International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 06 | June 2021 www.irjet.net p-ISSN: 2395-0072

# Automatic Image Annotation and Autonomous Vehicles

**Aditi Modi[1], Mishma Toppo[2], Nidhi S Nair[3], P N Jayanthi[4]**

*[1, 2, 3]Students, Dept. of Electronics and Communication, R V College of Engineering*
*[4]Professor, Dept. of Electronics and Communication, R V College of Engineering, Bangalore, India*

---

**Abstract -** *Autonomous driving technologies have long since progressed from wireless control signals and electronically guided vehicle paths to a true self autonomy. This has been made possible through momentous developments in computer vision and deep learning technologies, allowing machines to better mimic human behaviour. Automatic image annotation is one of many computer vision sub-domains that have been a key part of this growth. It serves as a training tool as well as an important control mechanics in modern driverless systems. It allows on-board vehicular control systems to perceive the vehicle's surroundings in a manner similar a human - by recognising what an object is and clustering it into its closest classes. The paper is intended to serve as a brief look into the functioning, architecture and capabilities of two commonly implemented deep learning models for visual computer learning.*

***Key Words***: *Computer Vision, Automatic Image Annotation, Autonomous Vehicles, YOLO, Mask R-CNN, Object Detection, Instance Segmentation.*

## 1.INTRODUCTION

Computer vision is how a machine is able to accept, process and interpret the information in its vicinity. It finds application in many autonomous technologies, particularly in the manufacturing and automotive industries, where either a single or combination of its many sub-domains are employed as part of regular operations - these are object classification, object localisation, object detection, semantic segmentation and instance segmentation. Any of these processes can be boiled down to a model that relates input visual data to the state of its surroundings, a learning algorithm that can closely (as possible) describe the relation between the two and an inference algorithm that will use the trained model to make new decisions [1].

In an autonomous vehicle context, computer vision is the core of many Advanced Driver Assist Systems (ADAS) that are being developed with the promise of preventing accidents, reducing emissions, transporting the mobility-impaired and reducing driving related stress [2].

## 2. IMAGE ANNOTATION

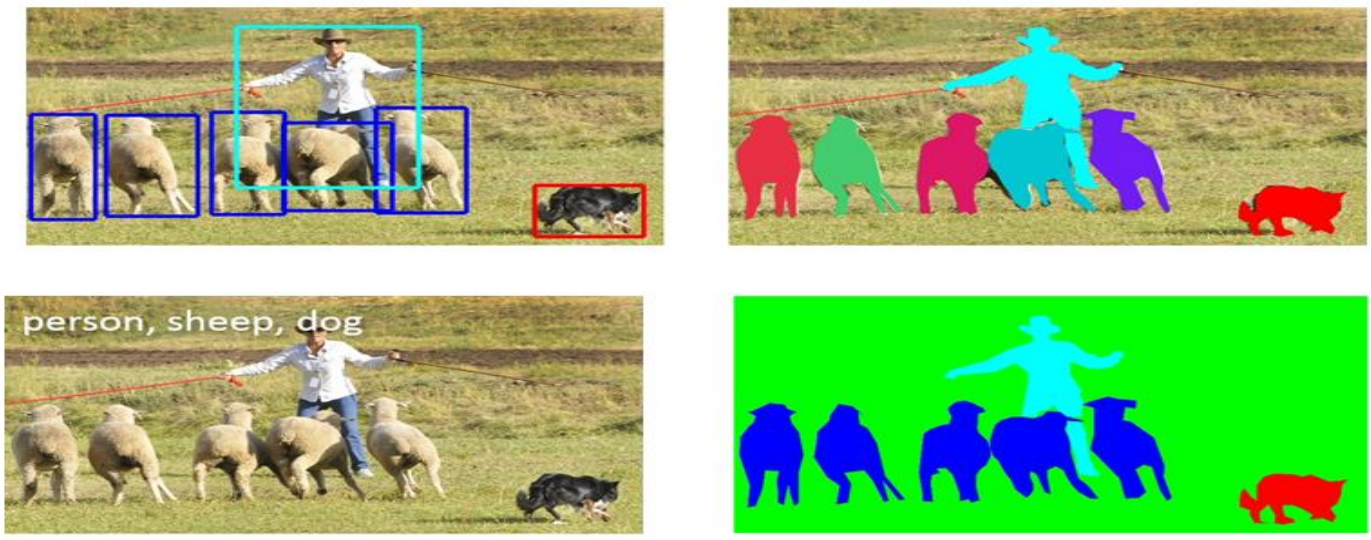


**Fig -1**: Image annotation types, example taken from Linet, al's paper on the development of the MS-COCO dataset [6]

Image annotation is the process by which input visual data is prepared for computer vision processes. It is a mostly human-powered task that assigns descriptive metadata to images, as labels or masks using tools like the VGG Image

Annotator [3], [4] and LabelMe [5]. Annotations are developed in-accordance to their usage in a mostly supervised machine learning scenario. Common image annotation types are image classification, 2D and 3D bounding boxes for objects, lines and splines for roads, polygons for irregular shapes and semantic segmentation for pixel-by-pixel marking of irregular and regular shapes. Image classification is the easiest of all annotations (figure (1a)) where single object images are assigned labels. Bounding boxes require annotators to draw rectangles around localised objects within images (figure 1b). Instance segmentation (figure 1d) is pixel-level segmentation of objects within an image. A variety of databases comprised of different annotation types, each serving as learning data for different end-use models. Some common examples are-

- Scene Understanding (SUN) database [7] and ImageNet database [8], [9] for image classification and computer vision models.
- Microsoft Common Objects in Context (MSCOCO) [6] for instance segmentation (figure 1d),
- PASCAL Visual Object Classes (VOC) [10] for bounding box data, and
- Cityscapes Dataset [11] for panoptic and semantic segmentation for autonomous vehicles.

### 2.1 Manual Annotation and Automatic Image Annotation

The process of image annotation begins with manual annotation, which is done by adding metadata to the image manually using various tools. This process is time consuming and tedious, especially when the set of images that need annotation is extremely large, having a person annotate is a cumbersome task. Such a model is also prone to human errors which can cause a depreciation in the accuracy. In order to overcome the issues in manual annotation, automatic annotation is taken into study. AI engineers predetermine the labels for the model manually using special image annotation software or tools, which involves defining regions in an image and creating text-based descriptions for each image annotated. Automatic image annotation is done after a training dataset is created, through manual annotation.

## 3. RELATED WORK

### 3.1 3D Object Detection

The model in taken into consideration for object detection was Alexey B's YOLOv4 [12], which is an extension of Redmon's YOLOv3 [13]. YOLO, also known as You Only Look Once, was developed by Joseph Redmon. The introduction of the YOLO real-time object detection is the cornerstone of image processing and recognition. This led to the development of faster and more accurate computer vision models.

### 3.2 Semantic Segmentation

Mask R-CNN, presented as a "conceptually simple, flexible and general framework for object instance segmentation" in the 2018 paper by He, et. al [14] is an extension of the Faster R-CNN model [15] with an added branch to predict segmentation masks in parallel with the existing branch for object classification and bounding box regression. This paper adapts and builds on W. Abdulla's [16] implementation of Mask R-CNN.

## 4. ARCHITECTURE AND COMPONENTS

### 4.1 YOLO (You Only Look Once)

YOLO is a real-time object recognition model that can detect multiple objects in a single frame. It is a single convolutional network that divides the input image into an S x S grid and predicts multiple bounding boxes and class confidence for each box. YOLO, unlike other detection models trains directly on the entire image, which optimises the detection performance, this makes YOLO extremely fast and the requirement of a complex pipeline is eliminated. The feature map of the YOLO output layer is designed to produce bounding box coordinates, the confidence score, and the class scores as outputs, and thus YOLO enables the detection of multiple objects with a single frame. Hence making the detection speed much faster than that of conventional methods. The processing of the grid unit causes localization errors to be large and the detection accuracy to be low, and thus making it unsuitable for autonomous driving applications. The solutions to these problems came with the introduction of YOLOv2 [17].YOLOv2 improves the detection accuracy compared to YOLO by using batch normalization for the convolution layer, and applying an anchor box, multi-scale training, and fine-grained features. The

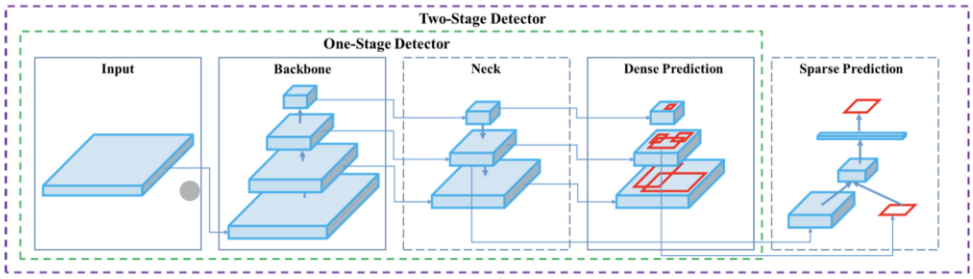


**Fig. 2**: Object detector [12].

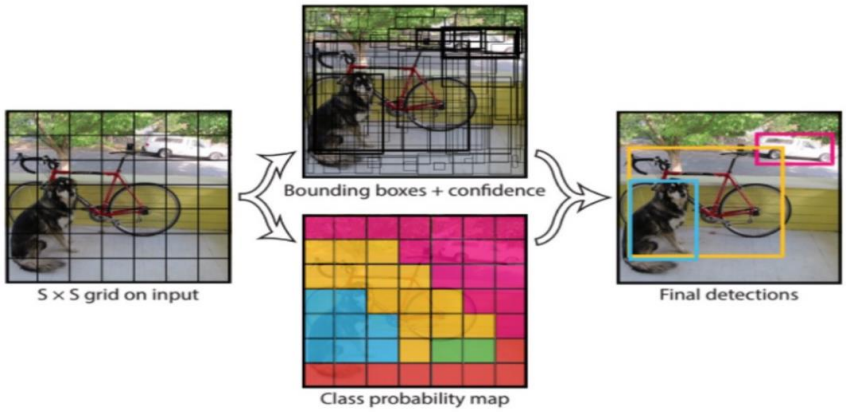


**Fig. 2**: S x S grid and predictions of bounding box for each cell [18].

YOLOv2, even though improved it failed to produce results that were acceptable for autonomous driving as detection of smaller objects such as traffic signs and far away objects were neglected. The model architecture consists of 24 convolutional layers followed by 4 pooling layers and 2 fully connected layers.[17] It applies the Leaky ReLu activation function after all layers except for the last one and uses dropout between the two fully connected layers in order to tackle over-fitting. Yolov3 model that was proposed improved the accuracy to great measures, this model's most prominent feature is the detection at three different scales in a similar manner as used in a feature pyramid network[19]. YOLOv4 was later introduced as a new and improved version of YOLO. The backbone architecture was upgraded and the modifications in the neck of the architecture lead the a 10% increase in the mAP (mean average precision) and a 12% increase in the FPS (frames per second) value[12]. From the paper [12] we can infer, the main goal was to design a faster and accurate model for object detection that can be trained and tested on any GPU to achieve real-time, high quality and convincing results.

### 4.2 Mask R-CNN

Mask R-CNN was introduced by He, *et. al* [14] as an improvement over existing visual deep learning architectures for region-based convolutional neural networks - R-CNN [21], Fast R-CNN [20] and Faster R-CNN [15]. The original R- CNN architecture used a selective search algorithm that would generate upto 2000 proposal regions for each image on which convolutional networks were run, which made this model very slow. Fast R-CNN improved on this by feeding the images directly to the network generating a convolutional feature map. Region proposals were then identified, reshaped into a fixed size when passed through a Region of Interest (RoI) pooling layer (see 4b) and fed into the fully connected layer. A final softmax activation [22] layer tightens the bounding box around the detected object. Faster R-CNN replaced selective search with a region proposal network that generated feature maps of smaller dimensions called anchors. Detected objects within anchor regions were considered only if it had an Intersection over Union (IoU) [23] of over 0.7 and discarded if it has on IoU of 0.3 or under. Acceptable anchor regions were then fed into the RoI pooling layer, then to the fully connected layer for softmax classification and bounding box regression. Mask R-CNN uses Faster R-CNN architecture for bounding box regression and adds to it a parallel operating branch that predicts what pixel in the image is part of the detected object. Mask generation is performed by a fully connected network (FCN) [22] that is applied on each region of interest in the image. An RoiAlign layer is introduced to accurately overlay extracted features from each RoI feature map over the input image.

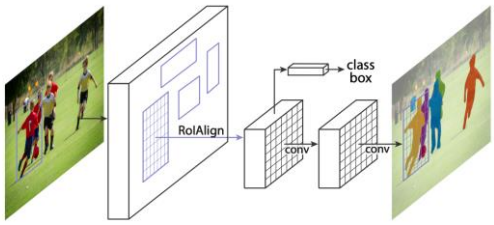


**Fig -4 (a)**: Mask R-CNN framework for instance segmentation [14]

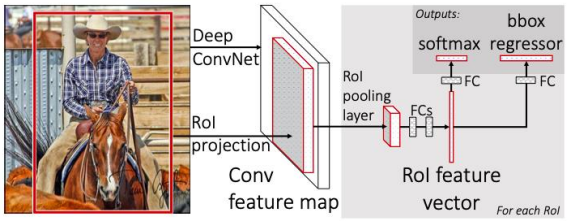


**Fig -4 (b)**: Fast R-CNN architecture for object detection. [20]

**Fig -4**: Instance segmentation and object detection framework used in Mask R-CNN.

## 5. SIMULATION AND DISCUSSION

The first stage of the project was to test images using the MS COCO [6] pre-trained weights, images from the Indian road driving dataset that was created was used to test the pre- trained YOLOv4 model. The model seemed to do well up until the objects were far ahead. The COCO dataset consists of 80 odd classes out of which 8 are relevant to autonomous driving, hence the urge to remove redundant classes and train the model on a separate, more relevant dataset, came up. Open images dataset was released by Google and has over 9 million images with both detection and segmentation masks [24] which can be used to train and validate models. The second stage of the project began with the training of the model, we shortlisted the classes to person, car, truck, traffic light, stop sign, motorcycle, bicycle and train, which are common road objects that can be seen. The pre-trained COCO layers where used to transfer learn and the model was set to train. After the training was completed, as seen in figure 5, we achieved an average training loss of 2.5. The new trained YOLOv4 model produced an mAP of 73.81\%. When training the Mask R-CNN model on the MSCOCO dataset, using the code and weights provided by W. Abdulla [16], we achieved an mAP of 53% . The model was then used to test images from the IDD dataset [25]. The model, then trained on images from the IDD and then tested, yielded an mAP of 71.1%.

| ClassId | Name | AP | TP | FP |
|---|---|---|---|---|
| 0 | auto | 98.38% | 122 | 2 |
| 1 | bus | 100% | 23 | 0 |
| 2 | car | 99.61% | 282 | 4 |
| 3 | rider | 98.82% | 217 | 12 |
| 4 | motorcycle | 97.54% | 317 | 6 |
| 5 | person | 99.37% | 157 | 6 |
| 6 | truck | 100% | 81 | 5 |
| 7 | traffic sign | 100% | 12 | 0 |
| 8 | bicycle | 100% | 2 | 0 |
| 9 | ambulance | 100% | 1 | 0 |
| 10 | traffic light | 0% | 0 | 0 |

**Table -1:** AP obtained per-class after training YOLOv4 on Open Images Dataset.

| | |
|---|---|
| Conf_threshold | 0.25 |
| Precision | 0.98 |
| Recall | 0.98 |
| F1-score | 0.98 |
| TP | 1214 |
| FP | 30 |
| FN | 19 |
| Avg IoU | 92.49 |
| mAP | 90.34% |

**Table -2:** mAP for the above results.

**Fig -5 (a)**. RoI proposals, **5 (b)** Bounding box regression.

**Fig -5 (c)**. Mask visualisation**.**

**Fig -5 (d)** Detection output.

**Fig -5**: Mask R-CNN operations visualised [16].

**Fig -5**: Detection results for YOLOv4.

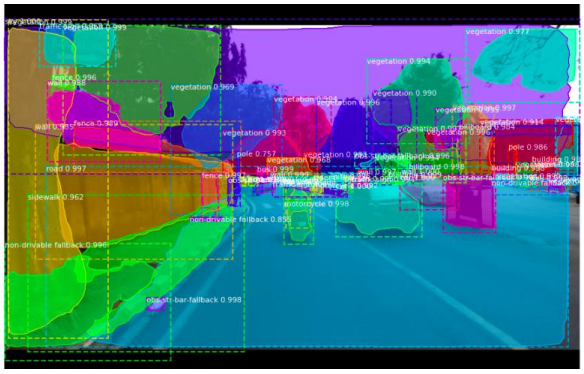

**Fig -5**: Detection results for Mask R-CNN.

## 6. CONCLUSION

Automatic image annotation is a technology that is a part of computer vision, it helps the machine see and assess the surroundings if not better but like us humans. There are many models and algorithms based on image annotation, how to judge which model is superior to the other is the major problem. In this paper, we describe object detection algorithms YOLOv4 and MASK R-CNN, which are considered to be some of the top performing models in object detection and classification. Though each model has its own strengths, a noticeable matter is that each model has its own constraints and issues. The models are compared based on the mAP and FPS value with pre-trained and trained weights.

The YOLOv4 model stands for speed and gives a stiff competition to the Mask R-CNN model which is able to semantically segment images along with creating a bounding box.

## REFERENCES

1. S. Prince, Computer Vision: Models Learning and Inference. Cambridge University Press, 2012.
2. T. J. Crayton and B. M. Meier, "Autonomous vehicles: Developing a public health research agenda to frame the future of transportation policy," Journal of Transport & Health, vol. 6, pp. 245–252, 2017, ISSN: 2214-1405. DOI: https : / / doi . org / 10 . 1016 / j . jth . 2017 . 04 . 004.[Online]. Available: https : / / www . sciencedirect . com / science/article/pii/S2214140517300014.
3. A. Dutta and A. Zisserman, "The VIA annotation software for images, audio and video," in Proceedings of the 27th ACM International Conference on Multimedia, ser. MM '19, Nice, France: ACM, 2019, ISBN: 978-1-4503-6889-6/19/10. DOI: 10 . 1145 / 3343031 . 3350535.[Online]. Available: https://doi.org/10.1145/3343031. 3350535.
4. A. Dutta, A. Gupta, and A. Zissermann, VGG image annotator (VIA), Version: 2.0.11, Accessed: 19/05/2021, 2016. [Online]. Available: http://www.robots.ox.ac.uk/vgg/software/via/.
5. B. Russell, A. Torralba, K. Murphy, and W. Freeman, "Labelme: A database and web-based tool for image annotation," en, International Journal of Computer Vision, vol. 77, Numbers 1-3, pp. 157–173, May 2008.
6. T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," CoRR, vol. abs/1405.0312, 2014.arXiv: 1405.0312. [Online]. Available: http://arxiv.org/abs/1405.0312.

7. J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 3485–3492. DOI: 10.1109/CVPR.2010.5539970.
8. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," en, in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009. DOI: 10 . 1109 /cvpr.2009.5206848.
9. G. A. Miller, "Wordnet: A lexical database for English," en, Communications of the ACM, vol. 38, no. 11, pp. 39–41, 1995.
10. M. Everingham, L. Van Gool, and C. Williams, "The pascal visual object classes (voc) challenge," en, Int J Comput Vis, vol. 88, pp. 303–338, 2010. DOI: 10.1007/s11263 - 009 - 0275 - 4. [Online]. Available: https : / / doi .org/10.1007/s11263-009-0275-4.
11. M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," CoRR, vol. abs/1604.01685, 2016. arXiv:1604 . 01685. [Online]. Available: http : / /arxiv.org/abs/1604.01685.
12. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," CoRR, vol. abs/2004.10934, 2020. arXiv: 2004.10934. [Online]. Available: https://arxiv.org/abs/2004.10934.
13. J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv, 2018.
14. K. He, G. Gkioxari, P. Dollár, and R. B. Girshick,"Mask R-CNN," CoRR, vol. abs/1703.06870, 2017. arXiv: 1703 . 06870. [Online]. Available: http : / / arxiv.org/abs/1703.06870.
15. S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," CoRR, vol. abs/1506.01497, 2015. arXiv: 1506.01497. [Online]. Available: http://arxiv.org/abs/1506.01497.
16. W. Abdulla, Mask r-cnn for object detection and instance segmentation on keras and tensorflow, https : / /
17. github.com/matterport/MaskRCNN, 2017.
18. J. Redmon and A. Farhadi, Yolo9000: Better, faster, stronger, 2016. arXiv: 1612.08242[cs.CV].
19. J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," CoRR, vol. abs/1506.02640, 2015. arXiv:1506.02640. [Online]. Available: http://arxiv.org/abs/1506.02640.
20. T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," CoRR, vol. abs/1612.03144, 2016. arXiv: 1612 . 03144. [Online]. Available: http : / / arxiv.org/abs/1612.03144.
21. R. Girshick, Fast r-cnn, 2015. arXiv: 1504 . 08083[cs.CV].
22. R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," CoRR, vol. abs/1311.2524, 2013. arXiv: 1311 . 2524. [Online]. Available: http : / /arxiv.org/abs/1311.2524.
23. A. Amidi and S. Amidi, CS 230 - Convolutional Neural Networks Cheatsheet. [Online]. Available: https : / /stanford . edu /~shervine / teaching / cs - 230 / cheatsheet -convolutional-neural-networks (visited on 05/30/2021).
24. A. Rosebrock, Intersection over Union (IoU) for object detection, en-US, Nov. 2016. [Online]. Available: https:/ / www. pyimagesearch. com / 2016 / 11 / 07 / intersection -over - union - iou - for - object - detection/ (visited on 05/30/2021).
25. A. Kuznetsova, H. Rom, N. Alldrin, J. R. R. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, and V. Ferrari, "The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale," CoRR, vol. abs/1811.00982, 2018. arXiv: 1811.00982. [Online]. Available: http://arxiv.org/abs/1811.00982.
26. G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, and C. Jawahar, en, in IDD: A Dataset for Exploring Problems of Autonomous Navigation in Un-constrained Environments - IEEE Winter Conf. on Applications of Computer Vision (WACV), 2019. [Online]. Available: http://idd.insaan.iiit.ac.in/.