 was a more advanced version of the originally proposed yolo algorithm. This version used Darknet-19 for feature extraction; this architecture has 19 convolutional layers followed by 5 max pool layers; further, a softmax layer was used for the classification of the objects and simultaneously introduced the concept of the anchors. In this version, the anchors gave the size of the objects relative to the final feature maps. In comparison with version one, the second version was a major enhancement considering the fact that the first version only predicted 98 bounding boxes per image whereas version 2 exceeded a thousand. This meant that the recall rate was drastically increased. Further features like batch normalization were also introduced to normalize the input layer by slightly altering the activation values. The resolution was increased from 224 x 224 to 448x 448, this greatly helped in increasing the detection accuracy of smaller or clustered objects. To further increase the precision the image was now divided into a 13x13 grid, which is smaller when compared with version one (smaller grids mean higher precision) [3]

The Yolo V3 was an incremental update to the previous version which uses Darknet-53 for feature extraction, the improvements of which were based upon its predecessor Darknet-19 included the use of residual connections, as well as the addition of more layers. A major enhancement was the use of logistic classifiers for every individual class instead of the traditional softmax layer. This enabled multi-label classification which meant that an object could be given two names and the model would give the probability for both. The anchors were also improved as the sizes of objects in the image get resized to the network size.[4]

Yolo V4 was an improvement on the version three algorithm. This version improved the mean average precision map (map) by 10% and increased the number of frames per second by 12%. The architecture was also changed and included 4 main components namely the input layer, the backbone, the neck, and the dense prediction layer.[5]

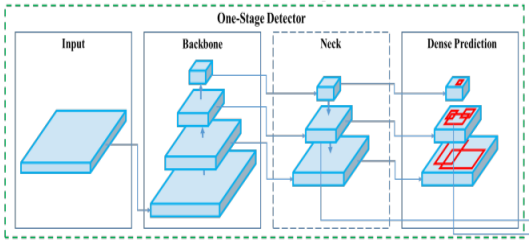

Fig.1 Single Shot Detector

The backbone layer is used for feature extraction. Version 4 used CSPDarknet53 for feature extraction. Models such as ResNet are pre-trained on datasets like ImageNet. These networks produce different features as the number of layers is changed. More layers mean more features. Feature selection plays a vital role in the later stages of image detection.

The neck in the architecture adds extra layers between the head and the backbone. These layers help in the extraction of feature maps at various stages of the backbone.Version 4 used Spatial pyramid pooling (SSP) and a modified version of path aggregation. The SSP module is used as a max

pooling layer over the feature map having different kernel sizes.

The head is the main part of the architecture that is in charge of detections i.e classification and regression which ultimately help in building the bounding boxes. This version of yolo applies the head network to each anchor box. The cost function in the paper is given as

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

hence the points are treated as independent variables however these lose the integrity of the object being detected. [5] To overcome this the IOU loss was proposed which took into consideration the area covered by the predicting bounding box and the actual bounding box or ground truth. Yolo v4 finally settled with the CIou loss for the bounding boxes as it gave better performance compared to others.[6]

The fifth version of yolo was released in the year 2020. This model overcame a lot of shortcomings that were present in the previous versions of yolo. In a recently published paper named yolo for autonomous landing and spot detection in faulty UAVs, it was found that the fifth version of yolo outperforms the previous versions of yolo algorithms in terms of accuracy. Further, they found that the speed of yolo v5 and yolo v4 were identical.[7]

It can be seen that yolo v5 has its roots seated in yolo v3, as both use PyTorch for feature extraction and training procedures, however, most of the architecture resembles yolo v4. In summary, version 5 is the best of version 3 and version 4. Yolov5 is pre-trained on the coco dataset and depends on Ultralytics for training the network. Yolov5 has contributed highly to the translation of the Darknet research to the PyTorch framework.

III. METHODOLOGY

Yolo has developed over a period of time through various versions to become not only fast but highly accurate. The fifth version of yolo makes training the network on a custom dataset with the fifth version of yolo very easy as it has integrated a lot of important libraries in PyTorch.

The fifth version of the yolo algorithm is pre-trained on two datasets: the COCO (common objects in context) dataset, which has more than eighty classes, and Pascal VOC.

The fifth version of yolo has various models that have pre-trained checkpoints on various parameters like speed, FLOPs (floating point operations), and maP val. The names of a few important models that are widely used are YOLO v5n (nano), YOLO v5s (small), YOLO v5m (medium), and YOLO v5l (large). The important point of difference between these models is the number of parameters that they take into consideration while training. [8] The fifth version of yolo is composed of 6 main modules on which the entire architecture is dependent. The modules start from the focus module which transforms space into depth. The space generally refers to the resolution of the image and increases the depth where depth is generally the number of channels. The next is the CBL module which consists of the activation functions such as relu, this is generally present in the hidden layers. The CBL module is followed by the CSP module. The CSP module is a convolutional neural network that forms the backbone for object detection. This is followed by another module that plays a vital role in object detection, this module is called the SPP module or the Spatial Pyramid Pooling module. This module is very important as it removes the fixed size constraint or the fixed resolution requirement of the image, i.e the input can be of any size, but the output obtained after convolutions will always be of the same size. Hence the SPP layer is added on top of the last convolutional layer in yolo. These modules are further followed by the concat and the up-sample modules. Yolo offers 4 network models namely V5s, V5M, V5L, and V5x. The only difference between these models is the depth_multiple and the width_multiple, these two parameters control the number of convolutional cores. [8]

The nano takes 1.9 million, small takes 7.2 million, medium takes 21.2 million, large takes 109.1 parameters into consideration, the common

point is the size of the images (pixels) can be up to 640. [8]

Upon careful comparison of these models and their parameters, we chose yolo v5s for the training of our network. Further, we chose yolo v5s because the CSPNet solves one of the major problems of gradient information being repeated which generally occurs in models that have large-scale backbones, further it integrates the gradient changes into a feature map, and by doing so it decreases the parameters and the required FLOPS (floating-point operations per second) of the model, this ensures the inference speed and accuracy as per the yolo standards and simultaneously reduces the model size.

In live fruit detection speed and accuracy is highly important as the objects are in clusters or in continuous motion due to shivering hands the focus keeps changing. having a compact model size determines its efficiency on devices that have poor camera quality. Further, the Yolov5 model also uses the applied path aggregation network (PANet) [10] as its neck to boost information flow between the layers making it more efficient.

The Yolo layer, generates 3 different feature maps of different sizes (18 × 18, 36 × 36, 72 × 72) this helps in achieving multi-scale [18] predictions, thus the model can handle small, medium, and big objects.[11] Generally, it is observed that the health of a fruit usually starts to deteriorate in a small-scale (black patches or change in texture) to the fruit going completely black or damp (dis-coloration or dis-figuration) having a multi-scale detection model ensures that various changes that occur during the fruit life cycle can be followed easily. These may include changes in color, texture, or even the shape. The architecture of yolo v5 can be understood with the help of the below figure.[8]

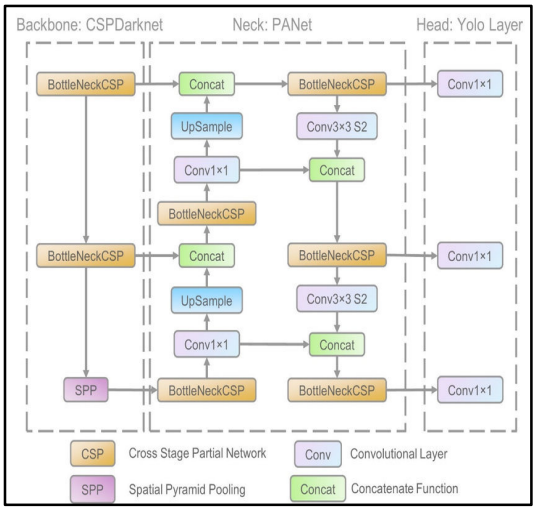

Fig.2 Yolo v3 block diagram

First, the image is fed to the backbone which we consider stage one of the architecture. This layer consists of the BottleNeck Cross Stage Partial Network and the Spatial Pyramid Pooling.

The BottleNeck CSP is used to formulate the image features. It solves the problem of duplicate gradients. Further using the CSP BottleNeck reduces the parameters and results in fewer FLOPS. This is highly important as it increases the speed of detection. Further, the BottleNeck layer reduces the number of feature maps or channels required in the network. If the BottleNeck is not used these feature maps would multiply in each layer or after convolution. The BottleNeck layer uses 1x1 convolutions instead of 3x3 convolutions, thus reducing the output features.

Further, the Spatial Pyramid Pooling (SPP) layer eliminates the requirement of the fixed size input image, which means that the input can be of any size, and the SPP layer will generate a fixed output which would be fed into the further layers. This layer helps in the computation of features from the entire image only once, these features are then pooled to generate a fixed-length output. This is extremely beneficial as it avoids the repetitive computation of the convolved features.

Concatenation is a major part of yolo, the feature maps are taken on two different instances from the BottleNeck CSP and are concatenated. To concatenate the data it must be first upsampled. The

upsampling is controlled by the stride. This can be understood as one pixel for the input having stride = 2 writes it to 4 pixels in the output, hence we can say that one pixel becomes 4 pixels in a dimension of 2x2.[12] Further logic like interpolation is applied to this layer. The dimensions of the final feature map are 1/32 times smaller than the spatial resolution of the input image.

As we know that version 5 of yolo is a combination of the best features of versions three and four, we can see that the BottleNeck CSP is a part of yolo v4 whereas the SPP layer followed by the convolution layers have been taken from version three. However, it must be noted that unlike the other versions of yolo, yolo v5 has no interconnected layers, which makes it not only fast but also suitable to train on models of various sizes.

IV. DATABASE

The first database we considered was a standard database called "Fruits 360". This database was created by putting fruits on a shaft that was connected to a slow-speed motor, this motor was rotated at 3 rpm (rotations per minute). A short movie of 20 seconds was recorded for each fruit. Further this database consists of images that have 100x100 dimensions, there are a total of 90483 images which are further split into train, test, and validation. The size of the training set is 67692 images per fruit, and the size of the testing set is 22688 images per fruit. The validation set consisted of 103 images. There are a total of 131 classes in this dataset [13]. Since our main focus is to deal with fruit health and recognition, we chose a few specific fruits that go through major alterations in shape, size, or color. These fruits included lemon, banana, apple, etc. Since these fruits only had a 360-degree view and didn't have any damage to them, we used this database only to test the performance of the algorithms on low-resolution images and train the model for the identification of these fruits. This was challenging as we were dealing with images that were only 100x100 pixels in size and the YOLO algorithm is majorly made to detect objects in real-time with a much higher resolution.

The second database that we took into consideration is called "Fruits Fresh and rotten for classification". This is a rather simple database that consists of two directories only namely test and train. The testing directory consists of 3 fruits namely apple banana and oranges, these are further classified into fresh and rotten fruits. There is no standard size that is followed in this database however, the maximum size of the images is 500 pixels. This database helped us in studying the various defects that could occur in the fruits from fungal infections in oranges that made them almost white to bananas that went completely black. The examples can be seen below.

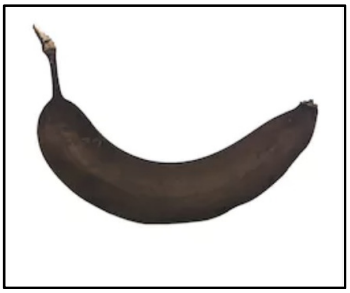

Fig.3 Rotten banana [14]

Fig. 4 Infected orange [14]

It was interesting to note that not all fruits on ripening changed their shape, some only changed their color and some only changed their shape.

We also implemented our own database consisting of 30 images which were taken using a macro camera. These images have a higher resolution compared to the one that we were previously using. We did this to test the accuracy of the algorithm being implemented. The fruit taken into consideration was a banana which was partially

rotten and showed a few defects. An example can be seen below. This database was also used to train the yolo algorithm for live detections as the images fulfilled all the criteria required for live detections.

Fig. 5 Damaged banana (a)

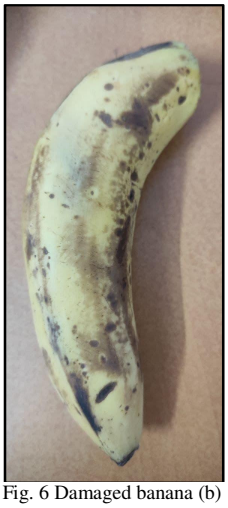

Fig. 6 Damaged banana (b)

V. IMPLEMENTATION AND RESULTS

This project was implemented using the latest version of yolo i.e yolo v5s. The image below is a comparison of human anatomy with that of the SSD (single shot detector). The head forms the first layer in which the image is fed, this layer is also called the input layer. This is the first layer in which the image is fed. The main function of the head is to locate the bounding boxes and perform classifications on the objects. It generally has one stage in yolo. In simpler terms, the head finds the regions where the objects may be present however it fails to tell us which object is present hence in this layer identification is not possible. The single-stage detectors are further subdivided into anchor-free detectors and anchored detectors.

The output from the head is fed into the backbone. This is the most important layer as it helps in the process of feature extraction. The previous versions of yolo have used various versions of the darknet for feature extraction. However, with the advancements in the yolo algorithms, the latest version has switched from the traditional darknet to the latest PyTorch for feature extraction.

The output from the backbone is further fed to the neck. The neck consists of additional layers which are inserted between the head and the backbone. The neck is the fully connected layer which has additional blocks such as SSP, ASPP, and SAM. The reason that the neck is placed after the backbone is that it collects the feature maps from all the stages of the algorithm.

After the feature extraction and feature collection are done the required data is then passed to the dense layer network. This network is composed of neurons. Here every neuron receives input from the neurons that are present in the previous layer. This layer is used to classify the objects or the images based on the input that is received from the convolutions that are done in the previous layers. Further, the dense layer also helps in performing various operations such as scaling, rotation, and changing the vector dimensions. As we have seen above, version 5 of yolo is made by combining the best features from the previous versions.

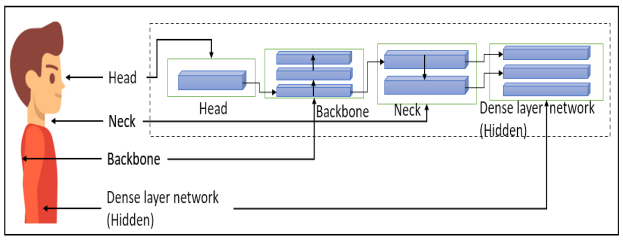

Fig. 7 Single Shot Detector working

Yolo V5 uses only convolutional layers hence it is sometimes referred to as a fully convolutional network. There are 75 convolutional layers in the yolo network. A convolutional layer with stride 2 is

used in the down-sampling process of the feature maps. The down-sampling helps in preventing the losses that may happen to low-level features during the feature pooling process. 1 x1 convolutions are generally used in the yolo network hence the size of the predicted map is exactly the same as the size of the feature map before it.

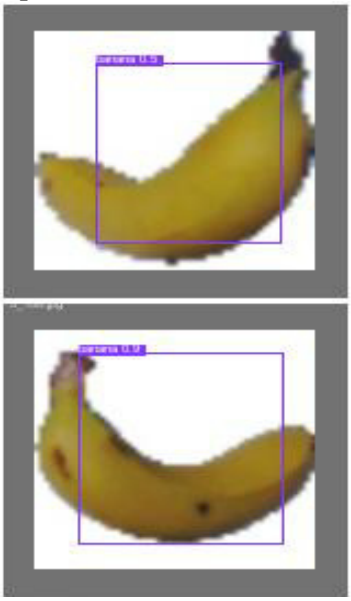

Fig. 8 Banana [Fruits 365]

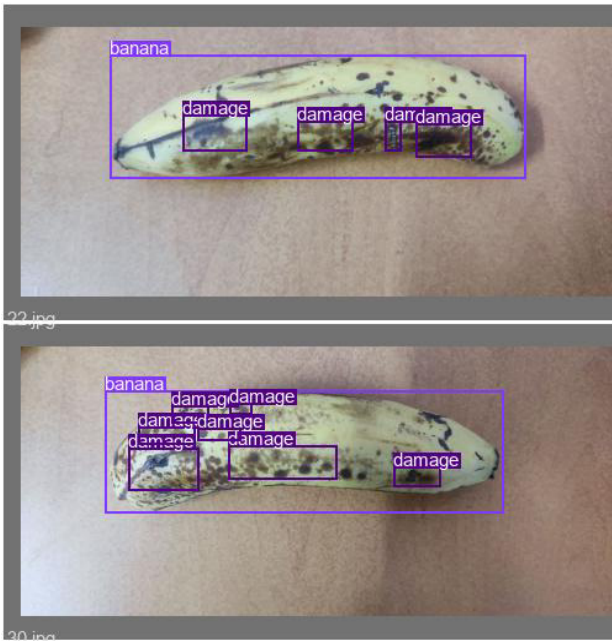

Fig.9 Detected damage
output for the self-database (A)

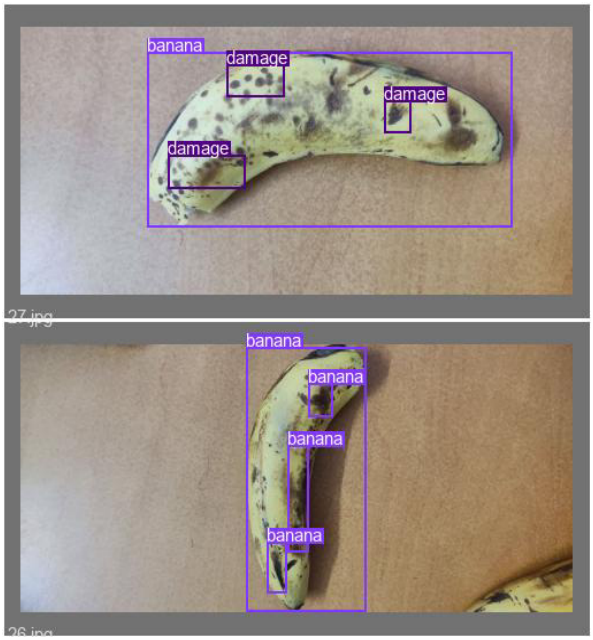

Fig. 10 Detected damage
[output for the self-database (B)]

VI. CONCLUSION

We were successfully able to detect fruits with lower-resolution images as inputs. Further, we also detected damage to the fruits in real-time and with an image as an input. We were also able to test the damages on a standard database called Fruits fresh and rotten fruits. Further, we were also able to detect damages on fruits in real time using our own database. With the help of these detections, we can successfully determine the health of the fruit and thus its shelf life.

VII. FUTURE SCOPE

There has been very little research that has been done on fruit health detection using image recognition. Usually, these tests are done with the help of a glucometer which measures the level of sugar present in the fruit, using this data one can easily determine the shelf life of the fruit. However, determining the health or shelf life of the fruit using object detection techniques still remains a major challenge due to the lack of standard databases and research. In this paper we have considered a few

fruits whose color and shape are easy to distinguish as they progress through their lifecycle, the research can be broadened by studying more fruits and their ripening processes such as pomegranate, litchis, etc.

ACKNOWLEDGMENT

We owe deep gratitude to our mentor Mr. Pradeep Tiwari for providing us with an opportunity to learn and providing us with all the knowledge and guidance required for the proper progression of the project. We are extremely thankful to him for giving his time, clearing our doubts, and guiding us throughout. He has also helped in writing the reports and explaining the format that should be used in the report. His constant encouragement and suggestions helped us achieve our goal. We heartily thank our mentor, Mr. Pradeep Tiwari, for his constant guidance and keen supervision during the entire tenure. This experience will surely help us in the future to work on any project we will be a part of. Further, we would also like to thank our college-Mukesh Patel who allowed us to work on such an interesting topic for two semesters.

REFERENCES

[1] Medium. (2018). Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3. [online] Available at: https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088
https://github.com/krishnaik06/Advanced-CNN-Architectures

[2] Redmon J,DivvalaS,GirshickR,etal.You only look once:Unified,real-time object detection[C]//Proceedings of the IEEEconference on computer vision and pattern recognition.2016:779-788.

[3] Redmon J,Farhadi A.YOLO9000:better,faster,stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition.2017:7263-7271.

[4] Redmon J,Farhadi A.YOLOv3:an incremental improvement[J].arXiv:1804.02767,2018.

[5] Bochkovskiy, Alexey & Wang, Chien-Yao & Liao, Hong-yuan. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection.

[6] UnitBox: An Advanced Object Detection Network https://arxiv.org/abs/1608.01471

[7] Nepal, U.; Eslamiat, H. Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. Sensors 2022, 22, 464. https://doi.org/10.3390/s22020464

[8] https://github.com/ultralytics/yolov5

[9] Xu, Renjie& Lin, Haifeng & Lu, Kangjie& Cao, Lin & Liu, Yunfei. (2021). A Forest Fire Detection System Based on Ensemble Learning. Forests. 12. 217. 10.3390/f12020217. https://www.researchgate.net/figure/The-network-architecture-of-Yolov5-It-consists-of-three-parts-1-Backbone-CSPDarknet_fig1_349299852

[10] Wang, K.; Liew, J.H.; Zou, Y.; Zhou, D.; Feng, J. Panet: Few-shot image semantic segmentation with prototype alignment. InProceedings of the IEEE International Conference on Computer Vision (ICCV 2019), Seoul, Korea, 20–26 October 2019; pp. 9197–9206.

[11] Fernandes, A.M.; Utkin, A.B.; Lavrov, A.V.; Vilar, R.M. Development of neural network committee machines for automaticforest fire detection using lidar. Pattern Recognit. 2004, 37, 2039–2047, doi:10.1016/j.patcog.2004.04.002

[12] https://github.com/pjreddie/darknet/blob/f6d861736038da22c9eb0739dca84003c5a5e275/src/blas.c#L334

[13] Mihai Oltean, Fruits 360 dataset: new research directions, Technical report, 2021.

[14] Database: https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification

