e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

FOOD DETECTION WITH YOLOV3

Swaraj Bhopatkar*1

*1Department of Information Technology B.K. Birla College of Arts, Commerce and Science (Autonomous) Kalyan, India.

ABSTRACT

This mission explores food detection using YOLOv3, the (state of the art and exceptional) set of PC commands for object detection. We're great at following YOLOv3 on (many exclusive varieties of people or matters) food set to (in a way that's close to reality or a real number) get awareness of unique food gadgets in photos and movies. The resulting model offers realistic uses in dietary view/surveillance, menu analysis, and meal protection (careful examination of something), making it a valuable beneficial component/asset for many companies.

I. INTRODUCTION

The obsession with observing snacks and meals has been a fun subject for experimenting with current machine learning techniques. The maximum goal could be a computer imaginative and prescient machine that could perform unusual ingredient sorting and location in real-time, which can be an IoT tool deployed on the AI aspect for a lot of food packages. With the modern improvement of convolutional neural network (CNN), the accuracy of photo classification has increased by leaps and bounds in recent years (in 2014). In many cases, AI can understand objects higher than a professional human. As a food detection technologist, deep learning is the destiny of food tracking.

Food detection serves an essential function in specific areas, including nutritional monitoring/surveillance, restaurant menu evaluation, and food safety. In this project, we explore (the location of) the use of YOLOv3, a (currently best available design) command set for an object detection computer for the unique activity of food detection. YOLOv3 (You Only Look Once version 3) is well known for its (running or visible at once, without any delay) performance and high (satisfying that it is very close to the truth or actual range) in object detection, making it an ideal wish for this software. YOLO takes a very unique approach. It applies one neural community to the overall photo. This community divides the image into regions and predicts bounding boxes and possibilities for each neighborhood. These bounding containers are weighted using expected probabilities. The interested reader should review the authentic "Look Only Once: Unified Real-Time Object Detection" [1].

YOLOv3 (You Only Look Once, Version 3) is a (running or visible immediately, no delay) object detection instruction set for laptop that identifies real spoken/specific objects in movies, live streams, photos. The YOLO laptop command set familiarization system uses features detected through a deep convolutional network associated with the nerves/mind to hit the object. Versions 1-3 of YOLO were created by means of Joseph Redmon and Ali Farhadi, and the 1/3 model of the YOLO system that acquires computer command set knowledge is a more (really very close or really wide) model of the original ML laptop command set. The first version of YOLO was created in 2016, and the model 3, which is much talked about in this newsletter, will be created two years later in 2018. YOLOv3 is an advanced model of YOLO and YOLOv2. YOLO is used using Keras or OpenCV deep learning libraries.

As traditionally with object detectors, the capabilities obtained using convolutional layers are superimposed on a classifier that enables detection (assertions about likely future events). In YOLO, (the claim about possible fate opportunities) is based on a convolutional layer that uses some mysterious twists. The YOLO computer instruction set is known as "the most effective look as soon as" due to the fact that its (claim approximately likely future occasion) uses some confusing twists; which means that the size of the map (an announcement of the approximately likely future opportunity) is exactly the scale of the characteristic map in front of it.

II. WORKING OF YOLOV3

YOLO is a Convolutional Neural/CNN for performing item detection (runs or can be viewed all at once, without delay). CNNs are primarily classifier-based systems that can process input images as structured series of records and capture patterns among them (see photo below). YOLO has the advantage of being much faster

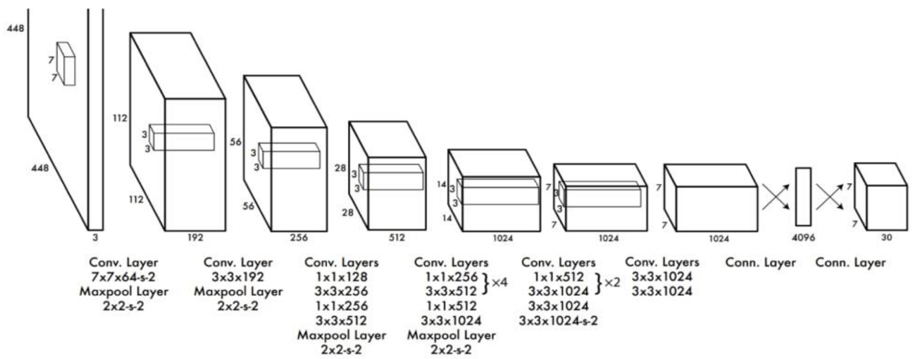
e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

than other networks and still maintaining itself (satisfying because it is very close to reality or the correct range). It enables the model to study the whole picture looking at time so that (statements of approximately feasible destiny activities) are (based on knowledge and knowledge acquisition) using the global big picture within the picture. There are big differences between YOLOv3 and older versions in terms of speed, (high) quality and level of class detail. YOLOv2 and YOLOv3 are worlds apart in terms of (quality very close to the truth or true number), speed, and (related to beautiful design and construction of buildings, etc.). YOLOv2 was released in 2016, two years before YOLO v3.

YOLO-V3 (related to the beautiful design and construction of buildings, etc.) Due to the great ideas from ResNet and FPN (Feature-Pyramid Network) (related to the beautiful design and construction of buildings, etc.), the YOLO-V3 feature is an extractor, called Darknet-53 (has 52 confusing twists) contains jump connections (like ResNet) and 3 (statements of a possible future event) heads (like FPN) - each processes an image on another (relative to space or existing in space) (compress or force into a smaller space) ion.

III. ARCHITECTURE OF YOLOV3

The YOLO computer instruction set takes an image as input and then uses a simple neural/brain-related deep convolutional network to detect objects in the image. (related to the beautiful design and construction of buildings, etc.) of the CNN model, which forms the most important part of YOLO, is shown below.



The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection.

Fig 1. The YOLO architecture

The first 20 convolution layers of the model are pre-trained with ImageNet by concatenating (short-lived only) average combinational and fully connected layers. Then this pre-trained model is transferred to perform detection because previous research has shown people that adding convolution and connected layers to a pre-trained network improves performance. The last fully connected layer YOLOs (describes a possible future event) both the class chance and bounding box coordinates.

YOLO divides the input image into an S X S grid. If the center of an object falls within a grid cell, that grid cell is responsible for detecting that object. Each grid cell (describes a possible future event) B bounding boxes and a confidence score for those boxes. These confidence scores reflect how certain the model is that the box contains the object and how (very close to the truth or actual number) it thinks the box is (a possible future event described).

YOLO (describes a possible future event) multiple bounding boxes per grid cell. At training time, we only want one bounding box (describe a possible future event) or to be responsible for each object. YOLO will assign one to (describe a possible future event) or be "responsible" for (describe a possible future event) to the object based on which (statement of a possible future event) has the highest current IOU. This leads to (focusing on doing one thing very well) between the bounding box (describe a possible future event) or Every (describe a possible future event) or gets better at predicting certain sizes (width:height, for example 4:3) or classes of objects, thus improving the overall recall score.

One of the key ways of doing things used in YOLO models is Non-Maximum Stop/Prevent (Action or Feeling) (NMS). NMS is a post-processing step used to improve (the quality of being very close to the truth or real number) and (very little waste in working or making something) object detection. In object detection, it is common to create multiple bounding boxes for a single object in an image. These bounding boxes may overlap or be placed in different positions, but they all represent the same object. NMS is used to identify and remove unnecessary or incorrect bounding boxes and to output one bounding box for each object in the image.YOLO v3 is the third version of the YOLO object detection computer instruction set. It was introduced in 2018 as an improvement over YOLO v2 to increase (very close to true or true number quality) the speed of the computer's instruction set.

One of the major improvements of YOLO v3 is the use of a new CNN (related to beautiful design and construction of buildings, etc.) called Darknet-53. Darknet-53 is a version of ResNet (related to the beautiful design and construction of buildings, etc.) and is intended for object detection tasks. It has 53 convolutional layers and can (with effort to achieve or obtain) (best design available now) results on various object detection test results. Another improvement in YOLO v3 is anchor boxes with different scales and (width:height, like 4:3) s. In YOLO v2, the anchor boxes were all the same size, which limited the ability of the computer instruction set to detect objects of different sizes and shapes. In YOLO v3, the boxes (Ship Attachment/Support Source and Security/TV Reporter) are reduced in size and resized (width:height, for example 4:3) to better match the size and shape of detected objects.

YOLO v3 also introduces the idea of "Function Pyramid Networks" (FPN). FPNs are CNNs (related to the beautiful design and construction of buildings, etc.) used to detect objects at many scales. It builds a pyramid of feature maps, with each level of the pyramid used to detect objects at a different scale. This helps improve the performance of small object detection because the model can see objects at many scales. In addition to these improvements, YOLO v3 can handle a wider range of object sizes and is more stable than previous versions of YOLO.

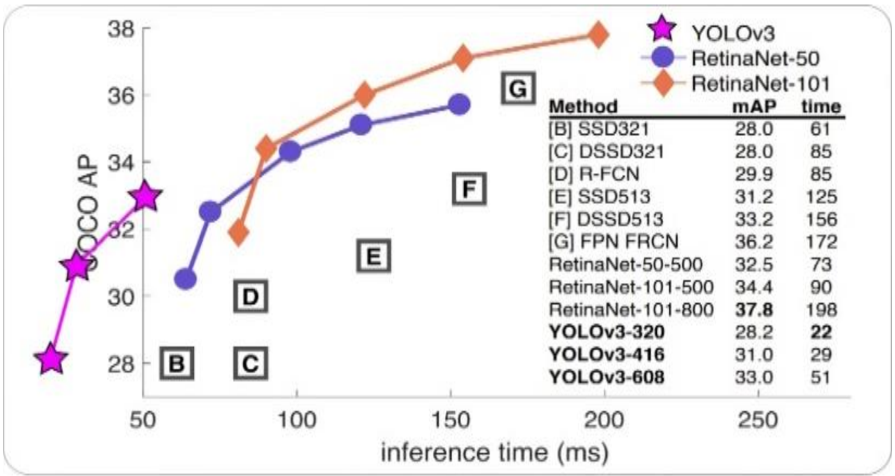


Fig 2. Results obtained by YOLOv3

IV. RESULT

In YOLOv3, Food Watcher will become the most advanced AI that can recognize common food (in progress or visible instantly, without any delay). AI will show more compassion for the human needs of these beautiful carbohydrate compounds (aka. foods).

Darknet YOLO training requires the following to function

- Bounding box file for each image
- Class name file for all category names
- Training data set for list of training images
- Dataset file validation for list of validation images
- Configuration file to specify the YOLO neural network
- Data location file to find all data information

Experimental materials, design and methods

• Data Collection

Pictures of foods and fruits which were easily available in our home were collected. 50 images were collected The images were taken with a mobile camera. The remaining images were obtained from various internet sources.

• Import and install libraries

Some popular Python libraries have been imported for matrix operations, rendering, and file manipulation

➢ Numpy: NumPy data formats include matrix and multidimensional arrays. NumPy can perform mathematical operations on arrays, such as statistical, algebraic, and trigonometric routines. It provides highly functional multidimensional arrays and even the necessary tools for computing and regulating arrays[2]. It is used in food detection with YOLOv3 to process the image data and perform the necessary field manipulations.

➢ Matplotlib: Matplotlib is a plotting library that can be useful for displaying (in your mind) the results of your food detection, such as displaying images with bounding boxes around detected foods. It is used to display the result of object detection, which involves drawing construction boxes around the food image and displaying the process image.

➢ TensorFlow: TensorFlow is a deep learning framework. YOLOv3 models are implemented in Darknet (another framework written in C/CUDA).

➢ Pandas: Panda is used for data manipulation or data analysis. It is a popular python library for data manipulation. It is not directly involved in food detection. It is used to analyze, organize and subsequently process the results of food detection.

➢ OpenCV: OpenCV is a computer vision library that provides various image and video processing functions. This library is used in food detection for image loading before and after processing.

When using YOLOv3 for food detection, the results typically involve identifying and locating different food items in an image or video.

➢ Object detection: YOLOv3 is able to detect multiple food items in an image or video (simultaneously). It can identify different types of food such as fruits, vegetables, fast food and more.

➢ Bounding boxes: YOLOv3 provides bounding boxes around each detected food. These boxes point to/show the location of the food in the image, making it easier to understand where each item is.

➢ Class labels: YOLOv3 also assigns class labels to each detected food item. These labels identify what type of food it is, such as "apple", "pizza", "hamburger", etc.

➢ Confidence Score: For each detection, YOLOv3 assigns a confidence score that represents how reliable the set of computer instructions (quality is very close to the truth or real number) of the detection. A higher confidence score usually indicates a more reliable detection.

➢ Real-time detection: YOLOv3 is known for its object detection capabilities (instant or displayable immediately, without any delay), which means it can quickly process images or video images and provide the results of food detection in the vicinity (run or display immediately, without any delay). delay).), making it suitable for applications such as monitoring/surveillance (self-service restaurant) of food selection or careful study of food intake in a video stream.

➢ Multiple classes: YOLOv3 can detect a wide range of food classes thanks to its pre-trained model or fine-tuning capabilities. Common food classes include hamburgers, pizzas, fruits, vegetables, and more. Explicit/specific classes and their (quality very close to the truth or real number) may change/vary depending on the training data and model settings.

➢ Localization: YOLOv3 is also effective in localizing food (in a way that is close to the truth or real number). It can accurately outline the shape and position of each food item in a picture.

➢ Customization: YOLOv3 can be tuned on custom food datasets, allowing you to train the model to recognize clearly stated/specific food types that are clearly associated or related to your application or research.

Using YOLOv3 for real-time food detection gives output like:

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

Fig 3. Food detection using YOLOv3 results

V. CONCLUSION

In this paper, the YOLOv3 algorithm is used to detect different foods. The YOLOv3 food detection project has (shown/demonstrated or demonstrated) (possible strength or capability within/capability) computer vision and deep learning methods in solving real-world food recognition and classification problems. The food detection project using YOLOv3 has shown humans (perhaps the power or capability within/possibility) of deep learning in solving real-world food recognition problems. As technology continues to advance, the application and (quality very close to the truth or actual number) of these models are expected to improve, making them valuable tools (helping to increase/show in a good way) healthier eating habits and (faster and more efficient) various processes food related.

VI. REFERENCES

[1] Un Liu and Xuewei Wang, "Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network", Facility Horticulture Laboratory of Universities in Shandong, Weifang University of Science and Technology, Weifang, China

[2] Fahad Jubayer, Janibul Alam Soeb, Abu Naser Mojumder, Mitun Kanti P, Pranta Barua, Shahidullah Kayshar, Syeda Sabrina Akter, Mizanur Rahman, Amirul Islam, "Detection of mold on the food surface using YOLOv5", Elsevier B.V. This is an open access article under the CC BY-NC-ND license Current Research in Food Science 4 (2021) 724–728

[3] Liquan Zhao, Shauiyang Li, " Object Detection Algorithm based On Improved YOLO v3", Electronics 2020, 9, 537

[4] Juan du, "understanding of object detection based on cnn family and yolo", iop publishing iop conf. Series: journal of physics: conf. Series 1004 (2018) 012029

[5] Kazim Raza1, Song Hong, "Fast and Accurate Fish Detection Design with Improved YOLO-v3 Model and Transfer Learning", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 2, 2020