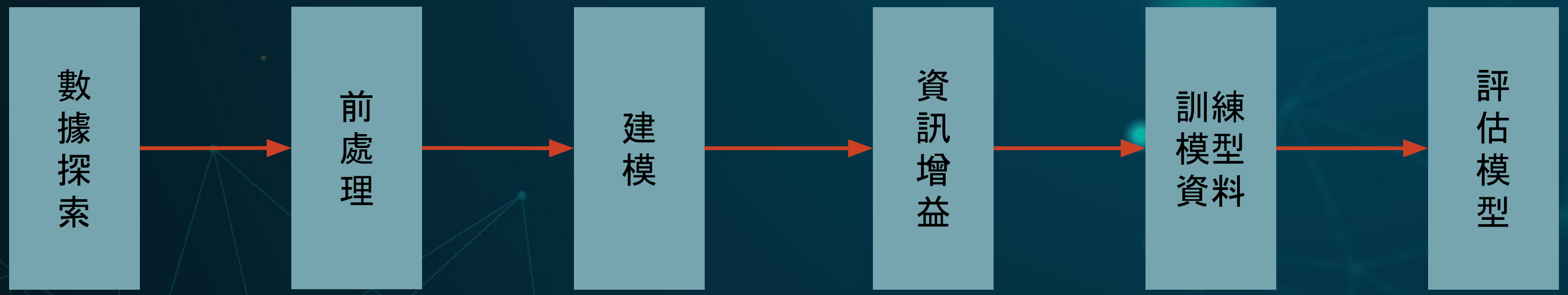
玉山手寫辨識

406170085 資數四 黃偉柏
407170288 資數三 李秉叡
407170460 資數三 關佳怡

目錄

流程圖



數據探索

前處理

建模

資訊增益

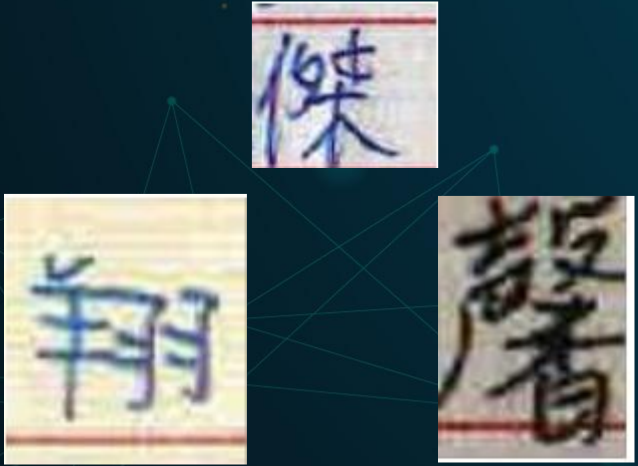
訓練模型資料

評估模型

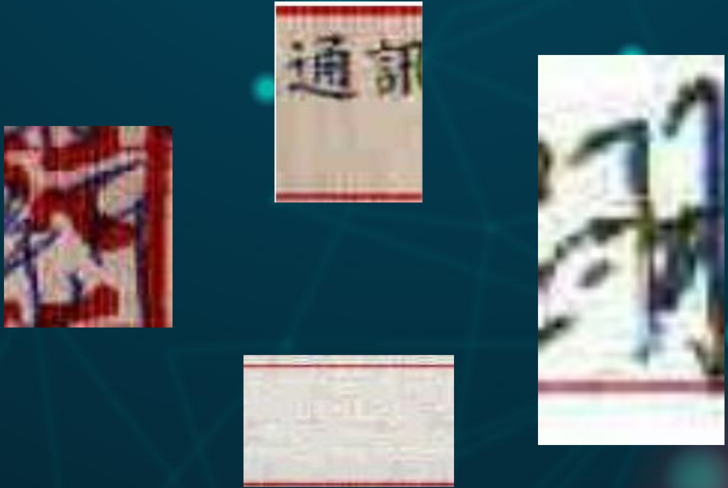
數據探索

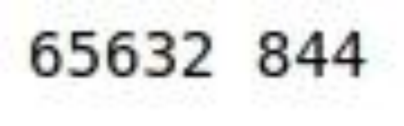
資料清理
(圖片說明)

正常



不正常

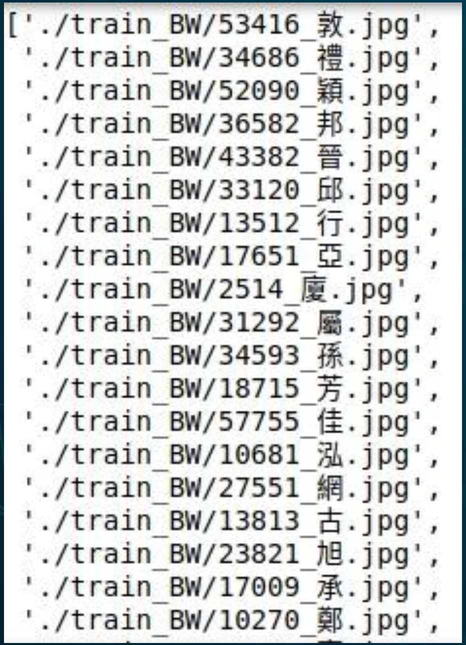




```
65632 844
```

圖片總數

文字(不重複)

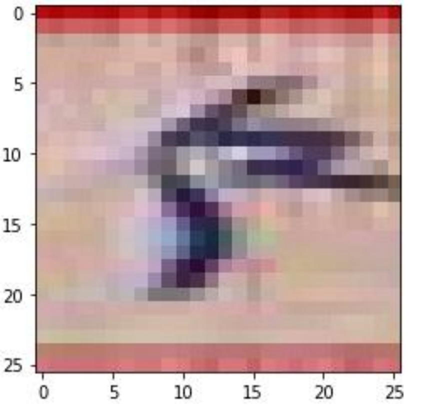


```
['./train_BW/53416_敦.jpg',
 './train_BW/34686_禮.jpg',
 './train_BW/52090_穎.jpg',
 './train_BW/36582_邦.jpg',
 './train_BW/43382_晉.jpg',
 './train_BW/33120_邱.jpg',
 './train_BW/13512_行.jpg',
 './train_BW/17651_亞.jpg',
 './train_BW/2514_慶.jpg',
 './train_BW/31292_屬.jpg',
 './train_BW/34593_孫.jpg',
 './train_BW/18715_芳.jpg',
 './train_BW/57755_佳.jpg',
 './train_BW/10681_泓.jpg',
 './train_BW/27551_網.jpg',
 './train_BW/13813_古.jpg',
 './train_BW/23821_旭.jpg',
 './train_BW/17009_承.jpg',
 './train_BW/10270_鄭.jpg',
```

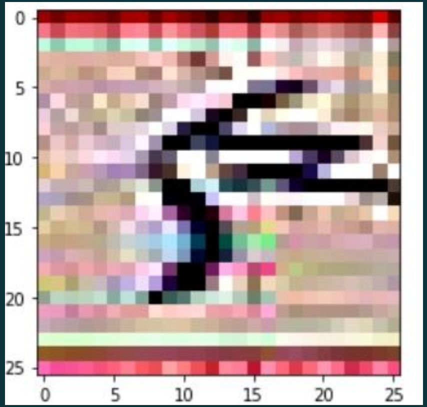
數據前處理

圖片處理

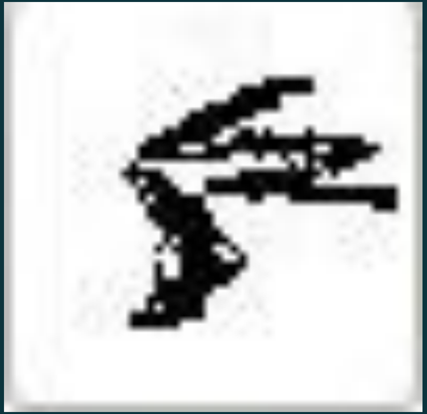
原版



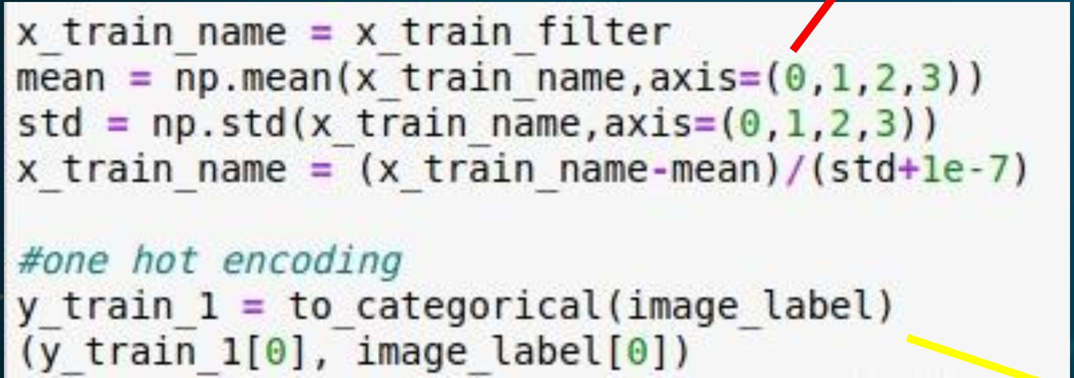
原版銳利化



先轉黑白在銳利化



原版

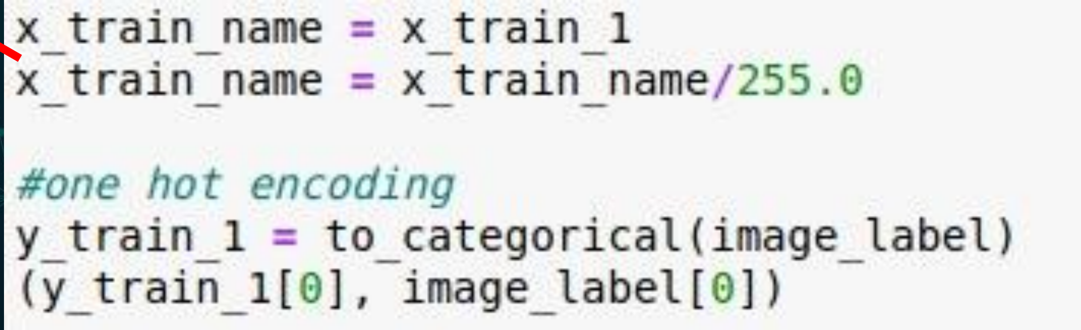


```
x_train_name = x_train_filter
mean = np.mean(x_train_name,axis=(0,1,2,3))
std = np.std(x_train_name,axis=(0,1,2,3))
x_train_name = (x_train_name-mean)/(std+1e-7)

#one hot encoding
y_train_1 = to_categorical(image_label)
(y_train_1[0], image_label[0])
```

Z-Score

讀熱編碼

特徵壓縮



```
x_train_name = x_train_1
x_train_name = x_train_name/255.0

#one hot encoding
y_train_1 = to_categorical(image_label)
(y_train_1[0], image_label[0])
```

先轉黑白
再銳利化

建立CNN模型

卷積層

卷積層

池化層

卷積層

卷積層

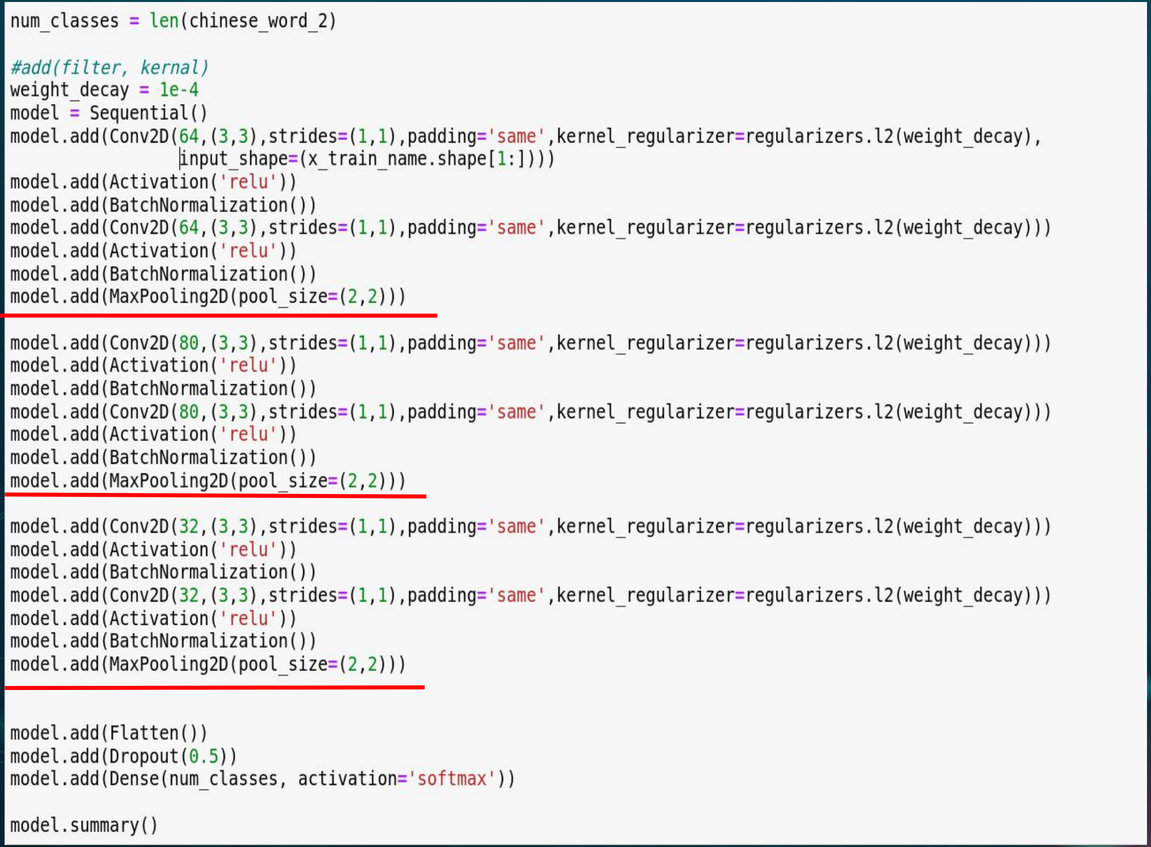
池化層

卷積層

卷積層

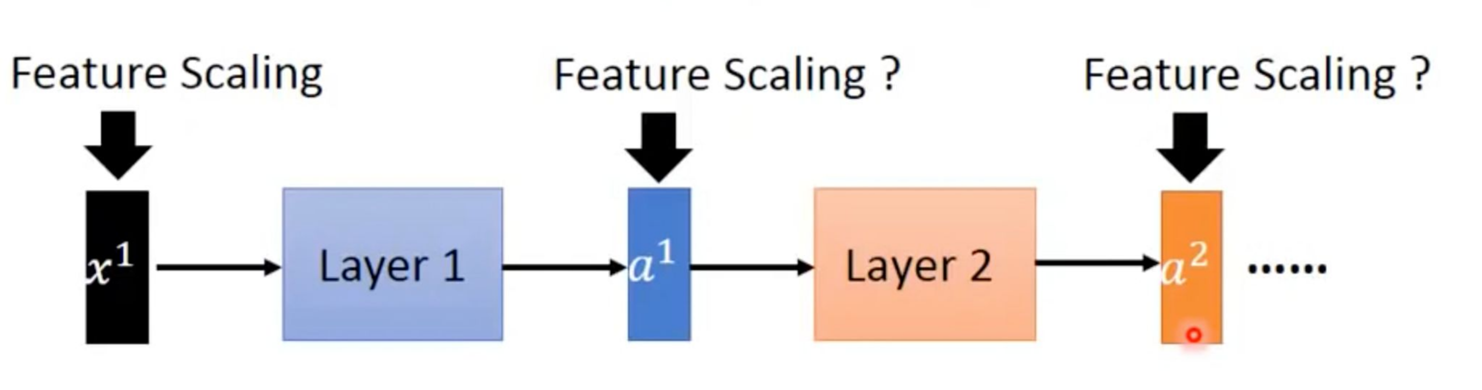
池化層

dropout



```
num_classes = len(chinese_word_2)

#add(filter, kernal)
weight_decay = 1e-4
model = Sequential()
model.add(Conv2D(64,(3,3),strides=(1,1),padding='same',kernel_regularizer=regularizers.l2(weight_decay),
                 input_shape=(x_train_name.shape[1:])))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(64,(3,3),strides=(1,1),padding='same',kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(80,(3,3),strides=(1,1),padding='same',kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(80,(3,3),strides=(1,1),padding='same',kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(32,(3,3),strides=(1,1),padding='same',kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(32,(3,3),strides=(1,1),padding='same',kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))


model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

model.summary()
```

Batch Normalization



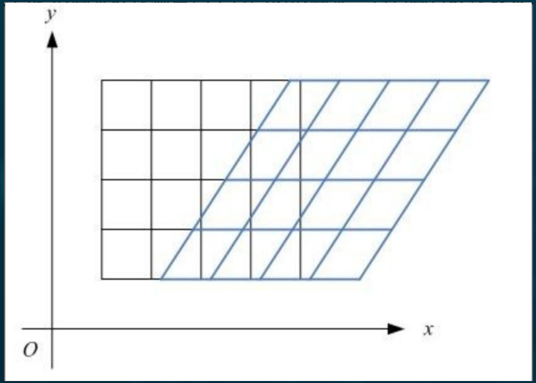
資訊增益

- width_shift_range & height_shift_range:

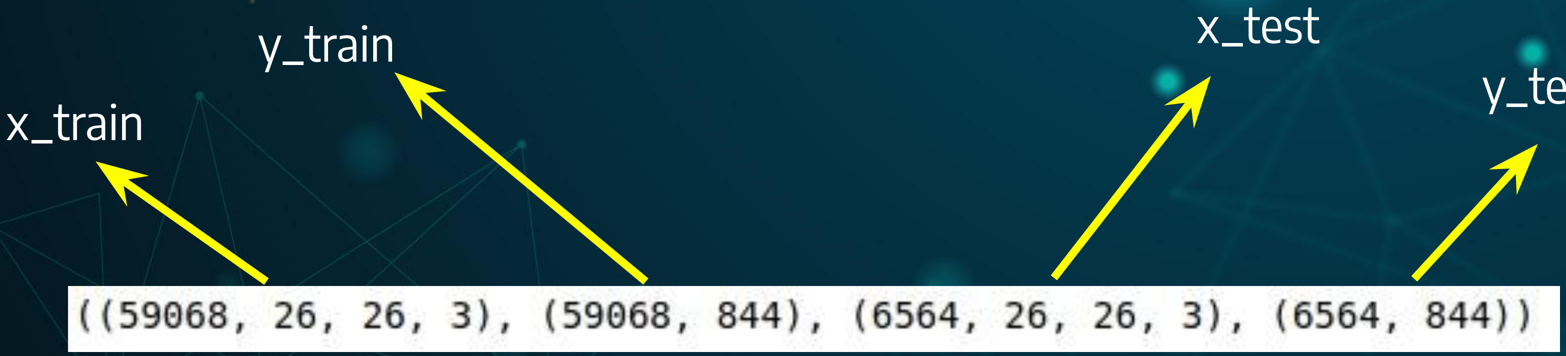
  水平、上下平移

- Shear_range:

  讓所有點的x坐標(或者y坐標)保持不變，而對應的y坐標(或者x坐標)則按比例發生平移



數據拆分



x_train

y_train

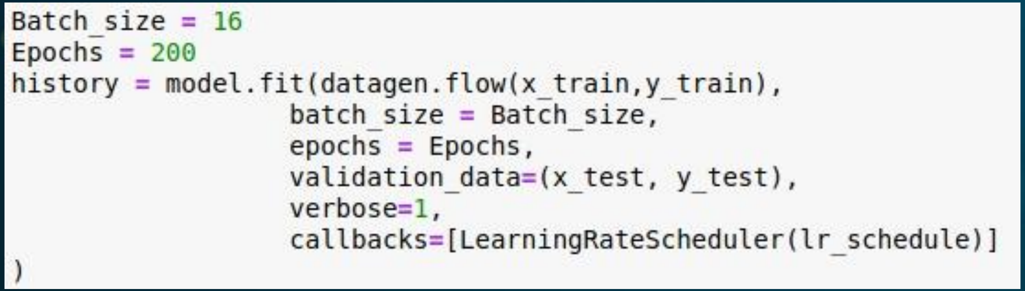
x_test

y_test

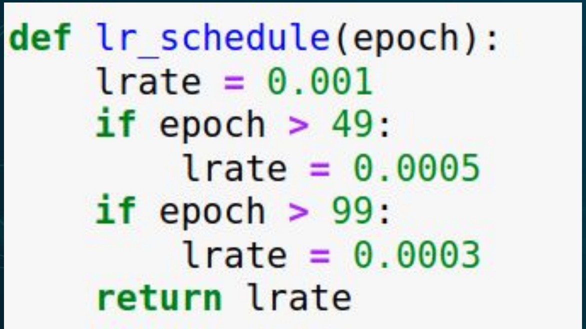
```
((59068, 26, 26, 3), (59068, 844), (6564, 26, 26, 3), (6564, 844))
```

訓練模型

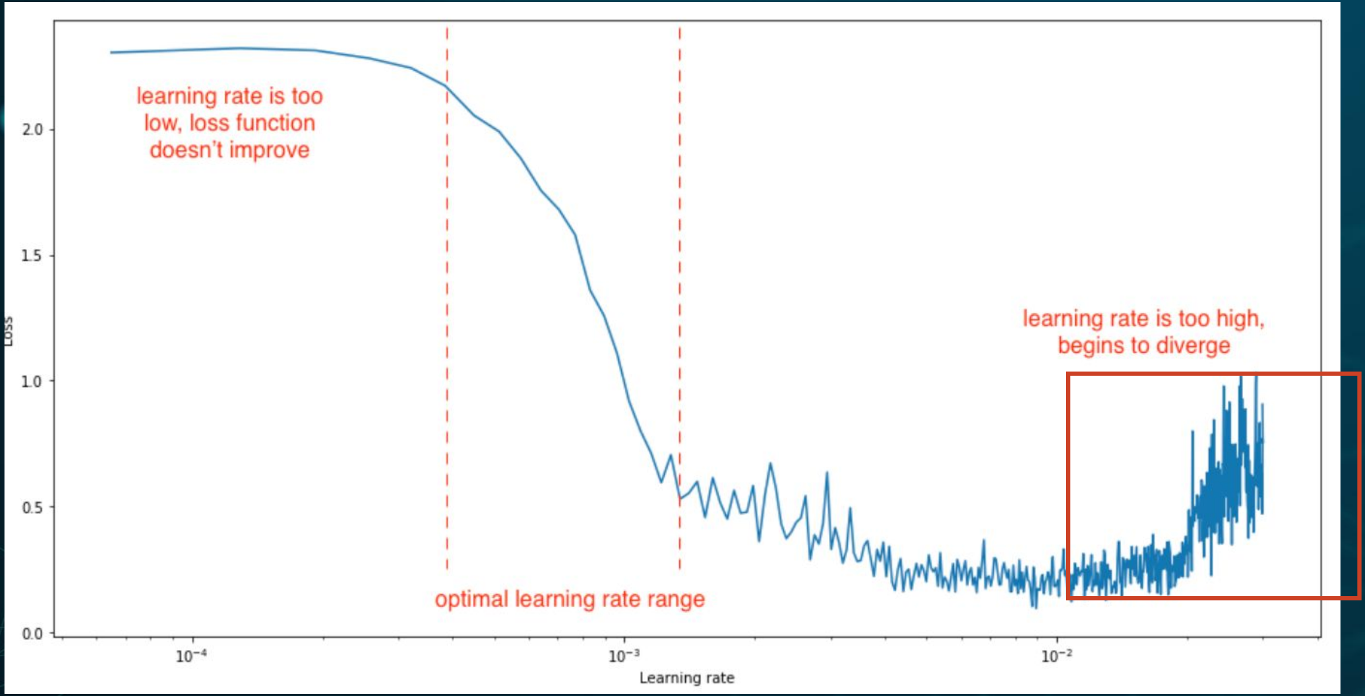
Batch_size = 16
Epochs = 200



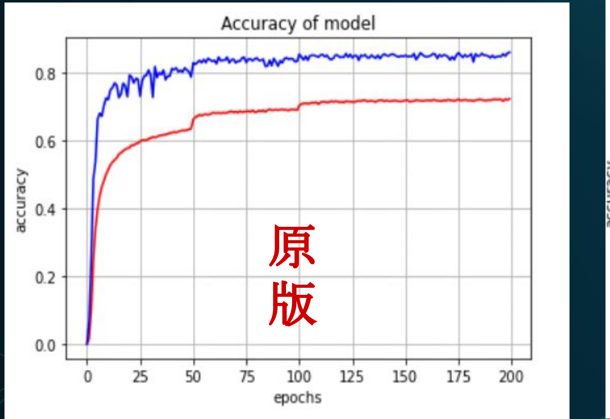
```
Batch_size = 16
Epochs = 200
history = model.fit(datagen.flow(x_train,y_train),
                  batch_size = Batch_size,
                  epochs = Epochs,
                  validation_data=(x_test, y_test),
                  verbose=1,
                  callbacks=[LearningRateScheduler(lr_schedule)]
)
```



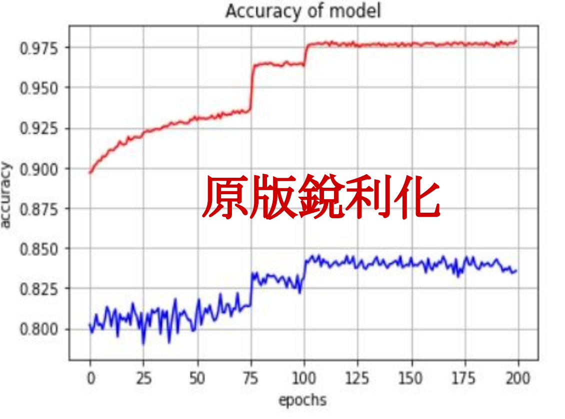
```
def lr_schedule(epoch):
    lrate = 0.001
    if epoch > 49:
        lrate = 0.0005
    if epoch > 99:
        lrate = 0.0003
    return lrate
```


learning rate is too low, loss function doesn't improve
optimal learning rate range
learning rate is too high, begins to diverge
Loss
2.0
1.5
1.0
0.5
0.0
10^{-4}
10^{-3}
10^{-2}
Learning rate

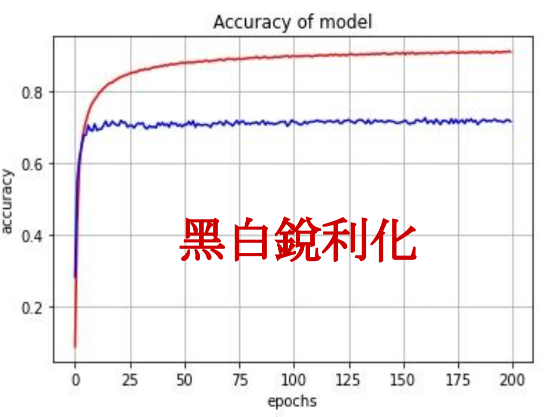
🟥 train 🟦 test



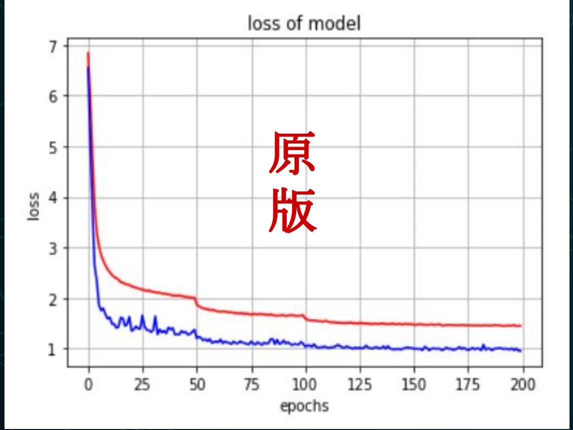
原版



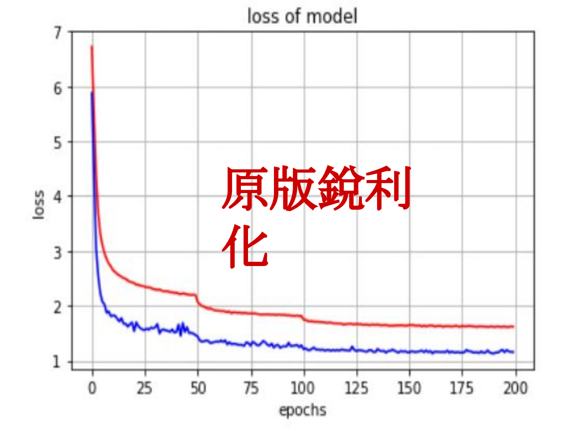
原版銳利化



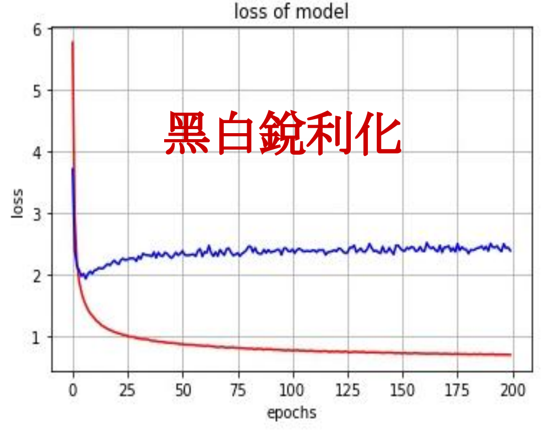
黑白銳利化



原版

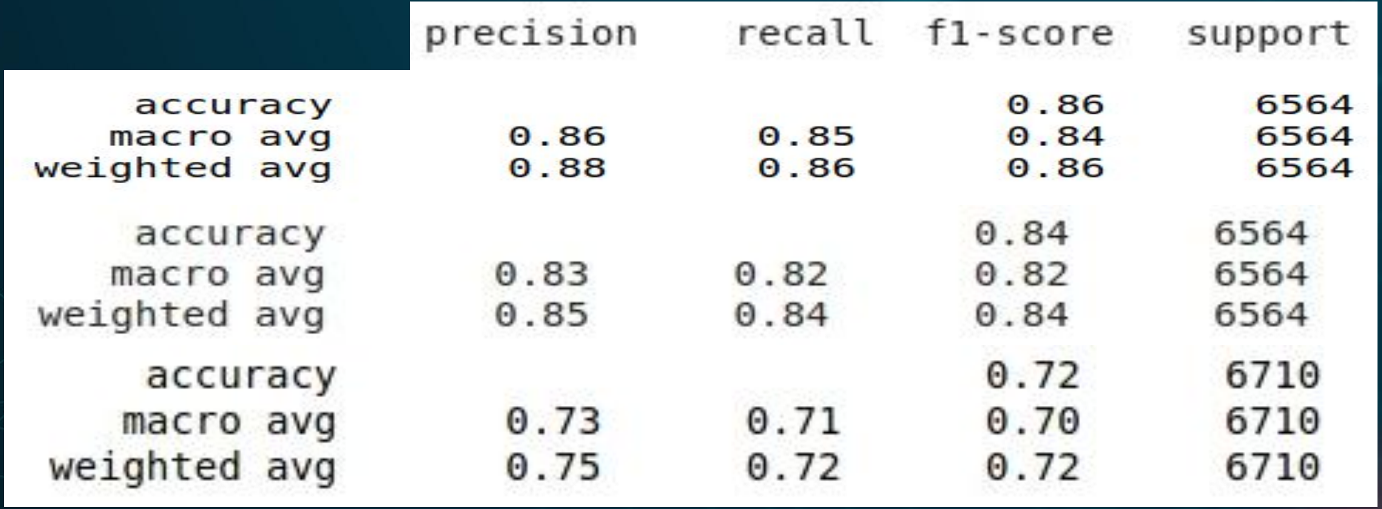


原版銳利化



黑白銳利化

測試集的評估指標



		precision	recall	f1-score	support
原版	accuracy			0.86	6564
	macro avg	0.86	0.85	0.84	6564
	weighted avg	0.88	0.86	0.86	6564
原版銳利化	accuracy			0.84	6564
	macro avg	0.83	0.82	0.82	6564
	weighted avg	0.85	0.84	0.84	6564
黑白銳利化	accuracy			0.72	6710
	macro avg	0.73	0.71	0.70	6710
	weighted avg	0.75	0.72	0.72	6710

工作內容與心得

THANKS!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

參考資料：

https://ithelp.ithome.com.tw/articles/10204032

https://www.youtube.com/watch?v=BZh1ltr5Rkg

https://zhuanlan.zhihu.com/p/30197320