



小波散射的數字分類

學校：東海大學 | 系級：應數三 | 姓名：溫宏岳 | 指導教授：黃皇男

專題：訊號與系統期末報告

目錄

- ◆ 小波圖像散射的分類模型
- ◆ 電腦眼中的數字圖形
- ◆ 卷積神經網絡 (CNN) 是什麼
- ◆ 池化 (Pooling) 產生的特徵向量
- ◆ 主成分分析模型 (PCA) 與預測
- ◆ 卷積神經網絡 (CNN) 的訓練與精確度分析
- ◆ 結論與心得
- ◆ 參考文獻
- ◆ Sample MATLAB Code

小波圖像散射的分類模型

小波圖像散射的分類模型有**數字分類**、**紋理分類**、**GPU加速比例圖**的計算和用於**聲音場景識別**的多模型後期融合系統等。

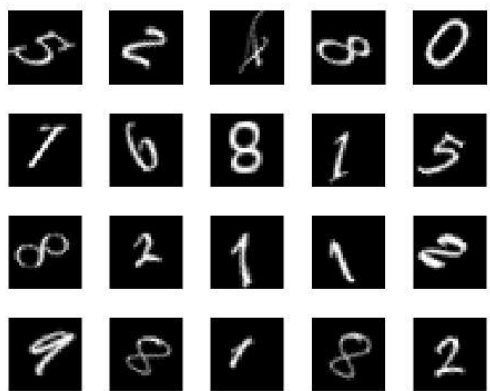


Fig1. 數字分類

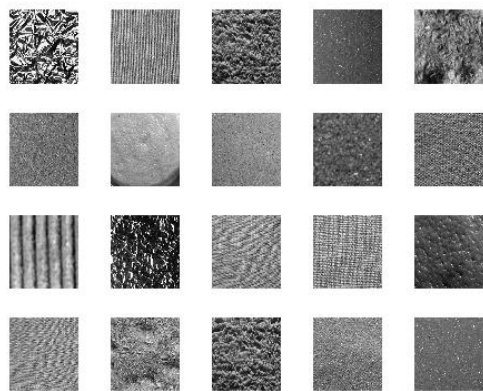


Fig2. 紋理分類

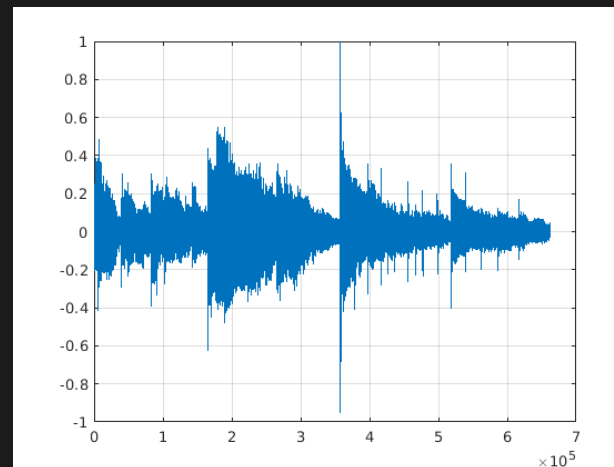


Fig3. GPU加速比例圖

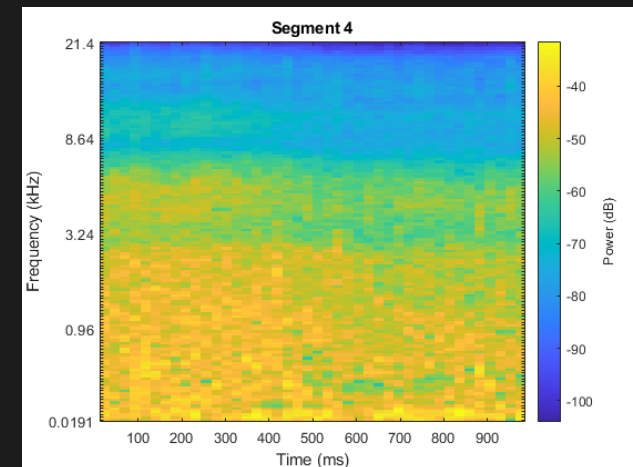


Fig4. 聲音場景識別

電腦眼中的數字圖形

一幅黑白靜止平面圖像（如照片）中各點的灰度值可用其位置坐標（ x, y ）的函數 $f(x, y)$ 來描述。

顯然 $f(x, y)$ 是二維連續函數，有無窮多個取值。這種用連續函數表示的圖像無法用電腦進行處理，也無法在各種數字系統中傳輸或存儲，所以我們必須將代表圖像的連續（模擬）信號轉變為離散（數字）信號。這樣的變換過程，稱為**圖像數字化**。

右側中的左圖是大家知道的灰階影像（8位元），分別是一個數字1和數字8（像素值皆為 $12 \times 12 = 144$ ）。人的眼睛看到的是左圖，電腦看到的是右圖。

那你們現在右側中的左圖右是怎麼顯示出來的呢？

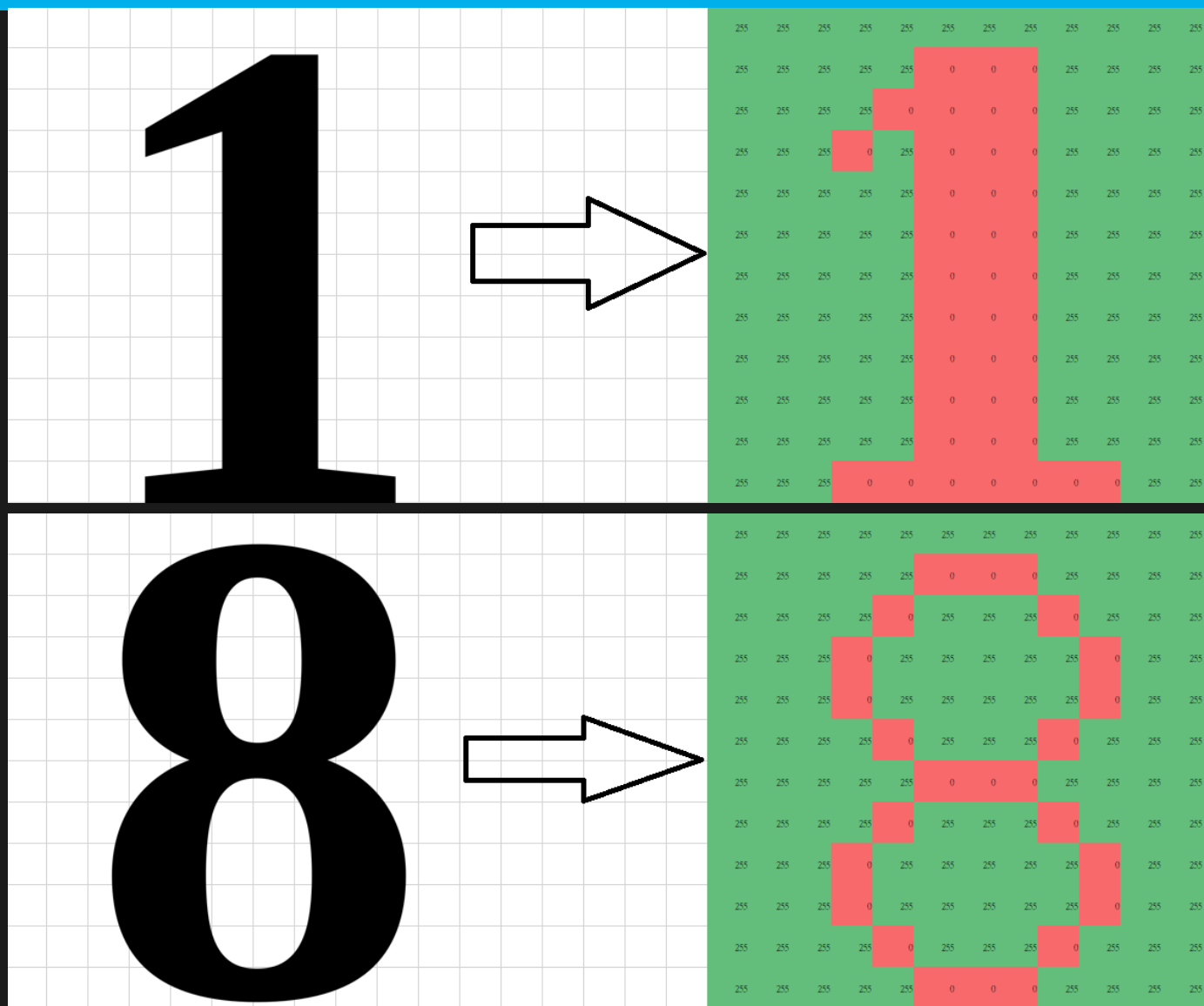


Fig5. 數字圖形

卷積神經網絡 (CNN) 是什麼

卷積神經網路 (CNN) 是法國人 Yann LaCun (揚-勒丘恩) 於貝爾實驗室任職時所提出，是一種深度學習結構， 他也是在那時蒐集與建立了 MNIST 手寫辨識資料集：卷積 (Convolution，又譯為旋積或摺積)

CNN 則是將離散卷積運算應用在人工神經網路，藉由導入多個卷積層擷取空間特徵來提升辨識率，透過隨機產生的卷積核心 (Kernel) 將一張圖片轉換成多張圖片。這些卷積核心如同濾鏡 (filter) 一般，卷積運算相當於將原始圖片通過不同濾鏡的過濾效果來提取或突顯各種空間特徵，例如邊緣與輪廓等等資訊。。

我們可將CNN分為三個層面：卷積層 (Convolution Layer)、池化層 (Pooling Layer)、全連接層 (Fully Connected Layer)

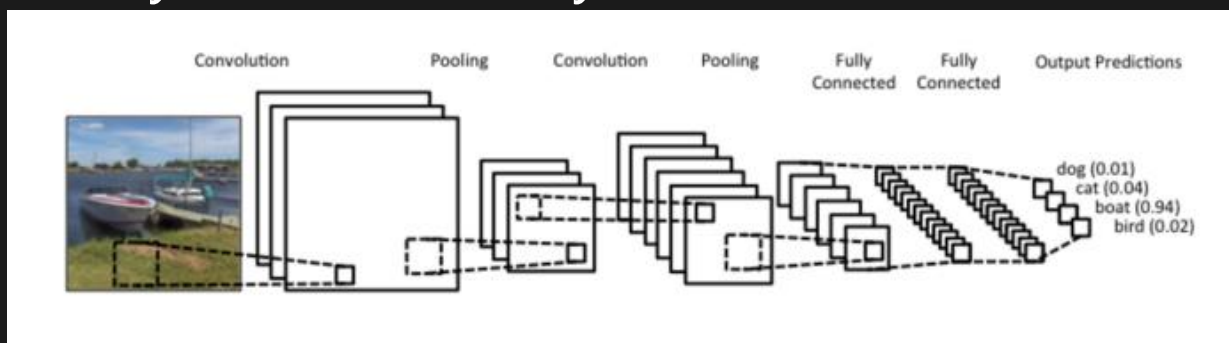


Fig6. CNN

池化 (Pooling) 產生的特徵向量

池化，目的只是在將圖片資料量減少並保留重要資訊的方法，把原本的資料做一個最大化或是平均化的降維計算。

這個例題使用10,000張圖像分成兩組，一組用於訓練，另一組用於測試。將80%的數據（即8,000張圖像）分配給訓練集，並保留其餘2,000張圖像進行測試。使用輔助函數 `helperScatImages` 去根據散射變換係數創建特徵向量。

最後，**全連接層**的部分就是將之前的結果平坦化之後接到最基本的神經網絡了

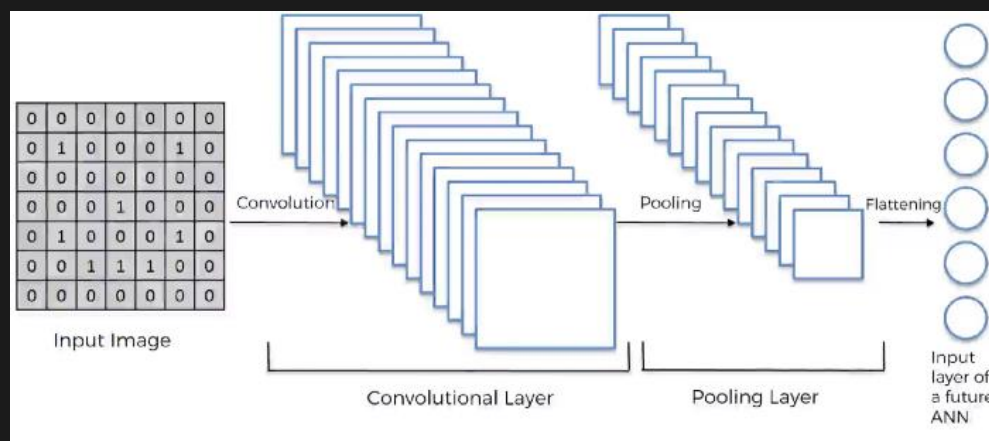


Fig7. Convolution 、 Pooling 、 Fully Connected

主成分分析模型（PCA）與預測

主成分分析（PCA）是一種統計分析、簡化數據集的方法。它利用正交變換來對一系列可能相關的變數的觀測值進行線性變換，從而投影為一系列線性不相關變數的值，這些不相關變數稱為主成分（Principal Components）。具體來說，主成分可以看做一個線性方程式，其包含一系列線性係數來指示投影方向。PCA對原始數據的正則化（regularization）或預處理敏感（相對縮放）。

PCA在數學的定義是：一個正交化線性變換，把數據變換到一個新的坐標系統中，使得這一數據的任何投影的第一大變異數在第一個坐標（稱為第一主成分）上，第二大變異數在第二個坐標（第二主成分）上，依次類推。

```
accuracy = sum(testImds.Labels == predlabels)./numel(testImds.Labels)*100  
  
accuracy = 99.6000
```

我們已經對測試數據進行了正確分類的99.6%。要查看如何對2,000張測試圖像進行分類，右側是混淆矩陣（測試集中有10個類別的200個例子）。

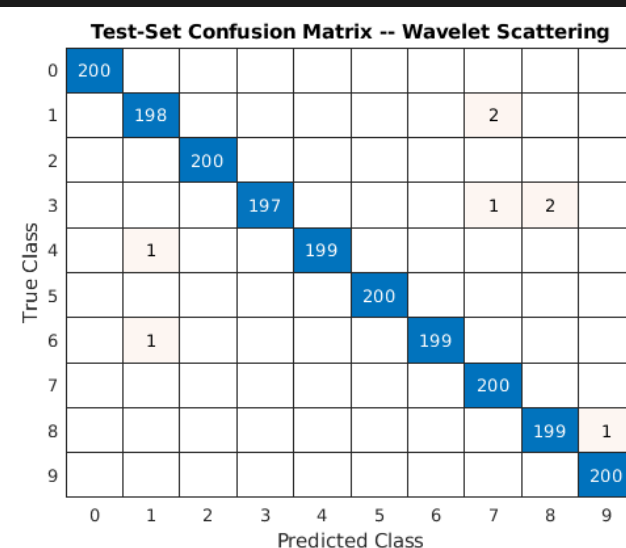


Fig8. PCA後產生的混淆矩陣

卷積神經網絡 (CNN) 的訓練與精確度分析

訓練結束時，CNN的訓練集執行率接近100%。

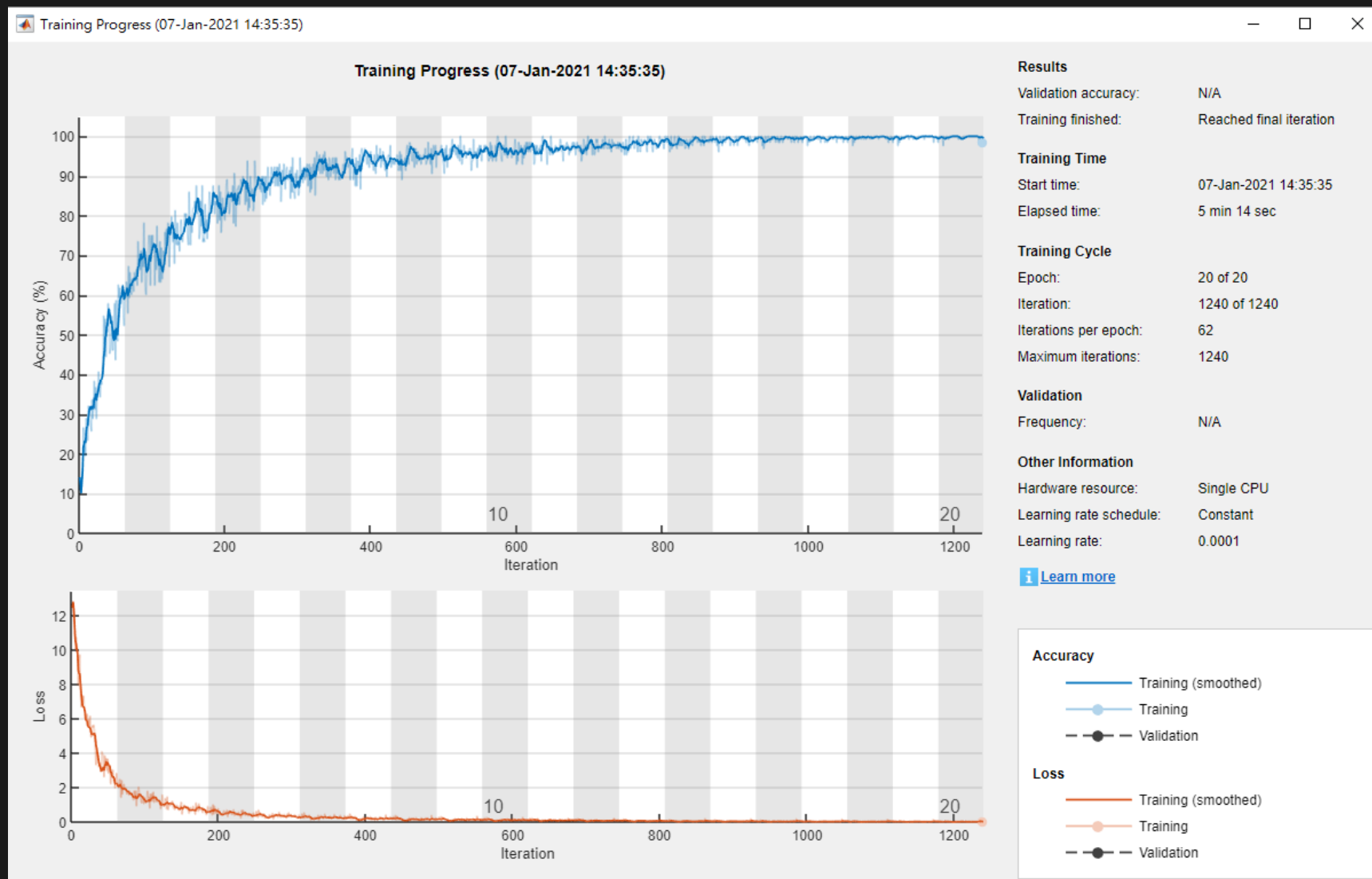


Fig9. 精確度圖

卷積神經網絡 (CNN) 的訓練與精確度分析

```
YPred = classify(net,testImds,'ExecutionEnvironment','cpu');  
DCNNaccuracy = sum(YPred == testImds.Labels)/numel(YPred)*100
```

DCNNaccuracy = 95.6500

CNN在保留的測試集上實現了95.5%的正確分類。
右側是混淆圖。

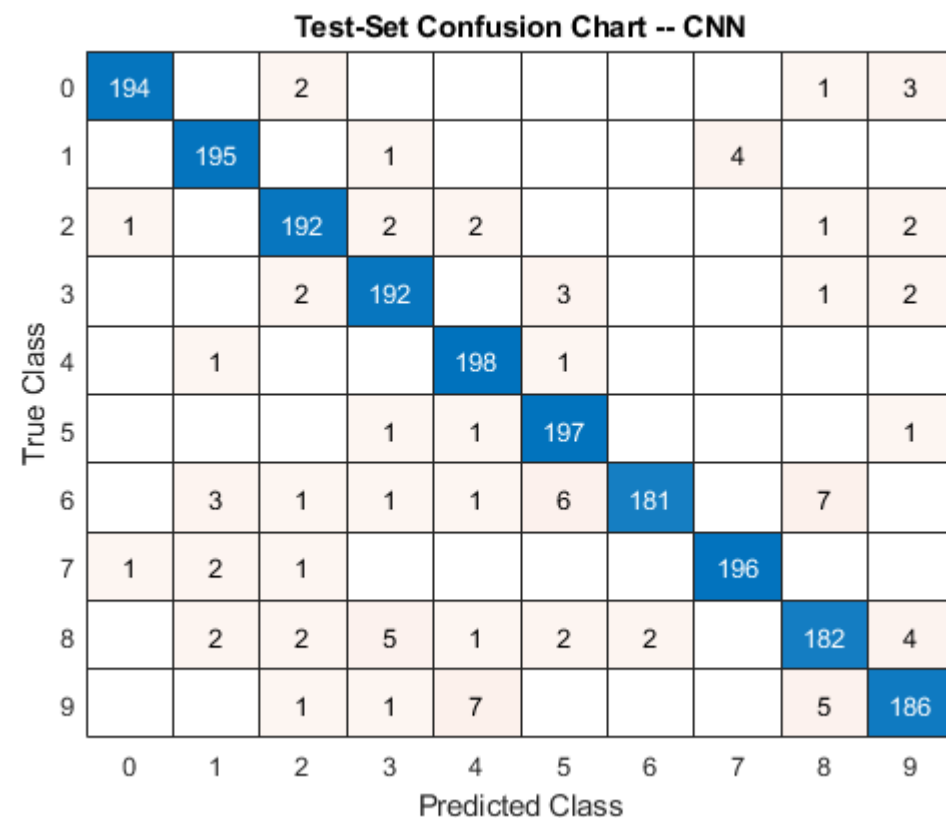


Fig10. CNN過後產生的混淆矩陣

結論與心得

這次介紹所使用小波圖像散射來創建數字圖像的低方差（low-variance）表示形式以進行分類。使用具有固定濾波器權重的散射變換和簡單的主成分分類器，我們在保留的測試集上實現了99.6%的正確分類。通過學習過濾器的簡單CNN，我們達到了95.5%的正確率。這個題目的目的僅演示如小波散射變換中提取的特徵向量可以產生強大數據來供卷積神經網絡（CNN）訓練與學習。

我們也可以推廣到更多實際層面的應用，正因為如此，涉及的理論跟操作細項會更加的複雜許多，像是可以有**制式化的字體**、**英文字體**、**手寫字體**或是**車牌辨識**等問題。



Fig11.有制式化的字體

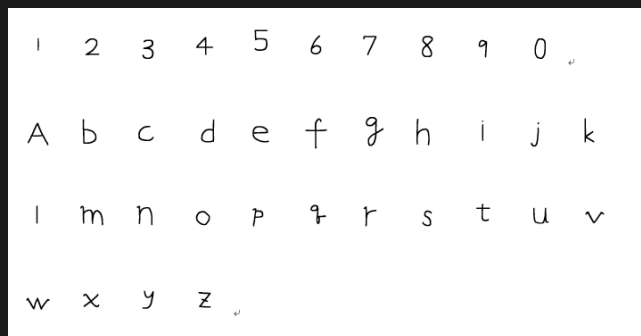


Fig12.手寫字體

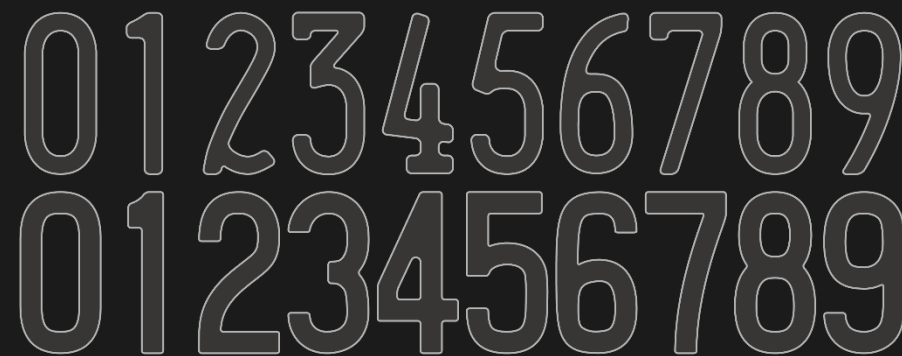


Fig13.車牌字體

參考文獻

- <https://www.mathworks.com/help/releases/R2020a/wavelet/examples/digit-classification-with-wavelet-scattering.html#WaveletScatteringDigitsExample-1>
- <https://chih-sheng-huang821.medium.com/卷積神經網路-convolutional-neural-network-cnn-卷積運算-池化運算-856330c2b703>
- <https://medium.com/jameslearningnote/資料分析-機器學習-第5-1講-卷積神經網絡介紹-convolutional-neural-network-4f8249d65d4f/>
- <http://yhhuang1966.blogspot.com/2018/04/keras-cnn.html>

Sample MATLAB Code

%Digit Images

```
digitDatasetPath =  
fullfile(matlabroot,'toolbox','nnet','nndemos','nndatasets','DigitDataset');  
Imds = imageDatastore(digitDatasetPath,'IncludeSubfolders',true,  
'LabelSource','foldernames');
```

```
figure  
numImages = 10000;  
rng(100);  
perm = randperm(numImages,20);  
for np = 1:20  
    subplot(4,5,np);  
    imshow(Imds.Files{perm(np)});  
end
```

%Wavelet Image Scattering Feature Extraction

```
sf = waveletScattering2('ImageSize',[28 28],'InvarianceScale',28, ...  
    'NumRotations',[8 8]);
```

```
if isempty(gcp)  
    parpool;  
end
```

Sample MATLAB Code

```
rng(10);
Imds = shuffle(Imds);
[trainImds,testImds] = splitEachLabel(Imds,0.8);
Ttrain = tall(trainImds);
Ttest = tall(testImds);
trainfeatures = cellfun(@(x)helperScatImages(sf,x),Ttrain,'UniformOutput',false);
testfeatures = cellfun(@(x)helperScatImages(sf,x),Ttest,'UniformOutput',false);
Trainf = gather(trainfeatures);

Trainf = gather(trainfeatures);

trainfeatures = cat(2,Trainf{:});
Testf = gather(testfeatures);

testfeatures = cat(2,Testf{:});

%PCA Model and Prediction
model = helperPCAModel(trainfeatures,30,trainImds.Labels);
predlabels = helperPCAClassifier(testfeatures,model);
```

Sample MATLAB Code

```
accuracy = sum(testImds.Labels == predlabels)./numel(testImds.Labels)*100

figure;
confusionchart(testImds.Labels,predlabels)
title('Test-Set Confusion Matrix -- Wavelet Scattering')

%CNN
imageSize = [28 28 1];
layers = [ ...
    imageInputLayer([28 28 1])
    convolution2dLayer(5,20)
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer];
```

Sample MATLAB Code

```
options = trainingOptions('sgdm', ...  
    'MaxEpochs',20,...  
    'InitialLearnRate',1e-4, ...  
    'Verbose',false, ...  
    'Plots','training-progress','ExecutionEnvironment','cpu');  
  
reset(trainImds);  
reset(testImds);  
net = trainNetwork(trainImds, layers, options);  
  
YPred = classify(net, testImds, 'ExecutionEnvironment', 'cpu');  
DCNNaccuracy = sum(YPred == testImds.Labels)/numel(YPred)*100  
  
figure;  
confusionchart(testImds.Labels, YPred)  
title('Test-Set Confusion Chart -- CNN')
```

謝謝聆聽~~