ISSN: 2278-4632
Vol-13, Issue-11, No.02, November: 2023

OBJECT DETECTION USING YOLO MODEL

Sake Pothalaiah Department of ECE Vignana Bharathi Institute of Technology Hyderabad, India pothalaiahs@gmail.com
Yerram Srinivas Department of ECE Vignana Bharathi Institute of Technology Hyderabad, India srinivasyerram06@gmail.com
Y.Raju Department of CSE Vignana Bharathi Institute of Technology Hyderabad, India raju.yeligeti@gmail.com

ABSTRACT-
The world in the 21st century is ever evolving towards automation. This upsurge seemingly has no decline in the foreseeable future. Image recognition is at the forefront of this charge which seeks to revolutionize the way of living of the average man. If robotics can be likened to the creation of a body for computers to live in, then image processing is the development of the part of its brain which deal with identification and recognition of images. In this paper developed an object detection algorithm using YOLO (**You Only Look Once)**. Our algorithm was trained on fifty thousand images and evaluated on ten thousand images and employed a 21 x 21 grid also programmed a text generator which randomly creates texts and URLs in an image. A record of useful information about the location of the URLs in the image is also recorded and later passed to the YOLO algorithm for training.

Keywords: Object Detection; Image Recognition; OCR; AI; YOLO.

INTRODUCTION:

In this chapter, introduce the background and focus area of our thesis. highlighted our motivations, scope, and thesis organization. will explore YOLO's potential in recognizing objects and creating bounding box around them.

Digitized images are often represented as a two-dimensional (2D) array of pixels values. Each pixel value which makes up the color scheme of the image is often influenced by an array of factors such as light intensity. Visual scene is projected unto a surface, where receptors (natural or artificial) produce values that depend on the intensity of incident light. These exciting concepts are however hard to implement. Forming an image leads to loss of details of information while collapsing a three-dimensional (3D) image into a two-dimensional image. Many other factors are responsible for why image recognition/ image processing is hard. Some of such factors are noise in the image (pixels values that are off from its surrounding pixels), mapping from scene to image etc. In recent years, during the Image Net Large Scale Visual Recognition Competition (ILSVRC, 2015), computers re going better than humans in the image classification task [5] Object detection is a computer vision implementation that makes a system (an algorithm) about to estimate the location of objects in a digitized scene such as an image or video. Usually, a bounded box is wrapped around the detected object which helps humans locate the object quicker than unprocessed images. For this discourse, an object is the representation of a physical object (URL) in an image. In image processing, it is an identifiable portion of an image that can be interpreted as a single unit [7]. This creates a sharp contrast to the layman's idea that an image or an object are interchangeable.

GENERAL OBJECT DETECTION MODEL

Figure 1 is an object detection structure which has a region proposal component followed by a CNN classifier. Researchers use region proposal methods to produce a bunch of candidate regions, each of which may contain one kind of object. Each region is then passed through the
CNN classification algorithm. The model is to convert a multiple object detection problem into a single object classification problem., these region proposal methods are much slower than classification part, which becomes the bottleneck for the whole system. The drawback of this structure is that cannot tradeoff accuracy for detection speed in time critical applications

ISSN: 2278-4632
Vol-13, Issue-11, No.02, November: 2023

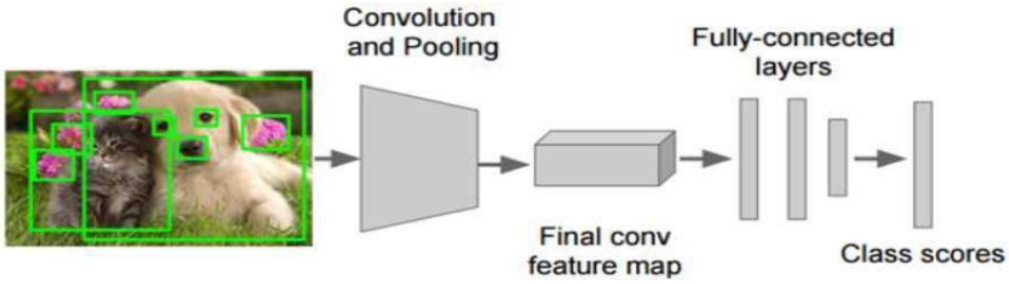

Figure 1. Structure of a CNN Based Object Detection Model

UNIFIED DETECTION MODEL – YOLO

The YOLO model was first brought into existence by Joseph Redmon in his paper "You only look once, Unified, Real-time object detection". The mechanism for the algorithm employs the use of a single neural network that takes a photograph as an Input and attempts to predict bounding boxes and class labels for each bounding box directly. Although this offered less predictive accuracy, which was mostly due to more localization errors, it boasted speeds of up to 45 frames per second and up to 155 frames person on speed optimized versions of the model [2].

"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 …" [4]

To begin with the model operates by splitting the inputted image into a grid of cells, where each cell is responsible for predicting a bounding box if the center of a bounding box falls within it. Each grid cell predicts a bounding box involving the x, y coordinate and the width and height and a metric of valuation of quality known as a confidence score. A class prediction is also based on each cell. To supply more emphasis an instance will be provided. For example, an image may be divided into a 7 × 7 grid and each cell in the grid may predict 2 bounding boxes, resulting in 94 proposed bounding box predictions. The class probabilities map and the bounding boxes with confidences are then combined into a final set of bounding boxes and class labels.

The YOLO was not without shortcomings, the algorithm had a number of limitations be- cause of the number of grids that it could run on as ll as some other issues which will be addressed subsequently. Firstly,the model uses a 7 × 7 grid and since each grid can only identify an object, the model restricts the maximum
number of objects detectable to 49. Secondly, the model suffers from what is known as a close detection model,
since each grid is only capable of detecting one object, if a grid cell contains more than one object it will be unable to detect it. Thirdly, a problem might arise because the location of an object might be more than a grid, thus, there exists a possibility that the model might detect the object more than once [8]. Due to the aforementioned problems encountered when running YOLO, it was fairly obvious that localization error and other problems of the system needed to be addressed. As a result of that, YOLOv2 was created as an improvement to deal with the issues and questions posed by its predecessor.
Therefore, localization errors as ll as errors of real re significantly addressed in the new version. The model was updated by Joseph Redmon and Ali Farhadi to further revamp model performance in their 2016 paper named "YOLO9000: Better, Faster, Stronger [6]".
STRUCTURE OF YOLO

YOLO is implemented as a convolution neural network and has been evaluated on the PASCAL VOC detection dataset. It consists of a total of 24 convolutional layers followed by 2 fully connected layers. The layers are separated by their functionality in the following manner:

1. First 20 convolutional layers followed by an average pooling layer and a fully connected layer is pre-trained on the ImageNet 1000-class classification dataset.
2. The pretraining for classification is performed on dataset with resolution 224 × 224.
3. The layers comprise of 1 × 1 reduction layers and 3 × 3 convolutional layers.

ISSN: 2278-4632
Vol-13, Issue-11, No.02, November: 2023

4. Last 4 convolutional layers follo d by 2 fully connected layers are added to train the network for object detection.
5. Object detection requires more granular detail hence the resolution of the dataset is bumped to 448 × 448.
6. The final layer predicts the class probabilities and bounding boxes.

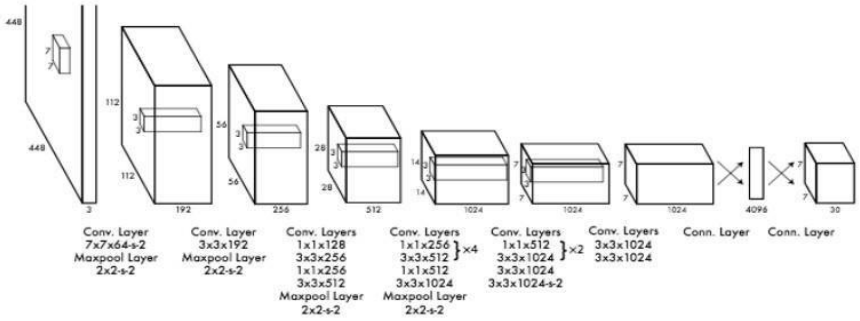

Figure 2. YOLO Structure.

The final layer uses a linear activation whereas the other convolutional layers use leaky ReLU activation. The input is 448 × 448 image, and the output is the class prediction of the object enclosed in the bounding box.

Network Structure

Figure 2. YOLO Network Architecture [5].

The whole system can be divided into two major components: Feature Extractor and Detector; both are multi- scale. When a new image comes in, it goes through the feature extractor first so that it can obtain feature embeddings at three (or more) different scales. Then, these features are feed into three (or more) branches of the detector to get bounding boxes and class information [3].

Intersection over Union (IoU)

YOLOs default metric for measuring overlap bet en two bounding boxes or masks is Intersection over union (IoU). Any algorithm that provides predicted bounding boxes as output can be evaluated using IoU [8]. If the prediction

is completely correct, $IoU = 1$. The lo r the IoU, the worse the prediction result. To apply Intersection over union, certain

parameters must be met to evaluate an (arbitrary) object detector, they are: The ground-truth bounding boxes these are the

hand labeled bounding boxes from the testing set that specify where in the image our object is and the predicted bounding boxes from our model. If these two sets of bounding boxes are present it is possible to apply Intersection over union. Thus, computing Intersection over Union can therefore be calculated as; The area of overlap divided by the area of union [9]

Plainly put, the intersection over union is a ratio of the similarity of the ground truth bounding box and the predicted bounding box, thus predicting a rough estimate on how much an Artificial intelligence can rely on the predictions made by the algorithm. This is an improvement over binary models which label predictions as either correct or incorrect. Also due to the variance in parameters bet en the model and the object it is quite unrealistic to have a 100% match bet en the (x, y) coordinates a predicted bounding box and the (x, y) coordinates of the ground truth box [7]. Thus, this equation ensures that boxes with a larger area of overlap get higher scores than those with lesser areas thus cementing Intersection over union as excellent metric for evaluating custom object detectors. Generally, any score greater than 0.8 is a good score [5]. This application of

ISSN: 2278-4632
Vol-13, Issue-11, No.02, November: 2023

intersection over union is used during the testing phase at the completion of the training [6].

Another interesting application of this model occurs where there is more than one bounding box for the same object. the metric helps to eliminate bounding boxes with lesser scores.

How this is done is that if there are two bounding boxes with very high confidence scores, then the chances are that the boxes are detecting the same object therefore the box with the higher confidence rating is selected. However, where confidence rating of the two boxes is low, it is likely that they are predicting separate objects of the same class, such as two different cars in the same picture. This application is used after the model has completed training and is being deployed [6]

As earlier mentioned, Intersection over union is a good metric for evaluating the quality of a task. Before further elucidation, it is pertinent to define some terminologies first, these terminologies are true positive, true negatives, false negatives, and false positives. Simply put, a true positive is any correctly drawn annotation with a value greater than 0.5, a true negative is when an F1 refuses to draw any annotation because there simply is not one to be drawn. There is no value here because no annotation is drawn, thus there is no way to calculate true negatives.

A false negative occurs where there are missing annotations while a false positive occurs where there are incorrectly drawn annotations that have an IoU score of less than 0.5. An equation known as accuracy is usually used to measure the performance of a task because it is an incredibly straight forward measurement as well as for its simplicity. it is simply a ratio of correctly drawn annotations to the total expected annotations (ground truth). Thus, it is calculated as being equal to the sum of True positive and True negative divided by the sum of True positive, False positive, True negative and False negative.

Direct Location Prediction

In the early versions of the YOLO model, there were no constraints on the location prediction. The predicted bounding box was not tethered therefore it could occur far from the original grid location. This anomaly resulted in a very unstable model [6]. The bulk of the instability resulted from predicting the (x, y) locations for the box. In region proposal networks the network predicts values t_x and t_y and the (x, y) center coordinates are calculated as:

$$x = (t_x * w_a) - x_a$$

$$y = (t_y * h_a) - y_a$$

For example, a prediction of $tx = 1$ would shift the box to the right by the width of the anchor box, a prediction of $tx = -1$ would shift it to the left by the same amount [6]. This formulation did not exist within any boundaries and as such, any anchor box could end up at any point in the image regardless of what location predicted the box. with random initialization it took the model an obscene amount of time to stabilize to predict sensible offsets [6]. The subsequent versions of YOLO diverged from this approach and devised a means to properly tackle the situation. YOLOv2 bounds the location using logistic activation.

Sigma (σ), which ensures that the value remains between 1 and 0 [7]. Given the anchor box of size (p_w, p_h) at the grid cell with its top left corner at (c_x, c_y), the model predicts the off- set and the scale, (t_x, t_y, t_w, t_h) and the corresponding predicted bounding box has center (b_x, b_y) and size (b_w, b_h). The confidence score is the sigmoid (σ) of another output t_o.Since the location prediction is constrained, the parameterization is easier to learn, making the network more stable. The employment of dimension clusters along with directly predicting the bounding box's center location improves YOLO by almost 5% over the version with anchor boxes n, m.

In digital image processing, the bounding box is the coordinates of a rectangle wishing which an object may be contained when it is placed over a page, a canvas, a screen, or any other similar bi-dimensional background [8]. In the field of object detection, a bounding box is usually used to

ISSN: 2278-4632
Vol-13, Issue-11, No.02, November: 2023

describe an object location.
The bounding box is a rectangular box that can be determined by the x and y axis coordinates in the upper-left
corner and the x and y axis coordinates in the lower-right corner of the rectangle [6]. The first version of
YOLO directly predicted all four values which describes a bounding box. the x and y coordinates of each
bounding box are de- fined relative to the top left corner of each grid cell and normalized by the cell dimensions such that the coordinate values are bounded between 0 and 1. However in YOLOv2 there was a shift in paradigm and the algorithm employed dimensional clusters in place of anchor boxes, 4 coordinates are predicted
for each bounding box, t_x, t_y, t_w, t_h, If the cell is offset from the top left corner of the image by (c_x, c_y) and the bounding box prior has width and height p_w, p_h, then the predictions correspond to

$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h}$$

[5]: Bounding Box prediction formula given below as .
ground truth box) minus our prediction: $\hat{t_*} - t_*$. This ground truth value can be easily computed by inverting the
equations above. YOLOv3 predicts an object ness score for each bounding box using logistic regression. This
should be 1 if the bounding box prior overlaps a ground truth object by more than any other bounding box prior is not the best but does overlap a ground truth object by more than some threshold the prediction is ignored. use the threshold of .5. usually, a system only assigns one bounding box prior for each ground truth object. If a bounding box prior is not assigned to a ground truth object it incurs no loss for coordinate or class predictions, only object ness [5].

The concept of a bounding box prior was introduced in YOLOv2. Previously, the model was expected to provide unique bounding box descriptors for each new image, a collection of bounding boxes is defined with varying aspect ratios which embed some prior information about the shape of objects are expecting to detect. Redmon offers an approach towards discovering the best aspect ratios by doing k-means clustering (with a custom distance metric) on all of the bounding boxes in the training dataset [5].Thus, instead of predicting the bounding box dimension directly, the task is reformulated to simply predict the offset from the bounding box prior in order to fine-tune the predicted bounding box dimensions. The result of which is that it makes the prediction task easier to learn. **Objectness (and assigning labeled objects to a bounding box)** "objectness" score $pobj$ is trained to approximate the intersection over union bet en the predicted box and the ground truth label. When the loss during training is calculated, Objects are matched to whichever bounding box prediction on the same grid cell produced the highest IoU score. For unmatched boxes, the only descriptor which will be included in the function is $pobj$.

Upon the introduction of additional bounding box priors in YOLOv2, it was possible to assign objects to whichever anchor box on the same grid cell has the highest IoU score with the labeled object.

YOLO (version 3) redefined the "objectness" target score p_{obj} to be 1 for the bounding boxes with highest IoU score for each given target, and the score 0 for all remaining boxes. However bounding boxes which have a high IoU score above a defined threshold but not the highest score when calculating the loss will not be included. This simply means

EXPERIMENT:

In real-life application, speed, accuracy, and resources tradeoffs must be made. The inter- net is a great resource for different classes of trained data ready for use. Unfortunately, there is not one class ready made for recognition of URLs in an image. Most tests and trainings are done with datasets,

ISSN: 2278-4632
Vol-13, Issue-11, No.02, November: 2023

and their results are measured in mean Average Precision (mAP) at Intersection over Union (IoU) threshold. IoU measures the overlap between two regions.

In this paper will highlight the custom generation of our training data from our word bank of 56000 (Figure 3) words. A dataset is a collection of data. Image datasets are used to train and benchmark object detection algorithms. For the algorithm to be trained, it needs to create images with URLs in them and store the coordinates of the URLs. This would be used by the training algorithm to know what is right (a URL) and use that to learn what is wrong (not a URL).

Python was our choice programing language and because YOLO is just a technique for detection using convolutional neural network, made use of pythons' keras library as ll.

Figure 3. Word bank used to generate our training image data.

Figure 4 shows a sample image generated from our word bank. This process is known as tagging, and ours is automated. A total of sixty thousand images re generated for this experiment, each having varying number of URL(s) in them and some having no URL at all, giving the algorithm a large enough sample size. A record of the x and y coordinate of any URL generated is stored and is later passed into our detection algorithm.

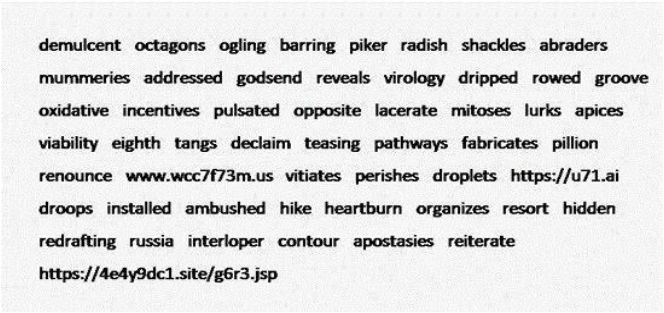

Figure 4. Sample generated image using our custom-built tagging tool.

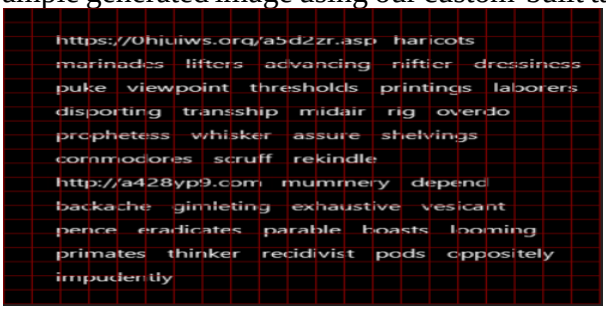

Figure 5. Sample grid generated by YOLO for our detection algorithm.

Training, according to Longman Dictionary , is the process of teaching or being taught the skills for a particular job or activity. In this context, it refers to the process of teaching an algorithm towards a specific task for which it will be used. These algorithms also learn from experience without being explicitly programmed. In our experiment, labelled ten thousand images, each with varying number of URLs and some with no URL at all. trained our algorithm using fifty thousand images. These images have varying
font size. Our image is sectioned into 21 × 21 grid cells (Figure 5), each is responsible for predicting K
bounding boxes. The grid cell was selected because would be working with only text.

ISSN: 2278-4632
Vol-13, Issue-11, No.02, November: 2023

An object is considered to lie in a specific cell only if the center co-ordinates of the anchor box lie in that cell. Due to this property, the center co-ordinates are always calculated relative to the cell, whereas the height and width are calculated relative to the whole image size. Using the Equation below, YOLO deter- mines the probability that a cell contains a certain class. The class with the maximum probability is chosen and assigned to that grid cell. This is repeated for all grid cells in the image. The probability that there is an object of certain class 'c' is:

$score_{c,i} = p_c \times c_i$

Setting 0.3 filter size and 0.7 IoU threshold, applied non-max suppression to select the bounding box with the highest IoU threshold. Batch normalization is then applied to add noise to the inputs of every

later, discouraging overfitting preventing our model from producing deterministic values for a given training example. later applied max pooling to down-sample out input (the batch normalized output) followed by our activation function. settled with Rectified Linear Units (ReLU) at the end of the training process, provided the trained algorithm with a new set of data (with ten thousand images) to test our result. This new dataset is modelled like the training data.

V. CONCLUSION:

This work aims to incorporate state of-the-art technique for object detection with the goal of achieving high accuracy with a real-time performance. A major challenge in many of the object detection systems is the dependency on other computer vision techniques for helping the deep learning-based approach, YOLO observed significant difference in the accuracy of URL detection when using an OCR software or our YOLO algorithm. However, YOLO algorithm would be best used to specify the region of interest before converting to texts which greatly improves accuracy when combined with OCR software.

REFERENCES:

1. Bekkouch Imad Eddine Ibrahim et al., "Few-Shot Object Detection: Application to Medieval Musicological Studies", Journal of Imaging, vol. 8, no. 2, pp. 18, 2022
2. Shuyu Miao et al., "Balanced single-shot object detection using crosscontext attention-guided network", Pattern Recognition, vol. 122, pp. 108258, 2022.
3. . Eike Jens Hoffmann, Karam Abdulahhad and Xiao Xiang Zhu, "Using Social Media Images for Building Function Classification", 2022.
4. Chang Xu et al., "RGB-T salient object detection via CNN feature and result saliency map fusion", Applied Intelligence, pp. 1-20, 2022.
5. Jun Jiang et al., "Abnormal behavior detection using streak flow acceleration", Applied Intelligence, pp. 1-18, 2022.
6. Juan Montenegro and Yeojin Chung, "Semi-supervised generative adversarial networks for anomaly detection", SHS b of Conferences, vol. 132, 2022.
7. M. Goumas, "The use of human behavioural cues by urban herring gulls", 2022
8. Michael Seedall, "Malicious Interlocutor Detection Using Forensic Analysis of Historic Data", Diss. University of Huddersfield, 2022.
9. Elizabeth S. Greene, Justin Leidwanger and Leopoldo Repola, "Ephemeral Heritage: Boats Migration and the Central Mediterranean Passage", American Journal of Archaeology, vol. 126, no. 1, pp. 79-102, 2022.
10. Hsuan Hsu and Kuo-Feng Tseng, "Facing the era of smartness: constructing a framework of required technology competencies for hospitality practitioners", Journal of Hospitality and Tourism Technology, 2022