Image Based Calorie Prediction using Yolov5

Submitted in partial fulfillment of the requirements
for the award of Bachelor of Engineering
degreeinComputer Science and Engineering.

By

SOLLETI SIVASAI (38110540)
GOPISETTY SAIPAVAN (38110170)

DEPARTMENT OF COMPUTER SCINECE

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY – 2022

ABSTRACT

For the last few decades, it has been the popular trend that people are putting more attention on improving their healthiness and regulating calorie intake for every meal, so that we build a model for calorie estimation of different food. In order to express our concerns on this issue, and with our great interests, we used object detection to estimate the calories count of some famous dishes. Based on the various food images collected by scrapping and previously defined calorie info, we built a image-based calorie prediction model, which can be accurately identify the name of the foods and provide their calorie intake, and finally offer meals plan advice for different groups of people. To identify the dish, we used the Yolo-V5(You only look Once) for real-time processing of object detection and classification. We used an annotator tool called “makesense.ai” to manually label our dishes with their respective original dish names. Using our proposed model, users can easily calculate the calorie intake of their desired foods by just simply clicking photos, which saves a lot of time

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1INTRODUCTION TO DEEP LEARNING

Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones. In human brain approximately 100 billion neurons all together this is a picture of an individual neuron and each neuron is connected through thousands of their neighboring neurons present in the brain. The question here is how we recreate these neurons in a computer. So, we create an artificial structure called an artificial neural net where we have nodes or neurons. We have some neurons for input value and some for-output value and in between, there may be lots of neurons interconnected in the hidden layer.

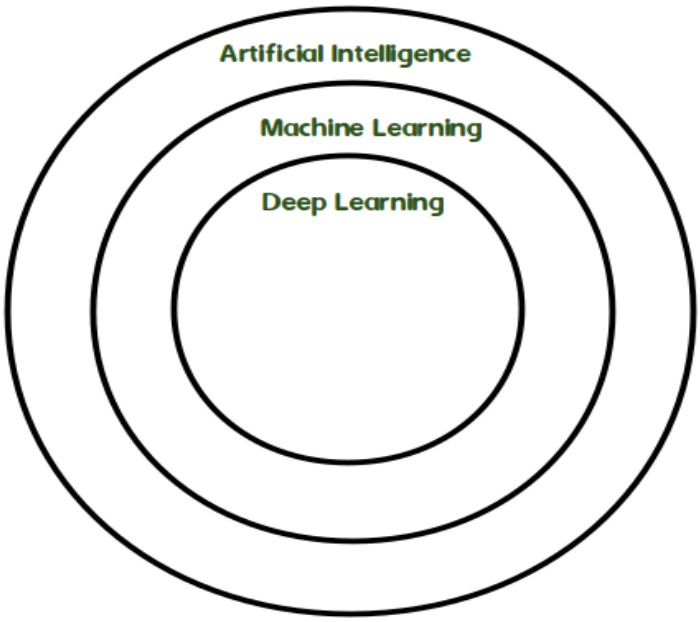


FIG:1.1 About deep learning

The key reason in the success of neural networks is growth by faster computers with larger memory. After having availability of larger datasets, another hurdle faced by researchers was how to process and store this amount of data. But with

today’s faster CPUs and GPUs and also with larger memory we have resources to work on larger datasets. The researchers now able to perform experiments on a very large scale and day by day new algorithms, concepts and results are found. Deep learning improved its ability in providing more accurate results and because of this its use is also increasing in real life too. Since researchers are now able to achieve almost human like results in tasks like voice recognition, object detection, image recognition etc., and many major IT companies now using deep learning for their real-world products.

Deep learning has applications since 1990s but at that many researchers refused to use it because to make it work perfectly and for better results one need a large dataset which can be fed to the network so that hidden layers can extract every abstract feature from it. But this all started to change by increasing digitization of society, more and more activities take place on computers, increase in computers connected together in networks and so on. The age of “Big Data” has made the implementation of deep learning much easier and effective**.**Deep learning is an approach to machine learning that has drawn heavily on our knowledge of human brain, applied maths and statistics. In recent years deep learning has seen tremendous growth in its popularity and usefulness.

Recently YOLO (You Only Look Once) becomes one of the most appealing approaches and has been a crucial factor in the variety of recent success and challenging Deep Learning applications such as object detection, and face recognition. Therefore, YOLO-V5 is considered as our main model for our challenging tasks of image classification. Specifically, it is used for is one of high research and business transactions. Food recognition and calories prediction application is used in different tasks of our real-life time purposes. Precisely, it willhelp in following proper diet and controlling the intake of calories.

1.2 OBJECT DECTECTION

FASTER R-CNN

Our object detection system, called Faster R-CNN, iscomposed of two modules. The first module is a deepfully convolutional network that proposes regions,and the second module is the Fast R-CNN detector that uses the proposedregions. The entire system is asingle, unified network for object detection Using the recently popular terminology of neuralnetworks with ‘attention’ mechanisms, the RPNmodule tells the Fast R-CNN module where to look.A Region Proposal Network (RPN) takes an image(of any size) as input and outputs a set of rectangularobject proposals, each with an objectless score.

This architecture is naturally implemented with an n×n convolutional layer followed

Object detection results (%) on the **MS COCO** dataset. The model is VGG-16.

method	proposals	training data	COCO val		COCO test-dev	
			mAP@.5	mAP@[.5, .95]	mAP@.5	mAP@[.5, .95]
Fast R-CNN [2]	SS, 2000	COCO train	-	-	35.9	19.7
Fast R-CNN [impl. in this paper]	SS, 2000	COCO train	38.6	18.9	39.3	19.3
Faster R-CNN	RPN, 300	COCO train	41.5	21.2	42.1	21.5
Faster R-CNN	RPN, 300	COCO trainval	-	-	**42.7**	**21.9**

Fig 1.2 Accuracy Table

SSD

The SSD approach is based on a feed-forward convolutional network that produces

a fixed-size collection of bounding boxes and scores for the presence of object class

instances in those boxes, followed by a non-maximum suppression step to produce the

final detections. The early network layers are based on a standard architecture used for

high quality image classification (truncated before any classification layers), which we

will call the base network2

. We then add auxiliary structure to the network to produce

detections with the following key features:

Multi-scale feature maps for detection We add convolutional feature layers to the end

of the truncated base network. These layers decrease in size progressively and allow

predictions of detections at multiple scales. The convolutional model for predicting

detections is different for each feature layer (cfOverfeat[4] and YOLO[5] that operate

on a single scale feature map).

Convolutional predictors for detection Each added feature layer (or optionally an existing feature layer from the base network) can produce a fixed set of detection predictions using a set of convolutional filters. These are indicated on top of the SSD network

architecture in Fig. 2. For a feature layer of size m × n with p channels, the basic element for predicting parameters of a potential detection is a 3 × 3 × p small kernel

that produces either a score for a category, or a shape offset relative to the default box

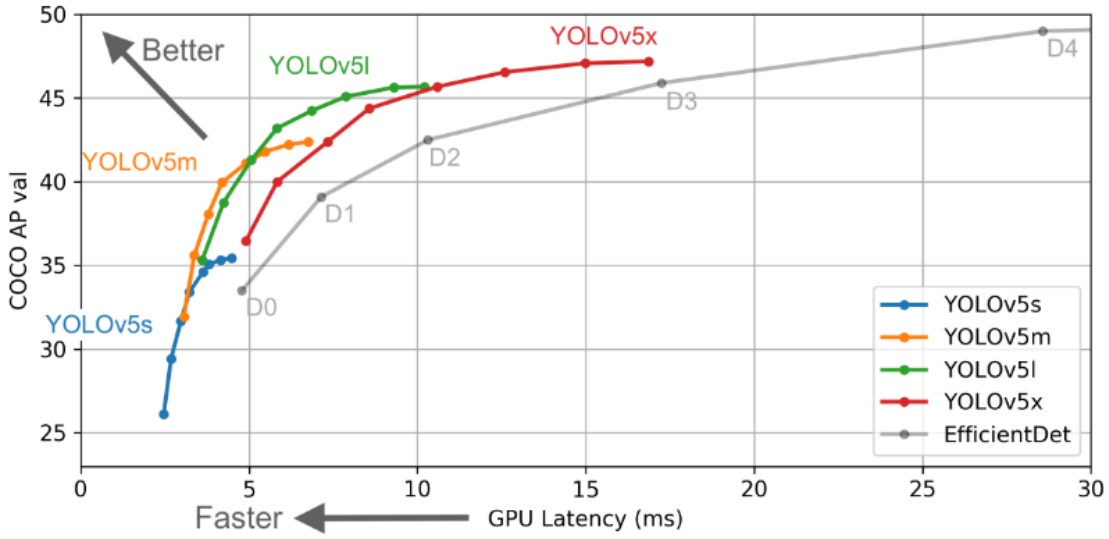
coordinates. At each of the m × n locations where the kernel is applied, it produces an

output value. The bounding box offset output values are measured relative to a defaultM

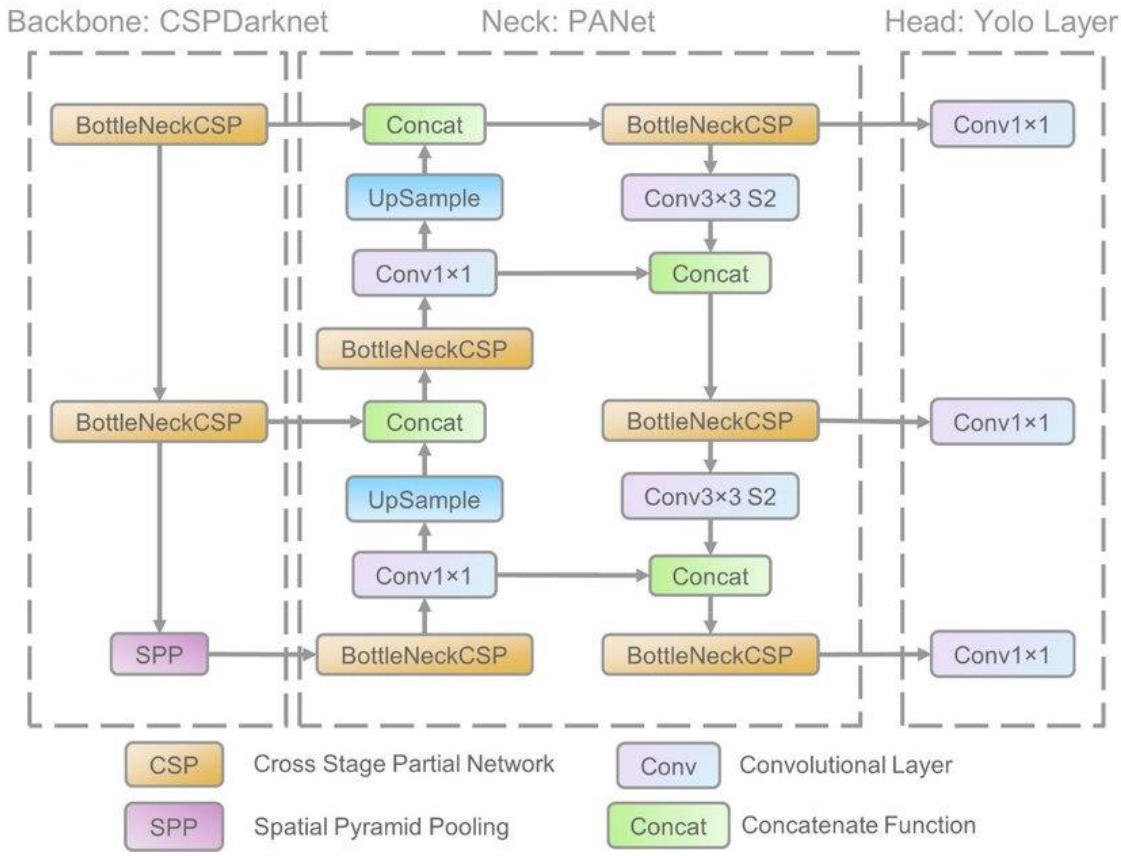
1.3 INTRODUCTION TO YOLO-V5

The deep learning community is abuzz with YOLO v5. YOLO (You Only Look Once) is a family of models that ("PJ Reddie") Joseph Redmon originally coined with a 2016 publication. YOLO models are infamous for being highly performant yet incredibly small – making them ideal candidates for real-time conditions and on-device deployment environments. This immediately generated significant discussions across Hacker News, Reddit and even GitHub but not for its inference

speed. The YOLOv5 implementation has been done in Pytorch in contrast with the previous developments that used the DarkNet framework. This makes it easier to understand, train with it and deploy this model. There is no paper released with YOLO-v5. The release of YOLOv5 includes five different model's sizes like YOLOv5s (smallest), YOLOv5m, YOLOv5l, YOLOv5x (largest). The inference speed and mean average precision (mAP) for these models is shared below:



1.2.1 YOLO-V5 ARCHITECTURE:



1. Backbone: Model Backbone is mostly used to extract key features from an input image. CSP (Cross Stage Partial Networks) are used as a backbone in YOLO v5 to extract rich in useful characteristics from an input image.

2. Neck: The Model Neck is mostly used to create feature pyramids. Feature pyramids aid models in generalizing successfully when it comes to object scaling. It aids in the identification of the same object in various sizes and scales. Feature pyramids are quite beneficial in assisting models to perform effectively on previously unseen data. Other models, such as FPN, BiFPN, and PANet, use various sorts of feature pyramid approaches. PANet is used as a neck in YOLO v5 to get feature pyramids.

3. Head: The model Head is mostly responsible for the final detection step. It uses anchor boxes to construct final output vectors with class probabilities, objectness scores, and bounding boxes.