

集成學習之口罩瑕疵辨識應用

王家恩^{1,a}, 何承遠^{1,2,b,*}

¹ 亞洲大學資訊工程學系

^{2*} 亞洲大學人工智慧博士學位學程

^a rtfgvb7415@gmail.com, ^btommyho@asia.edu.tw

摘要

2019年新冠疫情的爆發配戴口罩的人也日益增加，隨著口罩需求量的提升許多廠商紛紛投入口罩製造的行列，但傳統的口罩製造業產品的把關檢驗都是通過人工的方式一一檢查，此種方式不僅需要較多的人力，也需要較長的時間才能完成，儘管使用AI人工視覺的方式幫助分類，但許多較難分辨的錯誤還是難以糾正出來，模型準確度平均有89.5%、錯誤率為10.5%，於是本研究利用集成學習方式，希望藉由多個模型進行預測，並以投票表決的方式以多數決來決定預測結果，經過集成學習方式可將模型錯誤率降低16.2%，顯然集成學習方法可以有效的降低錯誤率並提升準確率，但光靠集成學習還是無法讓錯誤率下降太多，於是本研究針對集成學習中的模型進行權重調整，實驗是否能夠透過調整權重在提升準確度，經由實驗後發現，再經過調整權重後，最高可讓模型只剩7%的錯誤率，是現今平均錯誤率的2/3，明顯下降許多。

關鍵詞：深度學習、影像處理與應用、智慧製造技術與應用、集成學習。

1. 前言

2019年底COVID-19病毒在全球爆發並且流行，造成數億人感染，數百萬人死亡，人們為了避免感染做了很多防護措施，打疫苗、勤洗手、戴口罩，其中以戴口罩為最多人做的是其，因為口罩可以有效隔絕空氣中的病毒，可以降低人們感染的風險[1]，口罩的需求也因此增加，工廠也日以繼夜的趕工希望能夠產出更多的口罩供民眾使用。

而口罩在生產的過程當中，難免會出現瑕疵造成口罩無法使用、防護功能下降，或是外觀的不完整，目前大部分口罩工廠的品管都是依靠人工方式一一進行檢查，不僅需要較長的時間，也需要一定的人力成本。於是便希望藉由結合人工智慧(Artificial Intelligence, 縮寫AI)中的深度學習方法，來縮短口罩檢查所需的時間。

深度學習(Deep learning, 縮寫DL)是機器學習(Machine learning, 縮寫ML)中的一個子集合，而機器學習又是AI下的一個子集合，所以深度學習算是AI領域中的其中一種方法，機器學習與深度學習都是透過大量數據，讓機器模仿人腦來學習針對問題的推理與分類，而兩者的差異在於學習的方式[1]。機器學習需要先進行特徵擷取，以貓狗為例，假設可以判斷兩者差異的特徵有花色、耳朵形狀、臉型這三種，先將這些特徵擷取出來，再輸入訓練模型中；深度學習則不需要透過人為的特徵擷取，只需要將資料倒入模型訓練中，模型會自己擷取特徵並進行判斷。深度學習允許多個神經網路層組成的計算機網路具有多個抽象的數據表示，這項方法大大的提升了圖像辨識、語音識別及其他領域的突破。

本研究使用圖像辨識方法中的卷積神經網路[2]對口罩進行瑕疵檢測分類，在模型的選擇上，選用EfficientNets系列模型作為主要訓練模型，EfficientNets是Google於2019年提出了一個新的神經網路擴展方法，藉由平衡網路的深度、寬度、圖像解析度，深度指的是神經網路的層數[3]，寬度指的是神經網路的通道數，圖像解析度指圖像輸入神經網路的大小，透過三個維度的平衡進行有效的網路擴展。此外，Google使用神經網路搜索演算法(neural architecture search, NAS)得到一個新的基準網路(baseline Network)，並對其進行縮放，獲得一系列的模型稱為EfficientNets[3]。

會選擇此模型作為主要模型的原因是，EfficientNet在ImageNet上的表現比起其他模型都要來的優秀，與廣泛使用的ResNet-50相比，EfficientNet-B4將top-1準確度從76.3%提升到83.0%，並且計算量更少。EfficientNet-B7更在ImageNet上獲得84.3%的top-1準確度，並且模型縮小了8.4倍、速度增加6.4倍[3]。

但是在模型訓練的過程中發現，單一EfficientNet在口罩的瑕疵辨識上並不如預期，為了提升模型的準確度，決定利用集成學習方法[4]來提升模型的準確度，透過多個模型對一圖像進行分類並表決，採多數決方式決定分類結果，觀察其方法是否能夠有效提升模型對於圖片分類的準確度。

2. 研究方法

在此將介紹本研究的步驟，首先透過人工方式將口罩進行分類，主要分成七類其詳細內容會在下個章節做介紹，再將分類好的口罩用 insta 360 one R 運動攝影機對其進行拍照存檔，再將所有圖片依照 6:2:2 的比例分成 Train、Validation、Test 三個資料集，切分完成後將 Train 以及 Validation 資料集透過 image generator 進行資料擴增與資料處理。完成資料處理後，引入 EfficientNet B0~B6 模型，並讓其分別使用 Train 與 Validation 進行遷移式學習，且將學習好的模型儲存下來。最後透過 Ensemble learning 方法抽取奇數模型數量，利用 Test Data 對其進行驗證，觀察集成學習方法是否能夠提升模型對口罩分類的準確度，圖 1 為研究流程。

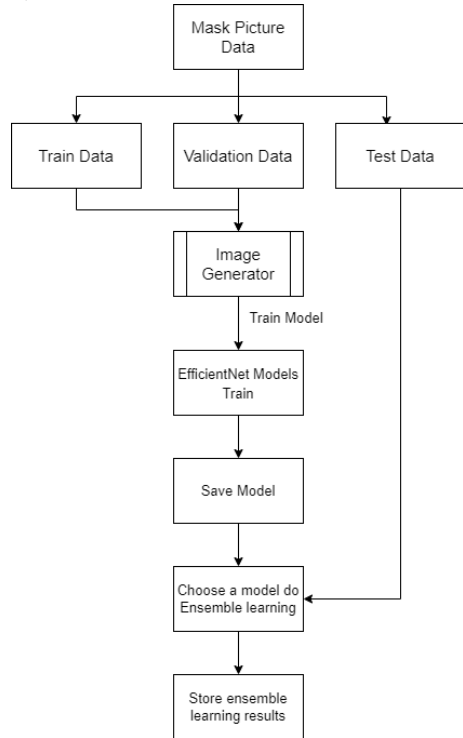


圖 1. 研究流程

3. 資料集介紹

本研究所使用的口罩資料可區分為 7 大類，分別是：正常口罩、耳掛瑕疵、無掛耳、鼻樑條瑕疵、無鼻樑條、掛耳焊點瑕疵、外觀瑕疵這七種下方為各分類的詳細介紹：

1. 正常口罩：無任何瑕疵，且可以正常使用的口罩。(如圖 2)



圖 2. 正常口罩

2. 耳掛瑕疵：口罩在製作時，掛耳沒有黏貼完全，造成耳掛斷裂或脫落。(如圖 3)



圖 3. 耳掛瑕疵

3. 無掛耳：口罩兩旁掛耳黏貼不完全脫落；或是生產時掛耳條使用完沒有及時補充造成。(如圖 4)



圖 4. 無掛耳

4. 鼻樑條瑕疵：口罩再進行超音波密封時，鼻樑條放置的位置不正確；或是鼻樑條裁切時的長度太短，造成超音波機器勿將其密封上，此種情形會造成超音波機器損毀。(如圖 5)

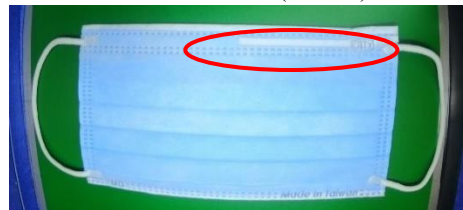


圖 5. 鼻樑條瑕疵

5. 無鼻樑條：口罩在密封時密封不完全，造成鼻樑條掉出；或是生產時鼻樑條使用完忘記補充。(如圖 6)

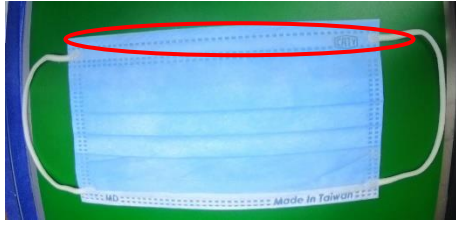


圖 6.無鼻樑條

6. 掛耳焊點瑕疵:此瑕疵屬於掛耳有確實黏貼,但是兩邊掛耳的位置有高低落差。這是因為掛耳在黏貼時會旋轉,造成位置偏移,此情形容易讓使用者在配戴口罩時會不舒服。(如圖 7)



圖 7.掛耳焊點瑕疵

7. 外觀瑕疵:外觀瑕疵只是單純的外觀不美觀,主要發生在口罩的上下緣,會出現布料不平整的問題,但使情形不影響口罩防護的性能。(如圖 8)



圖 8.外觀瑕疵

上述 7 類口罩先以人工的方式分類完成後,將所有口罩進行拍照建檔詳細數量(如表 1),所有口罩完成拍照建檔後,再將這些圖片資料依照 6:2:2 的比例拆分成 Train、Validation、Test 三個資料集,其中的 Train 與 validation 主要使用在模型訓練當中,而 Test 資料集則是使用在集成學習模型驗證當中。

表 1.口罩類別與數量

類別	類別 ID	數量(pics)
正常口罩	Normal	2058
耳掛瑕疵	Error_Ear	1562
無掛耳	No_Ear	1935
鼻樑條瑕疵	Error_Iron	2017
無鼻樑條	No_Iron	440
耳掛焊點瑕疵	Error_Solder	1934
外觀瑕疵	NG	2036

3.1 資料處理

資料處理部分本研究將其拆成兩個部份執行,第一是 3 個資料集都必須進行的圖片格式處

理;第二則是針對 Train Data 所做的數據增強以下將兩個做法分開來介紹。

3.1.1 圖片格式處理

圖片在進行訓練之前須將三個資料集中的圖片進行色彩變更、解析度變更、圖片數值規一化處理三種。

由於研究中所蒐集的口罩有很多種不同的顏色,且每個顏色的口罩數量也不相同,為了減低色彩對於模型所造成的影響,故將所有口罩顏色從彩色 RGB 圖像轉換成 GRAY 灰白圖像,降低顏色在模型學習中的重要性。(如圖 9)



圖 9.圖片 RGB 轉 GRAY 示意圖

解析度的部分,由於一開始在進行口罩拍攝建檔時所產出的影像為 1920 x 1080 大小的圖片,若是使用此大小讓模型進行訓練的話,會給 GPU 帶來很大的壓力,因為圖片越大所要花費的訓練資源就更多,若是 GPU 的記憶體太小的話很容易使模型訓練時記憶體不足,造成模型訓練中斷;而若圖片太小則會造成圖片許多特徵被容易被忽略掉。為了讓模型能夠順利訓練又能夠保留重要的特徵資訊,本研究測試了三種不同大小的圖片讓模型訓練並驗證,其中 240 x 140 的大小表現得最好(如表 2),最終決定將圖片統一縮小為 240 x 140 大小。

表 2.解析度選擇

圖片解析度	分類準確度
120 x 70	0.83534
240 x 140	0.91037
360 x 210	0.88995

最後是圖像數值歸一化的部分,將圖像矩陣的數值控制在 0~1 之間,這麼做的原因是為了凸顯每個圖像的差異,消除圖像中相似的部分,雖然更改後的圖片用肉眼無法清楚分辨出來,但在機器眼中卻是能夠凸顯其差異的部分。

3.1.2 數據增強

在使用深度學習時必須擁有大量的數據,才能訓練出一個好的模型,在資料蒐集的過程中,有些類別的口罩較不容易蒐集到像是無鼻樑條以及焊點錯誤這兩類,導致其資料量比其他類別

的要來的少，為了讓模型也能夠確實分辨出這些類別，也為了降低模型在訓練時過度擬合[5]，於是採用資料擴增方法，將 Train Data 進行數據增強，將其內的數據進行垂直及水平調整(如圖 10)。

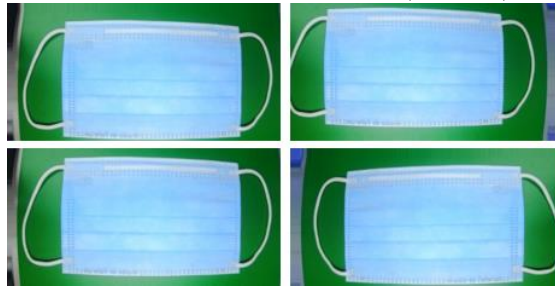


圖 10.圖片水平垂直移動範例

4. EfficientNet 模型訓練與測試

在模型訓練中，本研究引入 EfficientB0~B7 總共 8 個模型，分別對口罩資料進行訓練，在訓練開始之前針對每個模型做了些微的調整，原本模型輸出的 Dense 層擁有 1000 個分類神經元，將其改成只有 7 個神經元做分類，因為口罩的類別只有 7 類故作此調整(如圖 11)。

block7d_add (Add)	(None, 5, 8, 64@)	0	block7d_drop[0][0]
			block7c_add[0][0]
top_conv (Conv2D)	(None, 5, 8, 256@)	16384@0	block7d_add[0][0]
top_bn (BatchNormalization)	(None, 5, 8, 256@)	1024@	top_conv[0][0]
top_activation (Activation)	(None, 5, 8, 256@)	0	top_bn[0][0]
top_se_squeeze (GlobalAveragePo)	(None, 256@)	0	top_activation[0][0]
dropout (Dropout)	(None, 256@)	0	top_se_squeeze[0][0]
softmax (Dense)	(None, 7)	17927	dropout[0][0]
Total params: 64,115,614			
Trainable params: 63,804,887			
Non-trainable params: 310,727			

圖 11.調整 EfficientNet 輸出 Dense 層神經元數

調整好模型的神經層之後，在投入訓練前還必須設定模型訓練所需要的回調函數，在此針對三個回調函數進行設定，三個回調函數分別為：ModelCheckpoint、ReduceLROnPlateau、EarlyStopping 這三個，這三個函數各自負責了不同的任務。ModelCheckpoint 負責儲存最優秀的模型，每結束一次迭代就與前一次迭代比較若是成績優於前者則儲存並覆蓋之前的模型；ReduceLROnPlateau 用於定義學習率，模型在經過一定迭代後，效果就不會再有顯著的提升，因此借助 ReduceLROnPlateau 在訓練過程中縮小學習率，進而提升模型學習效果；最後 EarlyStopping 則適用於提早暫停模型訓練，當模型訓練到一定程度時就不會再提升，因此就沒有再繼續訓練的必要，故將其提早暫停。

所有參數設定完成以後便對模型進行訓練，訓練完成後，可以在表 3 當中看到，每個模型在每次迭代中的表現，每個模型的 loss 到最後都有很好的收斂，而 accuracy 的部分也越來越高表示準確率越精準，在理想狀態中，validation data 的

loss 與 accuracy 是越靠近 Train data 的結果越好，表示模型訓練時是往正向發展，但此圖表並不能夠明確的表示模型整體的準確度，因此必須透過 Test data 對模型進行測試。

表 3.模型迭代情形

模型	模型迭代
EfficientNetB0	
EfficientNetB1	
EfficientNetB2	
EfficientNetB3	
EfficientNetB4	
EfficientNetB5	
EfficientNetB6	
EfficientNetB7	

測試完成後，表 4 中分別存放模型的準確率以及模型分類的混淆矩陣，從中可以得知，每個模型的準確率大約都落在 0.87~0.90 之間，只有 EfficientB0 模型達到 0.91 的準確率，所有模型平均準確度達到 89.5%、錯誤率為 10.5%，並且在 Error_solder 那一類中分類錯誤的情況都很嚴重(如圖 12~圖 19)，藉此希望透過集成學習的方法來改善。

表 4.EfficientNets 準確率

模型	準確率	模型	準確度
B0	0.910379	B4	0.866194
B1	0.899124	B5	0.879949
B2	0.902876	B6	0.900792
B3	0.898707	B7	0.902042

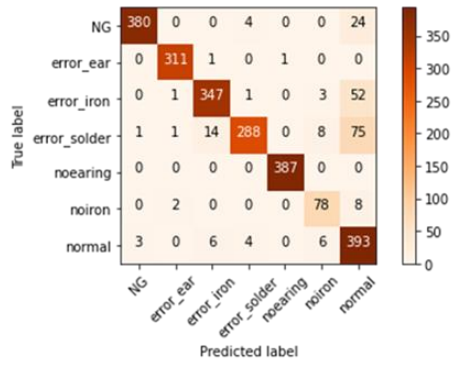


圖 12.B0 confusion matrix

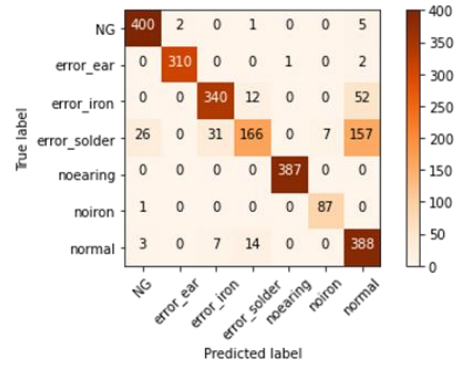


圖 16.B4 confusion matrix

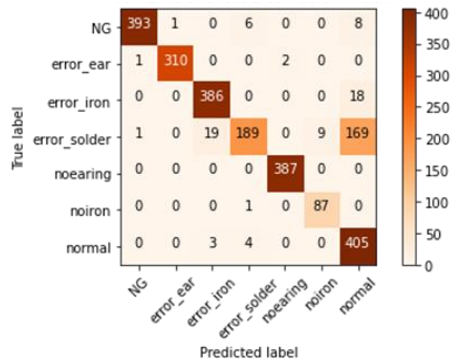


圖 13.B1 confusion matrix

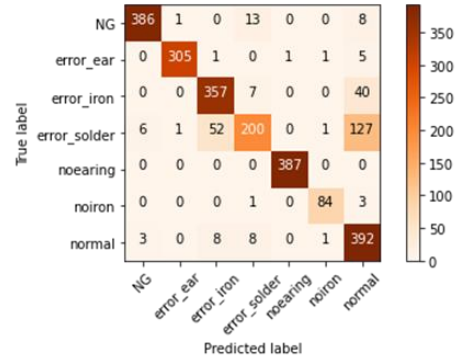


圖 17.B5 confusion matrix

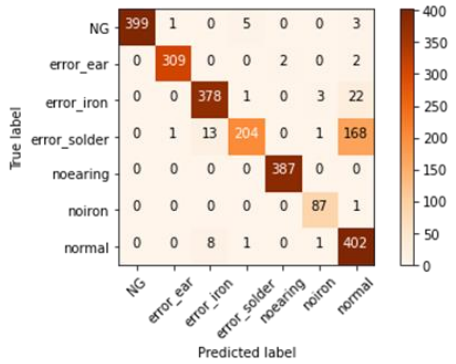


圖 14.B2 confusion matrix

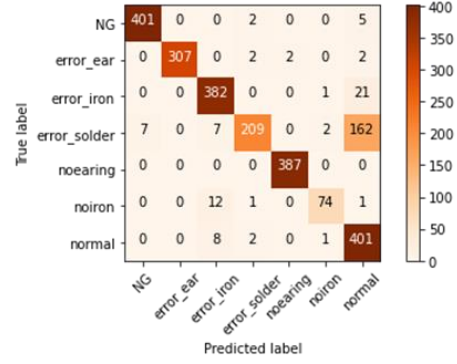


圖 18.B6 confusion matrix

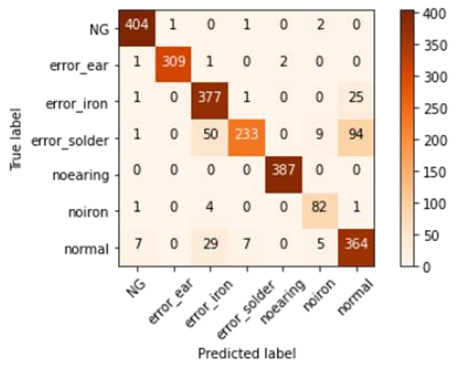


圖 15.B3 confusion matrix

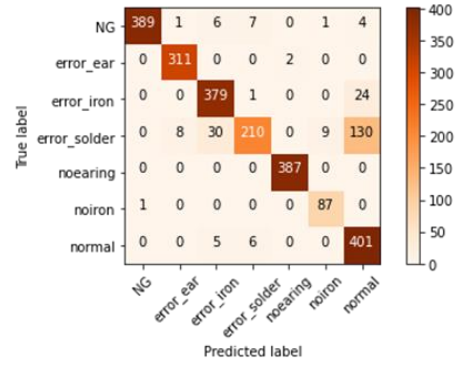


圖 19.B7 confusion matrix

5. 集成模型預測

完成所有的模型訓練及測試之後，依照測試所得到準確率來挑選模型組成集成學習團隊，在這邊總共搭配了 6 種集成學習模型，每個集成模型都是由不同數量不同組合的 EfficientNet 所組成，在模型編組的過程最重要步驟有兩個，第一是挑選成績較為優異的模型進行編組；第二是每個組合中的模型數量都必須是單數，會挑選優異模型的原因是因為希望藉由這些模型在提升對於物件的準確率，而組成模型要為單數的原因是，若是安排雙數模型進行組合的話，可能會有投票平手的情況發生，而投票若是平手模型則可能會隨機選取其中一個答案當成最終結果，故將所有集成模型都以單數組合而成(如表 5)。

表 5.集成學習測試準確率

集成學習組合	準確率
B0+B1+B2+B3+B5+B6+B7	0.919132
B0+B1+B2+B6+B7	0.918716
B0+B2+B6	0.913714
B0+B6+B7	0.917465
B0+B2+B7	0.918299
B2+B6+B7	0.915798

從表 5 中可以得知，模型透過組合成集成學習模型所測試出的準確率確實提升了一點點，原本單一模型的平均準確度為 89.5%、平均錯誤率為 10.5%，但透過集成後的平均準確率提升到 91.7%，錯誤率也下降了 16.2%，雖然只稍稍提升了一點點，但可以從混淆矩陣中發現，原本分類表現不好的 Error_solder 那一類正確數明顯提升了一些，就連 Normal 的分類準確度也稍稍提升一些(如圖 20~圖 25)，為了觀察是否能在提升集成學習的準確率，本研究針對只有 3 個模型組合的模型進行權重調整，會選擇只有三個模型的原因是因為權重比較好分配，依照個模型單一的表現來決定各模型在集成學習中的權重占比(如表 6)。

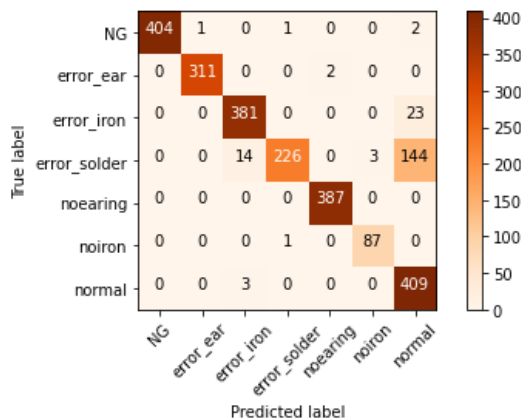


圖 20.B0+B1+B2+B3+B5+B6+B7 confusion

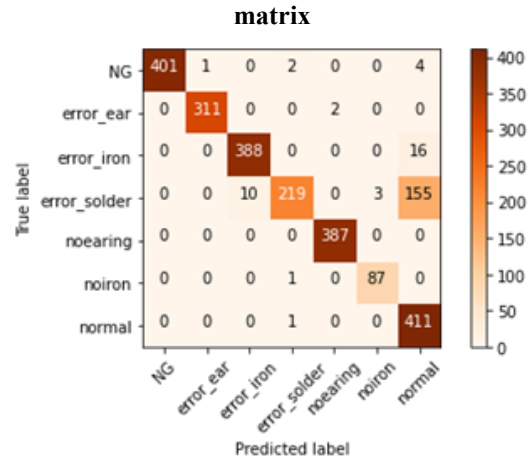


圖 21.B0+B1+B2+B6+B7 confusion matrix

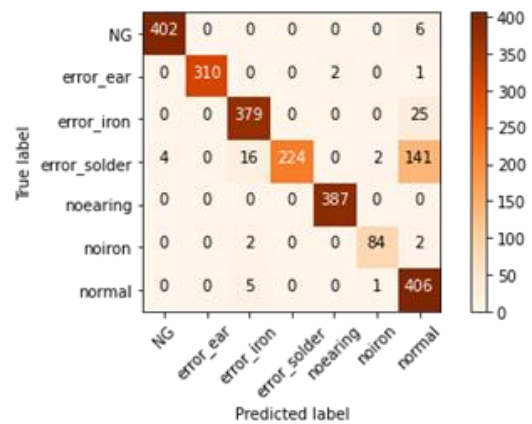


圖 22.B0+B2+B6 confusion matrix

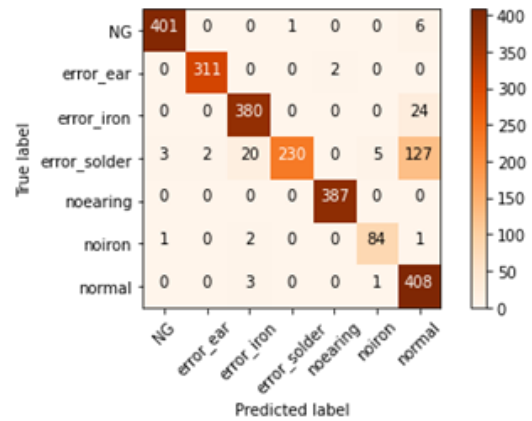


圖 23.B0+B6+B7 confusion matrix

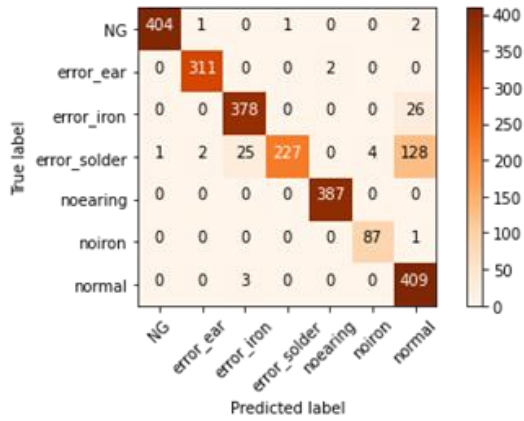


圖 24.B0+B2+B7 confusion matrix

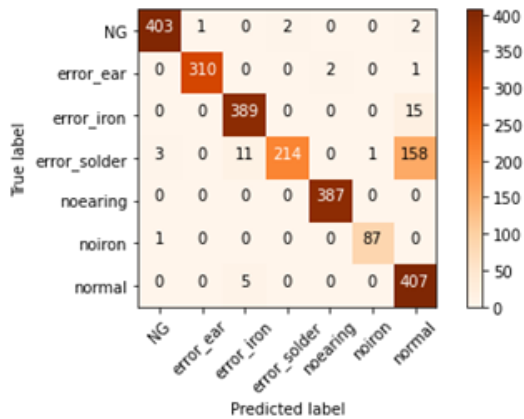


圖 25.B2+B6+B7 confusion matrix

表 6. 權重更改後準確率

集成學習組合	權重分配	準確率
B0+B2+B6	0.5+0.25+0.25	0.92621
B0+B6+B7	0.5+0.1+0.4	0.93038
B0+B2+B7	0.5+0.25+0.25	0.92997
B2+B6+B7	0.4+0.2+0.4	0.91663

表6中的權重是藉由表現較佳的模型獲得較多的權重所分配的，以此方式嘗試了多種不同的組合進行比較下，才確定其權重為最佳分配結果。從表6中可以發現，調整完權重之後整體的準確率也跟著提升了，從原本的0.91提升到0.92甚至是0.93，但最高就只能到0.93。平均準確度提升到了92.5%，並且最高可讓模型只有7%的錯誤率，比起初模型的2/3，Error_solder的正確數量也有明顯的提升很多(如圖26~圖29)。

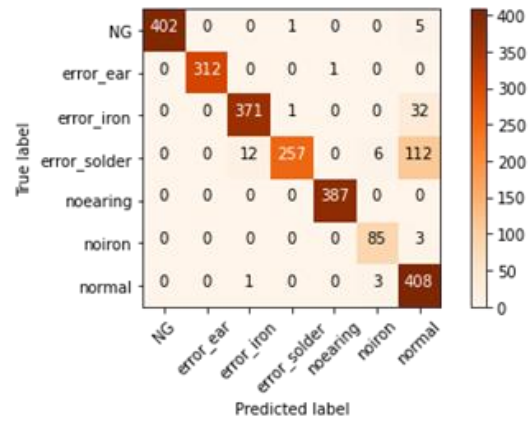


圖 26.B0+B2+B6 調整權重結果

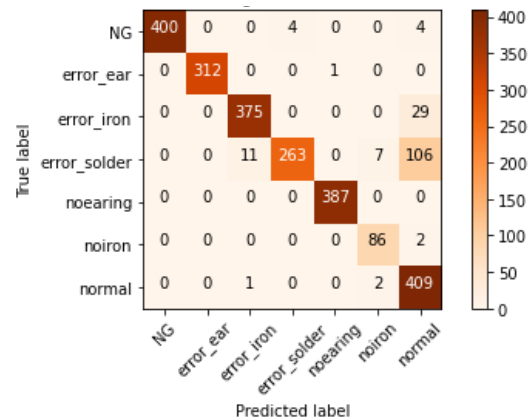


圖 27.B0+B6+B7 調整權重結果

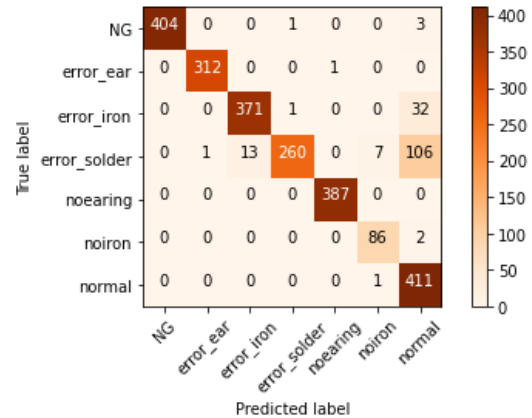


圖 28.B0+B2+B7 調整權重結果

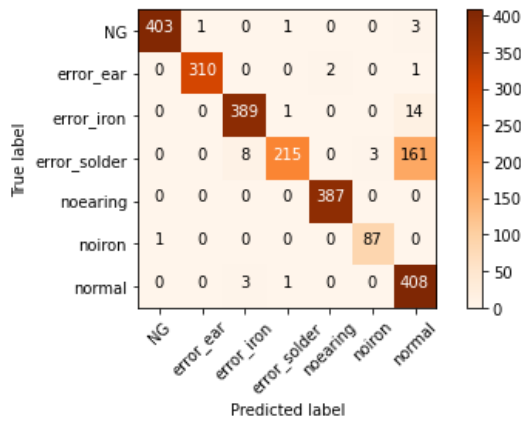


圖 29.B2+B6+B7 調整權重結果

6. 結論

在整個研究過程中可以發現，集成學習確實可以有效提升模型對於物件分類的準確率，可以將準確度從原本的 0.89 提升到 0.91；另外集成學習再搭配更改模型的權重更能夠改善模型的準確度，從 0.91 在提升到 0.92~0.93。但在整個研究中發現，不管是使用了集成學習，或者改變權重，Error_solder 的分類錯誤還是很多，模型很容易將其分類到 Normal 那一類去，後來得到的結論是，此類的特徵太過於不明顯，不容易讓機器學習到，若要能夠其更為準確的分類出來，解決辦法只有增加其資料量，或者是前期在做資料分類的時候訂定新的規則，焊點要偏移多大才能算是焊點錯誤，以上是本研究的所有觀點及結論。

致謝

本研究接受科技部編號：MOST 109-2221-E-468-008-MY3 研究計畫經費補助。

首先感謝科技部的補助，使我們能夠無所顧忌的完成此項研究。

也要感謝學校指導老師給予我們實驗上的建議，在實驗進行的過程當中，集成學習方法一度卡關準確度無法突破上去，感謝朱學亭老師的建議，透過改變權重的方式來提升集成學習準確度，也因為有老師的幫助實驗才得以順利完成。

更也感謝實驗室的同學，在程式撰寫有時會遭遇到瓶頸，同學都會伸出援手，給予有效的方

法及意見，使得研究得以進行下去，進而完成。

最後感謝主辦單位給予此機會，使我們能夠學習到更多人工智慧相關的技能，得以增進對深度學習的了解，以及相關技術的應用，也讓我們明白還有許多不足需要改進的地方。

參考文獻

- [1] O’Kelly, Eugenia et al. “How well do face masks protect the wearer compared to public perceptions?” medRxiv (2021).
- [2] LeCun, Yann et al. “Deep learning.” Nature vol. 521,7553 (2015): 436-44. doi:10.1038/nature14539
- [3] Tan, Mingxing and Quoc V. Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.” ArXiv abs/1905.11946 (2019): n. pag.
- [4] Huang, Faliang et al. “Research on Ensemble Learning.” 2009 International Conference on Artificial Intelligence and Computational Intelligence 3 (2009): 249-252.
- [5] Shorten, Connor and Taghi M. Khoshgoftaar. “A survey on Image Data Augmentation for Deep Learning.” Journal of Big Data 6 (2019): 1-48.
- [6] Fukushima, Kunihiko. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.” Biological Cybernetics 36 (2004): 193-202.
- [7] LeCun, Yann et al. “Gradient-based learning applied to document recognition.” (1998).
- [8] Oh, Kyoungsu and Keechul Jung. “GPU implementation of neural networks.” Pattern Recognit. 37 (2004): 1311-1314.
- [9] Albawi, Saad et al. “Understanding of a convolutional neural network.” 2017 International Conference on Engineering and Technology (ICET) (2017): 1-6.
- [10] Wang, Kunkun et al. “Application Research of Ensemble Learning Frameworks.” 2020 Chinese Automation Congress (CAC) (2020): 5767-5772.