

# 2022 年臺灣國際科學展覽會 優勝作品專輯

作品編號	190028
參展科別	電腦科學與資訊工程
作品名稱	以隨機噪音生成技術為基礎的驗證碼對抗式攻擊防禦機制
得獎獎項	一等獎 青少年科學獎 美國ISEF正選代表
就讀學校	臺北市立建國高級中學
指導教師	紀博文
作者姓名	賴柏丰、魏路德
關鍵詞	<u>驗證碼、Adversarial Attack、Random noising</u>

## 作者簡介



我是就讀建國中學的賴柏丰，我在國三的時候為了要學個一技之長而學習 Python，後來也有在網路上看大學教授上機器學習的課程。之後被魏路德約去參加科教館舉辦的未來之星營隊，並且得到了參加這次科展的機會。在做科展的過程中，由於我並沒有實際上從頭開始寫一個深度學習的程式，因此遇到了非常多的問題。這次的科展也算是我第一次將虛擬碼實際寫成程式，非常的有成就感。

大家好我是就讀建國中學普通班的魏路德，我從國三升高一的暑假開始學習 C 語言，便對程式設計感到好奇，直到參加了科教館舉辦的 2021 未來之星智慧科技營隊，才看到程式語言困難與複雜的一面。我的興趣是運動，曾經在台北市教育盃拿下冠軍，但是高中的我已經卸下運動員身份，準備探索資訊領域如海洋般無邊無際的知識。

## 摘要

網路上常常會使用驗證碼(CAPTCHA)防止自動化程序取得網站資源，而一般而言，若驗證碼是可以輕易取得，十分容易被深度學習網路破解。然而，對抗式攻擊(adversarial attack)可以騙過許多深度學習網路。因此，本研究目的為建立能夠破解對抗式攻擊的深度學習網路。主要包含三個部分：建立 Captcha breaker、使用對抗式攻擊影響 breaker、防禦對抗式攻擊。Captcha breaker 的部份使用模擬的目標驗證碼作為訓練資料，以解決訓練資料不足以及人工標籤的問題；而破解 adversarial attack 會使用 adversarial training 以及 random noising 的技術進行。

## Abstract

CAPTCHAs are often used on the internet to prevent users from gaining website resources by Automated procedures. Generally, CAPTCHAs are easily to be captured, so they are also easy to be broken by deep learning. However, adversarial attack can attack many deep learnings readily. Therefore, our project is going to come up with a deep learning, being able to solve adversarial attack. Our project contain three parts : establishing a CAPTCHA breaker, effecting breaker with adversarial attack and defending against adversarial attack. The part of Captcha breaker uses simulated target verification code as training data to solve the problem of insufficient training data and manul labeling; while solving adversarial attacks will use adversarial training and random noising techniques.

## 壹、研究動機

隨著科技的進步與網路的普及，過去許多需要民眾花費大量時間取得的資源，已經透過電子化，讓民眾可以透過各種網站來取用。而民眾為了快速且有效率的取得這些資源，開始利用網路爬蟲到網站上擷取資料。但是大量爬蟲造訪網站會導致網站癱瘓、負荷不了，因此，許多網站開始使用 Captcha，即驗證碼，分辨造訪網站的是機器還是人，並阻擋前者。但是由於有些驗證碼顏色與背景相似，不僅不易辨識，還需花費數秒，這足足比爬蟲慢上許多。舉例來說，疫苗預約系統需要利用網路預約，其中有一步驟為「圖形驗證碼」(圖一)，它要求使用者分辨驗證碼圖形中的數字及字母。雖然此驗證碼不難辨識，但是有六個字元，人類從辨識圖形到輸入數字，完成預約步驟的時間已比機器人晚了將近 10 秒。在大家同時競爭疫苗施打地點與種類的時段，這可能會讓使用者偏好的施打地點及疫苗種類被他人登記完畢，甚至將影響使用者施打時間。若因手動登記延遲接種時間，或是因疫苗廠牌不正確或不適應而引發副作用，更需付出人命的代價。由此可知使用機器人自動辨識驗證碼仍有其無可否定的重要性，而此技術也值得更深入的研究。

在對抗式攻擊(adversarial attack)技術還沒出現之前，工程師會大量收集驗證碼作為訓練驗證碼判別模型的資料，網路爬蟲便能順利運作，但在此技術出現後，驗證碼辨識系統又陷入困難，因為對抗式攻擊透過將驗證碼圖片加入肉眼無法觀察到的雜訊，使驗證碼外觀完整，卻能使驗證碼破解器無法正確辨識。這激起了我們的想要研發辨識系統的決心，希望能使驗證碼辨識系統復活，也讓網路爬蟲發揮功能，繼續為不方便操作電腦或是有迫切登入網站需求的使用者代勞。



圖一、COVID-19 公費疫苗預約平台的驗證碼

## 貳、研究目的

1. 以驗證碼作為判別模型訓練資料，訓練驗證碼判別模型，已達到能辨識基本文字行驗證碼之效果。
2. 針對判別模型進行對抗式攻擊，使該驗證碼不易被一般的驗證碼破解器識別。
3. 探討 adversarial training 和 random noising 兩種技術對於防禦 adversarial attack 的效果。

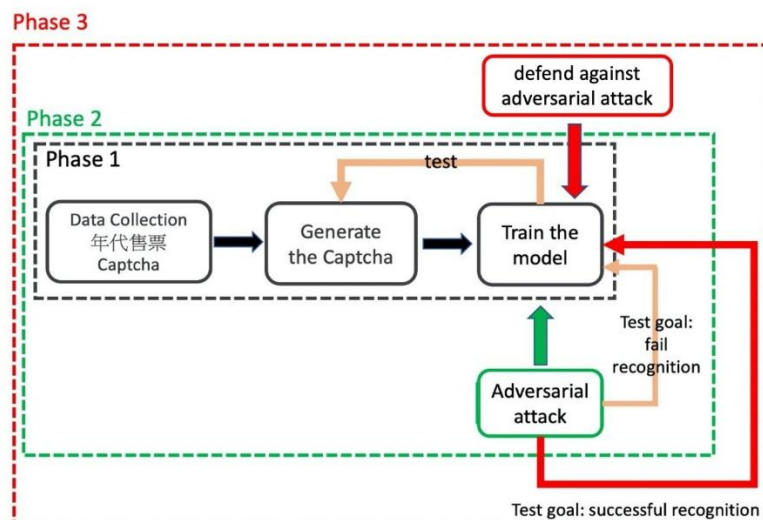
## 參、研究過程與方法

### 一、研究流程與架構

phase1 我們搜集年代售票網站的驗證碼，記錄其驗證碼特徵（如字元種類、大小、有無噪點等.....），並依照特徵生成相似的驗證碼，並以這些生成的驗證碼作為訓練深層神經網路判別模型的資料。訓練好模型後，我們會將先前生成好的驗證碼放入神經網路中測試其效果，如果神經網路成功辨識，則進入下一步驟。

phase2 我們將針對自己訓練的深層神經網路判別模型進行對抗式攻擊，並將被攻擊過的驗證碼放入神經網路測試。若神經網路成功的被誤導，辨識錯誤答案，我們將進行下一步。

phase3 我們會對判別模型加入一層隨機噪音層，並使用已被對抗式攻擊過的驗證碼圖像測試模型判斷結果。若模型判斷正確，則代表我們成功找出防禦對抗式攻擊的方法。



圖二、研究流程示意圖

## 二、研究設備與器材

### (一) 程式語言：python

使用模組：

#### 圖片生成與處理：

##### 1. PIL.Image：圖片生成與儲存

PIL(Pillow) 是 python 內建的影像處理套件，可用來做為基礎的繪圖工具，亦為將純數字矩陣轉換為圖片的工具之一。

##### 2. matplotlib：繪製圖表、顯示圖像

matplotlib 是 python 的繪圖庫，主要被用來做為數據視覺化與圖片編排的用途。

#### 數據處理與模型訓練：

##### 1. numpy：機器學習、影像處理的基礎套件

numpy 是一個 python 的擴充程式庫，支援高階大量的維度陣列與矩陣運算。

此外，基於精準度的問題，我們的 attacked image 會使用 numpy 內建的 npy 格式儲存。

##### 2. keras & tensorflow：機器學習

keras 屬於 tensorflow 的一種開放原始碼的高級 API，基於 python 高階深度學習的程式庫。

### (二) 訓練環境：google colab

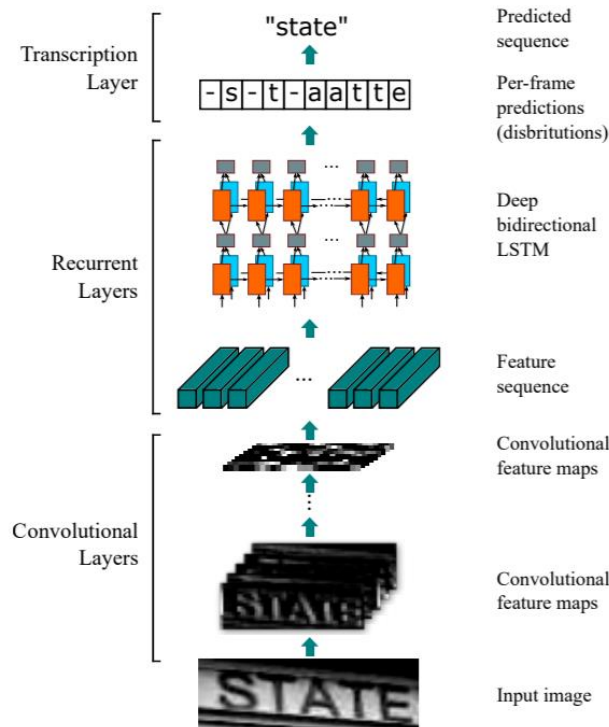
google colab 是 google 提供的一種基於 Jupyter Notebook 的線上 python 編寫工具，可以透過瀏覽器編寫程式，擁有免費的 gpu 可以使用，並且可以掛接雲端硬碟，獲取雲端的資料。而因為我們的電腦皆屬於文書機，沒有安裝 gpu，因此使用 google colab。

然而，由於我們使用的是 google colab 免費版，因此 google colab 會出現自動斷線、gpu 用量不夠的問題。因此，在進行實驗的時候我們會需要注意 gpu 的用量不能過多，否則被 google 自動斷線容易造成研究數據損失。

### 三、研究文獻探討

#### (一) 驗證碼判別模型--CRNN 模型(Convolutional Recurrent Neural Network)

CRNN 模型的概念為使用判斷圖片特徵的 CNN 模型，搭配上按時序切割進行判對的 RNN 模型，在兩模型組合處會將 CNN 模型的輸出轉換為 feature map，並輸入 RNN 模型，最終針對各個時間點計算 CTC Loss。[7][8]

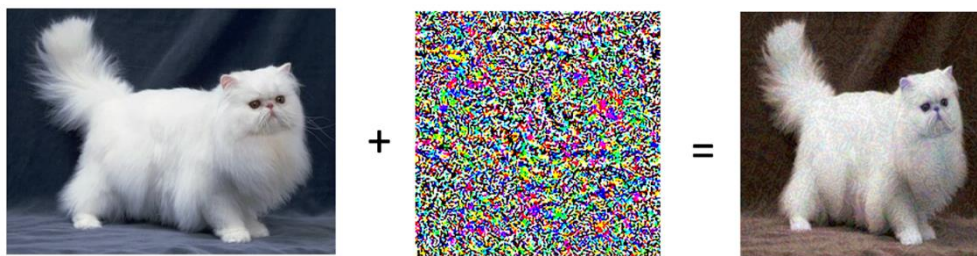


圖三、CNN 模型示意圖(擷取自[8])

#### (二) 對抗式攻擊(adversarial attack)：

對抗式攻擊是一種針對機器學習模型或深度學習網路，透過對於圖片加入人類難以觀察到的雜訊，引導模型對於該圖片產生錯誤的判斷。adversarial attack 主要分為白箱攻擊(White box attack)和黑箱攻擊(Black box attack)，其中，白箱攻擊是在攻擊者了解判別模型參數的情況下攻擊，黑箱攻擊則在不知參數的情況下，模擬目標模型，並對該模擬模型進行攻擊，其中以白箱攻擊的成功率較高。其攻擊方式其實非常難使人以肉眼察覺，因為其加入人的肉眼無法識別，機器卻會因此而誤判的雜訊，讓辨識模型無法正確辨識出樣本，因此在 AI 領域擁有一些用途。[5]

例如圖四中，將波斯貓的圖片打上雜訊，產出的圖片雖乍看之下仍為波斯貓，不過在模型 ResNet50 中判別結果卻是一種名為「西高地白梗」的狗。不過圖一中的雜訊為了讓人眼容易辨識，已被放大 100 倍，故肉眼仍能看出產出之圖片與原圖的些微差距。



圖四、對抗式攻擊示意圖

對抗式攻擊的攻擊手法，可用下列算式表示：

$\arg \max_{\mathbf{x}^{adv}} J(\mathbf{x}^{adv}, y)$  s.t.  $\|\mathbf{x}^{adv} - \mathbf{x}^{real}\|_{\infty} \leq \epsilon$ , 其中  $\epsilon$  為人眼難以辨識的最小範圍，我們攻擊的目的為：在對抗式樣本( $\mathbf{x}^{adv}$ )與真實樣本( $\mathbf{x}^{real}$ )視覺差距最小化的情形下，最大化其判斷答案與其真實樣本的誤差，故在算式中，我們可以看到  $J(\mathbf{x}^{adv}, y)$  為判斷答案與真實樣本的差距，而它是要被最大化(max)的。[1]

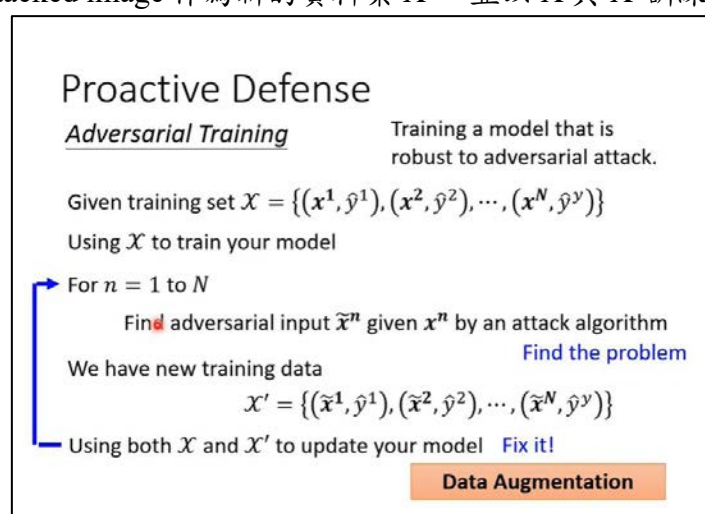
Adversarial attack 的詳細作法會在之後說明。

### (三)對抗式攻擊防禦機制

#### 1. 對抗式訓練(Adversarial training)：

Adversarial training 是一種主動防禦 adversarial attack 的方式，其詳細運行方式如下：

首先，使用原始的圖片資料集  $X$  訓練模型，之後使用 adversarial attack，生成相等數量的 attacked image 作為新的資料集  $X'$ ，並以  $X$  與  $X'$  訓練新的模型。[6]



圖五、adversarial training 虛擬碼(擷取自[6])

#### 2. 隨機製造雜訊(Random noising)：

此為一個主動防禦對抗式攻擊的方式，簡言之，其概念為：在模型每一層卷積層前面都加上 random noise layer，即在每次進行卷積前加上噪音，使模型在訓練時能夠學習如何處理或淡化噪音的影響，從而達到減緩 adversarial attack 的效果。

至於此技術的細節，則是使用 random self-ensemble (RSE)，讓 RSE 在每個積卷層(convolution layer)，包括訓練和測試階段，加入雜音層。Lecuyer 等人從差分隱私(Differential privacy)的角度來看隨機噪聲防禦機制，並提出了一種基於差分隱私的防禦，稱為 PixelDP，此防禦機制在 DNN 中加入 DP 雜音層，以其對輸入的預測中對分佈的變化施加 DP 界限，並具有小的、基於規範的擾動。DP 可以用來防禦 L1/L2 attacks 使用 Laplacian/Gaussian DP 機制。另外，有其他研究者被 DP 啟發，他為了排除對抗式攻擊的擾動影響，在樣本分類之前將對抗式樣本加入雜訊。[2][3]

#### 四、研究流程

##### (一) 模擬驗證碼

本研究選擇使用「年代售票」網站的驗證碼做為測試，主要因為年代售票的驗證碼同時具有「干擾線」、「噪點」、「空白背景」等三個驗證碼常見的特色，並且字數固定為4，字元的分布沒有重疊，分散的位置也較為均勻，對於驗證碼判別模型的訓練較為容易。

為了提供深度學習網路訓練資料，我們分析了年代售票驗證碼的特徵，如表一，並使用 Python 的 pillow 套件根據特徵進行模擬，效果如圖五所示。在驗證碼字元選擇的方面，由於驗證碼的文字通常會有一些變形，導致一些相像的字對於人類容易辨識錯誤，因此，大部分種類的驗證碼會將這些字元移除，在本研究中亦將這些字元移除，其中包括'O','O','Q','I','T','7'六種字元，剩餘 30 種字元。

我們會選擇使用人工生成驗證碼主要是因為有兩個好處：1. 進行機器學習的一大難點是獲得訓練資料，由於網站上的驗證碼的大量取得只能透過網路爬蟲的方式，然而使用網路爬蟲大量爬取網站資源會導致 IP 被網站封鎖。2. 在進行深度學習 classification 的訓練時，會需要同時為資料輸入標籤，而我們在訓練時會需要用到上千張驗證碼，逐一針對從網站上爬取的圖片進行人工標籤十分浪費時間。

年代售票	
顏色種類 (主要顏色)	隨機，高飽和度
顏色數量 (主要顏色)	4(四個文字各一)
尺寸	90x25 像素
扭曲	0
傾斜	[0,0]
傾斜角度相同	是
邊緣處理	無
大小(佔高比例)	0.5
高低變化(佔高比例)	[0,0.2]
字數範圍	4
字體	Calibri Light
用字範圍	數字、大寫英文字母
字元重疊程度	0
背景型態(干擾線；噪點)	[4,7]，隨機顏色；[20,40]，隨機顏色

表一、年代售票驗證碼特徵分析

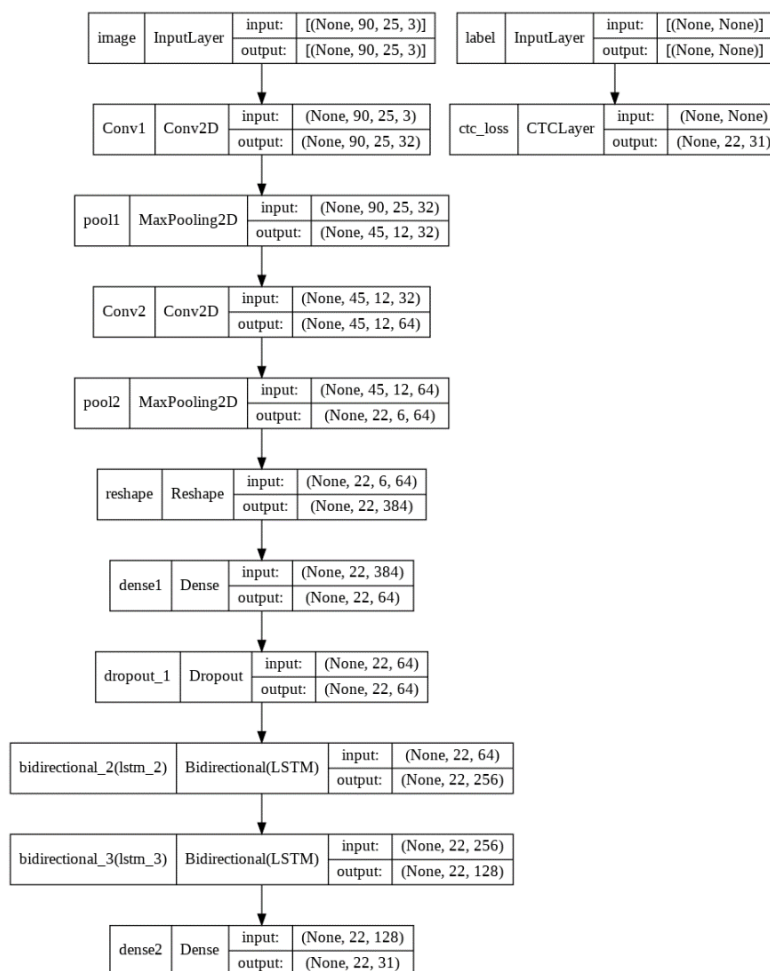


表二、年代售票驗證碼與模擬生成驗證碼對比

##### (二) 建立驗證碼判別模型

本研究使用 CNN+RNN 所組合成的深度學習網路作為驗證碼的判別模型，以下為此模型的結構圖





圖六、CRNN 模型結構圖(使用 keras 的 plot\_model 與 graphviz 生成)


另外，我們的模型有加上 EarlyStopping 的 Callbacks 參數，若 val\_loss 的值連續 15 輪沒有下降的話會強制結束訓練。

```
early_stopping_patience = 15
early_stopping = keras.callbacks.EarlyStopping(
    monitor="val_loss", patience=early_stopping_patience, restore_best_weights=True
)
```

圖七、Early stopping Callbacks 參數

### (三)針對驗證碼判別模型進行 adversarial attack

在訓練完判別驗證碼用的模型之後，我們可以使用 keras.backend 內建的 ctc\_batch\_cost 計算模型對於特定圖片、特定標籤的 loss 值，如同下圖所示。

圖片	
Loss - 8JCD	0.00576
Loss - 8BCD	16.12209


表三、CTC Loss 表示

首先，為求接下來對於 adversarial attack 描述的方便，我們定義了以下三種 label 型態以及三種 loss 型態：

- prediction：模型對於圖片判斷 loss 值最小的 label
- original label：captcha 圖片的正確答案
- target label：我們希望用 adversarial attack 將模型導向的目標
- predict loss：模型對於圖片判斷的最小 loss 值
- original loss：模型對於 original label 的 loss 值

- target loss：模型對於 target label 的 loss 值

舉例來說，若我們希望讓模型將一張'2DS2'(original label)的驗證碼判斷為'2DSU'(target label)，然而模型判斷結果為'2DSJ'(prediction)，則記錄方式如下表所示：

圖片	
predict loss – 2DSJ	0.36285
original loss – 2DS2	10.02384
target loss – 2DSU	5.03246

表四、自訂義 loss 型態演示

我們選擇 target label 的方式是在 original label 中隨機選擇位置，替換成其他隨機字元。由於 adversarial attack 的目的是為了使機器人對 captcha 判斷錯誤，這樣做既能節省資源也可以加快處理時間，同時也不失分辨網頁造訪者身分的功能。

我們會使用 adversarial attack，在圖片上加上不足以干擾人類辨識文字的雜訊，以達到引導模型將圖片判斷為特定標籤。然而，**adversarial attack 所生成的雜訊並非隨機生成**，而是依據模型的 loss 值進行 gradient descent。

Adversarial attack 詳細的作法為，我們會設定欲攻擊的圖片為  $x_o$ ，再生成一個與  $x_o$  相同形狀的前景  $x_f$  (此為 adversarial attack 的雜訊)，將  $x_o$  與  $x_f$  重疊後的圖像稱為  $x^{adv}$ ，使用 model 判斷  $x^{adv}$  對於 original label 以及 target label 的 loss 值，依照特定比例組合成

我們選擇設定 adversarial attack 的比重  $\alpha=0.8$ ，這樣可以讓程式注重在 target loss，而非持續地讓 original loss 下降，失去 target adversarial attack 的意義。另外，我們設定學習率  $\eta$  依照當前 target loss 值動態調整，如同下圖所示。

```
def lr_schedule(loss):
    if loss > 3:
        lr = 0.1
    elif loss > 1.5:
        lr = 5e-2
    elif loss > 0.7:
        lr = 1e-2
    elif loss > 0.3:
        lr = 5e-3
    else:
        lr = 1e-3
    return lr
```

圖八、adversarial attack 的 learning rate

而 adversarial attack 的最大變化值  $\epsilon$  與執行次數  $T$  則會依每次實驗的設計不同而更改。

---

**Algorithm** : Target adversarial attack

---

**Input** : original image, original label, target label, model

1. Let  $x_0, x_f =$  original image,  $\text{shape}(x_0)$  #  $x_f$  is an adversarial foreground
2. Let  $\text{label}_o, \text{label}_t =$  original label, target label
3. Initialize attack parameters  $\eta, \alpha, \epsilon, T$ ;
4. Let  $x^{\text{adv}}_0 = x_0 + x_f$
5. func calculate\_loss(pred, label):
6.     return loss(pred, label) #it depends on what kind of model it is
7. func Clip(array, eps):
8.     for x (0 to len(array)-1) do:
9.         if array[x]>eps:
10.             array[x] = eps
11.         else: pass
12.     return array
13. func gradient\_descent(loss, array):
14.     array =  $\partial \text{loss} / \partial \text{array}$
15.     return array
16. for t (0 to T-1) do:
17.      $\text{loss}_{\text{original}} = \text{calculate\_loss}(\text{model}(x^{\text{adv}}_t), \text{label}_o)$
18.      $\text{loss}_{\text{target}} = \text{calculate\_loss}(\text{model}(x^{\text{adv}}_t), \text{label}_t)$
19.      $\text{loss}_{\text{total}} = -(1-\alpha) * \text{loss}_{\text{original}} + \alpha * \text{loss}_{\text{target}}$
20.      $x_f = x_f + \eta * \text{gradient\_descent}(\text{loss}_{\text{total}}, x^{\text{adv}}_t)$
21.      $x^{\text{adv}}_{t+1} = x_0 + \text{Clip}(x_f, \epsilon)$

---

**Output** :  $x^{\text{adv}}_T$ , which is an adversarial attacked captcha image

---

圖九、adversarial attack 虛擬碼

## 肆、研究結果與討論

### 一、模型針對原始驗證碼圖片訓練結果

#### (一) 針對 captcha 圖片進行模型訓練

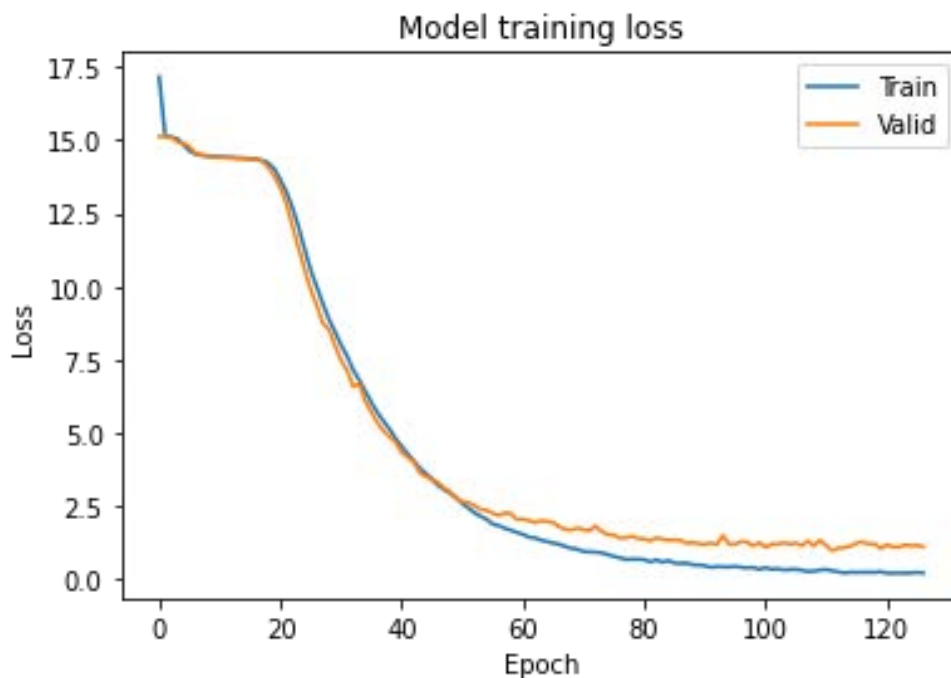
##### 1. 實驗目的

此步驟的目的為能夠有效辨識原始圖片的模型。

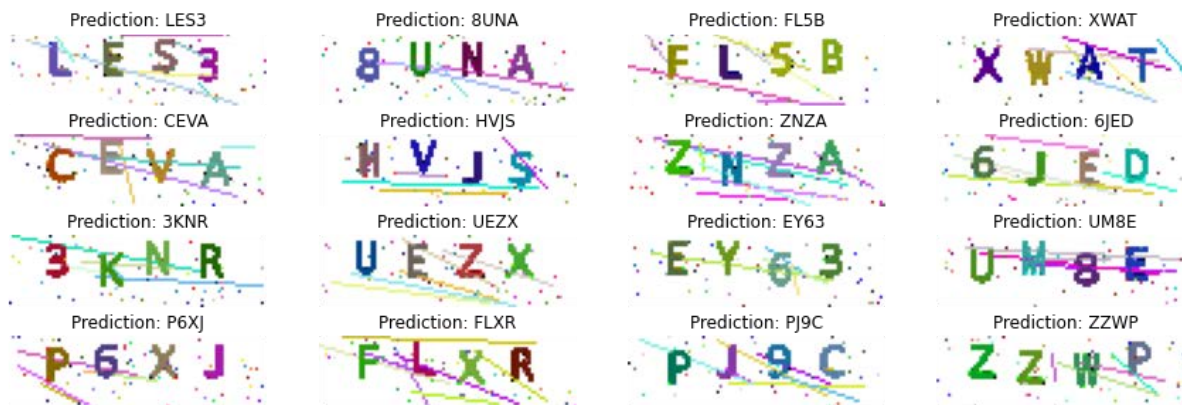
##### 2. 參數設定

模型如同上方參-三-(二)所描述，訓練資料集  $X=4000$ ，150 epochs

##### 3. 實驗結果



圖十、模型訓練 loss 值變化



圖十一、captcha 圖片與模型判斷對照表

#### 4. 結果討論

如同圖十所示，在訓練模型的過程中 train loss 和 validation loss 在 50 epoch 前差異不大，而到最後停止訓練時 validation loss 大概為 1.75，顯示略有 overfitting 的現象，然而因為驗證碼本身具有噪點、干擾線，我們認為這樣的結果影響並不大。

## 二、針對驗證碼判別模型進行 adversarial attack

### (一) 針對一張圖片進行 adversarial attack

#### 1. 實驗目的

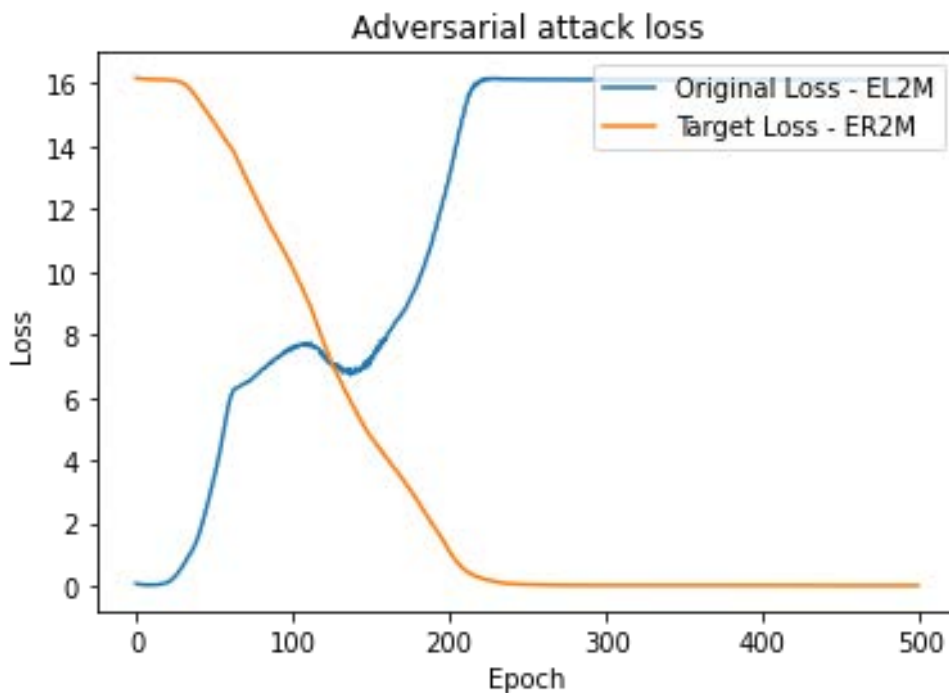
此實驗的目的為確認 adversarial attack 程式能夠在圖片上加上雜訊 (perturbation)，並正確的引導模型對圖片產生錯誤的判斷。在此實驗中發生的現象可能會因為模型、original image、target label 的不同而有所差異。我們會使用之前我們自己訓練的模型作為攻擊對象。

#### 2. 參數設定

$\epsilon = 25/500$ ， $T = 500$  epochs，original label = 'EL2M'，target label = 'ER2M'

#### 3. 實驗結果

模型對於 adversarial attacked image 的 loss 值如下圖所示：



圖十二、adversarial attack 的 loss 值變化

	original image	attacked image	perturbation
image			
original loss	0.06663	16.12094	X
target loss	16.18359	0.00444	X

表五、單張圖片 adversarial attack 效果圖

#### 4. 結果討論

如同上方圖表所示，大約在執行到 220 epoch 之後，模型對於 attacked image 的判斷就會從 original label，即'EL2M'，被引導至我們所設定的 target label，即'ER2M'。而比較兩種 loss 值的變化可以發現，target loss 有一直下降的趨勢，而 original loss 卻有一度有明顯下降，我們推測可能是因為我們設定比重  $\alpha=0.8$ ，因此 adversarial attack 的過程中會優先讓 target loss 下降，而非使 original loss 上升。

另外，比較表五中的 original image 和 attacked image 可以發現，兩張圖片之間雖然可以用肉眼看出些微的差距，然而並不會對文字辨識產生影響，這是受到了  $\epsilon$  的值的影響，這也是我們之後要探討的問題。

### (二) 探討最大變化值 $\epsilon$ 對於圖片視覺效果與模型判斷 loss 值的影響











#### 1. 實驗目的

此實驗的目的為探討最大變化值  $\epsilon$  對於圖片視覺效果與模型判斷 loss 值的影響。在 adversarial attack 中， $\epsilon$  的大小會影響 attacked image 與 original image 之間的差異，若  $\epsilon$  太大則可能會很大程度的影響人類辨識圖片，而  $\epsilon$  太小則可能導致 adversarial attack 效果不佳。因此，我們需要找到適當的  $\epsilon$  值。

## 2. 參數設定

T = 500 epochs，original label = '8JCD'，target label = '8BCD'

## 3. 實驗結果

epsilon	image	perturbation	original loss	target loss
0/255.		X	0.00576	16.12209
1/255.			0.00511	16.12050
5/255.			0.20685	15.45538
10/255.			14.61891	5.95429
25/255.			16.11742	0.00268
50/255.			16.11751	0.00224
100/255.			16.11760	0.00223
150/255.			16.11761	0.00224
200/255.			16.11760	0.00224
255/255.			16.11755	0.00218

表六、不同 epsilon 對 adversarial attack 的影響

## 4. 結果討論

根據上表可知，在  $\epsilon$  小於 10/255 時，adversarial attack 的效果並不明顯，主要是因為造成的擾動還不足以對模型的判斷產生巨大影響；而在  $\epsilon=10/255$  時，可以看到 original loss 已經升的非常高，然而 target loss 還沒有降到 1.0 以下，表示雖然 adversarial attack 已經可以看到成效，卻還沒有完全成功引導；而當  $\epsilon=25/255$  時，original loss 和 target loss 的值皆分別依預期的上升與下降。而之後的 perturbation、original loss 和 target loss 並沒有隨著  $\epsilon$  的上升而產生顯著的改變，可以推斷 adversarial attack 到  $\epsilon=25.255$  就幾乎停止了。

然而，考慮到 adversarial attack 可能會因為 target label 與模型的判斷相差過大而導致需要更多的擾動才能成功引導模型產生錯誤判斷，因此在後面的實驗我們將設定  $\epsilon$  為 50/255。

另外，雖然 adversarial attack 原先是希望能夠讓人類無法分辨其中的雜訊，然而，我們認為驗證碼與一般圖片不同，本身就會加入一定程度的雜訊，加入這種不足以影響辨識的雜訊並不會有太多影響。

### (三) 探討 loss 值影響模型判斷圖片的程度

#### 1. 實驗目的

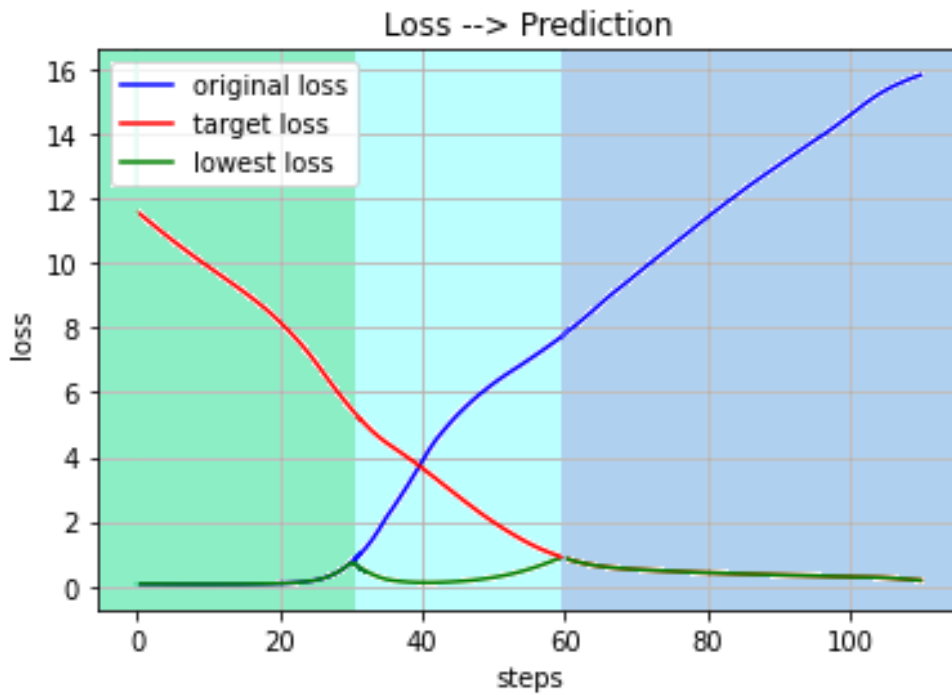
因為我們之後的實驗需要大量生成 attacked image，會消耗大量的資源與時間，為了加快生成的速度，我們決定在 adversarial attack 的 target loss 低於一定值，具有足夠效果時，便停止訓練。

因此在此實驗我們要尋找足以影響模型對圖片的判斷的最大 loss 值。

#### 2. 參數設定

original label = 'JLRX'，target label = 'JL4X'

#### 3. 實驗結果



圖十三、loss 值與 prediction 的關係圖

圖表的背景顏色依照模型對 attacked image 的判斷結果劃分，如下表所示：

顏色			
prediction	JLRX	JLHX	JL4X
step	~31	32~60	61~
交界點 original loss	0.80526		7.97255
交界點 target loss	5.02955		1.14596

表七、圖表顏色對照表

#### 4. 結果討論

觀察圖十三中可以發現，大約在 40~60 epoch 時，target loss 的值已經低於 original loss，然而此時 target loss 值並非模型的最小 loss 值，因此 attacked image 不會被判斷成 target label。

此驗證碼必須要將 target loss 降低到低於 1.1，而我們認為很可能會有其他測試資料需要更低的 loss 值才能成功 adversarial attack，因此我們將之後實驗的 maxloss 值設定為 0.5。

(四)探討需要多少 epoch 才能使 target loss 降到特定值(maxloss)以下

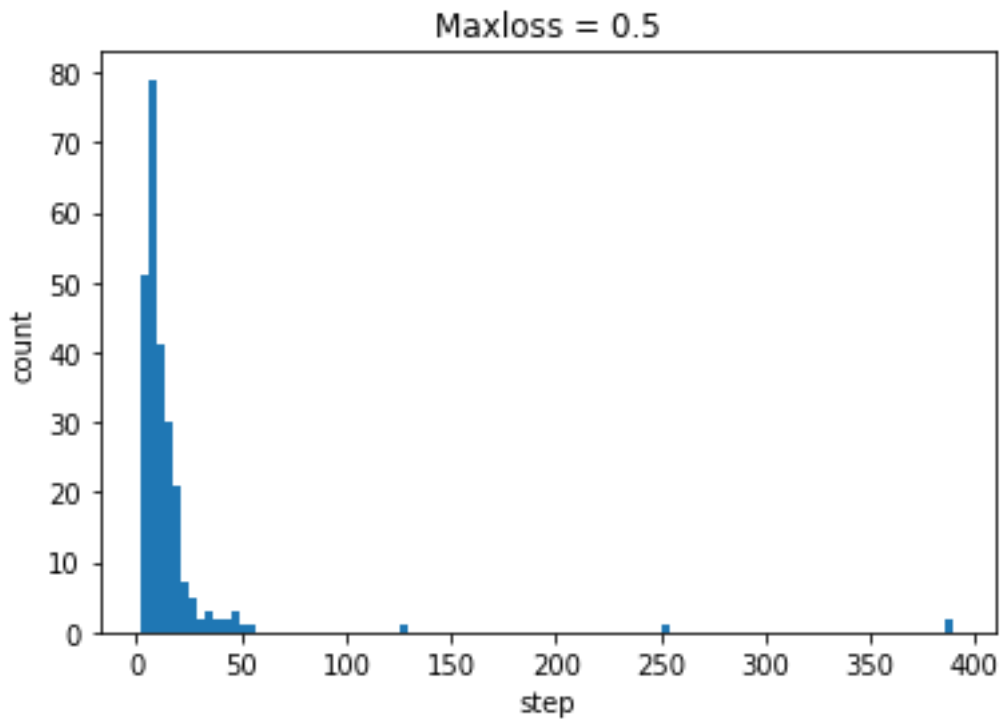
##### 1. 實驗目的

此實驗的目的為測試 adversarial attack 在設定了 maxloss 的情況下所需要使用的執行次數，以用來估計之後生成 adversarial training 資料集 X' 所需要的時間。

##### 2. 參數設定

$\epsilon=50/255$ ，測試資料集大小=200，maxloss=0.5

##### 3. 實驗結果



圖十四、到達 target loss = 0.5 所需的執行次數

#### 4. 結果討論

觀察圖十四可以發現，大部分情況下都只要 100 epoch 以內就可以完成攻擊，而也只有不到十筆資料需要使用到 100 個 epoch 以上。因此不會有資源過度使用的問題，我們可以使 adversarial training 資料集  $X'$  的大小與  $X$  相同，為 4000 張。



### 三、尋找對抗 adversarial attack 的方法

(一)使用 adversarial training 降低 adversarial attack 的效果

#### 1. 實驗目的

在此階段我們會使用先前使用的原始資料集 X 以及使用 adversarial attack 生成的資料集 X' 作為訓練資料，訓練判別模型。

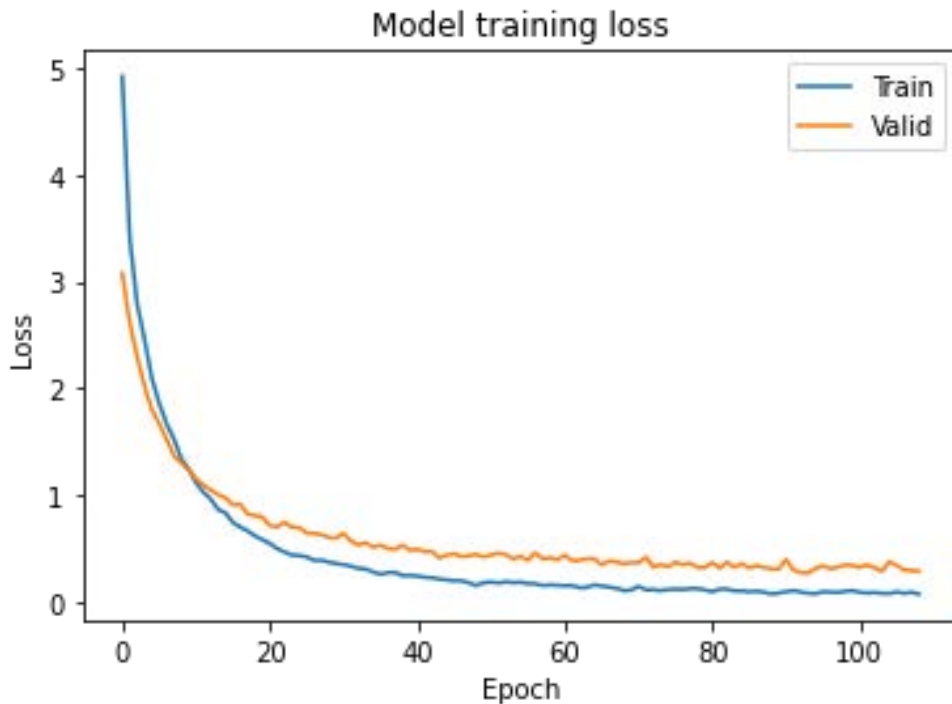
#### 2. 參數設定

attacked image :  $\epsilon=50/255$  ,  $\text{maxloss}=0.5$

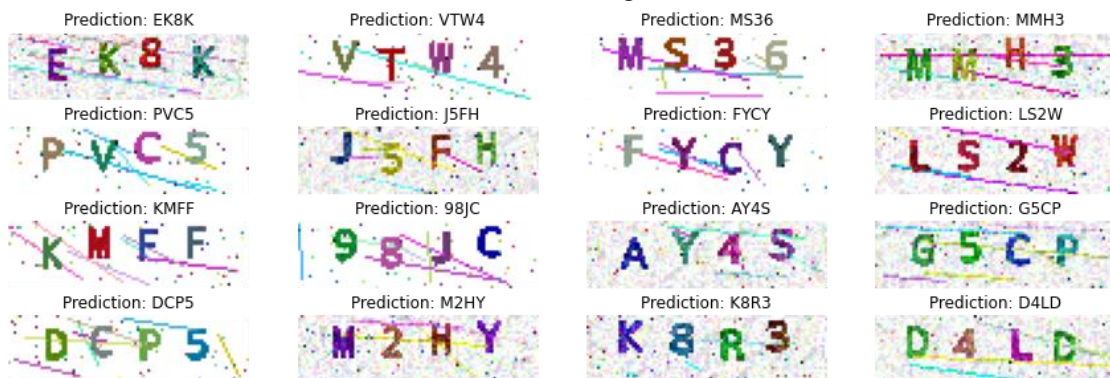
training : 訓練資料集  $X=4000$  ,  $X'=4000$  , 150 epochs

我們這裡會使用在 肆 - 一 訓練出的模型繼續訓練，因為使用的資料集有一半 (X) 是完全一模一樣的，另一半 (X') 也是 X 加上雜訊，差異並不大。

#### 3. 實驗結果



圖十五、adversarial training 的 loss 值變化



圖十六、captcha 圖片與模型判斷對照表

#### 4. 結果討論




由圖表可知，一開始訓練出來的的模型對於這個新的資料集判斷效果並沒有很好，然而在訓練到大約第 10 epoch 後 loss 值都降到了 1 以下，最終 train loss 和 validation loss 分別為 0.1 和 0.5。

而觀察圖十六可以得知模型在判斷 attacked image 和原始圖片上的效果相當接近。

## 伍、研究過程中的問題

### 一、圖片儲存問題

由於 adversarial attack 是以極為精細的雜訊對模型進行干擾，而目前常見的圖片儲存方式 JPEG、PNG 都會對圖片進行壓縮，因此，將 attacked image 轉換為圖片會導致細節損失，從而導致 attack 的效果降低，如下表所示：

		成功引導	造成干擾	失敗
圖片				
predict	prediction	22AR	LAJ5	2DF8
	loss	0.05740	0.90172	0.47412
original	label	224R	2AJ5	2DF8
	loss	12.13469	5.78667	0.47412
target	label	22AR	MAJ5	2D48
	loss	0.05740	5.80375	10.16371
比例		0.1	0.8	0.1






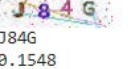
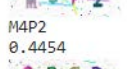



表八、attacked image 在壓縮成圖檔後

由上方表格可以得知，大部分的情況下 adversarial attack 在損失部分細節後依然能夠干擾模型進行判斷，然而這樣並非我們希望的效果。

因此，在儲存 attacked image 時，我們會使用 python 的 numpy 模組內建的 npy 格式儲存，以免造成細節的損失。

### 二、adversarial attack 的可轉移性問題

adversarial attack 是一種對圖片造成干擾，使模型對於該張圖片判斷錯誤。然而，若不是面對該模型的話，則很有可能導致原先的攻擊失效，如下圖所示：

模型 A				模型 B			
	24J6	24J6	0.4546		24J6	Z4J6	0.1558
	24J6	Z4J6	1.6539		24J6	Z4J6	9.7167
	558L	558X	0.4996		558L	558X	0.9238
	558L	558X	7.6872		558L	558X	7.4982
	9EG3	9EG9	0.3944		9EG3	9EG9	0.0377
	9EG3	9EG9	1.7016		9EG3	9EG9	8.4468
	CBYM	C4YM	0.4191		CBYM	C4YM	0.1226
	CBYM	C4YM	13.5332		CBYM	C4YM	14.6379
	F89Z	G89Z	0.4607		F89Z	G89Z	0.0577
	F89Z	G89Z	1.3380		F89Z	G89Z	12.4770
	J84G	JJ4G	0.4909		J84G	JJ4G	0.1548
	J84G	JJ4G	15.9821		J84G	JJ4G	20.7542
	M3P2	M4P2	0.4454		M3P2	M4P2	0.0258
	M3P2	M4P2	15.5152		M3P2	M4P2	13.9852
	R2SD	82SD	0.4812		R2SD	82SD	0.0690
	R2SD	82SD	9.7625		R2SD	82SD	13.5378
	U8SX	UESX	0.4866		U8SX	UESX	0.0371
	U8SX	UESX	3.6067		U8SX	UESX	9.4495
	X3BH	XDBH	0.4660		X5BH	XDBH	0.3290
	X3BH	XDBH	15.1282		X3BH	XDBH	3.2197
			0.4660				16.1038

表九、相同訓練參數的模型針對模型 A 進行攻擊的圖片判斷效果

上方圖片中，由於版面問題，我們並沒有將各個數值的意義標出，而上方表格中圖片中的意義如同下表所示。

圖片		
prediction	original label	target label
predict loss	original loss	target loss

一般而言，adversarial attack 會使用 ensemble 的方式進行，即透過同時攻擊多個模型來強化的效果，以達到成功誤導其他未被攻擊的模型的效果。後續我們會嘗試使用 ensemble 或是其他技術來加強 adversarial attack。

## 陸、結論

本研究透過模擬網站的驗證碼作為深度學習網路的訓練資料，訓練針對該驗證碼的判別模型，並成功的使用 adversarial attack 引導該模型對特定圖片做出特定的判斷，最終也實測了 adversarial training 的效果。

我們成功的使用 CRNN 模型訓練出對於我們自己生成的驗證碼的判別模型，雖然有些許 overfitting 的現象，然而整體對於驗證碼的準確率不低於 0.9。至於 adversarial attack 的部分，對圖片增加雜訊的最大變化值在  $\epsilon=50/255$  以上就足以非常成功的引導模型產生錯誤的判斷，而在設定 maxloss=0.5 的情況下，也能夠快速精準的進行 attack。然而，目前我們的研究中，adversarial attack 有三個問題：1. 目前只能針對特定模型做 adversarial attack，如果遇到其他模型效果會很差；2. 由於 adversarial attack 在轉換成圖片時被壓縮，就會損失很多細節，就會失去一定效果；3. 我們使用的 adversarial attack 技術若要部署到擁有較少資源的網站上的話，需要使用其他 adversarial attack 的演算法將 attack 所需的時間和資源再降低。

對抗式攻擊以及針對其的防禦之間的關係與近期討論度高的對抗式生成網路(Generative Adversarial Network) 十分相似，提升防禦技術的同時也能夠再給予對抗式攻擊新的目標模型，進一步提升對抗式攻擊的強度。因此，對抗式攻擊與針對其的防禦方法算是相輔相成的。

本研究的成果亦可應用在現實生活中，例如，經過 adversarial attack 的驗證碼可以進一步提供網站辨別機器人的方法。網站可以透過使用 adversarial attack，生成一張驗證碼，並引導驗證碼判別模型產生完全不可能會被真人寫出的答案，例如將 '2ACL' 轉換為 'KLMN'，而網頁端就可以在網頁造訪者輸入 'KLMN' 時直接將該網路 IP 封鎖，阻止其後續的行動。



圖十七、adversarial attack 正常輸入與阻斷 IP 示意圖

## 柒、未來展望

### 一、adversarial attack 上的加強：

在我們研究中，adversarial attack 有著低可轉移性、難以存取、生成速度緩慢的問題，我們希望可以在後續的研究中繼續進行改善

### 二、adversarial attack 防禦機制：

學術界中有很多用來防禦 adversarial attack 的機制，例如 random noising、模糊化等等，我們希望在未來可以多加嘗試、比較不同方法間的效果，並嘗試以 adversarial attack 攻破。

### 三、實做以 adversarial attack 驗證碼為基礎的阻擋爬蟲機制：

在現今的網站中有非常多種類的驗證碼，然而，為了防止機器人辨識驗證碼，驗證碼做得越來越扭曲，反而使人類無法辨識。使用對抗式攻擊製作驗證碼可以同時做到人類肉眼易辨識以及在結論中所提及，透過引導錯誤答案封鎖機器人 IP 位置。

## 捌、參考資料

- [1]Shao, R., Shi, Z., Yi, J., Chen, P. Y., & Hsieh, C. J. (2021). Robust Text CAPTCHAs Using Adversarial Examples. arXiv preprint arXiv:2101.02483.
- [2]Kui, R., Zheng, T., Qin, Z., Liu, X. (2020). Adversarial Attacks and Defenses in Deep Learning. Engineering, 6, 346-360.
- [3]Liu, X., Cheng, M., Zhang, H., Hsieh, C. (2018). Towards Robust Neural Networks via Random Self-ensemble. arXiv:1712.00673
- [4] 陳新，朱致伶(2020)。利用深度學習改善自拍人像構圖。2020 台灣國際科學展覽會第 60 屆。
- [5] Hung-yi Lee(2021)。【機器學習 2021】來自人類的惡意攻擊 (Adversarial Attack) (上) – 基本概念 [Video file]。Retrieved from <https://www.youtube.com/watch?v=xGQKhbjrFRk>
- [6] Hung-yi Lee(2021)。【機器學習 2021】來自人類的惡意攻擊 (Adversarial Attack) (下) – 類神經網路能否躲過人類深不見底的惡意？ [Video file]。Retrieved from <https://www.youtube.com/watch?v=z-Q9ia5H2Ig>
- [7] OCR：CRNN+CTC 開源加詳細解析。Retrieved from <https://www.ycc.idv.tw/crnn-ctc.html>
- [8] Shi, B., Bai, X., & Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE transactions on pattern analysis and machine intelligence, 39(11), 2298-2304.

## 【評語】 190028

此作品以隨機噪音生成技術為基礎的驗證碼來對抗攻擊防禦機制。此作品研究立意創新，應用層面廣，分析內容豐富且周詳。在複審中，參賽者表現優秀，論述與表達清楚，針對評審們的疑問都可以有效地回答，展現其對研究內容與挑戰的深度了解。鼓勵參賽者未來可在系統與實驗持續進步。