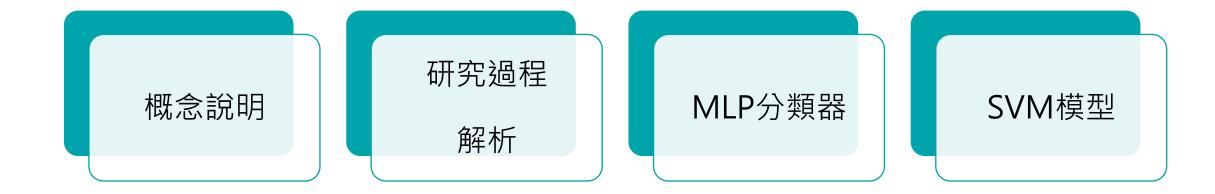
314337 類神經網路 期末報告 MLP分類器應用於瑕疵檢測

創新AI碩一 111C71008 何哲平



Outline 重點大綱





概念說明

研究動機及目的

為何選擇這主題

- A. MLP的實際應用 / 研究領域
 - → 作為模型成效的基本指標
- B. 支援工管所同學課程期末報告
 - →「機器視覺原理與應用」課程
 - → AOI影像 與 電路瑕疵檢測
- C. 對於MLP的興趣,更深入研究



研究動機及目的

Requirement:

判斷電路板是否為瑕疵

左:電路板照片VS.

• 右:電路板設計稿



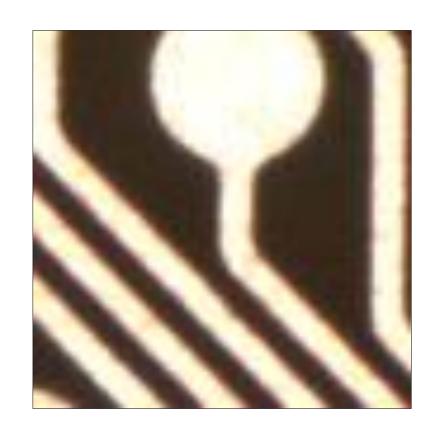


小試身手

• 解釋Label、說明瑕疵與正常的差異

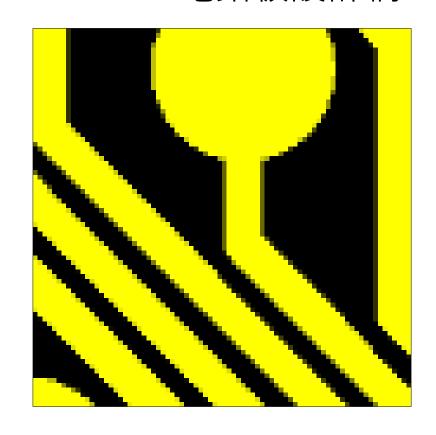
圖像預處理後

電路板照片



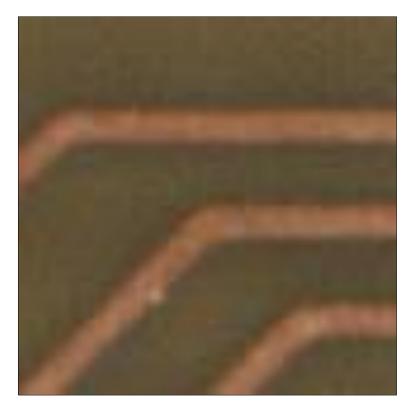
OK正常

電路板設計稿



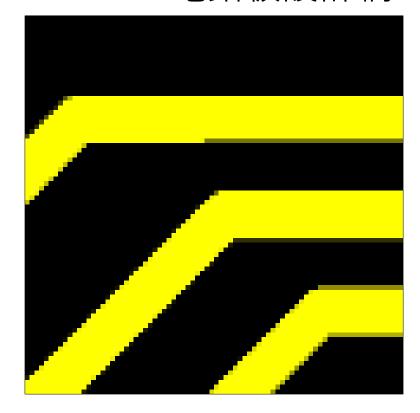
圖像預處理後

電路板照片



OK正常

電路板設計稿



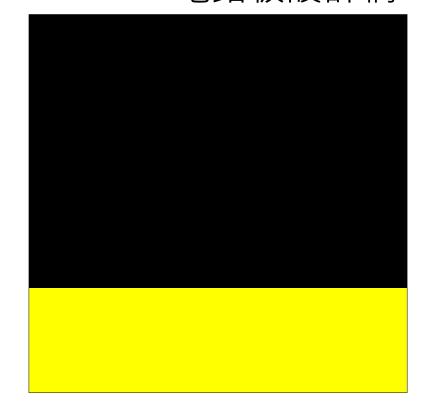
圖像預處理後

電路板照片



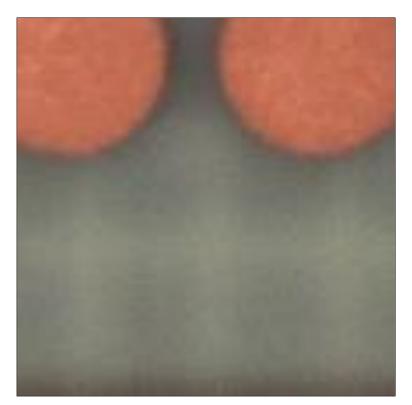
NG瑕疵

電路板設計稿



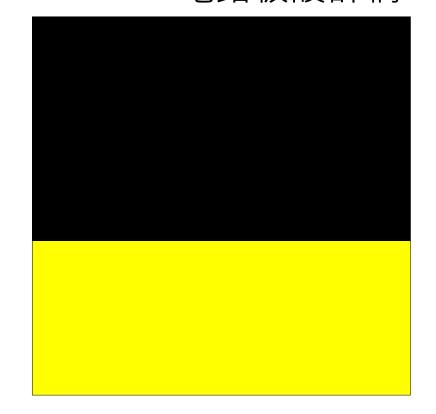
圖像預處理後

電路板照片



NG瑕疵

電路板設計稿



圖像預處理後

電路板照片



NG瑕疵

電路板設計稿



研究過程解析

資料處理流程

1.) 讀取

讀取電路板的圖像數據。

2.) 預處理

使用高斯濾波、邊緣強化、二值化,

以改善圖像質量。

3.) 形態學處理

使用膨脹、侵蝕等,

消除噪聲並突顯目標特徵。



4.) 檢測 及 結果顯示+保存

應用MLP多層感知器來檢測瑕疵·

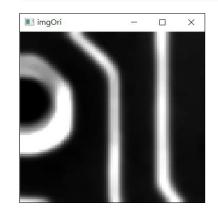
並保存分析記錄。

讀取圖像&前處理

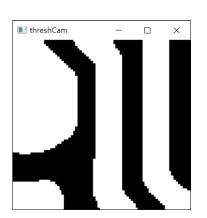


利用python cv2 套件

- 讀取資料
- 前處理
- ▶高斯濾波
- ▶邊緣強化
- ➤二值化 (Threshold)
- ▶去雜訊、增強比對



圖像比對

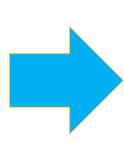


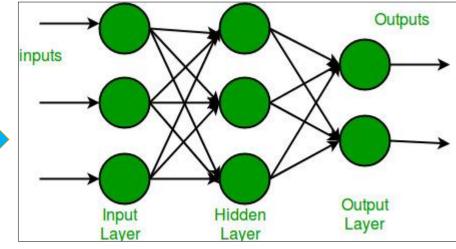
困難點:如何轉換資料型態??



HOW?

sepal_leng	sepal_wid1	petal_leng	petal_widt	species
5.1	3.5	1.4	0.2	setosa
4.6	BL	1.4	0.2	setosa
5.3	XV.	1221.5	0.2	setosa
5	3.3/	1 14	0.2	setosa
7	3.2	74	\$\\\	versicolor
6.4	3.2	4.5	さべい	versicolor
6.9	3.1	4.9	1.5	versicolor
5.5	2.3	4	1.3	versicolor





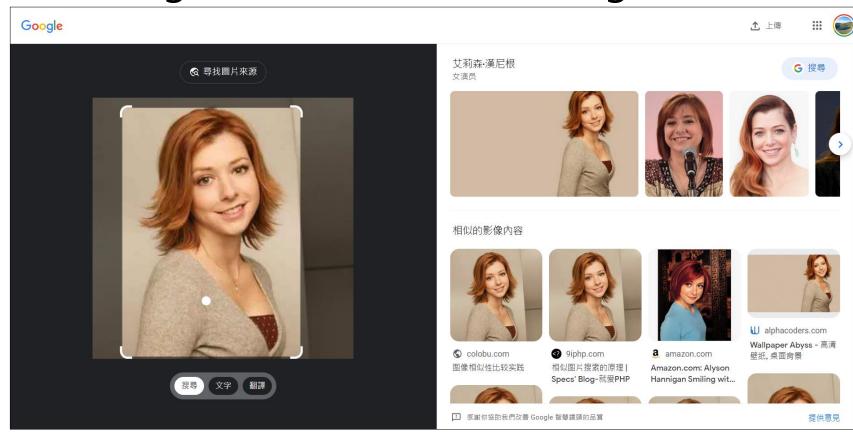
圖像相似度算法 Hash Algorithm

透過比對不同圖片之間的Hash值,計算出圖片之間的相似性。

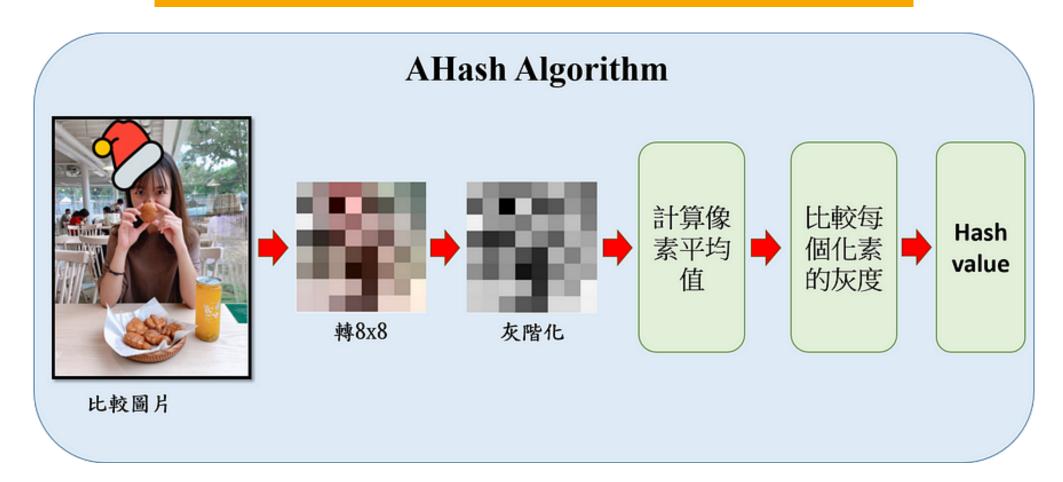
數值越小,表示兩張圖片的相似度越高;數值越大表示相似度越低。

- Ahash 均值雜湊演算法:採用圖片的平均像素為計算標準。
- Phash 感知雜湊演算法:計算低頻的均值哈希值來比對影像。
- Dhash 差值雜湊演算法:每一個像素點均與像素平均值進行比對。

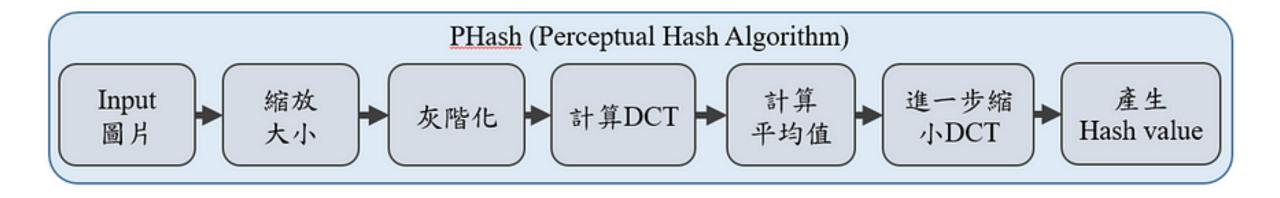
Hash Algorithm 實際應用:Google以圖搜圖



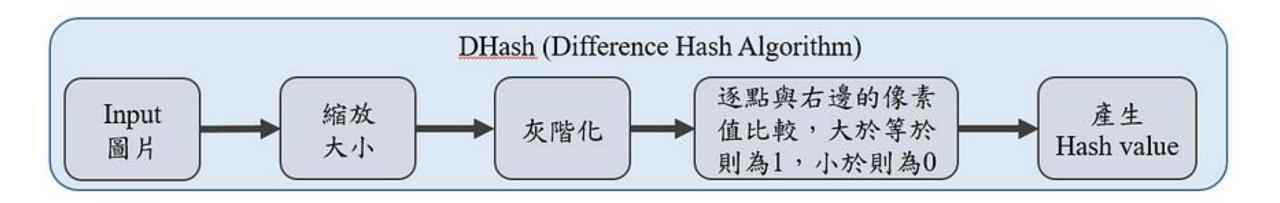
Ahash 均值雜湊演算法

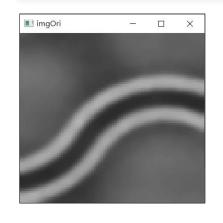


Phash 感知雜湊演算法

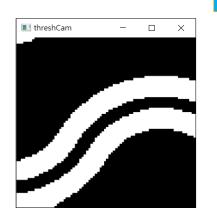


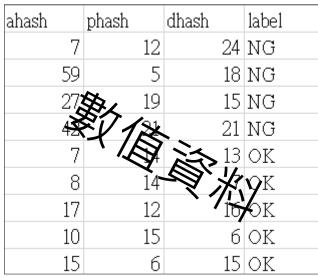
Dhash 差值雜湊演算法

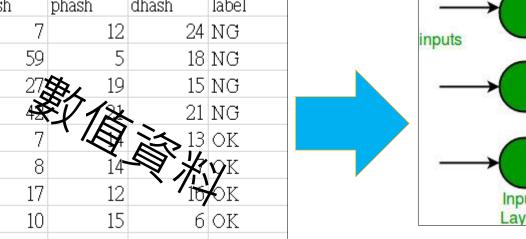


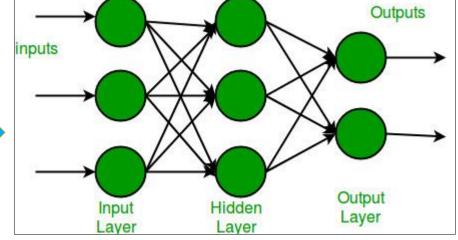


圖像比對





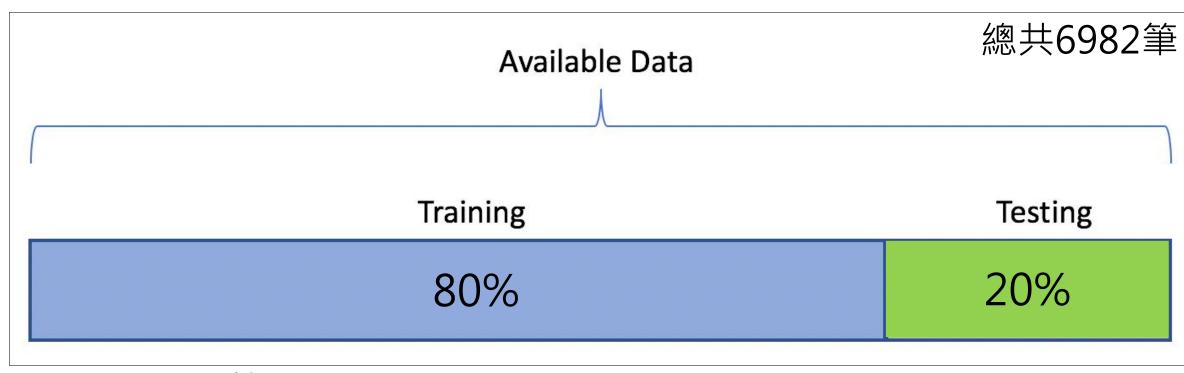




取得Hash數值資料

• 資料統計

label	count
OK	4565
NG	2417
Sum	6982



Train: 5585筆 Test: 1397筆

Parameters

• Train_Size : 80%

• Test_Size : 20%

• shuffle : True

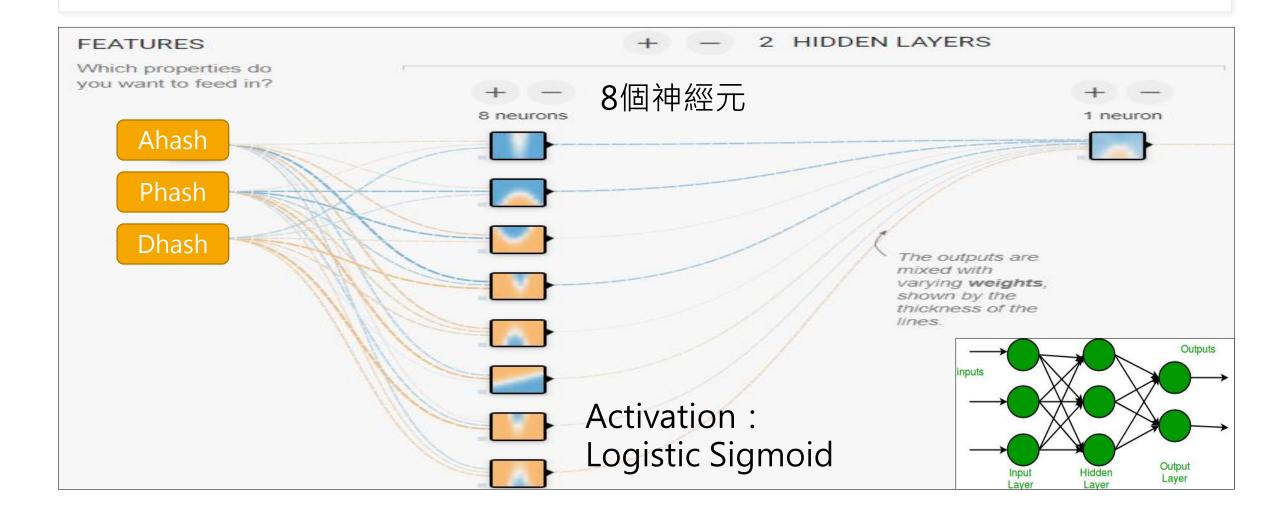
• stratify:抽樣比例依照原始label分布

```
# train_test_split
train, test = train_test_split(data, train_size=0.8, test_size = 0.2, shuffle=True, stratify=data['label'], random_state=1)
```

• 資料統計

label	Train 訓練集資料	Test 測試集資料
OK	3652	913
NG	1933	484
Sum	5585	1397

MLP Classifier 建立模型



MLP Classifier 建立模型

HyperParameters

- hidden_layer_sizes: 指定隱藏層層數+每層單元數 設定(8,) → 僅一層 8個神經元 的隱藏層。
- activation:隱藏層的激活函數
 設定 <u>logistic</u> sigmoid 函數 f(x) = 1 / (1 + exp(-x))
- solver:優化器 設定 adam 隨機梯度優化器

MLP Classifier 建立模型

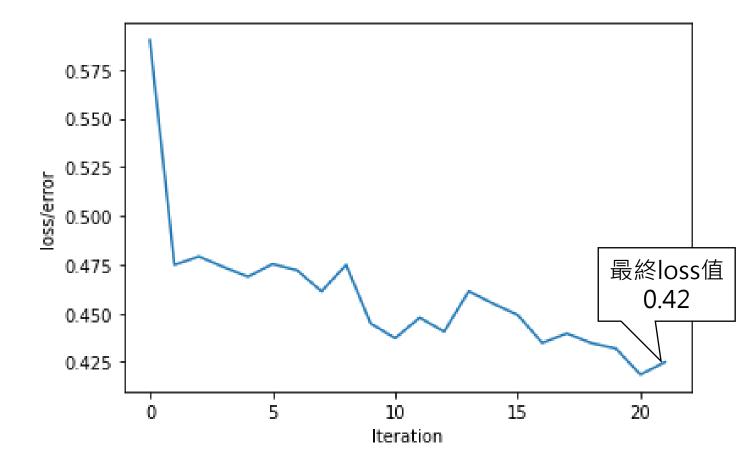
HyperParameters

Ŧ

- learning_rate \ learning_rate_init:設定學習率為恆定(constant)且數值為0.2。
- batch_size:批量大小,設定為200。
- max_iter:最大迭代次數,設定為200。
- early_stopping: True,當驗證分數沒有提高時,提前停止來終止訓練。

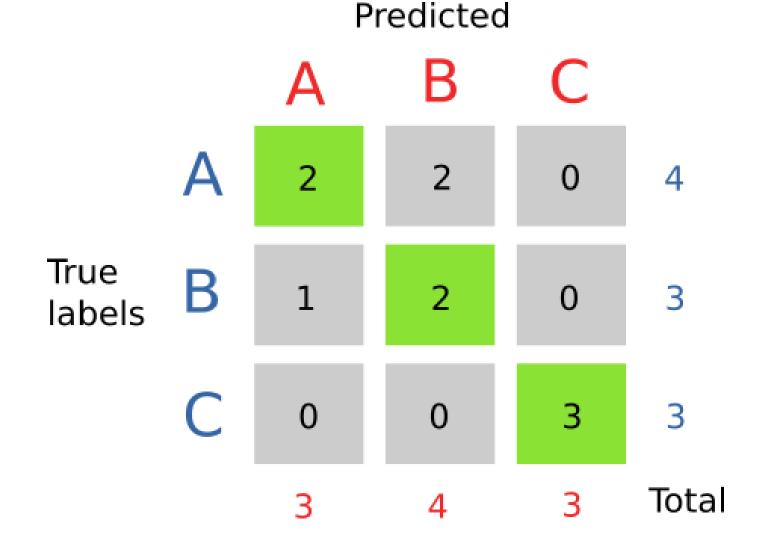
MLPClassifier

Loss Curve 誤差曲線



評估指標

- 正確率 Accuracy: 有多少比例的樣本預測對了
- 精確率 Precision: 預測為正的樣本中有多少預測對了
- **召回率 Recall**: 真實正的樣本有多少被預測對了
- **F1_Score** : 綜合考量 Precision與Recall



模型成果: Testing測試資料

• Accuracy : 82.61%

- 答對1154個
- 答錯243個

資料不平衡,NG與OK數量不一致

		預測值			
		Predict_NG	Predict_OK	Sum	
實	Label_NG	295	189	484	
際	Label_OK	54	859	913	
值	Sum	349	1048	1397	

檢視NG資料

➤ Precision: 84.53%

➤ Recall: 60.95%

➤ F1-Score : 70.83%

檢視OK資料

➤ Precision: 81.97%

➤ Recall: 94.09%

➤ F1-Score : 87.61%

模型成果: Training訓練資料

• Accuracy : 82.51%

• 答對4608個

• 答錯977個

資料不平衡,NG與OK數量不一致

		預測值			
		Predict_NG	Predict_OK	Sum	
實	Label_NG	1146	787	1933	
際	Label_OK	190	3462	3652	
值	Sum	1336	4249	5585	

檢視NG資料

➤ Precision: 85.78%

➤ Recall: 59.29%

>F1-Score : 70.11%

檢視OK資料

➤ Precision: 81.48%

➤ Recall : 94.8%

➤ F1-Score : 87.63%

模型成果: All所有資料

• CV交叉驗證:81.95% (CV=483)

• Accuracy : 82.53%

• 答對5762個

• 答錯1220個

資料不平衡,NG與OK數量不一致

		預測值			
		Predict_NG	Predict_OK	Sum	
實	Label_NG	1441	976	2417	
際	Label_OK	244	4321	4565	
值	Sum	1685	5297	6982	

檢視NG資料

➤ Precision: 85.52%

➤ Recall : 59.62%

➤ F1-Score : 70.26%

檢視OK資料

➤ Precision: 81.57%

➤ Recall: 94.65%

➤ F1-Score : 87.63%

Resample

- 利用sklearn.resample 隨機抽取OK正常的資料 使資料平衡
- 其它超參數保持不變
- 觀察資料平衡後模型的正確率

sklearn.utils.resample

 $sklearn.utils.resample(*arrays, replace=True, n_samples=None, random_state=None, stratify=None)$

source

Resample arrays or sparse matrices in a consistent way.

The default strategy implements one step of the bootstrapping procedure.

Parameters

*arrays : sequence of array-like of shape (n_samples,) or (n_samples, n_outputs)

Indexable data-structures can be arrays, lists, dataframes or scipy sparse matrices with consistent first dimension.

replace : bool, default=True

Implements resampling with replacement. If False, this will implement (sliced) random permutations.

n_samples : int, default=None

Number of samples to generate. If left to None this is automatically set to the first dimension of the arrays. If replace is False it should not be larger than the length of arrays.

random_state : int, RandomState instance or None, default=None

Determines random number generation for shuffling the data. Pass an int for reproducible results across multiple function calls. See Glossary.

stratify: array-like of shape (n_samples,) or (n_samples, n_outputs), default=None

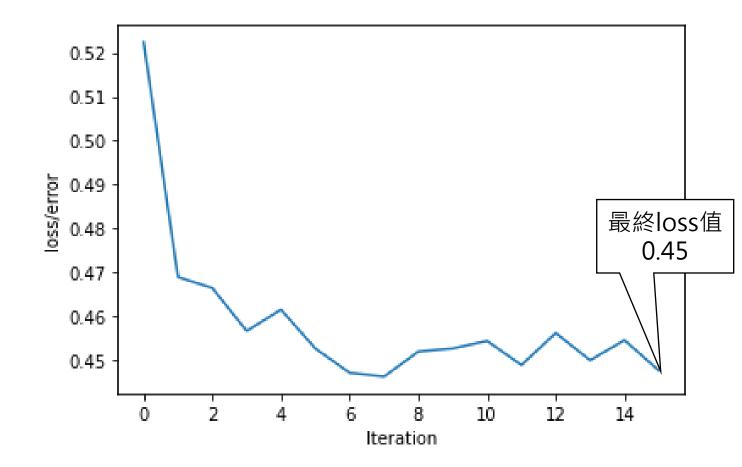
If not None, data is split in a stratified fashion, using this as the class labels.

Undersampling Copies of the minority class Original dataset Original dataset

• 資料統計

label	Train 訓練集 資料	Test 測試集 資料	Sum
OK	1933	484	2417
NG	1934	483	2417
Sum	3867	967	4834

Loss Curve 誤差曲線



模型成果: Testing測試資料

• Accuracy : 80.46%

- 答對778個
- 答錯189個

資料平衡,NG與OK數量一致

	預測值				
		Predict_NG	Predict_	OK	Sum
實	Label_NG	347		136	483
際	Label_OK	53		431	484
值	Sum	400		567	967

檢視NG資料

➤ Precision: 86.75%

➤ Recall: 71.84%

➤ F1-Score : 78.6%

檢視OK資料

➤ Precision : 76.01%

➤ Recall: 89.05%

➤F1-Score : 82.02%

模型成果:

Training訓練資料

• Accuracy : 80.37%

• 答對3108個

• 答錯759個

資料平衡,NG與OK數量一致

	預測值				
		Predict_NG Predict_OK Sun			
實	Label_NG	1420	514	1934	
際	Label_OK	245	1688	1933	
值	Sum	1665	2202	3867	

檢視NG資料

➤ Precision: 85.29%

➤ Recall: 73.42%

>F1-Score: 78.91%

檢視OK資料

➤ Precision: 76.66%

➤ Recall: 87.33%

➤ F1-Score : 81.64%

模型成果: All所有資料

• CV交叉驗證: 79.27% (CV=483)

• Accuracy : 80.39%

• 答對3886個

• 答錯948個

資料平衡,NG與OK數量一致

		預測值				
		Predict_NG Predict_OK Sum				
實	Label_NG	1767	650	2417		
際	Label_OK	298	2119	2417		
值	Sum	2065	2769	4834		

檢視NG資料

➤ Precision: 85.57%

➤ Recall: 73.11%

➤ F1-Score : 78.85%

檢視OK資料

➤ Precision: 76.53%

➤ Recall: 87.67%

➤ F1-Score : 81.72%

模型比較

資料不平衡 VS. 資料平衡

	資料不平衡	資料平衡		
All Data 所有資料				
Accuracy Score	82.53%	> 80.39%		
Cross Validation	81.95%	> 79.27%		
(NG) F1-Score	70.26%	< 78.85%		
(OK) F1-Score	87.63%	> 81.72%		
Testing 測試資料				
Accuracy Score	82.61 %	> 80.46%		
(NG) F1-Score	70.83%	< 78.6%		
(OK) F1-Score	87.61%	> 82.02%		
Training 訓練資料				
Accuracy Score	82.51 %	> 80.37%		
(NG) F1-Score	70.11%	< 78.91%		
(OK) F1-Score	87.63%	> 81.64%		



比較SVM模型

直接使用Pretrained Model

sklearn.svm.SVC

klearn.svm.SVC(*, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, re=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', break_ties=False, 'ate=None) 1 [source]

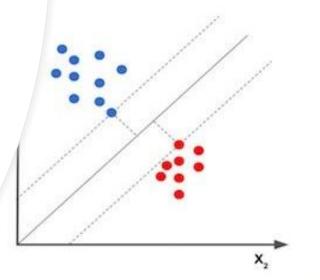
'ector Classification.

tation is based on libsym. The fit time scales at least quadratically with the number of samples and may be ond tens of thousands of samples. For large datasets consider using LinearSVC or SGDClassifier instead, possibly transformer or other Kernel Approximation.

pport is handled according to a one-vs-one scheme.

precise mathematical formulation of the provided kernel functions and how gamma, coef@ and degree affect corresponding section in the narrative documentation: Kernel functions.

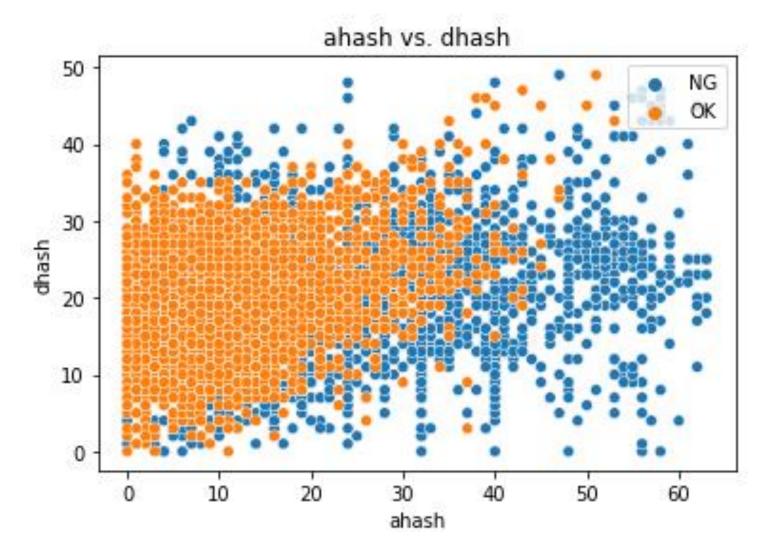
Iser Guide.



SUPPORT **VECTOR MACHINE**

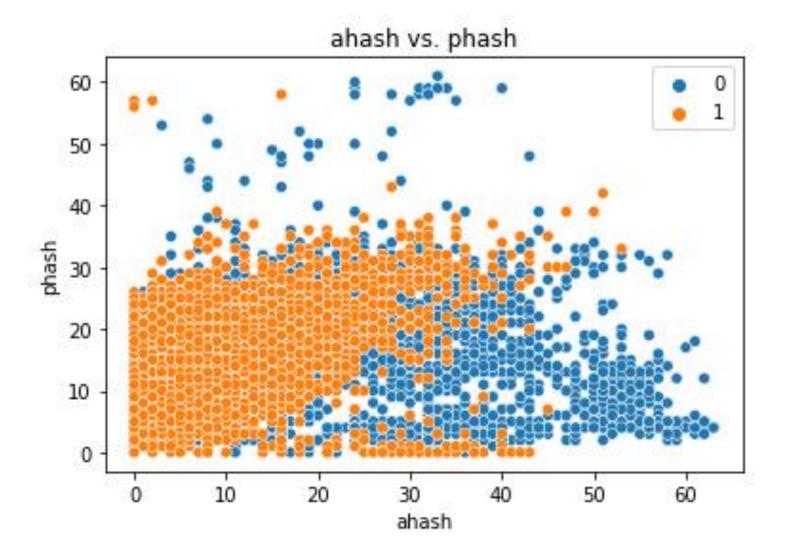
Scatter Plot 散佈圖

ahash VS. dhash



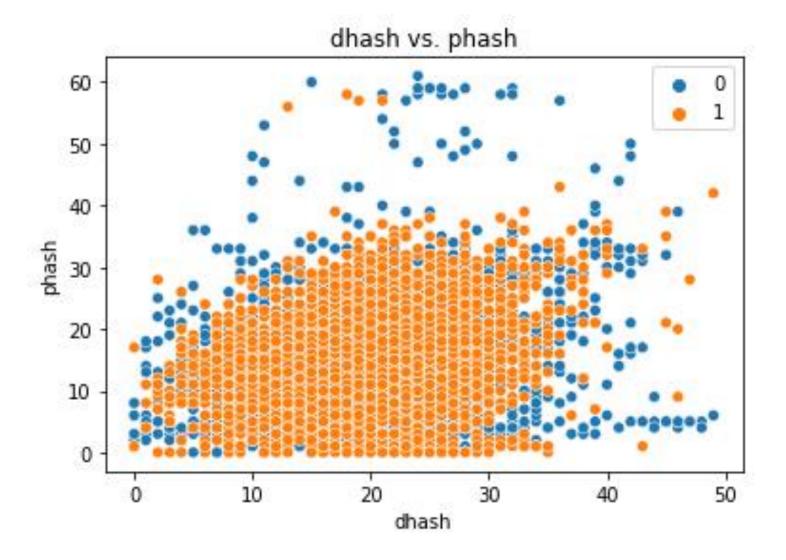
Scatter Plot 散佈圖

ahash VS. phash

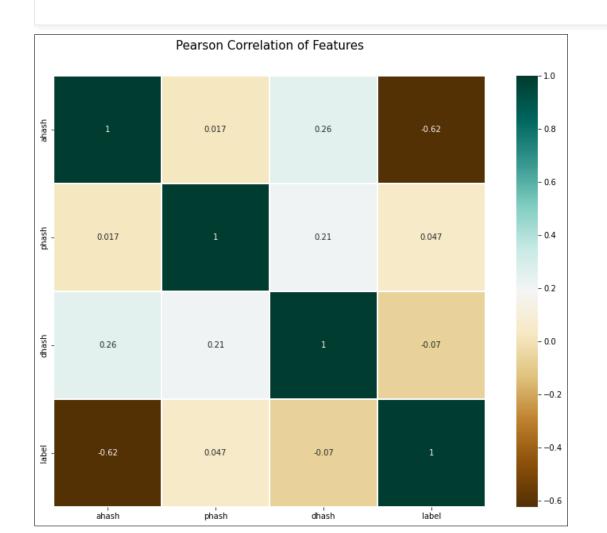


Scatter Plot 散佈圖

dhash VS. phash



決定兩個重要特徵



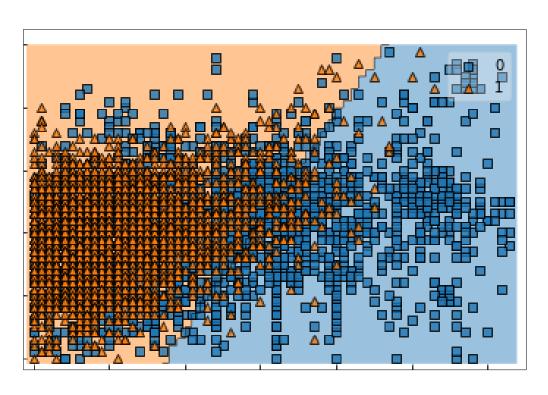
	label	
ahash	-0.62373	
phash	0.046891	
dhash	-0.06998	
label	1	

選擇特徵: ahash & dhash

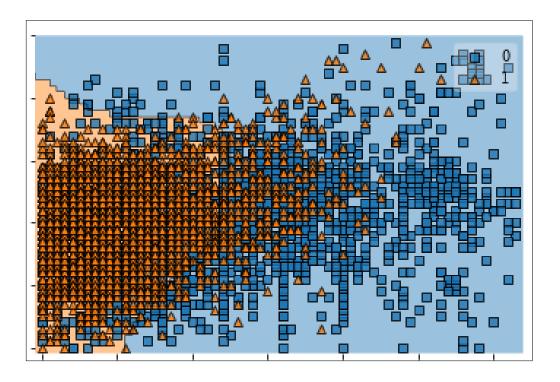
SVM模型

直接使用Pretrained Model

• 線性 clf = LinearSVC()



• 非線性 *clf = SVC(kernel="rbf")*



模型成果: 線性SVM

• Accuracy : 79.03%

• 答對5518個

• 答錯1464個

	預測值				
	Predict_NG Predict_OK Sur				
實	Label_NG	1084	1333	2417	
際	Label_OK	131	4434	4565	
值	Sum	1215	5767	6982	

檢視NG資料

➤ Precision: 89.22%

➤ Recall: 44.85%

➤ F1-Score : 59.69%

檢視OK資料

➤ Precision: 76.89%

➤ Recall: 97.13%

➤ F1-Score : 85.83%

模型成果: 非線性SVM

• Accuracy : 82.78%

• 答對5780個

• 答錯1202個

	預測值				
	Predict_NG Predict_OK Sui				
實	Label_NG	1604	813	2417	
際	Label_OK	389	4176	4565	
值	Sum	1993	4989	6982	

檢視NG資料

➤ Precision: 80.48%

➤ Recall : 66.36%

➤ F1-Score : 72.74%

檢視OK資料

➤ Precision: 83.7%

➤ Recall: 91.48%

➤ F1-Score : 87.42%

模型比較

線性SVM VS.非線性SVM

	SVM線性	SVM非線性 ®
Accuracy Score	79.03%	< 82.78%
答對	5518個	< 5780個
答錯	1464個	> 1202個
	NG	
Precision	89.22%	80.48%
Recall	44.85%	66.36%
F1-Score	59.69%	72.74%
	OK	
Precision	76.89%	83.70%
Recall	97.13%	91.48%
F1-Score	85.83%	87.42%

模型比較

非線性SVM VS. MLP

	非線性SVM	MLP
Accuracy Score	82.78%	82.53%
答對	5780個	5762個
答錯	1202個	1220個

研究重點摘要

01

利用hash比較圖 片相似性,並轉換 為數值。 02

藉由MLP分類是否 為瑕疵 03

藉由SVM分類是 否為瑕疵

結論

需要再討論

- Label的爭議
 - → (右圖)我們認為是OK正常,但廠商卻說是瑕疵
 - → 因此標準設定為正確值(Accuracy Score)達九成即可
- Accuracy / Precision / Recall / F1_Score 看重哪一個
 - → Precision:預測為瑕疵,有多少預測對了?
 - → Recall:實際為瑕疵,有多少被預測對了?

