## 

## ITI107 Deep Learning Networks

## Assignment Report

**Done by:**

**Yap Jing Yang 21B583W**

**Ng Wei Xiang 21B576M**

#### **Table of Content**

* **Introduction**
* **Data Collection & annotation**
* **Selecting the Pre-trained Models**
* **Using pre-trained model: SSD MobileNet v2 320x320 (Setting up, training, results and testing)**
* **Using pre-trained model: SSD ResNet50 V1 FPN 640x640 (Setting up, training, results and testing)**
* **Using pre-trained model: EfficientDet D0 512x512**  **(Setting up, training, results and testing)**
* **Conclusion**
* **Individual Contributions & Learning Journey**

## Introduction

For this assignment, my partner is Ng Wei Xiang, 21B576M. We chose the following five object classes to work on:

Class 1: Lion

Class 2: Elephant

Class 3: Leopard

Class 4: Buffalo

Class 5: Rhino

For the pre-trained models, we decided on the following models:

1. SSD MobileNet v2 320x320 (used by Yap Jing Yang)
2. SSD ResNet50 V1 FPN 640x640 (used by Yap Jing Yang)
3. EfficientDet D0 512x512 (used by Ng Wei Xiang)

All details pertaining to this project can be found on our project repository on GITHUB. (https://github.com/jingyang022/ITI107\_Assignment)

## Data Collection & Annotation

The dataset can be downloaded from our project repository. (<https://github.com/jingyang022/ITI107_Assignment/tree/main/datasets>)

The dataset consists of 1000 images and 1000 annotations in PASCALVOC format.

The 1000 images consist of:

1. 200 lion images
2. 200 elephant images
3. 200 leopard images
4. 200 buffalo images
5. 200 rhino images

The buffalo, elephant and rhino images are taken from the Kaggle African Wildlife dataset. (<https://www.kaggle.com/biancaferreira/african-wildlife>)

The lion and leopard images are downloaded from the Internet via Web scraping tools.

The annotation of the images are done using the Roboflow annotation tool.

We also uploaded a sample picture and video featuring all five animals to test the model.

## Selecting the Pre-trained Models

For the pre-trained models, we decided on the following models:

1. SSD MobileNet v2 320x320 (used by Yap Jing Yang)
2. SSD ResNet50 V1 FPN 640x640 (used by Yap Jing Yang)
3. EfficientDet D0 512x512 (used by Ng Wei Xiang)

Details of using the individual models will be covered later by the respective trainers. We will also discuss our training process and experimental results, as well as screenshots of mAP, loss and evaluation images from Tensorboard.

The pre-trained models can be downloaded from the following link: (<https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md>)

## Using pre-trained model: SSD MobileNet v2 320x320

## (Covered by Yap Jing Yang)

**Setting up & Training the Model**

For the training of the model, I followed the guide as per the lab session number 4 covered by Mr Mar. The link can be found here: (<https://github.com/nyp-sit/iti107/blob/main/session-4/custom_training_with_tfod_api.md>).

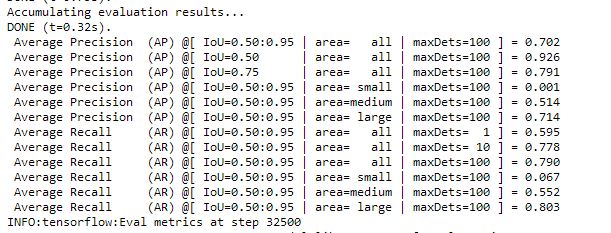
I configured the pipeline.config file as per my needs. I chose a batch size of 10 and use a learning rate of 0.04. I trained the model for about two hours and stopped training when the validation loss has more or less plateaued out.

**Results of the training**

I had tabulated the results in a table and provided screenshots of the mAP, loss and evaluation images from Tensorboard.

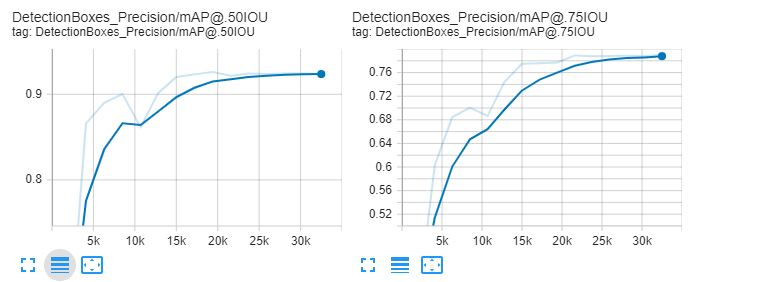
|  |  |  |  |
| --- | --- | --- | --- |
| **mAP@0.5IOU** | **mAP@0.75IOU** | **mAP@small objects** | **mAP@large objects** |
| 0.926 | 0.791 | 0 | 0.714 |

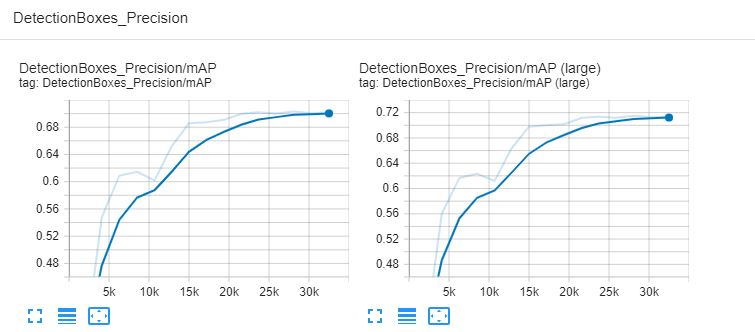
These values are extracted from the screenshot of the evaluation results.

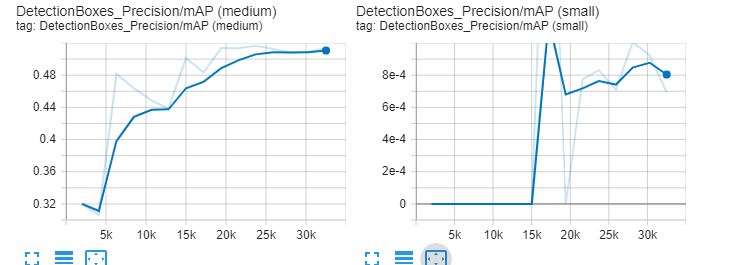


My model mAP@0.5 IOU seems quite high at 0.926 and is decent of about 0.8 for mAP@0.75IOU.

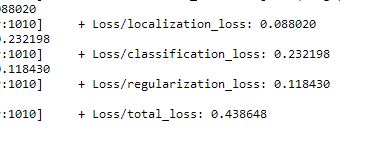
However, my model mAP on small objects is zero and 0.714 for large objects. The reason might be due to the fact that the object class in the images is big and there are probably no small objects inside the picture.

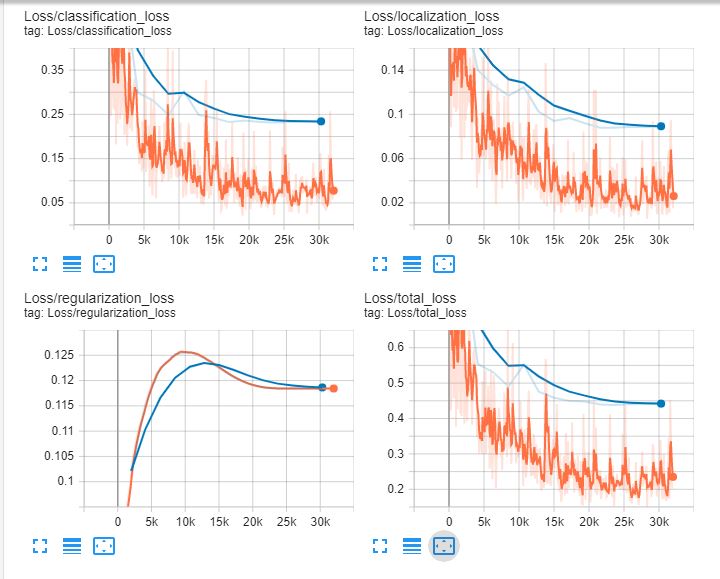




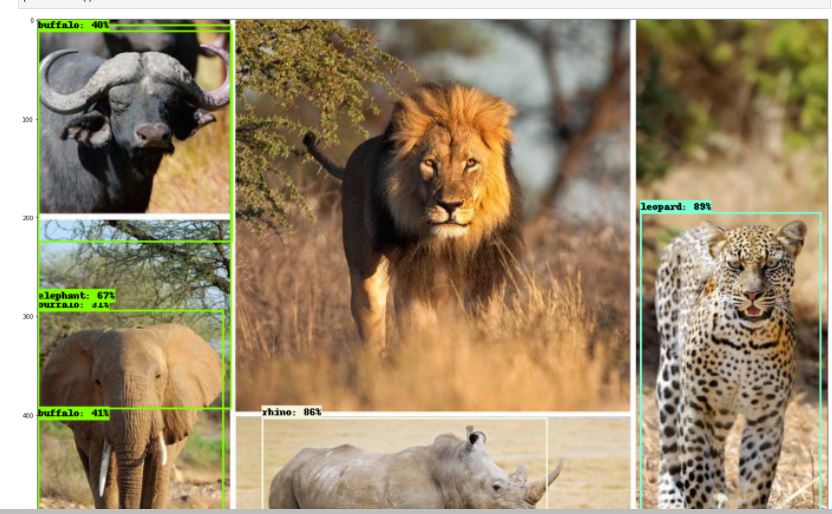


As for the loss, the total loss is 0.438648 which seems decent.





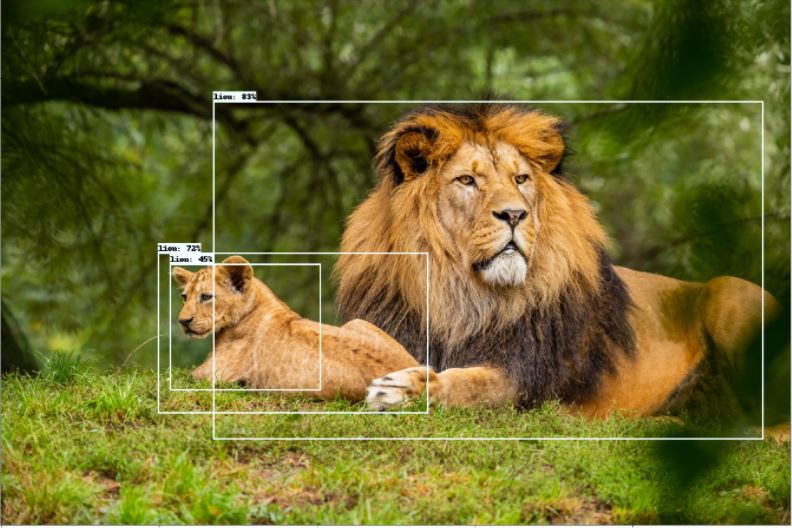
**Testing the model on the test image and video clip**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Buffalo** | **Elephant** | **Lion** | **Rhino** | **Leopard** |
| 40% | 67% | No detection | 86% | 89% |

The results are not so good. The model is unable to detect the lion, it could due to the colour of the lion coincides closely with the background. In the next picture, the model is able to detect the lions as the background is mainly green in colour.

For the video, the results are even worse. Although the model is able to detect the animals in the video, it has classified the animals wrongly.



## Using pre-trained model: SSD ResNet50 V1 FPN 640x640

## (Covered by Yap Jing Yang)

**Setting up & Training the Model**

For the training of the model, I followed the guide as per the lab session number 4 covered by Mr Mar. The link can be found here: (<https://github.com/nyp-sit/iti107/blob/main/session-4/custom_training_with_tfod_api.md>).

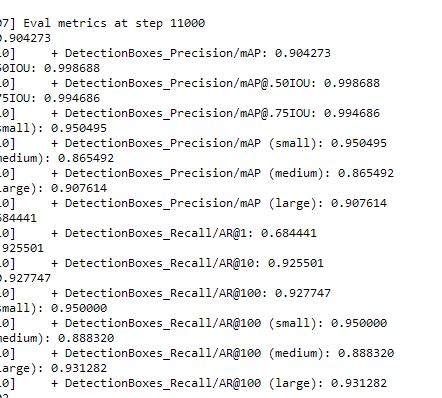
I configured the pipeline.config file as per my needs. I chose a batch size of 10 and use a learning rate of 0.04. I trained the model for about five hours and stopped training when the mean mAP reached a pretty high value.

**Results of the training**

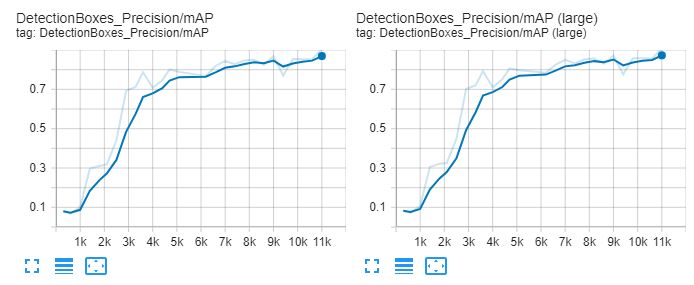
I had tabulated the results in a table and provided screenshots of the mAP, loss and evaluation images from Tensorboard.

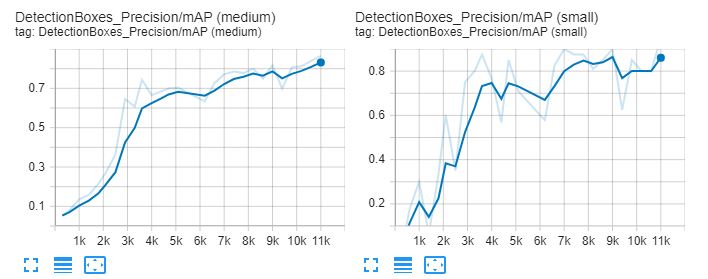
|  |  |  |  |
| --- | --- | --- | --- |
| **mAP@0.5IOU** | **mAP@0.75IOU** | **mAP@small objects** | **mAP@large objects** |
| 0.998 | 0.994 | 0.95 | 0.907 |

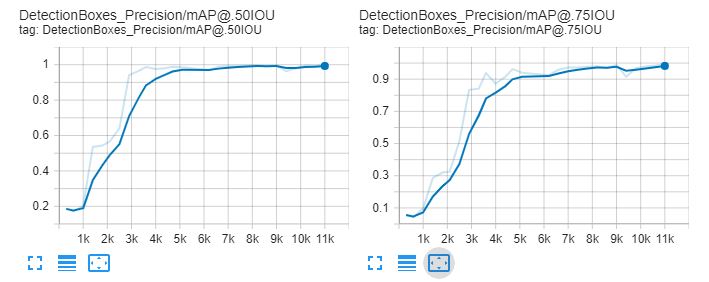
These values are extracted from the screenshot of the evaluation results.



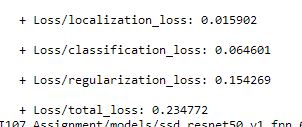
This model is better than the SSD MobileNet v2 320x320 model as mAP values are higher and it is able to detect small objects.

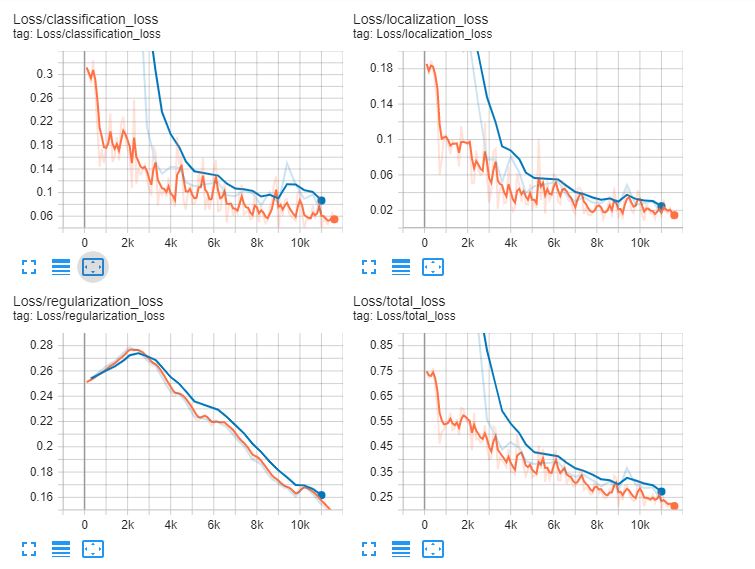






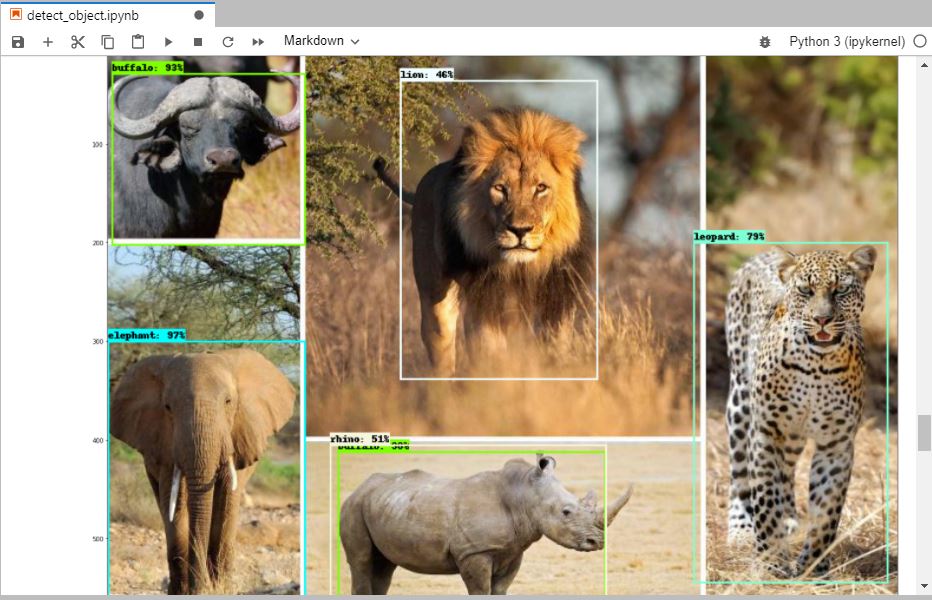
As for the loss, the total loss is 0.234772 which seems decent.





This model does not seems to be overfitting as the curves almost close to converge at the end of the training session.

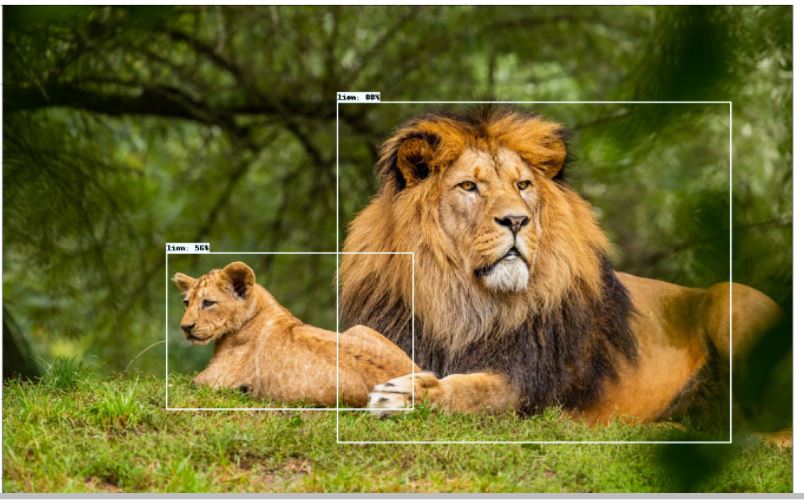
**Testing the model on the test image and video clip**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Buffalo** | **Elephant** | **Lion** | **Rhino** | **Leopard** |
| 93% | 97% | 46% | 51% | 79% |

As compared to the SSD MobileNet v2 320x320 model, this model has a much better performance as it can detect and classify all five animals correctly. Still this model cannot detect the lion very well. In the next picture, the model is able to detect the lions better as the background is mainly green in colour.

For the video, the results are the same as the SSD MobileNet v2 320x320 model. Although the model is able to detect the animals in the video, it has classified the animals wrongly.



## Using pre-trained model: EfficientDet D0 512x512

## (Covered by Ng Wei Xiang)

(To be filled by Wei Xiang)

## Conclusion

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Mean mAP** | **mAP@**  **0.5IOU** | **mAP@**  **0.75IOU** | **mAP@**  **small objects** | **mAP@**  **large objects** |
| EfficientDet D0 512x512 |  |  |  |  |  |
| SSD ResNet50 V1 FPN 640x640 | 0.904 | 0.998 | 0.994 | 0.95 | 0.907 |
| SSD MobileNet v2 320x320 | 0.702 | 0.926 | 0.791 | 0 | 0.714 |

discussion

## My Contributions & Learning Journey (Yap Jing Yang)

(To be filled by JY)

## My Contributions & Learning Journey (Ng Wei Xiang)

(To be filled by Wei Xiang)