Final project：Compression deepfake detect

組員：李權恩、汪玄同、吳玫萱

一、介紹：

近年來 deepfake 議題快速地受到重視，而現在也已經有不少辨識方法能達到 90%以上的辨識精確度，但移到現實世界場景下使用時卻大大的不如預期，其中關鍵的原因之一就是在社群網路上的影像多數都是經過壓縮的，經過壓縮的圖片會出現類似馬賽克的色塊問題，而視覺上也就會變得比較模糊，同時也會造成深度學習方法在進行真偽辨識時的效能大幅減弱，故如何克服壓縮照片難以辨識的問題再現實場景下是十分重要得問題，故對此我們針對被應用於壓縮影像的兩個先前研究[1, 2]進行實踐、變化以取得相應特徵再結合應用，並且使用 Logistic、Xgboost、SVM、KNN 與 MLP 共五種模型進行分群，以達到真偽辨識的目標。

我們的研究分為兩個部分，第一部分受[1]啟發，我們先將圖片成多個 8*8 的區塊，在將其轉移至頻率域，然後將所有 8*8 的區塊編碼為 1~64 的位置，將每個位置視為一個變數，對除了代表亮度的第一個位置外的所有位置進行係數估計，最後每張圖片皆會得到一個 63 維度的特徵提取向量，並用運該向量進行真為分群。

我們研究的第二部分為實踐[2]，因為大多偽造人臉的方法皆使用 GAN，該方法的目標是對卷機痕跡進行偵測，具體來說是對每一張圖片訓練出一個核函數使其代表輸入圖片的卷基痕跡，然後使用一個分類器去對每個圖片所得的核函數做分群，並在測試階段藉由這個核函數分群結果判斷圖片真偽，以達到最終判別壓縮圖片真偽的目的。

最後我們除了兩個分法分別的特徵分群結果，也會結合兩個方法所得的特徵去進行分群，因為第一部份的研究是針對頻預空間特徵進行特徵萃取，而第二部分則是針對 RGB 色彩空間進行特徵探索，我們將兩者結合可兼顧兩個不同空間的特徵資訊，幫助模型進行真偽辨識，以下我們將詳細的介紹兩個方法並展現我們的研究結果。

以下是我們研究的主要重點整理：

- 同時針對頻域與 RGB 色彩空間進行特徵提取，可以結合兩者的特徵資訊幫助模型進行真偽圖片辨識。
- 將圖片資訊轉至頻率空間，藉由提取頻譜資訊區分真假圖片，提高分群效能，而避免模型學習人臉長相去記憶標籤匹配的傾向。

二、 研究方法

2.1. 特徵提取方法一:Detecting Gan DCT Anomalies

文章中將圖片切割成8×8的小塊，並使用 DCT 將多個小塊轉成平率空間的資料，藉由偽造圖片可能會有較少的高頻特徵判斷圖片是否為偽造圖片，下列為特徵提取的流程。

假設輸入圖片$H \times W \times C$維度

步驟一:

使用 zero padding 將圖片的H,W擴增為 8 的整數倍，得到的圖片維度假設維度維$(h * 8) \times (w * 8) * C$，將圖片分割成$(h * w * C)$個$8 \times 8$的小塊

步驟二:

對分割後的每個8×8的小塊使用 2d-DCT 轉到頻率空間中，對每個小塊使用 Zig-Zag 拉直成 64 維度的向量，將得到的所有向量進行合併可以得到下方的矩陣

$$\begin{bmatrix} d_{1,0} & \cdots & d_{1,63} \\ \vdots & \ddots & \vdots \\ d_{h*w*c,0} & \cdots & d_{h*w*c,63} \end{bmatrix}$$

矩陣維度為$(h * w * C) \times 64$

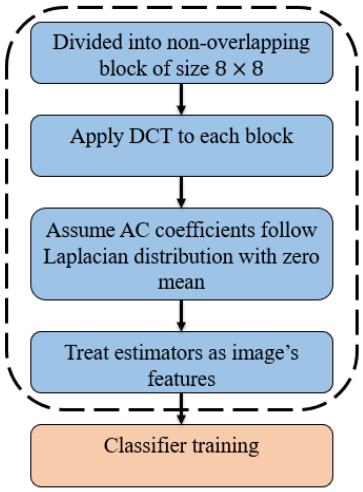
步驟三:

使用下列式子估計向量$\beta = [\beta_1, \beta_2, \dots, \beta_{63}]$

$$\beta = \frac{1}{h * w * C} \sum_{h*w*C}^{i=1} [|d_{i,1}|, \dots, |d_{i,63}|]$$

最後用β作為特徵提取後的向量。

之所以只使用$d_{i,1} \sim d_{i,63}$是因為 DCT 的結果第一個元素代表的是圖片的整體亮度，後面的元素才代表低頻率和高頻率的特徵，所以最後得到的β為 63 維度的向量，下圖為流程圖



2.2 特徵提取方法二:Convolutional Traces

因為大多偽造人臉的方法皆使用 GAN，而 GAN 中的 generator 在模型的最後一層皆會使用 transpose convolution，文中假設使用 transpose convolution 生成的圖片皆會滿足以下式子

$$I[x,y] = \sum_{s,t=-\alpha}^{\alpha} k_{s,t} * I[x+s, y+t] \dots\dots\dots (1)$$

其中I為生成的圖片，k為進行 convolution 的 kernel，$k_{s,t}$為 kernel 的第 s 列 t 行的值，這個式子可以理解為只要是經過 transpose convolution 生成的圖片，皆可以找到一個 kernel k對這個圖片做捲積仍然會得到這個圖片。

文章中假設若一張圖片滿足式子(1)，則圖片的每一個像素都會屬於高斯分佈

$$M_1 \sim \text{Gaussian}\left(\sum_{s,t=-\alpha}^{\alpha} k_{s,t} * I[x+s, y+t], \sigma^2 \right)$$

若不是偽造的假圖則會屬於$M_2 \sim \text{Uniform}(0,255)$，下圖為文章中的偽代碼，$M_1$和$M_2$分別代表兩個分佈

Algorithm 1: Expectation-Maximization Algorithm

Data: Image I
Result: $\vec{k}$
Initialize N //Kernel size
Initialize σ_0
Set $\vec{k}$ random of size NxN
Set R, P, W matrices with 0 values of the same size as I
Set p_0 as 1/size of the range of values of I
for $n = 1;\ n < 100\ n+ = 1$ **do**
  //Expectation Step
  for $\forall$ *values in* I **do**
    $R[x,y] = \left| I[x,y] - \sum_{s,t=-\alpha}^{\alpha} k_{s,t} I[x+s, y+t] \right|$
    $P[x,y] = \frac{1}{\sigma_n \sqrt{2\pi}} e^{-\frac{R[x,y]}{2\sigma_n^2}}$
    $W[x,y] = \frac{P[x,y]}{P[x,y]+p_0}$
  //Maximization Step
  Calculate $k_{s,t}^{(n+1)}$ as shown in the formula 7

演算法中每次迭代更新 kernel k可分為兩部分，第一部分為 Expectation step 第二部分為 Maximization step，下列為提取特徵的流程

假設輸入一張H * W * C的圖片I

步驟一:
初始化一個$3 \times 3 \times C$的 kernel k並設定正中間的值固定為 0

步驟二(Expectation step):
用以下式子計算$R[x,y]$, $P[x,\ y]$, $W[x,y]$

$$\mathrm{R[x,y]} = \left| \mathrm{I[x,y]} - \sum_{s,t=-\alpha}^{\alpha} k_{s,t} * I[x+s, y+t] \right|$$

$$\mathrm{P\{I[x,y]\ \mid\ I[x,y] \in M_1\}} = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{(\mathrm{R[x,y]})^2}{2\sigma^2}}$$

$$\mathrm{W[x,y]} = \frac{\mathrm{P[x,y]}}{\mathrm{P[x,y]} + \mathrm{P_0}}$$

W[x, y]中的$\mathrm{P_0}$為 uniform(0,255)中任一可能值的機率值，所以$\mathrm{P_0}$為$\frac{1}{256}$

步驟三(Maximization step):

使用下列式子更新 kernel k和σ^2，第一個式子為更新$k_{i,j}$的式子第二式子為更新σ^2的式子

$$\sum_{s,t=-\alpha}^{\alpha} k_{s,t}\left(\sum_{x,y} w[x,y] * I[x+i,y+j] * I[x+s,y+t]\right) = \sum_{x,y} w[x,y] * I[x+i,y+j] * I[x,y]$$

$$\sigma^2 = \frac{\sum_{x,y} R^2 \times \frac{P_0}{(P+P_0)^2} \times P \times R^2}{\sum_{x,y} \frac{P_0}{P+(P_0)^2} \times P \times R^2}$$

重複步驟二和三直到 kernel 收斂，將得到的 kernel 除了正中間 0 的部分拉直為一個向量可得到$8 * C$的向量，由於此次實驗輸入圖片皆為 RGB 三通到圖片，故提取的特徵向量長度為 24。

三、 實驗設置

我們使用 Faceforensics++[3]資料集進行實驗，其中包含 1000 部真影片，和 4000 部假影片，而這 4000 部假影片中又分為 DeepFakes (DF)、Face2Face (F2F)、FaceSwap (FS)、NeuralTextures (NT)四種偽造方法各 1000 部。
並選用 Logistic、Xgboost、SVM、KNN 與 MLP 共五種模型進行分群，其中包含四種機器學習方法與一種深度學習方法皆去探討特徵的分群結果。

實驗部分分為兩種，第一種使用原始的 Faceforensics++影片截下的幀進行實驗，第二部份我們進一步將實驗所用的每一幀進行臉部截取，因為我們的特徵主要目的是將真臉與假臉分開，所以在除了偽造部分的像素依然是真的，這會導致模型擷取下來的特徵即使在偽造圖上也有來自真實圖片的資訊，成為干擾模型的雜訊，在參考的兩篇原始論文中他們的實驗資料集使用整張偽造的圖片進行實驗，這樣也保證了他們的實驗不會因此受到干擾，故在第二部份我們對每一幀進行了人臉徵測與截取，試圖進一步增加實驗效能。

我們的訓練集與測試資料集依照 FaceForensics ++的原始設定，其中訓練部分真影片共 720 部每個影片取一張圖片，而假影片共分四種 Deepfake、Face2Face、FaceSwap、NeuralTextures 其中每種偽造方法各取 180 個影片，每個影片取一張圖片共 720 張圖片，這樣取的目的是達到真偽訓練資料平衡，以免偽造圖片數量遠大於真圖，使得模型都預測為偽圖的情況。而測試集部分，真影片共 140 部每個影片取一張圖片，共 140 張圖片，而假影片共分四種

Deepfake、Face2Face、FaceSwap、NeuralTextures 其中每種偽造方法各 140 個影片，每個影片取一張圖片共 560 圖片。

而在截臉的部分則在相同基礎下去截臉，我們使用真圖片去進行截臉，並將所得的截取範圍座標運用到同一影片同一幀的真偽圖片上，但是因為截臉會有無法徵測到人臉的部分，故若該影片的每一幀皆無法徵測到人臉則跳過該影片，最終在截臉部分的訓練與測試資料集數量分別為訓練部分真影片共 691 部每個影片取一張圖片，共 691 張圖片，而假影片共分四種 Deepfake、Face2Face、FaceSwap、NeuralTextures 其中前三種偽造方法各取 172 個影片、NeuralTextures 取 175 個影片，每個影片取一張圖片共 691 張圖片。而測試集部分，真影片共 132 部每個影片取一張圖片，共 132 張圖片，而假影片共分四種 Deepfake、Face2Face、FaceSwap、NeuralTextures 其中每種偽造方法各 132 個影片，每個影片取一張圖片共 528 圖片。

四、 實驗結果

這裡分別展示沒有進行臉部截取與有進行臉部截去下方法一(DCT)、方法二(Convolutional Traces)和兩者結合後(Concat)在 Logistic、Xgboost、SVM、KNN 與 MLP 共五種模型下的效能。

4.1 沒有進行臉部截取的資料實驗結果

ACC AUC F1	Raw			c40		
	DCT	**Conv-trace**	**Concat**	**DCT**	**Conv-trace**	**Concat**
Logistic	0.7314 0.7873 0.6659	0.,5114 0.5045 0.4579	0.7114 0.7792 0.6444	0.5814 0.5994 0.5230	0.5014 0.5116 0.4556	0.5557 0.5632 0.4981
Xgboost	0.6914 0.7886 0.6375	0.8000 0.5000 0.4444	0.6586 0.7424 0.6015	0.5286 0.5160 0.4531	0.8000 0.5000 0.4444	0.5429 0.5166 0.4723
SVM	0.5586 0.5634 0.5046	0.4286 0.5063 0.4107	0.5929 0.5500 0.5146	0.4786 0.5348 0.4502	0.5586 0.5045 0.4780	0.4829 0.5348 0.4528
KNN	0.3771 0.5705 0.3769	0.3700 0.5205 0.3676	0.3757 0.5321 0.3736	0.3271 0.4884 0.3267	0.3543 0.4812 0.3504	0.3571 0.5311 0.3566
MLP	0.6700	0.5071	0.5471	0.7014	0.4971	0.4671

	0.7533 0.6148	0.5456 0.4693	0.6692 0.5155	0.5821 0.5489	0.5068 0.4534	0.5900 0.4433

這邊我們可以看到總結下來方法二 DCT 模型進行特徵萃取的效能最好，原始圖片部分以使用 Logistic 模型在準確率與 F1 score 為評估標準時有較好的結果，而 Xgboost 模型則在使用 AUC 為評估標準時有較好的結果，壓縮圖片部分以使用 MLP 模型在準確率與 F1 score 為評估標準時有較好的結果，而 Logistic 模型則在使用 AUC 為評估標準時有較好的結果。

但整體效能上其實並不理想，與原始論文效能也相差甚遠，在 DCT 和 Convolutional Traces 原始論文中都分別有 90%以上的 F1 score 和準確度，這讓我們探索這樣的結果上有什麼是我們沒有注意到需要改進的，故我們進一步做了進行了臉部截取資料的實驗，如同第三節實驗設置中提到的，因為我們的特徵主要目的是將真臉與假臉分開，所以如果在除了偽造部分以外的像素依然是真的，這會導致模型擷取下來的特徵即使在偽造圖上也有來自真實圖片的資訊，成為干擾模型的雜訊，在參考的兩篇原始論文中他們的實驗資料集使用整張偽造的圖片進行實驗，這樣也保證了他們的實驗不會因此受到干擾，故進行 4.2 節先將資料進行臉部截取後的實驗以達到跟論文較皆信的實驗標準，並且期待能達到提升整體實驗性能的目標。

4.2 進行臉部截取的資料實驗結果

ACC AUC F1	**Raw**			**c40**		
	DCT	**Conv-trace**	**Concat**	**DCT**	**Conv-trace**	**Concat**
Logistic	0.8076 0.8229 0.7536	0.5045 0.4972 0.4516	0.7788 0.7936 0.7221	0.6136 0.5630 0.5251		
Xgboost	0.8591 0.8580 0.8078	0.5212 0.5530 0.4809	0.8515 0.8589 0.8012	0.5818 0.5707 0.5118		
SVM	0.5682 0.6278 0.5322	0.4393 0.5104 0.4189	0.6045 0.6051 0.5449	0.5545 0.5426 0.4942		
KNN	0.3606 0.5322 0.3599	0.3742 0.5407 0.3729	0.3621 0.5331 0.3613	0.3500 0.5057 0.3487		
MLP	0.8742 0.9124	0.6333 0.4979	0.8348 0.8455	0.7394 0.5702	0.6742 0.5023	0.7182 0.5907

	0.8161	0.4966	0.7382	0.5756	0.5036	0.5646

根據上表我們可以發現進行截臉後的未經壓縮圖片其 DCT 在 Logistic、Xgboost 和 MLP 都有十分明顯的進步，而在 c40 壓縮影像的實驗結果在 DCT 特徵擷取方法在 MLP 分類下也有較佳的結果，可以發現針對偽造部分的擷取確實對模型訓練是有幫助的，而其原因如上所述也是較容易直觀理解的，而 DCT 主要是根據頻域中的高頻部分也就是照片中的細節紋理處特徵訊息去進行真偽辨識的，這點在我們的實驗中可以發現，在其 63 維特徵向量中高頻處的參數會具有較高的權重，也就是說高頻資訊對真偽辨識目標較為重要，但壓縮圖片會造成細節紋理的消失，轉至 DCT 頻域時高頻處資訊就會變少，同時造成真偽辨識效能下降，並且可能較難突破，同時在 DCT 特徵擷取方法的參考論文[1]中其實驗是針對單一偽造方法與真圖片進行真偽辨識的，而我們使用了 FF++中的四種偽造方法當作偽造圖片的資料來源，故這樣的真偽辨識條件是更加困難的，而這也是更值得我們去研究、挑戰的地方。

令外在 Convolutional Traces 部分，經過截臉後的特徵對模型效能並未獲得助益，其中我們所考慮的可能原因有兩個方向，一部分是在 Convolutional Traces 原始論文中實驗都是針對單一的偽造方法去進行真偽分群，可能在提取卷機痕跡的過程中來自不同的偽造方法會使其卷機痕跡特徵上會有較大的差異，而僅以核 k 中的 8 個參數(3*3 和函數去掉中心位置)並不足以表現其特徵差異，也因此難以用其來區分出真偽圖片，另一個方面則是在 Convolutional Traces 原始論文中所使用的資料數量是巨大的如下圖所示，其資料量光偽造圖片就皆超過 3000 張，真圖片加上假圖片數量較我們的實驗數據量真偽各 720 張大了很多，但因為期訓練時間成本較大，我們對現在使用的訓練集與測試集全部資料共 2140 張進行 Convolutional Traces 特徵提取就需要 5 個多小時，也就是說光進行一張照片的特徵提取就需要約 9 秒鐘的時間，所以在這次的研究中我們無法使用較大的資料量對這個造成效能低落的可能原因進行驗證。

Method	Number of images generated	Size	Data input to the network	Goal of the network	Kernel size of the latest Convolution Layer
GDWCT [5]	3369	216x216	CELEBA	Improves the styling capability	4x4
STARGAN [6]	5648	256x256	CELEBA	Image-to-image translations on multiple domains using a single model	7x7
ATTGAN [15]	6005	256x256	CELEBA	Transfer of face attributes with classification constraints	4x4
STYLEGAN [19]	9999	1024x1024	CELEBA-HQ FFHQ	Transfer semantic content from a source domain to a target domain characterized by a different style	3x3
STYLEGAN2 [20]	3000	1024x1024	FFHQ	Transfer semantic content from a source domain to a target domain characterized by a different style	3x3

Table 1. Details of Deepfake GAN architectures employed for analysis. For each one is reported: all images generated, the generated image sizes, the original input used to train the neural network, the goal of the network and the kernel size of last convolutional layer.

五、發現與檢討

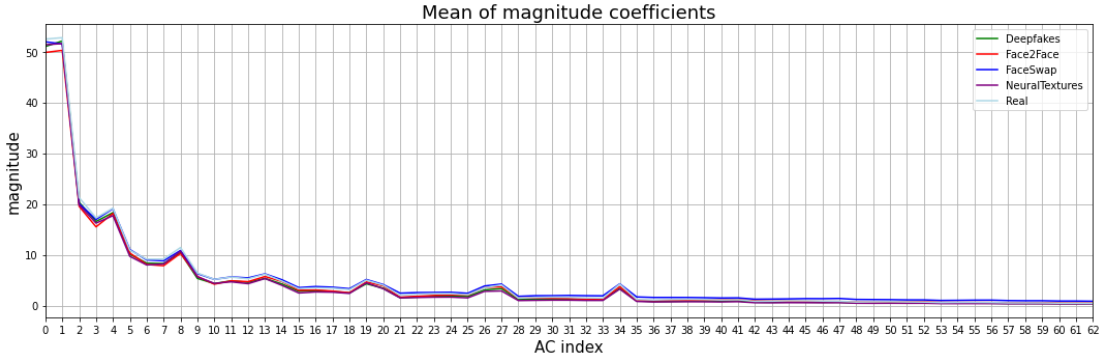
在前一節探討了實驗結果，其實都沒有像參考文獻所呈現的結果，在本節我們主要想探討此兩種方法在本次使用的資料集上出現的狀況，以及一些造成模型效能不佳的潛在因素。最後再簡短的說明我們認為根據目前的結果，未來應如何接下去進行分析。

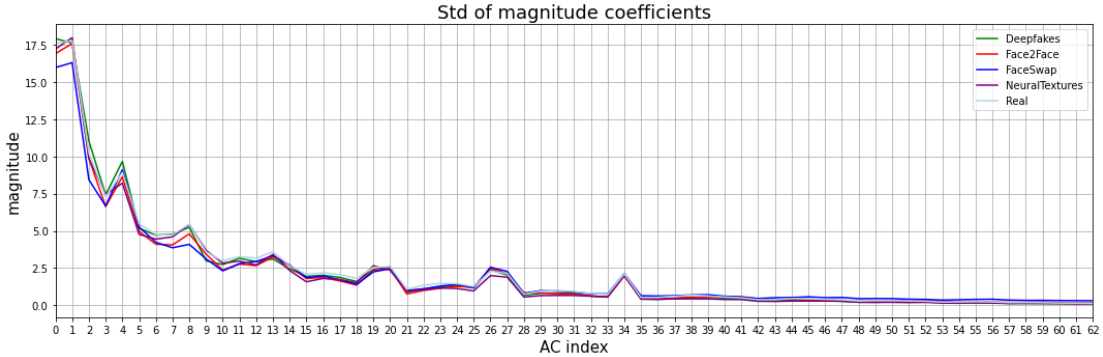
1. 截臉

   此篇實驗所使用的資料集主要是使用換臉技術，言下之意是除了人臉以外的頭髮、脖子、服飾及背景等等都是原圖，而兩篇參考文獻的方法皆是操作在完全由模型生成的偽造圖片上，因為訓練資料上性質的不同，先使用截臉技術進行局部擷取、取特徵及模型訓練，有得到模型正確率的提升，特別在 MLP 當中正確率來到了 9 成。

2. DCT 特徵

   下圖是在 c40 資料集下各個不同類型圖片的 AC 係數平均數折線圖及標準差折線圖，由此我們可以發現只有在較低頻的地方才有些微的差異，其餘部分不同類型的圖片幾乎是重疊在一起的，或許因為圖像壓縮的關係，高頻的資訊較不充足導致無法明顯區分。



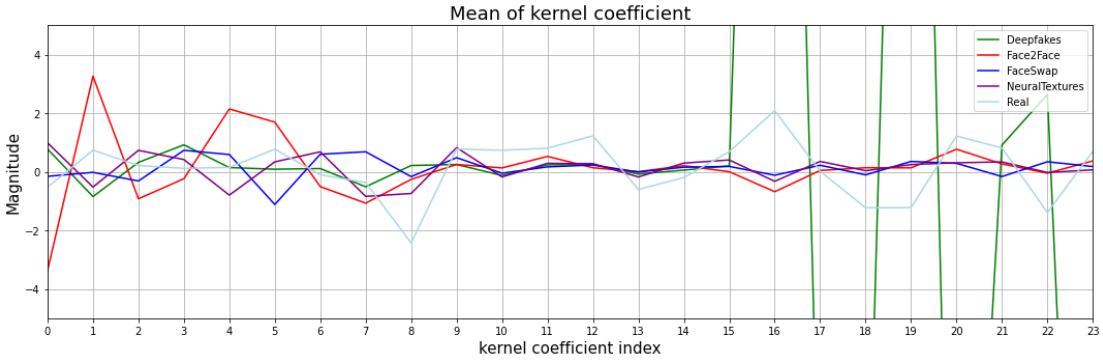


3. Conv-trace 特徵

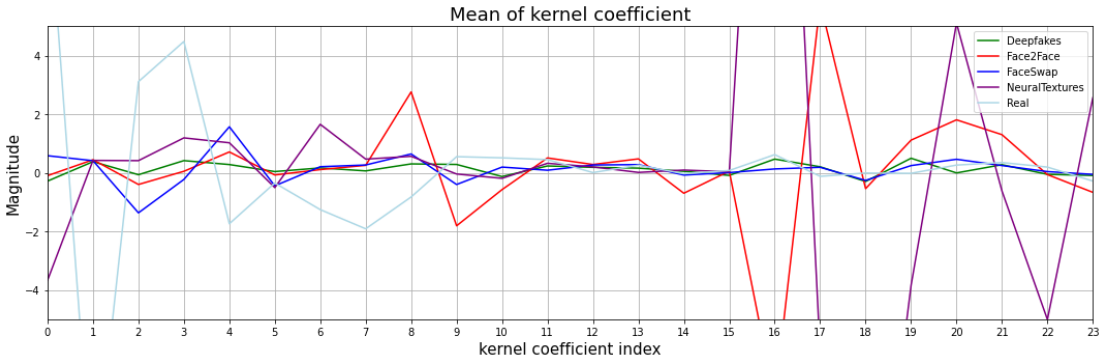
   在 Conv-trace 上，即使我們利用演算法為每一張圖片找到一組 kernel 後，也很難判斷 kernel 之間有特殊的趨勢，下圖分別為 raw 資料集與 c40 資料集每張圖在進行截臉操作再估計出的 3x3 kernel，各項係數的平均數及標準差，透

過下圖比較我們可以發現 raw 資料集與 c40 資料會是兩個不同的狀況，我們沒有辦法找到一個一致的特性來判斷不同偽造技術的係數分布。

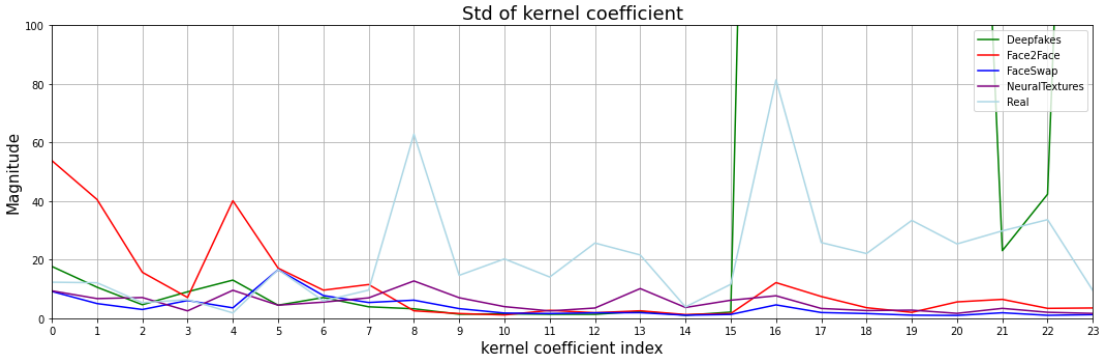
C40：



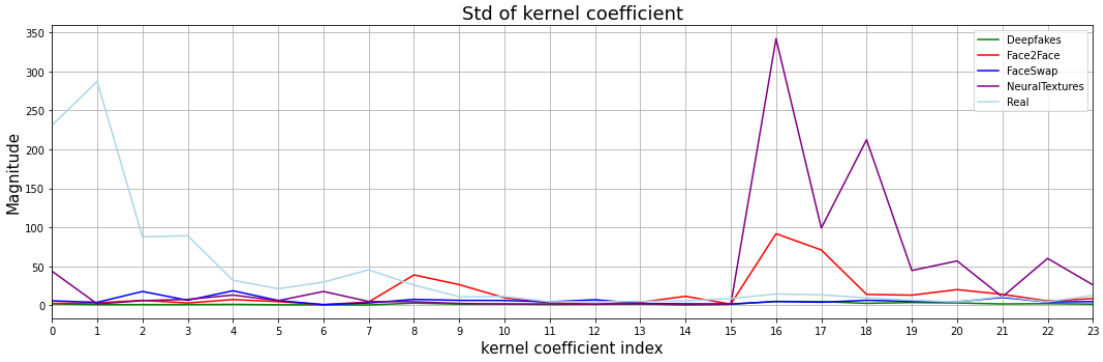
Raw：



C40：



Raw：



4. 羅吉斯迴歸

   在 raw 資料集當中有約 0.8 的預測正確率是不錯的表現，但在 c40 資料集下，我們實驗以 DCT 特徵訓練此模型需要調整相當大的迭代次數，由此推測在影像經由壓縮過後，各項頻率資訊的損失可能造成 DCT 頻率沒有顯著的不同，所以模型無法快速收斂、預測正確率也不佳。

5. 判斷能力

   下表我們進一步整理利用 MLP 模型對測試資料集不同類型的圖片類型的預測正確率，所有圖片皆有進行截臉操作，表格最左列代表該圖片類型，數值代表對該類型的圖片判斷正確的比率。特別需要注意的是訓練階段其實已經有考慮將資料筆數平衡，偽造影像數量與真實影像數量相同，但模型仍舊在真實影像上判斷不佳。

分類正確率	Raw	C40
Deepfake	96/132	91/132
Face2face	89/132	87/132
Faceswap	99/132	81/132
Neuraltexture	93/132	92/132
real	43/132	50/132

六、參考資料

1. Giudice, O., L. Guarnera, and S. Battiato, *Fighting deepfakes by detecting GAN DCT anomalies.* arXiv preprint arXiv:2101.09781, 2021.
2. Guarnera, L., O. Giudice, and S. Battiato. *Deepfake detection by analyzing convolutional traces*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020.
3. Rossler, A., et al. *Faceforensics++: Learning to detect manipulated facial images*. in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.