

資工系實務專題研究計畫成果報告(專題題目)

專題編號：106-CSIE-S004

執行期限：105年第1學期至106年第1學期

指導教授：陳彥霖教授

專題參與人員：103590004 林可均

103590010 施婉翎

一、摘要

現代家庭獨身一人養多隻寵物的生活方式越來越常見，然而飼養者時有外出上的困擾，為了實現這種不同寵物不同食物要餵養的概念，我們的目的就是作出一個能夠辨識貓狗的餵食器。

由於需要將兩個不同的寵物分開，我們使用攝影機視訊串流，加上目標追蹤分析技術與視訊動態來偵測貓狗的位置和移動，當寵物靠近飼料碗時，我們使用深度學習(Deep Learning)來辨識靠近飼料碗的是貓或狗，再以 Arduino 控制，從飼料槽內倒出一次份量的飼料。

關鍵詞: Deep Learning、Tensorflow、貓狗辨識、OpenCV、Arduino Uno、Android App

二、緣由與目的

現在有許多家庭家中，同時養狗與養貓，或同時養三隻以上的寵物，當飼主不方便帶著寵物，家中又沒有人可以照顧時，就需要一個「寵物餵食器」。當飼主不在家，寵物因為飢餓而走到飼料碗面前時，寵物餵食器便會將飼料倒下一次的份量，而同時飼養狗與貓的家庭，寵物餵食器提供兩個以上的飼料槽，能夠分別裝上狗食和貓食，並在寵物靠近飼料碗時，偵測飢餓的寵物是貓或是狗，再倒出屬於牠的飼料。

三、研究報告內容

我們的流程圖如圖1、圖2，抓取影

像的部分採用 OpenCV 擷取影像，分析與比對影像使用深度學習，投遞飼料使用 Arduino Uno 控制板，詳細說明如下。

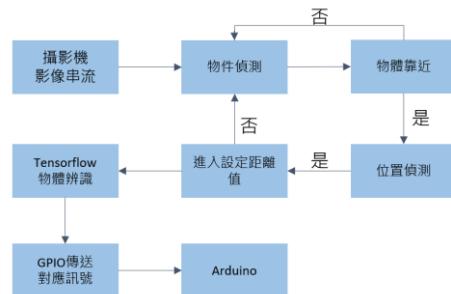


圖1. 軟體流程圖

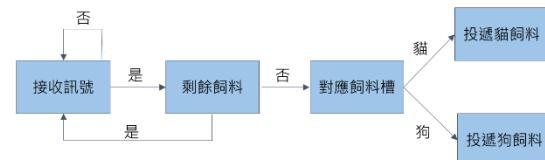


圖2. 硬體流程圖

(一)、Deep Learning[1]

Deep Learning 是用於自動地學習一些特徵，是無監督學習的一種[2]，概念源於人工神經網絡的研究，它採用了和神經網絡相似的分層結構，系統有包括輸入層、很多隱層、輸出層組成的多層網絡，只有相鄰層節點之間有連接，也就是上一層的輸出作為下一層的輸入，通過構建具有很多隱層的機器學習模型和海量的訓練數據，從而最後能夠提升分類或預測的準確性[3]。

(二)、物件追蹤

物件追蹤通常採用 OpenCV 做為開發背景，以兩種方式判別移動物體，第一種物件偵測功能為電腦對動態影像進行分

析，先區分前景(移動物件)與背景(固定不動)，再進一步標記前景之不同物件[4]，第二種則是先選取要追蹤的物體，再計算該物體的直方圖與該物體的反向投影，最後回傳追蹤的區域[5]。我們的專題採用第一種方法。而為了減少些微光影變化對抓取大型物體的影響，則會加上對物件大小的基本限制。

(三)、嵌入式系統

為了有效提高作品的便利性與對 Tensorflow 在 GPU 上面的運算，我們使用 NVIDIA Jetson TX2 作為開發的嵌入式平台[6]，同時在資料傳輸部分，使用 GPIO 接口傳輸至 Arduino。

(四)、攝影機視訊串流

我們的影像是用 opencv 擷取影像拍到的部分，再將影像交由 Deep Learning 模組來比對是否有貓狗待在飼料機前，進而投遞飼料。

(五)、Arduino

我們使用 Arduino 來接收 NVIDIA Jetson TX2 所發送的訊號，收到訊號後控制該飼料槽的馬達轉動，使飼料掉落。

四、結論

(一)、Deep Learning 辨識貓與狗

在飼主使用的前期，機器判斷錯誤的機率會比較大，但我們透過前期人為的修正或是使用手動的方式給飼料，讓機器漸漸地觀察出寵物之間特徵的不同，讓判斷錯誤的機率大幅降低，然後我們再將這方面的運算，透過雲端大量收集貓狗的數據，精進系統整體的辨識率，這樣不僅可以降低機器的成本、降低使用者的成本，在未來使用自動的方式給飼料時，也能夠更滿足寵物的需求。

(二)、Arduino Uno 硬體開發

我們製作出(圖 3)的外觀，並且使用 Arduino[7]來控制飼料槽的閘門開或關，

當收到訊號時，便打開相對應的閘門，倒出一次份量的飼料。



圖3. 全自動貓狗餵食器外觀

(三)、攝影機視訊串流

在攝影機偵測到寵物時將影像截取(如圖 4)，並同時將訊號傳給 Opencv 做物體偵測。

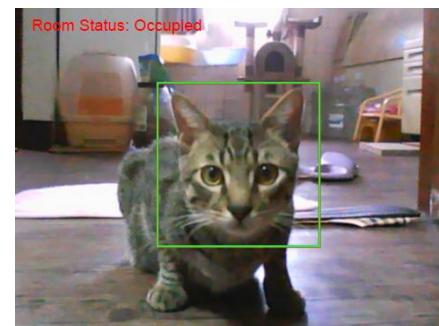


圖4. 偵測移動物體

(四)、追蹤分析貓狗位置

動態偵測貓狗的位置和移動，與原本背景中的樣本影像做比對，以分辨貓狗是否靠近飼料碗。當確定有貓狗靠近鏡頭時，再將影像作物種判別。

五、參考文獻

- [1] Deep Learning - https://en.wikipedia.org/wiki/Deep_learning
- [2] Lei Jimmy Ba and Rich Caruana. Do Deep Nets Really Need to be Deep?
- [3] TensorFlow - https://www.tensorflow.org/versions/r0.12/get_started/index.html

[4] Object Tracking 物件偵測 - http://geniusvision.net/manual/object_tracking.htm

[5] OpenCV Camshift -
<https://yulun.me/2013/opencv-csharp-emgu-object-tracking-camshift-algorithm/>

[6] NVIDIA Jetson TX2 - https://elinux.org/Jetson_TX2

[7] Arduino - <http://arduino.cc/>