Journal of Information Technology Management

Print ISSN: 2008-5893

Online ISSN: 2423-5059

Comparing the Performance of Pre-trained Deep Learning Models in Object Detection and Recognition

Omar Ibrahim Obaid

Department of Computer Science, College of Education, AL-Iraqia University, Baghdad, Iraq. E-mail: alhamdanyomar23@gmail.com

Mazin Abed Mohammed*

*Corresponding Author, Ph.D., College of Computer Science and Information Technology, University of Anbar, Ramadi, 31001, Iraq. E-mail: mazinalshujeary@uoanbar.edu.iq

Akbal Omran Salman

Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq. E-mail: Akbal.O.Salman@mtu.edu.iq

Salama A. Mostafa

Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Johor, 86400, Malaysia. E-mail: salama@uthm.edu.my

Ahmed A. Elngar

Faculty of Computer & Artificial Intelligence, Beni-Suef University, Beni-Suef City, 62511, Egypt; College of Computer Information Technology, American University in the Emirates, United Arab Emirates. E-mail: elngar_7@yahoo.co.uk

Abstract

The aim of this study is to evaluate the performance of the pre-trained models and compare them with the probability percentage of prediction in terms of execution time. This study uses the COCO dataset to evaluate both pre-trained image recognition and object detection, models. The results revealed that Tiny-YoloV3 is considered the best method for real-time applications as it takes less time. Whereas ResNet 50 is required for those applications which require a high probability percentage of prediction, such as medical image classification. In general, the rate of probability varies from 75% to 90% for the large objects in ResNet 50. Whereas in Tiny-YoloV3, the rate varies from 35% to 80% for large objects, besides it extracts more objects, so the rise of execution time is sensible. Whereas small size and high percentage probability makes SqueezeNet suitable for portable applications, while reusing features makes DenseNet suitable for applications for object identification.

Keywords: Deep Learning, Image Recognition, Object Detection, Pre-trained Models.

Journal of Information Technology Management, 2022, Vol. 14, No.4, pp. 40-56
Published by University of Tehran, Faculty of Management
doi: https://doi.org/ 10.22059/jitm.2022.88134
Article Type: Research Paper

Received: April 13, 2022
Received in revised form: May 25, 2022
Accepted: June 28, 2022
Published online: July 13, 2022

Introduction

Object detection and image recognition are the key challenges in the systems of computer vision because of the diversity that each particular image or object where the object is exhibited could have like the brightening or the main position of the object. The investigation of Artificial Neural Networks (ANNs) is the starting of the notion of deep learning (DL) (Nisa, et al., 2021). DL Approaches have the capability to learn from experience, it is robust and boosts the performance by modifying the alteration in the environment, at present tough to train. Deep learning techniques have attracted a lot of researchers' interests due to their deep-seated capability to overcome the disadvantages of conventional techniques based on features of handcrafted. The technology of DL is a common word in the current time because of state-of-the-art technology outcomes acquired in the object detection and image recognition domain.

The huge and the free publicly datasets and potent Graphics Processing Units (GPUs) are two causes that have made deep learning technologies have such publicity. The need for huge datasets and strong resources to carry out the training have satisfied at the current time. Figure 1 illustrates the surprising rise in DL with regard to computer vision in the last lustrum (Sheu, J.S. and Chen-Yin, H., 2019). Image prediction and classification have been generally investigated domains in the field of computer vision which has done marked outcomes in the wide-world contests with the support of deep learning techniques (Krizhevsky, et al., 2012). The researchers have developed models of deep learning for object detection by the inspiration of outcomes in the area of image classification which has also accomplished remarkable outcomes (Ren, et al., 2017).

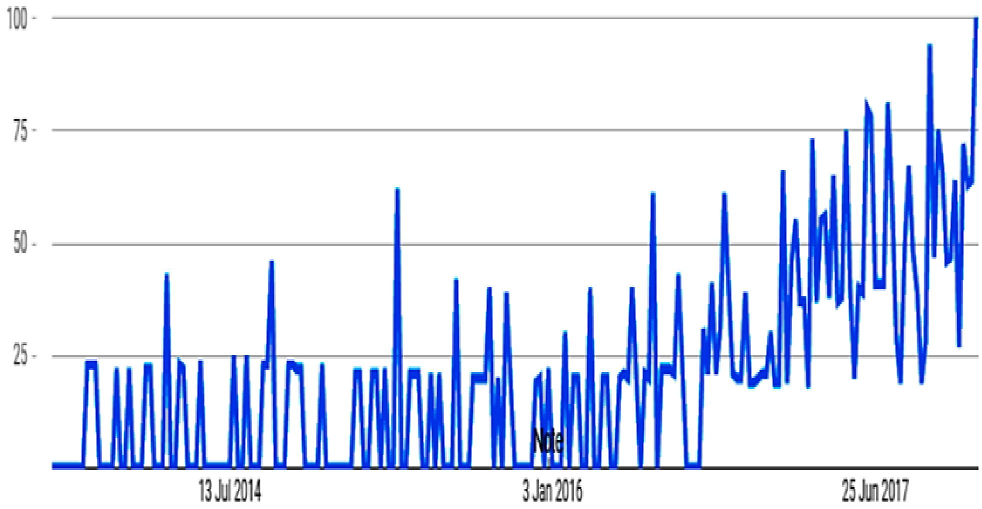

Figure 1. The surprising rise in DL concerning computer vision

This research aims to evaluate the pre-trained models based on convolutional neural networks (CNNs) models for image recognition and object detection. The works mentioned in (Huh, et al., 2016), (Yosinski, et al., 2014) have already discussed the impact of utilizing the pre-trained models on the images of the same scope; however, this was done on the datasets of ImageNet. In this work, we evaluate four pre-trained models in the same scope but on the coco datasets. The rest organization of this study is Section 2 presented a literature review that includes CNN models, object detection, and most related work. Materials and methods for data set used and CNN models parameters setting used in section 3. Section 4 showed the experimental results for image recognition and object detection outcomes. Finally, the summaries and recommendations for future work have been concluded in the conclusion section.

Literature Review

Convolutional Neural Networks (CNNs)

The CNNs were first presented in 1989 (Y. LeCun, et al., 1989), it is a famous deep learning approach that was inspired by the normal technique of visible conception for the organisms, and it was comprehensively utilized for object detection. Technically, CNN is a sort of feed-forward neural network and it is acted on the precept of sharing weights. Figure 2 shows the typical block diagram of CNNs (Sermanet, et al., 2011).

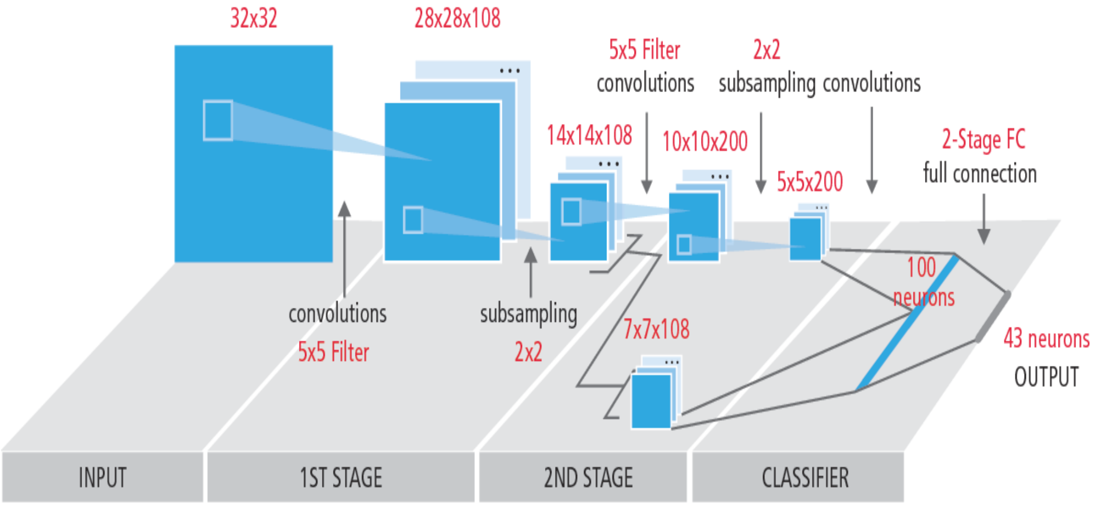

Figure 2. Typical block diagram of CNNs (Sermanet, et al., 2011)

The convolutional neural network consists of convolutional, pooling, and dense layers pursued by a SoftMax layer for the image classification (Alex Krizhevsky, et al., 2012). CNN has various sequent layers in a way that the output of the layer is the input for the next layer. Convolution means that one formula can overlap with other one and it is a mix of two formulas got multiplied. The feature maps can be obtained from the image which convolved with the activation function.

For the purpose of decreasing the unpredictability of spatial for the network, pooling layers are managed along with feature maps to obtain abstracted feature maps. This procedure is recurrent for the ideal number of filters as feature maps are made. Ultimately, feature maps are treated with wholly linked layers to obtain the outcome of image classification showing trust results for the labels of the predicted class (Sheu, J.S. and Chen-Yin, H., 2019).

Object Detection

Computer vision is an important area of artificial intelligence. It is formed from diverse sides like object detection and image recognition. The topic of object detection is extensively investigated in computer vision topics. The main objective is to find the pattern of the object within a given image. The process of object detection basically discovers an object by applying a recognition algorithm for a given image (Awan, et al., 2021). Object detection is the first footstep in each efficacy of visional recognition. Figure 3 shows the YOLO network which was used in order to detect objects for a given image (Redmon, et al., 2016).

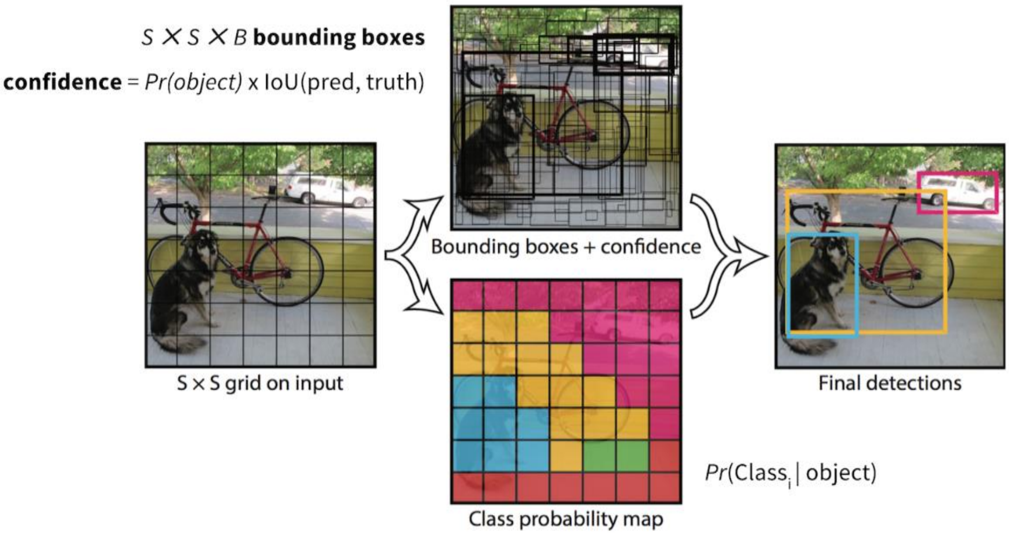
$S \times S \times B$ **bounding boxes**
confidence = *Pr(object)* x IoU(pred, truth)
Bounding boxes + confidence
S × S grid on input
Final detections
Pr(Class$_i$ | object)
Class probability map

Figure 3. YOLO network for object detection (Redmon, et al., 2016**)**

The process of detecting a class with a single pattern from a given image is known as object detection of a single-class. On the other hand, the process of detecting all the objects patterns for a given image is known as object detection of a multi-class. Furthermore, localization of the objects is required for the process of object detection, while the process of classification does not. YOLO network is splitting the image into a grid, and for each cell of the grid, the bounding boxes are predicted.

Related Work

The fast progression in the field of convolutional neural networks (CNNs), noticeable outcomes have been obtained using object pre-trained models (detectors) based on (CNNs). It became a modern direction in the literature of detection. The detection branch and backbone network are two parts of convolutional neural networks based on detectors. Object detection-based backbone network is generally scrounged from ImageNet (Russakovsky, et al., 2015). It was a formal dataset for evaluating the power of deep CNN. AlexNet (Mujahid A, et al., 2021) was on top in trying to raise convolutional neural network deepness.

GoogleNet (Szegedy, et al., 2015) suggests a modern block of inception to include extra various features. ResNet (Xie, et al., 2017) utilizes a set of CNN layers to substitute the conventional convolution. It minimizes the parameters and raised the accuracy altogether. DenseNet (Lal, S., et al., 2021) concatenates various layers densely, and it minimizes the parameters whereas considering accuracy competitively. On the other hand, object detection is based on the detection branch generally connected to the based model which has been trained for the dataset of ImageNet classification. One-phase detector and two-phase detector are two various logic designs for object detection (Z. Li, et al., 2018).

The first one is immediately utilizing the backbone for the prediction of the object pattern. For instance, YOLO (Redmon, et al., 2016), (Redmon, J, Farhadi, A, 2016) utilizes a straightforward effective backbone DarkNet and thereafter makes the detection simplified as it is a problem of regression. RetinaNet (Lin, et al., 2017) utilizes ResNet as a requisite extractor of feature, and thereafter includes Focal-Loss to tackle the problem of imbalanced class sourced by the utmost ratio of foreground-background.

On the other hand, the two-phase detector is first predicting a lot of motions depending on the backbone, and thereafter an extra classifier is included for the motion of regression and classification. Faster R-CNN (Ren, et al., 2015) immediately produces motions from the backbone by utilizing Region-Proposal Network (RPN). Feature Pyramid Network (FPN) (Lin, et al., 2016) builds pyramids of features by taking advantage of multi-scale inherently, particularly (FPN) using U-shape frame and hence combining the output of multi-layers, and yet still scrounges conventional ResNet with absent of more research.

Methodology

Materials

The images used in this study are taken from the Common Objects in Context (COCO) dataset which is a large-scale dataset for image recognition and object detection. This dataset exists in Ref (Cocodataset.org, 2022) with the (2017 Val images [5k/1GB]). This dataset is basically for detection and thus, is suitable for the objectives of this study. On the other hand, a python library called Image AI (ImageAI", Imageai.org, 2022) is used in this study. The library is basically for computer vision tasks, and it enables researchers and developers to construct applications and readily combines state-of-the-art deep learning technologies.

This library depends on several important libraries which should be installed as well to integrate the process of object recognition and detection. Furthermore, the pre-trained models which have been used in this study are located in (OlafenwaMoses/ImageAI", GitHub, 2022) and includes ResNet 50 and Tiny-YoloV3 which are used for the object detection process. The models used in this study were trained on COCO datasets which means that these models can recognize and detect about 80 various types of popular daily objects. In addition, SqueezeNet and DenseNet models have been used for the image recognition process. These models are trained on the ImageNet-1000 dataset which means that these models can recognize and predict about 1000 various objects in provided images.

ResNet 50

In this section, we discuss the ResNet 50 model. This model stands for Residual Network and it is a convolutional neural network that has 50 layers deep it was trained based on the COCO dataset. This means that this network can classify and detect about 80 various types of popular daily objects. A wide scope of images has a great impact on this model and thus, it has learned substantial representations of features. Figure 4 shows the architecture of the ResNet network with Feature Pyramid Network (FPN) (K. He, et al., 2016), (T.-Y. Lin, et al., 2017).

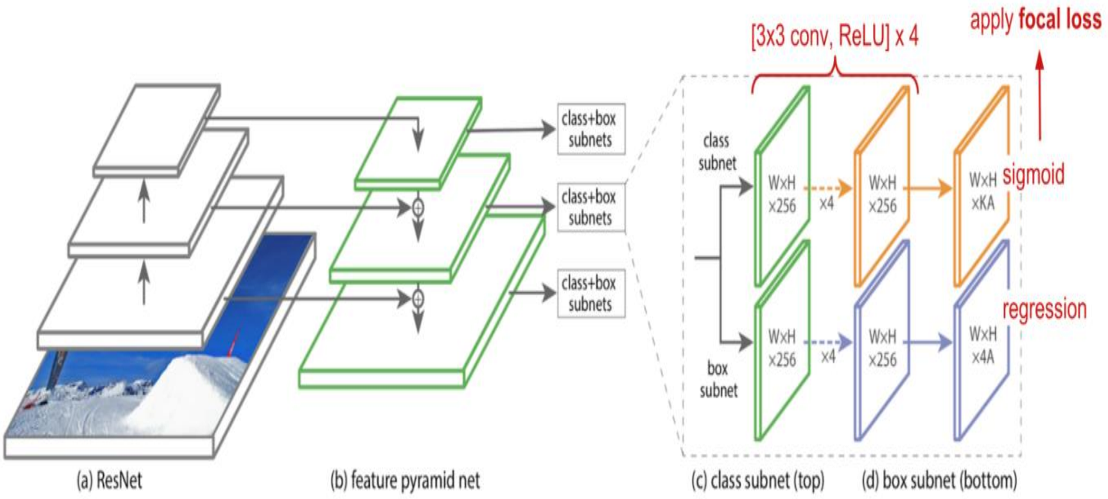

Figure 4. The architecture of the ResNet network with Feature Pyramid Network (FPN) (K. He, et al., 2016), (T.-Y. Lin, et al., 2017)

Every grade of the pyramid could be utilized for discovering objects at various levels. Fully convolutional networks are used for FPN to get better multi-level predictions. Furthermore, FPN is constructed upon the architecture of ResNet and it serves as the backbone for the layer of RetinaNet. The backbone is taking charge of calculating the convolutional map of the feature. The classification of the object is done using the first subnet on the output of the backbone. Otherwise, the regression of the bounding box is done using the second subnet. Furthermore, the focal loss is used to address the scenario of detection for the one-phase detector which has an excessive imponderable through the training process amidst foreground and background classes (Lin, et al., 2017).

Tiny-YoloV3

Unlike other conventional algorithms which have been used for the detection process, Yolo utilizes one convolutional network for the whole picture and then, class prediction and bounding boxes are predicted together for those boxes. However, the tiny-Yolov3 model is a new version of YoloV3 which has used the DarkNet53 architecture. Tiny-yolov3 minimizes the convolutional layers' number and the keyframe of this model consists of seven convolutional layers (D. Xiao, et al., 2017). The following Fig. 5 illustrates the architecture of tiny-yolov3 (Ma J, Chen L, Gao Z, 2017). In this model, the feature extraction process is done by using a number of 1x1 and 3x3 convolutional layers. It utilizes the layer of pooling rather than the YoloV3 convolutional layer. In the process of training, the loss function used in this model is selfsame which has been used in YoloV3. The loss function is fundamentally shaped by the center of the prediction frame (x, y), size of prediction frame (w, h), class of prediction (class), and the confidence of prediction (confidence) (D. Xiao, et al., 2017).

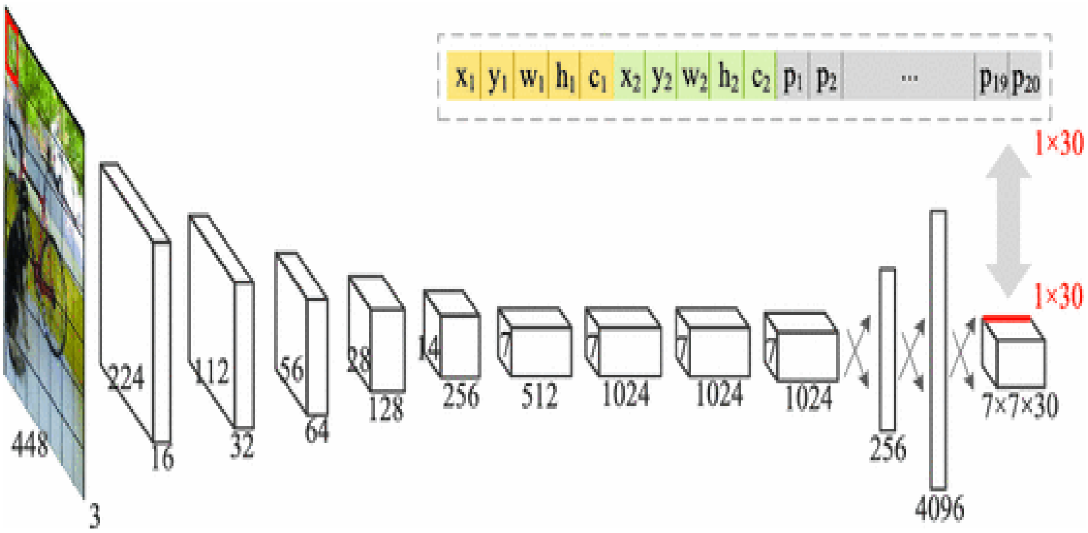

Figure 5. The architecture of tiny-Yolov3 [28]

SqueezeNet

In this section, SqueezeNet architecture will be discussed. This model has little parameters when maintaining accuracy compared to other models. Its main architecture consists of an independent convolutional layer pursued by eight modules of fire, and it ends with the ultimate convolutional layer. Then, the number of filters is progressively maximized per module of fire from the start until the network ends up. However, max pooling is implemented in this model. Figure 6 shows the architecture of SqueezeNet (Guo, et al., 2017).

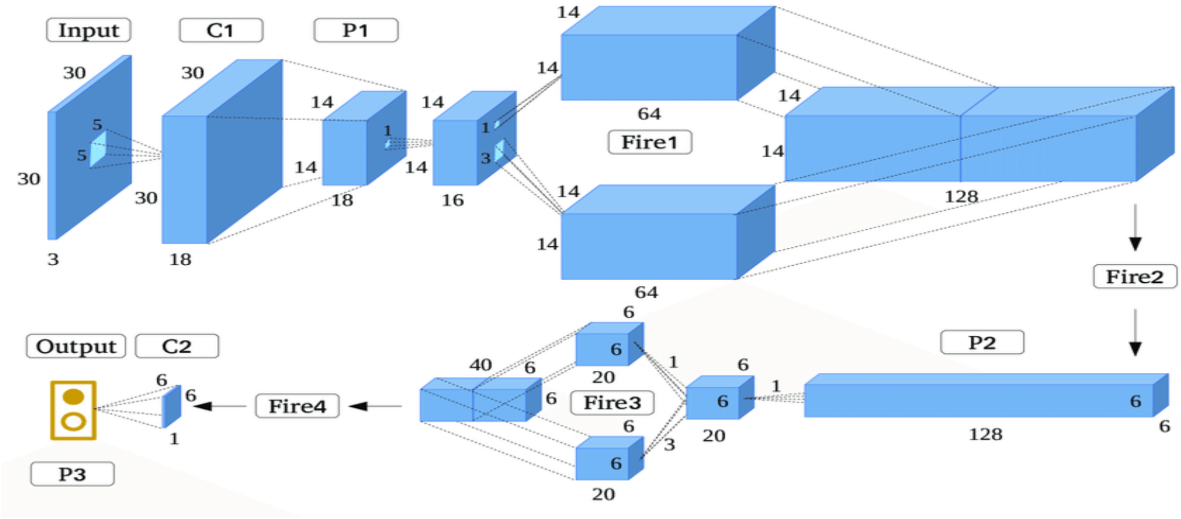

Figure 6. The architecture of Squeeze Net (Guo, et al., 2017).

The module of fire is constituted of 1x1 filters which exist in the squeeze-convolutional layer. Then, it feeds it up into an extended layer which has a blend of 1x1 and 3x3 filters. The fire frame seems like a fire burning over a matchstick. However, it reduces the volume of input canals resulting in minimizing the parameters' number. Larger maps of activation are obtainable for convolutional layers in the last phase by using the process of downsampling. Thus, it drives a rising accuracy of classification. Eventually, the pooling layer is utilized to generate the output immediately (Guo, et al., 2017).

DenseNet

Unlike the conventional algorithms which use N layers having N connections, one in the midst of every layer and its next layer, this model connects every layer to another one using the feed-forward method (G. Huang, et al., 2017). It includes a convolutional layer called feature-layer which is responsible for taking the feature of low-level from the images. In addition, it has diverse dense blocks, and layers of transition in the midst of these closed blocks. Dense-layer gets the outputs from the past layer and gets them concatenated in the deepness of the dimension. Figure 7 illustrates the architecture of DenseNet (G. Huang, et al., 2017).

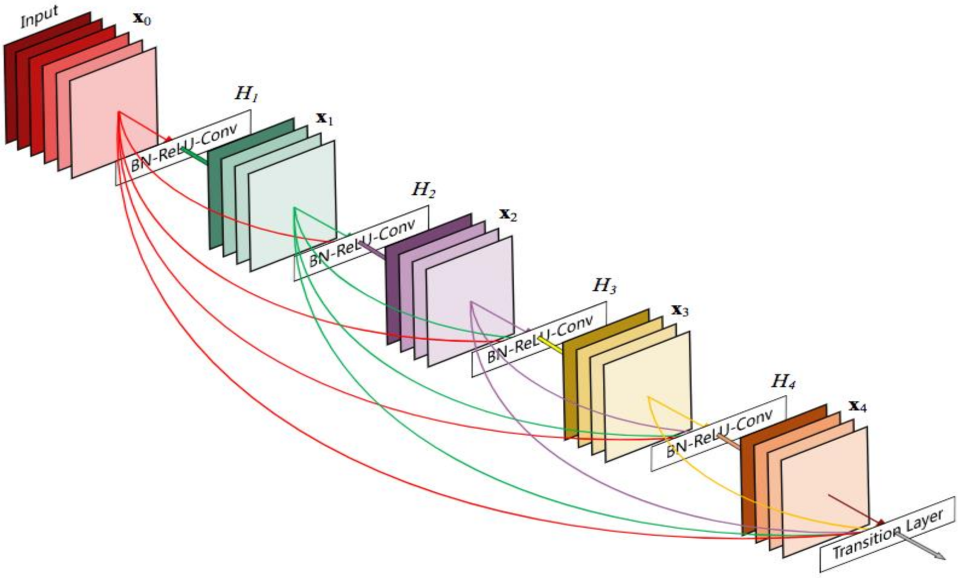

Figure 7. The architecture of DenseNet [30]

In DenseNet, a bottleneck-layer which is a 1x1 convolutional-layer is used in order to enhance the computational efficacy by minimizing the input of feature-maps. This process gives a stable input of deepness for the second convolutional layer. Furthermore, a transition layer is taking charge of minimizing the size of output (feature-maps) for each layer to extract the high grade of features. However, dense blocks hold the size and deepness. To minimize the size to the mid, the transition layer which includes a 2x2 moderate pooling layer and a 1x1 convolutional layer is used in this model. Finally, the efficacy of parameters is obtainable due to the few parameters used in DenseNet.

Experiments

Environment Setup

The experiments of this study were conducted on Windows 7 OS with 4 GB RAM. In addition, Python 3.7 with the Anaconda platform has been utilized for experiments to be carried out. Other packages like Keras, Tensorflow, and OpenCV are also utilized. On the other hand, four models have been used in this study and they had explained in section 3.1 clearly. Furthermore, the evaluation of these models was done on the COCO dataset.

Implementation Details

Since pre-trained models are used, the training phase is passed and the testing phase proceeds to evaluate the pre-trained models. These models are: resnet50_coco_best_v2.0.1 (145MB), tiny-Yolo (33.9MB), squeezenet_weights_tf_dim_ordering_tf_kernels (4.8MB), DenseNet-BC-121-32 (31.6MB)). In the testing phase, a given image which was taken from COCO dataset should be as an input to the pre-trained models, whereas the output should be the main image with the percentage probability and particular bounding-boxes around the detected objects. Figure 8 illustrates the framework of object detection and image recognition.

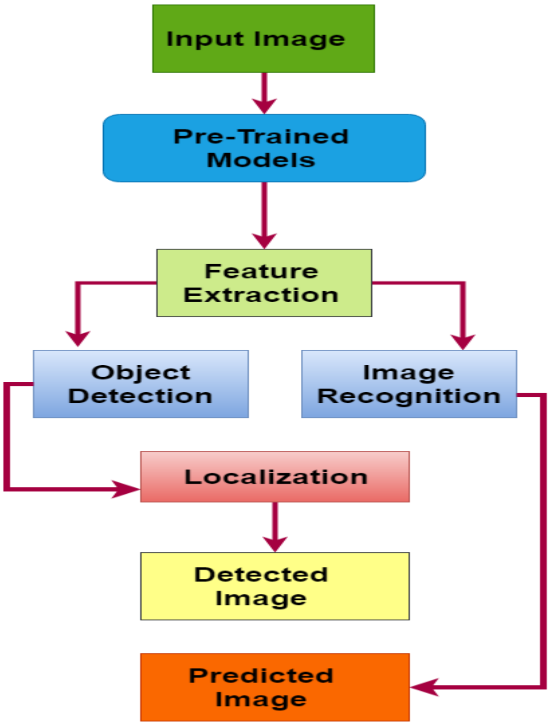

Figure 8. Proposed object detection and image recognition framework

Results and Discussion

Object Detection Part

We discuss in this part the results regarding object detection models to assess the abilities of these models, and this can be derived by the time taken to execute a given sample of images and the percentage probability for the detected objects. ResNet 50 and TinyYoloV3 are executed and Figure 9 illustrates the taken time for execution.

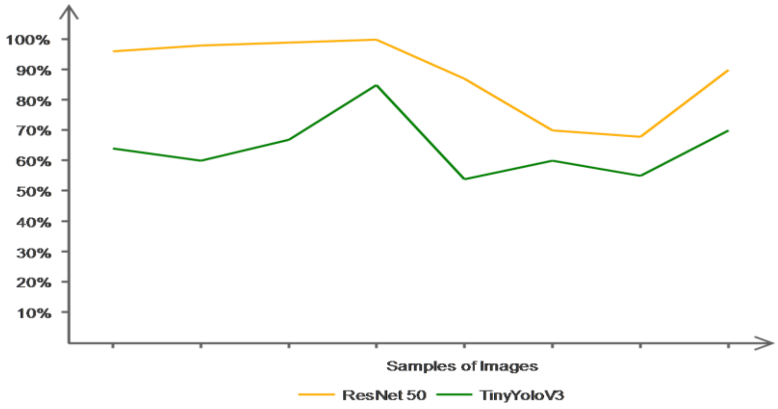

Figure 9. The execution time of ResNet 50 and TinyYoloV3

It is clear that tiny-YoloV3 outperforms on ResNet 50 in terms of speed since it takes less time as it is observed in the figure 9 but at the same time, it lacks probability percentage in the prediction, especially for small objects. In general, the rate varies from 75% to 90% for the large objects in ResNet 50. Whereas in Tiny-YoloV3, the rate varies from 35% to 80% for large objects generally. Figure 10 shows a sample of images for the detected objects with their probability percentage for both ResNet 50 and tiny-YoloV3.

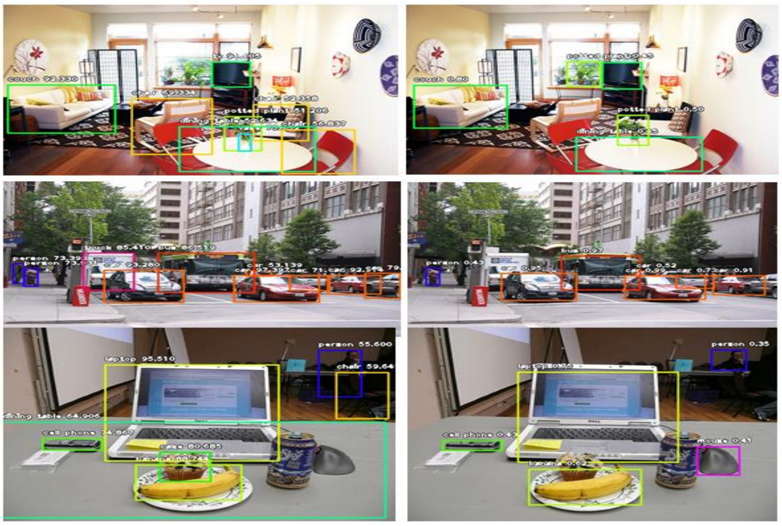

Figure 10. Object detection using ResNet 50 (left) and TinyYoloV3 (right)

Despite the lack of speed in execution time, ResNet 50 has a great performance in the probability percentage and comprehensiveness of objects, thus, the significance of the backbone is clarified. As we can see in figure 10, ResNet 50 extracts more objects so the rise in execution time is sensible. On the other hand, TinyYoloV3 consider the best model to be picked up for the applications when the speed of execution time is needed like in real-time applications. While ResNet 50 is required for those applications which need high accuracy of prediction due to the nature of the scope like medical image classification. Figure 11 illustrates the probability percentage of ResNet 50 and TinyYoloV3 for different images.

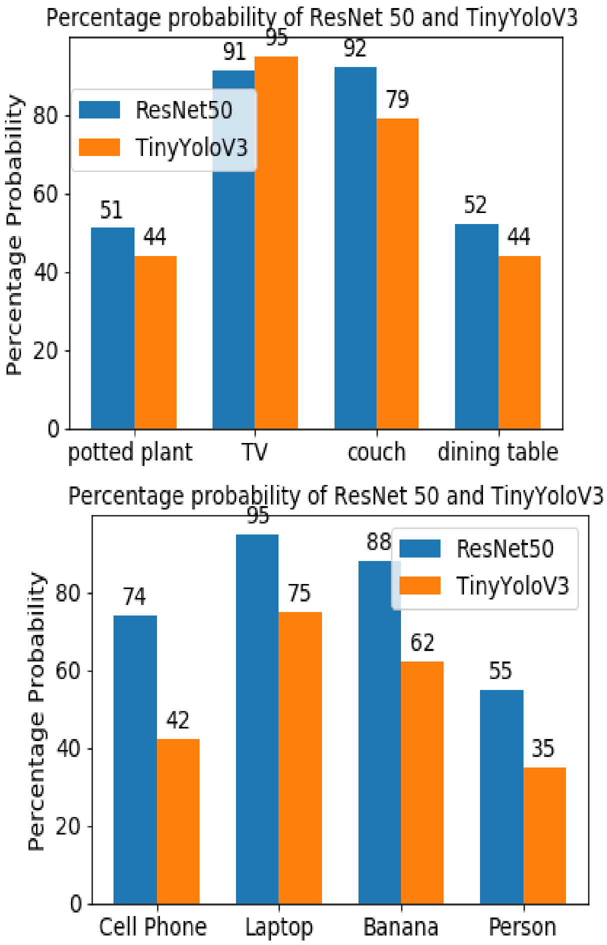

Figure 11. The probability percentage of ResNet 50 and TinyYoloV3

Image Recognition Part

In this part, we debate the results concerning image recognition models to evaluate the capabilities of these models, and this can be achieved by the execution time for a given sample of images and also the probability percentage for the detected objects. Squeeze Net and DenseNet are executed and figure 12 illustrates the taken time for the execution.

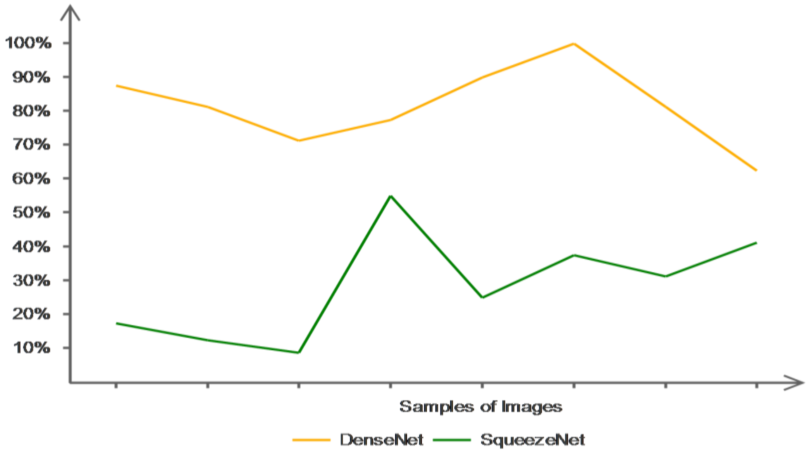

Figure 12. The execution time of Squeeze Net and Dense Net

As seen in the experiment, Squeeze Net outperforms on DenseNet in terms of execution time speed. Squeeze Net model has little parameters when maintaining accuracy compared to other models, so less time in the execution is sensible. Whereas DenseNet model has also fewer parameters but at the same time has diverse dense blocks so the execution time is raised. Figure 13 shows the probability percentage of Squeeze Net and DenseNet for different images.

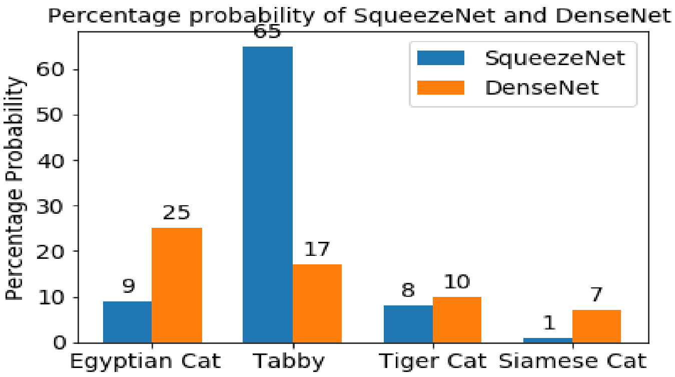

Figure 13. The probability percentage of Squeeze Net and DenseNet

Furthermore, the size of the model is minimized in Squeeze Net model due to the little number of fully-connected layers besides the fire modules. In addition, its small size along with less execution time makes it suitable for the applications of mobile. Furthermore, DenseNet is adequate for the applications of identification due to the reuse of features. Finally, reducing the parameters and reusing features have a great impact on the size of the model compared to other different models. Finally, Table 1 illustrates the comparison between image recognition models and object detection models in terms of execution time. It is obvious that image recognition models take less time than object detection models but it gives less probability percentage at the same time.

Samples	Image Recognition		Object Detection	
	DenseNet	SqueezNet	ResNet 50	TinyYolo V3
1	62 sec	44 sec	96 sec	64 sec
2	57 sec	7 sec	98 sec	60 sec
3	65 sec	10 sec	132 sec	67 sec
4	70 sec	14 sec	182 sec	85 sec

Conclusion

In this study, an experiential comparison of four object detection and recognition models based on deep convolutional neural network is introduced. We have analyzed the major issues of these models, the execution time, and the probability percentage of prediction. The models of object detection used in this paper were pre-trained on the COCO dataset, whereas the models of image recognition were pre-trained on the ImageNet-1000 dataset to detect objects and predict the items of a given image. We found in the object detection part that Tiny-YoloV3 outperforms ResNet 50 in terms of execution time. Furthermore, ResNet 50 has a great performance of percentage probability of prediction. Moreover, we found in the image recognition part that the Squeeze Net outperforms the DenseNet in terms of execution time and the percentage probability of prediction. Thus, we have clarified that each model serves better in some applications based on the execution time and percentage probability of prediction. The future work considers using Tiny-YoloV3 for real-time object recognition of a drone system.

Conflict of interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article

References

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” In: Advances in Neural Information Processing Systems 25, Ed, by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Curran Associates, Inc., pp. 1097–1105, 2012.

Awan, M.J., Masood, O.A., Mohammed, M.A., Yasin, A., Zain, A.M., Damaševičius, R. and Abdulkareem, K.H., Image-Based Malware Classification Using VGG19 Network and Spatial Convolutional Attention. *Electronics*, *10*(19), p.2444. 2021.

Cocodataset.org, “COCO - Common Objects in Context,” [Online], Available at: http://cocodataset.org/#download [Accessed 14 June, 2022].

D. Xiao, F. Shan, Z. Li, B. T. Le, X. Liu and X. Li, "A Target Detection Model Based on Improved Tiny-yolov3 Under the Environment of Mining Truck," in IEEE Access, doi: 10.1109/ACCESS, 2928603, 2019.

G. Huang, Z. Liu, L. v. d. Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, pp. 2261-2269, doi:10.1109/CVPR.2017.243, 2017.

Guo, Zhiling et al, “Village Building Identification Based on Ensemble Convolutional Neural Networks”, Sensors (Basel, Switzerland) vol, 17,11 2487, Doi: 10.3390/s17112487, 30 Oct, 2017.

Huh, M., Agrawal, P, and Efros, A. A, “What makes ImageNet good for transfer learning?,” arXiv preprint arXiv,1608.08614, 2016.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” In CVPR, 2, 4, 5, 6, 8, 2016.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” In Advances in Neural Information Processing Systems, 1097–1105, 2012.

Lal, S., Rehman, S.U., Shah, J.H., Meraj, T., Rauf, H.T., Damaševičius, R., Mohammed, M.A. and Abdulkareem, K.H., Adversarial attack and defence through adversarial training and feature fusion for diabetic retinopathy recognition. *Sensors*, *21*(11), p.3922. 2021.

Lin, T.Y, Doll´ar, P, Girshick, R, He, K, Hariharan, B, Belongie, S, “Feature pyramid networks for object Detection,” arXiv preprint, arXiv:1612.03144, 2016.

Lin, T.Y, Goyal, P, Girshick, R, He, K, Doll´ar, P, “Focal loss for dense object detection,”arXiv preprint, arXiv:1708.02002, 2017.

Ma J, Chen L, Gao Z, “Hardware Implementation and Optimization of Tiny- YOLO Network,” In Zhai G, Zhou J, Yang X, (eds) Digital TV and Wireless Multimedia Communication, Communications in Computer and Information Science, vol 815, Springer, Singapore, IFTC 2017.

Mujahid A, Awan MJ, Yasin A, Mohammed MA, Damaševičius R, Maskeliūnas R, Abdulkareem KH. Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model. *Applied Sciences*,11(9):4164, 2021.

Nisa, M.U., Mahmood, D., Ahmed, G., Khan, S., Mohammed, M.A. and Damaševičius, R., Optimizing Prediction of YouTube Video Popularity Using XGBoost. *Electronics*, *10*(23), p.2962. 2021.

OlafenwaMoses/ImageAI", GitHub, [Online], Available at: "ImageAI", Imageai.org, [Online], Available at http://imageai.org/, [Accessed 14 June, 2022].

Redmon, J, Divvala, S, Girshick, R, Farhadi, A., "You only look once: Unified, real- time object detection," In: CVPR, 2016.

Redmon, J, Farhadi, A, "Yolo 9000: Better, faster, stronger," arXiv preprint, arXiv:1612.08242, 2016.

Ren, S, He, K, Girshick, R, Sun, J, "Faster r-cnn: Towards real-time object detection with region proposal Networks," In Advances in neural information processing systems, pp. 91– 99, 2015.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE transactions on pattern analysis and machine intelligence, 39 (6): 1137– 49, 2017.

Russakovsky, O, Deng, J, Su, H., Krause, J., Satheesh, S, Ma, S, Huang, Z, Karpathy, A, Khosla, A, Bernstein, M., et al, "ImageNet large scale visual recognition challenge," International Journal of Computer Vision, 115 (3), 211–252, 2015.

Sermanet, Pierre, and Yann LeCun, "Traffic Sign Recognition with Multi Scale Networks," Courant Institute of Mathematical Sciences, New York University. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6033589, 2011.

Sheu, J.S. and Chen-Yin, H., 2019, "Combining Cloud Computing and Artificial Intelligence Scene Recognition in Real-time Environment Image Planning Walkable Area," Advances in Technology Innovation, 5(1), p.10,2019.

Szegedy, C, Liu, W, Jia, Y, Sermanet, P, Reed, S, Anguelov, D, Erhan, D Vanhoucke, V, Rabinovich, A "Going deeper with convolutions," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9, 2015.

T.-Y. Lin, P. Doll´ar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object Detection," In CVPR, 1, 2, 4, 5, 6, 8, 2017.

Xie, S, Girshick, R, Doll´ar, P, Tu, Z, He, K, "Aggregated residual transformations for deep neural networks," In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5987–5995, IEEE, 2017.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," In Neural Comput. 1.4 pp. 541–551. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1. 4.541, Dec, 1989.

Yosinski, J, Clune, J, Bengio, Y, and Lipson, H, "How transferable are features in deep neural networks?" In Advances in neural information processing systems, pages 3320{3328}, 2014.

Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. "DetNet: Design backbone for object detection," In ECCV, 2018.

Bibliographic information of this paper for citing:

Obaid, Omar Ibrahim; Mohammed, Mazin Abed; Salman, Akbal Omran; Mostafa, Salama A. & Elngar, Ahmed A. (2022). Comparing the Performance of Pre-trained Deep Learning Models in Object Detection and Recognition. *Journal of Information Technology Management,* 14 (4), 40-56. https://doi.org/ 10.22059/jitm.2022.88134