Experiment

Dataset Creation: First, I cleaned unnecessary images from the provided dataset. I found 2 image formats in the dataset, png and jpeg. I changed all the image format into jpg. Then I used makesense.ai to create dataset in YOLO format for the given image dataset. Also, I spilt the dataset 80% for training and 20% for testing. Makesense.ai is a webapp which provides a free image annotation tool.

Model Selection: Transfer Learning is a machine learning approach where a pretrained model can be trained again with a different dataset. For this object detection task, the Transfer Learning method is applicable. A pretrained YOLO v3 is chosen for this task. I chose the darknet 53 model to train the pretrained weight.

Observations

1st phase Training and Testing:

First, I trained the model for 1000 epochs with the following parameters:

learning_rate = 0.001

decay = 0.0005

batch size = 64

momentum = 0.9

Test result metrics:

ap = 96.16% (**TP** = 32, **FP** = 1)

mean average precision (mAP) = 0.961597

precision = 0.97, **recall** = 0.86, **F1-score** = 0.91

Second, I trained the model for 2000 epochs with similar parameters:

learning_rate =0.001

decay =0.0005

batch size = 64

activation =leaky relu

momentum =0.9

Test result metrics:

ap = 91.50% (**TP** = 33, **FP** = 1)

mean average precision (mAP) = 0.915033

precision = 0.97, **recall** = 0.89, **F1-score** = 0.93

Total 2000 iterations took 4 hours to execute with the help of colab GPU.

2nd phase Training and Testing:

Like 1st phase training, First I trained the model for 1000 epochswith the following parameters:

learning_rate = 0.0001

decay = 0.0001

batch size = 64

momentum = 0.5

Test result metrics:

ap = 49.21% (**TP** = 0, **FP** = 0)

mean average precision (mAP) = 0.492075,

precision = -nan, **recall** = 0.00, **F1-score** = -nan

Second, I trained the model for 2000 epochs with similar parameters:

learning_rate = 0.0001

decay = 0.0001

batch size = 64

momentum = 0.5

Test result metrics:

ap = 57.76% (**TP** = 18, **FP** = 8)

mean average precision (mAP) = 0.577585

precision = 0.69, **recall** = 0.49, **F1-score** = 0.57

Due to colab GPU usage policy, I can not train again.

So, I my observation is within these 2-phase training process, **learning_rate** = 0.001, **decay** = 0.0005, **momentum** = 0.9

and 2000 epoch, the model performs best. Because the metrics like precision, recall and f1-score is higher.

I also Build a API for koala detection. The trained model can be found in this link:

https://drive.google.com/file/d/1ZKLG1QFjwIl4h94vc7fcUbQaZn0Q6LuB/view?usp=sharing