# A Survey of Deep Learning Object Detection Models for Business Intelligence Applications

**Vinay Kumar T M[1], Varun Kumar V[2], T Nikhil[3], Sameeksha M[4]**

[1,2,3,4]Student, Department of Computer Science and Engineering, Dayananda Sagar College of Engineering Bangalore, India.

**Abstract**

The ever-rising applications of Business Intelligence techniques in the present world demands integration with other deep learning techniques such as Object Detection, Natural Language Processing, etc. With the inclusion of Object Detection, business intelligence can provide intelligible insights into make business experience better. With a comprehensive elucidation of training time complexities and the multiple factors that play a vital role in its variation, this paper aims to provide a report of object detection techniques. The single-stage and two-stage detectors are separately taken into consideration, while explaining the pertinence of its use cases.

**Keywords:** Object Detection, Convolutional Neural Network, Training Time, Business Intelligence

## 1. Introduction

Business Intelligence(BI)[1] is an ever-growing field that is exceptionally dependent on data collection, data manipulation, data analysis and inference on data. Deep Learning[2] and Business Intelligence go hand-in-hand to revolutionize the use of data to yield inventions that can change the world. The field of BI, wherein technological manoeuvres have created wonders through data-driven decision making, still strives for refinement that can enhance customer satisfaction and ensure global security, to name some examples. Such systems, on which a large number of the populace is dependent, need data that is guaranteed to be authentic and most importantly, precise. Not even an instance of fault can be tolerated, as it can lead to catastrophic consequences, truncating data dependency of large and small organizations.

Data science has enriched business intelligence in that it volunteers insights and mechanisms to surpass the best existing technologies in the contemporary world through data collection, data warehousing, and data analytics. Though these methods provide robust results for concerns that prevail in the current world of BI, there are certain drawbacks. The fundamental drawback that can be deduced is the lack of large scale data during the development stage. Data inadequacy can prove to be a hard lined impediment in security ascertainment, or any other business intelligence goal for that matter, as it impedes the performance of data-dependent systems. Given enough data, wonders can be achieved by leveraging the techniques of Deep Learning and Machine Learning.

Technological enhancements throughout the years have also paved the way for an exponential growth of cyber crimes. To the contrary, they have manifested innovative solutions to these problems as well. In the present world of data-driven digital systems, Deep Learning plays a vital role in gathering methodologies that can put the crime rates to an all-time low, and at the same enhance the efficiency of said systems. That includes analysis of text-based applications[3] to apprehend derogatory remarks and hate crimes; verification of images for fowl and macabre contents that may harm sentiments of the people.

Object detection, most of all, has a lot to offer in the field of business intelligence as it helps organizations gain insights using visual data, either in the form of images or videos. For example, object detection can prevent leakage of confidential information by keeping a track of suspicious accounts on the Internet and taking necessary actions that might mitigate the consequences. Video Surveillance can be reinforced with object detection for anomaly detection, thereby subsidising crime and improving security measures. Customer Behaviour Analysis can also be performed by tracking the brands and endorsements customers follow on a regular basis to implicitly discern their likes and dislikes to offer recommendations.

Within the vast multitude of Object Detection techniques available, it is prerequisite to understand the applicability and practicality of each one of them in different Business Intelligence scenarios. Some of them may require a fast technique, with a compromise for accuracy, while others may need the opposite. There are innumerable techniques of Object Detection that differ in the speed and accuracy with which they perform. Selecting a suitable technique hence becomes an absolutely meticulous task that demands comprehensive knowledge about each technique and the nuanced differences between others. This research paper presents a comparative analysis of the various object detection techniques to infer their speed, accuracy and efficiency in different use cases and design recommendations for unique scenarios.

At the outset, applications of object detection in BI are illuminated upon with pertinent examples, emphasizing their roles in inventory management, security, quality control and what's more. It is imperative that knowledge of how object detection exerts its influence in BI be conveyed to underline its importance. Following the introduction, the datasets, such as COCO, ILSVRC and PASCAL VOC([4],[5],[6]), used to train high-precision object detection techniques are described along with the conventional metrics used to measure the efficacy of object detection models. With common metrics, it becomes convenient to compare and contrast the techniques and draw conclusions based on it. Precision and recall are the intrinsic metrics used in almost every dataset for object detection and inferences on the same are made in the succeeding sections. Furthermore, the impact of using other metrics and their drawbacks are also borne with. The next section includes numerous object detection techniques such as R-CNN, SSD, YOLO ([7],[8],[9]) etc., amongst other variations and novelties for analytical manifestation of how distinctive every one of them are. Performance in terms of speed, training-time and inference time along with memory requirements are analysed in detail. How the use of different algorithms at each step of the detection pipeline makes a substantial influence on the techniques' time and space efficiency are also discussed. The variations in optimization techniques, hardware accelerators and their impact on performance are studied to evaluate the contrast between the methods.

Based on the comparative analysis that follows, inference regarding the merits of each method are elucidated and relevant recommendations are contributed in aiding a selection that is suitable for business intelligence use cases. Concomitantly, the trade-off between training time and model performance is reviewed upon to get a better understanding of the factors that actuate them. It can be concluded that the fine-tuning of neural networks used in training an object detection technique is what brings about the differences, and that with proper fine-tuning, wonders can be achieved in the field of object detection. Errors that are quite frequent in undermining performance can be circumvented through scrutiny are mentioned and explained[10]. The paper also consists of a broad study of the shortcomings of different methods and advances to rectify them[11]. As a result, we conclude with the pros and cons of the methods and review their suitability for business intelligence.

## 2. Two-Step Methods

### 2.1 R-CNN

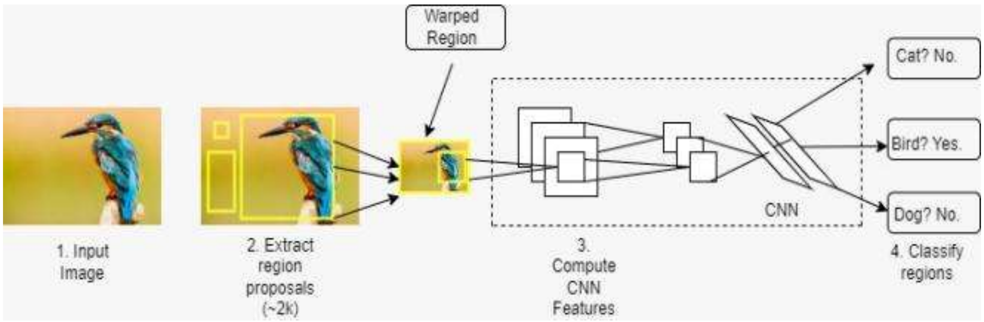
R-CNN[7] introduced the concept of two-stage object detectors: the first stage being region proposals and the second being non-maximum suppression[12] to correctly identify objects and draw bounding boxes around them. Using Krizhevsky's[13] Convolutional model as a backbone network for feature extraction and supervised learning on image-level annotations, R-CNN was able to achieve an mAP(mean Average Precision) of 53.3% on the PASCAL VOC dataset. Figure 1 is a sample image of how R-CNN pipeline works.

**2.1.1 Region Proposals Module** High resolution images of order ~1000X600 are provided as input. Selective Search[9] is used to generate around 2000 region proposals for each image in about 2 seconds. The grouping of proposals is continued till an entire region is a proposal hence leading to proposals of various sizes. Selective search[14] produces drastically low but accurate object proposals than other exhaustive searches and is thus computationally very efficient and saves time.

**2.1.2 Object Detection Module** The images studded with region proposals are fed as input after converting them into the size of $227^{X}227$. Extraction of a 4096-dimensional feature vector from each region proposal is conducted using Caffe implementation of CNN[13]. During compression, the pixels are warped into a tight bounding box in such a way that there are exactly p pixels in each box (p=16, generally).

**2.1.3 Classification Model** SVMs are trained for each class which are used to generate respective scores. Considering these obtained scores, a greedy non-maximum suppression is applied to reject a region if it has Intersection over Union overlap lesser than the threshold value of 0.5.

**Figure 1: R-CNN System Computes Region Proposals And Classifies Which Class The Object Belongs To**



The high performance of the technique is attributed to the feature sharing among the variety of categories during training. Additionally, features extracted are low-dimensional compared to the UVA detection system[15] which uses features of an order of 360K as opposed to those of 4K in R-CNN. Moreover, the only class-specific computations are between features, SVM weights and non-maximum suppression[28], signifying that R-CNN can scale to thousands of object classes without the mAP perishing.

The 1000-way classification layer of the CNN is replaced with a randomly initialized (N+1)-way classification layer where N is the number of classes, plus 1 for background. For PASCAL VOC dataset

[6], N=20 and for ILSVRC 2013 dataset[5], N=200.

The bounding box regression method is very much similar to that of deformable parts models[10] with a key difference in that R-CNN uses features extracted from CNN and not DPM([16],[17]) part locations.

R-CNN achieved an mAP score of 53.3% on PASCAL VOC 2011/12 dataset with N=20 and 31.4% mAP on the ILSVRC2013 dataset with N=200. Using [18] as baseline improved the mAP to 66.0%, but took almost 7 times more computation power, hence the decline in efficiency.

The time taken to compute region proposals and features per image is roughly 13s on a GPU and 53s on a CPU. The memory requirements are also dropped down by 90 times using GPU. 10K detectors manage to run in about a minute on a CPU using R-CNN. Fine-tuning on the ILSVRC dataset took 13 hours using Caffe on the NVIDIA Tesla K20.

**Drawbacks of R-CNN**

1. R-CNN follows a pipeline procedure where it first fine tunes a ConvNet[19] using log loss, then SVMs are fit on these ConvNets, which act as object detectors followed by bounding-box regression; which is a tedious and time-consuming task.
2. The input shape of images necessarily need to be of resolution 227X227 in order to be compatible with the first layer of the CNN.
3. Training is time and space expensive.
4. Despite selective search being a well sought out method for region proposals, the myriad of region proposals are redundant and time consuming.

## 2.2 SPP-NET

The R-CNN technique had some major shortcomings; one of which is that the fully connected layers of the CNN required the input be provided in a fixed shape. But the Convolutional Neural Network backbone of Krizhevsky[13] produces output in different shapes and sizes. As a solution, R-CNN warps the images into a fixed bounding box and then conducts the forward pass of the neural network. This method of warping might lead to geometric distortion of pixels and might prove to be a substantial impediment during object detection.

The Spatial Pyramid Pooling layer[20] uses pooling layers at the top of the last layer of the convolutional network to avoid this problem. As a result, state-of-the-art accuracies using only a single full-image representation and no fine tuning have been achieved. The pooling layer takes the features extracted by the CNN as input and generates fixed-length representations, which are used as input by the fully connected layers for further predictions, subduing the necessity of additional computations. While training, this helps reduce over-fitting and increases scale-invariance. The maintenance of spatial information by pooling in local spatial bins makes SPP superior over the Bag of Words[9] method. The advantage of multi-level pooling is due to its robustness to the variance in object deformations and spatial layout, and not because it has more parameters.

SPP-Net has achieved state-of-the-art classification accuracies on Caltech101 and Pascal VOC 2007 datasets. The 5-scale result on Object Detection was 59.2%, which is significantly better than that of R-CNN with immense speed improvement which comes by running the convolutional network only one on each image of the dataset in contrast to the repeated application on every warped region in R-CNN. Feature extraction of 1-scale images takes ~0.05s, compared to the 14s in R-CNN. The overall testing time is ~0.5s. Use of different backbone architectures lead to slight changes in training and testing time.

EdgeBoxes[21] method computes the region proposals quicker than Selective Search[14], but is not up to par with accuracy.

## 2.3 Fast R-CNN

Though SPP-Net obviates the necessity of fixed-size inputs for the fully connected layers, it still maintains the complex pipeline of R-CNN, and hence the time complexity. The fine-tuning algorithm proposed in SPP-Net cannot update the CNNs that precede the spatial pyramid pooling layer.

Fast R-CNN[22] technique presents a new method of training the Convolutional neural networks where training is single-staged using a multi-task loss. Convolutional feature maps are first generated by the CNN which takes image and object proposals as input. For each object proposal, the RoI(region of interest) pooling layer extracts a fixed-length feature vector using the feature maps. The output layer is designed as (K+1), which is K object classes and one for background, which outputs four real-valued numbers for each of the K object classes. Each RoI is defined by a four-tuple (r, c, h, w) that specifies its top-left corner (r, c) and its height and width (h, w). Pooling is applied to each feature map channel independently. On the very deep model of VGG-net, Fast R-CNN trains 9 times faster and is 213 times faster at test time that R-CNN. A top accuracy on PASCAL VOC dataset of 66% is achieved. The speed lag in R-CNN was overcome in Fast R-CNN by computing the feature maps only once per image and sharing features while training.

Since Selective Search[14] generates large number of region proposals, it is imperative to use a speed booster such as Single Value Decomposition(SVD). The use of SVD gives a simple compression to the network and boosts the speed when RoIs are large in number. Such intense remodelling reduces the training time from 84 hours for that of R-CNN with VGG-Net as backbone to 9.5 hours for Fast R-CNN. Fast R-CNN is almost three times faster than SPP-Net and ten times with the truncated SVD integrated to it. Truncated SVD reduces detection time by nearly 30% without severely impacting the mAP, and without fine-tuning. Yet, multi-scale approach to training increases mAP by only a small amount at a larger cost of compute time.

## 2.4 Faster R-CNN

Faster R-CNN[23] originated from the inception of Region Proposal Networks(RPN) that share full-image convolutional features with the detection network, thus enabling near cost-free region proposals. An RPN is simply a convolutional network which is delegated the job of predicting object bounds and object-ness scores simultaneously. It is trained end-to-end to generate high-quality region proposals for Fast R-CNN to detect objects. Using attention mechanisms, the RPN tells the object detector where to look in the image. The Faster R-CNN works with only 300 object proposals as opposed to the 2000 in R-CNN. For computation of 2000 proposals, it takes only 300ms for RPN, but 2s for Selective search.

The artfully low marginal cost for computing proposals can be attributed to the feature sharing between convolutions at test-time. Novel “anchor” boxes were introduced to serve as references at multiple scales and aspect ratios. An anchor is centred at the sliding window and is associated with a scale and aspect ratio; which by default is assumed to be 3 and 3, respectively, yielding 9 anchors at each sliding position. These anchors are translation-invariant which implies that they are capable of detecting objects irrespective of their locations in the image. While SPP-Net[20] uses pyramid pooling to deal with the problem of multi-scale images, Faster R-CNN[23] uses a ‘pyramid of anchors’ technique which results in sliding windows of multiple scales and sizes that annuls the necessity of pyramid pooling.

The technique used to unify RPN and Fast R-CNN as detection network is alternative training, which alternates between fine-tuning for the region proposal task and fine-tuning for object detection, while keeping the proposals fixed. Another way would have been to join the two networks during training (Approximate joint training) that involves SGD (Stochastic Gradient Descent) and back propagation. Regardless of it producing near same results as the Fast R-CNN, the method reduces training time by 25-50%. The RPN method takes only about 10ms to generate proposals per image. Approximate joint training reduces the training time by 25-50%. The whole system takes up nearly 200ms for both proposals and detection. On the COCO dataset, the testing time is also nearly 200ms per image.

mAP of 70.4% was achieved on the PASCAL VOC test set using Faster R-CNN. On the union of PASCAL VOC 2007 trainval and 2012 trainval the mAP is 73.2%.

## 2.5 Miscellaneous

### 2.5.1 R-FCN

R-FCN[24] proposed position-sensitive feature maps to address the dilemma between translation-invariance in image classification and translation-variance in object detection.Using the 101-layer ResNet[18] architecture as backbone, R-FCN yields an mAP of 83.6% on the PASCAL VOC 2007 and 82.0% on the PASCAL VOC 2012 dataset at a test time speed of 170ms per image, which is faster than that of Faster R-CNN[23]. R-FCN is almost 2.5 times faster than Faster R-CNN at test time and 20 times faster on the COCO dataset at 53.2% mAP.

### 2.5.2 Cascade R-CNN

Cascade R-CNN[25] consists of sequence of object detectors trained with increasing IOU thresholds to be sequentially more selective against close false positives during inference, was proposed to address the dilemma caused by noisy detections and performance degradation caused by increasing IoU thresholds. It is evaluated on the MS-COCO dataset with an AP threshold ranging from 0.5 to 0.95 with an interval of 0.05.

Experiments were performed with three popular baseline detectors: Faster-RCNN with backbone VGG-Net, R-FCN and FPN with ResNet[18] backbone. Compared to the state-of-the-art detectors like Faster R-CNN etc., Cascade R-CNN has about 50 million more parameters in terms of architectural design due to the increase in cascade stages and therefore takes up more per image training and testing time(difference is in the order of 10^-2 seconds). A significant rise in mAP is observed with Cascade R-CNN because of the reduction of false positive occurances. On the COCO test-dev, Cascade R-CNN achieved an mAP of 42.8%, surpassing the state-of-the-art detectors.

### 2.5.3 Feature Pyramid Networks

FPN[26] is a top-down architecture with lateral connections developed to build high-level semantic feature maps at all scales. The run-time is nearly 6 FPS on a GPU. A minimum of 7 point increase in AP as compared to Faster R-CNN is observed. The postulation of feature sharing increases training time by almost twice, but reduces test-time.

Inference time was monitored to be 0.148s and 0.172s for Resnet-50 and Resnet-101[18], respectively. In spite of having a lighter overhead, a small extra cost is introduced from the extra layers of the Feature Pyramid Network. Table 1 gives a consolidated insight of the mAP of various object detection methods.

**Table 1: mAP comparison of Object detection methods.**

| Sl.no | Method | Backbone | mAP |
|---|---|---|---|
| 1 | Cascade R-CNN | Resnet-101 | 42.8% |
| 2 | YOLO | Resnet-101 | 52.7% |
| 3 | R-CNN | Resnet-101 | 58.5% |
| 4 | SSD | Resnet-101 | 60.9% |
| 5 | Fast R-CNN | Resnet-101 | 70.0% |
| 6 | Faster R-CNN | Resnet-101 | 73.2% |
| 7 | DSSD | Resnet-101 | 81.5% |
| 8 | Retina-net | Resnet-101 | 82.9% |

## 3 Single-Step Detectors

### 3.1 SSD

Single shot Multi-Box detector[8] is a single-stage object detector that eliminates the requirement of an independent region proposal mechanism by encapsulating all the computation within a single network. Not only does it simplify the training and increase the speed of detection, also demonstrates competitive results to two-step object detectors. SSD has better accuracy for input images of smaller size than those used in other single-stage detectors.

The model consists of a feed-forward convolutional network that produces fixed-size collections of bounding boxes and confidence scores for each box, followed by non-maximum suppression to produce the finalized detection results. Furthermore, convolutional layers are added to the base network to obtain predictions of detections at multiple scales. The default boxes play a role similar to that of anchor boxes used in Faster R-CNN[23], a subtle difference with the former being that, here, it is applied to several feature maps of different resolutions.

Unlike two-step methods which use intersection over Union, SSD uses jaccard overlap to match ground truth boxes to any default boxes above the threshold of 0.5. The training objective is derived from MultiBox[27] and is extended to handle multiple object categories. Utilizing feature maps from several different layers in a single network for predictions aids the handling objects of different scales while also sharing parameters across all object scales. An additional step of negative hard mining is encouraged, wherein the negative samples are sorted using the highest confidence loss for each box and the top ones are picked in such a way that the ratio between negatives and positives is at the most 3:1.

Ground truth information is assigned to specific outputs in a fixed set of detector outputs during training. Boxes with jaccard overlap greater than 0.5 are matched, which allows the network to predict high scores for multiple overlapping default boxes rather than requiring it to pick only the one with maximum overlap.

An mAP of 74.3% is achieved on PASCAL VOC 2007 test dataset at 59 fps for an input of 300X300 and 76.9% for 512X512. Using a faster base network can further ameliorate the speed and make the system robust to images of different resolutions as well. The high speed and commendable accuracy leads to the conclusion that speed vs accuracy trade-off is concretely improved by SSD[8].

### 3.2 DSSD

The deconvolutional single-shot detector[28] was introduced to augment high-level context knowledge into state-of-the-art single-stage object detectors. It is achieved by adding deconvolutional layers to the SSD network to improve accuracy and most importantly, detect objects of small scale. The deconvolutional layers increase the resolution of the feature maps after each layer of convolution.

The use of combined feature maps not only draw a substantial memory requirement but also impede the speed of a detector. The deconvolutional module is responsible for integrating the feature maps and the deconvolutional layers, thus improving speed and efficiency. The deconvolution module is inspired by Pinheiro, who suggested that a factored version of the deconvolution module for a refinement network has the same accuracy as a more complicated one and the network will be more efficient. Changes made to the module for the purpose of object detection include an addition of batch normalization layer after each convolutional layer and replacing the use of bilinear sampling with deconvolutional layer.

By keeping the training policy same as that of SSD, the VGG backbone is replaced with Residual-101. The deconvolutional module is trained first with the weights of the original SSD model's weights frozen, followed by fine-tuning the entire network. The Residual-101 model without the prediction module worked better than VGG for higher resolution input images. The replacement of VGG with Residual-101 barely makes an impact for small input images, but for large images attributed to spatial information. Adding the deconvolutional layer on top of the backbone improves the AP for smaller objects.

mAPs of 81.5% and 80.0% were achieved on the PASCAL VOC dataset of the years 2007 and 2012 respectively. An mAP of 33.2% was achieved on COCO, which thereby outperformed the state-of-the-art R-FCN[12]. Though it maintains a speed advantage over R-FCN, DSSD is slower than SSD. DSSD was able to present with such high performance only after exacting large training time. Hence, the lag in inference time. The increase in training time is because of the increase in the number of default bounding boxes.

### 3.3 YOLO

You Only Look Once(YOLO)[9] frames object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. The entire detection pipeline has been boiled down to a single network, which predicts bounding boxes and class probabilities directly from images in one evaluation; hence accentuating that end-to-end optimization can be done directly on the detection performance. The YOLO model is less likely to predict false positives for background compared to other state-of-the-art detectors, but makes more localization errors. The exclusion of a complex pipeline speeds up the detection process significantly. The base network performs at a speed of 45 fps with no batch processing on a Titan X GPU. Ever since, many YOLO upgradations have been proposed[29].

Yet another remarkable feature of YOLO is that it takes the image as a whole and reasons globally around it to implicitly encode contextual information about classes and their appearance. This does not happen in sliding-window based methods because they only look for objects within the region proposals, thus limiting global context. YOLO is a highly generalized method that does not break down when applied to new domains or unexpected objects. The only downside is that it trails in detection accuracy. In spite of achieving remarkable speed in detection, its accuracy is below par.

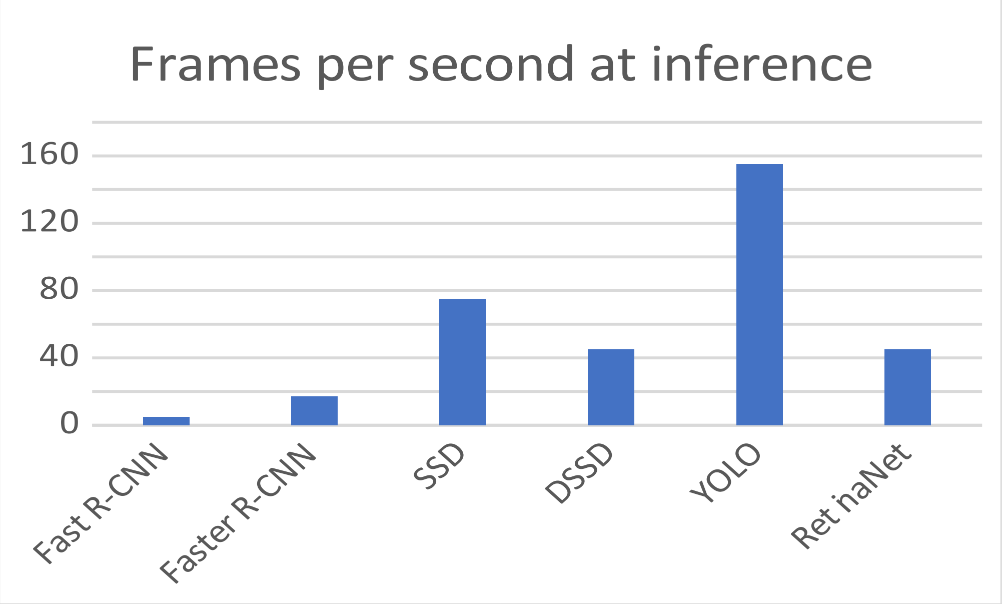
To remedy early divergence caused by the use of sum-squared error, the loss from bounding-box predictions is increased and the loss from confidence predictions is decreased. A multi-part loss function is to be optimized during training, which penalizes an error only if a grid cell contain an object.

YOLO struggles with grouped objects and objects of small size. A grid cell can predict only two bounding boxes and one class. Despite being good at generalization, it does not generalize well with new aspects and configurations. All types of errors are treated as the same by the loss function; which is not healthy for the performance because a small error in a large bounding box may be passable, but a large error in a small bounding box may be catastrophic.

Fast YOLO has an mAP of 52.7%, whereas YOLO(main) pushes it further to 63.4%. YOLO is the fastest known object detector but lags in accuracy. Despite having a greater speed of 155 FPS, Fast YOLO has a substandard mAP than that of SSD(52.7%). On combining with R-CNN, bounding boxes that are predicted to match the ground truth value in both the methods were boosted, mAP increased to 75%.

After training the network for a week, a single crop top-5 accuracy of 88% is achieved on the ImageNet 2012 validation set. On the PASCAL VOC dataset, only 98 bounding boxes and class probabilities per image are predicted. Figure 2 gives an insight of the processing speeds of different methods.

**Figure 2: Comparison of Frames Per Second Processing of Different Methods**



### 3.4 Corner-Net and Center-Net

Corner-Net [30] takes a different approach to object detection which identifies objects as a pair of key-points: top-left corner and bottom-right corner, particularly, hence eliminating the requirement of anchor boxes used in previous single-stage methods in order to compete with two-stage detectors. Heat-maps around the corner key-points are predicted using a single convolutional neural network along with an embedding vector for each detected corner. An additional corner-pooling approach to understand context was introduced to ameliorate the prediction of corners and their unification.

During training, the input and output image resolutions were set to 511X511 and 128X128, respectively, and an AP of nearly 40% was achieved. After random, customary data augmentation procedures, PCA(Principal Component Analysis) is applied to input images. While testing, after non-maximal suppression was applied, the top 100 top-left and bottom-right corners were selected for final predictions. Penalty for negative locations is reduced within a certain radius determined by the object that is predicted as positive, rather than keeping it high. The hourglass network as the backbone is crucial for the working of Corner-Net which developed bounding boxes of better quality viz.-a-viz. a better precision for a higher threshold of IoU.

Center-Net [31], which is an extension of Corner-Net detects each object as a triplet, rather than a pair of key-points improving both precision and recall. The design includes two customized modules called as cascade corner pooling and centre pooling which enrich the information collected for both top-left corners and bottom-right corners and provide more recognizable information at the centre regions, respectively.

The weak ability of Corner-Net to refer to the global information of the object leads to wrong estimation of object boundaries. CenterNet was designed to equip the CornerNet method with an additional key-point for detection which holds information about the centre region of a proposal. The intuition was that if a predicted bounding box has a high IoU with the ground truth value, then the probability that the centre key-point in its central region is predicted as the same class is high, and vice versa. This can be enforced during inference to verify whether that proposal belongs to the same class of object. For large objects, ‘centre pooling’ can extract richer internal visual patterns, which is abundant in larger objects, to inculcate centre knowledge. ‘Cascade corner pooling’ helps improve average recall(AR) and thus reduce incorrect bounding boxes by deducing boundary information.

The fault discovery(FD) rates(the complement of average precision and a representation of how many of the bounding boxes have an IoU less than threshold value) of several CornerNet IoUs were extensively researched to decide on a suitable IoU rate that minimizes FD. These rates when compared to Corner-Net dropped from ~30% to 4.5% in Center-Net. For small incorrect bounding boxes, the FD rates dropped from ~60% to 9%, and are accredited to the infusion of knowledge about the centres.

For inference, both original and horizontally flipped images with original resolutions are provided as input for the network. Bounding boxes detected in the horizontally flipped images are flipped back and mixed with those of the original image. The redundant boxes are deleted using soft nms###, then the top 100 bounding boxes are selected based on their scores as final detection results.

The input image size is set to 511X511, the same as Corner-Net, by using the Hourglass-52 backbone, single-scale testing AP improved by 3.8% and multi-scale testing improved by 4.1% than Corner-Net. The quality of improvement is mainly because of small objects, since centre information of small objects has been better reformed than that of larger objects with the Center-Net intuition. The reduction of incorrect bounding boxes improves average recall of the system, thus improving the confidence of those boxes with accurate locations but low scores.

The main premise of Center-Net; inculcation of knowledge about the centre region of object works well with smaller objects more than large ones because it is easier to locate the centre of a smaller object.

With hourglass-104 as the backbone, inference time of Corner-Net and Center-Net was observed to be roughly 300ms and 340ms respectively. Center-Net speeds up to 270ms with hourglass-52 backbone along with improvement in accuracy.

### 3.5 Retina-Net

The trailing accuracy of one-shot detectors motivated the research for Retina-net[32], which introduced a novel concept of focal loss that reduces the focal loss of well-classified examples. The class imbalance during training was found to be the leading impediment for one-stage detectors. Retina-net demonstrated comparable accuracies with that of state-of-the-art two-stage detectors, surpassing all the previous one-stage detectors.

After taking images as inputs, the model uses anchors to predict proposals. The use of FPN with the backbone architecture avoids the burden of taking into consideration the variable scales of images. Anchors are similar to that of RPN, which predicts a class probability and 4 bounding box offsets. If IOU>0.5, the anchor is assigned the ground truth values, else, if IOU<0.4, it is regarded as background class. Also, if 0.4<IOU<0.5, then the anchor is ignored during training.

Along with the main network, two subnetworks are used in Retina-net, the first of which predicts the class of the region and the second one predicts the coordinate offsets. Each convolutional layer has a ReLU activation function and maintains the same channel size as the input feature map. Sigmoid activations are attached to the output feature map.

Training is performed using SGD, with the initial learning rate set to 0.01 and is divided by 10 after the count of examples exceeds 60k and again at 80k. The only data augmentation performed is that of horizontal image flipping.

With the Resnet-101-FPN backbone, Retina-net achieves an AP of 39.1 on the COCO test-dev, running at 5 fps. Retina-net boasts the state-of-the-art accuracy in object detection.

## 4. Conclusion

By analysing the trends and turns, or even anomalies within images or videos, object detection techniques provide insight on how decision-making, overall efficiency and customer experience can be improved in business intelligence practices. The computational complexity of training an object detection model is of pivotal concern, and dictates the efficacy of these applications. Acquiring in depth knowledge about the training time and space complexities plays a significant role in ensuring that the right technique is used for the right circumstances. This paper, with its descriptive interpretations of various object detection techniques including R-CNN, SSD, and YOLO([7],[8],[9]); provides a profound elucidation that can help understand the varying factors that determine the applicability of the existing object detection models. With this intuition, the use of object detection in the field of business intelligence can be magnified for societal benefit.

## 5. References

1. Watson, H.J. and Wixom, B.H., “The current state of business intelligence”. *Computer*, *40*(9), pp.96-99, 2007.
2. LeCun, Y., Bengio, Y. and Hinton, G., “Deep learning”. *nature*, *521*(7553), pp.436-444, 2015.
3. Zhang, L., Wang, S. and Liu, B., “Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery”, 8(4), p.e1253, 2018.
4. “COCO: Common Objects in Context”. http://mscoco.org/dataset/ #detections-leaderboard 2016.
5. J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. FeiFei. “ImageNet Large Scale Visual Recognition Competition 2012 “(ILSVRC2012). http://www.image-net.org/challenges/LSVRC/2012/ 2012.
6. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. “The PASCAL Visual Object Classes (VOC) Challenge”. IJCV, 2010.
7. R. Girshick, J. Donahue, T. Darrell, and J. Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In CVPR, 2014.
8. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. “SSD: Single shot multibox detector”. arXiv:1512.02325v2, 2015.

9. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: Unified, real-time object detection". In CVPR, 2016
10. D. Hoiem, Y. Chodpathumwan, and Q. Dai. "Diagnosing error in object detectors". In ECCV. 2012
11. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks". In ICLR, 2014
12. Neubeck, A. and Van Gool, L., August. "Efficient non-maximum suppression". In *18th international conference on pattern recognition (ICPR'06)* (Vol. 3, pp. 850-855). IEEE, 2016.
13. A. Krizhevsky, I. Sutskever, and G. Hinton. "ImageNet classification with deep convolutional neural networks". In NIPS, 2012
14. J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. "Selective search for object recognition". IJCV, 2013.
15. Dean, T., Ruzon, M.A., Segal, M., Shlens, J., Vijayanarasimhan, S. and Yagnik, J., "Fast, accurate detection of 100,000 object classes on a single machine". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1814-1821), 2013.
16. P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. "Object detection with discriminatively trained part based models". TPAMI, 2010
17. R. Girshick, P. Felzenszwalb, and D. McAllester. "Discriminatively trained deformable part models, release"
18. K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In CVPR, pages 770–778, 2016.
19. K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". arXiv preprint, arXiv:1409.1556, 2014
20. K. He, X. Zhang, S. Ren, and J. Sun. "Spatial pyramid pooling in deep convolutional networks for visual recognition". In ECCV, 2014.
21. C. L. Zitnick and P. Dollar, "Edge boxes: Locating object ´ proposals from edges," in ECCV, 2014
22. R. Girshick. "Fast R-CNN". In ICCV, 2015
23. S. Ren, K. He, R. Girshick, and J. Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks." In NIPS, 2015
24. J. Dai, Y. Li, K. He, and J. Sun. "R-FCN: object detection via region-based fully convolutional networks". In NIPS, pages 379–387, 2016.
25. Cai, Z. and Vasconcelos, N., "Cascade r-cnn: Delving into high quality object detection". In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6154-6162) 2018.
26. Tsung-Yi Lin, Piotr Dollar, Ross B Girshick, Kaiming He, ´ Bharath Hariharan, and Serge J Belongie. "Feature pyramid networks for object detection". In IEEE Conference on Computer Vision and Pattern Recognition, 2017.
27. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: "Scalable object detection using deep neural networks". In: CVPR, 2014
28. C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. "Dssd: Deconvolutional single shot detector". arXiv preprint arXiv:1701.06659, 2017.
29. J. Redmon and A. Farhadi. "YOLO9000: Better, faster, stronger." In CVPR, 2017
30. Law, H. and Deng, J., "Cornernet: Detecting objects as paired keypoints". In *Proceedings of the European conference on computer vision (ECCV)* (pp. 734-750) 2018.

31. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q. and Tian, Q., 2019. "Centernet: Object detection with keypoint triplets". *arXiv preprint arXiv:1904.08189, 3,* p.6, 2019.
32. Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P., "Focal loss for dense object detection". In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988)2017.