

## 2019 大專校院軟體創作競賽作品設計測試文件

### 一、 系統名稱

AIDog—校園犬 AI 辨識系統

### 二、 系統目的與範圍

大學校園多屬開放性校園，大部分的大學校園或多或少皆有流浪犬的出現。與其採取被動性驅趕的消極措施，有不少校園會採取「以狗制狗」的策略，留下、照顧留在學校已較長期、較親人的犬隻，使牠們建立地域性，驅趕新進的流浪犬，同時也期望能建立一個「生命平等」的教學環境。但校園犬的管理實為一件不易之事，在部分幅員廣大的校園，校園犬的數量甚至多到眼花撩亂，想認識校園犬的新生或是同學們也因為毛色類似很容易誤認。所以我們團隊設計了這個系統，希望藉由人工智慧來提供一認識校狗的便利方式。

**本系統的主要目的：**

- (1)在校園內辨識校園犬及得到新通報新的校園犬加以命名辨識。
- (2)幫助使用者可以易於認識、辨認校園犬。
- (3)更提供管理者一個資訊化、便利的數據統計方式。

#### 1. 系統非功能需求

非功能需求編號	非功能需求描述
AIDOG-NF-001 快速回傳結果	<b>快速顯示出結果:</b> 當相機移動至目標之時，辨識計算結果應該要在 300 毫秒內回傳至螢幕上，不拖延使用者使用的時間。
AIDOG-NF-002 清楚明瞭	<b>辨識結果簡單易懂:</b> 不給予不重要資訊，將辨識符合度最高前三名顯示出來即可，不須顯示過多複雜且困難的背景資料。
AIDOG-NF-003 辨識處理	<b>極速的辨識過程:</b> TensorFlow 辨識需在 100 毫秒內完成，經過模組的辨識。
AIDOG-NF-004	<b>處理資料快速:</b> OpenCV 去背需在 2 秒內完成，並且會在伺服器上完成，不影響使用者的體驗性。

快速去背	
AIDOG-NF-005 模組大小	<b>模組在軟體佔用空間小:</b> 不能使用 tensorflow 輸出的巨型 pb 檔，在盡量不壓縮辨識品質的情況下將較小的辨識檔案放入測試軟體中。
AIDOG-NF-006 結果展示	<b>最高的辨識結果清晰看到:</b> 在輸出結果的 text 框框，最有可能的結果會顯示最大。
AIDOG-NF-007 通報處理	<b>可以同時傳多個訊息:</b> 許多人同時傳送緊急通報訊息時，伺服器可以承載的住。
AIDOG-NF-008 相機開啟	<b>啟動應用程式時同時開啟相機:</b> 在開啟程式並在使用者允許授權拍照權限後，迅速的啟動使用者裝置的相機。

## 2. 系統功能需求

功能需求編號	功能需求描述
AIDOG-F-001 開啟相機	<b>相機啟動時開啟：</b> 在開啟應用程式時，就會詢問使用者是否授權程式使用相機，如果允許了權限就可以馬上開啟。
AIDOG-F-002 相機掃描	<b>相機啟動後掃描物體:</b> 讓使用者更直觀的使用此應用程式，使用掃描不使用拍照的方式讓使用者能更快的知道結果。
AIDOG-F-003 偵測	<b>偵測物體：</b> 使用者可以及時將相機隊到的目標物體和資料庫內的特徵比對來辨識該物體
AIDOG-F-004 辨識	<b>辨識物體：</b> 將使用者瞄準的目標凍結並輸入到模組裡面，經過 mobilenetV1 的辨識來比對特徵。
AIDOG-F-005 回傳結果	<b>回傳辨識結果:</b> 將辨識結果以 text 方式，顯示在使用者的應用程式介面上，並回報最高機率及第二和第三高機率的可能結果，如果辨識出不是狗的話回傳接近 0 的解答，也會主動建議使用者向管理端回報一些突發狀況。
AIDOG-F-006	<b>側邊欄：</b> 可以做出側滑的動作來開啟另外一個介面，側邊會根據使用者拉取的速度來顯示速度。

側滑介面	
AIDOG-F-007 事件通報	<b>事件通報</b> ：能讓使用者以文字、照片等方式通知管理者校園犬有異常的行為，會提供咬人、狗被欺負等常見的問題可以快速選擇來通報。也會在系統辨識到陌生犬隻時主動建議使用者向管理者通報。

一般性的系統功能操作使用案例(Use case, UC)之劇本(Scenario)描述如下。

使用案例編號：AIDOG-UC001	使用案例名稱：權限問題
系統反應動作	使用者操作動作
a.顯示是否給予該應用程式拍照權限	
	b.選擇允許該應用程式獲取權限
c.系統自動啟動相機來感知物體	
	d.使用者看到相機所呈現物體

使用案例編號：AIDOG-UC002	使用案例名稱：瞄準資料庫內狗隻
系統反應動作	使用者操作動作
	a.將相機對準已存在資料庫的目標狗隻。
b.在系統 tflite 辨識模組內凍結目標畫面。	
c.轉換成可以輸入進模組的格式。	
d.進入 mobilenetV1 模組內辨識。	
e.輸出文字格式的結果。	
	f.在 text 框看見排名前三相似的狗隻結果。

使用案例編號：AIDOG-UC003	使用案例名稱：瞄準非資料庫內狗隻
系統反應動作	使用者操作動作
	a.將相機對準未存在於資料庫的狗隻。
b.在系統 tflite 辨識模組內凍結目標畫面。	
c.轉換成可以輸入進模組的格式。	

d.進入 mobilenetV1 模組內辨識。	
f.以圖片特徵來分析與資料庫內狗隻何者最相似。	
	g.在 text 框看見排名前三相似的狗隻結果，但相似度可能非常低。

<b>使用案例編號：AIDOG-UC004</b>	<b>使用案例名稱：瞄準人類或其他物體</b>
<b>系統反應動作</b>	<b>使用者操作動作</b>
	a.將相機對準人類或是其他動物甚至是物體。
b.在系統 tflite 辨識模組內凍結目標畫面。	
c.轉換成可以輸入進模組的格式。	
d.進入 mobilenetV1 模組內辨識。	
f.以圖片特徵來分析與資料庫內狗隻何者最相似。	
	g.在 text 框看見排名前三相似的狗隻結果，但相似度可能非常低。

<b>使用案例編號：AIDOG-UC005</b>	<b>使用案例名稱：使用者側邊欄</b>
<b>系統反應動作</b>	<b>使用者操作動作</b>
	a. 使用者滑出側邊欄。
b. 以拉動速度顯示側邊欄介面。	
	c. 使用者看到緊急通報介面。

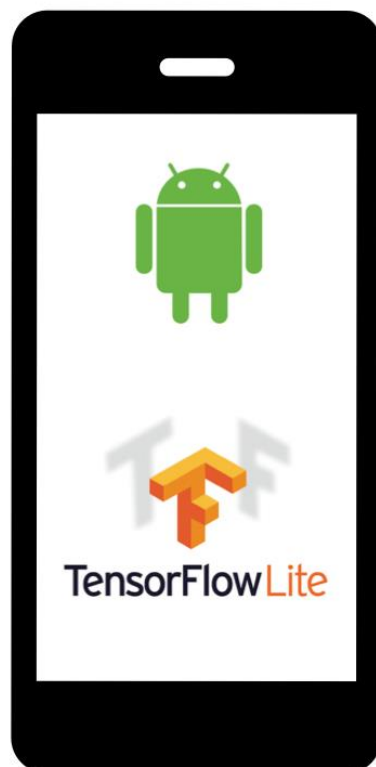
<b>使用案例編號：AIDOG-UC006</b>	<b>使用案例名稱：使用者關閉側邊欄</b>
<b>系統反應動作</b>	<b>使用者操作動作</b>
	a. 使用者將側邊欄滑回原始位置。
b. 以拉動速度將側邊欄介面隱藏回測邊中。	
c. 系統顯示原始相機介面。	
	d. 使用者看見相機介面。

<b>使用案例編號：AIDOG-UC007</b>	<b>使用案例名稱：使用者緊急通報</b>
<b>系統反應動作</b>	<b>使用者操作動作</b>
a. 顯示側邊欄緊急通報頁面	
	b. 使用者輸入緊急通報文字。

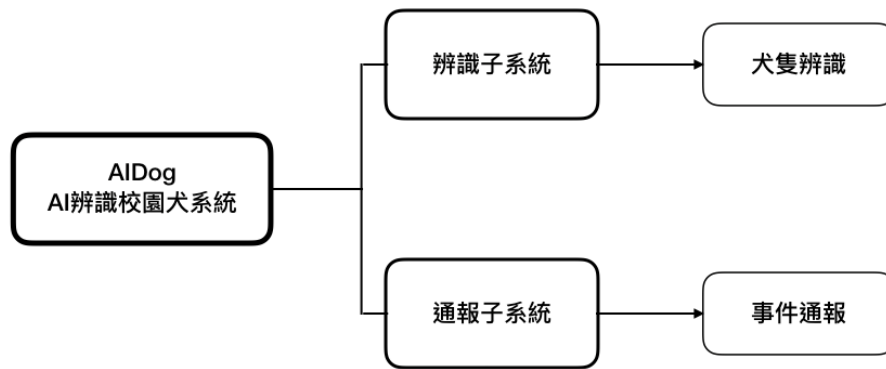
	c. 使用者按下傳輸按鈕
d. 系統傳送緊急通報事件給管理者帳號以及 email。	

使用案例編號：AIDOG-UC008	使用案例名稱：使用者通報未送出
系統反應動作	使用者操作動作
a. 顯示側邊欄緊急通報頁面	
	b. 使用者輸入緊急通報文字。
	c. 使用者未按下傳輸按鈕。
	d. 使用者將側邊欄縮小頁面。
e. 系統顯示相機畫面且 <u>不傳送</u> 通知。	

### 三、系統架構設計



▲圖一、系統架構



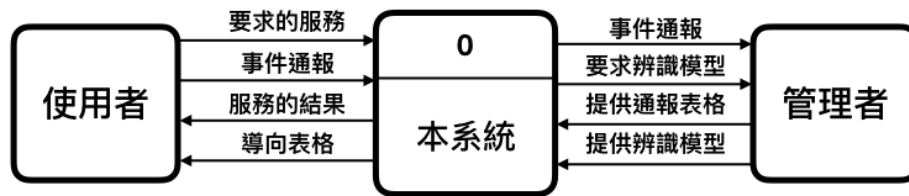
▲圖二、系統架構圖

系統架構圖說明：

辨識子系統：為本系統最重要也是最主要的功能，提供使用者利用相機拍攝之後進行辨識。

通報子系統：此子系統提供使用者向管理端進行事件通報，此子系統包含校園犬事件通報、陌生犬通報。

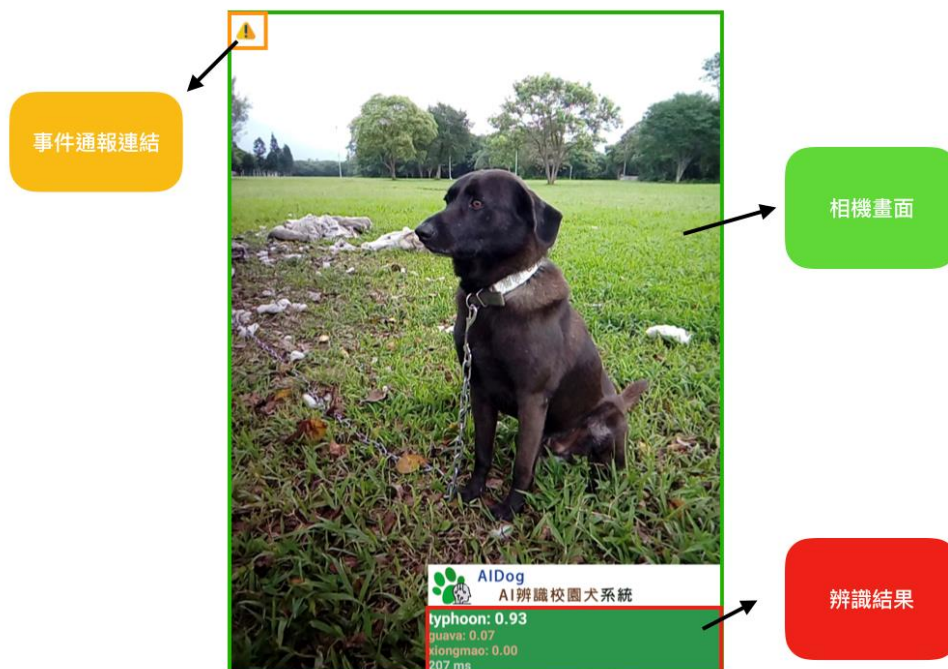
校園犬事件通報：此子功能提供使用者向進行校園犬異常行為(如：無故吠叫、追車、追人、咬人等)通報，使用者可使用相機攝影、相簿、文字輸入來提供事發當下狀態。



▲圖三、系統環境圖

#### 四、系統介面設計

開啟應用程式默認的第一畫面，此介面主要以相機為主，讓使用者可以在開啟應用程式時就馬上使用相機並達成辨識的目的。



▲圖四、系統主介面設計

本系統主介面如圖四，主要可以分為三個部分：

- (1) 相機辨識部分：在此部份是整個程式最主要的部分，顯示目前相機捕捉到的畫面。
- (2) 結果顯示部分：下方的文字框是顯示辨識結果的 text 物件，內容包括前三名的辨識結果，以及產生此辨識結果所花費的時間。
- (3) 通報部分：提供使用者事件通報的表單。

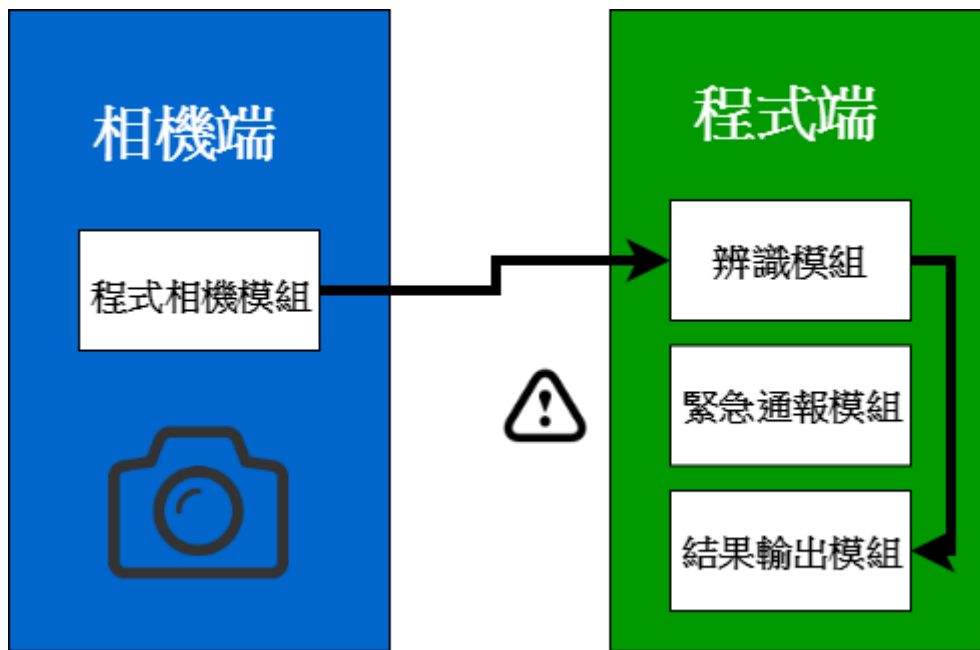


▲圖五、緊急通報側邊欄按鈕

## 五、軟體架構設計

圖為此專案的軟體架構，由兩大部分組成。





▲圖六、軟體架構

(1)相機端:

- a.以 Android 機器為主軸開啟相機。
- b.偵測手機鏡頭內目標，以凍結圖片的方式傳入模組

(2)手機端:

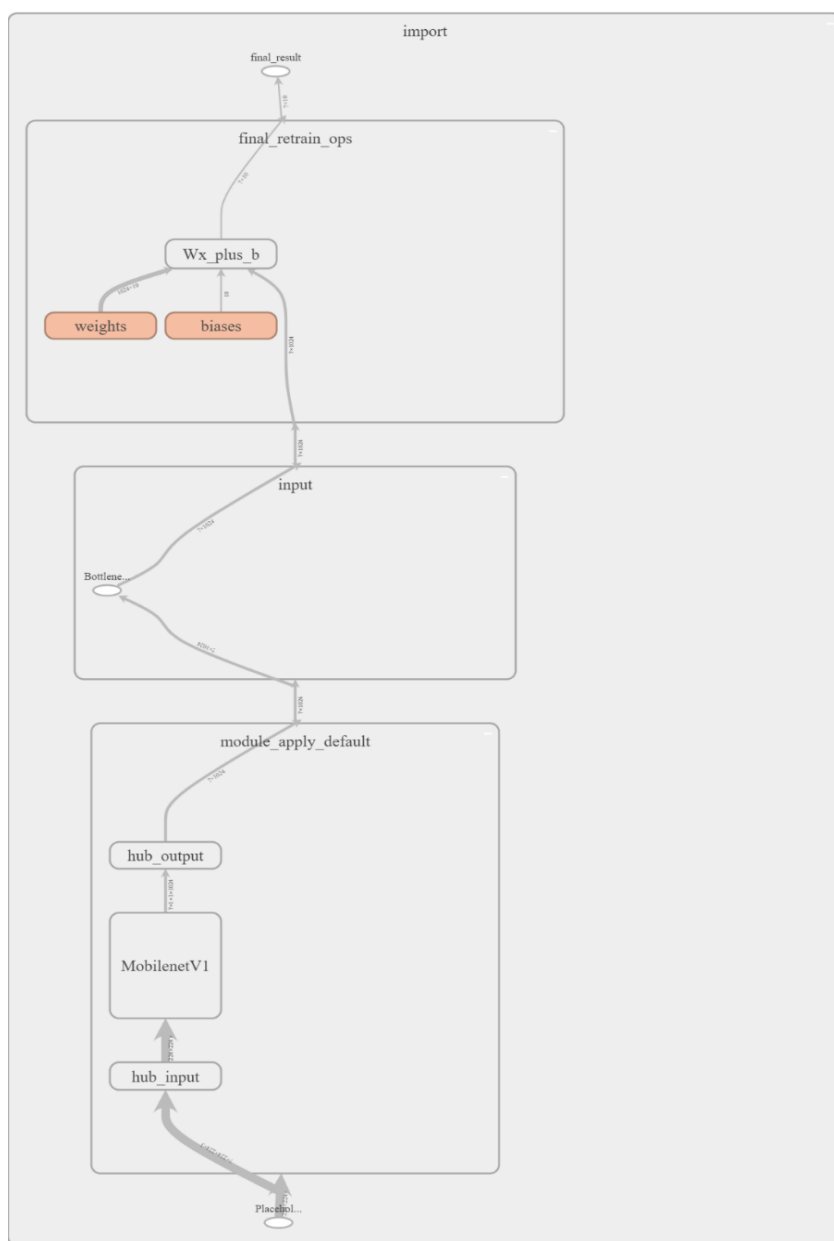
- a.tensorflow lite 讀取 Android 相機輸入的圖片檔案。
- b.將圖片以 text 輸出至下方 text 框。
- c.使用者可以點急緊通報按鈕來通報事件。

## 六、 軟體模組設計

(1)辨識模組

將由來源(--image)所讀取的圖片凍結並且輸入經由輕量神經網路 MobileNetV1 所設計的 module，如圖所示，並在模組內進

行特徵比對，並且分類，執行完後將分類結果來進行文字的輸出，輸出結果是以 1 為完全相似的基準。

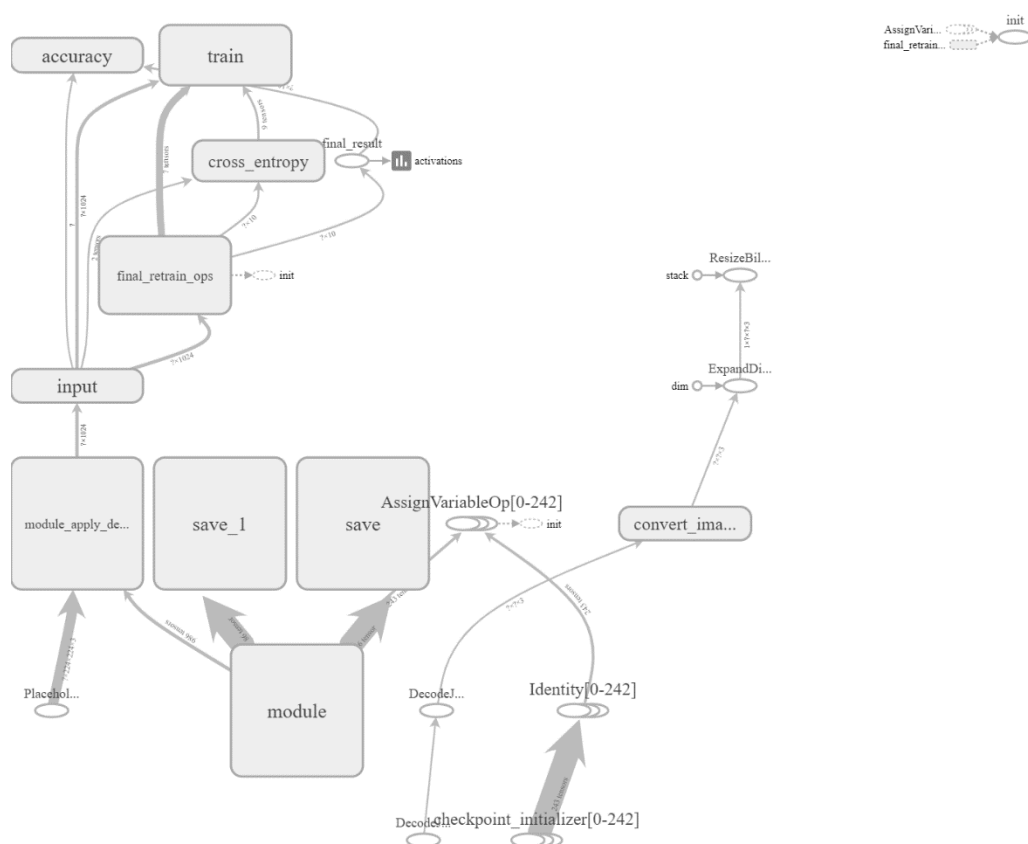


▲圖七、MobileNet 模型架構圖

## (2) 程式訓練模組

運用 google tensorflow 官網所提供的 hub 工具，以 mobileNet 的架構，和自己要訓練的狗隻資料，經過一定的處理之後，輸

入進程式裡面，再讓訓練重複多次，達到更精準的成績。



▲圖八、TensorFlow hub 訓練架構圖

### (3) 圖片處理去背模組

運用 OpenCV，將圖片內未指定的區域全部以黑幕罩住，只框出本體的部分，來讓訓練模組專注於圖片的本體。



▲圖九、OpenCV 去背範例圖

#### (4) 程式相機模組

運用 CameraActivity 執行開啟相機的工作。

#### (5) 程式通報模組。

使用一個緊急圖示來開起緊急通報的介面。

### 七、 開發環境

#### (1) Spyder



▲圖十、Spyder的Logo

[https://zh.wikipedia.org/wiki/Spyder#/media/File:Spyder\\_logo.svg](https://zh.wikipedia.org/wiki/Spyder#/media/File:Spyder_logo.svg)

本專案使用 Spyder 來做為 TensorFlow 訓練模組的環境，Spyder 是一個使用 Python 語言的開放原始碼跨平台科學運算整合開發環境。Spyder 整合了 NumPy，SciPy，Matplotlib 與 IPython，以及其他開源軟體，在介面方面擁有非常多功能，還有工作區和製作圖表的功能，非常符合本專案的需求。

## (2) Android studio



▲圖十一、Android Studio的Logo

[https://zh.wikipedia.org/wiki/Android\\_Studio#/media/File:Android\\_Studio\\_icon.svg](https://zh.wikipedia.org/wiki/Android_Studio#/media/File:Android_Studio_icon.svg)

**Android Studio** 是一個為 Android 平台開發程式的整合式開發環境。2013 年 5 月 16 日在 Google I/O 上發布，可供開發者免費使用。本專案是目的是讓使用者使用手機辨識犬隻，Android Studio 有簡單易懂的介面使開發移動端應用程式更為便利。

### (3) OpenCV



▲圖十二、OpenCV的Logo

[https://zh.wikipedia.org/wiki/File:OpenCV\\_Logo\\_with\\_text.png](https://zh.wikipedia.org/wiki/File:OpenCV_Logo_with_text.png)

OpenCV 全稱是 Open Source Computer Vision Library，是一個跨平台的電腦視覺庫。由英特爾公司發起並參與開發，大多用於開發實時的圖像處理、電腦視覺以及圖形辨識程式。在本專案中，我們將用於訓練機器模型時圖片批量去背，以及使用者上傳需辨識圖像時自動去背。

### (4) TensorFlow Lite

#### 1. TensorFlow



### ▲圖十三、TensorFlow的Logo

<https://zh.wikipedia.org/wiki/File:TensorFlowLogo.svg>

TensorFlow是一個開源軟體庫，用於各種感知和語言理解任務的機器學習。最初由Google Brain團隊開發，用於Google的研究和生產。其提供了Python、C++、Haskell、Java、Go和Rust等API。第三方包可用於C#、Julia、R和Scala。廣泛的應用程式使用TensorFlow作為基礎，其中它已成功實現自動化圖像字幕軟體。2015年10月26日，Google正式啟用了由TensorFlow提供支援的RankBrain，處理大量的搜尋查詢，替換和補充傳統的靜態演算法搜尋結果。

而TensorFlow lite是他的專案下的一個分支，在本專案中，我們將使用TensorFlow lite來做為嵌入Android的方案，該專案擁有著輕量級、跨平台、快速三個特點，很適合我們在行動裝置上使用。

## (5) TensorFlow Hub

TensorFlow Hub 是一個機器學習的函式庫，可以用於實現在不同任務下重複使用同種模型及架構，達到更加泛用的效果。本專案使用 TensorFlow Hub 來做遷移學習和重新訓練圖片。

## 八、 測試案例設計

### 1.應用程式測試

#### (1)辨識模組

1. TFlite model:測試動物圖片是否能更改顯示結果。
2. TFlite label:測試輸出文字是否為 label 裡的文字。
3. TFlite:測試辨識目標是否等同於出現最高機率的狗隻。

#### (2)圖片處理去背模組

1. ProcessDogPic:測試輸入須處理的圖片是否可以正常地出現

以黑色蓋住來達到去背結果。

(3)程式相機模組

1. CameraActivity:測試相機是否能被應用程式起動。

(4)程式通報模組

1. Warning:測試是否可以成功通報及跳轉到通報問卷。

## 2.系統**功能與非功能**測試

(1) AIDOG-NF-001

測試結果回傳時間在 300 毫秒以內

(2) AIDOG-NF-002

測試結果輸出正常無亂碼

(3) AIDOG-NF-003

(4) AIDOG-NF-004

測試圖片在伺服器內去背速度

(5) AIDOG-NF-005

(6) AIDOG-NF-006

(7) AIDOG-NF-007

測試多人同時通報是否能承載

(8) AIDOG-NF-008

(9) AIDOG-F-001

測試首次安裝此應用程式手機是否能開啟相機

(10) AIDOG-F-002

測試目標狗隻是否準確辨識

(11) AIDOG-F-003

(12) AIDOG-F-004

測試非目標是否讓辨識結果較低

(13) AIDOG-F-005

(14) AIDOG-F-006

(15) AIDOG-F-007

## 3.驗收測試



- (1) AIDOG-UC001  
測試是否出現權限問題，並開啟相機
- (2) AIDOG-UC002  
測試瞄準目標
- (3) AIDOG-UC003  
測試瞄準非目標
- (4) AIDOG-UC004  
測試瞄準無相關物體
- (5) AIDOG-UC005  
測試側邊欄開啟
- (6) AIDOG-UC006  
測試側邊欄關閉
- (7) AIDOG-UC007  
測試使用緊急通報
- (8) AIDOG-UC008  
測試是否能發送緊急通報

## 九、 測試報告

### 1.應用程式測試

測試模組案例名稱	測試結果	測試人員	測試時間
辨識模組(1)	Pass	陳聖勳	3/20
辨識模組(2)	Pass	陳聖勳	3/20
辨識模組(3)	Pass	陳聖勳	3/20
圖片處理去背模組	Pass	陳聖勳	3/18
程式相機模組	Pass	林駿丞	3/22
程式通報模組	Pass	林駿丞	3/22

### 2.系統功能與非功能測試

測試結果回傳時間在 300 毫秒以內

測試者	執行成功次數	總試驗次數	測試時間
同學 A	10	10	3/25
同學 B	20	20	3/25
同學 C	10	10	3/27

同學 D	15	15	3/28
------	----	----	------

在非模擬器開發的裝置上，使用者使用的測試裝置(Android 機器)上基本上皆可以達成在 300 毫秒內辨識的結果。

#### 測試結果輸出正常無亂碼

測試者	執行成功次數	總試驗次數	測試時間
同學 A	10	10	3/25
同學 B	20	20	3/25
同學 C	10	10	3/27
同學 D	15	15	3/28

在測試的過程，有可能會因輸出文字格式和程式設計不同而出現亂碼的狀況，使用英文作為輸出讓使用者測試讓測試結果正常。

#### 測試圖片在伺服器內去背速度在 2 秒以內

測試者	執行成功次數	總試驗次數	測試時間
陳聖勳	35	50	3/20
林駿丞	20	30	3/20

在伺服器有時可能會執行其他任務或是輸入的圖片較難解析而讓在伺服器內訓練的模型速度變慢，只達到 68.75%的成功率。

#### 測試多人同時通報是否能承載

測試者	執行結果	總試驗次數	測試時間
同學 A	Pass	10	4/01
同學 B	Pass	3	4/01
同學 C	Pass	10	4/01
同學 D	Pass	2	4/01

同學同時傳送緊急通報，皆成功。

#### 測試首次安裝此應用程式手機是否能開啟相機

測試者	執行結果	總試驗次數	測試時間
同學 A	Pass	1	3/25
同學 B	Pass	1	3/25
同學 C	Pass	1	3/27
同學 D	Pass	1	3/28

測試首次開啟是否會因裝置不同而造成問題，而受測試者的裝置皆沒出現問題。

### 測試目標狗隻是否準確辨識

測試者	執行成功次數	總試驗次數	測試時間
同學 A	25	30	3/25
同學 B	46	50	3/25
同學 C	38	40	3/27
同學 D	43	45	3/28

多次測試後，受測試者用不同的裝置在校園內測試真實的狗隻，除了部分長得較相似的黑狗以外，辨識結果達到 92.1212% 準確。

### 測試非目標是否讓辨識結果較低

測試者	執行成功次數	總試驗次數	測試時間
同學 A	10	10	3/25
同學 B	14	15	3/25
同學 C	11	11	3/27
同學 D	16	17	3/28

在拍攝其他地方而非狗隻時，相似度辨識變得非常分散，除了一兩次辨識錯誤，其他次是成功的。

### 測試是否能發送緊急通報

測試者	執行成功次數	總試驗次數	測試時間
同學 A	3	3	3/25
同學 B	1	1	3/25
同學 C	5	5	3/27
同學 D	10	10	3/28

緊急通報個受試者皆成功傳送。

## 3. 驗收測試

### 測試在各 use case 模擬使用者的行為是否能成功通過

測試案例編號	測試結果	測試人員	測試時間
AIDOG-UC001	Pass	陳聖勳	4/02
AIDOG-UC002	Pass	陳聖勳	4/03
AIDOG-UC003	Pass	陳聖勳	4/03
AIDOG-UC004	Pass	陳聖勳	4/03
AIDOG-UC005	Pass	林駿丞	4/05
AIDOG-UC006	Pass	林駿丞	4/05
AIDOG-UC007	Pass	陳聖勳	4/07
AIDOG-UC008	Pass	林駿丞	4/05

在驗收及二次測試皆通過各功能的測試。