Object Detection Using YOLO

Sagar Srujan Somepalli, Vamsi Kishore Nallagopu, Ankitha Devi Gangavarapu, Sai Chaitanya Reddy Tiyyagura, Praveen Chowdary Vemasani
Thummala Sumanth Kumar
JNTU Anantpur

Abstract

Computer vision has a lot of interesting applications and object detection is one of the most interesting application. With the advance computer vision techniques, the objects present in the images can be identified in seconds with great accuracy. Hundreds of images can be processed in a few minutes to detect objects in those images. There are many algorithms available now through which this object detection can be performed very fastly. YOLO is one of these popular object detection methods.

In more traditional ML-based approaches, computer vision techniques are used to look at various features of an image, such as the color histogram or edges, to identify groups of pixels that may belong to an object. These features are then fed into a regression model that predicts the location of the object along with its label.

On the other hand, deep learning-based approaches employ convolutional neural networks and YOLO to perform end-to-end, unsupervised object detection, in which features don't need to be defined and extracted separately.

Because deep learning methods have become the state-of-the-art approaches to object detection, these are the techniques we'll be focusing on for the purpose of our research.

Literature Review of papers included in this report:

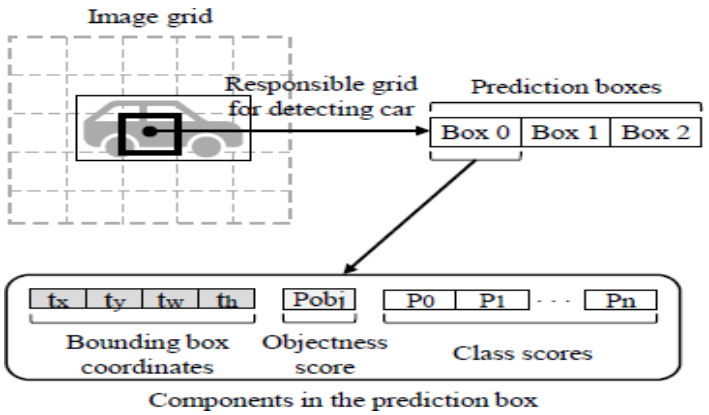
1	Gaussian YOLOv3: An Accurate and Fast Object Detector Using Localization Uncertainty for Autonomous Driving
2	Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection
3	YOLO-based Threat Object Detection in X-ray Images
4	YOLO v3-Tiny: Object Detection and Recognition using one stage improved model
5	Pedestrian Detection Based on YOLO Network Model
6	YOLOv4: Optimal Speed and Accuracy of Object Detection
7	Comparison of CNN and YOLO for Object Detection
8	Detection of Non-Helmet Riders and Extraction of License Plate Number using Yolo v2 and OCR Method
9	CPU Based YOLO: A Real Time Object Detection Algorithm

Paper 1: Gaussian YOLOv3: An Accurate and Fast Object Detector Using Localization Uncertainty for Autonomous Driving
Problem statement:

In this paper authors proposed a method for improving the object detection accuracy of real-time operation by modeling the bounding box of YOLOv3, which is the most representative of one- stage detectors, with a Gaussian parameter and redesigning the loss function.

Methodology:

In this research article the prediction feature map of YOLOv3 has three prediction boxes per grid, where each prediction box consists of boundary box coordinates the objectness score, and class scores. YOLOv3 outputs the objectness and class as a score of between zero and one. An object is then detected based on the product of these two values. The working of this model shown in the figure below:



Results:

The proposed method in this article can detect objects that YOLOv3 cannot find, thereby increasing its TP. These positive results are obtained due to loss attenuation effect in the learning process, so the learning accuracy for boundary box can be improved, which enhances the performance of objectness. The results show that Gaussian YOLOv3 can complement incorrect object detection results found by YOLOv3. Also, the results show that Gaussian YOLOv3 can accurately detect boundary box of object inaccurately detected by YOLOv3. Based on these results, Gaussian YOLOv3 can significantly reduce the FP and increase the TP. So, by using the proposed method of this article driving stability and efficiency are improved and fatal accidents can be prevented.

Paper 2: Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection

Problem statement:

Governments are facing many challenges to protect people from COVID-19 in many countries. As people of many countries are forced by laws to wear face masks in public, masked face detection is important to face appplications, such as object detection. To fight and win in the battle against coronavirus pandemic, Governments need guidance and surveillance on people in public areas, specifically in crowded places to ensure that wearing face masks laws are applied.

Methodology:

This research article used YOLOv2 with ResNet-50 for the feature extraction and detection in the training, validation, and testing phase for object detection. For a high-performance detector estimating the number of anchor boxes are important. Data and labels were flipped horizontally to increase the masked faces in dataset. This article used mean Intersection over Union (IoU) to estimate the number of anchor boxes.

In this article, ResNet-50 used as a deep transfer model for feature extraction. ResNet-50 has 16 residual bottleneck blocks each block has convolution size 1x1, 3x3, and 1x1 with feature maps (64, 128, 256, 512, 1024). The detection network (YOLO v2) is a convolutional neural network contain few convolutional layers, transform layer, and finally output layer. The transform layer extracts activations of convolutional layer and improves the steadiness of the deep neural network. The transform layer converts the bounding box forecast to be in outlines of the target box.

Results:

Authors do split of dataset up to 70% training data, 10% data for validation, and rest of 20% data for testing. The configuration of YOLO v2 with ResNet-50 with initial learning rate (σ) and the number of epochs was 60. The mini-batch size of the detector is set to 64. In terms of optimizer technique, Stochastic Gradient Descent with momentum and Adam is chosen to be their optimizer to enhance detector performance.

Paper 3: YOLO-based Threat Object Detection in X-ray Images
Problem statement:

To detect the threat objects manually in an X-ray machine is a ridiculous task for the baggage inspectors in bus stops, airports and train stations. Objects that are inside the baggage seen by X- ray machine are commonly difficult to recognize when rotated. Due to this, there is are high chances of missed detection, especially during rush hour.

Methodology:
This article focused on You Only Look Once (YOLOv3) architecture to detect threat objects. YOLOv3 predicts the bounding boxes using dimension clusters as anchor boxes. The features extractor used by YOLOv3 called Darknet-53. It has 53 convolutional layers trained on ImageNet dataset, which contains 1000 classes of images. Darknet-53 are combinations of convolutional layers with sizes 1×1, 3×3, and residual network. The final layer of the model used average pooling to downsample the output and softmax activation function. The proposed model was trained using stochastic gradient descent using a learning rate of 0.001 and 300 epochs.

Results:

The given paper showed that YOLOv3 was not accurate for detecting thin objects like wires. Overall, these results indicate that although transfer learning is a must-try approach in training a small amount of dataset, it does not always improve the accuracy of the model. These experiments confirmed that using a YOLOv3 model, the best mAP (52.40%) can be obtained by training the model from scratch instead of using the transfer learning technique in a small set of data. In addition, training multi-scale images improved the detection performance by 14% while increasing the image size improved the performance by only 12%.

Paper 4: YOLO v3-Tiny: Object Detection and Recognition using onestage improved model
This paper overviews the one stage and two stage detectors with their benefits and drawbacks. The two stage models focus on accuracy while one stage models focus on speed, so a YOLO v3- Tiny is proposed and compared to the previously available models.

HOG, CNN, RCNN, Fast RCNN and Faster RCNN are discussed and their pros and cons are described. Moreover, one stage models like SDD, YOLO (v1, v2, v3) are also discussed and their performance is compared. YOLO v3-Tiny is about 442% faster than other variants of YOLO model.

YOLO v1 uses Darknet Framework and dataset is ImageNet-1000, YOLO v2 uses Darknet-19 but reduces accuracy by 2% and YOLO v3 uses Darknet-53 and it has 53 convolutional layers. YOLO v3-Tiny is a lighter version of YOLO v3 but is more suitable for object detection.

Conclusion:

To conclude, if the accuracy is concerned, then RCNN is the best choice to make but if the accuracy is not the concerned then YOLO v3 is best to choose to obtain speedy results. But if the requirement is the best accuracy and less running time, then SSD is the best to choose.

Paper 5: Pedestrian Detection Based on YOLO Network Model
Problem Statement

Pedestrian Detection is one of the important tasks in some real-life problems. Most deep learning models are not enough good to detect pedestrians most accurately. In this paper, the YOLO structure is improved by adding three layers and named as YOLO-R. The role of the Route layer is to pass the pedestrian characteristic information of the specified layer to the current layer, and then use the Reorg layer to reorganize the feature map so that the currently-introduced Route layer feature can be matched with the feature map of the next layer.

Methodology

First the image is divided into a grid of SxS. If the pedestrian is in a grid, the grid is responsible for detecting the pedestrian. Each detection box has 5 predicted values (X, Y, W, H, Conf) and each grid predicts the pedestrian's conditional probability. At the time of detection, the conditional probability is multiplied by the predictive value of different detection box confidences to obtain the confidence score for each detection box pedestrian category.

Results:

The experiments results show that YOLO-R is much more accurate that YOLO v2 when tested on INRIA data set. YOLO-R is better in terms of Precision and Accuracy.

Paper 6: YOLOv4: Optimal Speed and Accuracy of Object Detection
Problem Statement

Nowadays the most recent accurate models need a large computational resource like GPUs for training purposes. In this paper a CNN model will be proposed that will require only one conventional GPU.

Methodology:

The main objective is to develop a fast-operating object detector for real time systems. A detector usually consists of two parts, a pre-trained backbone and a head for prediction. The head part is further categorized into one stage model and two stage models. The one stage models are RCNN, Fast RCNN etc. and two stage models are YOLO, SDD etc. There are different layers between head and backbone to collect feature maps, called neck of detector. To increase data variability, brightness, contrast, hue, saturation, noise, scaling, cropping, flipping, and rotating of an image are done in data augmentation.

The main objective is to find the best balance among the no of convolutional layers, the resolution of input network, the parameters and number of output layers. There are two possible options: For GPU, small number of groups (1 - 8) in convolutional layers: CSPResNeXt50 / CSPDarknet53 are used. And for VPU grouped-convolution are used, including models: EfficientNet-lite or MixNet etc. YOLO V4 is used with Backbone of CSPDarknet53, Neck: SPP PAN and Head: YOLOv3:

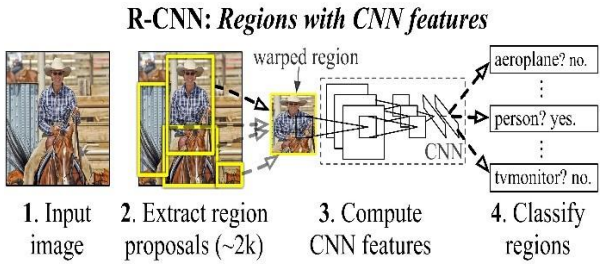
Conclusion

YOLO V4 is fastest in terms of both speed and accuracy to all alternative detectors available and a large number of features can be used to improve the efficiency of detector and classifies.

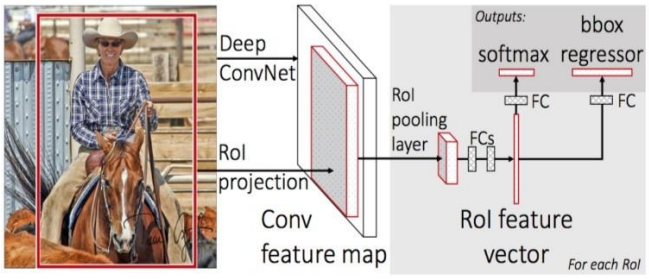
Paper 7: Comparison of CNN and YOLO for Object Detection

This paper is a comparison of three main object detection techniques in computer vision which includes R-CNN, Fast R-CNN and YOLO. Unlike machine learning, deep learning proves to be a more compact process in which the classification and localization occurs in a single trail after the image is captured. The techniques like R-CNN, Fast R-CNN, YOLO have emerged eventually making the process faster and compact.

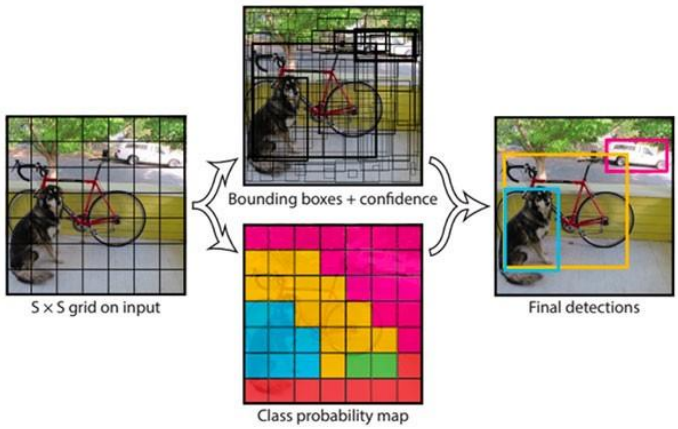
Region with CNN (RCNN): RCNN generates features in a region using CNN. The algorithm proposed selective search to extract just 2000 regions from the entire input image. These regions are referred to as region proposals. The subsets of these regions are identified in the image and are employed for classification of the objects in the regions. Therefore, instead of classifying big number of regions, just 2000 regions can be worked with. All these regions are generated by employing the selective search algorithm. The last layer of the CNN consists of the features which are extracted from the entire image and these features are further fed to a classification algorithm like support vector machine. This algorithm classifies the objects lying within the region proposal network.



Fast R-CNN: R-CNN model took a huge amount of time to train the network. To prevent this problem, another faster object detection algorithm was introduced known as Fast R-CNN. The input image is fed to the CNN algorithm which further generates convolutional feature map. The model takes the entire image as an input with a set of different object proposals. The network firstly processes the entire input image using more than one convolutional and max pooling layers so as to produce the convolution feature map for the entire image. The region proposals are generated using an algorithm such as Edge Boxes. For each of the object proposals, a region of interest (RoI) pooling layer converts the object proposals into a feature vector of a particular size. These feature vectors are fed to a set of fully connected layers attached to the two output layers, one layer produces softmax probability estimate and the other output layer gives four real value numbers for the K object classes. The bounding box coordinates are refined for one of the K classes using SVM trained using CNN features



YOLO: In contrast to the previous object detection algorithms, instead of complete image, the network looks at the portions of the image possessing greater possibilities that the object is present. This algorithm deploys bounding boxes and classifies the image hence predicting possibilities for these boxes using a unit convolutional network. YOLO works on input image by splitting it into an SxS grid, in each of which "m" bounding boxes are marked. In these bounding boxes, the network predicts class probability and offset values. The bounding boxes exceeding the threshold value for class probability are selected and are instrumental in locating the object inside the image.



COMPARISONS BETWEEN DIFFERENT OBJECT DETECTION TECHNIQUES

R-CNN	Fast RCNN	YOLO
Based on classification	Based on classification	Based on regression
Uses 2000 conv Nets for each region	Uses single deep convent	Uses single convolution network for entire image
Uses selective search algorithm	Uses selective search algorithm	---
Needs 49 seconds to test one image	Needs 2.3 seconds to test one image	Needs less than 2 second to test on one image
Slow, cannot be implemented real time	Faster than RCNN	Can run real time
Uses SVM for classification	Uses softmax for classification	Uses regression for classification
Produces bounding boxes	Produces bounding Box regression head and classification head	Produces bounding box, prediction contextual concurrently
Can find small objects	Can find small objects	Struggles to find small objects that appear in groups

CONCLUSION

RCNN, Fast RCNN and YOLO are the common techniques employed for object detection. RCNN and Fast RCNN are slower than YOLO but can detect small objects. YOLO is good at regression than classification. YOLO has difficulty in classifying small objects. Both RCNN and Fast RCNN fails to perform real time detection but YOLO can perform real time classification with good speed.

Paper 8: Detection of Non-Helmet Riders and Extraction of License Plate Number using Yolo v2 and OCR Method

Problem Statement

In this research work, a Non-Helmet Rider detection system is built which attempts to satisfy the automation of detecting the traffic violation of not wearing helmet and extracting the vehicles' license plate number. The main principle involved is Object Detection using Deep Learning at three levels. The objects detected are person, motorcycle/moped at first level using YOLOv2, helmet at second level using YOLOv3, License plate at the last level using YOLOv2. Then the license plate registration number is extracted using OCR (Optical Character Recognition).

Methodology:

The methodology is based on two conditions: First case is when the rider is wearing a Helmet and Second case is when the rider is not wearing a Helmet.

At first step, the frames are collected at regular intervals. After this, an input frame is passed to YOLOV2 model, where detection of objects is based on classes such as 'Motor bike' or 'Person'. At the output, image with required class detection along with confidence of detection through bounding box and probability value is obtained. Only the detected objects are extracted and stored as separate images and named with class name and image number in order. Once the person-motorcycle pair is obtained, the person images is given as input to helmet detection model. While testing this model, some false detections were observed. So, the person image was cropped to get only top one-fourth portion of image.

The cropped image is passed to **helmet detection model** to detect if a person is wearing helmet or not. If the helmet is found, then license plate detection model is not executed. But if the person is not wearing a helmet, this image is passed to the license plate detection model. Before applying OCR directly to extracted license plate image, pre-processing is done get output for better accuracy. Hence the image was rotated to get better results and then OCR technique is applied.

Results:

Results obtained are discussed here for two cases. They are, **Case 1:** When the motorcycle/moped rider is wearing helmet. **Case 2:** When the motorcycle/moped rider is not wearing helmet and License plate is detected

Sl. No	Detection model	Number Plate Detection	Threshold value
1	YOLO v2(Without Helmet)	Yes	0.5
2	YOLO v2(With Helmet)	No	0..87

Conclusion

A Non-Helmet Rider Detection system is developed where a video file is taken as input. If the motorcycle rider in the video footage is not wearing helmet while riding the motorcycle, then the license plate number of that motorcycle is extracted and displayed. Object detection principle with YOLO architecture is used for motorcycle, person, helmet and license plate detection. OCR is used for license plate number extraction if rider is not wearing helmet.

Paper 9: CPU Based YOLO: A Real Time Object Detection Algorithm
Problem Statement

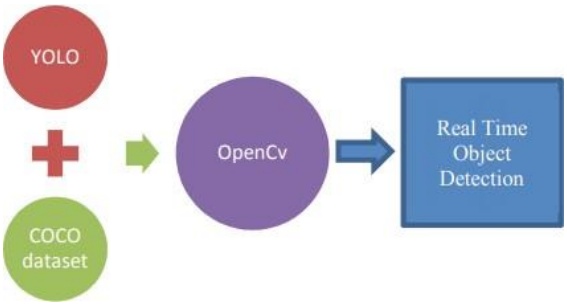
In this paper, we are proposing a model named "CPU Based YOLO" to run YOLO [7] algorithm on non-GPU computer with OpenCV. Our model is able to execute videos from external source or from webcam with minimum 10.12FPS, 80-99% confidence and with 31.05% mAP that is suitable for real time application within low cost and less effort.

Methodology:

To develop our model we need to install a deep learning framework where we will run the YOLO algorithm.

TensorFlow, DarkFlow, Darknet and OpenCv will be used to execute our model. For building CPU Based YOLO model for real time CPU based object detection, we will use YOLOV3 with DarkFlow and OpenCv.

The Architecture of the model is given below:



First the YOLOv3 and COCO dataset is loaded through OpenCV for real-time detection. When the video was given as input, the fps was too low. Several blob size for finding perfect FPS in CPU based computers was used. Using several different blob sizes, the final outcome was that low blob size increase the FPS but decrease the detection accuracy.

Results:

After performing several experiments in several computers for finding an optimum value of blob size, the outcome was 31.05% mAP and 16FPS (maximum) which is good enough for low configuration CPU based computers.

Computer /Laptop	RAM	Dataset	mAP	FPS
AMD Ryzen™ 3 2200G CPU 3.50GHz	8GB	COCO [14]	31.05%	16
Intel® Core™ i3-5010U CPU @ 2.10GHz	4GB	COCO	31.05%	7.7

Conclusion:

This model can be applied to normal Desktop or Laptop for executing YOLO. Thus, real time object detection can be deployed for several purposes like video surveillance, traffic monitoring, face tracking etc.