

The background features abstract, organic shapes in shades of pink, red, and yellow, separated by thick black wavy lines. There are also some white decorative elements like swirls and dots.

英語手語字母辨識實驗 以 MNIST 公開資料為例

109034537 邱靖中

大綱

- 研究目的
- 資料介紹
- 研究流程與結果
- 結論





01

研究目的



Why

為了瞭解深度學習運用於影像辨識之原理並理解如何運用於手語字母辨識領域

Where

住院病患以字母作為需求選擇，使機器能夠以字母判別病患想要什麼

Who

對於需要使用手語表達與溝通的族群

When

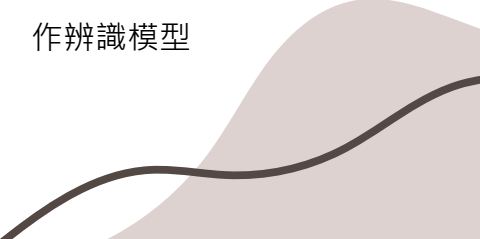
可以運用於口語表達障礙需要以字母作為回答之情境

What

實作一個手語字母辨識模型

How

運用 Python 與CNN 框架實作辨識模型





02

資料介紹

MNIST 公開資料集

- 以 0~24 代表英文字母 A~Y
- 以 784 (28x28) 像素值呈現
- 訓練資料共 27,455 筆、測試資料共 7172 筆



label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	pixel10	pixel11	pixel12	pixel13	pixel14	pixel15	pixel16	pixel17	pixel18	pixel19	pixel20	pixel21	pixel22	pixel23	pixel24
3	107	118	127	134	139	143	146	150	153	156	158	160	163	165	159	166	168	170	170	171	171	171		
6	155	157	156	156	156	157	156	158	158	157	158	156	154	154	153	152	151	149	149	148	147	146		
2	187	188	188	187	187	186	187	188	187	186	185	185	185	184	184	184	181	181	179	179	179	178		
2	211	211	212	212	211	210	211	210	210	211	209	207	208	207	206	203	202	201	200	198	197	195		
13	164	167	170	172	176	179	180	184	185	186	188	189	189	190	191	189	190	190	187	190	192	193		
16	161	168	172	173	178	184	189	193	196	202	206	208	212	214	217	219	220	222	226	227	227	228		
8	134	134	135	135	136	137	137	138	138	138	139	138	138	139	138	139	139	139	139	138	138	138		
22	114	42	74	99	104	109	117	127	142	152	155	162	165	168	177	140	170	181	182	185	188	189		
3	169	174	176	180	183	185	187	188	190	191	191	191	191	190	190	189	189	189	189	186	186	185		
3	189	189	189	190	190	191	190	190	190	189	188	187	187	186	185	184	184	183	180	180	180	178		
18	133	135	141	146	150	155	158	159	163	165	166	167	169	171	172	172	172	172	172	171	171	172		
10	0	25	38	40	41	46	50	56	69	81	91	96	99	104	110	115	118	122	123	126	130	131		
16	87	91	99	116	132	142	147	153	160	164	167	170	174	176	179	180	180	179	178	180	184	185		
22	80	98	121	39	53	94	100	107	110	128	144	152	162	165	187	158	133	184	186	190	197	196		
20	127	127	128	130	132	133	133	133	135	136	136	135	135	134	133	129	134	130	130	128	127	127		
16	86	87	89	93	104	114	122	131	137	140	142	146	151	155	159	162	164	166	168	172	173	174		
17	118	120	128	135	139	145	149	150	153	156	159	161	164	165	167	168	168	169	170	169	170	170		
13	223	225	226	227	228	229	230	230	230	231	232	231	231	232	232	231	232	233	236	236	231	232		
13	189	193	195	197	201	206	208	213	215	218	221	221	223	223	225	226	225	225	225	225	227	227		
19	173	174	176	177	176	177	177	177	177	177	175	175	175	175	175	174	173	172	172	170	170	170		
18	149	150	150	150	150	151	151	151	151	150	149	148	147	144	144	144	143	141	142	139	137			
21	131	135	139	143	145	146	149	152	153	156	157	158	159	162	162	163	165	166	166	168	167	168		
16	90	98	105	110	118	124	126	127	131	136	139	142	145	146	149	151	153	155	155	156	158	159		
23	85	90	97	103	119	130	138	147	152	157	161	164	163	167	171	176	178	180	182	184	185	188		
3	131	138	144	146	148	150	152	153	154	156	156	156	156	156	157	156	158	156	155	154	153	153		



03

研究流程
與結果

研究架構



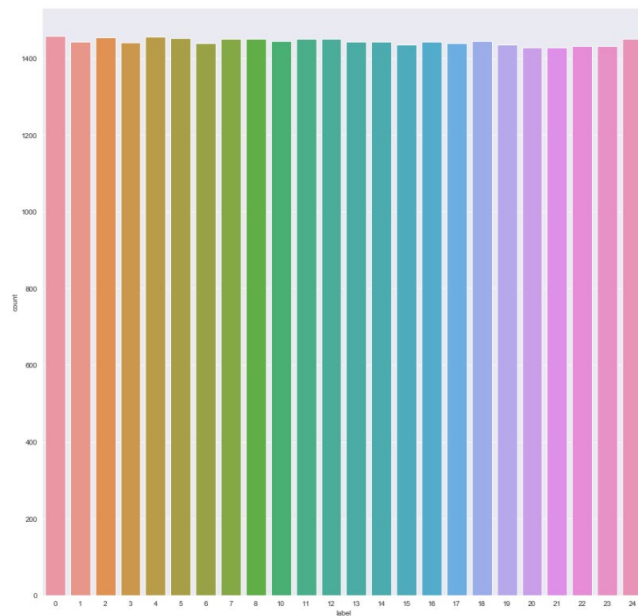
資料前處理 - 資料視覺化

- 將原始資料合併
- 以 seaborn 套件呈現
- 資料沒有不平衡問題

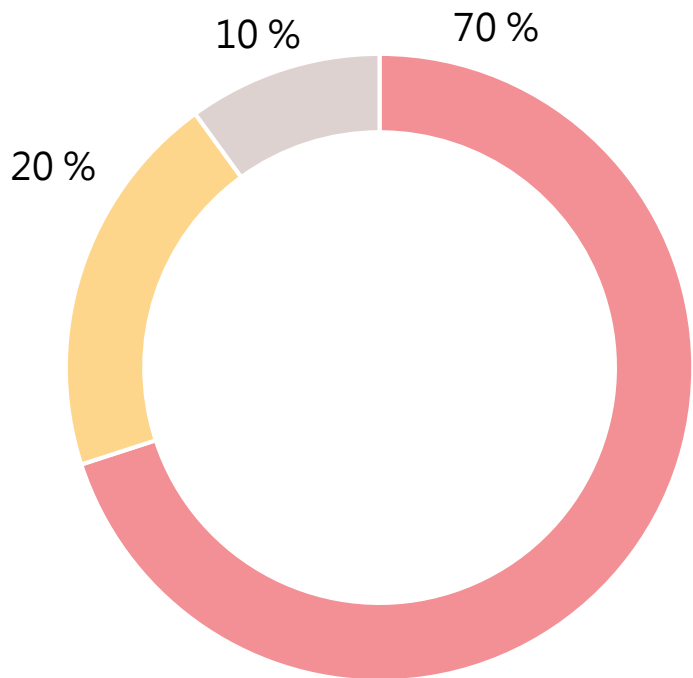
```
train_df = pd.read_csv("D:/IIE/Project3/sign_mnist_train.csv")
test_df = pd.read_csv("D:/IIE/Project3/sign_mnist_test.csv")

raw_data = pd.concat([train_df, test_df], axis=0)
```

```
plt.figure(figsize = (15,15)) # Label Count
sns.set_style("darkgrid")
sns.countplot(raw_data['label'])
```



資料前處理 - 資料分割



- 訓練資料集：24239 筆
- 驗證資料集：6925 筆
- 測試資料集：3463 筆

```
def datasplit(dataframe,ratio1,ratio2):  
    return dataframe[0:(round(len(dataframe)*(ratio1)))] ,dataframe[(round(len(dataframe)*(ratio1))):(round(len(dataframe)*(ratio1+ratio2)))]  
train_all, validation_all, test_all =datasplit(raw_data,0.7,0.2)  
  
#特徵值  
train_x = train_all.drop(columns=['label']).values  
validation_x = validation_all.drop(columns=['label']).values  
test_x = test_all.drop(columns=['label']).values  
  
#目標值  
train_y = train_all['label'].values  
validation_y = validation_all['label'].values  
test_y = test_all['label'].values  
print(len(train_x), len(validation_x), len(test_x))  
print(len(train_y), len(validation_y), len(test_y))
```

資料前處理 - One-hot Encoding

運用 One-hot Encoding 方式將目標值轉換為二維的形式，以利電腦處理作業的效率

```
from sklearn.preprocessing import LabelBinarizer
label_binarizer = LabelBinarizer()
train_y = label_binarizer.fit_transform(train_y)
validation_y = label_binarizer.fit_transform(validation_y)
test_y = label_binarizer.fit_transform(test_y)
```

train_y

```
array([[0, 0, 0, ..., 1, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 1],
       [0, 1, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

資料前處理 - 資料標準化

由於色碼各個像素值範圍介於 0~255，因此將所有特徵值除上 255，標準化以縮減數據間的跨度

```
train_x
array([[107, 118, 127, ..., 204, 203, 202],
       [155, 157, 156, ..., 103, 135, 149],
       [187, 188, 188, ..., 195, 194, 195],
       ...,
       [155, 159, 163, ..., 222, 221, 221],
       [145, 147, 148, ..., 181, 180, 180],
       [ 50,  14,  52, ..., 169, 169, 169]], dtype=int64)
```



```
# Normalize the data
train_x = train_x / 255
validation_x = validation_x / 255
test_x = test_x / 255
```

```
train_x
array([[0.41960784, 0.4627451, 0.49803922, ..., 0.8, 0.79607843,
        0.79215686],
       [0.60784314, 0.61568627, 0.61176471, ..., 0.40392157, 0.52941176,
        0.58431373],
       [0.73333333, 0.7372549, 0.7372549, ..., 0.76470588, 0.76078431,
        0.76470588],
       ...,
       [0.60784314, 0.62352941, 0.63921569, ..., 0.87058824, 0.86666667,
        0.86666667],
       [0.56862745, 0.57647059, 0.58039216, ..., 0.70980392, 0.70588235,
        0.70588235],
       [0.19607843, 0.05490196, 0.20392157, ..., 0.6627451, 0.6627451,
        0.6627451 ]])
```

資料前處理 - 資料增強

- 為了避免模型過擬合
- 隨意調整角度、壓縮圖片

```
datagen = ImageDataGenerator(  
  
    rotation_range=10,  
    zoom_range = 0.1,  
    width_shift_range=0.1,  
    height_shift_range=0.1,  
    horizontal_flip=True,  
    vertical_flip=False)  
  
datagen.fit(train_x_cnn)
```

模型建立 - CNN 架構建立

- 一層卷積層，神經元數為 25
- 一層池化層
- 一層卷積層神經元數為 75
- 一層 Dropout 層，比率為 0.5
- 一層池化層
- 一層卷積層神經元數為 100
- 一層 Dropout 層，比率為 0.5
- 一層扁平層
- 一層全連接層，神經元數為 128
- 一層 Dropout 層，比率為 0.5
- 一層輸出層，神經元數為 24

- 激活函數為 ReLU
- 損失函數為 Categorical
- 學習率為 0.01

```
learning_rate_reduction = ReduceLROnPlateau(monitor='val_accuracy', patience = 2, min_lr=0.01)
```

```
model = Sequential()  
model.add(Conv2D(25, (3,3), strides = 1, padding = 'same', activation = 'relu', input_shape = (28,28,1)))  
model.add(BatchNormalization())  
model.add(MaxPool2D((2,2), strides = 2, padding = 'same'))  
  
model.add(Conv2D(75, (3,3), strides = 1, padding = 'same', activation = 'relu'))  
model.add(Dropout(0.5))  
model.add(BatchNormalization())  
model.add(MaxPool2D((2,2), strides = 2, padding = 'same'))  
model.add(Conv2D(100, (3,3), strides = 1, padding = 'same', activation = 'relu'))  
model.add(Dropout(0.5))  
model.add(BatchNormalization())  
model.add(MaxPool2D((2,2), strides = 2, padding = 'same'))  
  
model.add(Flatten())  
model.add(Dense(units = 128, activation = 'relu'))  
model.add(Dropout(0.5))  
model.add(Dense(units = 24, activation = 'softmax'))  
model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])  
model.summary()
```

模型建立 - 超參數調整

- 調整 Activation Function

	Training Set	Validation Set	Testing Set
softmax	58.99%	78.12%	74.62%
ReLU	90.89%	96.82%	96.15%
Sigmoid	86.54%	67.15%	67.13%

模型建立 - 超參數調整

- 調整 Batch Size

	Training Set	Validation Set	Testing Set
10	90.89%	96.82%	96.15%
50	96.52%	98.71%	98.46%
100	97.53%	98.17%	97.31%

模型建立 - 超參數調整

- 調整 Learning Rate

	Training Set	Validation Set	Testing Set
0.01	96.52%	98.71%	98.46%
0.001	93.59%	97.39%	96.62%
0.0001	94.74%	99.49%	99.24%

模型建立 - 超參數調整

- 調整 Loss Function

	Training Set	Validation Set	Testing Set
categorical	94.74%	99.49%	99.24%
binary	99.49%	99.98%	99.98%



04

結論

研究結果



公開資料準確度高

運用 **CNN** 方式建立MNIST
手語字母辨識模型可以獲得
高達 **99.98%** 之準確度



外來照片準確度問題

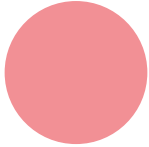
由於本實驗僅根據公開資料進行訓練與評估模型，如果以外來照片進行測試會產生預測不準的問題

未來展望



蒐集不同資料集

蒐集來自不同來源的手語靜態圖建立模型，以避免環境過於單純而產生模型過擬合的問題



結合手掌捕捉技術

結合手掌動態捕捉的技術，以**3D** 抓取手部關鍵點之方式建立手語辨識預測模型

The background features abstract, organic shapes in shades of pink, red, and yellow. A grey, wavy shape with white dashed lines is positioned at the top. A yellow shape with white dots is on the right. A red shape with white dots is at the bottom left. A black line curves across the middle. The text "Thanks for listening" is centered in a bold, black, sans-serif font.

Thanks for listening