

Real-Time Waste Classification and Localization with YOLOv3

Yug Gajjar¹, Dhruv Bhavsar², Pratik Awaik³, Naresh Patel⁴ and Jagruti Naik⁵

¹⁻³Gujarat Technological University, Ahmedabad, India

{yug.gjr, dhruvbhavsar095, pratikawaik125}@gmail.com

⁴⁻⁵ Vishwakarma Government Engineering College, Electronics & Communication Department, Ahmedabad, India

Email: {naresh, jnaik}@vgecg.ac.in

Abstract—Waste management for solids as is in the current state-of-the-practice, cannot keep up with the waste generation rates of countries, and the currently employed methods of managing solid waste have adverse effects on the environment. In this paper, we propose an alternate approach to how we currently manage solid waste by classification of waste by its type with the use of object-detection systems. We propose an object-detection model that frames object-detection as a regression problem, and it is made to recognize and classify waste, which it does with 89.22% confidence, this helps streamline the process of solid waste management.

Index Terms— waste classification; YOLO; logistic regression; bounding box prediction.

I. INTRODUCTION

Solid waste management is an issue that has implications for every person in every country. With the daily per capita waste generation in high-income countries projected to increase by 19 percent by year 2050[1], and for low-income and middle-income countries, by more than 40 percent [1], efficient handling of waste becomes a priority.

The currently employed processes to handle waste have a negative impact on the environment and on the quality of life for the people in a country. These inefficient processes also pollute the environment, for instance the commonly employed method of open waste burning causes air pollution, and to add to the problem, toxic materials from the waste spread carcinogens [2] into the air. In low-income countries, over 90 percent of the waste is either dumped or burned [1].

Thus, efficient handling, processing and recycling of waste becomes a necessity for protecting the environment and the health of citizens, classifying the waste into the different classes which need to be recycled or processed using different methods increases the efficiency of the waste management or recycling process. In this paper, we propose an object-detection system able to recognize and classify waste in real-time by video stream analysis.

II. RELATED WORK

In recent times, there have been a large number of approaches proposed for object-detection like [9] and [10], most of which re-purpose classifiers or localizers to perform object-detection by applying the model to an image at multiple locations and scales where the high scoring regions of the test image would be considered detections.

This paper proposes an approach to classify objects that utilizes YOLO [7], which is a real-time object-detection system that models object-detection as a regression problem.

Approaches to object-detection like R-CNN systems use region proposal methods to generate potential bounding boxes in a test image, and then run a classifier on these proposed bounding boxes, which detects objects. Unlike these approaches, YOLO approaches object-detection as a single regression problem, it produces the coordinates of bounding boxes and class probabilities directly from the test image's pixels.

III. YOU ONLY LOOK ONCE - YOLO

You Only Look Once (YOLO) is a real-time object- detection system. It is a single convolutional neural network that simultaneously predicts multiple bounding boxes and classes of the entire image in the single scan. The network has 24 convolutional layers followed by two fully connected layers. In YOLO, 1×1 reduction layers were used, followed by 3×3 convolutional layers.

Currently, there are four versions of YOLO (v1, v2, v3 and v4). The original YOLO network is also termed as YOLOv1. YOLOv2 was an improved version of YOLOv1, with the advantages of speed with the introduction of batch normalization, anchor boxes and a high-resolution classifier compared to YOLOv1. In YOLOv3, a better feature extractor was introduced with the introduction of 53 convolutional layers trained on ImageNet. The accuracy of the YOLOv3 is better than YOLOv2 but it is slower due to more layers.

A. Limitations of older YOLO versions

- In YOLOv1, only 49 objects could be detected by the 7×7 grid since each cell can only detect a single object.
- Since each cell in YOLOv1 cannot predict more than one object, this results in a close object-detection problem occurring when near small objects exist in the same cell.
- If the object is somewhat large and located in more than one cell, it will be detected more than once in YOLOv1.
- YOLOv1 suffers from the high localization error and low recall.
- YOLOv2 cannot detect small objects in an image in an accurate way. Because, YOLOv2 architecture is still missing some of the most important components that are now staples in most of the state-of-the-art detection algorithms like the residual blocks, skip connections and up-sampling.
- The output prediction is carried out at the last layer.

IV. YOLOv3

YOLOv3 applies a singular neural network to the full image, dividing the image into regions, and predicting bounding boxes and probabilities for each region. The bounding boxes are weighted by the predicted probabilities. This approach to object detection has numerous advantages over classifier-based systems, for instance, since this approach analyzes the complete image at the test time, its predictions are informed by global context in the image. Furthermore, since this approach makes predictions with a single network evaluation, it is extremely fast compared to other classifier-based systems R-CNNs which use regional proposal methods to first generate potential bounding boxes in a test image and then run a classifier on these proposed bounding boxes.

A. YOLOv3 Architecture

The most recent version of YOLO is YOLOv3 shown in Fig. 1, which is much denser than the previous two versions and it is more accurate while compromising the speed of computation. The main contribution here is YOLOv3 able to get more meaningful semantic information during the training process from the up-sampled features. YOLOv3 limitations and upgrades are summarized as follows:

YOLOv3 is a hybrid approach between the network used in YOLOv2 and residual network. Consequently, it has some short-cut connections which allows us to get more fine-grained information from the earlier feature map which leads to better performance for small objects.

YOLOv3 performs detections at 3 different scales (levels), i.e., 3 YOLO layers unlike YOLOv1 and YOLOv2 which predict the output at the last layer. At each scale, 3 anchor boxes are used to predict 3 boxes for any grid cell. The first YOLO layer has anchors larger than 60×60 , the 2nd layer has the anchors larger than 30×30 and the 3rd layer has the remaining boxes. It uses non-maximum suppression to detect the object only one time with the best score and remove any redundancy predictions for the same objects.

	Type	Filters	Size	Output
	Convolutional	32	3×3	256×256
	Convolutional	64	$3 \times 3 / 2$	128×128
1x	Convolutional	32	1×1	
	Convolutional	64	3×3	
	Residual			128×128
	Convolutional	128	$3 \times 3 / 2$	64×64
2x	Convolutional	64	1×1	
	Convolutional	128	3×3	
	Residual			64×64
	Convolutional	256	$3 \times 3 / 2$	32×32
8x	Convolutional	128	1×1	
	Convolutional	256	3×3	
	Residual			32×32
	Convolutional	512	$3 \times 3 / 2$	16×16
8x	Convolutional	256	1×1	
	Convolutional	512	3×3	
	Residual			16×16
	Convolutional	1024	$3 \times 3 / 2$	8×8
4x	Convolutional	512	1×1	
	Convolutional	1024	3×3	
	Residual			8×8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 1. YOLOv3 architecture

B. YOLOv3 Limitations

YOLOv3 has more layers compared to YOLOv1 and YOLOv2, so it takes more time in training and prediction. Other than this, the close objects problem is handled, but some limitations still exist for detection of very close objects.

V. DATA ANNOTATIONS

Before training the model, the dataset needs to be manually annotated with the bounding box information of 6 classes (cardboard, glass, metal, paper, plastic, trash). The data annotation was done using vott. YOLOv3 takes a column vector as an input containing the coordinates of bottom left corners and top right corners of the bounding box in the training data. The software generates bottom left coordinates (Xmin, Ymin) and top right coordinates (Xmax, Ymax). The coordinates are rounded off to fit into YOLOv3 format.

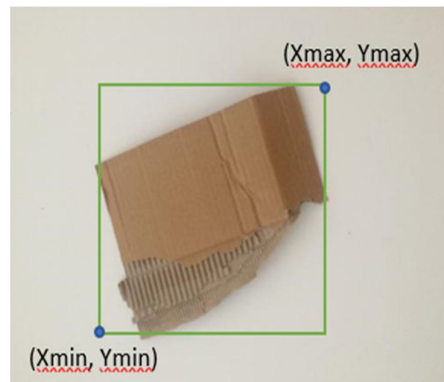


Figure 2. Co-ordinates for bounding box

VI. TRAINING

The network predicts 4 coordinates 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:

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_w e^{t_w}$$

$$b_h = p_h e^{t_h}$$

YOLOv3 predicts an “object-ness” score for each bounding box using logistic regression, which is the confidence that some object exists in a given box. This should be 1 if the bounding box prior overlaps a ground truth object by more than any other bounding box prior. if the bounding box prior is not the best but does overlap a ground truth object by more than some threshold, we ignore the prediction.

For class prediction, YOLOv3 uses logistic regression instead of SoftMax regression because it is unnecessary for good performance, SoftMax imposes that each box has only one class, which cannot be true in every cases. YOLOv3 uses binary cross-entropy to calculate loss for the class predictions.

For feature extraction it uses hybrid approach between the network used in YOLOv2, Darknet-19, and that newfangled residual network stuff. Our network uses successive 3x3 and 1x1 convolutional layers but now has some short-cut connections as well and is significantly larger. It has 53 convolutional layers.

TABLE I. TRAINING OF THE MODEL

Epoch	Training loss	Validation loss
5	52.1490	47.7583
10	29.1849	28.1734
20	19.8145	20.1082
30	17.2815	17.3645
40	16.0997	16.4305
50	15.4969	15.7186
60	12.4057	12.6906
70	10.3502	10.7470
80	8.8683	9.1421
90	8.0992	7.9723
102	7.1185	7.1049
102	7.1185	7.1049

This model can predict the output with 89.22% confidence.

VII. PREDICTION

A. Logistic Regression

In previous versions of YOLO, SoftMax regression method was used to predict the output from the model weights so if more than one objects are there in the image then only one object which was detected first will be given as the result.

In YOLOv3, this disadvantage has been overcome by the logistic regression method, threshold amount is decided and based on those multiple labels can be classified.

YOLO v3 is a variant of darknet in which 53 layers are of ImageNet and for prediction another 53 layers are added, so in total we have 106 layers available in YOLOv3.

YOLO performs predictions based on first down-sampling the image then layer by layer performing the up-sampling of an image.

B. Up-sampling and Down-sampling in YOLOv3

YOLO v2 used a custom deep architecture Darknet- 19, originally a 19-layer network supplemented with 11 more layers for object detection. With a 30-layer architecture, YOLO v2 often struggled with small-object detection. This was attributed to loss of fine-grained features as the layers down-sampled the input. To remedy this, YOLO v2 used an identity mapping, concatenating feature maps from a previous layer to capture low level features.

The newer architecture boasts of residual skip connections and up-sampling. The most salient feature of YOLO v3 is that it makes detections at three different scales. YOLO is a fully convolutional network and its eventual output is generated by applying a 1×1 kernel on a feature map. In YOLO v3, the detection is done by applying 1×1 detection kernels on feature maps of three different sizes at three different places in the network.

The shape of the detection kernel is:

$$1 * 1 * (B * (N + \text{object-ness feature} + C))$$

Here, B is the number of bounding boxes a cell on the feature map can predict, N is the number of bounding box attributes, and C is the number of classes

C. Bounding box prediction

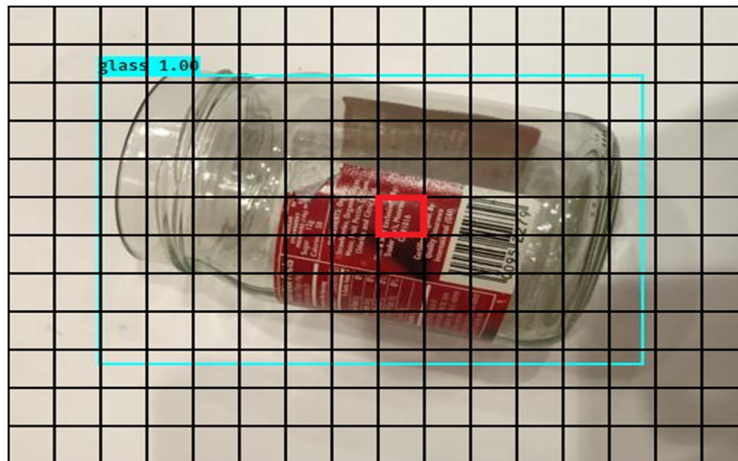


Figure 3. Grid in YOLOv3

In the above figure you can see the grids created on the result image which was predicted by YOLOv3. Here we have total classes $C = 6$ for prediction and number of bounding boxes are $B = 3$, so according to the formula we have the kernel size of $1 \times 1 \times 33$. The feature map produced by this kernel has identical height and width of the previous feature map, and has detection attributes along the depth as described above.

Stride of the network, or a layer is defined as the ratio by which it down-samples the input. in the above figure one red bounded box is there, the structure of that bounded box will look like this below figure:

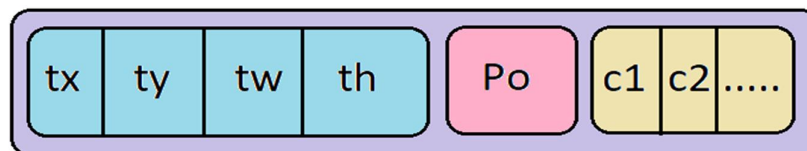


Figure 4. Attributes of bounding box

In the above figure tx , tw , ty , th represents the coordinates of the predicted bounding box and P_o stands for the objectiveness of the given bounding box with the actual object and $c1$, $c2$ are the classes which are going to be predicted by the object-ness score.

D. Prediction by layer in YOLOv3

YOLOv3 makes prediction at three scales, which are precisely given by down-sampling the dimensions of the input image by 32, 16 and 8 respectively.

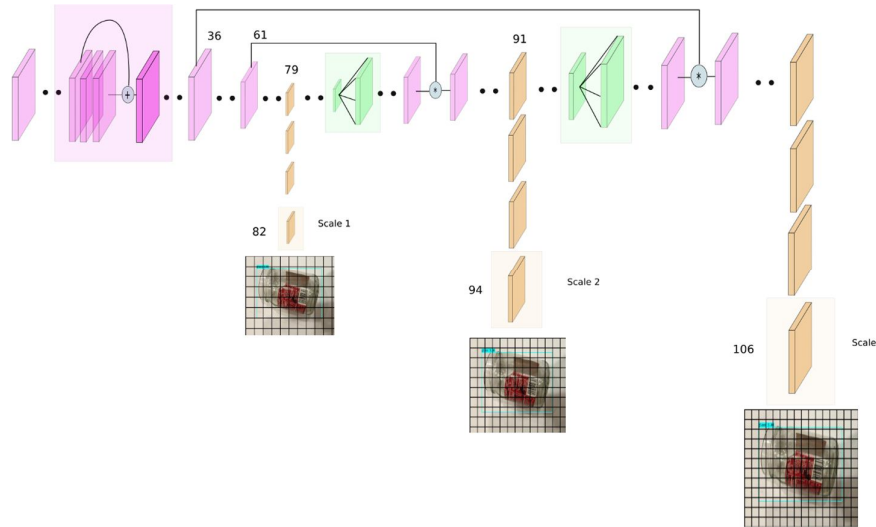


Figure 5. Layered Prediction in YOLOv3

The first detection is made by the 82nd layer. For the first 81 layers, the image is down sampled by the network, such that the 81st layer has a stride of 32. we have an image of 512×512 , the resultant feature map would be of size 16×16 . One detection is made here using the 1×1 detection kernel, giving us a detection feature map of $16 \times 16 \times 33$.

Then, the feature map from layer 79 is subjected to a few convolutional layers before being up-sampled by a factor of 2, to dimensions of 32×32 . This feature map is then depth concatenated with the feature map from layer 61. Then the combined feature maps are again subjected a few 1×1 convolutional layers to fuse the features from the earlier layer (61). Then, the second detection is made by the 94th layer, yielding a detection feature map of $32 \times 32 \times 33$.

A similar procedure is followed again, where the feature map from layer 91 is subjected to few convolutional layers before being depth concatenated with a feature map from layer 36. Like before, a few 1×1 convolutional layers follow to fuse the information from the previous layer (36). We make the final of the 3 at 106th layer, yielding feature map of size $64 \times 64 \times 33$.

VIII. EXPERIMENT AND RESULTS

The datasets for this study were collected from the TACO dataset. The datasets were subjected to an 80%–20% split, where 80% was treated as the train set and 20% as the test set.

Below are some of the results from the test dataset:



Figure 6. Test set images



Figure 7. Test images after prediction

IX. CONCLUSIONS

Through the development of this study's model, and empirically evaluating the performance of the proposed model, we found that for classifying waste into different classes, through framing object-detection as a regression problem with the help of YOLOv3 algorithm, evaluated on the public TACO dataset, the results indicate that the proposed model is well suited for waste classification in real-time, with an 88.92% confidence of correct classification.

REFERENCES

- [1] Kaza, Silpa; Yao, Lisa C.; Bhada-Tata, Perinaz; Van Woerden, Frank. 2018. What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050. Urban Development; Washington, DC: World Bank. © World Bank. <https://openknowledge.worldbank.org/handle/10986/30317> License: CC BY 3.0 IGO.
- [2] Cogut, A. (2016). Open burning of waste: a global health disaster. R20 Regions of climate action.
- [3] Redmon, J., & Farhadi, A. (2018). Yolo3: An incremental improvement. arXiv preprint arXiv:1804.02767.
- [4] Aly, G. H., Marey, M., El-Sayed, S. A., & Tolba, M. F. (2021). YOLO Based Breast Masses Detection and Classification in Full-Field Digital Mammograms. *Computer Methods and Programs in Biomedicine*, 200, 105823.
- [5] De Carolis, B., Ladogana, F., & Macchiarulo, N. (2020, May). Yolo trashnet: waste detection in video streams. In 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS) (pp. 1-7). IEEE.
- [6] Adedeji, O., & Wang, Z. (2019). Intelligent waste classification system using deep learning convolutional neural network. *Procedia Manufacturing*, 35, 607-612.
- [7] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779- 788).
- [8] Redmon, J., & Farhadi, A. (2018). Yolo3: An incremental improvement. arXiv preprint arXiv:1804.02767.
- [9] Kang, Z., Yang, J., Li, G., & Zhang, Z. (2020). An automatic waste classification system based on deep learning. *IEEE Access*, 8, 140019-140029.
- [10] Rabano, S. L., Cabatuan, M. K., Sybingco, E., Dadios, E. P., & Calilung, E. J. (2018, November). Common waste classification using mobilenet. In 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM) (pp. 1-4). IEEE.