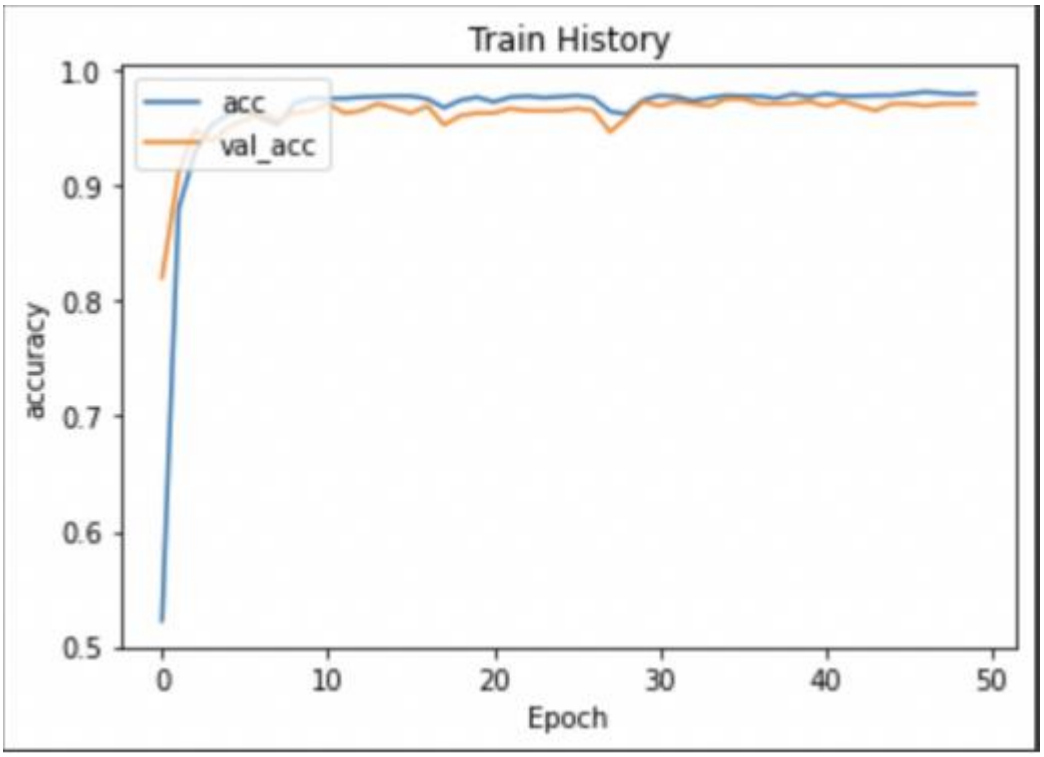
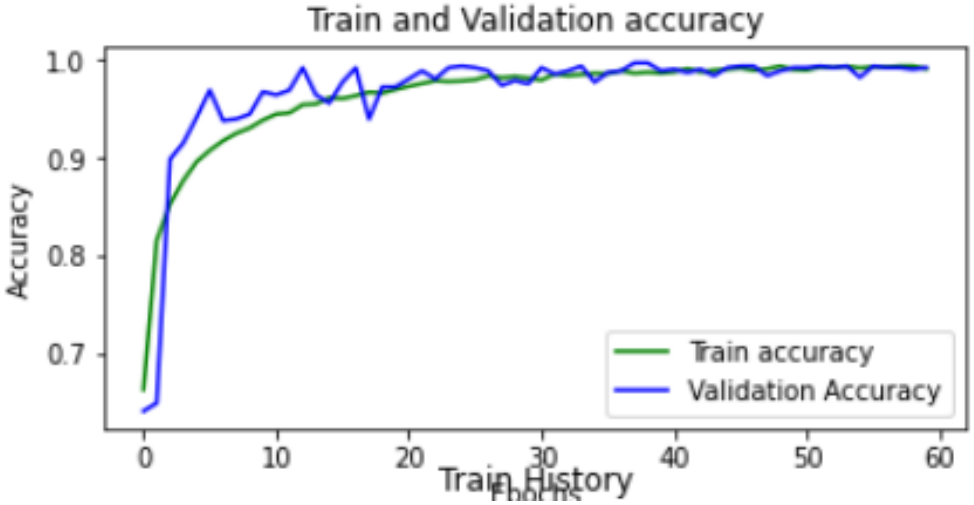
一、 工件辨識準確率及損失率請附上

(1) Train accuracy history 圖

Ex ：

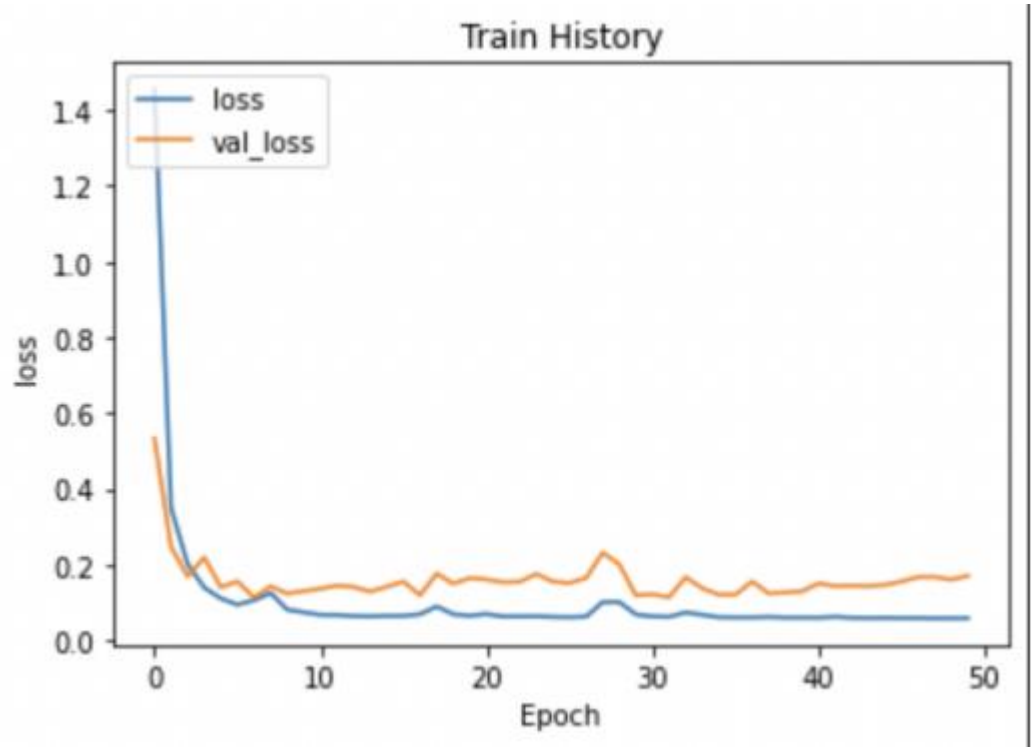


Train accuracy history 圖:

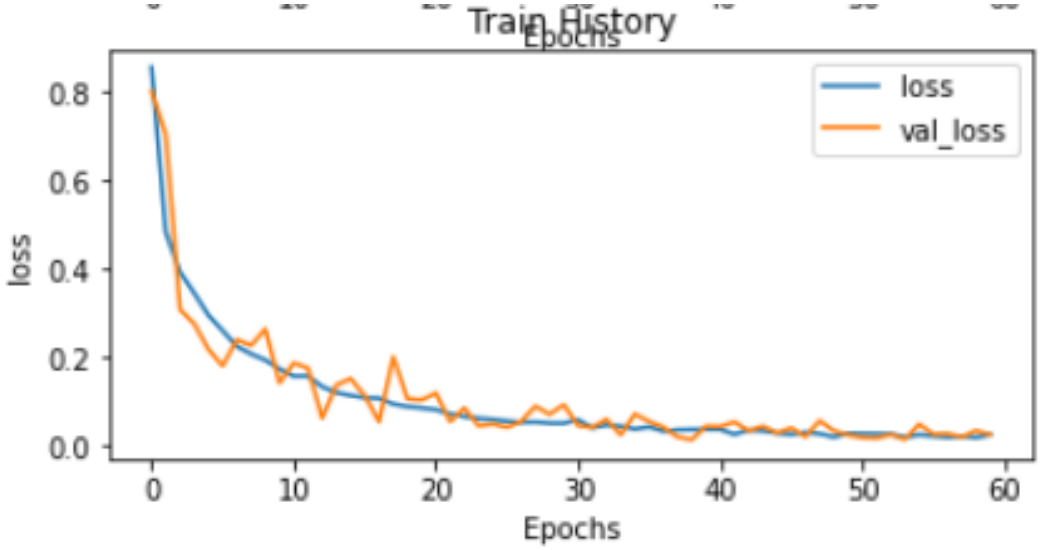


(2) Train loss history 圖

Ex ：



Train loss history 圖:



(3) Test loss ,Test accuracy 截圖:

```
Test loss: 0.058736883103847504
Test accuracy 0.9835957884788513
```

二、請詳述你如何實作期末專題，包括資料前處理、選擇模型建立、調整參數…等等

最初進行的是資料的前處理，由於這次給予的資料包和講義中的範例在路徑上不太一樣，因此要讀取檔案需要採用不一樣的方式：依序讀取四個工件的資料夾，並且依據當前資料夾給予讀取的圖片不同的 Label。另外，由於未知原因，無法像講義中一樣直接對讀取到的圖片進行正規化與 reshape。在仔細研究過錯誤信息以及上網找了一翻資料後，我在進行正規化與 reshape 前，先將圖片用 resize 縮小到(28,28)的大小後，方才能夠進行正規化以及標準化。

完成資料前處理後，我開始建立模型，最初我選擇的是依照講義中的方式進行模型建立，只在數量上修改成 4 個工件之數量。完成建模後，我開始嘗試運行，且為了在更短的時間內得到結果，我最初將 Epochs 設定為 10。令我比較驚訝的是，在 Epochs 為 10 且參數大多沒變的情形下，accuracy 竟然就能夠達到 0.8 左右。

之後我開始嘗試提升 accuracy，由於之前助教提到過，若想提高 accuracy，可以在前處理的時候對圖片進行一些調整，使 accuracy 更高。當時我想到的方法是，將讀取到

的圖片旋轉三個角度，得到該圖片 0 度、90 度、180 度、270 度的樣子後，再將它們都丟到 x_data 與 y_data 中。令我意外的是，accuracy 並沒有提升，反而降低到了 0.2 左右。

之後我還有嘗試將圖片取鏡面後再輸入，結果一樣，accuracy 下降了取多。接著我還有嘗試對參數進行更改，不論是對 filter 的數量進行增加或減少、放大 kernel size、增加更多的卷積層與池化層，或者是減少 Dropout 又或是增加或減少 Dense，無一例外都導致了 accuracy 的降低，只有在小小的改動了一下 kernel 時才沒有對 accuracy 有太大的影響。且雖然增加 Epochs 次數可以提高 accuracy，但是到達一定次數後果並不明顯，且時間花費極長。

之後我想到，講義中給的例子是辨識手寫字，我想到，由於是數字，且又是手寫的，因此圖片是黑白的或許並不會對 accuracy 造成甚麼影響。但是本次辨公件給予的圖片顏色非常相近，且還有反光，因此我將原本讀取黑白圖片的模式改成讀取彩色圖片的模式。果不其然，在改成讀取彩色圖片後，在 Epochs 為 10 的情況下，accuracy 便上升

到了 0.94 左右，大大的提升，看來彩色的圖片確實可以提告 accuracy。

接著，我又嘗試將 Epochs 的次數進行提高，發現在次數大於 50 次後，accuracy 大於 0.99 後便不會有太大的提升，且在這種 accuracy 極高的情況下，測試 Test 資料的 accuracy 反而會有些微的下降。因此最後我將 Epochs，成功率維持在 0.98 左右。

此次的 Midterm Project 製作過程大概就是這些，不得不說，就算在 Epochs = 10 的情形下進行訓練，花的時間一樣不少，感覺其實蠻多時間花在等待上。另外我有試過在讀取圖片的時候用 multiprocessing 的方式來增加讀取的速度，結果嘗試了幾次發現似乎是因為這會造成很大的負擔，導致我多次被 Colab 踢出來，浪費許多時間。