ISSN:2348-2079

Volume-6 Issue-1

International Journal of Intellectual Advancements and Research in Engineering Computations

Yolo intelligent vision based on machine learning

V.Logeswari[1], S.Deepak[2], S. Karthick Narayanan[3],S.Manikandan[4],S.Mohan Raj[5]

[1]Assistant Professor, [2345]UG Students

Department of Electronics and Communication Engineering, Nandha Engineering College, Erode-52

[1]vlogesivam@gmail.com,[2]deepaksethu97@gmail.com,[3]karthickn2010@gmail.com,

[4]mohansmart006@gmail.com, [5]manikandan95@gmail.com.

Tamil Nadu, India.

***ABSTRACT*---This project is about YOLO object detection.**

Priorworkonobjectdetectionrepurposesclassifierst oper- form detection. We frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. Asingleneuralnetwork predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. Our unified architecture is extremely fast. Our base YOLO model processes images in real-time at 45 frames

per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the map of other real-time detectors. Comparedtostate-of-the-artdetectionsystems,YOLO makes more localization errors but is far less likely to predict false detections where nothing exists. Finally, YOLO learns very general representations of objects. It outper- forms all other detection methods,including DPM and R- CNN, by a wide margin when generalizing from natura images to art work on both the Picasso Data set and the People- Art Dataset.

KEYWORDS: Ubuntu, Python3.5, OpenCV 3, Keras tensor flow, Sublime, Web-camera

I.INTRODUCTION

Humans glance at an image and instantly know what objects are in the image, where they are, and how they interact. The human visual system is fast and accurate, al- lowing us to perform complex tasks like driving with little conscious thought. Fast, accurate, algorithms for object de- tection would allow computers to drive cars in any weather withoutspecializedsensors,enableassistivedevicestocon- veyreal-timesceneinformationtohumanusers,andunlock the potential for general purpose, responsive robotic systems. Current detection systems repurpose classifiers to per- form detection. To detect an object, these systems take a

1. Resize image. 2. Run convolutional network. 3. Non-max suppression.

Dog: 0.30

Person: 0.64

classifier for that object and evaluate it at various locations and scales in a test image. Systems like deformable parts models (DPM) use a sliding window approach where the classifier is run at evenly spaced locations over the entire image. MorerecentapproacheslikeR-CNNuseregionproposal methodstofirstgeneratepotentialboundingboxesin ageandthenrunaclassifieronthesepro posedboxes. After classification, post-processing is used to refine the bounding box, eliminate duplicate detections, and rescore the box based on other objects. These complex pipelines are slow and hard to optimize because each individual component must be trained separately. We reframe object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. Using our system, you only look once (YOLO) at an image to predict what objects are present and where they are. YOLO is refreshingly simple. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection. First,YOLOisextremelyfast. Sinceweframedetection as a regression problem we don't need a complex pipeline. We simply run our neural network on a new image at testtime to predict detections. Our base network runs at 45 frames per second with no batch processing on a Titan X GPU and a fast version runs at more than 150 fps. This means we can process streaming video in real-time with less than 25 milliseconds of latency. Furthermore, YOLO achieves more than twice the mean average precision of other real-time systems. Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during training and test time so it encodes contextual information about classes as well as their appearance. Fast R-CNN, a top detection method, mistakes background patches in an image for objects because it can't see the larger context. YOLO makes less than half the number of background errors compared to Fast R-CNN. Third,YOLOlearnsgeneralizablerepresentationso objects. When trained on natural images and tested on art- work, YOLO outperforms top detection methods like DPM and R-CNN by a wide margin. Since YOLO is highly generalizable it is less likely to break down when applied to new domains or unexpected input.

II.RELATED WORKS

We unify the separate components of object detection into a single neural network. Our network uses features from the entire image to predict each bounding box. It also predicts all bounding boxes for an image simultaneously. This means our network reasons globally about the full image and all the objects in the image. The YOLO designenablesend-to-endtraining andreal-timespeedswhile maintaining high average precision. Our system divides the input image into S×S grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. Eachgridcellpredictsboundingboxesandconfidence scoresforthoseboxes.
Theseconfidencescoresreflecthow confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts. Formally we define confidence as truth pred. If no object exists in that cell, the confidence scores should be zero. Otherwise we want the confidence score to equal the intersection over union (IOU) between the predicted box and the ground truth. Each bounding box consists of 5 predictions: x, y, w, h, and confidence. The coordinates represent the center of the box relative to the bounds of the grid cell. The width andheightarepredictedrelativetothewholeimage.

III.PROPOSED SYSTEM ARCHITECTURE

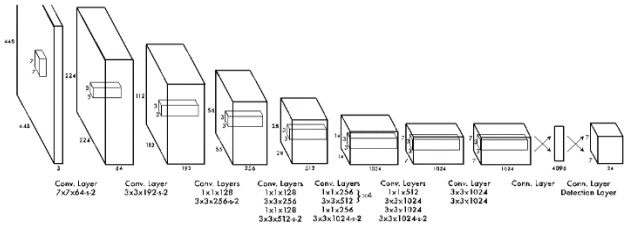
We implement this model as a convolutional neural net- work and evaluate it on the PASCAL VOC detection dataset. The initial convolutional layers of the network extract features from the image while the fully connected layers predict the output probabilities and coordinates. Our network architecture is inspired by the GoogleNet model for image classification. Our network has 24 convolutional layers followed by 2 fully connected layers. However, instead of the inception modules used by GoogleNet we simply use 1×1 reduction layers followed by 3×3 convolutional layers.

A.TRAINING

YOLO predicts multiple bounding boxes per grid cell. At training time we only want one bounding box predictor to be responsible for each object. We assign one predictor to be "responsible" for predicting an object based on which prediction has the highest current IOU with the ground truth. Thisleadstospecializationbetweentheboundingbox predictors. Each predictor gets better at predicting certain sizes, aspect ratios, or classes of object, improving overall recall.We train the network for about 135 epochs on the training and validation data sets from PASCAL VOC 2007 and2012. Whentestingon2012wealsoincludetheVOC2007 test data.

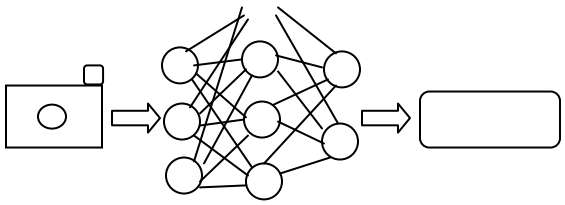
YOLO shares some similarities with R-CNN. Each grid cell proposes a potential bounding boxes and scores those boxes using convolutional features. However, our system puts spatial constraints on the grid cell proposals which helps mitigate multiple detections of the same object.

Our system also proposes far fewer bounding boxes, only 98 per image compared to about 2000 from Selective Search. Finally, our system combines these individual components into a single, jointly optimized model.



B. EXPERIMENTS

First we compare YOLO with other real-time detection systems on PASCAL VOC 2007. To understand the differences between YOLO and R-CNN variants we explore the errors on VOC 2007 made by YOLO and Fast R-CNN, one of the highest performing versions of R-CNN. Based on the different error profiles we show that YOLO can be used to rescore Fast R-CNN detections and reduce the errors from background false positives, giving a significant performance boost. We also present VOC 2012 results and compare map to current state-of-the-art methods. Finally, weshowthatYOLOgeneralizestonedomainsbetterthan other detectors on two artwork datasets**.**You will need a webcam connected to the computer that OpenCV can connect to or it won't work. If you have multiple webcams connected and want to select which one to use you can pass the flag.



Camera Neural network Monitor

CONCLUSION AND FUTURE WORK:

We introduce YOLO, a unified model for object detection. Our model is simple to construct and can be trained directly on full images. Unlike classifier-based approaches, YOLOistrainedonalossfunctionthatdirectlycorrespond s to detection performance and the entire model is trained jointly. Fast YOLO is the fastest general-purpose object detectorintheliteratureandYOLOpushesthestate-of-the-artin real-time object detection. YOLO also generalizes well to new domains making it ideal for applications that rely on fast, robust object detection.YOLO is a fast, accurate object detector, making it ideal for computer vision applications.

RESULT AND DISCUSSION:

The resulting system is interactive and engaging. While YOLO processes images individually, when attached to a webcam it functions like a tracking system, detecting objects as they move around and change in appearance.We use a totally different approach. We apply a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

REFERENCES:

[1] S. Gidaris and N. Komodakis. Object detection via a multi- region & semantic segmentation-aware CNN model. CoRR, abs/1505.01749, 2015

[2] Z.ShenandX.Xue. Domoredropoutsinpool5featuremaps for better object detection. arXiv preprint arXiv:1409.6911, 2014.

[3] S. Ren, K. He, R. B. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. CoRR, abs/1504.06066, 2015.

[4] M. A. Sadeghi and D. Forsyth. 30hz object detection with dpm v5. In Computer Vision–ECCV 2014, pages 65–79. Springer, 201.

[5] P. Viola and M. J. Jones. Robust real-time face detection. International journal of computer vision, 57(2):137–154, 2004.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learningforimagerecognition. ArXivpreprintarXiv:1512.03385, 2015.

[7] K. Simonyan and A. Zisserman. Very deep convolutional networks forlarge-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[8] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: To- wards real-time object detection with region proposal net- works. arXiv preprint arXiv:1506.01497, 201.

[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 2015.

[11] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 1440–1448, 2015.