**Training a Custom Object Detection Model**

Name: Goh Dai Yong Adison

**Introduction**

This project aims to create a custom object detection model by leveraging on transfer learning from the pre-trained **CenterNet Resnet50 V1 FPN 512x512** model. The pre-trained model comprises of the ResNet-50 architecture as the feature extractor while applying the feature pyramid network (FPN) to extract features from multiple scales, and uses the CenterNet head as the final layer to generate the center point heatmap for object detection.

The custom model will be fine-tuned by training on a task-specific dataset, and aims to detect two classes: **“person”** (which is also an existing class from the COCO dataset) and **“lightsaber”**. The lightsaber is a laser sword used by characters in the Star Wars movie series.

**Data Collection & Annotation**

A total of 500 images were sourced based on image searches on Google and Bing. 250 images were used for training initially in the first two runs, while another 250 images were added for the last two runs (total four runs). The Bulk Image Downloader extension on Google Chrome was used to screen, identify and download relevant images from these searches. The bulk of the images (about 80%) generally depict pictures of Star Wars characters wielding a lightsaber (ie. both classes present in the same image), while about 15% of images consists of lightsaber(s) only and 5% with Star Wars character only. As the model was never trained specifically on lightsabers, a greater number of images depicting lightsabers were included in the dataset for training. Once the images were downloaded, the LabelImg package was used to annotate the respective class bounding boxes in each image and then saved as individual XML files.

The link to the images and annotations used for fine-tuning were uploaded in Google Drive ([Link](https://drive.google.com/drive/folders/1zdeC46f_ytv_74d5MDiWL0MvLF3dSCXc?usp=sharing)).

**Training & Evaluation**

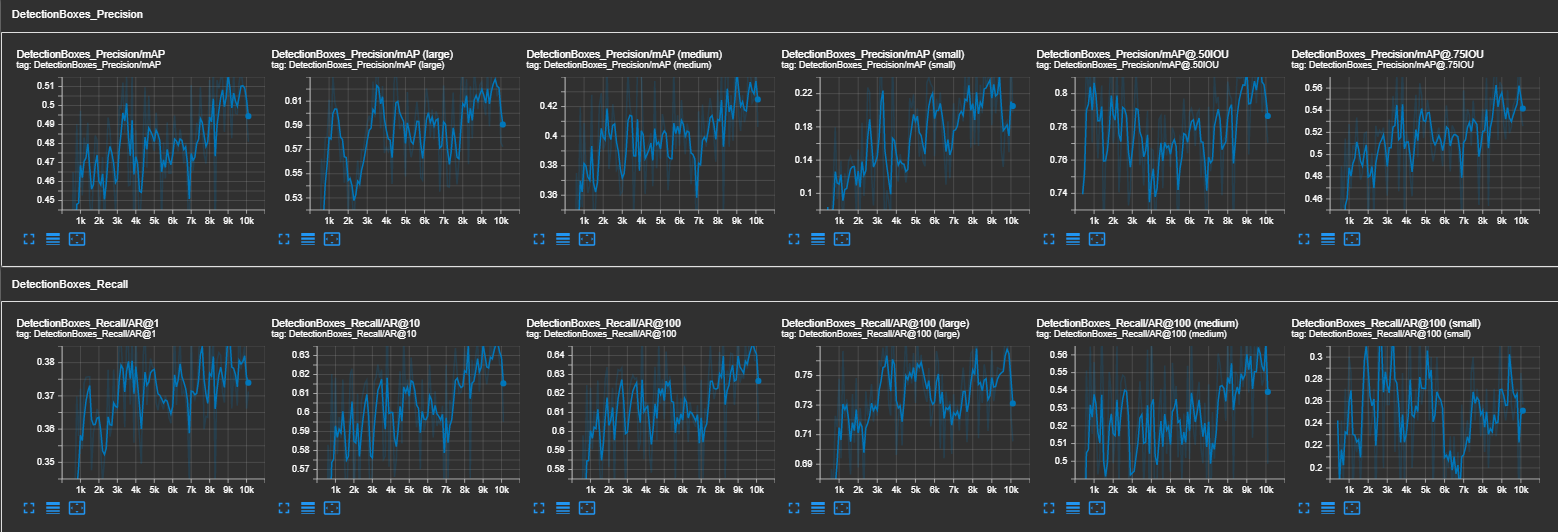
Four training runs were performed with different pipeline configurations.

Run 1

*Parameters changed:*

* num\_classes: 2
  + This is based on the number of classes – “person” & “lightsaber”
* batch\_size: 10 (default: 128)
  + Tried batch\_size = 32 initially but ran into out-of-memory error. Low batch\_size allows training to proceed with limited GPU memory resource, however this may lead to the model taking longer to converge.)
* total\_steps: 25000 (default: 250000)
  + As the dataset is small with 250 images for run1, fewer steps are likely required for training the model. A large number of steps will also increase risk of overfitting.
* Adam optimiser
  + learning\_rate\_base: 0.001 (default, no change)
  + warmup\_learning\_rate: 0.00025 (default, no change)
  + warmup\_steps: 2500 (default: 5000)
  + As the Adam optimiser is popular and generally works well for image processing, no change is made to the type of optimiser.

*Results of Run 1:*





About 10000 steps were performed in Run 1 before it was terminated. Training was ended early as the mAP was decreasing and total evaluation loss was increasing.

The mean average precision (mAP) was highest at about step 9000 at approximately 0.515, which was based on minimum box overlap intersection-over-union (IOU) of 0.7. At step 9000, mAP at 0.5 IOU was above 0.81. This mAP will be used as a benchmark to be improved on in subsequent runs.

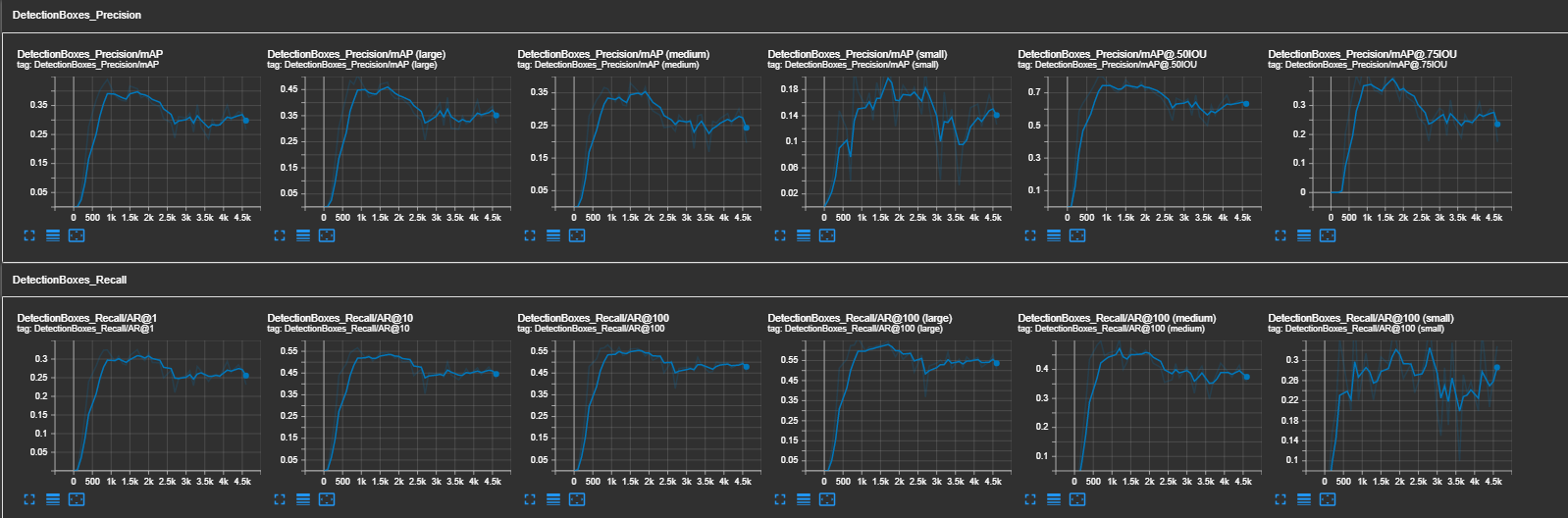
The mAP and loss tended to fluctuate quite significantly, this may be contributed by a learning rate that is too high or increasing too quickly. In addition, the total training loss (orange line) continues to decrease but the total evaluation loss (blue line) remained somewhat constant after step 2000 (fluctuating between 3 and 4). This suggests that the model may be overfitting, possibly due to a small dataset of 250 images and that the number of steps in Run 1 may be too much.

Run 2

*Parameters changed:*

* Adding more data\_augmentation\_options
  + Applied: random\_vertical\_flip and random\_rotation90
  + Existing: random\_horizontal\_flip, random\_crop\_image, random\_adjust\_hue, random\_adjust\_contrast, random\_adjust\_saturation, random\_adjust\_brightness, random\_absolute\_pad\_image
  + Image augmentation was used to randomly apply transformations to input images. The existing data augmentations options were all relevant to the dataset, as ‘person’ and ‘lightsaber’ can be found in varying positions, sizes and colours. Additional augmentation options including vertical flipping and rotation were introduced as a ‘person’ or ‘lightsaber’ could also be found in various horizontal and vertical positions, for example in battle scenes. This provides the existing small dataset with a larger variance, a larger variety of images allows the model to generalise and better detect objects.
* Increase in base learning rate
  + learning\_rate\_base: 0.005 (increased from 0.001)
  + warmup\_learning\_rate: 0.0001 (decreased from 0.00025)
  + warm\_steps: 3000 (increased from 2500)
  + The initial warmup learning rate was decreased to first allow the model to make small changes in the weights, this is to prevent higher losses at the start from drastically changing the original model’s weights. The gradual increase over more steps to a higher base learning rate is intended to prevent the model from being stuck at local minimums and to also prevent large fluctuations in the loss or mAP.

*Results of Run 2:*





Run 2 was terminated early at about 4700 steps as the mAP was decreasing and total evaluation loss increasing. The mAP was highest and total evaluation loss the lowest between steps 1000 to 2000. However, the key metric mAP at 0.5 IOU failed to perform better in Run 2 (0.72) compared to Run 1 (0.81). The mAP for Run 2 was also much poorer, going up to 0.39. The mAP and evaluation loss started becoming poorer as the learning rate increased, especially at about step 2700 where the learning rate is nearing the base learning rate of 0.005. This suggests that the learning rate may be high or increased too quickly, resulting in instability in the training.

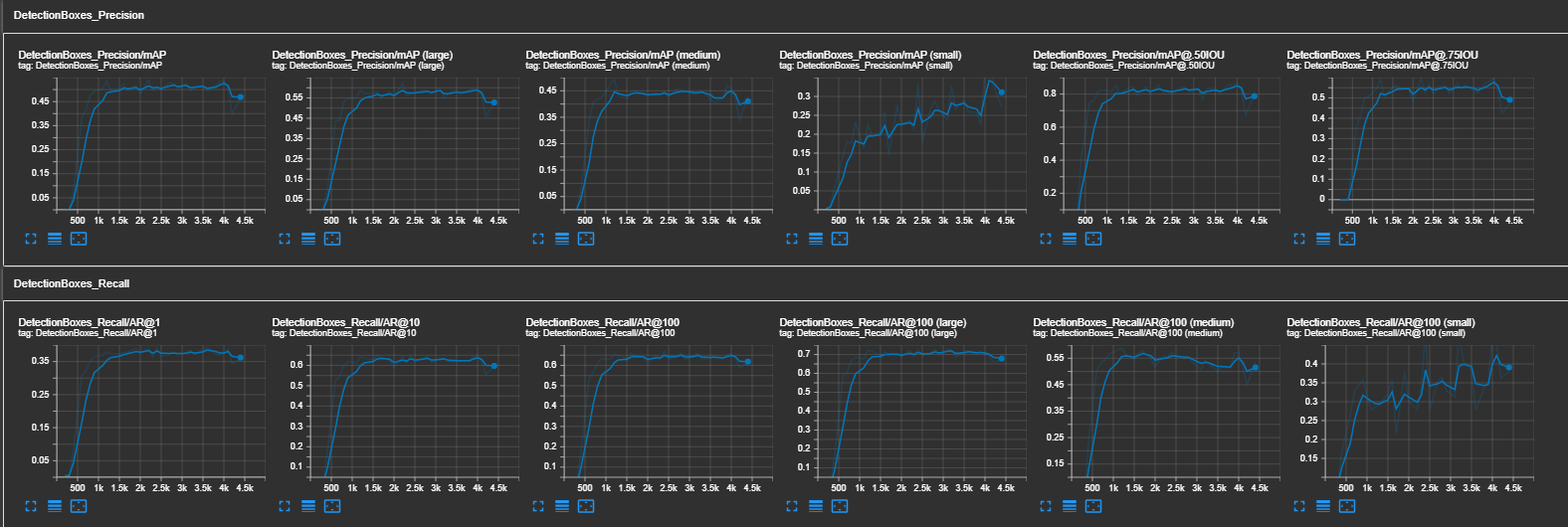
At this point, it is not clear if the new data augmentation techniques of random vertical flip and rotation may have contributed to the poorer results. But it is unlikely as these augmentation techniques are relevant to the dataset. The training and evaluation also started to deviate more after 1000 steps, suggesting that there could be some degree of overfitting possibly due to the small training sample size of 250. To address this, more image samples will also be added to the dataset in subsequent runs.

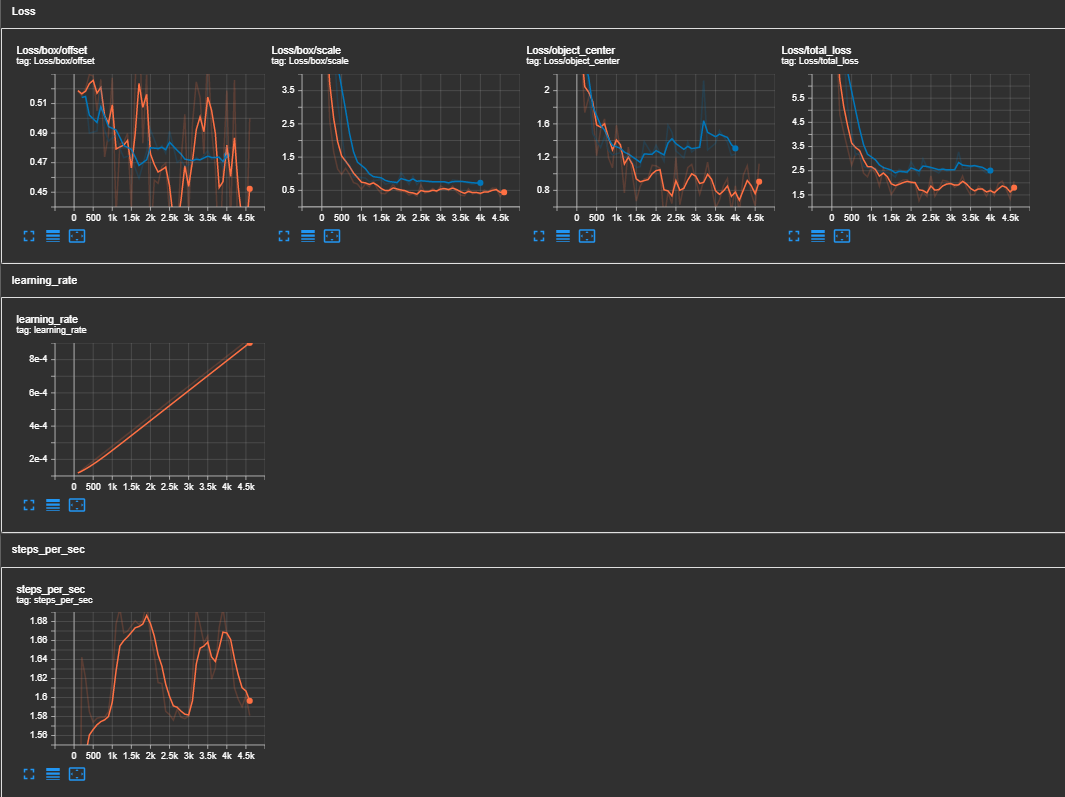
Run 3

*Parameters changed:*

* Additional 250 training images & annotations were added to the dataset, with a current total of 500.
* Reduce learning rate
  + learning\_rate\_base: 0.001 (reduced from 0.005, same as run 1)
  + warmup\_learning\_rate: 0.0001 (same as run 2)
  + warmup\_steps: 5000 (increased from 3000)
  + The base learning rate is reduced and warmup steps increased to allow the learning rate to increase slowly over a longer period of time. This is to prevent instability in training as seen in the previous runs, while allowing the model to converge towards the minimum. However, this may also result in slower convergence but this is a lesser issue.

*Results of Run 3:*





Run 3 was terminated earlier after about 4500 steps, as the mAP and total evaluation loss has plateaued and did not change much after step 1500. The training and evaluation loss was similar and less overfitting was also seen in this run, this is likely contributed by the increased size of the training dataset. However, running more steps will likely increase the risk of overfitting and hence stopping at about 4500 steps was appropriate.

The mAP at 0.5 IOU was highest at about step 4000 with a value of 0.8669, which was a 5-point improvement from run 1. This was likely contributed by the larger sample size, which allowed the model to learn from more examples and therefore to better generalise and detect persons and lightsabers. In addition, the lower and slow increase in the learning rate also contributed to less fluctuations and more stability in the total loss and mAP.

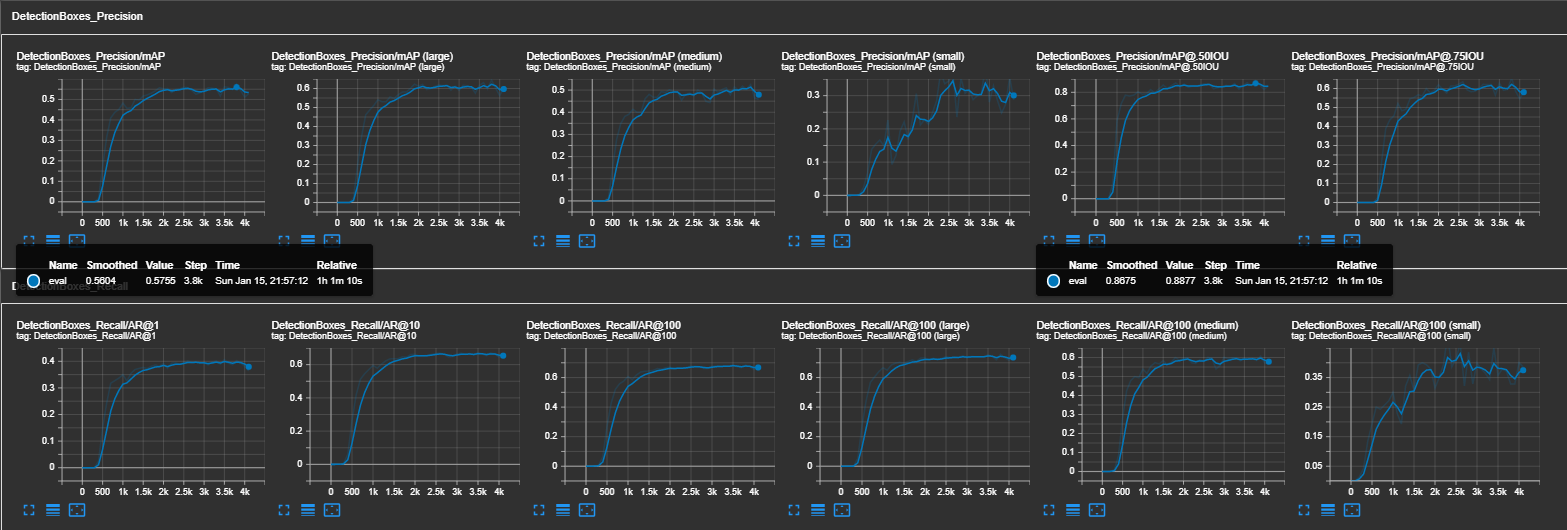
Run 4

*Parameters changed:*

* Increased batch size from 10 to 16
* Reduce learning rate and rate of increase
  + warmup\_learning\_rate: 0.00005 (reduced from 0.0001)
  + warmup\_steps: 7500 (reduced from 5000)
  + The starting learning rate was reduced and warmup steps increased in order to slow down the starting learning rate of the model. This is to allow minor corrections to the model weights at the start, especially when the loss is greater, in order to prevent large changes to the model weights during finetuning. This is important as the model was already pretrained on the COCO dataset and small finetuning of weights reduces the risk of the model forgetting any useful information that was learnt. While reducing the learning rate may increase time to convergence, this will be balanced out by increasing the batch size.

*Results of Run 4:*

Run 4 was terminated earlier after approximately 4300 steps. This was done as the mAP and loss have plateaued and remained steady after about 2000 steps, and the training and evaluation loss started to increase and diverge more after 4000 steps.





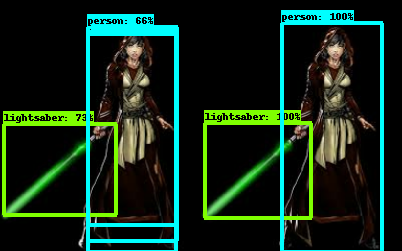
The results for Run 4 also improved compared to Run 3. The model performed best at step 3800, where the mAP at IOU 0.5 was 0.8877 (compared to 0.8669 in Run 3) and mAP 0.5755 (compared to 0.5365 in Run 3). This results show that the model is able to correctly identify and locate objects in an image at a fairly good accuracy of 88.77%. This improvement from run 3 was likely contributed by the lower initial and slow increase in the learning rate, which allowed the model to converge towards the minimum point. However, whether this minimum point was the global minimum or a local minimum could be not decisively determined, as the learning rate is still fairly low upon when the training was stopped. Nevertheless, the previous runs where learning rate was higher displayed some instability in learning and did not perform better or much poorer (as with run 2). Hence, this run suggests that a lower initial learning rate is useful and better for training.

Future Work

At this point, I was about to reach the provisioned AWS GPU resource time limit.

Other options for hyperparameter tuning I would have tried included using different optimisers such as stochastic gradient descent (SGD), root mean square propagation (RMSprop) and adaptive gradient algorithm (Adagrad). In these runs, I decided to focus on the Adam optimiser as it generally works well for object detection tasks and is more advanced compared to optimisers like SGD.

In addition, I will also introduce non-maximum suppression (NMS) to address duplicate predictions. In viewing the evaluated images in Tensorboard, the model had duplicate predictions in some of the images, such as the one below where there were duplicate ‘person’ classes. A IOU threshold of about 0.7 for the NMS algorithm will allow these duplicate predictions to be identified and removed.



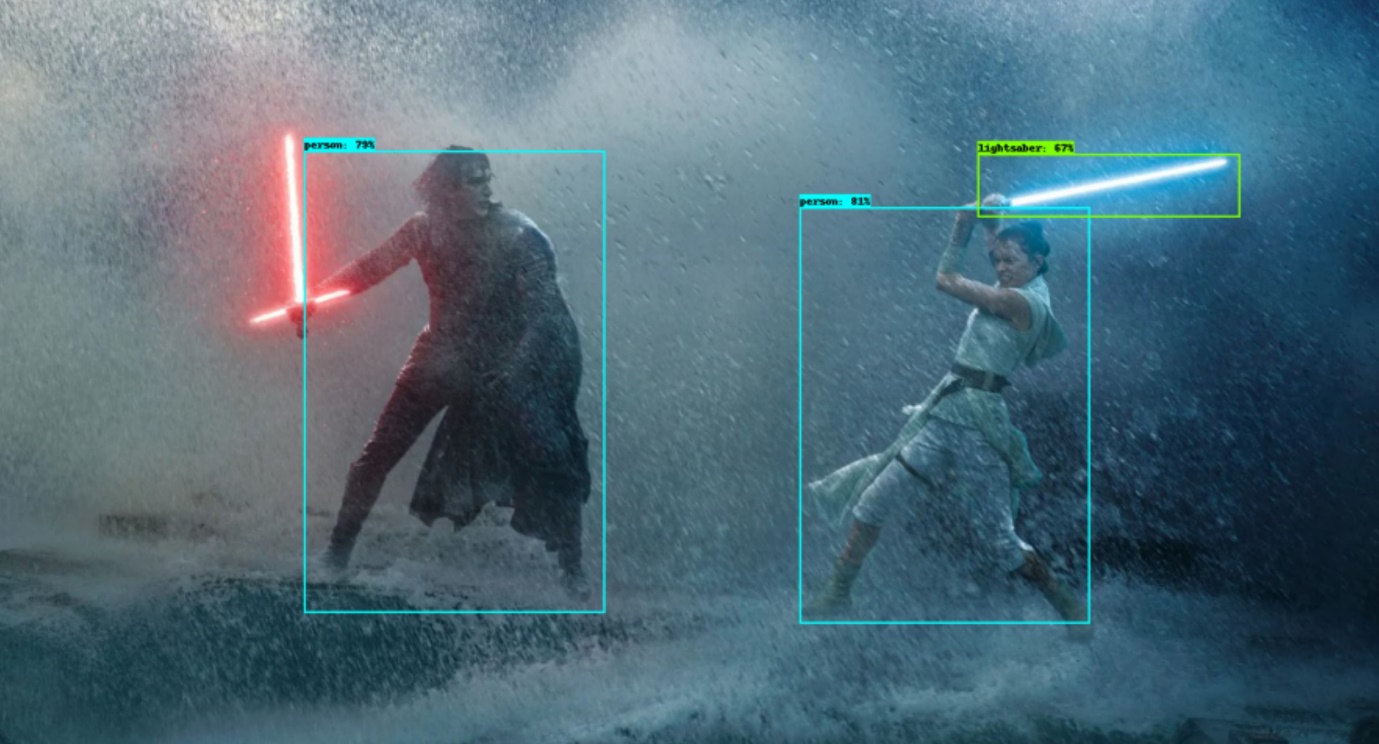
Adjusting the hyperparameters for the anchor box generators was not done as it is not applicable for CenterNet, which is an anchor-free object detection model. Instead of using anchor boxes, CenterNet detects objects by directly predicting the center point of the objects in an image using a center point heatmap.

**Conclusion**

The model trained in Run 4 performed the best with the highest mAP at IOU 0.5 of 0.8877. The corresponding Step 3800 (checkpoint 38) was exported and used to perform object detection on the testing image and video.

The test video can be found here: <https://youtu.be/Trdo-OBcIYs>

Test image:



It appears that both persons were accurately detected, but only one of the two lightsabers were detected. This may be due to the fact that Kylo Ren’s (red) lightsaber is of a unique shape with two smaller laser beams coming out from the side, as compared to most lightsabers which have a similar shape as Rey’s (blue) lightsaber. To address this, training with more examples of the red lightsaber may allow the model to better pick out these unique lightsabers.