A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework

To cite this article: S A Sanchez *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **844** 012024

View the article online for updates and enhancements.

A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework

S A Sanchez [1], H J Romero [1] y A D Morales [1]

[1]Corporación Universitaria Antonio José de Sucre, Sincelejo, Colombia,
Faculty of Engineering Science, GINTEING Research Group

sergio_sanchez@corposucre.edu.co

Abstract. Advances in parallel computing, GPU technology and deep learning facilitate the tools for processing complex images. The purpose of this research was focused on a review of the state of the art, related to the performance of pre-trained models for the detection of objects in order to make a comparison of these algorithms in terms of reliability, ac- curacy, time processed and Problems detected The consulted models are based on the Python programming language, the use of libraries based on TensorFlow, OpenCv and free image databases (Microsoft COCO and PASCAL VOC 2007/2012). These systems are not only focused on the recognition and classification of the objects in the images, but also on the location of the objects within it, drawing a bounding box around the appropriate way. For this research, different pre-trained models were re- viewed for the detection of objects such as R-CNN, R-FCN, SSD (single- shot multibox) and YOLO (You Only Look Once), with different extractors of characteristics such as VGG16, ResNet, Inception, MobileNet. As a result, it is not prudent to make direct and parallel analyzes between the different architecture and models, because each case has a particular solution for each problem, the purpose of this research is to generate an approximate notion of the experiments that have been carried out and conceive a starting point in the use that they are intended to give.

1. Introduction

Increased Industry 4.0 or better known as the fourth industrial revolution promises great advances and technological challenges, the concept of artificial intelligence is the central protagonist of this transformation, related to the analysis of large volumes of data (Big Data) and use algorithms for processing. Generating great expectations due to the impact that this technology can conceive that promises to change the paradigm of machines. Today the words artificial intelligence, machine learning and learning Deep are widely used, but there is no clear difference between each, to the point of believing that is the same concept.

Artificial intelligence is simulating machine behavior and human reasoning for this, use different techniques including machine learning, for example; Siri voice assistants and Alex, on the other hand, Machine Learning is the ability of computers to learn by itself from data and experiences or subsets of techniques using statistical methods to enable machines to learn by itself. In its most complex use uses Deep learning, for example, predictive analytics in autonomous vehicles, face detection of persons with judicial positions, among others.

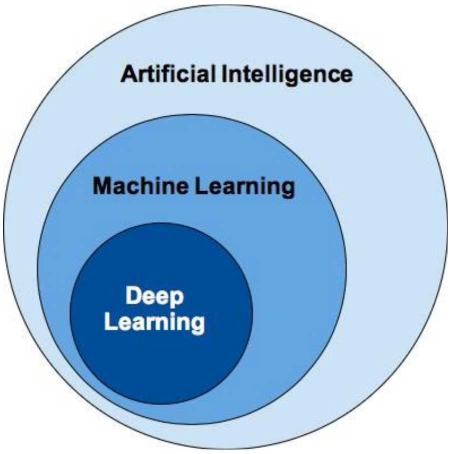

Figure 1. Relationship of the concepts of artificial intelligence, learning and deep learning machine.

With the emergence of the cognitive computing era, many researchers seek to incorporate human processes such as learning and communication with the purpose of reproducing them in machines. For many years, this technology has been surpassing different challenges, achieving rapid computation times (real time) and data processing as a human. Today we are talking about a new technology that is coming into great boom, which is Deep Learning, based on a series of neural networks that resemble the human brain. It consists of a set of unsupervised algorithms that form layers of artificial neurons to determine hidden features in a data set. In Figure 2, we observed some artificial intelligence techniques for the analysis of large volumes of data, from the most traditional to the most complex used today [2].

1.1. Using neural networks in Deep Learning

Neural networks have been crucial to teach computers to think and understand the world the way humans do, retaining innate advantages such as speed, ac- curacy and lack of bias. Artificial neural networks are widely used for their properties of nonlinear character, adaptability, generalization, fault tolerance, and scalability task decomposition. But today have disadvantages such as complexity in the design of architecture, lots of parameters to adjust difficulty and train networks.

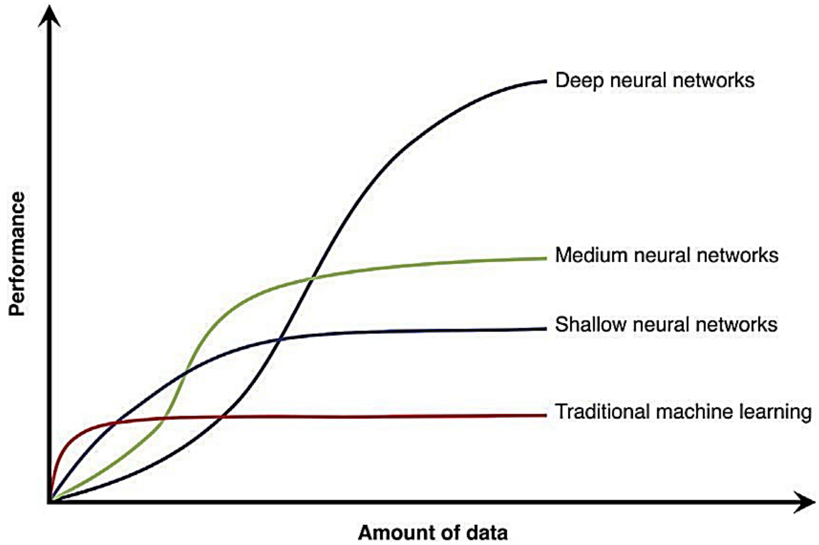

Figure 2. Artificial intelligence techniques vs. data volume.

Artificial neural networks can be classified according to their network topology and by learning algorithm. Depending on the topology can be the type of the (hidden or visible) layers, input or output and the directionality of the connections of neurons. According to the algorithm can be monitored, unsupervised, competitive or reinforcement. Furthermore, neural networks can be classified according to its architecture, pre-fed Neural Networks, convolutional and recurrent [3] networks.

The pre-fed neural networks were the first to be deployed, for its simple model. In these networks, the information is shifted in one direction, in this architecture include pioneers were perceptron multilayer perceptron and [4].

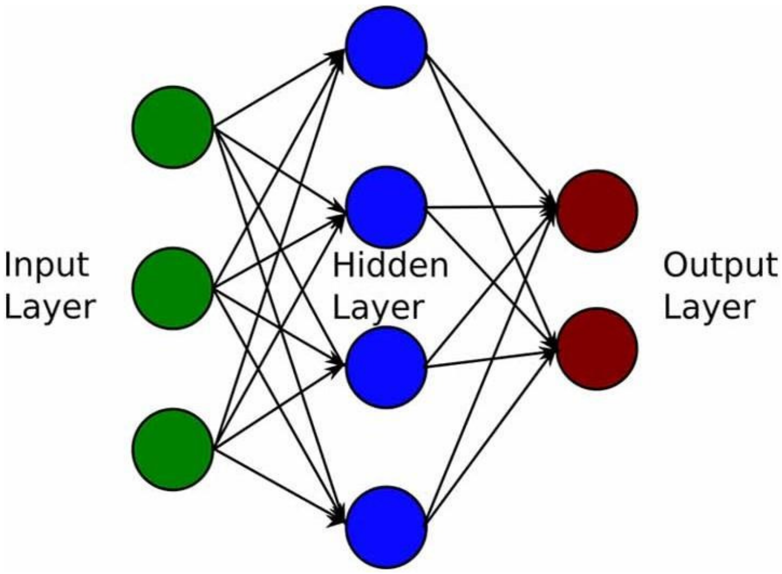

Figure 3. Topology of an artificial neural network.

The convolutional neural network (CNN) are very similar to the multilayer perceptron, are made up of neurons having weights and biases, which can be punished and learn from the entries. All this supported in a loss function or cost over the last layer. Some specific implementations of such networks are: ResNet, VGG16, Xception, Inception V3, among others [5].

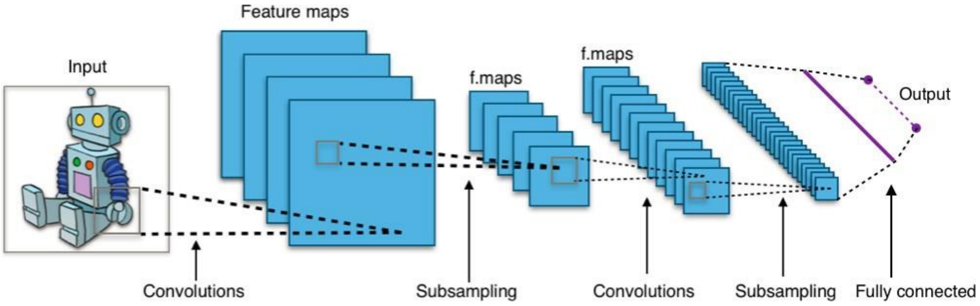

Figure 4. convolutional neural networks.

Recurrent Neural Network (RNN) using feedback loops, which allow information to persist. These apply the same operation to each element of a sequence of input data (text, voice, video, etc.), whose output depends on the input data as past information, characterized by memory capacity. The RNN conventional networks have problems in their training because backward gradients tend to grow or fade over time, because the gradient of the error depends not only present but also past mistakes. This causes the difficulty of memorizing dependencies for a long time. That is why architectures and learning methods have been developed that avoid these problems such as LSTM, neurological Turing machines and memory networks [6].

There are several types of training neural networks, such as manual tuning, learning algorithms and genetic algorithms. The most common problems during training are underfitting, overfitting and local minimum.

2. Methodology

In recent years, the architecture of the deep networks has been a significant progress for the moment Keras and TensorFlow dominates with different-pre- trained models already included in libraries, among these include VGG16, VGG1, ResNet50, Inception V3, Xception, MobileNet. The VGG and AlexNet 2012 networks follow a typical pattern of classical convolutional networks. MobileNet is a simplified architecture Xception architecture, optimized for mobile applications. The following architectures; ResNet, Inception and Xception have become a reference point for subsequent studies of artificial vision and learning for its versatility Deep [7].

There are many factors that explain the revolution of deep learning, among these factors is highlighted; availability of sets of huge data and quality, parallel computing (GPU) features efficient activation for backpropagation, new architectures, new regularization techniques that allow train more extensive networks with less danger of overshooting, robust optimizers and software platforms with large communities like TensorFlow, Theano, Keras, CNTK, PyTorch, Chainer and Mxnet. All this has allowed solving problems easier. Today the Python programming language has great importance in Machine Learning compared to other languages because of its support for Deep learning framework.

Within this framework include TensorFlow which is a library of open source software for machine learning that allows you to deploy computing in CPU or GPU, developed by Google, using graphs flow data, PyTorch uses Python language and has the support of Facebook, Theano is a Python library that supports mathematical expressions involving tensioners, CNTK are a set of tools developed by Microsoft, open for Deep learning code, Keras is a library of neural networks

high level created by Francis Chollet, member of Brain google equipment that lets you choose whether the models that are built will be executed in Theano, TensorFlow or CNTK. Keras and TensorFlow can construct models of three different ways; using a sequential model, a functional API and pre-trained models.

Earlier we talked about the different architectures (MobileNet, Inception, ResNet, among others), now we discuss models for object recognition and Keras TensorFlow; Faster-CCN R, R-FCN, SSD and YOLO. These models are classified based detectors in the region (Faster R-CNN, R-FCN, FPN) and single shot detectors (SSD and YOLO), start from different paths, but they look very similar now fighting for title faster and more accurate detector.

There are different metrics that can improve object detection algorithms based on more accurate positioning, faster speed and more accurate classification; metrics that stand out are: Intersection over Union (IoU), mean average precision (MAP) and rendered frames per second (FPS). Intersection over Union (IoU)It is an indicator that determines how close the predicted picture of the real picture [9]. The average metric average accuracy (MAP) is the product accuracy and recovery detection bounding boxes. It's a good combined measure of how sensitive the network to objects of interest and how well it avoids false alarms. The higher the score the map, the more precise the network, but this has a cost of execution speed [10]. Processed frames per second (FPS) is used to judge how fast is the system [11].

a. Datasets

Architectures and above models need data lately have focused on free data sets posted on the web, like Microsoft COCO (common objects in context) and PAS- CAL Visual Object Classes (VOC). Microsoft COCO is a dataset of 300,000 images with common objects 90 supported on an API that provides different models of object detection, which compensates for the speed and accuracy based on bounding boxes suitable objects [12]. PASCAL Visual Object Classes (VOC) is a reference point in the visual recognition of object categories and detection. It consists of a set of standard image data, annotations, and evaluation procedures [13]. Organized since 2005 to the present.

b. Comparison Between Deep Learning Algorithms for Object Detection

It is difficult to define a fair feature of different object detectors, each case of real life can have different solutions to reach a decision concerning the accuracy and speed, it is necessary to know other factors that affect performance; the type of feature extractor, steps out of the extractor, income resolutions images, strategy coincidence and threshold (as predictions are excluded when calculating the loss), Threshold IOU no maximum suppression ratio of positive anchor and negative, number of proposals or predictions of frame coding limit, increased data set of training data, using multi-scale images training or testing (with clipping), map layer features for object detection, It is important to note that technology is constantly evolving, any comparison can become obsolete quickly.

A comprehensive review of scientific papers and academics on the performance of different models object detection with TensorFlow framework where investigations found the following was performed; The authors Shaoqing Ren Kaiming He, Ross Girshick, and Jian Sun in his research entitled "Faster R-

CNN: Towards Real-Time Object Detection with Region Proposal Networks" analyzed several test set where different results were found based on the as medium accuracy (mAP) and the number of predictions using the R-CNN Faster method evidenced in table 1 [15].

Table 1. Performance Model Faster R-CNN based metrics speed, accuracy and num- bers predictions using database COCO MS [16].

Method	# proposals	data	mAP (%)
SS	2000	12	65.7
SS	2000	07++12	68.4
RPN+VGG, shared	300	12	67.0
RPN+VGG, shared	300	07++12	70.4
RPN+VGG, shared	300	COCO+07++12	75.9

Better average precision measurement (MAP) of the model is evidence Faster R-CNN based on RPN 300 + VGG methods predictions using data bases MS COCO + PAS-CAL VOC 2007+ PAS-CAL VOC 2012. In the following table, focused tests were performed on a K40 GPU with the test set PASCAL VOC 2007.

Table 2. Performance Model Faster R-CNN function metrics speed, accuracy and numbers predictions database PASCAL VOC 2007 [17].

model	system	conv	proposal	region-wise	total	rate
VGG	SS + Fast R-CNN	146	1510	174	180	0.5 fps
VGG	RPN+ Fast R-CNN	141	10	47	198	5 fps
ZF	RPN+ Fast R-CNN	31	3	25	59	17 fps

Can be determined in Table 2, the models made by RPN + Fast R-CNN and RPN + Fast R-CNN systems were the most efficient, with a rate of rendered frames per second of 5 and 17 fsp and a rate prediction 10 and 3 respectively. On the other hand, the authors Jifeng Dai, Yi Li, Qinghua He, Jian Sun in his re- search entitled "R-FCN: Object Detection via Region-based Fully Convolutional Networks" compared with detectors based on faster R-CNN, consumption com- puter is expensive, in contrast to NGF-R technique is based on convolutional calculations are shared across the image [18].

In Table 3, the results obtained according to the medium accuracy mea- surement (MAP) and the response time, which worked with the database MS COCO test and PAS-CAL VOC 2007, 2012. The code is evidenced available at the following link: https://github.com/daijifeng001/r-fcn.

Table 3. Result sobtained from the R-R-CNN and FCN models based metrics speed and accuracy.

	training data	mAP (%)	test time (sec/img)
Faster R-CNN	07++12	73.8	0.42
Faster R-CNN	07++12+COCO	83.8	3.36
R-FCN multi-sc train	07++12	77.6	0.17
R-FCN multi-sc train	07++12+COCO	82.0	0.17

better average precision measurement (MAP) is evidenced by the R-CNN method using the data bases MS COCO + PAS-CAL VOC 2007+ PAS-CAL VOC 2012, with 83.8 mAP and a testing time of 3.36 sec/img.

Another method was analyzed SSD, where the authors Wei Liu, Dragomir Angelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Yang Cheng-Fu, Alexan- der C. Berg conducted a study entitled “SSD: Single Shot MultiBox”; presenting a method with a deep neural network, where the output space defined by a set of predetermined tables on different aspect ratios and scale is discretized.

The network generates scores for each category of object in each frame and produces default settings for that box delimiter closely matches the shape of the object. In addition, the network combines multiple predictions feature maps with different resolutions to naturally handle objects of various sizes. This makes it simple to train SSD with different databases [20].

In Table 4, the results based on the measurement of average precision (MAP) and the response time in databases PASCAL VOC 2007, 2012 and MS COCO shown using input images with a resolution of 300 300 and 512 512 [21].

Table 4. Performance SSD method using databases PASCAL VOC 2007, 2012 and MS COCO.

Method	**VOC2007 test 07+12 07+12+COCO 0.5 0.5**		**VOC2012 test 07++12 07++12+COCO 0.5 0.5**		**COCO test-dev2015 trainval35k 0.95 0.5 0.75**		
SSD300	74.3	79.6	72.4	77.5	23.2	41.2	23.4
SSD512	76.8	81.6	74.9	80.0	26.8	46.5	27.8
SSD300*	77.2	81.2	75.8	79.3	25.1	43.1	25.8
SSD512*	79.8	83.2	78.5	82.2	28.8	48.5	30.3

In the above table it shows that the SSD method has better performance using input images with resolution of 512 x 51 databases PASCAL VOC 2007, 2012 + MS COCO.

In Table 5, we find the results based on the measurement of average precision (MAP) and FPS with databases PASCAL VOC 2007, 2012 and MS COCO using input images with a resolution of 300 300 and 512 512 and Faster methods R- CNN, YOLO and SSD [22].

Table 5. Results in the test set PASCAL VOC 2007, 2012 and MS COCO using input images with different resolutions, implementing the YOLO, Faster R-CNN and SSD [23] methods.

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	6000	1000X600
Fast YOLO	52.7	155	1	98	448X448
YOLO (VGG16)	66.4	21	1	98	448X448
SSD300	74.3	46	1	8732	300X300
SSD512	76.8	19	1	24564	512X512
SSD300	74.3	59	8	8732	300X300
SSD512	**76.8**	**22**	**8**	**24564**	**512X512**

In the above table the yield of YOLO, Faster R-CNN and SSD methods where it validates that each method has its advantages and disadvantages, concerning MAPs, FPS parameters, the number of predictions and image resolution.

Finally, the authors Joseph Redmon and Ali Farhadi University of Washing- ton and Allen Institute for AI, conducted a study entitled “YOLO9000: Better, Faster, Stronger” YOLO is a method of detecting objects that uses the latest technology in time real, which can detect more than 200 classes and 9,000 different categories of objects. This novel method has various improvements, as is YOLOv2 and YOLOv3, using a training method multiscale, is a relatively new and very efficient technology standard detection tasks databases and MS COCO PASCAL VOC. Exceeding methods as faster and SSD RCNN with ResNet [23].

In Table 6, the results evidence function as medium accuracy (mAP), FPS and databases PASCAL VOC 2007, 2012 and MS COCO using input images with different resolutions, implementing methods Faster R-CNN, Yolo and SSD Table 6. Performance of YOLO, Faster R-CNN and SSD methods works metrics speed and accuracy using the test set PASCAL VOC 2007 [24].

In Table 7, the results evidenced in function of the measured average precision (MAP) and databases PASCAL VOC 2007 and 2012, using the methods Faster R-CNN, YOLO and SSD.

It is not wise to make a parallel analysis of the above article, these experiments are performed in different environments. But the purpose of this article is to have a general notion about these methods.

In Figure 5, the results of input images is shown with dimensions of 300 x 300 and 512 x 512 using free databases IMAGEnet, PASCAL VOC 2007, 2012 and MS COCO for different methods making comparisons based on the metric precision. The YOLO method has different results for input images of 288 x 288, 416 x 461 and 544 x 544. The higher resolution images for the same model have better Map but are slower to process [26].

Table 6. Performance of YOLO, Faster R-CNN and SSD methods works metrics speed and accuracy using the test set PASCAL VOC 2007 [24].

Detection Frameworks	Train	mAP	FPS
Fast R-CNN	2007+2012	70.0	0.5
Faster R-CNN VGG-16	2007+2012	73.2	7
Faster R-CNN ResNet	2007+2012	76.4	5
YOLO	2007+2012	63.4	45
SSD300	2007+2012	74.3	46
SSD500	2007+2012	76.8	19
YOLO v2 288 x 288	2007+2012	69.0	91
YOLO v2 352 x 352	2007+2012	73.7	81
YOLO v2 416 x 416	2007+2012	76.8	67
YOLO v2 480 x 480	2007+2012	77.8	59
YOLO v2 544 x 544	2007+2012	78.6	40

Table 7. Results in test set PASCAL VOC 2007, 2012 using the R-CNN, and SSD YOLO [25] method.

Method	data	mAP	aero	bike	...	bird	boat
Fast R-CNN	07++12	68.4	82.3	78.4	... ...	70.8	**52.3**
Faster R-CNN	07++12	70.4	84.9	79.8	...	74.3	**53.9**
YOLO	07++12	57.9	77.0	67.2	... ...	57.7	**38.3**
SSD300	07++12	72.4	85.6	80.1	...	70.5	**57.6**
SSD512	07++12	74.9	87.4	82.3		75.8	**59.0**
ResNet	**07++12**	**73.8**	**86.5**	**81.6**		**77.2**	**58.0**

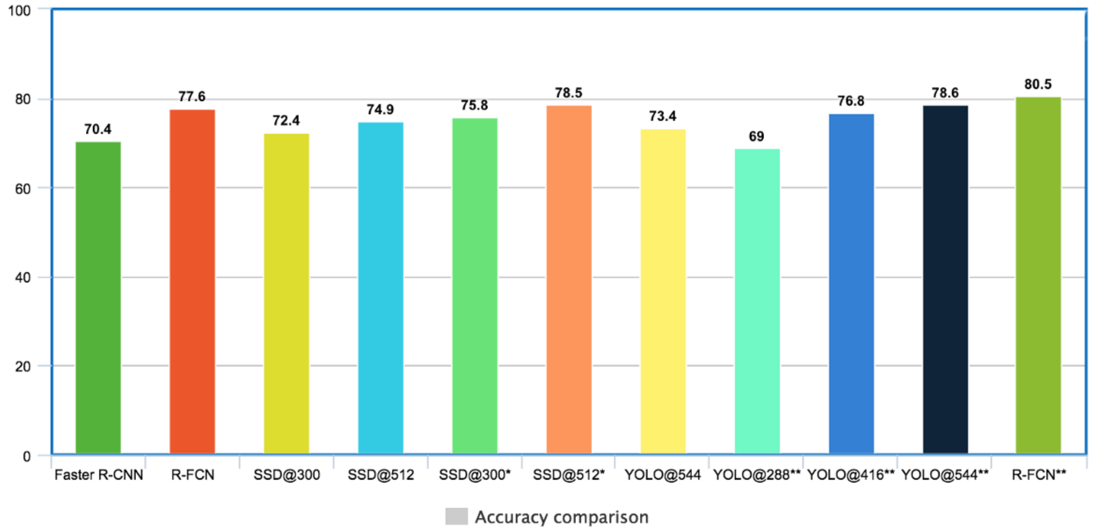

Figure 5. Comparison accuracy Faster R-CNN, R-FCN, SSD and YOLO models using input images with different resolutions.

In Figure 6, one can observe the resolutions of the input images and the feature extractors vary the speed of the detectors. Then the metric FPS highest and lowest reported by the above information is displayed, however, the result can then be very biased, in particular when measured at different map [27].

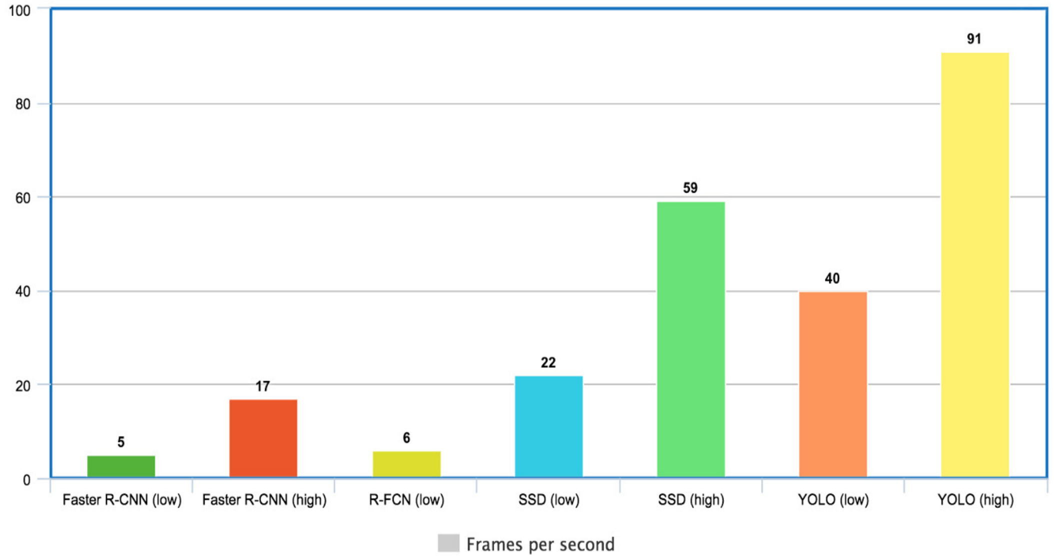

Figure 6. Comparison of frames processed per second (FPS) implementing the Faster R-CNN, R-FCN, SSD and YOLO models using input images with different resolutions.

In recent years, many results are measured exclusively with the data set MS detection COCO objects. This data set is more difficult to detect objects and, generally, detectors reach a much lower mAP. To measure the accuracy of these models has been used AP, where 0.5 denoting a fair and detection 0.95 indicating a very accurate detection. Here are some comparisons of key detectors. This can be seen in Figure 7 [28]. Table 11.

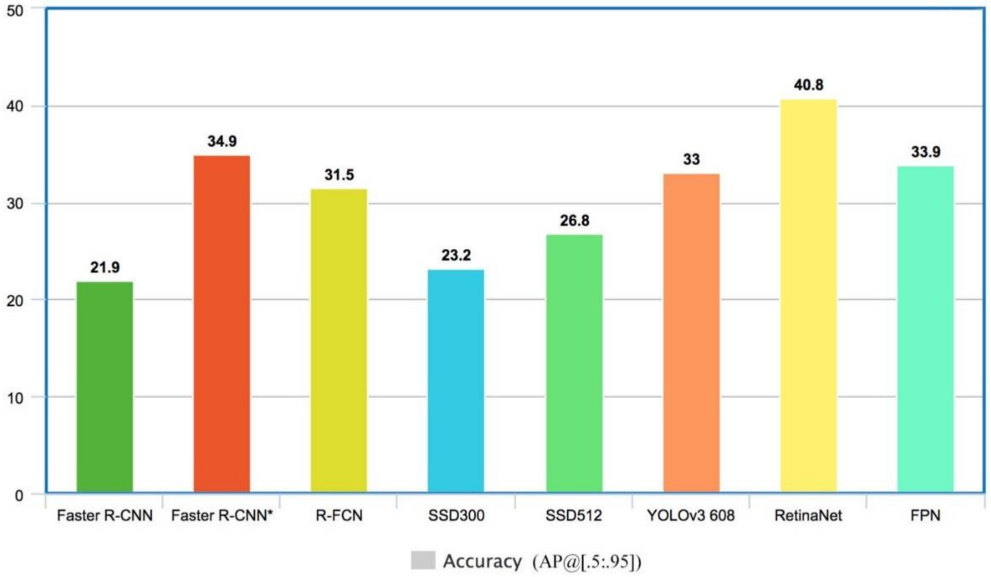

Figure 7. Comparison of accuracy Faster R-CNN, R-FCN, SSD and YOLO models using the database MS COCO.

The above items are studying how the resolution of the input images and feature extractors affects accuracy and speed detector. Overall, Faster R-R-FCN CNN and compared with SSD and YOLO models are slightly slower but more accurate. YOLO SSDs and methods have difficulty detecting small objects, but are faster.

In Table 9 the advantages and disadvantages of methods Fast R-CNN, Faster R-CNN, R-FCN, SSD and YOLO for detecting objects in images will be high- lighted by various experiments conducted by a variety of authors investigated in Deep Learning area.

Table 8. Advantages and disadvantages of some methods for detecting objects in images.

Method	Authors	Advantage	Disadvantages
Fast R-CNN	(Girshick, 2015)	The calculation of the characteristics of CNN is done in a single iteration, achieving that the detection of objects is 25 times faster than the RCNN method (it requires 20 seconds on average to analyze an image).	The use of an external candidate region generator creates a bottleneck in the detection process.
Faster R-CNN	(Renet al., 2015)	The RPN method allows object detection to be almost real-time, approximately 0.12 seconds per image.	Despite the efficiency of the algorithm, it is not fast enough to be used in applications that require real time, as would be the autonomous vehicles.
R-FCN	(Dai et al., 2016)	The test time of R-FCN is much faster than that of R-CNN	R-FCN has a competitive mAP but lower than that of Faster R-CNN.
Mask R-CNN	(He et al., 2017)	The location of the objects is more precise, when making a segmentation of the objects in the images.	Its execution time is greater than that used by the Faster-RCNN method, therefore, it can not be be implemented in applications that require real time.
YOLO	(Redmon et al., 2015)	The location of objects is very efficient, allowing its use in real-time applications.	The method has difficulties to correctly detect small objects.
SSD	(Liu et al., 2016)	The use of a single network, makes the location of the objects faster than the Fast-RCNN and Faster-RCNN methods.	The detection accuracy of the objects is lower compared to the Fast-RCNN and Faster-RCNN methods.

3. Conclusion

This research has been conducted a systematic review other than 50 scientific papers, where the recent progress of deep learning networks object detection is evident, deep models have significantly improved performance, but there are still many challenges and challenges. Parallel computers and GPU have greatly reduced training time of artificial neural networks and pre-trained models they have succeeded in reaching loaded into the devices reduced computation times, without having to use a GPU. Allowing to have a starting point for researchers seeking to venture into this line and not have to build and train an object detector from scratch, which would require a long time.

The difference between the detectors is shrinking. Single shot detectors use more complex designs in order to be more precise and regions based detectors accelerate the operations to be faster. The detector Yolo single shot, for example; associated features of other detectors, the specific difference may not be the basic concept but the details of implementation.

Detectors based on regions such as CNN and Faster R-R-FCN show a small advantage if speed precision is needed in real-time, single-shot detectors are here for real-time processing. But applications must verify if it meets your requirements accurately. The difference YOLO and SSD model is simply that the model does not include YOLO feature maps of varying size that makes SSD, and also uses its own custom base architecture. The YOLO and SSD methods have difficulty detecting small objects. In the case of these methods, the accuracy of detection of objects is smaller as compared to Faster-RCNN and R-FCN methods.

The R-FCN, YOLO and SSD models are faster on average, but cannot beat the R-CNN faster in accurately if speed is not a concern. feature extractor and resolution of the input image significantly affect the accuracy of the models. For large, SSD and YOLO objects can beat CNN and Faster R-R- FCN in precision with lighter and faster extractors. Although many researchers have made great strides in detection methods of objects, there are still many challenges that must be overcome. Deep learning will have a prospective future in a wide range of applications.

References

[1] Wang, L., & Sng, D. (2015). Deep Learning Algorithms with Applications to Video Analytics for A Smart City: A Survey 1-8. Retrieved from. http://arxiv.org/abs/1512.03131

[2] Orera Floria, JM. (2015). Development of a system for detecting people in indoor environments using fish-eye camera in overhead and algorithms based on Deep Learning plane. pp 5.

[3] Chaves, D., Saikia, S., & Fern, L. (2018). A Review of Methods Sistem tica to Locate Objects in Images Automatically, 15.

[4] Auer, P., Burgsteiner, H., & Maass, W. (2008). A learning very simple universal rule for approximators Consisting of a single layer of perceptrons Chair for Information Technology. Theoretical Computer Science, 015 879 1-29.

[5] Zhao, B., Li, X., Lu, X., and Wang, Z. RNN-CNN Architecture for Multi-Label Weather Recognition, pp-5. (2019).

[6] Manuel, U., Alcocer, R., Tello-loyal, E., Bertha, A., & Alvarado, R. (2018). NEURAL NETWORK BASED MODEL NEURAL NETWORK BASED rcurrent LSTM- MODEL FOR summary, 40 (130) 962-974.

[7] Sundar, KVS (2018). Evaluating Training Time of Inception-v3 and Resnet-50, 101 Models using CPU and GPU across TensorFlow. Second International Confer- ence on Electronics, Aerospace and Communication Technology (ICECA), (Iceca).

[8] Sun, C., & Murphy, K. (2017). Speed / accuracy trade-offs for modern convolutional object detectors, pp-11.

[9] Tsoi, N., Gwak, J., Reid, I., & States, U. (2019). Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression.

[10] Li, Y., Huang, C., Ding, L., Li, Z., Pan, Y., & Gao, X. (2019). Deep learning in bioinformatics: introduction, application, and perspective in big data was. White-filgueira, B., Garc, D., Fern, M., Brea, M., & Paula, L. Deep Learning- Based Visual Tracking on Multiple Object Embedded System for Mobile Edge IoT and Computing Applications 1-8.

[11] T.-Y. Lin, M. Maire, Belongie S. J. Hays, P. Perona, D. Ra- mannan Dolla'r P., and C. Lawrence Zitnick. (2014). Microsoft COCO: Common objects in context. In ECCV, 1 May.

[12] M. Everingham, L. Van Gool, CK Williams, J. Winn, and A. Zisserman. (2010). The visual object classes pascal (voc) challenge. International Journal of Computer Vision, 88 (2): 303-338,

[13] Wang, L., & Sng, D. (2015). Deep Learning Algorithms with Applications to Video Analytics for A Smart City: A Survey 1-8. Retrieved from http://arxiv.org/abs/1512.03131.

[14] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real- Time Object Detection with Region Proposal Networks. IEEE Transac- tions on Pattern Analysis and Machine Intelligence, 39 (6), pp 1. https://doi.org/10.1109/TPAMI.2016.2577031

[15] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real- Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39 (6), 1137-1149. p.8 https://doi.org/10.1109/TPAMI.2016.257703.

[16] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real- Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39 (6), 1137-1149. p 9. https://doi.org/10.1109/TPAMI.2016.2577031.

[17] Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object Detection Region- based via Convolutional Networks Fully. Retrieved from. P1. http://arxiv.org/abs/1605.06409.

[18] Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object Detection Region- based via Convolutional Networks Fully. Retrieved from. p7. http://arxiv.org/abs/1605.06409.

[19] Liu, W., Angelov D., Erhan, D., Szegedy, C. Reed, S., Fu, C., & Berg (2016). AC SSD: Single Shot MultiBox Detector, pp-1

[20] Liu, W., Angelov D., Erhan, D., Szegedy, C. Reed, S., Fu, C., & Berg (2016). AC SSD: Single Shot MultiBox Detector, pp-7.

[21] Liu, W., Angelov D., Erhan, D., Szegedy, C. Reed, S., Fu, C., & Berg (2016). AC SSD: Single Shot MultiBox Detector, pp-11.

[22] Liu, W., Angelov D., Erhan, D., Szegedy, C. Reed, S., Fu, C., & Berg (2016). AC SSD: Single Shot MultiBox Detector, pp-15.

[23] Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger. Pp-1.

[24] Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger. Pp-4.

[25] Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger. Pp-5.

[26] Sun, C., & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors .

[27] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learn- ing for image recognition. arXiv preprint arXiv: 1512.03385, 2015.two, 4,5.

[28] Sundar, (2018). KVS. Evaluating Training Time of Inception-v3 and Resnet-50, 101

Models using CPU and GPU across TensorFlow. 2018 Second International Conference on Electronics, Aerospace and Communication Technology (ICECA), (Iceca), 1964-1968.

[29] Sun, C., & Murphy, K. (2017). Speed / accuracy trade-offs for modern convolutional object detectors, pp-11.

[30] Chaves, D., Saikia, S., & Fern, L. (2018). A Review Sistem atic M ethods for Automatically Locate Objects in Images, p15.

[31] Tsoi, N., Gwak, J., Reid, I., & States, U. (2019). Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression.

[32] Li, Y., Huang, C., Ding, L., Li, Z., Pan, Y., & Gao, X. Deep learning in bioinformatics: introduction, application, and perspective in big data.

[33] White-filgueira, B., Garcia, D., Fern, M., Brea, M., & Paula, L. Deep Learning-

[34] Based Visual Tracking on Multiple Object Embedded System for Mobile Edge IoT and Computing Applications 1-8 (2019)

[35] T.-Y.Lin, M. Maire, Belongie S. J. Hays, P. Perona, D. Ra- mannan Dolla'r P., and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In ECCV, 1 May (2014)

[36] TYLin and P. Dollar. Ms coco api.https: // github. com / pdollar / coco, 2016.5