Cadla Services India Private Limited

BenchMarks and Suggestions on Models

Dataset used : Pulses(black eyed beans, chana dal, kidney beans, masoor dal, moong dal, toor dal, urad dal)

1)

EfficientNetsB0 : Trained the last dense layer. Use the library EfficientNet

BenchMarks : Training accuracy - 96%

: Validation accuracy - 92%

Dataset trained : Pulses(black eyed beans, chana dal, kidney beans, masoor dal, moong dal, toor dal, urad dal)

Comments : The model is really good.It is working very accurately as validation accuracy is quite good. In future this architecture can be considered as it's a new model so chances of getting updates on it is quite high.

Current Situation : Since it is a new model , so tensorflow lite doesn't support all its operations. therefore right now we can't convert it into tensorflow lite. In near future after tensorflow gets more updates, we might be able to convert it into tensorflow lite form.

Fine tuning the above model(EfficientNetsB0)

: Retrained the model from the layer name "multiply_16"

BenchMarks : Training accuracy - 97%

: Validation accuracy - 93%

General Remarks on EfficientsNets : This Nets has 8 variants starting from B0 to B7. Weights from B0 to B3 has been made public for transfer-learning.Other Weights will be made public in the near future.

Reference : https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

: https://pypi.org/project/efficientnet/

: https://www.dlology.com/blog/transfer-learning-with-efficientnet/ (For fine tuning)

2)

ResNet101 : Library is not present in the keras. Source code can be found in the Github but it contains some bugs. So in the near future it is expected to be included in keras in its next version. The model statistics is good and is expected to behave good, So we can consider it.

Comments : The models is a little heavy. So it's latency time in Raspberry Pi might be a little more.

Reference : https://keras.io/applications/#resnet

3)

MobileNetV2 : Trained the last dense layers

BenchMarks : Training accuracy - 90%

: Validation accuracy - 74%

Comments : The model is good as it is lightweight.The validation accuracy is fair enough but not an appreciable one.

Fine tuned the model for the full layers

BenchMarks : Training accuracy - 94%

: Validation accuracy - 77%

Remarks : Currently we are unable to convert the model to tflite form because one operation "FusedBatchNormV3" on the model is till now supported by tflite.

Comments :The model is light-weight. It is expected that the next version of tensorflow will contain the lite which will support the above operation.

4)

NASNets : Trained the last dense layers

BenchMarks : Training accuracy - 94%

: Validation accuracy - 83%

Fine tuned the model for the full layers

BenchMarks : Training accuracy - 94.5%

: Validation accuracy - 92.5%

Remarks : Currently we are unable to convert the model to tflite form because one operation "FusedBatchNormV3" on the model is till now supported by tflite.

Comments : It is expected that the next version of tensorflow will contain the lite which will support the above operation.

5)

SqueezeNets : Trained for the full layers

BenchMarks :Training accuracy - 95%

: Validation accuracy - 90%

Remarks : It's an old network so some library got outdated. So it is recommended to use the source code instead of the library. I have also provided the source code.

Comments : The model is very good when we compare its size vs performance. The model is not big. Its tflite is not big , it's only 5mb. So further work can be on this model to improve its accuracy.

6)

ShuffleNets : Trained the last dense layers

BenchMarks : Training accuracy - 90%

: Validation accuracy - 16%

Fine tuned the model for the full layers

BenchMarks : Training accuracy - 91%

: Validation accuracy - 20%

Remarks : The model doesn't seem to be behaving good for the dataset. Further work might improve the validation but till now it's not worthwhile.