COMPARISON OF COMPUTER VISION ARCHITECTURES PERFORMANCE TO IMPROVE THE HEALTH PROTOCOL VIOLATIONS DETECTION

[1]Stephen Royanmart Patrick, [2]Robertus Setiawan Aji Nugroho
[1,2]Informatics Engineering, Faculty of Computer Science,
Soegijapranata Catholic University
[2]nugroho@unika.ac.id

ABSTRACT

The COVID-19 pandemic is a big problem for the world. Many things have been done to solve the problem of Covid-19, one of which is the prevention of transmission. Prevention of the transmission of COVID-19 has been carried out by many methods, one of which is the creation of a detection system based on computer vision technology.

To improve the performance of the system, researchers conducted special research that compared the performance between 3 architectures, Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures. SSD ResNet50 V1 FPN was found as the best model in this test. That is because in two experiments the researcher got that model has consistency in performance. In the first experiment, mean average precision, mean average precision of medium images, mean average precision of small images, average recall, average recall for large images, average recall of medium images, and an average recall of small images SSD ResNet50 V1 FPN has the better results than others. In the second experiment, mean average precision, mean average precision of large images, mean average precision of medium images, mean average precision of small images, average recall, average recall of medium images, and an average recall of small images.

Keywords: Computer Vision, Health Violations Detection, Pandemic, Artificial Intelligence

INTRODUCTION

Until now, the Covid-19 virus has claimed a large number of victims. This causes a lot of losses, both economic losses, and other losses. On the other hand, the administration of the Covid-19 vaccine is still in progress and takes time. The best solution to this problem is the prevention of transmission.

To prevent the spread of the Covid-19 virus, experts around the world have tried various ways. One way is to give an appeal for the use of health protocols in the form of the use of masks and social distancing. Unfortunately, not all communities responded and obeyed the appeal. It should also be realized that experts cannot fully monitor people who are active. The role of the general public such as business owners, security units, and other parties is very much needed in monitoring the use of these health protocols.

Computer vision technology can be very helpful in monitoring violations of the use of masks and social distancing. For this, an effective architecture is needed in detecting objects in the image. From this research, the researcher wants to explore “which architectures are more effective for

mask detection from 3 architectures, 'Faster R-CNN ResNet50 V1 640x640', 'SSD ResNet50 V1 FPN 640x640 (RetinaNet50)' and 'SSD MobileNet V2 320x320'"

LITERATURE STUDY

Based on Zou et al. paper [1], object detection is an important computer vision task that deals with detecting instances of visual objects of a certain class (such as humans, animals, or cars) in digital images. For object detection, there are some models and applications that can be used. One of the models that are popular now is deep learning. Based on LeCun et al. paper [2], "Computational models with numerous processing layers can learn multiple degrees of abstraction for data representations using deep learning. Speech recognition, visual object recognition, object detection, and a range of other industries like medicine discovery and genomics have all benefited greatly from these techniques. The backpropagation technique describes how a computer should alter its internal settings to compute the representation in each layer from the representation in the previous layer, allowing it to find intricate structures in large data sets".

To develop the deep learning model, there are some frameworks that can be used. The examples are TensorFlow and PyTorch. Based on Google Trends, specifically for Indonesia in 1 year until Oct 22, 2021 [3], TensorFlow is more popular. Based on the tensorflow.org website [4], "TensorFlow is an open-source machine learning platform that runs from start to finish. It features a robust ecosystem of tools, libraries, and community resources that enable researchers to advance the state-of-the-art in machine learning and developers to quickly build and deploy ML-powered apps".

Specifically to mask detection, until now some research has already been done using various architectures. An example is face mask detection that was already developed by Chowdary et al [5]. They implemented InceptionV3 for face mask detection. InceptionV3 is a 48 layered convolutional neural network architecture developed by Google. They use 1570 images that consist of 785 simulated masked facial images and 785 unmasked facial images. From this dataset, 1099 images of both categories are used for training, and the remaining 470 images are used for testing the model. The implementation of InceptionV3 architecture has high accuracy (up to 100%). Although the accuracy is so high, based on Bianco et al. paper [6], there are some architectures that need more light memory consumption and have more FPS. An example is the MobileNet architecture.

Because of the problem from the previous paragraph, the researcher thinks that the researcher needs to learn about the other algorithm that is more effective for the detection to make a decision about what architecture will use for implementation. The researcher learns about the implementation of other object detection problems. Alising [7] implemented several object detection architectures on mobile devices. The researcher reviewed the performance of Post-it detection using several models like Tiny YOLO V2, SSD MobileNet V1, SSD MobileNet V2, SSD Inception V2, Faster RCNN Inception V2, and Faster RCNN ResNet50 that are implemented through Android devices. He used Tensorflow Lite for the experiment. He used many Post-it

images dataset from Bentouch and Instagram. He wanted to know the performance of each model. From this research, I got some knowledge that for this case Faster RCNN and SSD have better mAP (Mean Average Precision) than Tiny YOLO V2. Faster RCNN ResNet50 99,33%, Faster RCNN Inception V2 96.69%, SSD Inception V2 96.82%, SSD MobileNet V2 91.90%, SSD MobileNet V1 91.16% & Tiny Yolo V2 87.57%. For the inference time, Faster RCNN ResNet50 got 20018 ms, Faster RCNN Inception V2 got 4105 ms, SSD Inception V2 got 716 ms, SSD MobileNet V2 438 ms, SSD MobileNet V1 got 454 ms & Tiny Yolo V2 got 515 ms. From this paper, I consider learning more about SSD MobileNet V2 the next time.

In the other research, Campoverde and Barros [8] wanted to compare the performance of Deep Neural Networks (DNN) for detecting 6 classes based on urban actors (car, bus, truck, bicycle, motorcycle, person). The researchers used "videos which were captured from the side and had the total count of each mobility actor" and "collected images from 2 cities" for the dataset. They want to know about the precision, recall, and inference time of SSD models using Tensorflow lite. Unfortunately, because they only used the SSD model, I can't compare the other architecture's performance from this paper.

Choe et al. [9] also develop some object detection systems that can classify the parrot species from android devices. The researchers used many datasets which consist of 4 species. In 1 species, there are 2800 images for training, and 700 for validation. They wanted to know about the accuracy of detecting parrot species. The researchers use eight models: ResNet50, NASNetMobile, InceptionResNetV2, and InceptionV3, with two initialization pre-trained ImageNet weights or random numbers. Results of the pre-trained Imagenet scheme for ResNet50, ResNet50 got an F1 score of 94% from A. Chlotoptera detection, 90% from C. Galerita detection, 84% from C. Goffinianan detection & 89% from P. Erithacus detection. For the random numbers train, ResNet50 got an F1 score of 94% from A. Chlotoptera detection, 59% from C. Galerita detection, 56% from C. Goffinianan detection & 81% from P. Erithacus detection. Results of the pre-trained Imagenet scheme for InceptionResNetV2, InceptionResNetV2 got an F1 score of 98% from A. Chlotoptera detection, 95% from C. Galerita detection, 93% from C. Goffinianan detection & 97% from P. Erithacus detection. For the random numbers train, InceptionResNetV2 got an F1 score of 87% from A. Chlotoptera detection, 76% from C. Galerita detection, 68% from C. Goffinianan detection & 78% from P. Erithacus detection. Results of the pre-trained Imagenet scheme for NASNet Mobile, NASNet Mobile got an F1 score of 100% from A. Chlotoptera detection, 94% from C. Galerita detection, 93% from C. Goffinianan detection & 100%% from P. Erithacus detection. For the random numbers train, NASNet Mobile got an F1 score of 95% from A. Chlotoptera detection, 77% from C. Galerita detection, 65% from C. Goffinianan detection & 83% from P. Erithacus detection.

In Deepa et al. research [10], the researchers wanted to know about many models (YOLO, SSD, Faster RNN) performance for real-time tennis ball tracking. The researcher used datasets from the tennis court, which consist of videos and images which were collected from various angles and lighting conditions. They wanted to know about the best model, which is evaluated by

the processing time. For the training, the researchers used around 5000 images and used 200.000 steps. For the evaluation, the researchers use approximately 10 images. From this research, I got the knowledge that SSD architecture is a much efficient and comparatively more accurate algorithm with less computational speed for this particular task.

In Fan et al. research [11], the researchers wanted to develop a smartphone application based on the classification of rock lithology. The researchers used 3795 images which consist of 30 types of rock. 3046 images for training, 381 for verification, and 365 for testing. They wanted to know about the best model, which is evaluated by accuracy. From this paper, I got the knowledge that SqueezeNet was more stable than the MobileNet and ShuffleNet model for classifying rock lithology. But if we need faster train time, ShuffleNet can be the alternative.

In Ghoury et al. research [12], the researchers wanted to compare 2 models SSD Mobilenet V1 and Faster R-CNN performance for detecting grape disease. The researchers used 543 images for training and 113 images for testing which consist of 4 classes (Healthy Grape, Diseased Grape, Healthy Grape Leaf, and Diseased Grape leaf). They wanted to know about the best model based on accuracy and time. From this paper, I got the knowledge that the overall Faster R-CNN model has better accuracy (95.57%) than the SSD MobileNet V1 model (59.29%) for the grape disease classification.

In Hung and Kien research [13], the researchers wanted to improve the system that can classify fish species using some models. The researchers used many images which consist of 1187 yellow tuna images, 1415 striped bass images, 1131 brook bass images, and 1170 snapper images. 80% of images were used for training, and 20% of images were used for testing. They used Average Precision and Average Recall values as the indicator. This paper is useful for my research because I can know about the comparison of Faster-RCNN-Inception V2, SSD Mobilenet, and SSD Inception V2 (from precision/recall side and time side). From this paper, I got the knowledge that mixed SSD and MobileNet models have good classification results. But the SSD-InceptionV2 overall has better performance.

In [14], the researchers wanted to develop the novel images classification with some methods (DenseNet, MobileNets, Dense1-MobileNet, Dense2-MobileNet). The researchers used so many datasets. First from Caltech-101 (9145 images with 102 classes), 1500 images for testing, and the rest for training. The second from The Uebingen Animals datasets (22742 images with 21 classes), 2000 images for testing, and the rest for training. For the evaluation, the researchers used an accuracy indicator. This paper is useful for my research because I can learn about the comparison of detection performance from some methods. The main limitation of this paper is, this paper uses accuracy for the indicator. Sometimes if the number of datasets is very different, accuracy parameters are worse than precision/recall. This paper will fit my research because I got a lot of information, especially in the implementation of the Dense MobileNet model.

In [15], the researchers wanted to improve the deep learning system for detecting tomato disease (change VGG16 to res101). The researchers used 4178 images which consist of 1 class of healthy tomato images and 4 classes of tomato disease. For the evaluation, the researchers used

the mean average precision (mAP) indicator. This paper is useful for my research because I can learn about the comparison of detection performance from some methods. The main limitation of this research is the time it takes longer. If using Faster RNN-res101, the model needed 462ms. It's longer than the Faster RNN-Mobile model (142). But on the mAP side, Faster RNN-res101 gave the better mAP.

After reviewing several papers, I decided to explore more about the SSD (Single Shot MultiBox Detector) model. From the Liu et al. research [16], the SSD model was created as a result of the researcher's desire to create a model that can detect objects in photos using only one deep neural network. PASCAL, VOC, COCO, and ILSVRC datasets were employed by the researchers. The researcher employed the mean average precision (mAP) indicator to assess the situation. This paper is valuable for my research because it informs me about the SSD approach in detail. The SSD method generates a fixed-size collection of bounding boxes and scores for the existence of object class instances in those boxes using a feed-forward convolutional network, followed by a non-maximum suppression step to produce final detections.

RESEARCH METHODOLOGY

Literature Study

Before the researcher decides to implement the fixed architecture that will be used, the researcher decided to read several scientific pieces of literature about object detection. From this step, the researcher expects roughly what architecture will be used.

Data Collection

The dataset was obtained from the Kaggle website (https://www.kaggle.com/andrewmvd/face-mask-detection). The existing dataset is used as training data for the artificial intelligence model which consists of three classes, namely wear a mask, not wear a mask & wear a mask but wrong. The total dataset is 853 images & 853 XML annotated files.

Models Config Preparation

Before the training and evaluation, the researcher needs to prepare the config that will be implemented in the framework. To all config, the researcher set 8 batch sizes for training & 5000 to the number of steps.

Conversion of the RAW Dataset to TFRecord Data

The data that was used was converted from the images to the TF Record format. Based on the TensorFlow website [17], "The TFRecord format is a basic binary record storage format". This action made the data easier to process in the next step.

Framework Design for Training & Evaluating

The detection system is a system that is trained to detect the use of masks (Wear a mask, not wear a mask, wear a mask but wrong & back face). The researcher made one framework Python code and implement three models which consist of "Faster R-CNN ResNet50 V1 640x640" / "SSD ResNet50 V1 FPN 640x640 (RetinaNet50)" / "SSD MobileNet V2 320x320". For the training and evaluation, the researcher did 2 experiments. First using 2:8, 5:5 and 8:2 data scale, for the second using 2:2, 5:2 and 8:2 data scale.

RESULTS

RAW Data

Experiment 1 (2:8, 5:5, 8:2 data scale)

Table 1. Experiment 1 - RAW output of Faster R-CNN ResNet50 V1 architecture

Faster R-CNN ResNet50 V1 640x640 (5000 steps, 8 Batch Size)								
Data Scale	**mAP**	**mAP Large**	**mAP Medium**	**mAP Small**	**AR @100**	**AR @100 Large**	**AR @100 Medium**	**AR @100 Small**
2:8	0,1516	0,5235	0,2565	0,07701	0,3129	0,6565	0,4781	0,2134
5:5	0,221	0,6474	0,3246	0,1544	0,4149	0,7512	0,571	0,3195
8:2	0,2635	0,7802	0,3605	0,186	0,4403	0,7998	0,5579	0,3625

Table 2. Experiment 1 - RAW output of SSD ResNet50 V1 FPN architecture

SSD ResNet50 V1 FPN 640x640 (5000 steps, 8 Batch Size)								
Data Scale	**mAP**	**mAP Large**	**mAP Medium**	**mAP Small**	**AR @100**	**AR @100 Large**	**AR @100 Medium**	**AR @100 Small**
2:8	0,2322	0,5058	0,297	0,1795	0,3694	0,6983	0,4847	0,2898
5:5	0,3364	0,6123	0,3701	0,3022	0,5105	0,7886	0,6007	0,4423
8:2	0,3985	0,7733	0,4461	0,3633	0,5471	0,7883	0,6389	0,4806

Table 3. Experiment 1 - RAW output of SSD MobileNet V2 architecture

SSD MobileNet V2 320x320 (5000 steps, 8 Batch Size)								
Data Scale	**mAP**	**mAP Large**	**mAP Medium**	**mAP Small**	**AR @100**	**AR @100 Large**	**AR @100 Medium**	**AR @100 Small**
2:8	0,153	0,5913	0,2655	0,07084	0,2981	0,7017	0,4983	0,1775
5:5	0,1676	0,6234	0,2615	0,09134	0,3437	0,7163	0,5458	0,2177
8:2	0,1604	0,5273	0,2696	0,09545	0,3602	0,8016	0,5307	0,2447

Experiment 2 (2:2, 5:2, 8:2 data scale)

Table 4. Experiment 2 - RAW output of Faster R-CNN ResNet50 V1 architecture

Faster R-CNN ResNet50 V1 640x640 (5000 steps, 8 Batch Size)								
Data Scale	**mAP**	**mAP Large**	**mAP Medium**	**mAP Small**	**AR @100**	**AR @100 Large**	**AR @100 Medium**	**AR @100 Small**
2:2	0,1626	0,4342	0,2823	0,08499	0,3363	0,6423	0,4845	0,2318
5:2	0,2359	0,7342	0,3034	0,2097	0,4146	0,7525	0,5181	0,3506
8:2	0,2604	0,7699	0,3485	0,1938	0,417	0,7911	0,5078	0,3536

Table 5. Experiment 2 - RAW output of SSD ResNet50 V1 FPN architecture

SSD ResNet50 V1 FPN 640x640 (5000 steps, 8 Batch Size)								
Data Scale	**mAP**	**mAP Large**	**mAP Medium**	**mAP Small**	**AR @100**	**AR @100 Large**	**AR @100 Medium**	**AR @100 Small**
2:2	0,2377	0,4762	0,2831	0,2065	0,392	0,6563	0,4621	0,347
5:2	0,3437	0,6705	0,3268	0,347	0,5145	0,7084	0,5538	0,488
8:2	0,3967	0,8006	0,4261	0,371	0,5753	0,8296	0,6233	0,5324

Table 6. Experiment 2 - RAW output of SSD MobileNet V2 architecture

SSD MobileNet V2 320x320 (5000 steps, 8 Batch Size)								
Data Scale	**mAP**	**mAP Large**	**mAP Medium**	**mAP Small**	**AR @100**	**AR @100 Large**	**AR @100 Medium**	**AR @100 Small**
2:2	0,1376	0,4714	0,2282	0,07639	0,3381	0,7661	0,4985	0,2249
5:2	0,1725	0,5703	0,2924	0,1004	0,3765	0,7687	0,5422	0,2701
8:2	0,1667	0,5233	0,2785	0,1002	0,3626	0,7736	0,502	0,2745

Explanation

Experiment 1 (2:8, 5:5, 8:2 data scale)

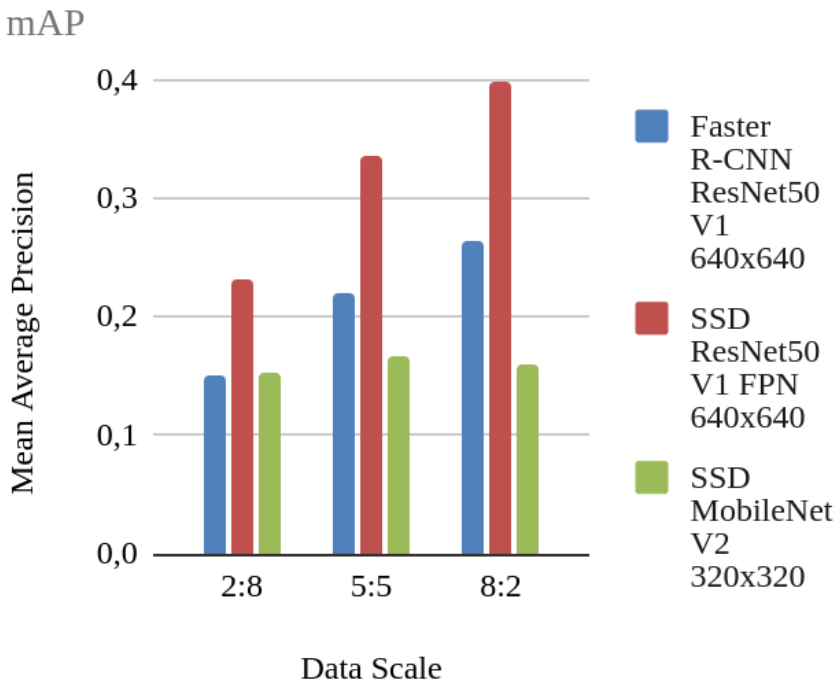

Figure 1. Experiment 1 - Mean average precision of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

Figure 1 shows the data of mean average precision from the evaluation section. From the chart above, we know that SSD ResNet50 V1 FPN architecture has the biggest mean average precision for detection. In a specific situation (2:8 data scale), Faster R-CNN ResNet50 V1 and SSD MobileNet V2 classification loss approximately the same. From this situation, overall we know that **SSD ResNet50 V1 FPN is more effective** than others in this case.

From another perspective, overall we know that more data makes the system more accurate. That's because data can know about various types of data (also avoid overfit).

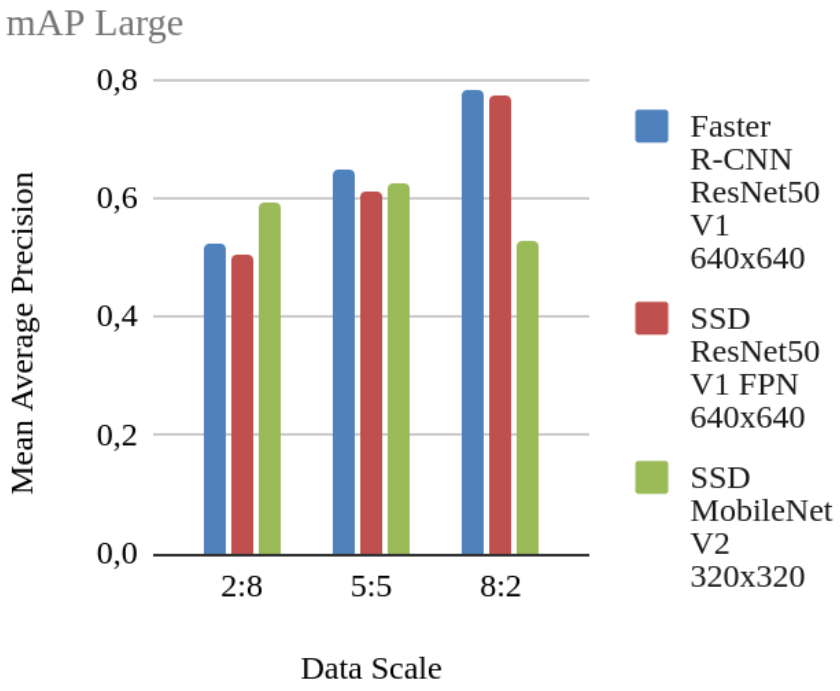

Figure 2. Experiment 1 - Mean average precision of large images size detection of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

Figure 2 shows the data of mean average precision of large images size detection from the evaluation section. From the chart above, we know that SSD MobileNet V2 architecture has the biggest mean average precision (specifically for large-size images) in the 2:8 data scale. On another scale, Faster R-CNN ResNet50 V1 has the biggest mean average precision (specifically for large images). From the overall experiment, we know that **Faster R-CNN ResNet50 V1 is more effective** than others in this case.

From another perspective, overall we know that more data made the system more accurate to detect large images (only SSD MobileNet V2 that gave the different result). That's because data can know about various types of data (also avoid overfit).

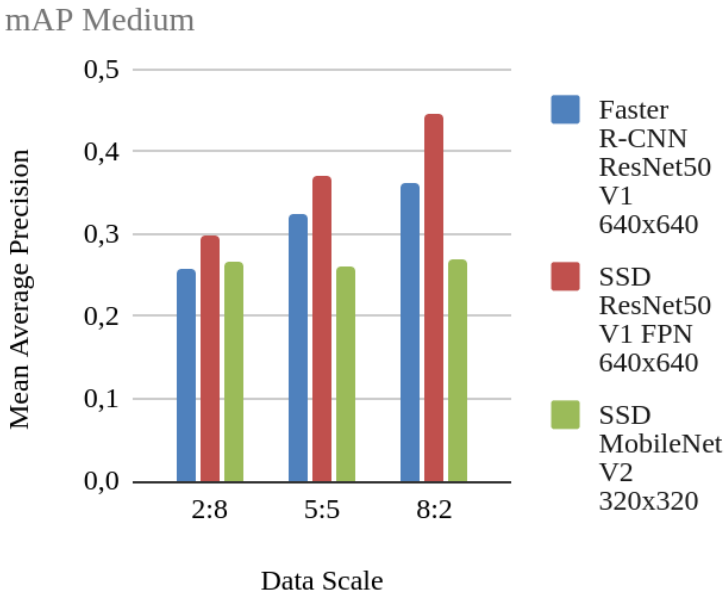

Figure 3. Experiment 1 - Mean average precision of medium images size detection of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

Figure 3 shows the data of mean average precision of medium images size detection from the evaluation section. This data is the data from the evaluating section. From the chart above, we know that SSD ResNet50 V1 FPN architecture has the biggest mean average precision (specifically for medium size images) for detection in all data scales. From this situation, overall we know that **SSD ResNet50 V1 FPN is more effective** than others in this case.

From another perspective, overall we know that more data made the system more accurate to detect medium size images (only SSD MobileNet V2 that gave the different result). That's because data can know about various types of data (also avoid overfit).

Figure 4 shows the data of mean average precision of small images size detection from the evaluation section. From the chart above, we know that SSD ResNet50 V1 FPN architecture has the biggest mean average precision (specifically for small size images) for detection in all data scales. From this situation, overall we know that **SSD ResNet50 V1 FPN is more effective** than others in this case.

From another perspective, overall we know that more data made the system more accurate to detect small-size images. That's because data can know about various types of data (also avoid overfit).

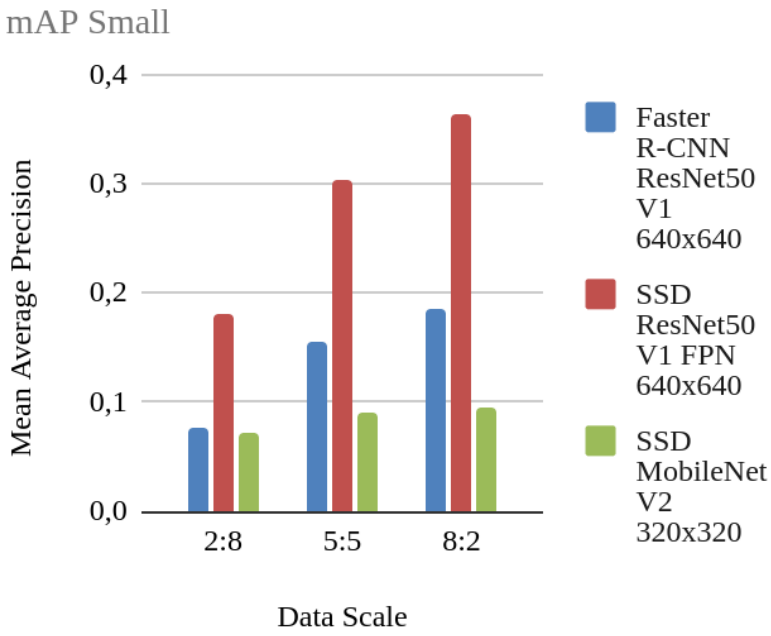

Figure 4. Experiment 1 - Mean average precision of small images size detection of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

Figure 5 shows the data of average recall from the evaluation section. From the chart above, we know that SSD ResNet50 V1 FPN architecture has the biggest average recall for detection in all data scales. From this situation, overall we know that **SSD ResNet50 V1 FPN is more effective** than others in this case.

From another perspective, overall we know that more data made the system more accurate to detect images. That's because data can know about various types of data (also avoid overfit).

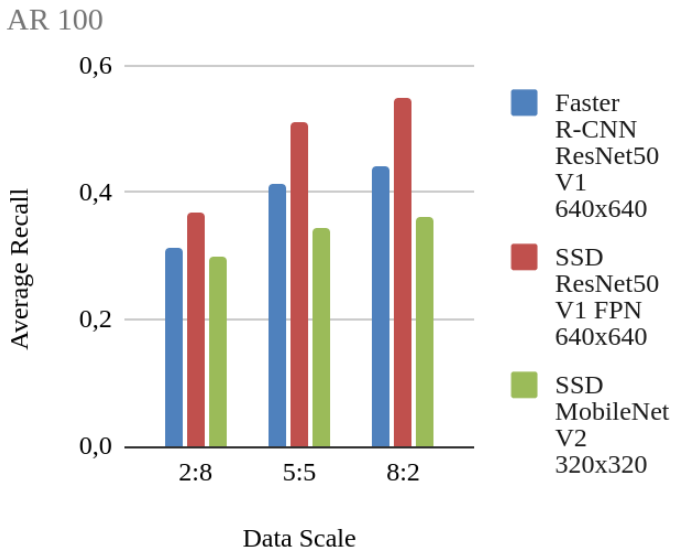

Figure 5. Experiment 1 - Average recall of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

Figure 6 shows the data of average recall of large images size detection from the evaluation section. From the chart above, we know that SSD ResNet50 V1 FPN architecture has the biggest average recall for detection in the 2:8 and 5:5 data scales. But on the 8:2 scale, SSD MobileNet has the biggest average recall. Overall from this situation, overall we know that **SSD ResNet50 V1 FPN is more effective** than others in this case.

From another perspective, overall we know that more data made the system more accurate to detect large-size images (only SSD ResNet50 V1 FPN gave the different result). That's because data can know about various types of data (also avoid overfit).

Figure 7 shows the data of average recall of medium images size detection from the evaluation section. From the chart above, we know that SSD ResNet50 V1 FPN architecture has the biggest average recall for detection overall. But in the 2:8 scale, SSD MobileNet V2 is more accurate. Overall from this situation, overall we know that **SSD ResNet50 V1 FPN is more effective** than others in this case. From another perspective, for the Faster R-CNN ResNet 50 V1 & SSD MobileNet V2, the data scale didn't do much to increase recall.

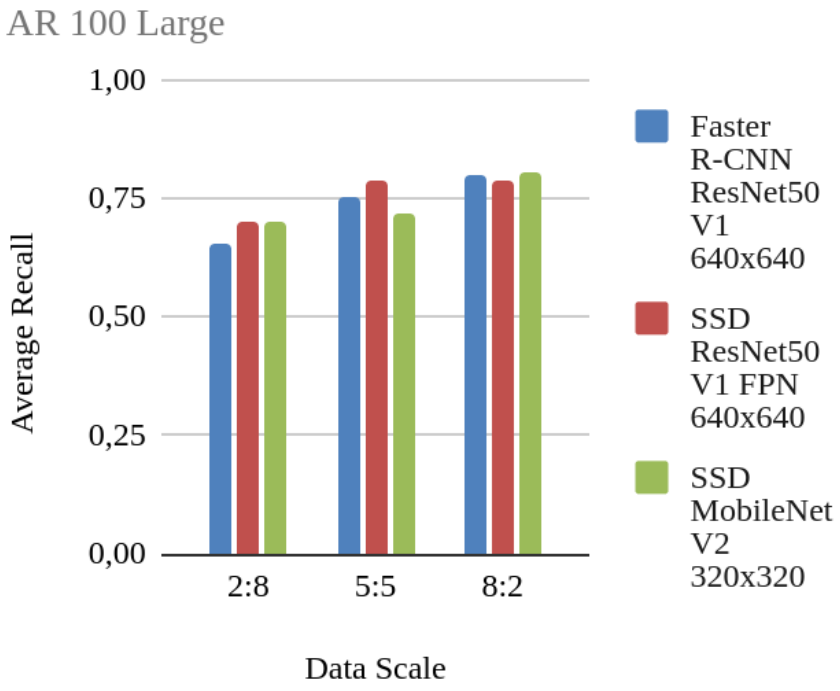

Figure 6. Experiment 1 - Average recall of large image size detection of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

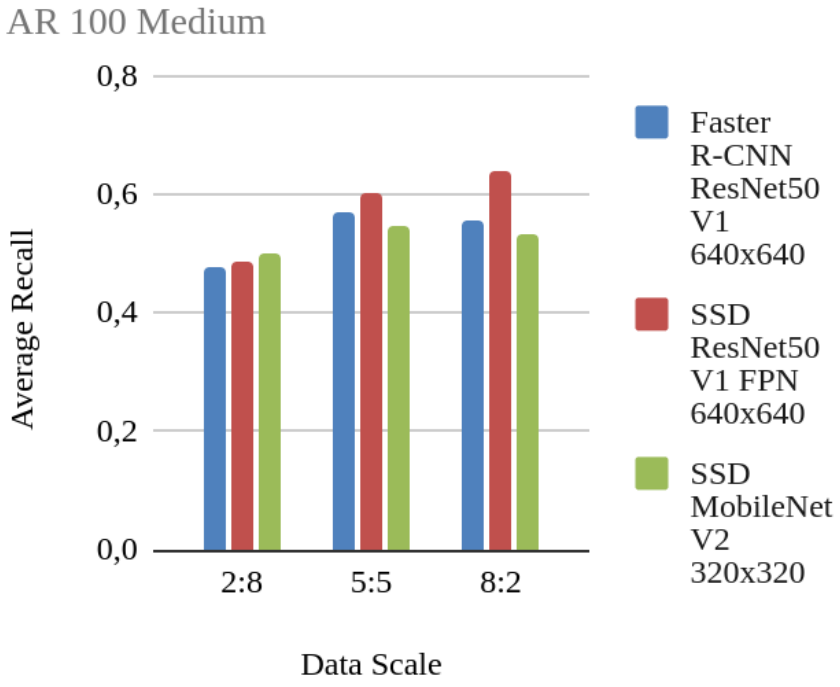

Figure 7. Experiment 1 - Average recall of medium image size detection of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

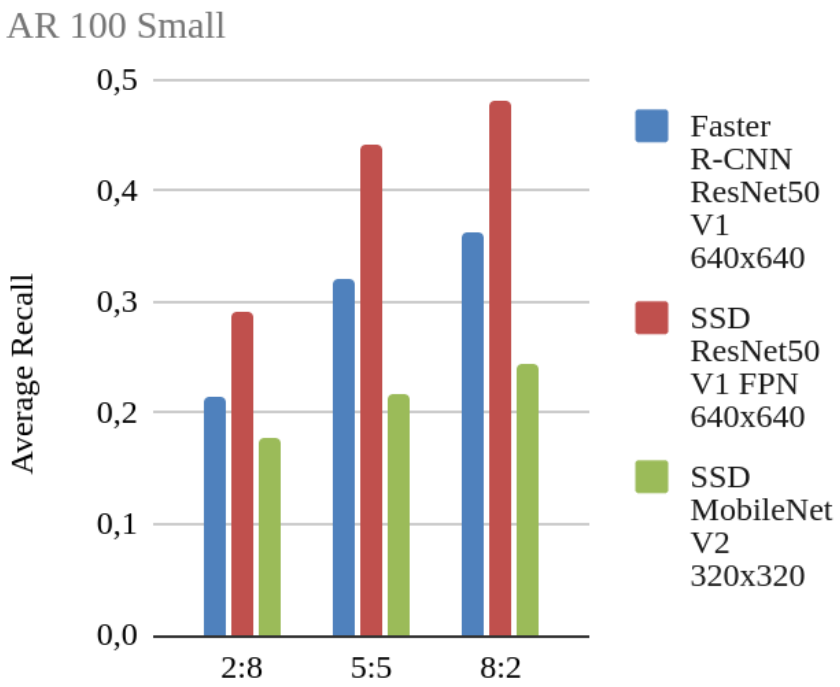

Figure 8. Experiment 1 - Average recall of small image size detection of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

Figure 8 shows the data of average recall of small images size detection from the evaluation section. From the chart above, we know that SSD ResNet50 V1 FPN architecture has the biggest average recall for all of the experiments. From this situation, overall we know that **SSD ResNet50 V1 FPN is more effective** than others in this case.

From another perspective, overall we know that more data made the system more accurate to detect small-size images. That's because data can know about various types of data (also avoid overfit).

Experiment 2 (2:2, 5:2, 8:2 data scale)

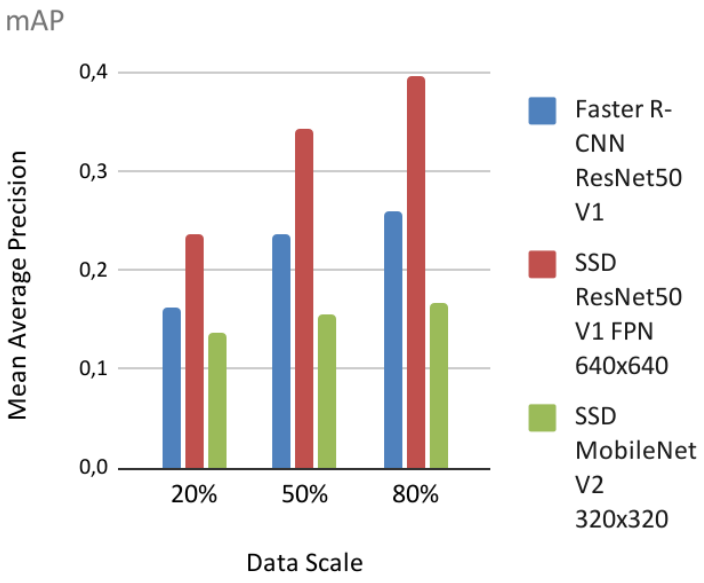

Figure 9. Experiment 2 - Mean average precision of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

Figure 9 shows the data of mean average precision from the evaluation section. From the chart above, we know that SSD ResNet50 V1 FPN architecture has the biggest mean average precision for detection. From this situation, overall we know that **SSD ResNet50 V1 FPN is more effective** than others in this case.

From another perspective, overall we know that more data made the system more accurate to detect objects. That's because data can know about various types of data (also avoid overfit).

Figure 10 shows the data of mean average precision from the evaluation section. From the chart above we know that overall SSD ResNet50 V1 FPN architecture has the biggest mean average precision for detection (20% & 80% data scale). From this situation, overall we know that **SSD ResNet50 V1 FPN is more effective** than others in this case.

From another perspective, overall we know that more data made the system more accurate to detect large images (only SSD MobileNet V2 that gave the different result). That's because data can know about various types of data (also avoid overfit).

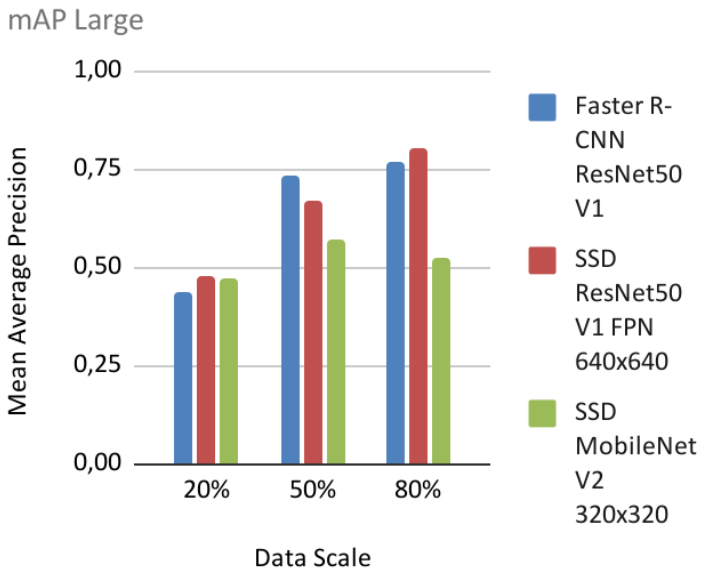

Figure 10. Experiment 2 - Mean average precision of large images size detection of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

Figure 11 shows the data of mean average precision of medium images size detection from the evaluation section. This data is the data from the evaluating section. From the chart above, we know that SSD ResNet50 V1 FPN architecture has the biggest mean average precision (specifically for medium size images) for detection in all data scales. From this situation, overall we know that **SSD ResNet50 V1 FPN is more effective** than others in this case.

From another perspective, overall we know that more data made the system more accurate to detect medium size images. That's because data can know about various types of data (also avoid overfit).

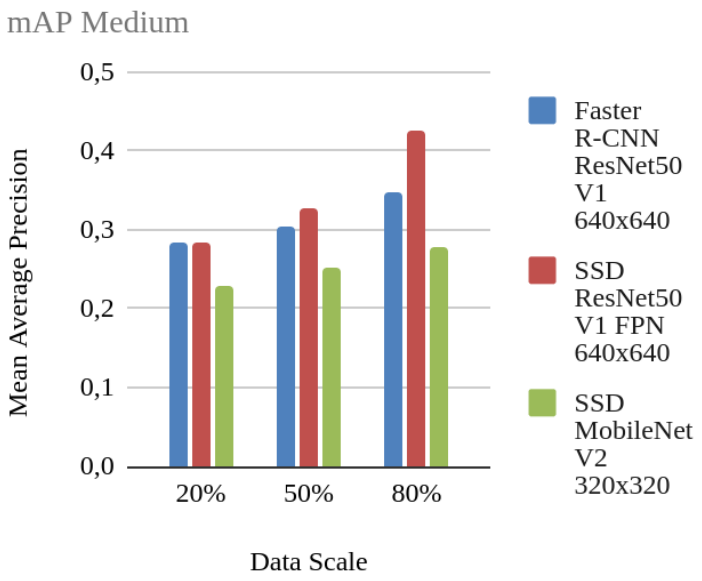

Figure 11. Experiment 2 - Mean average precision of medium images size detection of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

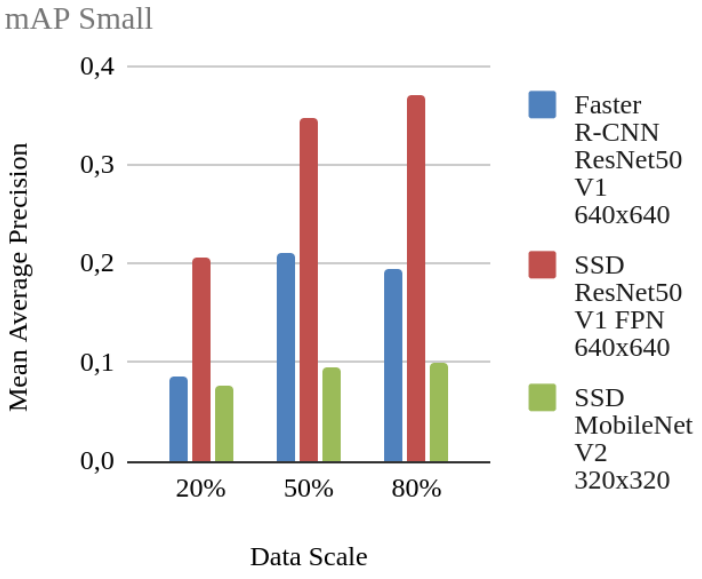

Figure 12. Experiment 2 - Mean average precision of small images size detection of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

Figure 12 shows the data of mean average precision of small images size detection from the evaluation section. From the chart above, we know that SSD ResNet50 V1 FPN architecture

has the biggest mean average precision (specifically for small size images) for detection in all data scales. From this situation, overall we know that **SSD ResNet50 V1 FPN is more effective** than others in this case.

From another perspective, overall we know that more data made the system more accurate to detect small-size images (Only Faster-RNN ResNet50 V1 that gave different results. 50% data scale better than 80% data scale). That's because data can know about various types of data (also avoid overfit).

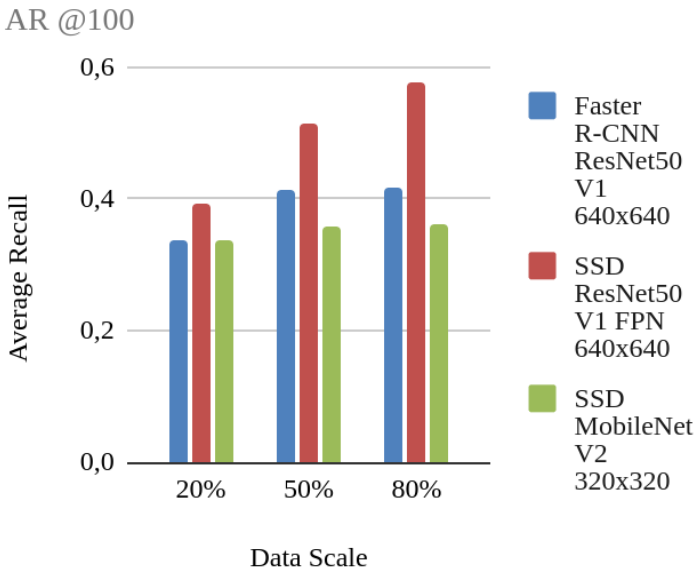

Figure 13. Experiment 2 - Average recall of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

Figure 13 shows the data of average recall from the evaluation section. From the chart above, we know that SSD ResNet50 V1 FPN architecture has the biggest average recall for detection in all data scales. From this situation, overall we know that **SSD ResNet50 V1 FPN is more effective** than others in this case.

From another perspective, overall we know that more data made the system more accurate to detect images. That's because data can know about various types of data (also avoid overfit).

Figure 14. Experiment 2 - Average recall of large image size detection of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

Figure 14 shows the data of average recall of large images size detection from the evaluation section. From the chart above, we know that the results are so random. From the chart above, we know that SSD MobileNet V2 architecture has the biggest average recall for the large images in the 2:2 data scale. Faster R-CNN ResNet 50 V1 architecture has the biggest average recall for the large images in the 2:2 data scale. SSD ResNet50 V1 FPN architecture has the biggest average recall for the large images in the 8:2 data scale. From this situation, we know that the score of the three architectures is approximately the same.

From another perspective, overall we know that more data made the system more accurate to detect images. That's because data can know about various types of data (also avoid overfit).

Figure 15 shows the data of average recall of medium images size detection from the evaluation section. From the chart above, we know that SSD ResNet50 V1 FPN architecture has the biggest average recall for detection overall. But in the 20% data scale experiment, SSD MobileNet V2 is more accurate. Overall from this situation, overall we know that **SSD ResNet50 V1 FPN is more effective** than others in this case. From another perspective, for the Faster R-CNN ResNet 50 V1 & SSD MobileNet V2, the data scale didn't do much to increase recall.

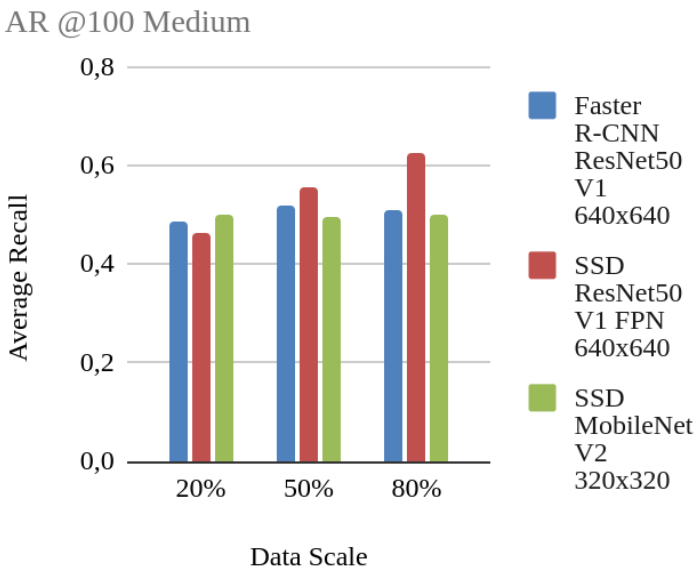

Figure 15. Experiment 2 - Average recall of medium image size detection of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

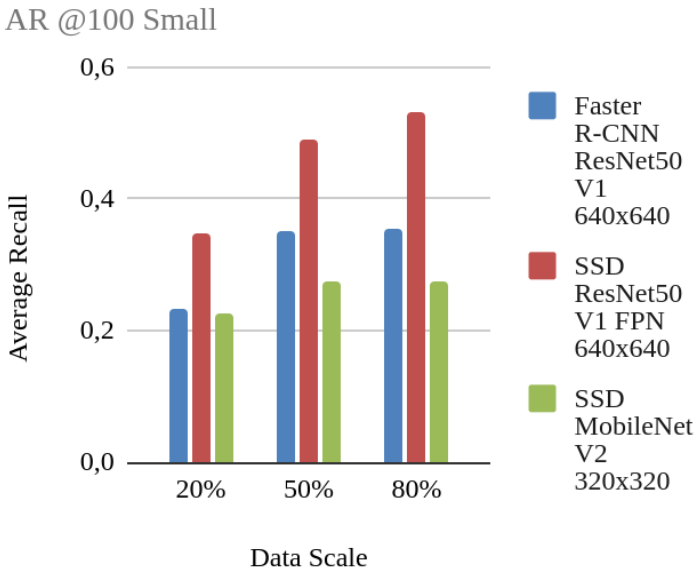

Figure 16. Experiment 2 - Average recall of small image size detection of Faster-RNN ResNet50 V1, SSD ResNet50 V1 FPN & SSD MobileNet V2 architectures

Figure 16 shows the data of average recall of small images size detection from the evaluation section. From the chart above, we know that SSD ResNet50 V1 FPN architecture has the biggest average recall for all of the experiments. From this situation, overall we know that **SSD ResNet50 V1 FPN is more effective** than others in this case.

From another perspective, overall we know that more data made the system more accurate to detect small-size images. That's because data can know about various types of data (also avoid overfit).

CONCLUSION

We can improve the performance of the "System for detection and recapitulation of health protocol violations based on computer vision technology that is integrated with websites and smartphone applications" by changing the object detection model. From the experiment, we know that SSD ResNet50 V1 FPN is the most effective model. Although the researcher did two times experiment, the results were approximately the same. In the first experiment, mean average precision, mean average precision of medium images, mean average precision of small images, average recall, average recall for large images, average recall of medium images, and an average recall of small images SSD ResNet50 V1 FPN has the better results than others. Also in the second experiment, mean average precision, mean average precision of large images, mean average precision of medium images, mean average precision of small images, average recall, average recall of medium images, and an average recall of small images SSD ResNet50 V1 FPN has the better results.

Compared to the SSD MobileNet V2 architecture, even though both use SSD architecture, it is evident that the addition of the ResNet50 V1 FPN architecture makes for better performance than the addition of MobileNet V2 architecture. Compared to the Faster R-CNN ResNet50 V1, it is evident that the addition of the ResNet50 V1 FPN architecture can make SSD architecture more accurate than Faster R-CNN architecture.

For the next research, the researcher has suggestions about the dataset. It's better if in the next research the dataset is more varied. Also better if in the next research the researcher uses more steps for the training.

REFERENCES

[1] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," *ArXiv190505055 Cs*, May 2019, Accessed: Oct. 22, 2021. [Online]. Available: http://arxiv.org/abs/1905.05055

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[3] "Google Trends," *Google Trends*. https://trends.google.co.id/trends/explore?geo=ID&q=tensorflow,pytorch (accessed Oct. 22, 2021).
[4] "TensorFlow." https://www.tensorflow.org/ (accessed Oct. 22, 2021).
[5] G. J. Chowdary, N. S. Punn, S. K. Sonbhadra, and S. Agarwal, "Face Mask Detection using Transfer Learning of InceptionV3," *ArXiv200908369 Cs Eess*, Oct. 2020, Accessed: Oct. 09, 2021. [Online]. Available: http://arxiv.org/abs/2009.08369
[6] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark Analysis of Representative Deep Neural Network Architectures," *IEEE Access*, vol. 6, pp. 64270–64277, 2018, doi: 10.1109/ACCESS.2018.2877890.
[7] O. Alsing, "Mobile Object Detection using TensorFlow Lite and Transfer Learning," Independent thesis Advanced level, KTH ROYAL INSTITUTE OF TECHNOLOGY, SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE, 2018. [Online]. Available: http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A1242627
[8] A. Campoverde and G. Barros, "Detection and Classification of Urban Actors Through TensorFlow with an Android Device," in *Information and Communication Technologies of Ecuador (TIC.EC)*, Cham, 2020, pp. 167–181. doi: 10.1007/978-3-030-35740-5_12.
[9] D. Choe, E. Choi, and D. K. Kim, "The Real-Time Mobile Application for Classifying of Endangered Parrot Species Using the CNN Models Based on Transfer Learning," *Mob. Inf. Syst.*, vol. 2020, p. e1475164, Mar. 2020, doi: 10.1155/2020/1475164.
[10] R. Deepa, E. Tamilselvan, E. S. Abrar, and S. Sampath, "Comparison of Yolo, SSD, Faster RCNN for Real Time Tennis Ball Tracking for Action Decision Networks," in *2019 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, Apr. 2019, pp. 1–4. doi: 10.1109/ICACCE46606.2019.9079965.
[11] G. Fan, F. Chen, D. Chen, Y. Li, and Y. Dong, "A Deep Learning Model for Quick and Accurate Rock Recognition with Smartphones," *Mob. Inf. Syst.*, vol. 2020, p. e7462524, May 2020, doi: 10.1155/2020/7462524.
[12] S. Ghoury, C. Sungur, and A. Durdu, "Real-Time Diseases Detection of Grape and Grape Leaves using Faster R-CNN and SSD MobileNet Architectures," Apr. 2019. [Online]. Available: https://www.researchgate.net/publication/334987612_Real-Time_Diseases_Detection_of_Grape_and_Grape_Leaves_using_Faster_R-CNN_and_SSD_MobileNet_Architectures
[13] P. D. Hung and N. N. Kien, "SSD-Mobilenet Implementation for Classifying Fish Species," in *Intelligent Computing and Optimization*, Cham, 2020, pp. 399–408. doi: 10.1007/978-3-030-33585-4_40.
[14] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, "A Novel Image Classification Approach via Dense-MobileNet Models," *Mob. Inf. Syst.*, vol. 2020, p. e7602384, Jan. 2020, doi: 10.1155/2020/7602384.
[15] Y. Zhang, C. Song, and D. Zhang, "Deep Learning-Based Object Detection Improvement for Tomato Disease," *IEEE Access*, vol. 8, pp. 56607–56614, 2020, doi: 10.1109/ACCESS.2020.2982456.
[16] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, Cham, 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.
[17] "TFRecord and tf.train.Example | TensorFlow Core," *TensorFlow*. https://www.tensorflow.org/tutorials/load_data/tfrecord (accessed Oct. 20, 2021).