International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 04 Issue: 08 | August -2020 ISSN: 2582-3930

Object Detection and Segmentation, A Survey using Yolo v3

Ankit Singh, Sandesh M, Apoorv Anupam, Rakshith G H, *Student, Dept. Of Computer Science,*

Rajesh Kumar S, *Associate Professor, Dept. Of Computer Science*

Cambridge Institute of Technology

Special thanks to Praveen Kumar, for helping us curate this paper

Abstract—Object detection in recent years has evolved to object localization and segmentation, making use of a highly constructed backend network for better results. Object segmentation includes localizing pixels to a targeted class and helps in outlining the detected object. In this paper, there is a display of the various techniques used in the past for object localization and segmentation, and giving a better approach to obtain segmented and location results. The primary dataset used is from the ImageNet 2012 challenge for classifying images, segmenting and perform object localization andYOLO v3 architecture, the major neural network module, for a series of functional model steps which includes Image augmentation, Batch Normalization, and cut off per epoch iteration. The training methods provides better boundary boxes for locating and segmenting. The steps of the proposed detection algorithms are described and illustrated.

Index Terms—object detection, localization, segmentation, YOLOv3, neural network.

1 Introduction

Object recognition, in the form of machine vision, is the ability of software to identify objects, places, people, writing actions in images. Computers can use machine vision technology coupled with camera and text input software to achieve image recognition. Image recognition is used to generate large amounts of virtual machine objects, such as documenting image content through meta-tags, searching image content, navigating autonomous robots, self-driving vehicles, risk avoidance programs. The goal of this field is to teach machines to understand (recognize) the content of an image just like humans do. Image segmentation is a computer-based visualization program designed to facilitate image updating by dividing visual input into graphical sections or parts of objects and creating a set of pixels or “large pixels”. Image segmentation filters into pixels into larger parts, eliminating the need to view each pixel as a preview.

Convolution Neural Network (CNN) is one of the most popular ways of doing object recognition. It is widely used, and most state-of-the-art neural networks used this method for various object recognition related tasks such as image classification. This CNN network takes an image as input and outputs the probability of the different classes. If the object is present in the image then its output probability is high else the output probability of the rest of classes is either negligible or low. The advantage of Deep learning is that we don’t need to do feature extraction from data as compared to traditional methods.

Object recognition algorithms like YOLO use bound boxes to show parts of an image that contain something and then divide it. This restricts their ability as they do not provide any information about the condition of the item.

But making use of techniques such as batch normalization, code Net, Image training, improvised learning rate to help us with dropout and cut off, which makes this instance segmentation stand out. With so many computer vision functions, it is not enough to simply point to a class of objects. These tasks require the classification of images, showing the composition of an item, and marking an item from an image. Image segmentation allows for a consistent understanding of objects within the image.

Image detection picks us where image recognition leaves and essentially uses image recognition at its heart. On top of

recognizing what is in the image, it tries to find the local position of classes on images. In Image classification, it takes an image as an input and outputs the classification label of that image with some metric (probability, loss, accuracy, etc). For Example: An image of a cat can be classified as a class label "cat" or an image of Dog can be classified as a class label "dog"with some probability.

1.1 Object Localization:

This algorithm locates the presence of an object in the image and represents it with a bounding box. It takes an image as input and outputs the location of the bounding box in the form of (position, height, and width). Object Detection: Object Detection algorithms act as a combination of image classification and object localization. It takes an image as input and produces one or more bounding boxes with the class label attached to each bounding box. These algorithms are capable enough to deal with multi-class classification and localization as well as to deal with the objects with multiple occurrences.

Image segmentation is a further extension of object detection in which we mark the presence of an object through pixel-wise masks generated for each object in the image.

Image segmentation is a computer-based visualization program designed to facilitate image updating by dividing visualinput into graphical sections or parts of objects and creating a set of pixels or "large pixels". Image segmentation filters into pixels into larger parts, eliminating the need to view each pixel as a preview. This technique is more granular than bounding box generation because this can help us in determining the shape of each object present in the image. This granularity helps us in various fields such as medical image processing, satellite imaging, etc. There are many image segmentation approaches proposed recently.

Instead of saying that a certain area has sheep, for example, parts of the image can identify where each sheep ends up and the next one starts. In many computer vision applications, the detection and recognition of objects is an important task. We introduce the methodology that could be followed to get better results on the entire process, initiating from detection to
segmentation.

2 LITERATURE REVIEW

[1]This paper includes methods for object detection combining top-down and bottom-up recognition image segmentation. This approach consists of majorly two steps- a hypothesis generation step and a verification step. Object deformation and background clutter, a Shape Context feature is designed which is more robust, this is the basic idea of top-down hypothesis generation step. This improved Shape Context generates a set of hypotheses of figure-ground masks and object locations. The second step, firstly set of feasible segmentations are computed that are consistent with top-down object hypotheses. To prune out false positives they propose a procedure called False Positive Pruning (FPP). The fact of false-positive regions typically not aligning with any feasible image segmentation is exploited. Method overview. There are three parts to the method (shaded rectangles). Codebook building (cyan) is the training stage, generating codebook entries with enhanced SC features and object masks. Top-down (blue) recognition generates multiple hypotheses in the input image through improved SC matching and voting.

The verification part (pink) aims to use bottom-up segmentation to verify those top-down hypotheses.

Round-corner rectangles are processes and the input / output dataare ordinary rectangles[2]. In this research paper analysis of the techniques of object recognition and segmentation with images and videos is carried out. Some applications are very obvious with regard to object recognition such as Robot navigation, Medical diagnosis, Protection, Industrial inspection and automation, Human-Computer interface, Information Recovery. Segmentation is now commonly used in diverse fields, such as image processing,video recognition, shadow detection, identification of human activity and many more. The paper includes various existing skills used in object recognition and segmentation, with organized representation and precision. Identification assumes information about recognition when carrying out the research.

When doing the analysis, identification concludes facts about the classification and segmentation of an object from the perspective of static and moving objects with respect to scrutinization. Most methods are based on the algorithmic and mathematical models. Results were winded up with the merits and demerits of existing methods and future scope liabilities in this area. [3] Semantic image segmentation, which is one of the primary technologies in the field of image processing and computer vision, has been introduced for several domains, such as medical and smart transport. There are several datasets that are published for researchers in the market to test their algorithms. With the recent emergence of Deep Neural Network (DNN), segmentation has made very tremendous progress, it is years of research topic. The paper addresses two approaches, the standard approach and recent DNN methods. The first part of the review of conventional methods and datasets consists of eight aspects of recent DNN-based methods, a completely convolutionary network, up sample ways, FCN combined with CRF methods, dilated convolution approaches, backbone network development, pyramid methods, multi-level and multi-stage methods, supervised, poorly supervised thus it draws a conclusion.

[4] Accuracy is never enough to satisfy the detection results of object detection. Efficiency and accuracy with regard to rapidity is discussed in the context. A real-time detection technique, making use of the YOLOv3 algorithm by deep learning techniques is demonstrated. Over 3 unique scales crosswise expectations are made at first. Utilization of the

identification layer made to highlight maps of 3 distinct sizes in recognition, this having stride 32, 16, 8 individually. This implies, with a partner contribution of 416 x 416, we will in general form a location on scales 13 x 13, 26 x 26 and 52x 52. To anticipate the jumping box article score, this makes use of strategic relapse. To foresee the classes that the bounding box may contain, a cross-entropy misfortune is utilized. After the forecast the certainty is determined. This results in performing multi-label classification for detecting objects from an image, increasing the average preciseness for tiny objects. It is much quicker than RCNN. MAP increased significantly. MAP increase has decreased localization errors.

2.1 Limitations of Current Work

This trade-off between precision and efficiency must be chosen according to the requirement. The problem includes classification as well as regression, which contributes to the simultaneous learning of the model. Which adds to the difficulty of the problem. The key problem in this issue is that of the output variable dimension that is induced by the variable number of artefacts that can be present in any given input image.

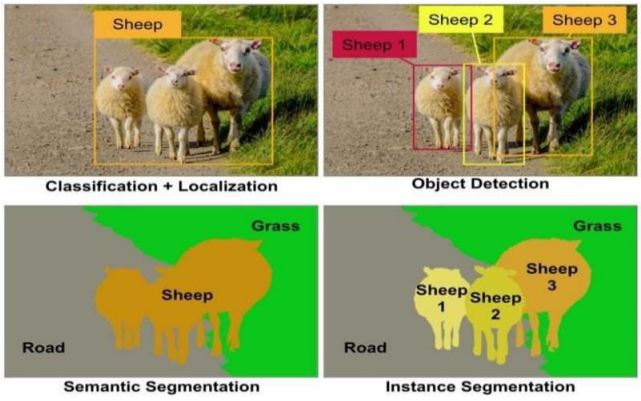
Every general function of machine learning requires a fixed input and output dimension to be trained in the model. Another significant barrier to the widespread adoption of object detection systems is the need to detect objects in real-time (30fps) while being precise.

2.2 Basic Theory

Object recognition is a general term to identify a set of computer vision related activities involving theidentification of objects in digital images.

Classification of images involves predicting the class of a single object in an image. Localization of an object refers to defining the position of one or more objects in an image and drawing a boundary box around its distance. Object detection combines these two tasks and locates one or more objects in an image, and classifies them.

In the realm of computer vision there are a wide variety of neural network applications. And with a bit of twist, they can effectively apply the same tools and techniques across a wide range of tasks. We'll walk through some of those applications and ways to approach them in this paper. The most popular of the four are: -



- Semantic segmentation
- Classification and localization
- Object detection
- Instance segmentation

Figure 1: Classification of Objects Based on Techniques

3. WORKFLOW

3.1 Dataset

ImageNet is a high-resolution, 22,000-resolution website of over 15 million images. The images were collected on the web and labelled by employee authors using the crowdsourcing tool Amazon Mechanical Turk. Since 2010, there has been an annual ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) as part of the Pascal Visual Object Challenge. For every 1000 segments ILSVRC uses an ImageNet subset of approximately 1000 images. There are almost 1.2 million training files, 50,000 accredited images and 150,000 test files in all. ImageNet comes with customizable images. The images are therefore limited to a fixed resolution of 256 / 256. The image is released due to a rectangular image and then wears the 256 some 256 media cover to the resulting image.

3.2 Batch Normalization

Whenever we want to train a network, we provide a batch of input from the complete dataset and these batch keep changing until the epoch is completed. In simpler words, the whole dataset is divided into batches of images with each image in a batch being processed parallelly on the GPU. Training is complicated by the fact that the inputs to each layer are affected by the parameters of all preceding layers. So, small changes in the network parameters amplify as the network becomes deeper. The training proceeds in steps, at each step considering a ***mini-batch***. Thus, each mini-batch has a different distribution from different parts of the dataset. So, the network has to continuously adapt to new distribution. whenever this input distribution changes, it is called **Covariate Shift**. Internal covariate shift simply means the changes inside network layers. Thus, our aim is to centralize these distributions (at least during training & validation) which vary on each mini-batch. This is the main reason why BN comes into the picture.Thus, Batch Normalization will help us in having to pass information which is more vividly accepted to next layers.

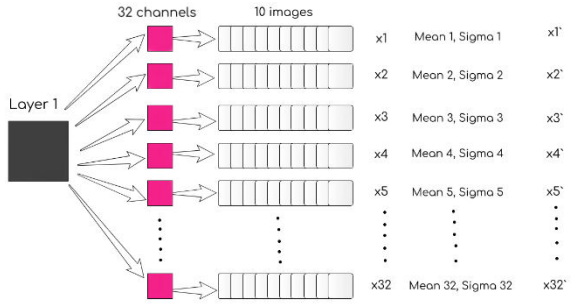


Figure 2: Batch Normalization on a simple network.

Benefits of Batch Normalization

The purpose of batch planning is to maximize network training. It has been shown to have many benefits:

1. Faster Training Networks - While each training will be slower due to less normalization statistics at the time of passing and additional hyperparameters for training while spreading backward. However, it should evolve very quickly, so training should be fast enough.
2. Allows higher read rates - Gradient emergence often requires smaller read rates for the network to interact. As the networks begin to deepen, the gradients become smaller during backpropagation, so it requires further validation. Using batch understanding allows for higherlevels of learning, which increases the speed at which networks train.
3. Makes instruments easy to get started - Getting started can be difficult, especially when building deep networks. Batch adjustment helps to reduce the sensitivity of the first one.
4. Enabling Multiple Functions-In certain situations, some activation functions don't work well. Sigmoid easily lose their gradient, meaning they can't be used
   in deep networks, and ReLUs frequently die off during training (stop learning altogether), so we need to be cautious about the range of values that are allocated to them. But since the values that flow through each application are controlled by a standard batch, offline malfunctions on deep networks often work too.
5. Simplify deep network architecture - Previous 4 points make it easy to build and fast train deep neural networks, and deep networks often produce better results.
6. Provides some implementation - Batch optimization adds a little bit of noise to your network, and in some cases, (e.g. start-up modules) has been shown to work as well as off function. You can view batch optimization as a general assumption, allowing you to limit your output to the network.

3.3 VGG16-Convolutional Network for Classification and Detection

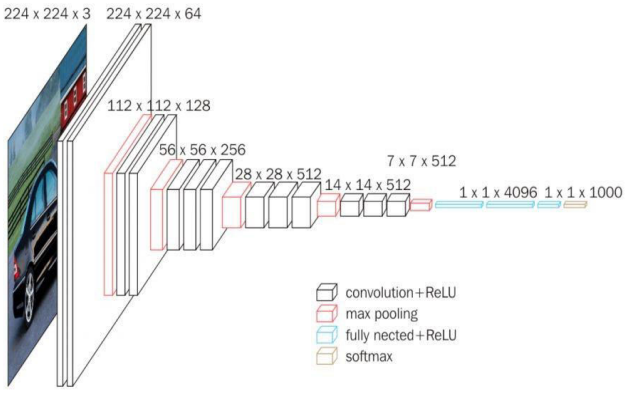


Figure 3: Architecture of VGG16

The cov1 layer input is not the default image size of 224 x 224 RGB. This image is passed through a layer stack (of conf.), Where the filters were applied with a very small reception field: 3 × 3 (the smallest size to hold left / right view, top/bottom, centre). In one configuration, it also works with 1 x 1 filters, which can be seen as a line change of input channels (followed by incompatibility). The cutting-edge line is cantered at 1 pixel; local suspension for cargo. the layer input is so large that the surface adjustment is retained after detection, e.g. 1-pixel padding of 3 × 3 Conv. layers. Spatial pooling is made up of five layers of water max, followed by some of these siblings. layers (not all bugs. layers are followed by max-pooling). Max-pooling is done in a 2 x 2-pixel window, with stride 2. The three Integrated Layer (FC) layers follow a series of permissive layers (with different depths in different designs): the first two have 4096 channels each, the third performs 1000 ILSVRC isolations and thus contains 1000 channels (each phase). The last layer is the soft-max layer. Optimization of fully integrated layers is the same across networks.

All hidden layers are fitted with rearrangement (ReLU) non-linearity. It is also notable that none of the networks (other than one) contain Local Response Normalization(LRN), simulations like this do not improve performance on ILSVRC data, but lead to increased memory usage and compilation time.

4. YOLOv3

YOLOv3 is current state of the art architecture and we tweaked it to the problem statement, thus, helped in attaining better results for boundary boxes, segmentation. This results into multiple and more highly probable outcomes.

4.1 YOLOv2 (YOLO9000) and YOLOv3

The model was updated by Joseph Redmon and Ali Farhadi in their 2016 paper entitled "YOLO9000: Better, Faster, Stronger," in an attempt to increase model performance.

SSD is a strong competitor for YOLO which at one point demonstrates higher accuracy for real-time processing.

Comparing with region-based detectors, YOLO has higher localization errors and the recall (measure how good to locate all objects) is lower. YOLOv2 is the second version of the YOLO with the objective of improving the accuracy significantly while making it faster. The addition of Batch normalization, High resolution classifier and more anchor boxes, Yolo-v2 certainly outperformed previous architecture.

The YOLO training composes of 2 phases. First, train a classifier network like VGG16. Then replace the fully connected layers with a convolution layer and retrain it end-to-end for the object detection. YOLO trains the classifier with 224 × 224 pictures followed by 448 × 448 pictures for the object detection. YOLOv2 starts with 224 × 224 pictures for the classifier training but then retune the classifier again with 448 × 448 pictures using much fewer epochs. This makes the detector training easier and moves mAP up by 4%.

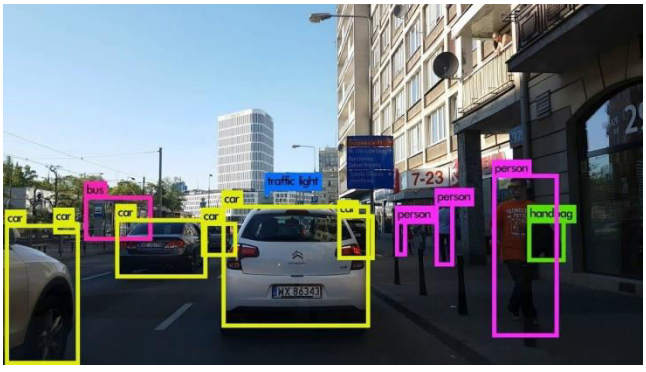


Figure 4: Objects detected by YOLO.

This modified YOLO predicts detections on a 13 × 13 feature map. While this is sufficient for large objects, it may benefit from finer grained features for localizing smaller objects. Faster R-CNN and SSD both run their proposal networks at various feature maps in the network to get a range of resolutions. It takes a different approach, simply adding a pass-through layer that brings features from an earlier layer at 26×26 resolution.

Make predictions on the offsets to the anchors. Nevertheless, if it is unconstrained, the guesses will be randomized again. YOLO predicts 5 parameters (Tx, ty, Tw, Th, and to) and applies the sigma function to constraint its possible offset range.

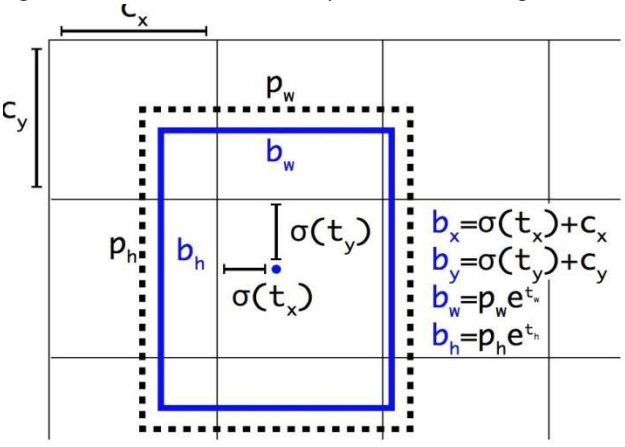


Figure5: Example of the Representation Chosen when Predicting Bounding Box Position and Shape Taken from:
YOLO9000: Better, Faster, Stronger.

With the use of k-means clustering (dimension clusters) and the improvement mentioned in this section, mAP increases 5%. The idea of mixing detection and classification data faces few challenges: 1-Detection datasets are small comparing to classification datasets. 2-Detection datasets have only common objects and general labels, like “dog” or “boat”, while Classification datasets have a much wider and deeper range of labels for example ImageNet dataset has more than a hundred breeds of dog like German shepherd and Bedlington terrier. It’s a little bigger but more accurate. Same as YOLO9000 the network predicts 4 coordinates for each bounding box, Tx, ty, Tw, Th. If the cell is offset from the top left corner of the image by (cx, cy) and the bounding box prior has width and height Pw, Ph.

YOLOv3 also predicts an objectness score(confidence) 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. For example (prior 1) overlaps the first ground truth object more than any other bounding box prior (has the highest IOU) and prior 2 overlaps the second ground truth object by more than any other bounding box prior. The 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 objectness. If the box does not have the highest IOU but does overlap a ground truth object by more than some threshold, ignore the prediction (They use the threshold of 0.5).

Multi labels prediction: In some datasets like the Open Image Dataset an object may has multilabel, for example an object can be labelled as a woman and as a person. In this dataset there are many overlapping labels. Using a SoftMax for class prediction imposes the assumption that each box has exactly one class which is often not the case (as in Open Image Dataset). A multilabel approach better models the data. For this reason, YOLOv3 do not use a SoftMax, instead it simply uses independent logistic classifiers for any class. During training it use binary cross-entropy loss for the class predictions. Using independent logistic classifiers an object can be detected as woman an as a person at the same time.

Small objects detection: YOLO struggled with small objects. However, with YOLOv3 show a better performance for small objects, and that because of using short cut connections. Using these connections method allows us to get more fine-grained information from the earlier feature map. However, comparing to the previous version, YOLOv3 has worse performance on medium and larger size objects.

You Only Look Once (YOLO) is an end-to-end network for object detection. YOLO splits the image to SxS block, and then each block is responsible for detecting those targets whose centre points fall within the grid. After detection, Non-Maximum Suppression is used for eliminating the duplicated bounding boxes. The 3-rd. generation of YOLO, YOLOv3 has integrated

many cutting-edge technologies, including residual block-based backbone, feature pyramid network like network head for multi-scale prediction, batch normalization, anchor boxes prediction, etc. Darknet53 is an efficient backbone for performing feature extraction. It is a very deep backbone that contains 53 convolutional layers and it also conveys many advanced structure including: (a) Residual blocks which add shortcut layers to make the network easier to train; (b) Inception structure that contains 3x3, 1x1 convolutional kernel which keep the respective field and decrease the computation cost; (c) Batch normalization layer makes the learning of layers in the network more independent of each other.

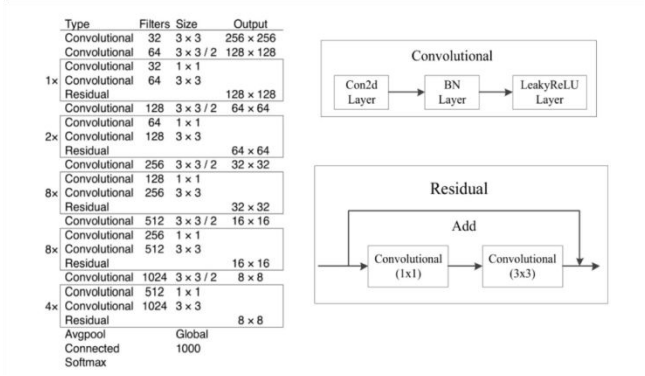


Figure 6: Darknet53 as feature extractor

With the high efficiency of the Darknet53, it is selected as the feature extractor for both YOLOv3 head and later segmentation used. The comparison of Darknet53 and other popular backbone which are widely using in similar tasks including Mask R-CNN.

Darknet53 backbone is pre-trained on the ImageNet [35] dataset for the general feature learning step. Darknet53 has basically the same Top-1 and Top-5 accuracy on ImageNet classification tasks with ResNet101.However, it has fewer billions of operations and higher billion floating-point opera22tions per second, which means it computes faster and has fewer calculations. These make Darknet 53 more efficient.

YOLOv3 head adapts feature pyramid network (FPN) like structure to improve the multi-scale prediction. It makes three predictions regarding each different scale. The output tensor conveys bounding boxes coordinates, each class' confidence scores and objects confidence (1 for object and 0 for non-object). The output of the YOLO head is post-processed by non-maximum suppression to eliminate the extra bounding boxes. The bounding boxes output is then prepared as ROI for the segmentation task. The first part is YOLOv3, which proposes the region of interest (ROI) and regresses for the classes and confidence scores. In the thesis, a modified YOLOv3 is implemented, which also outputs the feature maps from every last layer of the residual blocks. The second part is ROIAlign that takes different YOLO bounding box outputs and feature maps as inputs and generates the fixed-size ROI feature maps. The last part is Fully Convolutional Network (FCN), which transforms the ROI inputs to the semantic mask outputs.

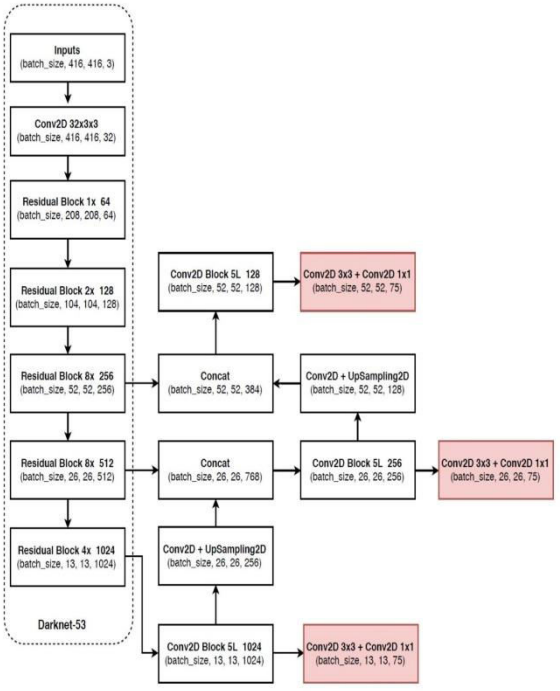


Figure 7: Yolo Head

5. METHODOLOGY

The approach lies in the following manner-

- We first fetch the ImageNet dataset and get the object localization boxes in a json format.
- The json formatted boxes and image data are separately handled and stored into a NumPy array.
- We build Yolo v3 architecture and while doing so, we add several tweaks such as, applying batch normalization after every layer except the last layer. Also, a similar technique is employed to having dropout of neurons after every layer with a drop probability of 15%.
- We also tune out dataset by passing it through a train generator function which performs image augmentation techniques on the image flow to the model input layer. We also prepare a validation data generator by employing similar image augmentation techniques.
- Training is performed using the base VGG neural network, treating the model further on YOLOv3. We had a custom loss function which could validate Yolo v3 after each validation epoch.

- The trained model is tested on a test bed where images are pre-processed according to the model's requirement during training phase.

6. RESULTS

The following results have been obtained using a CNN based model training on Google Collaboratory online tool, TensorFlow Keras framework, ImageNet dataset, Image Augmentations, Batch Normalization, Dropout and Data flow generators.

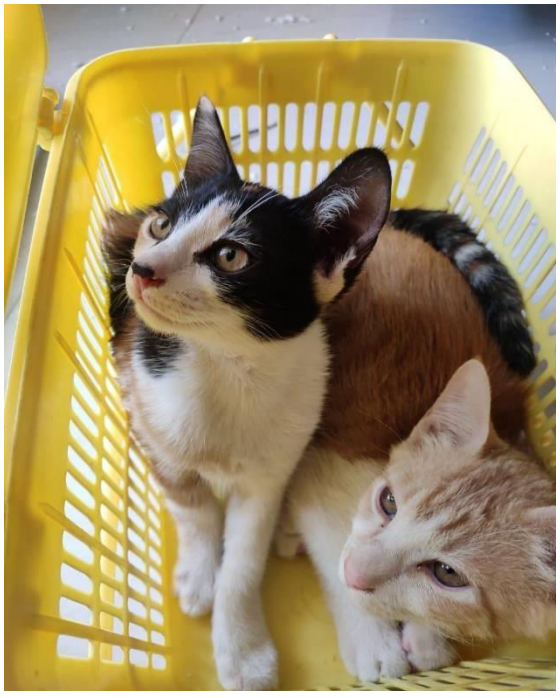


Figure 8: Two Cats

Images are fed post CNN model which is trained on object localization and trained on pixel segmentation. The model was made to learn to classify each pixel after localization. The images are uploaded in a zip format to the pre-trained model, all images can be taken real-time also, using any web camera, or mobile device as long as the GPU is able to handle the backend work.Accuracy is mentioned on each of the objects on the images with the help of YOLO v3 objectness and loss function probabilities. Detection, localization and segmentation can clearly be seen from the images, there are objects recognized falsely, with given small probability of that object being actually recognized. A lot of it depends on the hardware components as well.

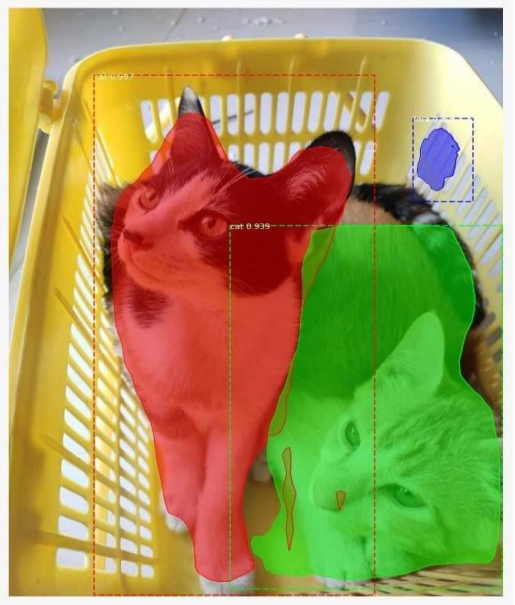


Figure 9: Segmented Output

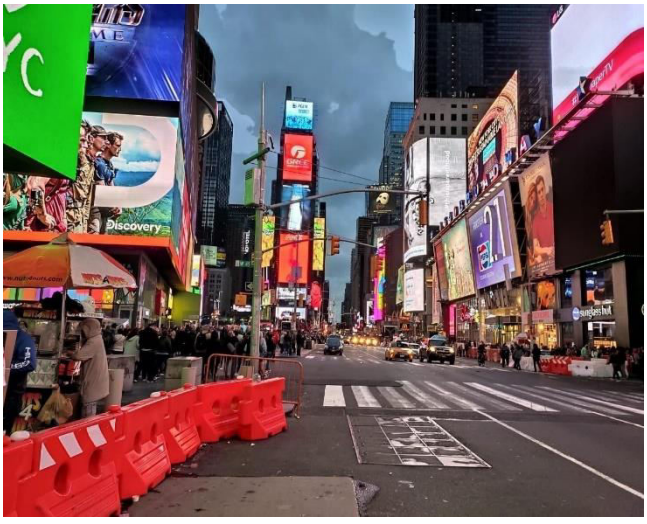


Figure 10: Time Square, NY, USA

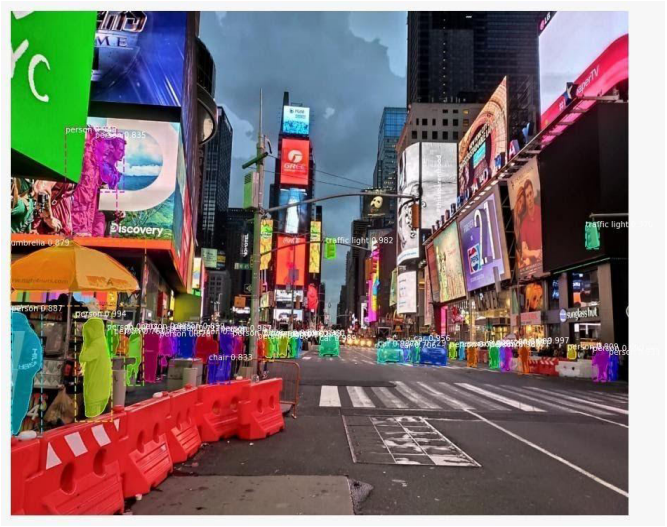


Figure 11: Segmented Output

5. Conclusion

A flexible model can be built on tools which are not very highly compute efficient. Tools, frameworks and the techniques used in this paper takes advantage of recent computer science and neural networking advancements. It can be used in real-time applications involving pre-processing object segmentation, such as Drone control, surveillance cameras, etc. We have exploited existing methodologies and used them to build a methodology to make real time localization and segmentation on a low hardware device. Since this is a very important area of research with significant consequences in all fields of life starting with medical images, it is difficult to overestimate the importance of this analysis. We display the importance of YOLOv3, usage of Image augmentation techniques, positive effect of Batch Normalization and CNN functional architecture. We conclude the model process with feeding the images and measuring the accuracy of training. A better idea of segmentation is achieved in the research.

REFERENCES

[1] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[2] Ross Girshick. Fast R-CNN. In International Conference on Computer Vision (ICCV), 2015.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NIPS), 2015.

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, ChengYang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In ECCV, 2016.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[7] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and ´ hierarchical image segmentation. TPAMI, 33(5):898–916, 2011. 3, 5

[8] G. Bertasius, J. Shi, and L. Torresani. High-for-low and low-for high: Efficient boundary detection from deep object features and its applications to high-level vision. In ICCV, 2015. 2

[9] G. Bertasius, J. Shi, and L. Torresani. Semantic segmentation with boundary neural fields. In CVPR, 2016. 2

[10]T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In ECCV, 2010. 1, 8 [5] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and ´ L. Van Gool. One-shot video object segmentation. In CVPR, 2017. 3, 8

[11]J. Chang, D. Wei, and J. W. Fisher III. A video representation using temporal superpixels. In CVPR, 2013. 2, 3

[12]J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In ECCV, 2016. 2, 3, 9, 10

[13]J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In CVPR, 2015. 3

[14]J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In CVPR, 2016. 2, 3, 5, 6, 9, 11

[15] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object detection via region based fully convolutional networks. In NIPS, 2016. 2 [11] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool. Real-time facial feature detection using conditional regression forests. In CVPR, 2012. 3

[16] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In BMVC, 2014. 8

[17] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. Jumpcut: Non-successive mask transfer and interpolation for video cutout. ACM Trans. Graph., 34(6), 2015. 2, 3 [14] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. TPAMI, 35(8):1915–1929, 2013. 2, 3

[18] M. Grundmann, V. Kwatra, M. Han, and I. A. Essa. Efficient hierarchical graph-based video segmentation. In CVPR, 2010. 1, 2, 3

[19] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. Joint 3d scene reconstruction and class segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 97–104, 2013. 3

[20] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In ECCV, 2014. 3

[21] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik. Hypercolumns for ´ object segmentation and fine-grained localization. In CVPR, 2015. 2, 4, 5