|  |  |  |
| --- | --- | --- |
| **Model** | **Mobile SSD** | **Faster RCNN** |
| Test Loss  Number of epochs | 1.5  5114 | 0.1016  7167 |

### Scores

**3. EXPERIMENTAL SETUP**

**3.1 Feature Extractors**

We will be discussing a few feature extractors to get the gist of the entire architecture, where Meta architectures are combined with feature extractors like VGG, Resnet101 etc. The way it works is that we first apply convolutional neural networks via feature extractors to extract all the high level information from the images. The feature extractors must be wisely selected as types of layers, no of parameters directly affect the speed and memory performance of the object detectors. We will compare four different feature extractors to combine with our Meta architectures and test their performances. Out of the four chosen feature extractors, apart from Mobile Net, every other feature extractor has an open source Tensorflow implementation, which one can use via GitHub. We compare VGG-16, Resnet101, Inception Resnet(v2)[6] ,which combines the optimization quality of ResNet with computational efficiency of inception modules and Mobile Net which achieved accuracies similar to VGG-16 in Image net competition with 1/30 of its computational cost and model size. As the name suggests, Mobile Net because of its low computational cost has been at the forefront of various vision applications in mobile devices. Its building blocks are depth wise separable convolutions which factorize a standard convolution into a depth wise convolution and a 1x1 convolution, effectively reducing both computational cost and number of parameters. In R-FCN and R-CNN , where we must choose the layer that must be used for predicting region proposals, we have used ‘Conv5’ layer in VGG-16, and ‘Conv\_4\_x’ layers in ResNet-101, for other feature extractors we choose similar layers. In SSD, following previous methodologies, we have also selected the topmost convolutional feature map and a higher resolution feature map at a lower level, then adding a sequence of convolutional layers with spatial resolution

**Model Details**

Here, we mention the intricate details of the different combination of Meta architectures with the feature extractors.

**3.6.1 Faster R-CNN**

Tensorflow’s standard ROI pooling is used, also batch normalization is used in all convolutional layers. Optimizer used is SGD with momentum set to 0.9. The learning rate was set as per the feature extractor.

**Inception Resnet:** The stride size is 8 in atrous mode and 16 otherwise. The features are extracted from the Mixed\_6a layers including the residual layers. Features maps are 17x17 size and learning rate is 1e-3.

**MobileNet:** The initial learning rate is 3e-3 and stride size is set to 16 pixels. We extract features from Conv2D\_11 layers and features maps of size 14x14 are used.

**SSD**

Batch normalization is used in all layers and the weights are initialized with a standard deviation of 0.03. The convolutional feature maps were added and all of them used for prediction, with the convolutional layers being added with a spatial resolution, decaying by a factor of 2.

**Inception Resnet:** The initial learning rate is set to 0.005, with a decaying factor of 0.8 after every 700k steps. The activation function ReLU is used. Mixed\_6a and Conv2d\_7b are used by appending with additional convolutional layers of depth 512,256,128 respectively.

**MobileNet:** The initial learning rate is set to 0.004 and we have used conv\_11 and conv\_13 layers with four additional layers with decaying spatial resolution with depths of 512,256,256 ,128.The activation function ReLU is used .

**Analysis**

It is noted that usually SSD model is faster than its counterpart; also Faster R-CNN leads to much slower times, taking more than 100ms per image and giving the most accurate models of SSD. It is also found that the intuition of feature extractors’ accuracy correlating with the mAP scores on the inn-house dataset seems to be true only for Faster R-CNN as SSD models as less reliant on the classification accuracy of their feature extractors. The number of region proposals for Faster R-CNN can be reduced without affecting the mAP score by much, in turn saving lots of computation. We found that, in Faster R-CNN Inception Resnet, the mAP score with 300 proposals was 34.2, and it gave a mAP score of 28.7 with just 10 proposals, although the sweet spot is when we used 50 proposals as we are able to achieve more than 93% accuracy of those trained with 300 proposals by saving 3 times the running time. As far as memory utilization is considered, Faster R-CNN inception Resnet consumes the most memory whereas SSD MobileNet consumes the least, unsurprisingly; it is correlating with the speed of these models. The models performed much better on bigger images with more

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Combination** | **mAP score** |  | **CPU time** |
|  |  |  | 40 |
| SSD MobileNet | 19 |  | |
| SSD Inception  Resnet | 20.3 | 80 | |
| Faster R-CNN  MobileNet | 19 | 118 | |
|  |  |  |  |
| Faster R-CNN Inception Resnet | 34.2 |  | 860 |

**5. CONCLUSION**

We have put forward a comparative study of state of the art object detectors which use convolutional neural networks. We have addressed their issues and performance on a common hardware and also tested different combinations of them, all on the in-house dataset. We discovered that SSD performs much better with light weight feature extractors on bigger images, competing with the most accurate of models. We also found that fewer proposals increase speed without compromising much on the mAP scores. We wish to try different combinations and find better results and sweet spots which can be applied to specific use cases.

**REFERENCES**

[1]Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016,October. Ssd: Single shot multibox detector. At European conference on computer vision (pp.

21-37). Springer, Cham.

[2]Dai, J., Li, Y., He, K. and Sun, J., 2016. R-fcn:Object detection via region-based fully convolutional networks. At Advances in neural information processing systems (pp. 379-387).

[3]Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99)

[4]Girshick, R., 2015. Fast r-cnn. At Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[5]Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. and Ghemawat, S., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

[6]N.Silberman and S.Guadarrama .Tf-slim: A high level library tzo define complex models in tensorflow. https://research.googleblog.com/2016/08/tf- slim-high-level-library-to-define.html,2016.

[7]T. Y. Lin and P. Dollar. Ms coco api. https://github.com/pdollar/coco, 2016.

[8]Ren, S., He, K., Girshick, R., Zhang, X. and Sun, J., 2017. Object detection networks on convolutional feature maps. IEEE transactions on pattern analysis and machine intelligence,39(7), pp.1476-1481.

[9]J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection.arXiv preprint arXiv:1506.02640, 2015.

[10]A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang,T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861,2017

[11] Nikhil Yadav , Utkarsh Binay. Comparative Study of Object Detection Algorithms.

International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 11 | Nov -2017