

# 影像資料處理：自動光學瑕疵檢測

(Automated Optical Inspection, AOI)

組別 7 - 人為刀組

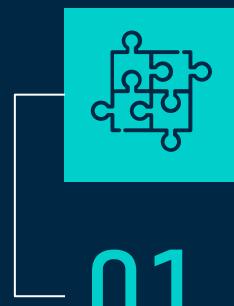
統計111 張婕琳 統計111 劉貞莉

統計111 侯登耀 統計111 邱瑞麒

統計112 曾庭勛 政治112 劉伊庭

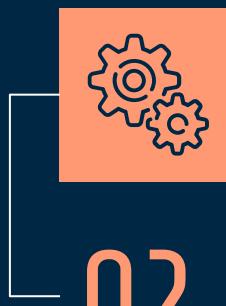
機器學習 鄭順林老師

# CONTENTS



01

資料介紹(p.04)



02

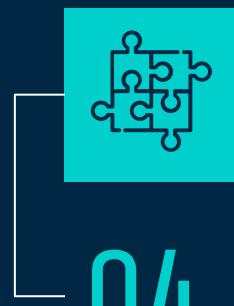
文獻回顧(p.06)



03

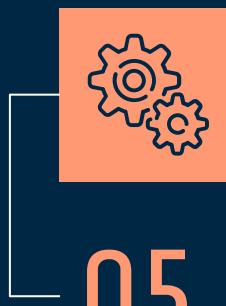
資料探索(p.61)

# CONTENTS



04

研究方法(p.74)



05

模型成果(p.77)



06

討論紀錄(p.157)

# 資料介紹

01

## ■ 問題描述 | 辨別瑕疵種類，

改良傳統上以人力使用光學儀器進行檢測的缺點

資料來源 | AIdea 人工智慧共創平台 (<https://aidea-web.tw>)

資料格式 | csv、PNG檔

資料筆數 | 1500筆

變數個數 | 4個 ( Label、圖像長、圖像寬、圖像灰階程度 )

新增變數 | 後續於模型成果—特徵創建頁面說明

資料維度 | 3 ( 圖像長、圖像寬、圖像灰階程度 )

資料解析度 | 512 x 512

# 文獻回顧

02

# 相關文獻

# Welding Defect Classification Based on Convolution Neural Network (CNN) and Gaussian Kernel

Agus Khumaidi<sup>1</sup> , Eko Mulyanto Yuniarno, Mauridhi Hery Purnomo  
2017 International Seminar on Intelligent Technology and Its Applications  
(ISITIA)

侯登耀

# 論文分析方法

- | 作者使用Welding Image Dataset，且defect總共有4種類別(分別為good, undercut, over spatter以及porosity)。  
每種類別各有30張data(共120張data)，每個類別取其中24張做為訓練data，剩下6張做為測試data。
- | 作者將原先 $320 \times 240 \times 3$ 的RGB圖片轉換成 $320 \times 240 \times 1$ 的灰階圖片，再使用CNN卷積神經網路進行分類。

| keras模型：

## 1. 卷積層(共2層)

卷基層用於萃取出特徵，而每一個卷積層都含有數層 $5 \times 5$ 的Gaussian kernel，其能夠過濾影像的灰階並產生10張新的影像，最後該卷積層會將這些影像進行線性轉換再傳到下一個卷積層。

## 2. 次取樣過程

透過ReLU做非線性轉換，使值不為負值。將照片拆成很多 $2 \times 2$ 的矩陣，並透過MaxPooling layer降維成 $158 \times 118 \times 10$ 的影像，並篩選特徵。將降維後的像素Flatten為向量，丟入64層的hidden layer，可得到4個類別的output vector。要分類時，進入activation function(sigmoid)進行分類，再配合gradient descent演算法調整參數。

# 論文分析結果

- | 作者藉由accuracy以及Mean squared error (MSE)去評估模型。
- | train accuracy以及validation accuracy隨著訓練iteration提升並在10次後趨於穩定。 validation accuracy落在95%~100%。而validation 的MSE平均約0.02，也是相當不錯。

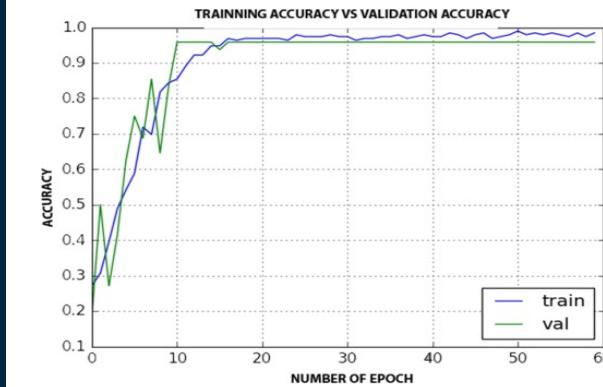


Fig. 5 Accuracy performance

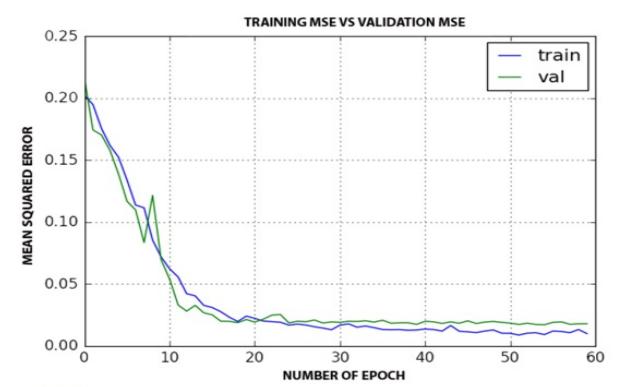


Fig. 6. MSE performance

# U-Net based Zero-hour Defect Inspection of Electronic Components and Semiconductors

Florian Kälber, Okan Köpüklü, Nicolas Lehment and Gerhard Rigoll (2021)

*In Proceedings of the 16th International Joint Conference on Computer  
Vision, Imaging and Computer Graphics Theory and Applications  
(VISIGRAPP 2021) - Volume 5: VISAPP, pages 593-601*

曾達勛

| 使用的資料集: DeepPCB dataset (Tang et al., 2019) and an internal semiconductor production line dataset

| 使用的分析方法:

□ **Preprocessing:** 進行image registration，將照片做完difference operation and a binarization後，貼在尺寸相同空白的模板上，之後使用SURF演算法在原始照片和上述所提的模板上，兩相比對後，再使用DBSCAN algorithm移除離群值。

□ **Data Augmentation:** 為了避免overfitting和資料的不足，每張圖片會用三種幾何的方式轉換，分別為rotation, cropping, and flipping (horizontal flipping and vertical flipping)，最重要的是，這些轉換不會改變照片上的label，在訓練時，選擇進行何種轉換的參數是被隨機指定，每次epoch改變一次此參數。

□ **Image Segmentation Models:** 使用以下模型，分別為U-Net、MobileNetV2和ShuffleNet並且用IoU(Intersection over Union)做評估。

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

| **Training Details:** 將資料集分割，80%用來training，剩下的20%用來做validation，訓練前，所有的輸入照片尺寸調為256x256。augmentation會隨機應用在這些照片，使用Adam optimizer 在參數 $\beta_1 = 0.9$ 、 $\beta_2 = 0.2$ 和learning rate  $\alpha = 0.0001$ ，使用BCE為損失函數，所有的模型都在epoch=200和batch size=16時進行訓練。  
使用OpenCV做資料預處理，使用Pytorch來建立CNN model。

$$BCE = -y_p \log \hat{y}_p - (1 - y_p) \log(1 - \hat{y}_p)$$

| **分析結果:** 這方法是基於fully convolutional U-Net architecture，這方法將照片視為二元圖像分類問題，每一個像素單獨進行二元分類，以確認照片中的異常，這方法在實驗中顯示出有良好的結果和性能。而且很彈性，因為相較於大多數現有的方法，這方法只取決於訓練的照片，舉例來說，一個用來偵測產品瑕疵的鏡頭，配合著缺陷偵測的模型，可設置在製造PCB產線的任意位置上，並能對設計和流程的改變做出快速反應，而且無須再用一批新的參考照片做訓練。

Table 3: Defect classification results of the chip defect dataset are compared between CNN classification models and U-Net based image segmentation models. On output images of the latter, a threshold is applied for classification.

Network	Acc.(%)	Prec.	Recall	F1-score
AlexNet	98.2	0.967	<b>0.988</b>	0.977
ResNet-101	98.3	0.988	0.967	0.977
MobileNetV2x0.5	<b>99.0</b>	0.990	0.985	<b>0.987</b>
ShuffleNetV2x0.5	93.4	0.911	0.918	0.914
UNet/MNetV2x1.0	<b>99.0</b>	0.996	0.977	0.986
UNet/SNetV2x0.5	98.4	<b>0.997</b>	0.960	0.978

Table 2: Defect detection results deepPCB dataset.

	ShuffleNetV2 x0.5 Encoder	MobileNetV2 x1.0 Encoder
Precision	0.915	<b>0.963</b>
Recall	<b>0.971</b>	0.965
F1-Score	0.942	<b>0.964</b>

# Fabric Defect Detection Using Activation Layer Embedded Convolutional Neural Network

Wenbin Ouyang, Bugao Xu, Jue Hou, and Xiaohui Yuan (2019)

*IEEE Access (Volume: 7)*

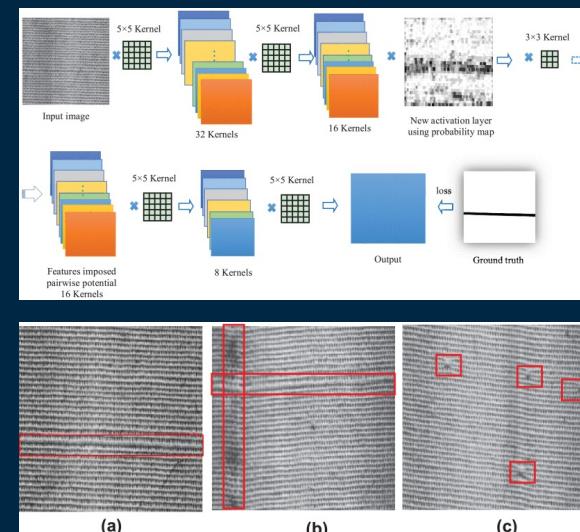
劉貞莉

# Research

- In this paper, a deep-learning algorithm was developed for an on-loom fabric defect inspection system by combining the techniques of image pre-processing, fabric motif determination, candidate defect map generation, and convolutional neural networks (CNNs).
- Method: Defect Probability Map Generation -> Pairwise Potential Activation Layer -> Network Architecture (7-layer CNN)

## Datasets

- Fabric defect dataset. This contains 1160 fabric images of  $500 \times 500$  pixels, which include horizontal defects (a), vertical defects (b), isolated defects (c), and defect-free fabrics.
- TILDA database as a verification set.  
Each TILDA image has a text description about defect areas in the image.



# Result

Defect results	Fabric defect dataset			TILDA database		
	Original image	Ground truth	PPAL	Original image	Ground truth	PPAL
1						
2						
3						

3-Metrics Evaluation	Precision	Recall	F1-Score	Precision	Recall	F1-Score
	0.9653	0.8636	0.9117	0.9225	0.8369	0.8776

- The average precision and recall of detecting defects in the existing images reached, respectively, over 90% and 80% at the pixel level and the accuracy on counting the number of defects from a publicly available dataset exceeded 98%.

# Position Determination of A Ball Grid Array by Automated Optical Inspection Method

Yi-Chuan Lin/ Kerwin Wang (2014)

*Proceedings of the 9th IEEE International Conference on Nano/Micro Engineered and  
Molecular Systems*

張健琳

## **Method:**

This paper introduces a novel automated optical inspection (AOI) method to measure the positions and diameters of micro-solder balls of a ball grid array (BGA).

## **Conclusion:**

the platform is capable of performing continuous image acquisition of a 15fps,  $640 \times 480$ , 8-bit grayscale video.

# Modern Architecture for Deep learning-based Automatic Optical Inspection

Johannes Richter, Detlef Streitferdt (2019)

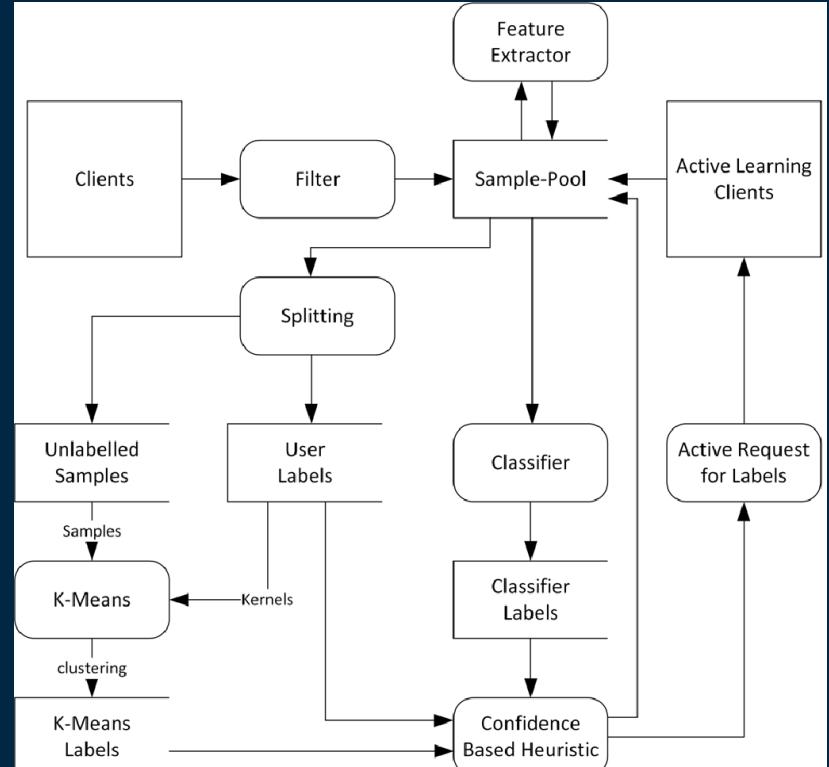
*2019 IEEE 43rd Annual Computer Software and Applications Conference*

邱瑞麒

- 本文獻的目的是透過主動學習和無監督式分群透過自動光學檢測圖(AOI)將尚未分類的通孔插裝技術(Through Hole Technology)以及印刷電路板(Printed Circuit Boards)分群，增加模型的樣本數，同時也訓練分類器。
- 用特徵提取、分群、集群分析、基於confidence value的選擇對未分類的影像進行分群。

$$c_i = 1.0 - \frac{d_{i0}}{d_{i1}}$$

$d_{i0}$  is the quadratic distance of the sample to its assigned centroid and  $d_{i1}$  the quadratic distance to its closest neighbor centroid.



- 特徵提取：用已標記的資料訓練自動編碼器(convolutional autoencoder)，形成隱藏層中的基本特徵。
- 集群分析：對未標記資料做PCA後再用K-means分群，最後每一個樣本都會獲得label和confidence value。
- 分類：在不做PCA的情況下使用分類器，並用相同的特徵去訓練不同的分類器，最後每一個樣本同樣會獲得label和confidence value。
- 主動學習：主要用來驗證集群分析和分類的假設，選擇confidence value 較高的label。
- 訓練模型：先透過移動、翻轉、旋轉、縮放或添加噪音來增加樣本數，之後，訓練會分多批次在多個epochs上運行，當產品質量達標或是檢測完目標數量後，停止訓練。新訓練出的分類器和集群分析結果會迭代至下一組未標記資料，讓預測結果更收斂。

# A High-Performance Deep Learning Algorithm for the Automated Optical Inspection of Laser Welding

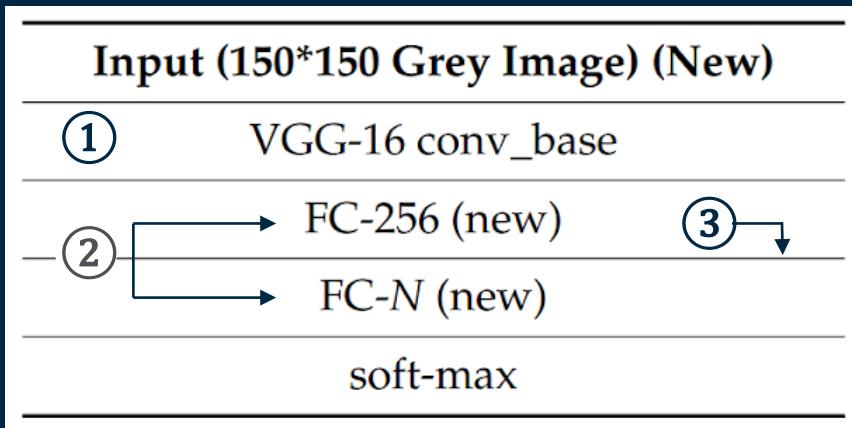
Yatao Yang, Longhui Pan, Junxian Ma, Runze Yang, Yishuang Zhu,  
Yanzhao Yang and Li Zhang (2020)

*Applied Sciences*

劉伊庭

# 研究方法

- 利用轉移學習 ( Transfer Learning ) 與預訓練 ( Pre-training ) 方法  
優化卷積神經網絡 ( CNN ) 的VGG-16模型。



優化後的 VGG-16 模型架構圖

## ① 卷積層

具有之前在 ImageNet 中訓練後保存的大量參數和權重，因此有強大的提取圖像邊緣和輪廓特徵的能力。

## ②全連結層 Fully Connected Layer

將兩個新的 FC 層替換原先VGG-16中的三個 FC 層。

③設置 $p=0.5$ 的dropout 率來正則化第一個 FC 層，目的為減少過擬合。

# 研究結果

- 使用 8941 張圖像（7217 張用於訓練、910 張用於驗證數據集、814 張用於最終測試數據集），訓練和驗證與最終測試準確率分別超過 99.9%、99.89% 和 99.87%。
- 由於採用預訓練的方法對模型進行訓練，因此：
  - ① 故障陽性率較低，僅為 0.16%
  - ② 訓練時間短，只需一小時，較為優化前少了兩小時
  - ③ 預測時間短，預測一張圖像只需 40 毫秒

# 關鍵文獻

# Textural Features for Image

Robert M. Haralick, K. Shanmugam, Its'Hak Dinstein (1973)

*IEEE Transactions on Systems, Man, and Cybernetics*  
(Volume: SMC-3, Issue: 6, Nov. 1973)

# 將圖像數字化

- 圖像轉變為數值時通常以  $(x,y)$  形式儲存
- $L_x = \{1, 2, \dots, N_x\}$  ;  $L_y = \{1, 2, \dots, N_y\}$
- $L_x \times L_y$  是分辨像元 (resolution cell) 的集合
- 數字圖像  $I$  是一個函數，為每個分辨像元分配灰度值  $G \in \{1, 2, \dots, N_g\}$
- 數字圖像  $I$  可進行各種二元分析，如編碼、復原、增強、分類等特定圖像處理任務。

# 尋找特徵

- 本研究的目標為開發基於分辨像元以及連續分辨像元的圖像分類方式
- 困難之處是如何**定義有意義的特徵**來描述來自分辨像元的圖片訊息
- 在尋找特徵時，會以在解釋彩色照片時使用的三個基本元素：**光譜 (spectral)、紋理 (textural)、上下文 (contextual features)**作參考依據
  - 光譜特徵描述電磁光譜可見光的各個波段平均色調變化
  - 紋理特徵包含有關波段內色調變化的空間分佈信息
  - 上下文特徵包含從被分析區域周圍的圖片數據塊中所獲得的信息

# 尋找特徵

- 若將黑白照片交由機器獨立處理時，**紋理和色調 (tone)** 最為重要
  - 色調的概念是基於圖像中分辨像元的不同灰度
  - 紋理與灰度色調的空間（統計）分佈有關
- 紋理和色調並非獨立的概念，兩者息息相關
- 紹理是物體表面的特性，蘊含表面結構排列及其與周圍環境關係的重要資訊，可用於辨別圖像，因此文欲提出一種計算圖像紋理特徵的快速方法，並討論這些特徵在區分不同類型圖像之效果。

# 色調與紋理

- 色調與紋理是構成每個離散色調特徵的分辨像元之空間模式。
- 當圖像中的一小塊區域：
  1. 離散灰度色調的特徵變化很小時，該區域主要屬性為色調。
  2. 離散灰度色調的特徵廣泛變化時，該區域主要屬性為紋理。
    - 1) 當沒有空間模式且特徵間的灰度變化很大時，會產生精細的紋理。
    - 2) 隨空間模式變得更明確且涉及越多的分辨像元，會產生粗糙的紋理。
- 獲取圖像紋理特徵的方式：假設紋理資訊是由一組灰度空間依賴矩陣指定，其矩陣針對圖像上相鄰分辨像元對之間的各種角度關係和距離進行計算。

# 灰度空間依賴矩陣

以  $N_x=3, N_y=3$  為例

135°

90°

45°

0°

0°

45°

90°

135°

6	7	8
5	x	1
4	3	2

resolution cell 解析度格

# 灰度空間依賴矩陣

		Grey Tone			
		0	1	2	3
Grey Tone	0	#(0,0) #(0,1) #(0,2) #(0,3)			
	1	#(1,0) #(1,1) #(1,2) #(1,3)			
	2	#(2,0) #(2,1) #(2,2) #(2,3)			
	3	#(3,0) #(3,1) #(3,2) #(3,3)			

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

0°

$$P_H = \begin{pmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

45°

$$P_{RD} = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

# 灰度空間依賴矩陣

		Grey Tone			
		0	1	2	3
Grey Tone	0	#(0,0) #(0,1) #(0,2) #(0,3)			
	1	#(1,0) #(1,1) #(1,2) #(1,3)			
	2	#(2,0) #(2,1) #(2,2) #(2,3)			
	3	#(3,0) #(3,1) #(3,2) #(3,3)			

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

0°

$$P_H = \begin{pmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

45°

$$P_{RD} = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

# 灰度空間依賴矩陣

		Grey Tone			
		0	1	2	3
Grey Tone	0	#(0,0) #(0,1) #(0,2) #(0,3)			
	1	#(1,0) #(1,1) #(1,2) #(1,3)			
	2	#(2,0) #(2,1) <b>#(2,1)</b> #(2,2) #(2,3)			
	3	#(3,0) #(3,1) #(3,2) #(3,3)			

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

0°

$$P_H = \begin{pmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

45°

$$P_{RD} = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

# 數據集描述

- 砂岩的顯微照片

243 個樣本數，由每張大小為  $64 \times 64$  的圖像塊組成，圖像的灰度被等概率量化為 16 個級別，其紋理特徵是從距離 1 灰度空間依賴矩陣計算而來，劃分為五個砂岩類別。

- $9 \times 9$  英寸 1:20000 的航空照片

170 個樣本數，每張圖像都被數字化成  $20 \times 50$  的陣列，劃分為八個土地利用類別。

- 多光譜衛星圖像

624 個樣本數，由每張大小為  $64 \times 64$  的圖像塊組成，其紋理特徵是根據多光譜圖像集中第二個 MSS 波段圖像的距離 1 灰度空間依賴矩陣計算而來，此外由於地面實況資訊來自彩色合成圖像，因此另外使用一組光譜顏色特徵對這些圖像進行分類，其是由四個光譜波段裡每個波段中  $64 \times 64$  圖像塊的灰度值之平均值與標準偏差組成，最終劃分為七個土地利用類別。

# 分類算法

## ■ 方法一：分段線性判別函數法 (Piecewise Linear Discriminant Function Method)

- 適用於樣本數量較多的數據集（顯微照片和衛星圖像）
- 數據集被任意劃分為訓練集和測試集
  - 訓練集的樣本被運用在開發
  - 測試集的樣本則進行檢測
- 使用一組超平面（決策邊界）將空間模式劃分為多個區域，這些超平面的位置由樣本模式確定，其中每個區域都會由特定類別的樣本模式主導。當一個新的模式被呈現出來進行識別時，會依據它所屬的區域被分配一個類別。

# 分類算法

- 對於涉及  $NR$  個類別的多類別問題，總共使用  $NR(NR - 1)/2$  個超平面來劃分模式空間。
- 這些超平面由一組權重向量  $W_j$  定義： $i = 1, 2, \dots, NR, j = 1, 2, \dots, NR, j > i$ ，將屬於第  $i$  個和第  $j$  個的樣本模式分開這些類別。
- 所得出的迴歸類型算法可用於獲得權重向量。確定超平面的位置後，新模式的分類如下：對於每個類別  $C_i$ ，當呈現新模式  $X$  時，給出正響應的超平面的數量  $V_i$ ，是使用其中的  $Z$  透過將值為 1 的分量添加到  $X$  獲得的增強模式向量來確定的，即  $z = \begin{bmatrix} 1 \\ x \end{bmatrix}$

$$V_i = \sum_{\substack{j=1 \\ j \neq i}}^{NR} \frac{|W_{ij}^T Z| + W_{ij}^T Z}{2|W_{ij}^T Z|}, \quad i = 1, 2, \dots, N_R$$

# 分類算法

## ■ 方法二：最小 – 最大決策規則 (Min-Max Decision Rule)

- 適用於每個類別具有較少樣本數量的數據集（航空照片）
- 透過使用數據集中除一個樣本之外的所有樣本獲得決策邊界，再對遺漏的樣本進行分類。數據集所有樣本均會經歷此過程，以獲取分類方案的整體準確性。

# 分類算法

- 決策規則：
- 如果  $b_{nk}$  和  $a_{nk}$  定義均勻分佈的最小值和最大值，則  $(x_1, x_2, \dots, x_n)$  被分配給類別  $k$  僅當：

$$b_{nk} \leq x_n \leq a_{nk}, \quad n = 1, 2, \dots, N$$

$$\prod_{n=1}^N \frac{1}{(a_{nk} - b_{nk})} \geq \prod_{n=1}^N \frac{1}{(a_{nj} - b_{nj})}$$

- 對於所有  $j$  滿足  $b_{nj} < x_n < a_{nj}, n = 1, 2, \dots, N$ 。如果不存在  $k$  滿足  $b_{nk} < x_n < a_{nk}, n = 1, 2, \dots, N$ ，則  $(x_1, x_2, \dots, x_n)$  被分配給類別  $k$  僅當：

(  $K$  為類別數量 )

$$\begin{aligned} & \sum_{n=1}^N \min \frac{\{|x_n - a_{nk}|, |x_n - b_{nk}|\}}{(a_{nk} - b_{nk})} \\ & \geq \sum_{n=1}^N \min \frac{\{|x_n - a_{nj}|, |x_n - b_{nj}|\}}{(a_{nj} - b_{nj})}, \quad j = 1, 2, \dots, K \end{aligned}$$

# 分析結果

## 1.顯微照片

- 143個樣本訓練集和100個樣本測試集
  - 使用訓練集的樣本導出分段線性判別函數，用來劃分五個砂岩類別
  - 對測試集的樣本進行分類，該分類方法在測試集上的總體準確率為 **89%**

## 2.航空照片

- 逐一計算 169 張圖像的最小和最大統計量，並使用最小 – 最大決策規則對遺漏的那張圖像進行分類。此過程重複 170 次將所有圖像完成分類
- 在這 170 張圖像中，有 140 張的圖像被正確分類，準確率為**82.3%**。

# 分析結果

## 3.衛星圖像

- 採用分段線性判別函數法識別該數據集中每個影像塊的土地利用類別。分類器的輸入變量包括四個紋理特徵和八個光譜特徵。
- 分類器在訓練集的 314 個樣本進行訓練，測試集中的310個樣本每個都被分配到其中一個土地利用類別，測試的總體準確率為 **83.5%**。
- 若此衛星圖像僅使用光譜特徵分類，分類準確率為 **74-77%**，低於使用光譜和紋理特徵組合的 83.5% 分類準確度，因此結果顯示，若將紋理特徵用作分類器的附加輸入，分類準確度可能會顯著提高。

# Classification of Solder Joint Using Feature Selection Based on Bayes and Support Vector Machine

Hao Wu, Xianmin Zhang, Hongwei Xie, Yongcong Kuang, Gaofei Ouyang (2013)

*IEEE Transactions on Components, Packaging and Manufacturing Technology*  
(Volume: 3, Issue: 3, March 2013)

# 研究問題

- 判別圖像分類，決定該圖片之瑕疵種類
  - 瑕疵種類分為：good solder (Gs), pseudo solder (Cs), solder insufficient (Si), component shifted (Ct), wrong component (Wc), and tombstone (Tb) 六種
- 建立各式有助於圖像判別之特徵
- 研究步驟分兩階段進行
  - 選取合適的特徵以建立 Bayesian Classifier 進行 Qualified Test
  - 通過 Qualified Test 者分類為 good solder，未通過者再次選擇合適的特徵，由SVM Classifier將圖片分類為 pseudo solder, solder insufficient, component shifted, wrong component, and tombstone.

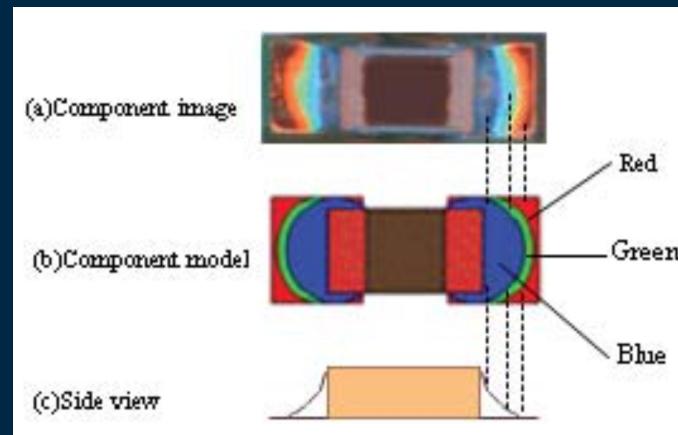
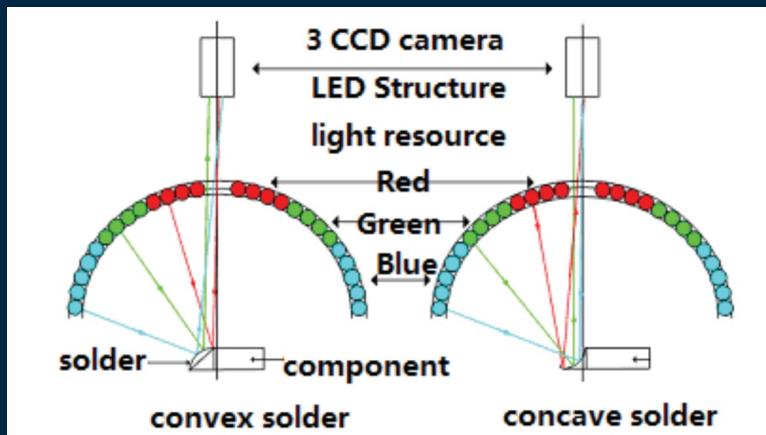
# 資料簡述

- 此文獻使用280個 type 0402 晶片組件 (chip components) , 以其不同瑕疵種類的焊接頭 (solder joint) 圖像作為研究樣本。每個晶片組件圖像分別有：
  - 組件體區 (component body area)  $100 \times 80$  像素
  - 焊料隆起焊盤 (solder pad area)  $60 \times 80$  像素
- 每個焊接頭都已由專業檢驗員事先檢驗過，並分類其屬類別，瑕疵種類與數量如右表所示
- 後述研究會切分成訓練集與測試集進行

Type	Number
Tombstone	19
Wrong component	30
Solder insufficient	30
Component shifted	44
Pseudo solder	48
Good solder	109
Total	280

# 資料取得

- 圖像資料是由 CCD color digital camera 和 three-color LED array illumination system 取得，其中 three-color 是指紅、綠、藍三色，範例圖像如下左所示
- 此種資料取得方式可以將 3D 資訊轉換成 2D 的彩色圖像，範例圖像如下右所示



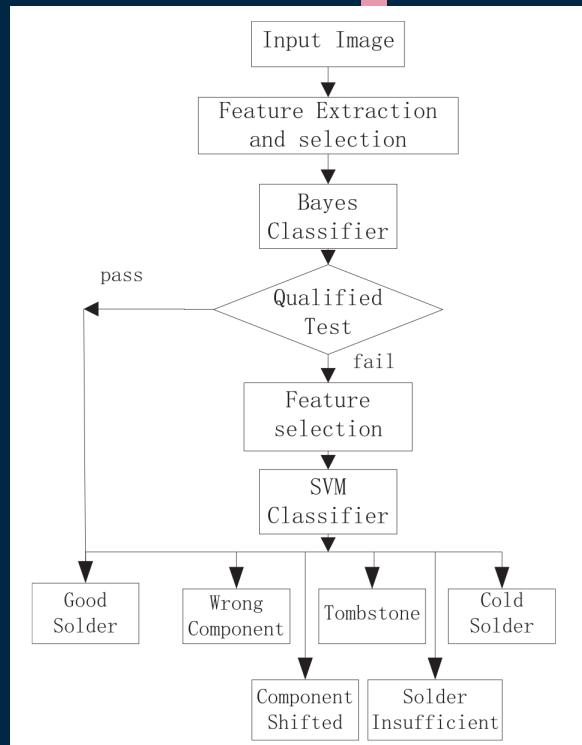
# 資料種類

- 此文獻資料根據不同的瑕疵進行分類，  
瑕疵種類分為六種：
  - good solder (Gs)
  - pseudo solder (Cs)
  - solder insufficient (Si)
  - component shifted (Ct)
  - wrong component (Wc)
  - tombstone (Tb)
- 範例圖像如右圖所示

Type	Image	Solder Model	Side View
Good solder			
Pseudo solder			
Solder insufficient			
Component shifted			
Wrong component			
Tombstone			

# 研究方法 — 研究流程

- 輸入圖像
- 特徵提取與選擇
- 貝氏分類器 (Bayesian Classifier)
- Qualified Test
  - pass
    - good solder
  - fail -> 特徵選擇 -> 支援向量機分類器 (SVM Classifier)
    - Tombstone
    - Wrong component
    - Solder insufficient
    - Component shifted
    - Pseudo solder



# 研究方法 — 特徵提取

- 此文獻提取色彩特徵與特徵匹配兩種類型的特徵，定義如下：
  - 色彩特徵 (Color Features)
    - 平均色彩飽和度 (average intensity value)
    - 明亮度百分比 (percentage of highlights)

$$X = (X_1, X_2, X_3, X_4, X_5, X_6)$$

$$X_j = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N I_r(x, y)$$

$$X_k = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N T_r(x, y) \times 100$$

- $X$ : 六維度的特徵向量
- $j$  與  $k$ : 1至6 個特徵向量
- $I_r(x,y)$ : 圖像第  $r$  個色框的色彩飽和度  
( $1 = \text{red}$ ,  $2 = \text{green}$ ,  $3 = \text{blue}$ )
- $T_r(x, y)$ : 每個色框的

# 研究方法 — 特徵提取

- 此文獻提取色彩特徵與特徵匹配兩種類型的特徵，定義如下：
  - 特徵匹配 (Template Matching Feature)
    - 為  $X7$  特徵
    - 演算法為將所需要區塊切割出來，並隱藏其他不需要的區塊，以此作為模板 (template, T) 與測試集 (search image, S) 比較的過程
    - 以正規化交互相關 (normalized cross correlation, NCC) 計算
      - ✓  $NCC(x,y) = +1$  or  $-1$  為 T 與 S 完美匹配
      - ✓  $NCC(x,y)$  絶對值大 表示 T 與 S 密切對應
      - ✓  $NCC(x,y)$  接近 0 表示 T 與 S 沒有很好的對應
      - ✓ 計算過程列於後

# 研究方法 — 特徵提取

- 此文獻提取色彩特徵與特徵匹配兩種類型的特徵，定義如下：
  - 特徵匹配 (Template Matching Feature)

$$\begin{aligned} X_7 &= \text{ncc}(x, y) \\ &= \frac{1}{n} \sum_{(u,v) \in T} \frac{t(u, v) - m_t}{\sqrt{\sigma_t^2}} \frac{s(x + u, y + v) - m_s(x, y)}{\sqrt{\sigma_s^2(x, y)}} \\ m_t &= \frac{1}{n} \sum_{(u,v) \in T} t(u, v) \\ \sigma_t^2 &= \frac{1}{n} \sum_{(u,v) \in T} (t(u, v) - m_t)^2 \\ m_s(x, y) &= \frac{1}{n} \sum_{(u,v) \in T} s(x + u, y + v) \\ \sigma_s^2(x, y) &= \frac{1}{n} \sum_{(u,v) \in T} (s(x + u, y + v) - m_s(x, y))^2 \end{aligned}$$

- $m_t$ : 模板的平均灰階值
- $\sigma_t^2$ : 模板的灰階值變異數
- $(x, y)$ : 模板在S的當前位置
- $(u, v)$ : 模板的像素位置
- $t(u, v)$ : 模板的灰階
- $s(x + u, y + v)$ : S的灰階

- 分別自 組件體區與焊料隆起焊盤提取上述 7 種特徵，故總特徵數為 14 個

# 研究方法 — 特徵選擇

- 總共 14 個特徵的情形下，此研究選擇以資訊增益 (information gain) 作為特徵選擇的方法
- 若目標屬性有  $c$  個不同的值，則與  $c$ -wise classification 相關的  $S$  熵 (entropy) 為定義為  $Entropy(S) = \sum_{i=1}^c -p_i \log_2 p_i$ , 其中  $p_i$  為  $S$  屬於類別  $i$  的比例
- 若目標屬性可由  $c$  個可能的值取得，則熵會與  $\log_2 c$  的值接近
- 已知屬性  $A$  值的情況下，期望會降低的熵可由  $Gain(S, A)$  表示
- $Gain(S, A) = Entropy(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$ ,  
其中， $values(A)$ : 屬性  $A$  所有可能值的集合， $S$ :  $S$  集合中屬性  $A$  為  $v$  值時的子集合
- 此法可計算該特徵在作為切割依據的機率，故機率越高，表示該項特徵越有價值

# 研究方法 — 貝氏分類器

- 此文獻根據前述的特徵提取，分兩階段進行分類
  - 第一階段：貝氏分類器 (Bayesian Classifier)
    - ✓ 分類焊接頭合格與否 (qualified or not)
    - ✓ 將總預期損失最小化
    - ✓ 需選擇能將後天機率  $p(w_i|x)$  極大化的類別， $w_i$ : 第 i 類別，x: 輸入的向量
    - ✓ 因此分類器可被表示為此判定函數 (decision function)
    - ✓  $d_i(x) = p(w_i|x) = p(x|w_i)p(w_i)$
    - ✓ 根據判定函數的輸出，能極大化判定函數的向量輸入便可以被分類到相對應的類別  $c_k = \{x|x \in C, d_x = \max_{i \in C}(d_i)\}$
    - ✓  $c_k$ : 第 k 個向量輸入的類別，C: 所有類別的集合

# 研究方法 — 支援向量機

- 此文獻根據前述的特徵提取，分兩階段進行分類
  - 第二階段：支援向量機 (SVM Classifier)
    - ✓ 分類焊接頭到相對應的瑕疵種類
    - ✓ 首先，演算法會透過 kernel function 生成的非線性轉換方法，把輸入的資訊轉換成高維度空間
    - ✓ 接著，找尋最佳化的線性分類
    - ✓ 不只是需要正確分割兩種類別，更要確保分類區間足夠大
    - ✓ 推導上經過計算 eigenvector, 解 quadratic programming, mapping kernel function, 使用 Lagrange multiplier 等步驟後，得出最終的判定函數

$$f(x) = \text{sgn}((w^*, x) + b^*) = \text{sgn}\left\{\sum_{i=1}^n \alpha^{i*} y_i K(x, x_i) + b^*\right\}$$

# 研究方法 — 支援向量機

- 此文獻根據前述的特徵提取，分兩階段進行分類
  - 第二階段：支援向量機 (SVM Classifier)
    - ✓ SVM 原設計為二元類別分類，而合併多個二元分類器後就能進行多類別之分類
    - ✓ 合併的方法有三：
      - one-against-one: 選擇兩種不同類別並建立所有二元組合的子分類器
      - one-against-rest: 選定一個類別後，跟剩下類別合併後進行比較
      - directed acyclic graph SVM (DAG-SVM): 將所有樣本分成兩大類，在兩大類裡面進行切割，直到兩大類中各剩下一個類別為止
    - ✓ one-against-one 與 DAG 分類準確性較高且分類速度快，而在此文獻只有六個類別需要分類的情況下，選擇 one-against-one方法

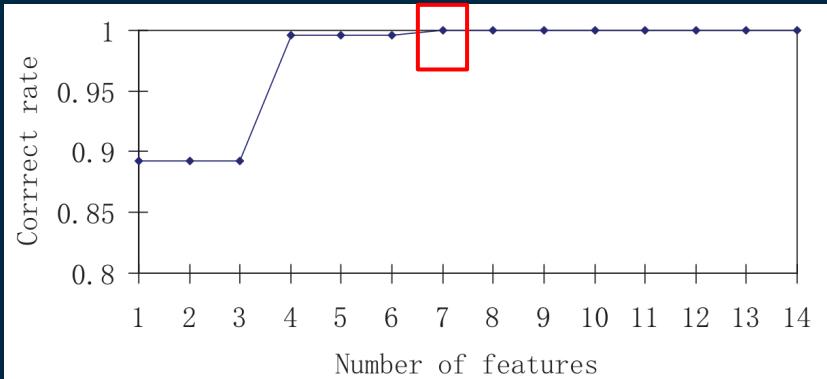
# 研究結果 — 特徵選擇

- 此文獻共有提取了 14 個特徵，依照前述資訊增益 (information gain) 的計算，將各個特徵的資訊增益由大到小排序如右表所示，發現第四個特徵資訊增益 1.4746 為首，表示第四個特徵最具有代表性，而第五個特徵資訊增益 1.3452 次之，以此類推
- 若文獻提及分類器選擇七個特徵，則表示選擇資訊增益前七大者，即第四、五、六、一、七、三個特徵

Rank	Feature	Information Gain
1	4	1.4746
2	5	1.3452
3	6	1.2918
4	1	1.2415
5	7	1.2221
6	3	1.1165
7	14	1.0772
8	2	0.8715
9	11	0.502
10	8	0.3863
11	10	0.3814
12	12	0.32
13	13	0.1308
14	9	0

# 研究結果 — 貝氏分類器

- 第一階段由貝氏分類器判斷焊接頭合規與否
- 由圖表可知，進行貝氏分類器時，選擇的特徵越多，正確率亦會提升
- 當選擇了七項特徵時，正確率已達 100%，故在此階段只需選擇七項特徵即可
- 多餘的特徵（選到八項、九項特徵等）已無意義，因無法再提供更多資訊



Number of Features	Correct Rate	Number of Features	Correct Rate
1	0.8929	8	1.00
2	0.8929	9	1.00
3	0.8929	10	1.00
4	0.9964	11	1.00
5	0.9964	12	1.00
6	0.9964	13	1.00
7	1.00	14	1.00

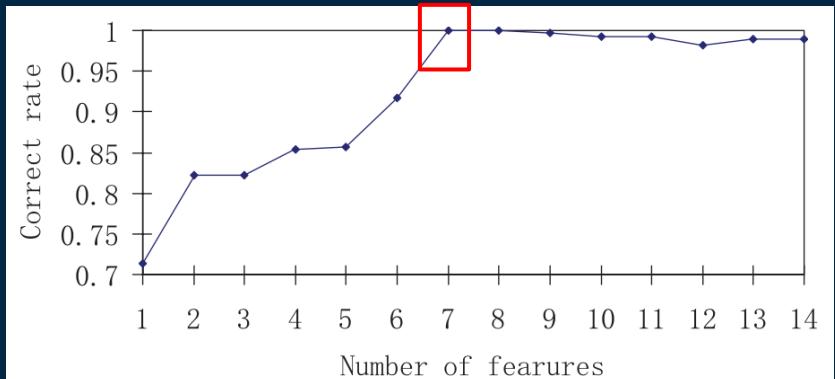
# 研究結果 — 支援向量機

- 第二階段由支援向量機分類器，分類焊接頭到相對應的瑕疵種類
- 首先決定要選用哪一種 kernel function
- 此文獻以四種不同的 kernel 測試
  - Linear ( $c=1$ )
  - Polynomial ( $c=1, g=2, d=3$ )
  - RBF ( $c=1, g=2$ )
  - Sigmoid ( $c=1, g=2$ )
- 右表顯示四種 kernel 進行分類的最高準確率，其中以多項式 (polynomial kernel SVM) 最佳，所以後續研究以多項式 kernel 進行
- 也表示此文獻的特徵間具有多項式關係 (polynomial relationship) 存在

Kernel (parameters)	Classification Accuracy
Linear ( $c=1$ )	99.29%
Polynomial ( $c=1, g=2, d=3$ )	100%
RBF ( $c=1, g=2$ )	86.43%
Sigmoid ( $c=1, g=2$ )	38.93%

# 研究結果 — 支援向量機

- 如前所示，本文獻的支援向量機選用多項式的 kernel function ( $\text{cost} = 1$ ,  $\text{gamma} = 2$ ,  $\text{degree} = 3$ )
- 由圖表可知，進行支援向量機分類器時，選擇的特徵越多，正確率亦會提升，但若超過九個（含）特徵開始，正確率便會降低
- 正確率會降低是因多餘的特徵所致，故在此階段選擇七項特徵



Number of Features	Correct Rate	Number of Features	Correct Rate
1	0.7143	8	1.00
2	0.8214	9	0.9964
3	0.8214	10	0.9929
4	0.8536	11	0.9929
5	0.8571	12	0.9821
6	0.9167	13	0.9893
7	1.00	14	0.9893

# 研究結果 — 補充比較其他分類法

- 為了比較第二階段的分類成效，此文獻有進行另外三種分類法：

- 鄰近演算法 (K-NN, K=1)
- 分類樹 (Tree)
- 反向傳播法 (Back Propagation, BP)

Classifier	Feature Number	Modeling Time(s)	Correct Rate
K-NN (K=1)	8	0.01	99.64%
Tree	7	0.04	99.28%
BP	7	0.66	100%
SVM	7	0.13	100%

- 右表顯示各個分類法所能達到的最高正確率，以及其所需的特徵數量即建模時間
- 發現 K-NN 與 Tree 所需建模時間短，但正確率低一些
- BP 與 SVM 正確率皆可達 100%，但 SVM 建模時間較短，故此文獻最後選擇以七個特徵進行 SVM 分類，以達到最佳檢驗成效

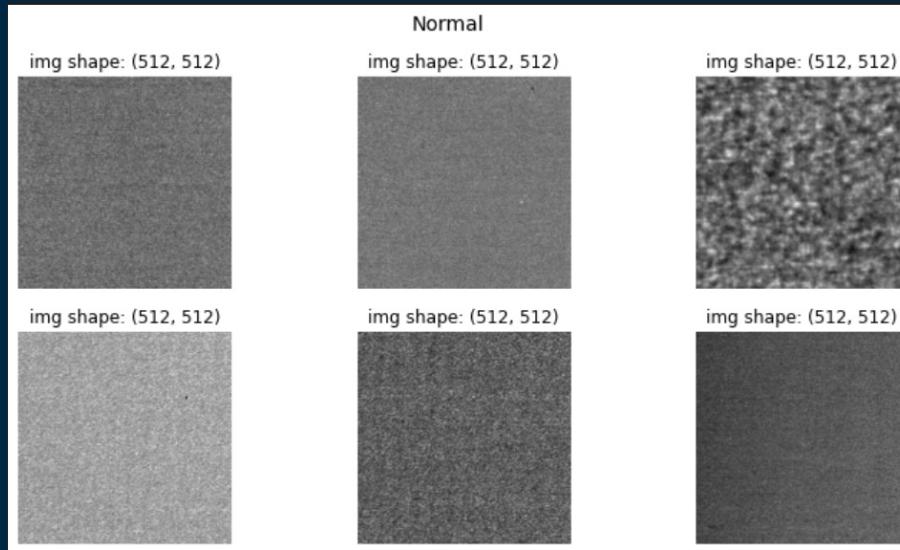
# 研究結論

- 此論文提供了兩階段的分類法，第一階段是二元分類，第二階段是多元分類
- 總共提取了 14 種特徵，以資訊增益大小來選擇合適的特徵
- 貝氏分類器 (Bayesian classifier) 判斷焊接頭合格與否
- 支援向量機分類器 (SVM classifier) 分類焊接頭到相對應的瑕疵種類
- 研究結果顯示，以特徵選擇的方法進行兩階段分類可以達到高識別率

# 資料探索

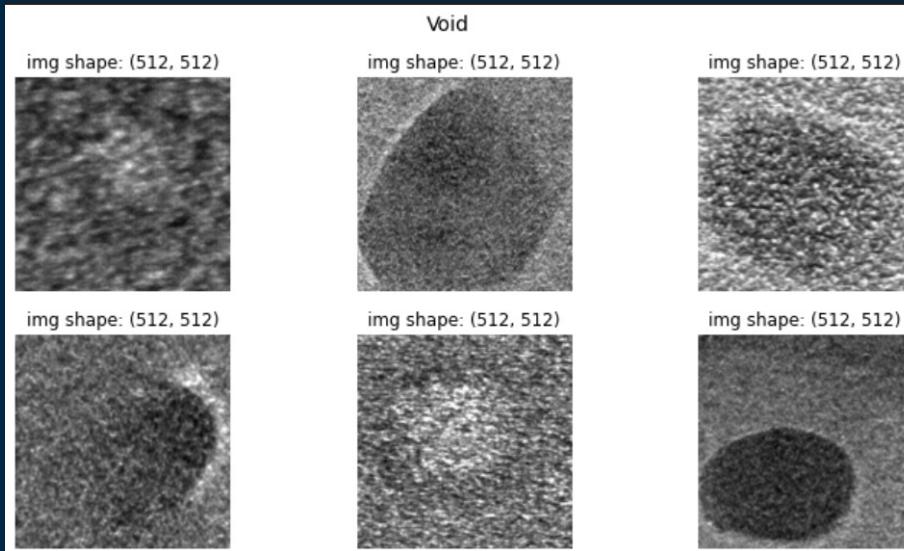
03

# 資料探索



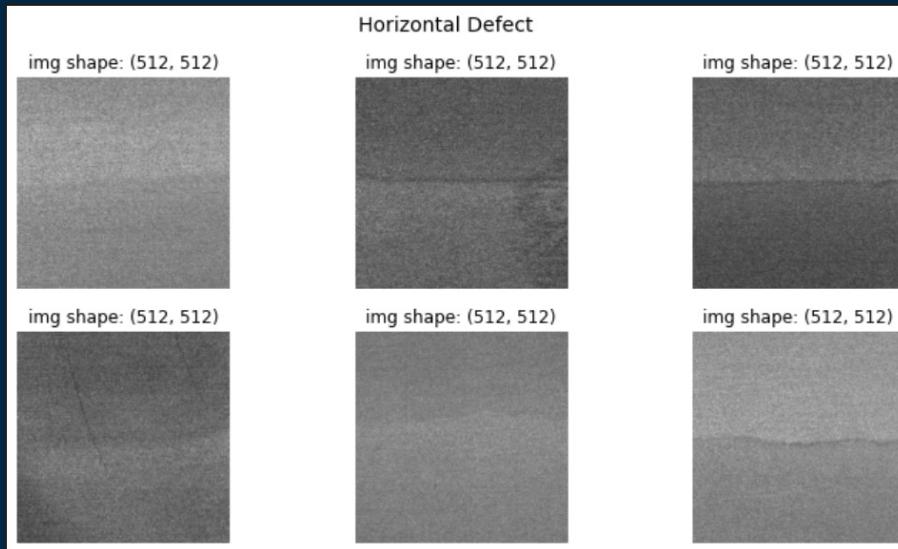
Label 0 : normal, 無瑕疵

# 資料探索



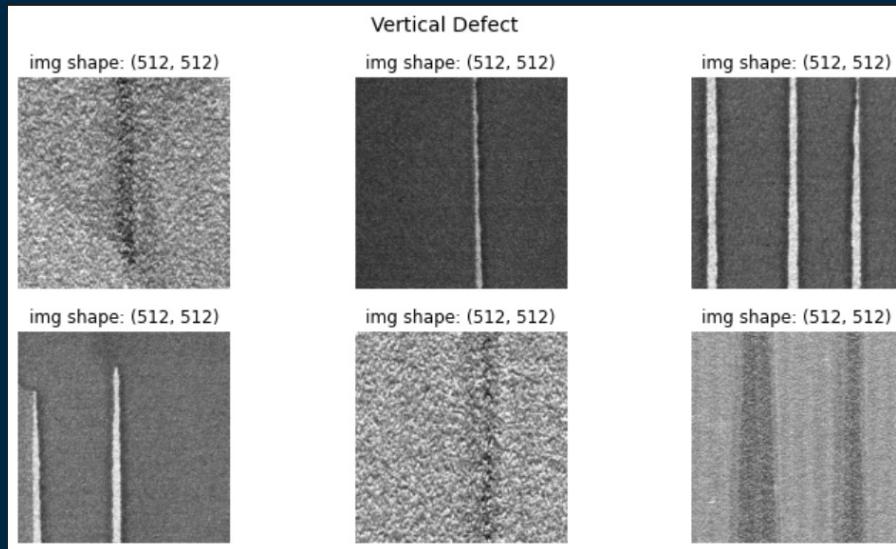
Label 1 : void, 空缺的

# 資料探索



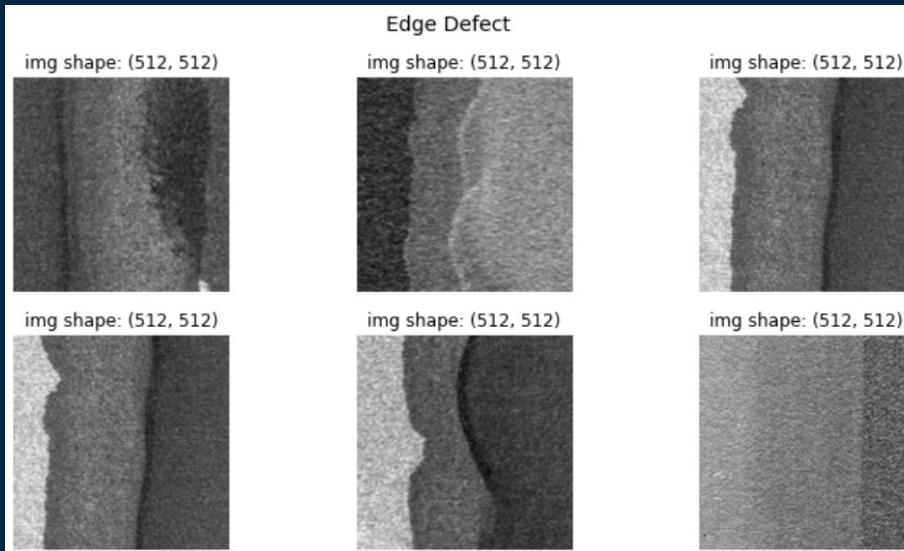
Label 2 : horizontal defect, 水平方向瑕疵

# 資料探索



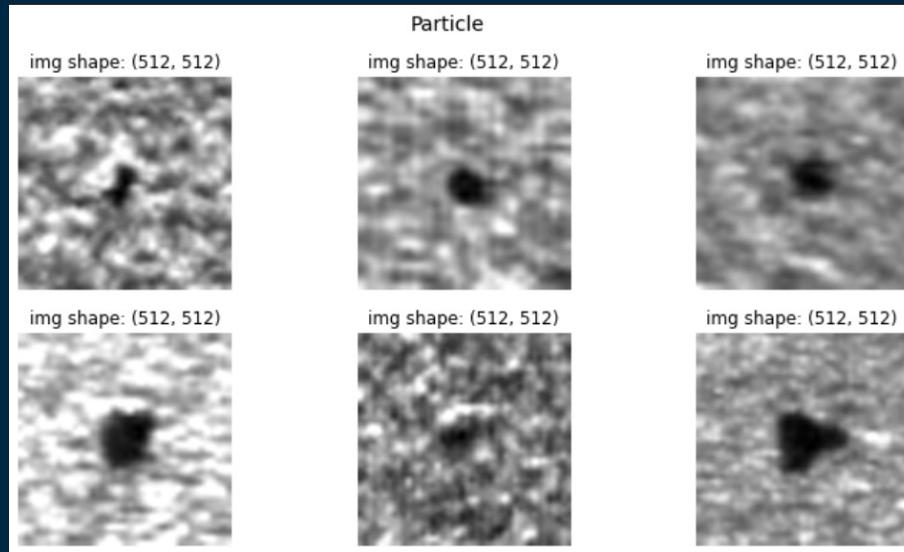
Label 3 : vertical defect, 垂直方向瑕疵

# 資料探索



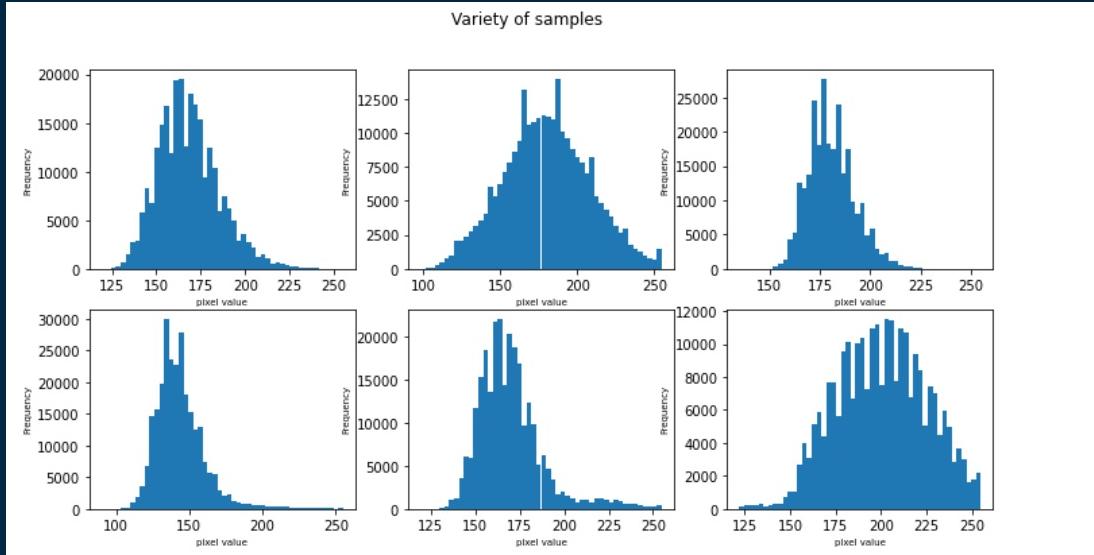
Label 4 : edge defect, 邊緣瑕疵

# 資料探索



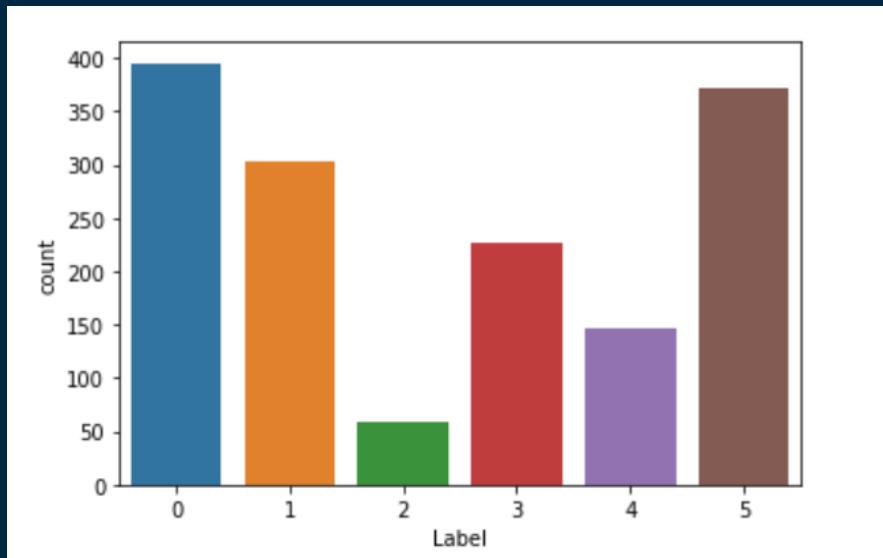
Label 5 : particle, 顆粒瑕疪

# 資料探索



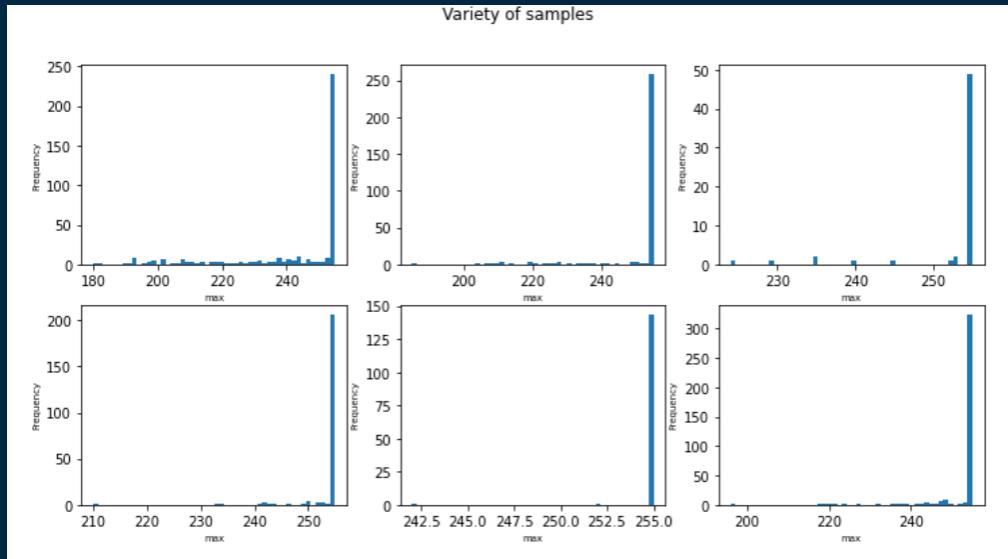
每個種類抽樣一張該張像素的分布

# 資料探索



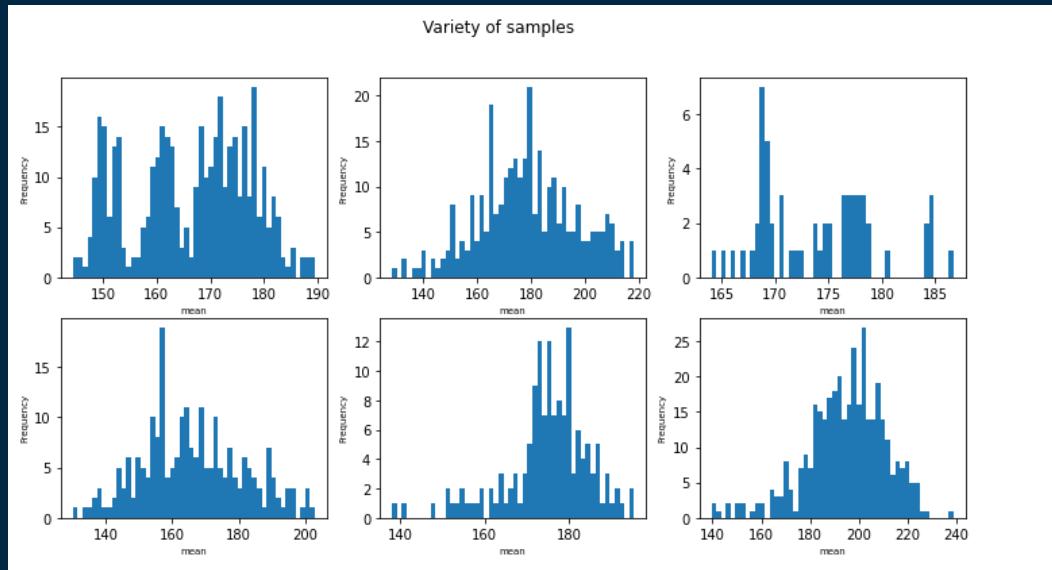
每個類別裡面有幾張照片

# 資料探索



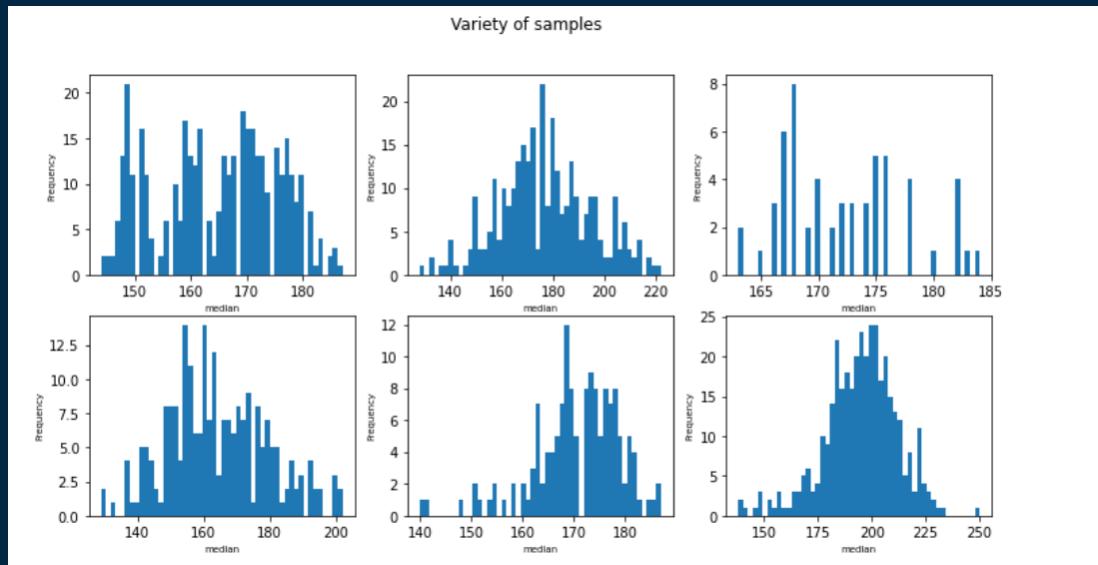
每個類別裡面每張照片的max

# 資料探索



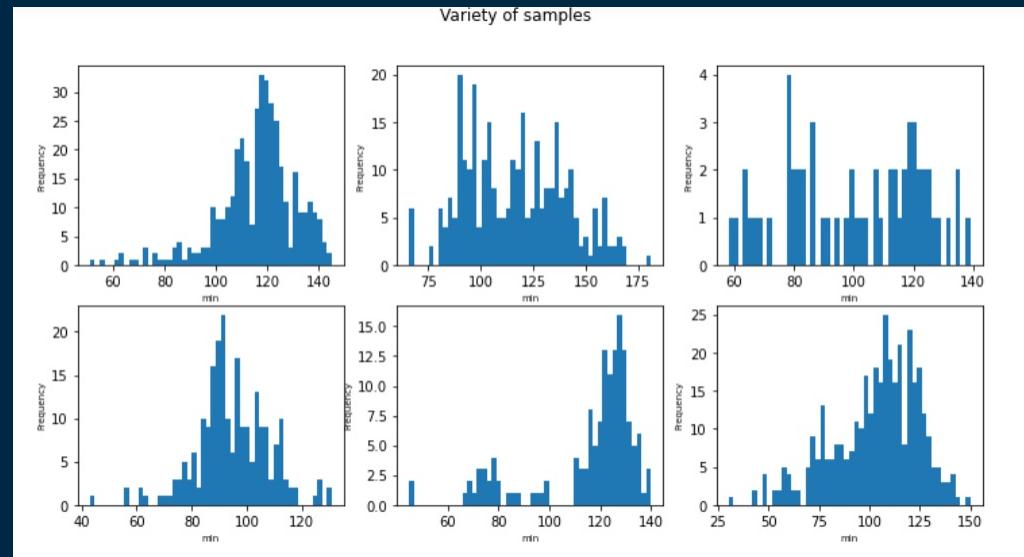
每個類別裡面每張照片的mean

# 資料探索



每個類別裡面每張照片的median

# 資料探索



每個類別裡面每張照片的min

# 研究方法

04

# 建模過程

- 我們將資料利用python中的mahotas package並使用函式haralic matrix篩選出影像0°、45°、90°、135°的13個features
- 其中，我們分別對影像做下列處理：
  - 將原始圖像四個方向的特徵向量取平均，當作最終的predictors
  - 影像切割 ( 2x2, 3x3, 4x4 ) 後取平均、變異、偏度、峰度
  - 像素合併 ( Average, Median, Gaussian, Bilateral ) 後取平均、變異、偏度、峰度
  - 詳細特徵創建部分將於個別模型成果處說明
- 接下來，我們利用Kfold( $k=5$ )對之後的模型進行交叉驗證(randomState訂為1082)
- 最後藉由confusion matrix、accuracy決定模型的優劣

# 13 Features Extracted by Haralick Matrix

1. Angular Second Moment [asm]
2. Contrast [con]
3. Correlation [cor]
4. Sum of Squares: Variance [var]
5. Inverse Difference Moment [idm]
6. Sum Average [sav]
7. Sum Variance [sva]
8. Sum Entropy [sen]
9. Entropy [ent]
10. Difference Variance [dva]
11. Difference Entropy [den]
- 12-13. Information Measures of Correlation [f12 and f13]

# 模型成果

05

原始圖像

# 特徵創建

(13 features)

feature 1

feature 2

feature 13

0°

45°

90°

135°

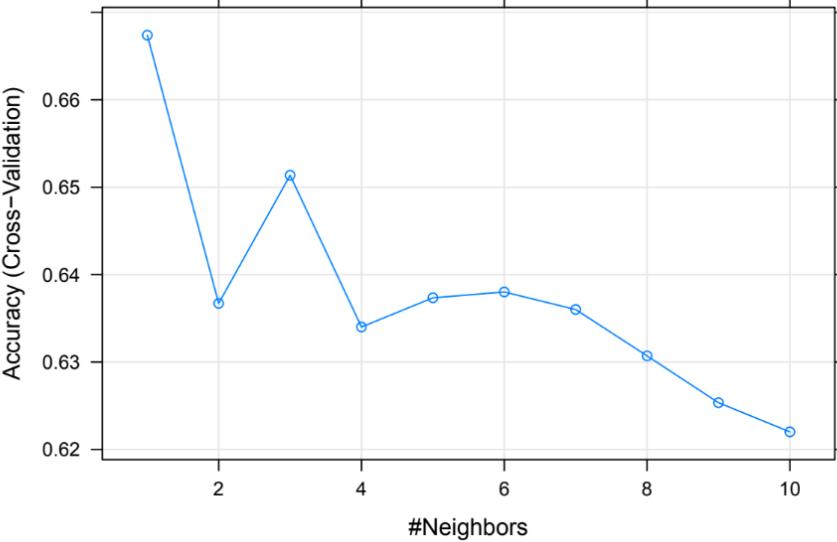
X0: 平均

X1: 平均

X12: 平均



# KNN



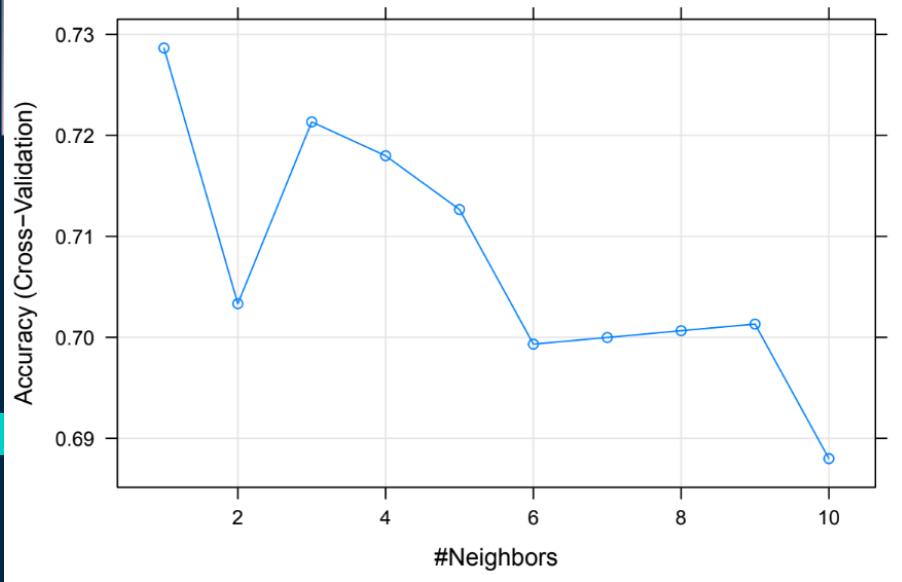
		Reference					
Prediction	0	1	2	3	4	5	
0	312	35	25	38	13	0	
1	28	131	0	19	47	51	
2	17	0	30	3	0	0	
3	34	19	3	149	9	2	
4	4	56	0	14	71	10	
5	0	62	0	4	6	308	

Accuracy (average) : 0.6673

We choose K=1 since the model has the highest accuracy.

Confusion matrix and model accuracy

# KNN (without $x_6$ )



		Reference					
Prediction	0	1	2	3	4	5	
0	314	28	24	31	8	0	
1	16	165	0	21	42	33	
2	26	0	30	1	0	0	
3	32	19	4	162	5	1	
4	7	44	0	12	88	3	
5	0	47	0	0	3	334	

Accuracy (average) : 0.7287

We choose K=1 since the model has the highest accuracy.

Confusion matrix and model accuracy

# Bagged CART

		Reference					
		0	1	2	3	4	5
Prediction	0	379	16	17	19	6	0
	1	5	261	0	17	13	4
2	2	0	41	1	0	0	0
3	9	10	0	177	8	0	0
4	0	8	0	13	119	0	0
5	0	8	0	0	0	0	367

Accuracy (average) : 0.896

Confusion matrix and model accuracy

# Bagged CART (without $x_6$ )

Reference							
Prediction	0	1	2	3	4	5	
0	375	15	17	18	0	0	
1	6	257	0	21	8	3	
2	6	0	41	0	0	0	
3	8	14	0	178	6	0	
4	0	9	0	10	132	0	
5	0	8	0	0	0	368	

Accuracy (average) : 0.9007

Confusion matrix and model accuracy

# Linear Discriminant Analysis

		Reference					
		0	1	2	3	4	5
Prediction	0	283	11	14	64	7	0
	1	55	256	0	16	1	0
		35	0	40	4	0	0
		22	5	4	109	7	0
		0	13	0	34	131	0
		0	18	0	0	0	371

Accuracy (average) : 0.7933

Confusion matrix and model accuracy

# Linear Discriminant Analysis (without $x_6$ )

Reference							
Prediction	0	1	2	3	4	5	
0	279	11	15	63	8	0	
1	55	255	0	18	3	0	
2	39	0	36	5	0	0	
3	22	5	7	106	7	0	
4	0	14	0	35	128	0	
5	0	18	0	0	0	371	

Accuracy (average) : 0.7833

Confusion matrix and model accuracy

# Quadratic Discriminant Analysis (without $x_6$ )

		Reference					
Prediction	0	0	1	2	3	4	5
		304	7	5	34	1	0
1	20	258	0	17	4	1	
2	57	0	52	10	0	0	
3	14	10	1	159	9	0	
4	0	16	0	7	132	0	
5	0	12	0	0	0	370	

Accuracy (average) : 0.85

Confusion matrix and model accuracy

# Naive Bayes

		Reference					
		0	1	2	3	4	5
Prediction	0	168	1	13	34	1	0
	1	67	174	0	13	13	4
2	98	0	43	32	0	0	
3	38	17	2	125	21	0	
4	24	93	0	23	111	0	
5	0	18	0	0	0	367	

Accuracy (average) : 0.6587

Confusion matrix and model accuracy

# Naive Bayes (without $x_6$ )

		Reference					
		0	1	2	3	4	5
Prediction	0	150	1	12	34	0	0
	1	77	170	0	14	12	6
2	90	0	43	25	0	0	
3	53	18	3	134	22	0	
4	25	93	0	20	112	0	
5	0	21	0	0	0	365	

Accuracy (average) : 0.6493

Confusion matrix and model accuracy

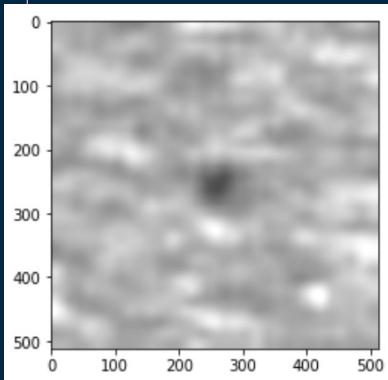
# Result Summary

Model	Predictor	Parameter	Accuracy
KNN	All Predictors	K=1	0.6674
KNN	Without $X_6$	K=1	0.7287
Bagged CART	All Predictors	Default	0.896
Bagged CART	Without $X_6$	Default	0.9007
LDA	ALL Predictors	Default	0.7933
LDA	Without $X_6$	Default	0.7833
QDA	Without $X_6$	Default	0.85
Naive Bayes	ALL Predictors	Default	0.6587
Naive Bayes	Without $X_6$	Default	0.6493

# 影像切割

# 切割方法

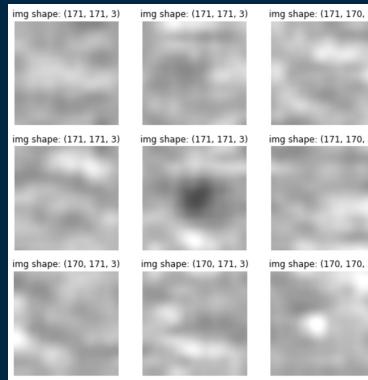
original



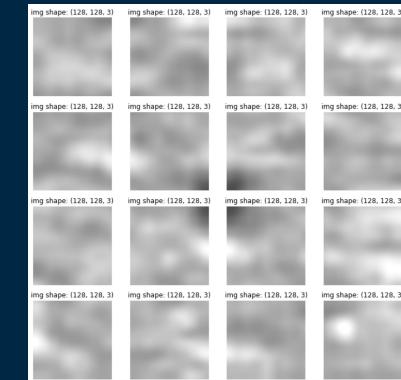
2x2



3x3



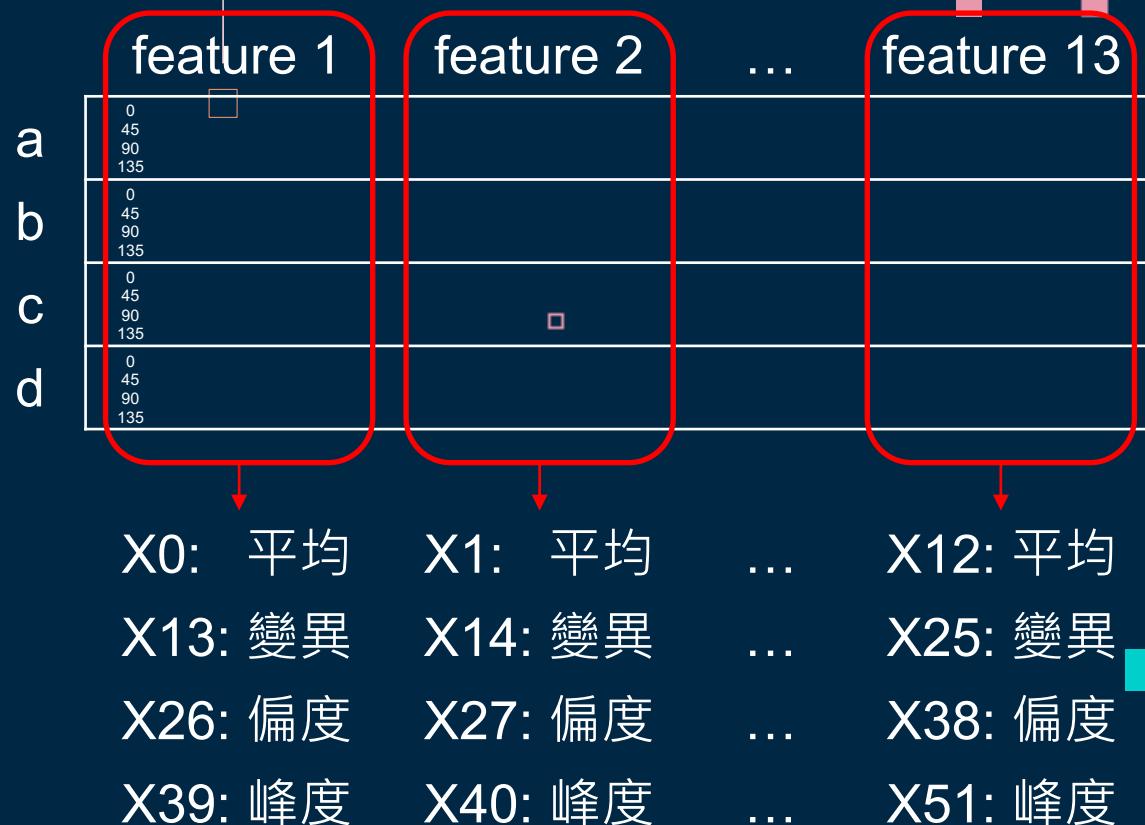
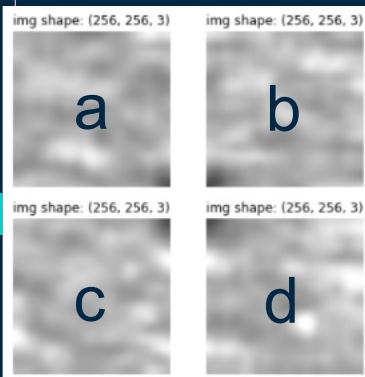
4x4



# 特徵創建

(52 features)

# 2x2為例



# 主成分分析

- 1. 計算所有變數的相關係數矩陣，進行主成分分析
- 2. 使用主成分分析的負荷量，計算出所有的主成分
- 3. 選擇15個主成分(可解釋約87%的變異)
- 4. 使用15個主成分去建立LDA、QDA模型

# 主成分分析

- 5. 用測試資料變數和訓練資料計算出來的負荷量，  
計算出測試資料的前15個主成分
- 6. 用測試資料的前15個主成分，套入之前所訓練的模型，  
進行預測

# Forward Selection

```
allNames <- names(data[, 1:52])
allVar <- paste("~", paste(allNames, collapse = " + "))

multi.fit = multinom(Label ~ 1, data = data, trace = F)
stepAIC(multi.fit, direction = "forward", trace = FALSE, scope = allVar)
```

- stepAIC: 透過AIC並依照**forward selection**去挑選變數加入到模型中

# 2x2切割 模型成果

15 principal components can explain 89.14% variation.

Model	Predictor	Parameter	Accuracy
mode	52		0.9553
Boosting Tree	52	n.trees=150, depth=3, shrinkage=0.1	0.9450
Forward Selection	52		0.9447
Random Forest	52	mtry=27	0.9300
Lasso	52	lambda=0.0031	0.9213
SVM	52	sigma=0.0179,C=1	0.8990
Penalized Multinomial Regression	52	decay=1e-0.4	0.8980
QDA	15 PCs	Default	0.8600
LDA	15 PCs	Default	0.7900
Naïve Bayes	52	Default	0.7690
KNN	52	K=5	0.5790

# 3x3切割 模型成果

15 principal components can explain 89.14% variation.

Model	Predictor	Parameter	Accuracy	Testing
mode	52		0.9633	
Forward Selection	52		0.9587	0.947
Random Forest	52	mtry=27	0.9540	
Boosting Tree	52	n.trees=150, depth=3, shrinkage=0.1	0.9540	
Lasso	52	lambda=0.0031	0.9373	0.934
SVM	52	sigma=0.0179,C=1	0.9280	
Penalized Multinomial Regression	52	decay=1e-0.4	0.9140	0.928
QDA	15 PCs	Default	0.8960	
LDA	15 PCs	Default	0.8420	0.855
Naïve Bayes	52	Default	0.8250	0.824
KNN	52	K=5	0.6170	0.632

# 4x4切割 模型成果

15 principal components can explain 89.14% variation.

Model	Predictor	Parameter	Accuracy
Boosting Tree	52	n.trees=150, depth=3, shrinkage=0.1	0.9610
Forward Selection	52		0.9593
mode	52		0.9593
Random Forest	52	mtry=27	0.9550
Lasso	52	lambda=0.0031	0.9413
Penalized Multinomial Regression	52	decay=0.1	0.9170
SVM	52	sigma=0.0189,C=1	0.9100
QDA	15 PCs	Default	0.9030
Naïve Bayes	52	Default	0.8330
LDA	15 PCs	Default	0.8270
KNN	52	K=5	0.6150

# 特徵創建

(每張子圖 a, b, ..., p 計算 52 features)



4x4

0°  
45°  
90°  
135°

feature 1

feature 2

feature 13

X0: 平均   X1: 平均   ...   X12: 平均  
X13: 變異   X14: 變異   ...   X25: 變異  
X26: 偏度   X27: 偏度   ...   X38: 偏度  
X39: 峰度   X40: 峰度   ...   X51: 峰度

# 特徵創建

(16子圖 x 52 features = 832 features)

4x4



a	$X_0, \dots, X_{51}$	$\rightarrow$	$X_0, \dots, X_{51}$
b	$X_0, \dots, X_{51}$	$\rightarrow$	$X_{52}, \dots, X_{103}$
...	...	...	...
p	$X_0, \dots, X_{51}$	$\rightarrow$	$X_{780}, \dots, X_{831}$

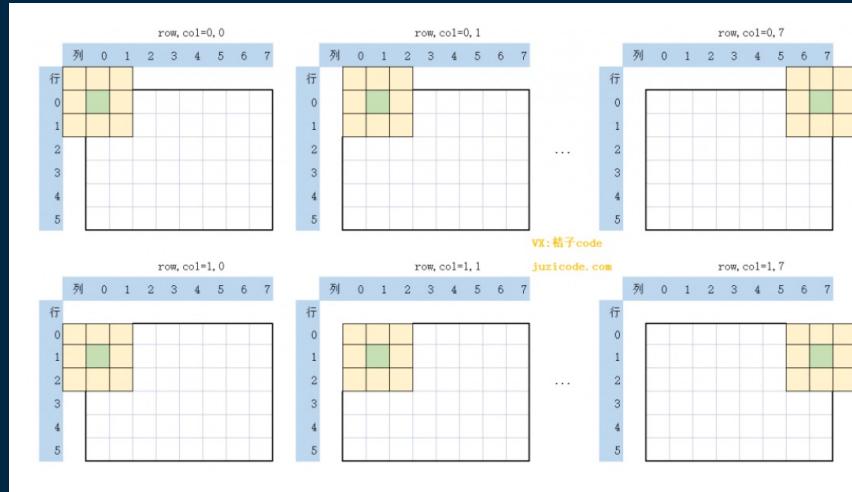
# 4x4切割 模型成果

Model	Predictor	Parameter	Accuracy
mode	832		0.9680
Forward Selection	832		0.9580
Random Forest	832	mtry=40	0.9407
Boosting Tree	832	n.trees=100, depth=3, shrinkage=0.1	0.9367
Lasso	832	lambda=0.0045	0.9073
QDA	19 PCs	Default	0.9067
SVM	832	sigma=0.0018, C = 1	0.9013
LDA	19 PCs	Default	0.7847
KNN	832	K=5	0.7727
Naïve Bayes	832	Default	0.7060

# 像素合併

# 平滑處理

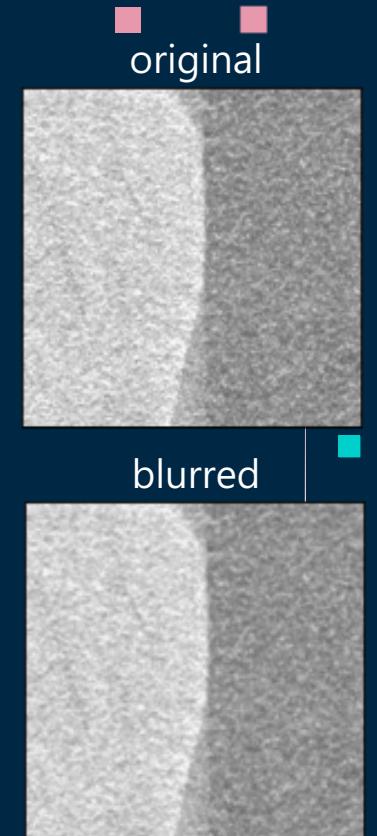
- 平滑處理用到“滑動窗口”的概念，下面以 $3 \times 3$ 的kernel為例，kernel會作用於原始圖像上的每一個像素，並沿著圖像逐個像素地進行計算，最終得到一幅新的類似於合併像素後的圖像。



# 平滑方法

## Average Blur

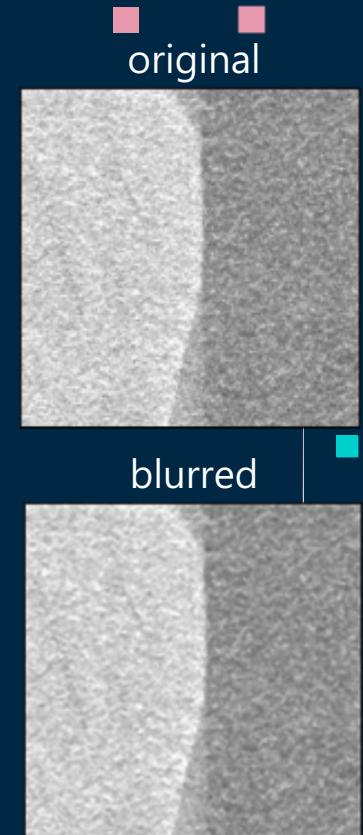
- 又稱為Normalized Box Filter
- 這是最簡單的一種模糊化方法
- 每個此kernel所在的最中央的那個點，  
我們把它調整為 $k \times k$ 窗口內所有其他點的平均值



# 平滑方法

## Median Blur

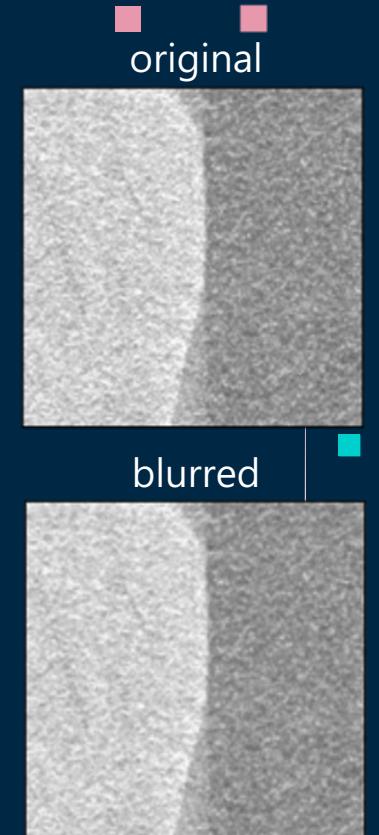
- 又稱Median Filter
- 此種模糊化的方法也經常應用於相片的除噪
- 每個此kernel所在的最中央的那個點，  
我們把它調整為 $k \times k$ 窗口內所有其他點的中位數



# 平滑方法

## Gaussian Blur

- 也稱為Gaussian filter
- 其作法是將給予各點不同的權值(color  $\sigma$ )，最後再以平均方式計算出中央點
- 其模糊化效果比起Averaging會比較明顯卻更自然



# 平滑方法

## Bilateral Blur

- 擁有Median filter的除噪效果
- 能保留圖片中的不同物件的輪廓
- 考慮像素之間幾何上的靠近程度(space  $\sigma$ )
- 考慮像素之間的光度及色彩差異(color  $\sigma$ )



# 特徵創建 (52 features)

feature 1

feature 2

feature 13

0°

45°

90°

135°

X0: 平均

X13: 變異

X26: 偏度

X39: 峰度

X1: 平均

X14: 變異

X27: 偏度

X40: 峰度

...

...

...

...

...

X12: 平均

X25: 變異

X38: 偏度

X51: 峰度

# average blur模型成果

Model	Predictor	Parameter	Accuracy
Penalized Logistic Regression	52	$\lambda = 0$	0.9520
KNN	52	K=5	0.7167
Random Forest	52	mtry = 27	0.9287
Boosting Tree	52	shrinkage = 0.1	0.9420
SVM	52	sigma = 0.0174, C = 1	0.9073

# median blur模型成果

Model	Predictor	Parameter	Accuracy
Penalized Logistic Regression	52	$\lambda = 0$	0.9460
KNN	52	K=5	0.7093
Random Forest	52	mtry = 2	0.9367
Boosting Tree	52	shrinkage = 0.1	0.9460
SVM	52	$\sigma = 0.0172, C = 1$	0.9153

# gaussian blur模型成果

Model	Predictor	Parameter	Accuracy
Penalized Logistic Regression	52	$\lambda = 0$	0.9560
KNN	52	K=5	0.7221
Random Forest	52	mtry = 2	0.9433
Boosting Tree	52	shrinkage = 0.1	0.9527
SVM	52	$\sigma = 0.0168, C = 1$	0.9013

# bilateral filter模型成果

Model	Predictor	Parameter	Accuracy
Penalized Logistic Regression	52	$\lambda = 0$	0.9467
KNN	52	K=5	0.6621
Random Forest	52	mtry = 2	0.9360
Boosting Tree	52	shrinkage = 0.1	0.9473
SVM	52	sigma = 0.0194, C = 1	0.9180

# 影像切割與像素合併

## 特徵創建 (364 features)

將前述的三種影像切割與四種像素合併方法所取得之各52個特徵合併

- 2x2影像切割 (X0, ..., X51)
- 3x3影像切割 (X52, ..., X103)
- 4x4影像切割 (X104, ..., X155)
- Average Blur (X156, ..., X207)
- Bilateral Blur (X208, ..., X259)
- Gaussian Blur (X260, ..., X311)
- Median Blur (X312, ..., X363) --> 共 $52 \times 7 = 364$  features

# 模型成果

Model	Predictor	Parameter	Accuracy
LASSO	364	lambda=0.0023	0.9780
Forward selection	364		0.9713
mode	364		0.9660
Boosting Tree	364	n.trees=150, depth=3, shrinkage=0.1	0.9607
Random Forest	364	mtry=183	0.9433
SVM	364	sigma=0.00189,C=1	0.8727
QDA	37 PCs	Default	0.8713
LDA	37 PCs	Default	0.8540
Naïve Bayes	364	Default	0.8047
KNN	364	K=5	0.6947

# 整體成果

# 整體成果 – 依全部模型之Accuracy排序

Method	Model	Predictor	Parameter	Accuracy
crop and blur	LASSO	364	lambda=0.0023	0.9780
crop and blur	Forward selection	364		0.9713
4x4 all	mode	832		0.9680
crop and blur	mode	364		0.9660
3x3	mode	52		0.9633
4x4	Boosting Tree	52	n.trees=150, depth=3, shrinkage=0.1	0.9610
crop and blur	Boosting Tree	364	n.trees=150, depth=3, shrinkage=0.1	0.9607
4x4	Forward Selection	52		0.9593
4x4	mode	52		0.9593
3x3	Forward Selection	52		0.9587

# 整體成果 – 各方法下之Accuracy排序

KNN

Naïve Bayes

LDA

QDA

Penalized Multinomial Regression

SVM

Random Forest

Boosting Tree

Mode

Forward selection

Lasso



# 整體成果 – KNN

Method	Predictor	Parameter	Accuracy
4x4 all	832	K=5	0.7727
gaussian	52	K=5	0.7221
average	52	K=5	0.7167
median	52	K=5	0.7093
crop and blur	364	K=5	0.6947
bilateral	52	K=5	0.6621
3x3	52	K=5	0.6170
4x4	52	K=5	0.6150
2x2	52	K=5	0.5790

# 整體成果 – Naïve Bayes

Method	Predictor	Parameter	Accuracy
4x4	52	Default	0.8330
3x3	52	Default	0.8250
crop and blur	364	Default	0.8047
2x2	52	Default	0.7690
4x4 all	832	Default	0.7060

# 整體成果 – LDA

Method	Predictor	Parameter	Accuracy
crop and blur	37 PCs	Default	0.8540
3x3	15 PCs	Default	0.8420
4x4	15 PCs	Default	0.8270
2x2	15 PCs	Default	0.7900
4x4 all	19 PCs	Default	0.7847

# 整體成果 – QDA

Method	Predictor	Parameter	Accuracy
4x4 all	19 PCs	Default	0.9067
4x4	15 PCs	Default	0.9030
3x3	15 PCs	Default	0.8960
crop and blur	37 PCs	Default	0.8713
2x2	15 PCs	Default	0.8600

# 整體成果 – Penalized Multinomial Regression

Method	Predictor	Parameter	Accuracy
4x4	52	decay=0.1	0.9170
3x3	52	decay=0.0001	0.9140
2x2	52	decay=0	0.8980

# 整體成果 – SVM

Method	Predictor	Parameter	Accuracy
3x3	52	sigma=0.0179, C=1	0.9280
bilateral	52	sigma = 0.0194, C=1	0.9180
median	52	sigma = 0.0172, C=1	0.9153
4x4	52	sigma=0.0189, C=1	0.9100
average	52	sigma = 0.0174, C=1	0.9073
4x4 all	832	sigma=0.0018, C=1	0.9013
gaussian	52	sigma = 0.0168, C=1	0.9013
2x2	52	sigma=0.0179, C=1	0.8990
crop and blur	364	sigma=0.0189, C=1	0.8727

# 整體成果 – Random Forest

Method	Predictor	Parameter	Accuracy
4x4	52	mtry=27	0.9550
3x3	52	mtry=27	0.9540
crop and blur	364	mtry=183	0.9433
gaussian	52	mtry = 2	0.9433
4x4 all	832	mtry=40	0.9407
median	52	mtry = 2	0.9367
bilateral	52	mtry = 2	0.9360
2x2	52	mtry=27	0.9300
average	52	mtry = 27	0.9287

# 整體成果 – Boosting Tree

Method	Predictor	Parameter	Accuracy
4x4	52	n.trees=150, depth=3, shrinkage=0.1	0.9610
crop and blur	364	n.trees=150, depth=3, shrinkage=0.1	0.9607
3x3	52	n.trees=150, depth=3, shrinkage=0.1	0.9540
gaussian	52	shrinkage = 0.1	0.9527
bilateral	52	shrinkage = 0.1	0.9473
median	52	shrinkage = 0.1	0.9460
2x2	52	n.trees=150, depth=3, shrinkage=0.1	0.9450
average	52	shrinkage = 0.1	0.9420
4x4 all	832	n.trees=100, depth=3, shrinkage=0.1	0.9367

# 整體成果 – Mode

Method	Predictor	Parameter	Accuracy
4x4 all	832		0.9680
crop and blur	364		0.9660
3x3	52		0.9633
4x4	52		0.9593
2x2	52		0.9553

# 整體成果 – Forward selection

Method	Predictor	Parameter	Accuracy
crop and blur	364	AIC: 448.9225	0.9713
4x4	52	AIC: 561.7828	0.9593
3x3	52	AIC: 592.634	0.9587
4x4 all	832		0.9580
2x2	52	AIC: 669.6916	0.9447

# 整體成果 – Lasso

Method	Predictor	Parameter	Accuracy
crop and blur	364	lambda=0.0023	0.9780
4x4	52	lambda=0.0031	0.9413
3x3	52	lambda=0.0031	0.9373
2x2	52	lambda=0.0031	0.9213
4x4 all	832	lambda=0.0045	0.9073

# CNN卷積神經網路

# 卷積神經網路-資料前處理

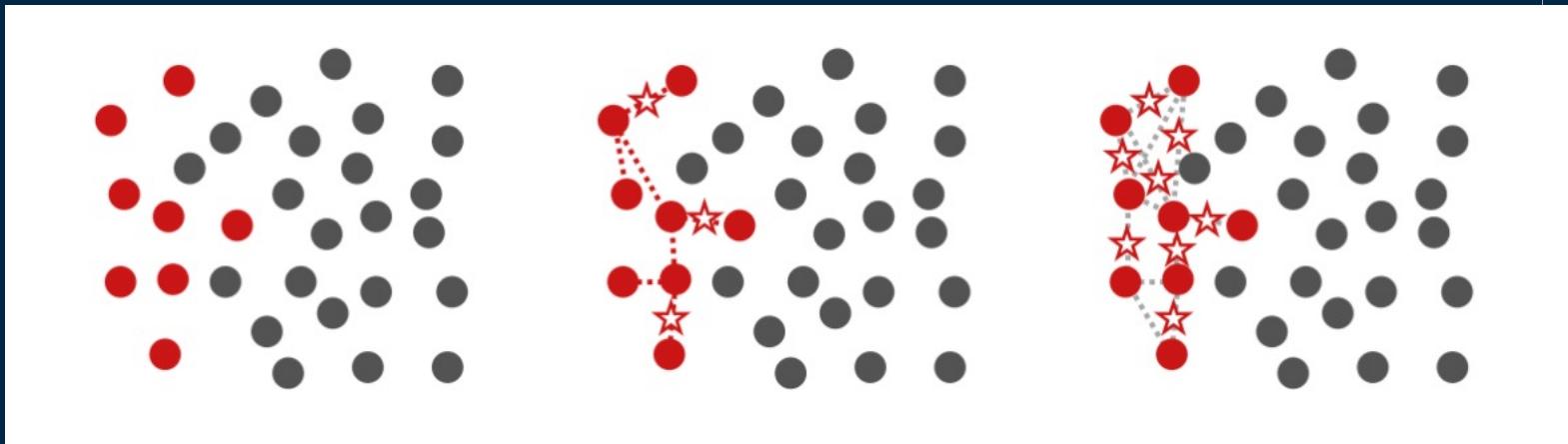
- - 1. 將原先 $512 \times 512$ 的圖片resize為 $128 \times 128$ 
    - 由於要縮小圖片，我們使用區域插值法
  - 2. 將 $128 \times 128$ 的圖片的所有像素進行正規化
    - 我們將每個像素除以255，使CNN在優化過程較順利
  - 3. 利用SMOTE改善資料各類別個數不平衡的問題

# 卷積神經網路-SMOTE(Oversampling)

- 1. 演算法的概念與實作簡單
- 2. 與Resampling相比擁有較低的解釋性

$$\Delta = \mathbf{x}_j - \mathbf{x}_i$$

$$\mathbf{x}_i^{(new)} = \mathbf{x}_i + \eta\Delta$$



# 卷積神經網路—模型訓練方法與配適方法

- - 1. 優化器訂為Adam，初始learning rate設為0.001
  - 2. Loss function訂為sparse\_categorical\_crossentropy
  - 3. Metrics僅使用accuracy做為評估標準
  - 4. 配適模型：
    - 20%的資料當驗證集，並設定30個epoch，每次看32筆資料

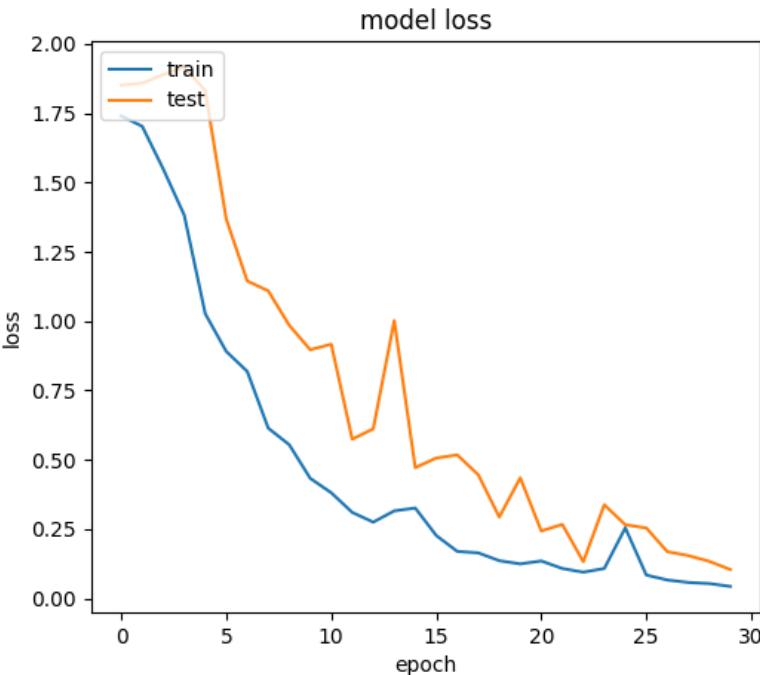
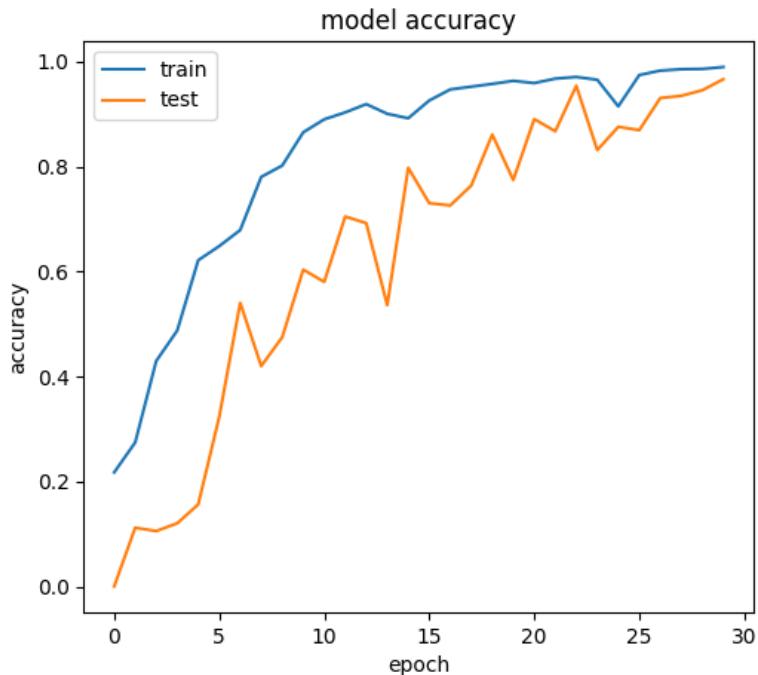
# 卷積神經網路—模型結構



$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

Layer (type)	Output Shape	Param #	
conv2d (Conv2D)	(None, 126, 126, 6)	60	Activation: relu
max_pooling2d (MaxPooling2D)	(None, 63, 63, 6)	0	
conv2d_1 (Conv2D)	(None, 61, 61, 16)	880	Activation: relu
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 16)	0	
flatten (Flatten)	(None, 14400)	0	
dense (Dense)	(None, 120)	1728120	
dense_1 (Dense)	(None, 84)	10164	
dense_2 (Dense)	(None, 6)	510	
activation (Activation)	(None, 6)	0	Activation: softmax(class=6)

# 卷積神經網路—Accuracy & Loss (ReLU)



# 卷積神經網路—Accuracy & Loss (RELU)

epoch: 1	loss: 1.717	accuracy: 0.2595	val_loss: 2.5977	val_accuracy: 0.0464
epoch: 2	loss: 1.5191	accuracy: 0.3745	val_loss: 2.099	val_accuracy: 0.0211
epoch: 3	loss: 1.2177	accuracy: 0.5585	val_loss: 1.9411	val_accuracy: 0.135
epoch: 4	loss: 0.9895	accuracy: 0.6334	val_loss: 2.2987	val_accuracy: 0.1329
epoch: 5	loss: 0.8268	accuracy: 0.6835	val_loss: 1.2905	val_accuracy: 0.3586
epoch: 6	loss: 0.686	accuracy: 0.7558	val_loss: 1.8573	val_accuracy: 0.3797
epoch: 7	loss: 0.6352	accuracy: 0.7706	val_loss: 1.5216	val_accuracy: 0.4304
epoch: 8	loss: 0.5745	accuracy: 0.7927	val_loss: 0.7788	val_accuracy: 0.6224
epoch: 9	loss: 0.5232	accuracy: 0.8117	val_loss: 1.0115	val_accuracy: 0.5084
epoch: 10	loss: 0.4736	accuracy: 0.8286	val_loss: 0.5652	val_accuracy: 0.7342
epoch: 11	loss: 0.4612	accuracy: 0.8386	val_loss: 0.7231	val_accuracy: 0.7004
epoch: 12	loss: 0.3184	accuracy: 0.8914	val_loss: 0.7558	val_accuracy: 0.6519
epoch: 13	loss: 0.2967	accuracy: 0.894	val_loss: 0.5602	val_accuracy: 0.7131
epoch: 14	loss: 0.3057	accuracy: 0.8924	val_loss: 0.5268	val_accuracy: 0.7658
epoch: 15	loss: 0.27	accuracy: 0.9056	val_loss: 0.5251	val_accuracy: 0.7658

epoch: 16	loss: 0.2064	accuracy: 0.9383	val_loss: 0.8268	val_accuracy: 0.7131
epoch: 17	loss: 0.1843	accuracy: 0.9409	val_loss: 0.4017	val_accuracy: 0.7869
epoch: 18	loss: 0.1385	accuracy: 0.9536	val_loss: 0.4795	val_accuracy: 0.7574
epoch: 19	loss: 0.1238	accuracy: 0.9647	val_loss: 0.2488	val_accuracy: 0.8755
epoch: 20	loss: 0.105	accuracy: 0.9715	val_loss: 0.1743	val_accuracy: 0.9283
epoch: 21	loss: 0.0829	accuracy: 0.9742	val_loss: 0.3837	val_accuracy: 0.8143
epoch: 22	loss: 0.0722	accuracy: 0.9826	val_loss: 0.2283	val_accuracy: 0.9051
epoch: 23	loss: 0.0798	accuracy: 0.9778	val_loss: 0.2109	val_accuracy: 0.8903
epoch: 24	loss: 0.0609	accuracy: 0.9831	val_loss: 0.2436	val_accuracy: 0.8692
epoch: 25	loss: 0.0442	accuracy: 0.9889	val_loss: 0.1657	val_accuracy: 0.9367
epoch: 26	loss: 0.0382	accuracy: 0.99	val_loss: 0.0566	val_accuracy: 0.9916
epoch: 27	loss: 0.0357	accuracy: 0.991	val_loss: 0.1235	val_accuracy: 0.9515
epoch: 28	loss: 0.0259	accuracy: 0.9953	val_loss: 0.078	val_accuracy: 0.9789
epoch: 29	loss: 0.0302	accuracy: 0.9953	val_loss: 0.0279	val_accuracy: 0.9937
epoch: 30	loss: 0.0217	accuracy: 0.9958	val_loss: 0.0874	val_accuracy: 0.9747

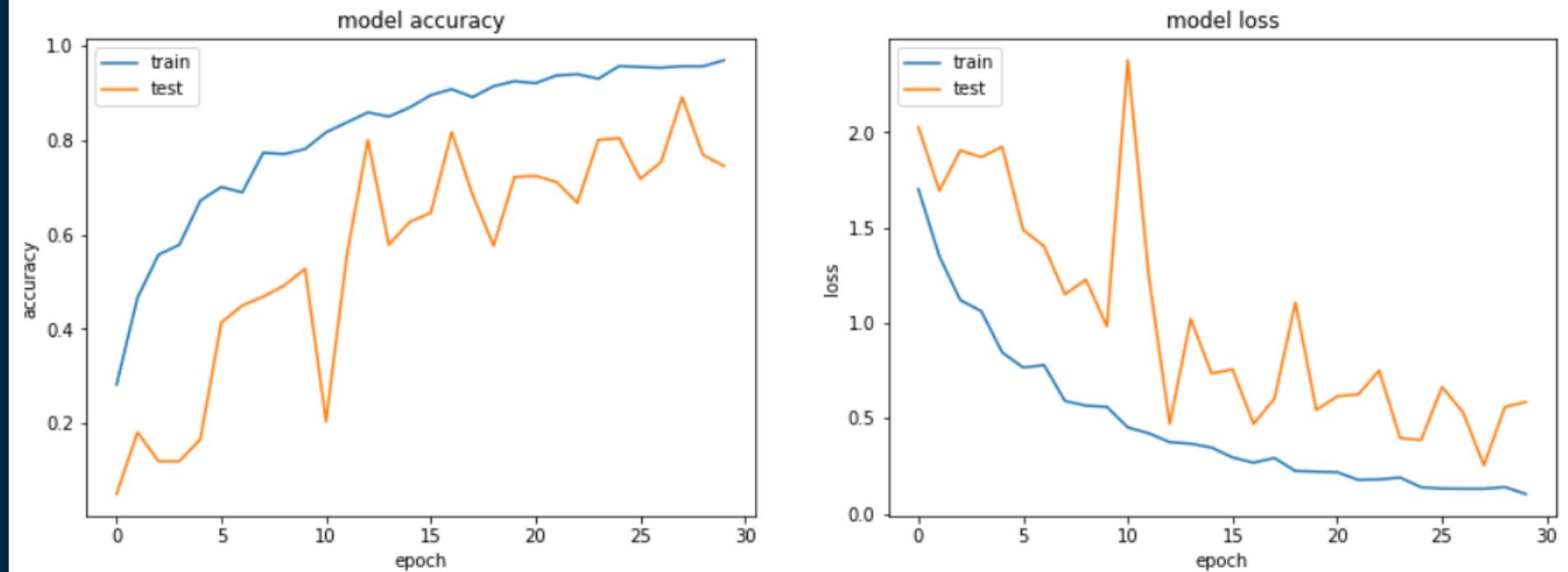
# 卷積神經網路—模型結構



$$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$$

Layer (type)	Output Shape	Param #	
conv2d_2 (Conv2D)	(None, 126, 126, 6)	60	Activation: tanh
max_pooling2d_2 (MaxPooling 2D)	(None, 63, 63, 6)	0	
conv2d_3 (Conv2D)	(None, 61, 61, 16)	880	Activation: tanh
max_pooling2d_3 (MaxPooling 2D)	(None, 30, 30, 16)	0	
flatten_1 (Flatten)	(None, 14400)	0	
dense_3 (Dense)	(None, 120)	1728120	
dense_4 (Dense)	(None, 84)	10164	
dense_5 (Dense)	(None, 6)	510	
activation_1 (Activation)	(None, 6)	0	Activation: softmax(class=6)

# 卷積神經網路—Accuracy & Loss (TANH)



謝謝大家

# 討論紀錄

06

## 討論紀錄列表

日期	時間	地點	參與名單
03.10	19:00-23:45	D-24	全體組員
04.02	21:00-23:00	線上 (Zoom)	全體組員
04.09	19:00-23:00	全家 台南常榮店	全體組員
04.09	23:00-23:30	全家 台南常榮店	部分組員
04.10	13:00-18:00	全家 台南常榮店	部分組員
04.15	13:00-14:30	線上 (Google meet)	全體組員
04.15	20:00-00:00	線上 (Google meet)	部分組員
04.17	20:00-21:15	線上 (Google meet)	全體組員

# 討論紀錄列表

日期	時間	地點	參與名單
05.07	21:00-23:30	線上 (Google meet)	全體組員
05.09	19:00-23:50	線上 (Line)	全體組員
05.10	11:10-11:30	老師辦公室	部分組員
05.10	13:00-14:00	線上 (Line)	全體組員
05.12	19:30- <b>24:00</b>	D-24	全體組員
<b>05.13</b>	<b>00:00-01:00</b>	D-24	全體組員
05.13	14:00-16:30	線上 (Google meet)	全體組員
05.16	11:00-12:00	線上 (Line)	全體組員

## 討論紀錄列表

日期	時間	地點	參與名單
05.19	11:00-13:00	線上 (Line)	全體組員
05.28	19:00-23:30	線上 (Google meet)	全體組員
06.02	19:00-24:00	線上 (Google meet)	全體組員
06.03	00:00-02:00	線上 (Google meet)	全體組員
06.03	15:00-18:00	線上 (Google meet)	部分組員
06.04	13:00-16:00	線上 (Google meet)	全體組員

# 討論內容

# 03.10 討論紀錄

時間 | 19:00-23:45

地點 | D-24

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾莛勛 劉伊庭

討論內容 | 1. 決定組別名稱、分享文獻查找進度

2. 彙整文獻內容 ( 張健琳 劉貞莉 劉伊庭 )

3. 討論資料內容

4. 討論資料探索方向

5. 進行資料探索程式撰寫 ( 侯登耀 邱瑞麒 曾莛勛 )

6. 簡報編輯與設計 ( 張健琳 劉貞莉 )

# 04.02 討論紀錄

時間 | 21:00-23:00

地點 | 線上 (Zoom)

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾莛勛 劉伊庭

討論內容 | 主持：曾莛勛

1. 分享關鍵文獻閱讀進度
2. 討論關鍵文獻不明白之處
3. 訂定下次討論時間與進度

# 04.09 討論紀錄

時間 | 19:00-23:00

地點 | 全家 台南常榮店

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾莛勛 劉伊庭

討論內容 | 1. 彙整與製作關鍵論文簡報 ( 張健琳 劉貞莉 劉伊庭 )

2. 資料前處理 ( 侯登耀 邱瑞麒 曾莛勛 )

3. 特徵選擇與模型建立 ( 侯登耀 邱瑞麒 曾莛勛 )

# 04.09 討論紀錄

時間 | 23:00-23:30

地點 | 全家 台南常榮店

參與名單 | 張健琳 劉貞莉 劉伊庭

討論內容 | 因當日未完成關鍵文獻回顧，討論後續分工內容：

a. 彙整與製作關鍵論文簡報

- Textural Features for Image ( 劉伊庭 )
- Classification of Solder Joint ( 劉貞莉 )

b. 關鍵論文之簡報設計 ( 張健琳 )

# 04.10 討論紀錄

時間 | 13:00-18:00

地點 | 全家 台南常榮店

參與名單 | 侯登耀 邱瑞麒 曾莛勛

討論內容 | 1. 資料前處理

2. 尋找Haralic matrix套件

3. 尋找適合的分類器

# 04.15 討論紀錄

時間 | 13:00-14:30

地點 | 線上 (Google meet)

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾莛勛 劉伊庭

討論內容 | 主持：侯登耀

1. 彙報各自分工進度
2. 討論與修改第一次報告錯誤之處
3. 進行python程式撰寫（侯登耀 邱瑞麒 曾莛勛）
4. 新增研究結果簡報（劉貞莉、劉伊庭）
5. 簡報設計（張健琳）

# 04.15 討論紀錄

時間 | 20:00-00:00

地點 | 線上 (Google meet)

參與名單 | 侯登耀 邱瑞麒 曾莛勛

討論內容 | 1. 匯出Haralic所生成的13個features

2. 利用Kfold切割資料

3. 配適模型

4. 資料彙整

# 04.17 討論紀錄

時間 | 20:00-21:15

地點 | 線上 (Google meet)

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾庭勛 劉伊庭

討論內容 | 主持：侯登耀

1. 再次確認明日報告內容

2. 討論後續報告修正 (未來展望, Future Outlook) 內容

- 1. 實踐第二篇論文提到的特徵選擇
- 2. 對四個不同角度特徵向量的處理
- 3. 解決training data各種類資料不平衡的問題
- 4. 增加其他分類模型(e.g. SVM, K-means等)

# 05.07 討論紀錄

時間 | 21:00-23:30

地點 | 線上 (Google meet)

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾庭勛 劉伊庭

討論內容 | 主持：侯登耀

1. 討論前次報告需修正之處
2. 討論影像分割與像素合併之方式
3. 決定分工內容與下次討論時間

# 05.09 討論紀錄

時間 | 19:00-23:50

地點 | 線上 (Line)

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾庭勛 劉伊庭

討論內容 | 1. 確定影像切割大小

2. 確定像素合併方式

3. 撰寫影像切割程式碼

4. 撰寫像素合併程式碼

# 05.10 討論紀錄

時間 | 11:10-11:30

地點 | 老師辦公室

參與名單 | 張健琳 劉貞莉 劉伊庭

討論內容 | 1. 詢問影像切割後的觀察值個數

2. 詢問影像切割後特徵擷取方式

# 05.10 討論紀錄

時間 | 13:00-14:00

地點 | 線上 (Line)

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾庭勛 劉伊庭

討論內容 | 1. 完善影像切割程式碼

2. 完善像素合併程式碼

3. 提取影像切割後的特徵

4. 提取像素合併後的特徵

# 05.12 討論紀錄

時間 | 19:30-05.13 01:00

地點 | D-24

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾庭勛 劉伊庭

討論內容 | 1. 彙報個人分工進行狀況與統整

2. 撰寫預測模型程式碼

3. 套入模型進行預測

4. 彙整模型預測結果

5. 第三次報告簡報製作與設計

# 05.13 討論紀錄

時間 | 14:00-16:30

地點 | 線上 (Google meet)

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾庭勛 劉伊庭

討論內容 | 1. 彙整 training data 預測結果

2. 撰寫預測 testing data 程式碼

3. 套入模型進行 testing data 預測

4. 彙整 testing data 預測結果

5. 第三次報告簡報製作與設計

# 05.16 討論紀錄

時間 | 11:00-12:00

地點 | 線上 (Line)

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾庭勛 劉伊庭

討論內容 | 1. 討論今日報告改進之處

2. 討論如何正確應用PCA

3. 下次進度須完成所有變數 ( 52 個變數 ) 的模型配適

# 05.19 討論紀錄

時間 | 11:00-13:00

地點 | 線上 (Line)

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾庭勛 劉伊庭

討論內容 | 1. 確認PCA應用方式

2. 撰寫PCA篩選變數與建立模型之程式碼

# 05.28 討論紀錄

時間 | 19:00-23:30

地點 | 線上 (Google meet)

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾莛勛 劉伊庭

討論內容 | 1. 增加 forward selection

2. 修改模型並僅套用PCA於LDA和QDA

3. 重新套入模型進行 training data 預測

4. 彙整 training data 預測結果

5. 簡報修改

# 06.02 討論紀錄

時間 | 19:00-06.03 2:00

地點 | 線上 (Google meet)

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾庭勛 劉伊庭

討論內容 | 1. CNN模型資料前處理

2. 撰寫CNN程式碼

3. 新增整體成果表格

# 06.03 討論紀錄

時間 | 15:00-18:00

地點 | 線上 (Google meet)

參與名單 | 張健琳 侯登耀 邱瑞麒 曾莛勛 劉伊庭

討論內容 | 1. 處理CNN資料不平衡問題

2. 調整CNN模型網路架構

3. 新增CNN模型accuracy & loss

# 06.04 討論紀錄

時間 | 13:00-16:00

地點 | 線上 (Google meet)

參與名單 | 張健琳 劉貞莉 侯登耀 邱瑞麒 曾庭勛 劉伊庭

討論內容 | 1. 完成CNN模型

2. 新增本周討論紀錄

3. 練習報告內容

# 分工內容

# 個人分工內容

姓名 | 邱瑞麒

- 分工 |
1. 進行資料探索程式撰寫
  2. 資料前處理
  3. 特徵選擇與模型建立
  4. 尋找Haralick matrix套件、適合的分類器
  5. 進行python程式撰寫
  6. 匯出Haralick所生成的13個features
  7. 利用Kfold切割資料
  8. 配適模型、資料彙整

# 個人分工內容

姓名 | 邱瑞麒

- 分工 | 9. 撰寫像素合併程式碼  
10. 完善像素合併程式碼  
11. 提取像素合併後的特徵  
12. 撰寫預測模型程式碼  
13. 套入模型進行預測  
14. 負責第三次上台報告  
15. 撰寫影像切割與像素合併之測試資料label程式碼  
16. CNN模型建構

# 個人分工內容

姓名 | 曾庭勛

- 分工 |
1. 進行資料探索程式撰寫
  2. 資料前處理
  3. 特徵選擇與模型建立
  4. 尋找Haralick matrix套件、適合的分類器
  5. 進行python程式撰寫
  6. 匯出Haralick所生成的13個features
  7. 利用Kfold切割資料
  8. 配適模型、資料彙整

# 個人分工內容

姓名 | 曾庭勛

分工 | 9. 撰寫像素合併程式碼  
10. 完善像素合併程式碼  
11. 提取像素合併後的特徵  
12. 撰寫預測模型程式碼  
13. 修改PCA篩選變數程式碼  
14. 撰寫影像切割不合併特徵之測試資料label程式碼  
15. CNN模型建構  
16. 負責第四次上台報告

# 個人分工內容

姓名 | 侯登耀

分工 | 1. 進行資料探索程式撰寫  
2. 資料前處理  
3. 特徵選擇與模型建立  
4. 尋找Haralick matrix套件、適合的分類器  
5. 進行python程式撰寫  
6. 匯出Haralick所生成的13個features  
7. 利用Kfold切割資料  
8. 配適模型、資料彙整

# 個人分工內容

姓名 | 侯登耀

- 分工 | 9. 撰寫像素合併程式碼  
10. 完善像素合併程式碼  
11. 提取像素合併後的特徵  
12. 撰寫預測模型程式碼  
13. 套入模型進行預測  
14. 彙整 testing data 預測結果  
15. CNN模型建構  
16. 負責第四次上台報告

# 個人分工內容

姓名 | 張健琳

- 分工 |
1. 彙整文獻內容 (第一次報告)
  2. 簡報編輯與設計 (第一次報告)
  3. 負責第二次上台報告
  4. 關鍵論文之簡報設計 (第二次報告)
  5. 撰寫影像切割程式碼
  6. 提取影像切割後的特徵
  7. 套入模型進行預測
  8. 彙整模型預測結果

# 個人分工內容

姓名 | 張健琳

分工 | 9. 簡報製作與設計 (第三次報告)

10. 撰寫像素合併之測試資料特徵擷取程式碼

11. 彙整新建模型預測結果

12. CNN模型建構

# 個人分工內容

姓名 | 劉貞莉

- 分工 |
1. 彙整文獻內容 (第一次報告)
  2. 簡報編輯與設計 (第一次報告)
  3. 彙整與製作關鍵論文簡報 Classification of Solder Joint
  4. 撰寫影像切割程式碼
  5. 提取影像切割後的特徵
  6. 套入模型進行預測
  7. 彙整模型預測結果
  8. 簡報製作與設計 (第三次報告)

# 個人分工內容

姓名 | 劉貞莉

分工 | 9. 彙整 training data 預測結果  
10. 撰寫像素合併之測試資料特徵擷取程式碼  
11. 撰寫影像切割與像素合併之測試資料label程式碼  
12. 撰寫影像切割不合併特徵之測試資料label程式碼  
13. 負責第四次上台報告

# 個人分工內容

姓名 | 劉伊庭

分工 | 1. 彙整文獻內容 (第一次報告 )  
2. 彙整與製作關鍵論文簡報Textural Features for Image  
3. 撰寫影像切割程式碼  
4. 提取影像切割後的特徵  
5. 套入模型進行預測  
6. 彙整模型預測結果  
7. 簡報製作與設計 (第三次報告 )  
8. 撰寫影像切割之測試資料label程式碼

# 個人分工內容

姓名 | 劉伊庭

分工 | 9. CNN模型建構

10. 負責第四次上台報告