

國立清華大學 電機工程學系

實作專題研究成果摘要

Implementing UAV Object Avoidance  
With Image Classification

以影像辨識實作無人機避障

專題領域：系統領域

組 別：B347

指導教授：呂仁碩

組員姓名：戴劭名、莊智宇

研究期間：2023 年 3 月 6 日至 2023 年 11 月 22 日止，共 8 個月

## 1. 報告摘要

在此專題中我們目標是以影像辨識完成無人機的自動避障，透過 Digilent Cora z7 板子使用圖像感測器 ov7670 的輸出畫面，將畫面經過深度學習的分類，分類成無人機飛行當下所需要的指令，達到無人機的避障功能。

## 2. 報告內容

### (1) 背景/動機

近年來無人機技術已相當成熟，我們認為要讓無人機擁有空拍以外更廣泛的應用，達到真正的「無人」，深度學習和神經網路技術的不斷進步，將其應用於無人機控制的研究已經成為一個引人注目的領域。本研究旨在進一步探索這方面的應用，以提高無人機的自主性和性能。基於神經網路的控制系統可以提供更靈活和自適應的飛行策略，使無人機能夠更有效地應對不同情境，並實現更高效的任務執行。

### (2) 研究目的

本專題旨在完成無人機的避障實作，運用無人機上搭載的 ov7670 鏡頭，讀取拍攝畫面，用機器學習將畫面分成不同類別，讓無人機依據畫面情境避開障礙。

### (3) 研究方法

#### 1.樣本收集、標誌、擴增

在室內取得環境照片，將照片分類，分類方式以我們認為當下位置無人機最適合避開牆壁的動作。在周圍離牆壁較遠或駛離牆壁方向時，分類為前行(Forward)，在離右方牆壁較近且有夾角時，分類為左轉(Yaw-Left)，在離左方牆壁較近且有夾角時，分類為右轉(Yaw-Right)，在離右方牆壁較近且平行時，分類為左移(Roll-Left)，在離左方牆壁較近且平行時，分類為右移(Roll-Right)。使用 pytorch 的 transform 將每張圖片複製二十張並且在五度範圍內隨機旋轉並調整明暗、飽和，這部的目的是為了模擬無人機飛行時會有的誤差

## 2.模型訓練

訓練模型啟發自論文展示的 DenseNet 模型，自嘗試訓練模型時，我們意識到 DenseNet 過於龐大，我們的擁有資源無法訓練，且我們所需的功能並不會需要使用到 DenseNet，最終我們選擇以 AlexNet 模型實作，AlexNet 足夠我們使用，訓練時間較短，且符合我們所擁有的記憶體資源，接者我們選擇 CrossEntropy、Adam 為損失函數以及最佳化方法，硬體加速器使用 google colab 提供的 T4 GPU，在測試集上達到 99.47%的準確率。

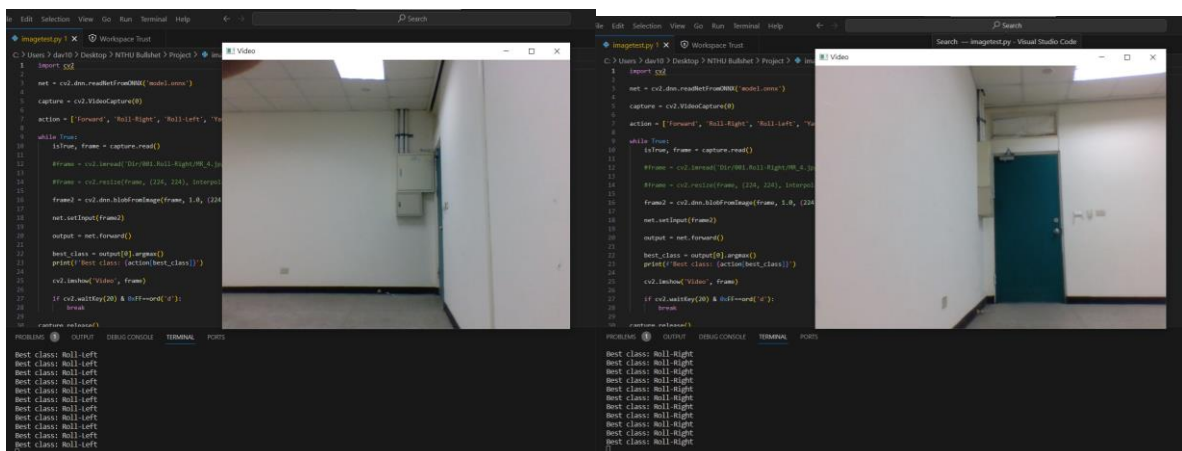
訓練過程我們以 pytorch 實作，首先我們載入需要的函式庫並聯結到雲端硬碟，在雲端上我們有自己的圖片集，我們定義圖片在進入模型之前必須經過變換，將圖片大小調整至 224x224 並轉換成 Tensor，然後建立資料集並將其分割，75%為訓練集，剩下 25%各半分在驗證集和測試集。再來我們建立 AlexNet 模型，區分圖片的方法是先將不同 class 的圖片先分類好，並在建立資料集時以資料夾名稱上 label 以節省時間。接下來我們將模型初始化，並移至 GPU 以便訓練，優化方面我們選擇用 Adam 並設 lr 為 0.001，執行訓練模型，我們分為 10 個 epoch 訓練，並且把驗證準確度最高的模型存下來，測試集結果:準確度 99.47%。

## 3.軟體模型測試

為了測試模型以及模擬使用 ov7670 所拍攝的影像，我們到測試場以電腦拍攝畫面，觀察模型分類畫面的表現。

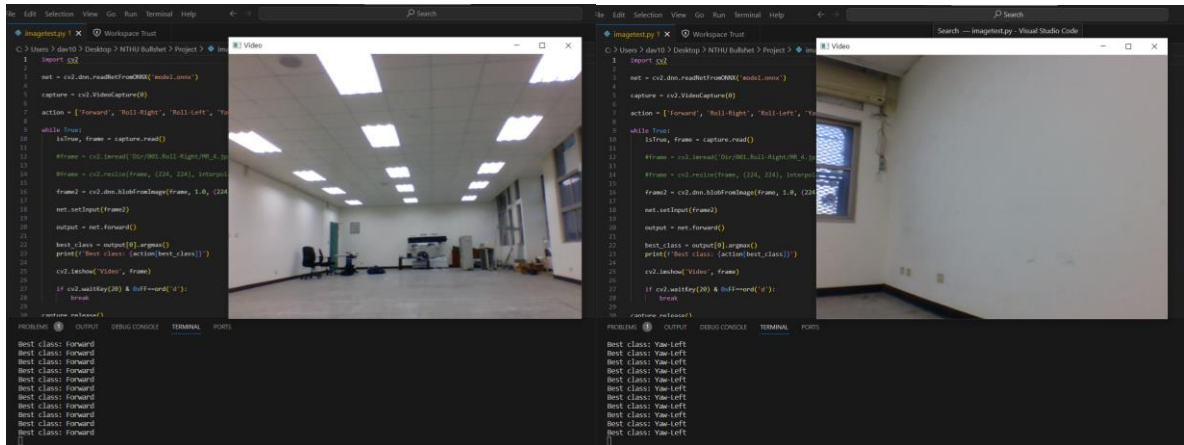
### (4) 研究結果

我們使用 python OpenCV 用筆電鏡頭在試飛測試場地做測試模型軟體



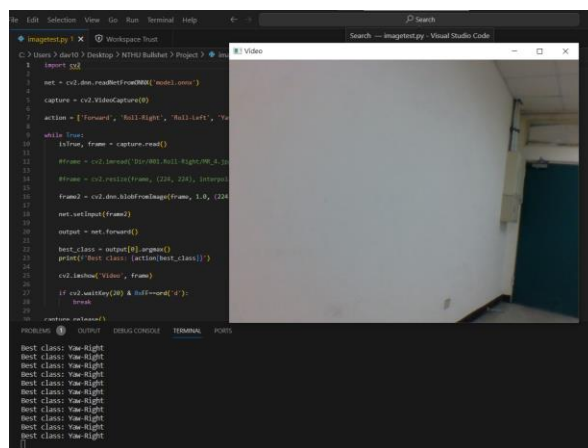
圖一、Roll-Left

圖二、Roll-right



圖四、Forward

圖三、Yaw-Left



圖五、Yaw-Right

## (5) 總結

本專題使用的影像辨識在模擬上可以給予無人機正確指令，成功的避開周遭牆壁，然而我們的指令都是方向上的指令，日後可以在模型中增加速度上的類別，使的無人機更加靈活。

### 3. 參考圖片

```
class AlexNet(nn.Module):
    def __init__(self, num_classes=1000):
        super(AlexNet, self).__init__()
        self.features = nn.Sequential(
            nn.Conv2d(3, 64, kernel_size=11, stride=4, padding=2),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
            nn.Conv2d(64, 192, kernel_size=5, padding=2),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
            nn.Conv2d(192, 384, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.Conv2d(384, 256, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.Conv2d(256, 256, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
        )
        self.avgpool = nn.AdaptiveAvgPool2d((6, 6))
        self.classifier = nn.Sequential(
            nn.Dropout(),
            nn.Linear(256 * 6 * 6, 4096),
            nn.ReLU(inplace=True),
            nn.Dropout(),
            nn.Linear(4096, 4096),
            nn.ReLU(inplace=True),
            nn.Linear(4096, num_classes),
        )

    def forward(self, x):
        x = self.features(x)
        x = self.avgpool(x)
        x = torch.flatten(x, 1)
        x = self.classifier(x)
        return x
```

圖六、AlexNet

### 4. 心得感想

在這次無人機專題中，很高興指導教授給予我們很多時間，從論文中找到我們自己想做的題目。在許多的論文閱讀中，我們不僅學到無人機、避障的相關知識，也訓練自己快速的掌握重點，理解一篇論文。除了指導教授與實驗室學長的協助之外，我們也得花時間從網路上找問題的解決方法，東拼西湊的變出我們要的結果，發現很多想法都需要技術來相輔相成，平時培養硬實力也是很重要的。