ITI107 – Deep learning Networks Assignment 2

Magdalene Cheong 23A467J (29 Dec 2023)

1) **Data collection and annotation process (Link to dataset: burger_fries_data)**

Source: Google

Method: LabelImg

80 images of burger and fries are collected. 17 were discarded due to poor image quality and poor object representation. Bounding boxes are drawn to classify the objects as 'burger' or 'fries'. The images and the accompanying xml files are then uploaded for training. A deliberate effort is put in to select burgers of different types – e.g. with veg/no veg, different meat colour, bun with and without sesame seeds. The same is done for fries: thin, thick, yellow and burnt. The orientation is also different. The burgers could be tilted up or down or positioned at an angle; the fries are either front-facing showing the tips or showing the long side. The background is a variety of colours and brightness. The purpose of these variations is to introduce invariance so that the developed model has a better detection accuracy, like not wrongfully detect the background as the labelled objects.

One challenge faced is deciding the extent of both objects co-existing within the same bounding box. Thus, a guideline is set such that the bounding box area is drawn in such a way that at least 85% of the area is the object to be labelled. Another challenge is the availability of images of reasonable resolution.

Examples of data with the bounding boxes:


2) **Experimentation**

Base Model:

Model	Batch Size	Learning Rate	classification_weight	Localization_weight
ssd_mobilenet_v2_320x320_coco17_tpu-8 - base model	10	0.039999999	1	1

This model is selected as the base model due to computational resource constraint, simplicity of the task and that real-time feedback is required at deployment. SSD-MobileNet-v2-FPNLite 320x320 is evaluated alongside with SSD-MobileNet-v2 320x320. The strength of the FPN is the ability to detect small objects. In the objects chosen, some items are relatively small, like mushroom, part of lettuce/tomato and fries oriented such that only the tips are visible. This model is evaluated to test if it helps to enhance detect accuracy above SSD-MobileNet-v2 320x320 capability.

Metrics:

Metrics captured are mAP at 0.5 IOU and total loss. The difference between eval loss and training loss is derived to compare overfitting. Data is collected at 2 points:

i) Where mAP starts to plateau

ii) Where eval loss starts to flatten

Legend:

For clarity and readability, in the report, the following are referred to with a shorter name:

i) SSD-MobileNet-v2 320x320 as 'mobilenet'

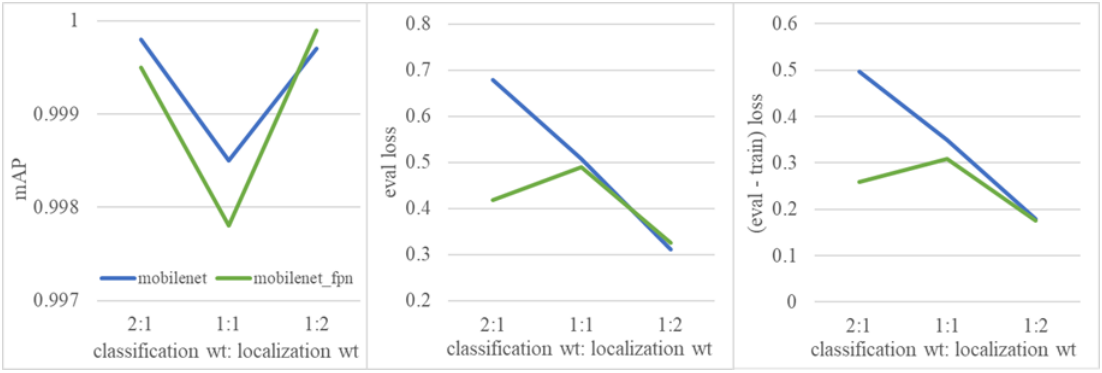
ii) MobileNet-v2-FPNLite 320x320 as 'mobilenet_fpn'

iii) mAP at 0.5 IOU as 'mAP'

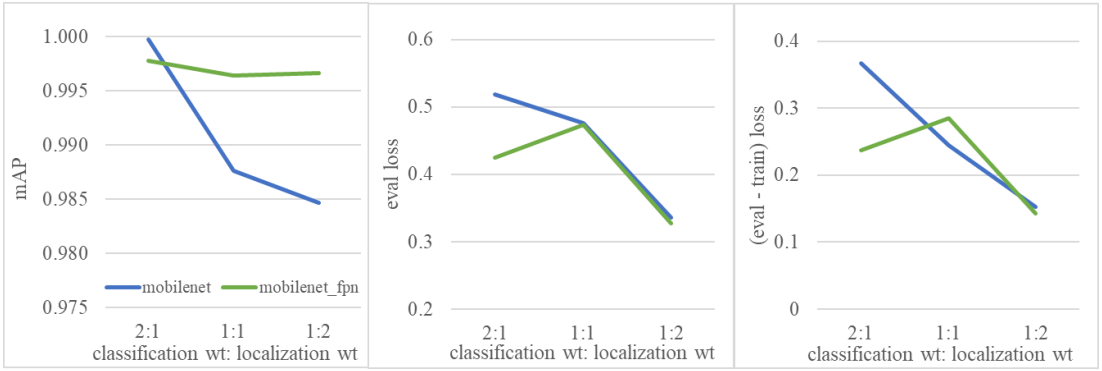
a) Classification and localization weights evaluation
Ratio of classification weight to localization weight is varied to understand their effects.

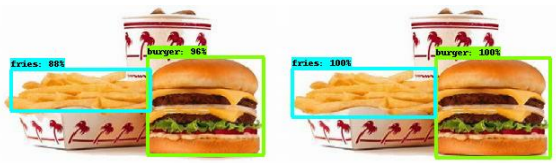
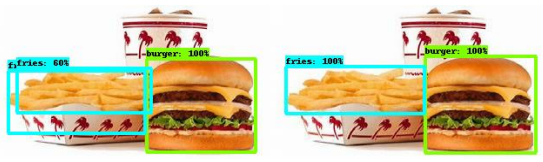
Classification_weight	Localization_weight	Ratio	Batch Size	Learning Rate
1	0.5	2:1	10	0.039999999
1	1	1:1	10	0.039999999
0.5	1	1:2	10	0.039999999

mAP approaching plateau



Eval loss starts to flatten



classification wt: localization wt = 1:2 (mobilenet)	**classification wt: localization wt = 1:2 (mobilenet_fpn)**
	

Discussion:

- mAP plateaus between 6.5k to 9.9k steps whereas eval loss starts to flatten between 3.8k to 6k steps for mobilenet. mAP is very good for all of these runs with min value of 0.985. By the time mAP plateaus, the model is already overfitting. Therefore, for the following runs, model performance will be compared only at the point eval loss starts to flatten.
- Both eval loss and (eval-train) loss are lowest for ratio 1:2 (Classification weight 0.5, localization weight 1.0). The lower the eval loss, the better is the performance. The lower the (eval-train) loss, the less overfitting there is.

- For each set of images, right object is manually labelled, left is outcome of training. From the train and eval images above, it appears that mobilenet_fpn did not detect accurately as the bounding box included the carton box in the image.
- According to the data (Ref: Expt Log in ITI107 Assignment 2 - Mag), recall (AR@1, AR@10, AR@100) for mobilenet is between 0.675 and 0.705 at the point eval loss flattens. Recall for mobilenet_fpn is between 0.68 – 0.70. It is inconclusive yet which model is better.
- Both models are overfitting as (eval-train) loss is +0.15.

⇨ Set classification weight as 0.5 and localization weight as 1.0

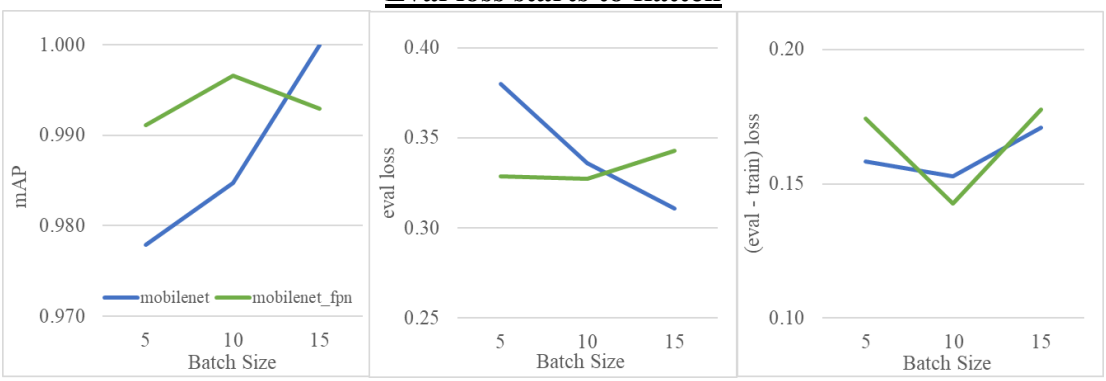
⇨ Objective for next set of runs is to reduce the overfitting.

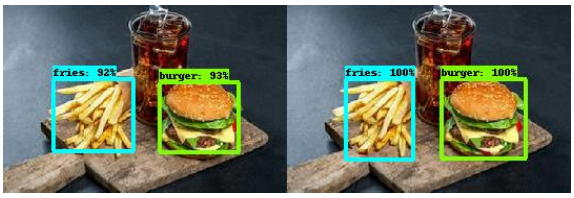
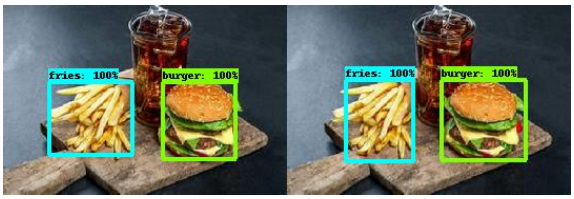
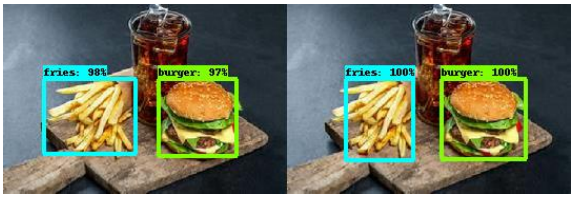
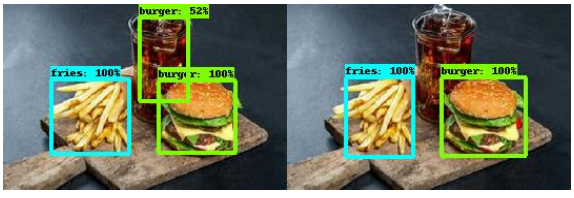
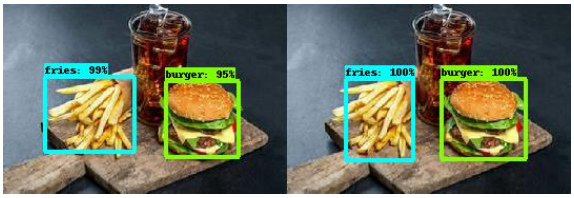
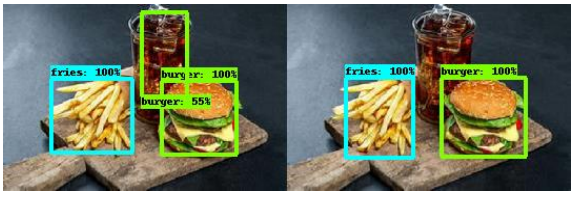
b) Batch size evaluation

Vary batch size to see if overfitting can be reduced.

Classification_weight	Localization_weight	Ratio	Batch Size	Learning Rate
0.5	1	1:2	5	0.039999999
0.5	1	1:2	10	0.039999999
0.5	1	1:2	15	0.039999999

Eval loss starts to flatten



Batch size	mobilenet	mobilenet_fpn
5		
10		
15		

Discussion:

- There is no remarkable difference between the 2 models.
- While eval loss looks better for larger batch size, the difference between eval loss and training loss is comparable. This means batch size plays no part in reducing overfitting.
- From the train and eval images above, mobilenet_fpn has again detected wrongly. For both batch size 10 and 15, it recognises the drink as burger.
- Even though the values of recall (AR@1, AR@10, AR@100) are comparable among the 3 batch sizes, it seems that at larger batch size, the chance of false positive is higher. Batch size 10 appears to be the margin for occurrence of false positive.
- While lower batch size of 5 has no false positive based on the limited data, the eval loss appears higher than larger batch size. The final model batch size will be either 5 or 10.

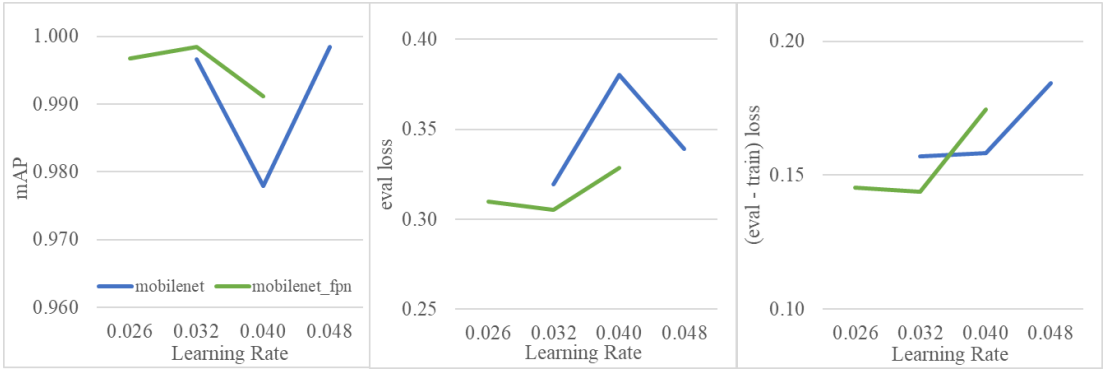
⇨ For the next evaluation on learning rate, batch size of 5 will be used so that wrong detection at batch size 10 and 15 for mobilenet_fpn will not interfere with the evaluation outcome.

c) Learning rate evaluation

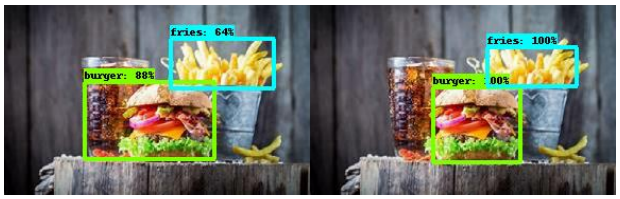
Vary learning rate in steps of 20% difference and observe effect on overfitting. Learning rate is represented in findings below in 3 decimal places for better readability.

Classification_weight	Localization_weight	Ratio	Batch Size	Learning Rate (LR)	LR (round to 3 dp)
0.5	1	1:2	5	0.025599999	0.026
0.5	1	1:2	5	0.031999999	0.032
0.5	1	1:2	5	0.039999999	0.040
0.5	1	1:2	5	0.047999999	0.048

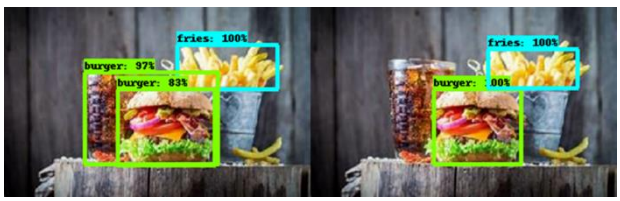
Eval loss starts to flatten

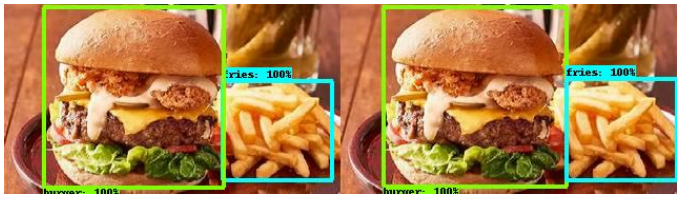


Low Learning Rate (0.032) (mobilenet)



Low Learning Rate (0.032) (mobilenet_fpn)





Discussion:

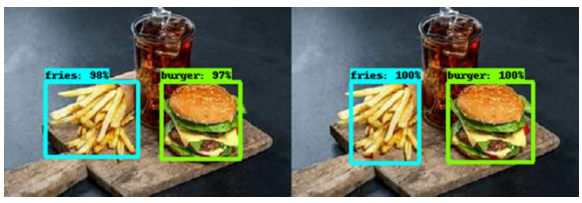
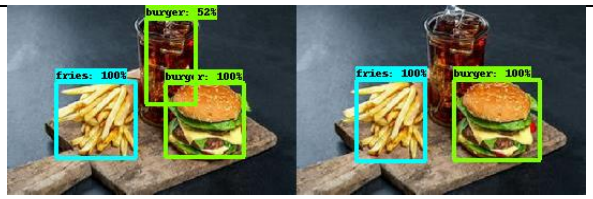
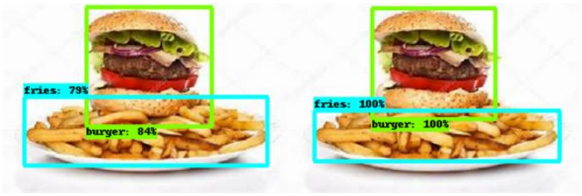
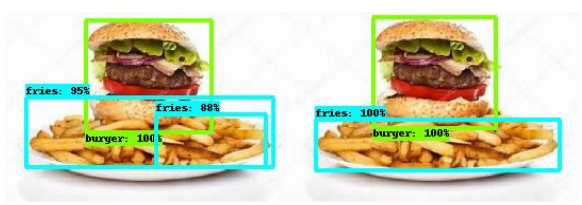
- mAP remains high for both models at different learning rates.
- (eval-train) loss appear to be slightly better for higher learning rate but it has a higher eval loss and the difference as compared to other learning rates is not large.
- At low learning rates below 0.032, both models are not detecting correctly. At learning rate 0.04, based on the limited data, there is no false positive for mobilenet but there is for mobilenet_fpn.
- The original learning rate of 0.04 appears to be the optimum setting against wrong detection.
- Learning rate 0.048 will not be used now unless it is further tested to check for convergence issue.

Conclusion:

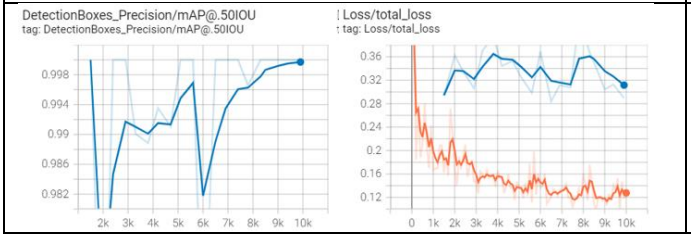
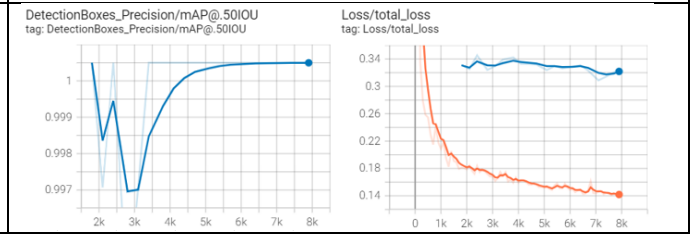
As discussed, classification wt and localization rate are best set at 0.5 and 1 respectively. It is also found that learning rate is optimal at 0.0399999991059303. From recall results table and eval images below, while batch size 5 has no wrong detection for both models, the recall is lowest. Batch size 10 appears to be the margin for wrong detection for SSD-MobileNet-v2-FPNLite. To be prudent and to ensure sufficient margin, batch size will be set at 10 and final model will be SSD-MobileNet-v2. Throughput for SSD-MobileNet-v2 is also higher for SSD-MobileNet-v2-FPNLite (7.45 steps/s vs 5.48 steps/s)

Recall results:

Batch size	SSD-MobileNet-v2				SSD-MobileNet-v2-FPNLite			
	AR@1	AR@10	AR@100	Wrong detection	AR@1	AR@10	AR@100	Wrong detection
5	0.670	0.697	0.702	No	0.700	0.725	0.730	No
10	0.695	0.710	0.718	No	0.712	0.734	0.734	Yes
15	0.715	0.717	0.720	No	0.710	0.730	0.720	Yes

SSD-MobileNet-v2	SSD-MobileNet-v2-FPNLite
	
	

mAP and Total Loss (final model parameters):

SSD-MobileNet-v2	SSD-MobileNet-v2-FPNLite
	

Though mAP and Total Loss charts do not appear to differ between the 2 models. For reasons discussed above, SSD-MobileNet-v2 is better.

Finally, table below compares final model against the base model:

Model	mAP	eval_loss	train_loss	(eval - train) loss	avg recall	Steps/s
ssd_mobilenet_v2_320x320_coco17_tpu-8 - base model	0.9876	0.4759	0.2306	0.2453	0.689	7.49
ssd_mobilenet_v2_320x320_coco17_tpu-8 - final model	**0.9847**	**0.3358**	**0.1830**	**0.1528**	**0.692**	**7.45**

The final model is an improvement from the model. It has equally high mAP and throughput and at much lesser overfitting. Recall however is not high for both. Training with a larger dataset and reduce the layer might help to improve the overfitting and recall.

Link to deployed model: https://huggingface.co/spaces/ITI107-2023S2/23A467J
Expt Log: ITI107 Assignment 2 - Mag