



In [1]:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np
import cv2
import time

from keras.models import load_model
model = load_model('my_model.h5')
#model.summary()
model.load_weights('my_model_weights.h5')

#如果 load_model 有錯誤訊息，請檢查 keras 和 tensorflow 的版本.
#建議使用 keras 2.2.4, tensorflow 1.8.0 or 1.12.0
```

Using TensorFlow backend.

WARNING: Logging before flag parsing goes to stderr.

W0814 11:16:46.537509 5000 deprecation\_wrapper.py:119] From C:\Users\USER\Anaconda3\envs\No-GPU\lib\site-packages\keras\backend\tensorflow\_backend.py:517: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.

W0814 11:16:46.560629 5000 deprecation\_wrapper.py:119] From C:\Users\USER\Anaconda3\envs\No-GPU\lib\site-packages\keras\backend\tensorflow\_backend.py:4138: The name tf.random\_uniform is deprecated. Please use tf.random.uniform instead.

W0814 11:16:46.591586 5000 deprecation\_wrapper.py:119] From C:\Users\USER\Anaconda3\envs\No-GPU\lib\site-packages\keras\backend\tensorflow\_backend.py:3976: The name tf.nn.max\_pool is deprecated. Please use tf.nn.max\_pool2d instead.

W0814 11:16:46.608501 5000 deprecation\_wrapper.py:119] From C:\Users\USER\Anaconda3\envs\No-GPU\lib\site-packages\keras\backend\tensorflow\_backend.py:131: The name tf.get\_default\_graph is deprecated. Please use tf.compat.v1.get\_default\_graph instead.

W0814 11:16:46.609540 5000 deprecation\_wrapper.py:119] From C:\Users\USER\Anaconda3\envs\No-GPU\lib\site-packages\keras\backend\tensorflow\_backend.py:133: The name tf.placeholder\_with\_default is deprecated. Please use tf.compat.v1.placeholder\_with\_default instead.

W0814 11:16:46.615479 5000 deprecation.py:506] From C:\Users\USER\Anaconda3\envs\No-GPU\lib\site-packages\keras\backend\tensorflow\_backend.py:3445: calling dropout (from tensorflow.python.ops.nn\_ops) with keep\_prob is deprecated and will be removed in a future version.

Instructions for updating:

Please use `rate` instead of `keep\_prob`. Rate should be set to `rate = 1 - keep\_prob`.

W0814 11:16:46.674457 5000 deprecation\_wrapper.py:119] From C:\Users\USER\Anaconda3\envs\No-GPU\lib\site-packages\keras\backend\tensorflow\_backend.py:174: The name tf.get\_default\_session is deprecated. Please use tf.compat.v1.get\_default\_session instead.

W0814 11:16:46.787352 5000 deprecation\_wrapper.py:119] From C:\Users\USER\Anaconda3\envs\No-GPU\lib\site-packages\keras\optimizers.py:790: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

W0814 11:16:46.872048 5000 deprecation.py:323] From C:\Users\USER\Anaconda3\envs\No-GPU\lib\site-packages\tensorflow\python\ops\math\_grad.py:1250: add\_dispatch\_support.<locals>.wrapper (from tensorflow.python.ops.array\_ops) is deprecated and will be removed in a future version.

Instructions for updating:

Use tf.where in 2.0, which has the same broadcast rule as np.where

In [2]:

```

cap = cv2.VideoCapture(0)
width = cap.get(cv2.CAP_PROP_FRAME_WIDTH)
height = cap.get(cv2.CAP_PROP_FRAME_HEIGHT)
print("Image Size: %d x %d" % (width, height))

while(True):
    tStart = time.time() #計時開始
    ret, img_1 = cap.read()
    img_2 = img_1[0:480,80:560]
    cv2.imshow('Camera',img_2)

    img_2_gray = cv2.cvtColor(img_2, cv2.COLOR_BGR2GRAY)
    img_3 = 255 - img_2_gray
    img_3 = img_3.astype('float32')
    img_3_min = np.amin(img_3)
    img_4 = img_3 - np.amin(img_3)
    img_5 = 255 * img_4 / (np.amax(img_4))
    kernel = np.ones((5,5),np.uint8)
    img_6 = cv2.dilate(img_5,kernel,iterations = 3)
    img_7 = cv2.resize(img_6,(28,28),1)
    img_8 = img_6.astype('uint8')
    cv2.imshow('input',img_8)

    #print(img_1.dtype, img_1.shape)
    #print(img_2.dtype, img_2.shape)

    x_test_image = np.reshape(img_7, (1,28,28))
    #print(np.amax(x_test_image))
    #print(np.amin(x_test_image))

    # convert 2-D 28x28 image to 4-D nx28x28x1 array
    x_Test4D=x_test_image.reshape(x_test_image.shape[0],28,28,1).astype('float32')
    # normalize the image numbers to 0~1
    x_Test4D_normalize = (x_Test4D / np.amax(x_test_image))
    #print(x_Test4D_normalize.shape)
    prediction=model.predict_classes(x_Test4D_normalize)

    filename = '%s%d%s' % ('./numbers/number_', prediction, '.jpg')
    img_9 = cv2.imread(filename)
    img_10 = cv2.resize(img_9,(480,480),1)
    cv2.imshow('inference',img_10)
    tEnd = time.time() #計時結束

    if cv2.waitKey(1) & 0xFF == ord('q'):
        cv2.imwrite('output.jpg', img_2)
        break

cap.release()
cv2.destroyAllWindows()

print('It takes %f sec for each frame' % (tEnd - tStart))

```

Image Size: 640 x 480

It takes 0.027086 sec for each frame