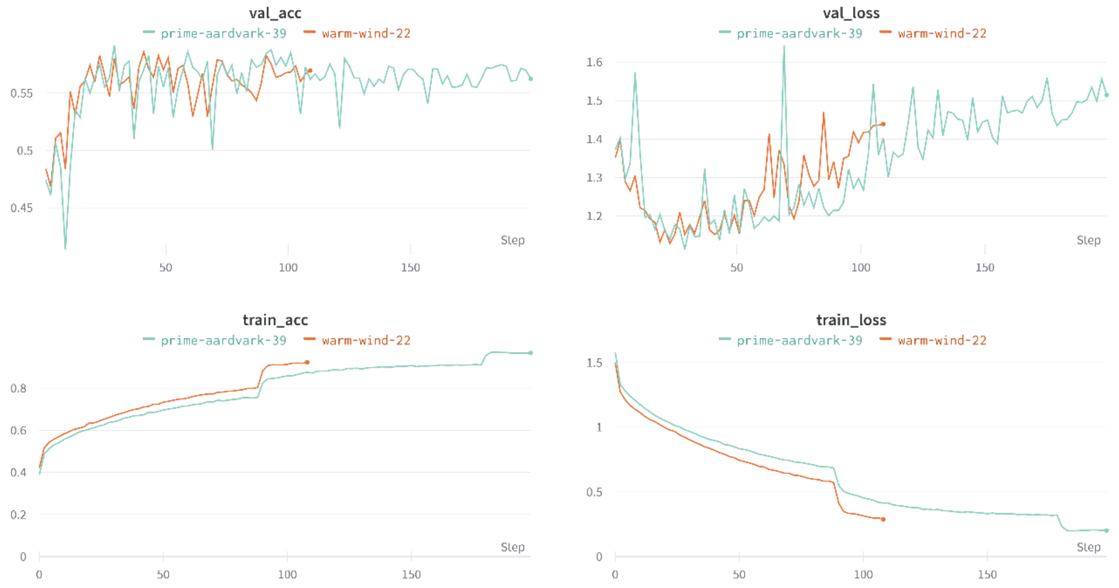
學號：R11921A16　系級：電機碩一　　姓名：何秉學

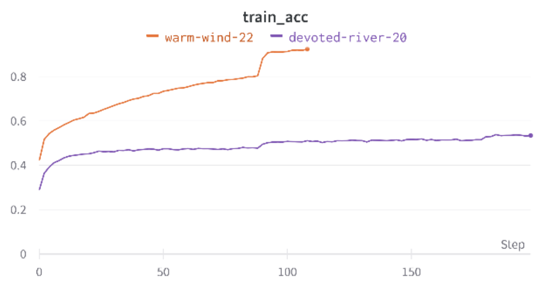
1. (1%) 實作 early-stopping，繪製 training, validation loss 的 learning curve，比較實作前後的差異，並說明 early-stopping 的運作機制

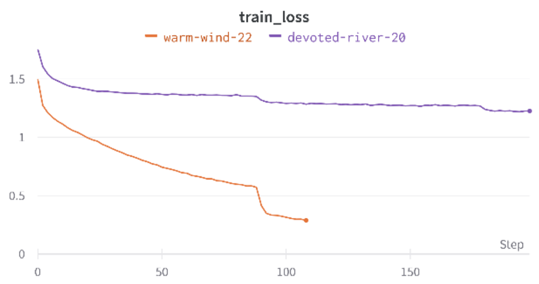
   As you can see below, if I use early stopping technique, it'll break the training loop when overfitting. The orange line is what I set early stopping with threshold 5. That is, if the model loss rise up 5 times consequently, then stop training. The other one doesn't set early stopping and you can see it'll complete the training loop even overfitting occur.

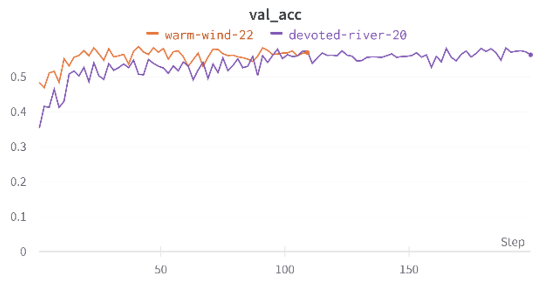


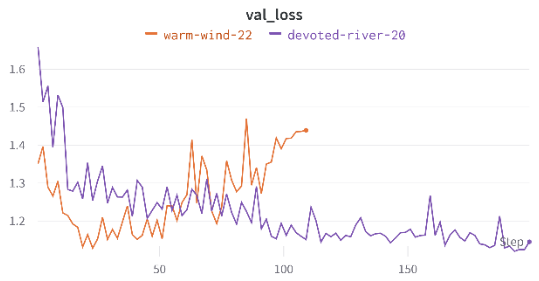
2. (1%) 嘗試使用 augmentation，說明實作細節並比較有無該 trick 對結果表現的影響(validation 或是 testing 擇一即可)，且需說明為何使用這些 augmentation 的原因。(ref: https://pytorch.org/vision/stable/transforms.html)

   As you can see below, if I use data augmentation, it can conquer overfitting. The other configurations are the same and the breakpoint of orange line is because of early-stopping. I set data augmentation technique on purple one and the others didn't.









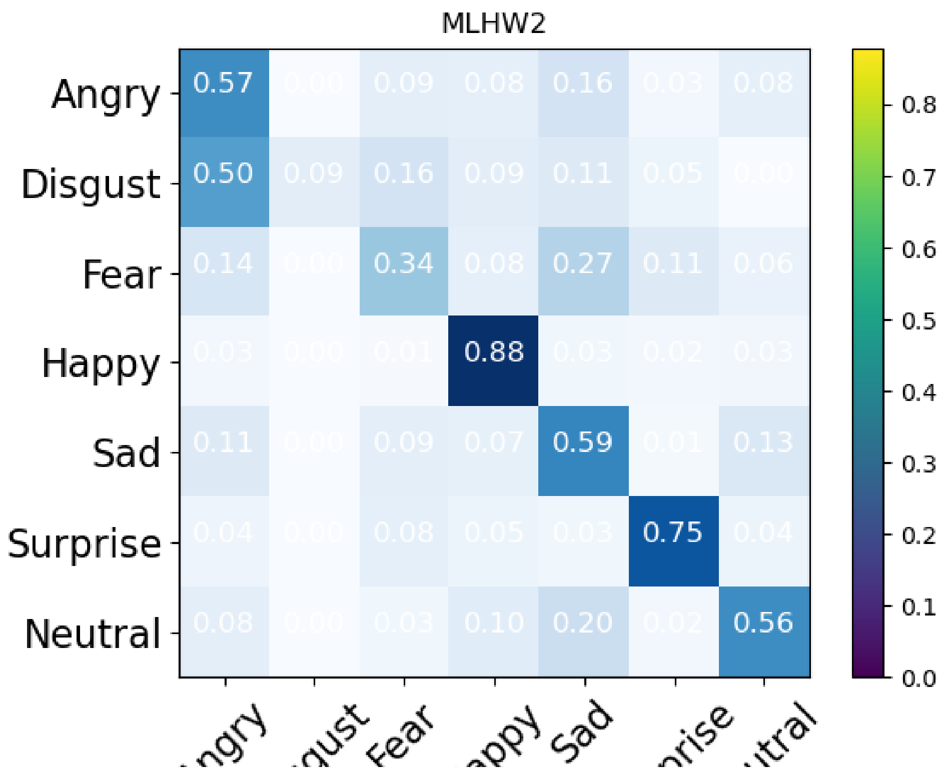
I use RandomChoice to choose transform_set randomly including RandomHorizontalFlip, ColorJitter, RandomRotation and used Resize to resize all images to 224*224 that for ResNet18 input size. ColorJitter will adjust the brightness, contrast, saturation and hue of the input image randomly. So, it can increase the diversity of training dataset properly. Though, lecture TA is not very suggestive to use RandomVerticalFlip skill on training image, because it'll transform the image that no human can recognize it. So, I use RandomHorizontalFlip instead.

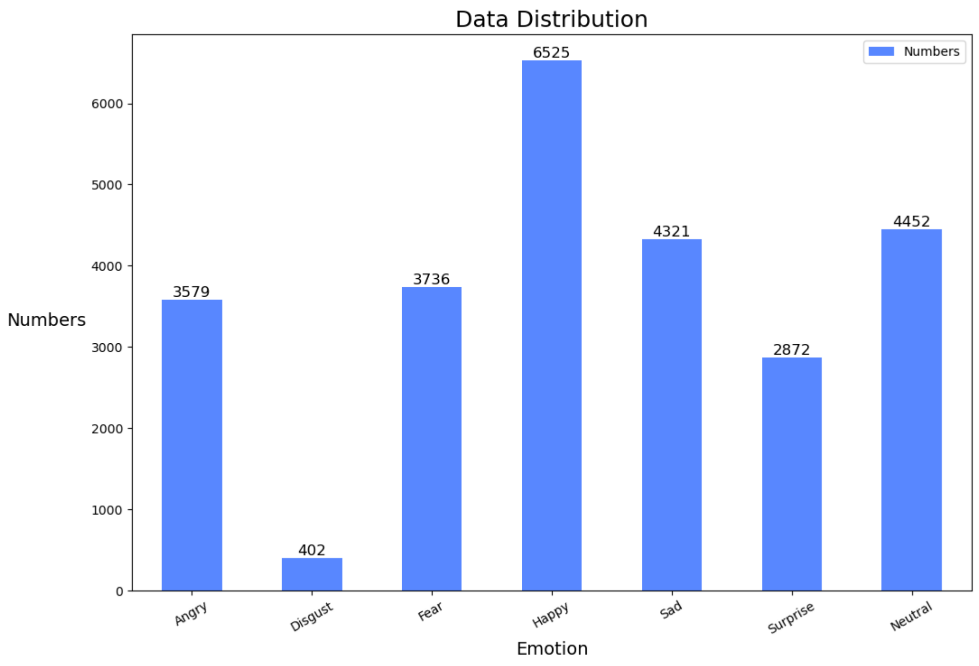
```
transform_set = [
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.ColorJitter(brightness=(0, 5), contrast=(0, 5),
saturation=(0, 5), hue=(-0.1, 0.1)),
    transforms.RandomRotation(30, center=(0, 0), expand=False),]

transform_aug = transforms.Compose([
    transforms.RandomChoice(transform_set),
    transforms.Resize(224)])
```

3. (1%) 畫出 confusion matrix 分析哪些類別的圖片容易使 model 搞混，找出模型出錯的例子，並分析可能的原因。

   (ref: https://en.wikipedia.org/wiki/Confusion_matrix)

   As you can see the confusion matrix below. The second class(emotion Disgust) is the worst result of the classification and the Happy class is the best. Also, the Fear class is not good enough. I think the main reason is data imbalance that shown below of second one. The prior probability of these two classes are 0.0155 and 0.1443 respectively. Under this circumstance, the model can't learn this class by enough images properly. And the bad result of Fear class. I think it's just not learn very well with bad model structure and bad configuration.





4. (1%) 請統計訓練資料中不同類別的數量比例，並說明：對 testing 或是 validation 來說，不針對特定類別，直接選擇機率最大的類別會是最好的結果嗎？(ref: https://arxiv.org/pdf/1608.06048.pdf , or hints: imbalanced classification)

The result of data distribution is shown above. The prior probability of the highest number training class is 6525/25887=0.252. If not targeting a specific category and just choose the Happy class, it would be worse than the result that generate by normal classification.

5. (4%) Refer to math problem:
https://hackmd.io/@lH2AB7kCSAS3NPw2FffsGg/r1otQp7Gi?fbclid=IwAR0cs5CajVy_zhDmHEDgze2V1_Jlxp95N45BF6hg1l6CgG-6IViYGAIGReE