Traffic Sign Classifier

為辨識德國的43道路號誌

我使用現有的DATA SET，有訓練、驗證和測試集，基本格式如下:

Number of training examples = 34799

Number of validation examples = 4410

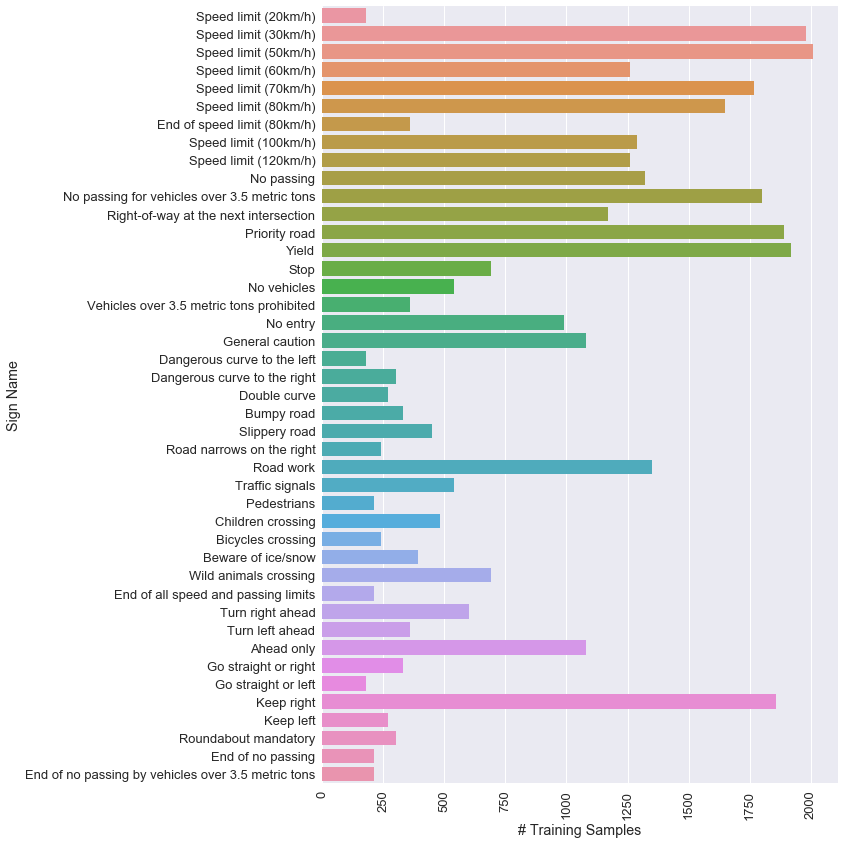
Number of testing examples = 12630

Image data shape = (32, 32, 3)

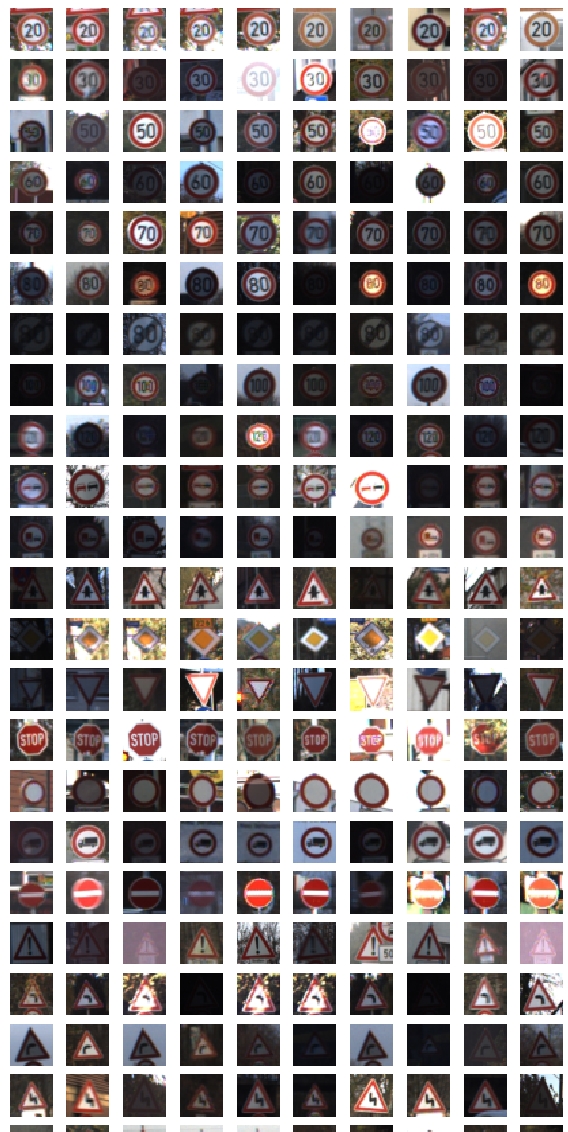
Number of classes = 43

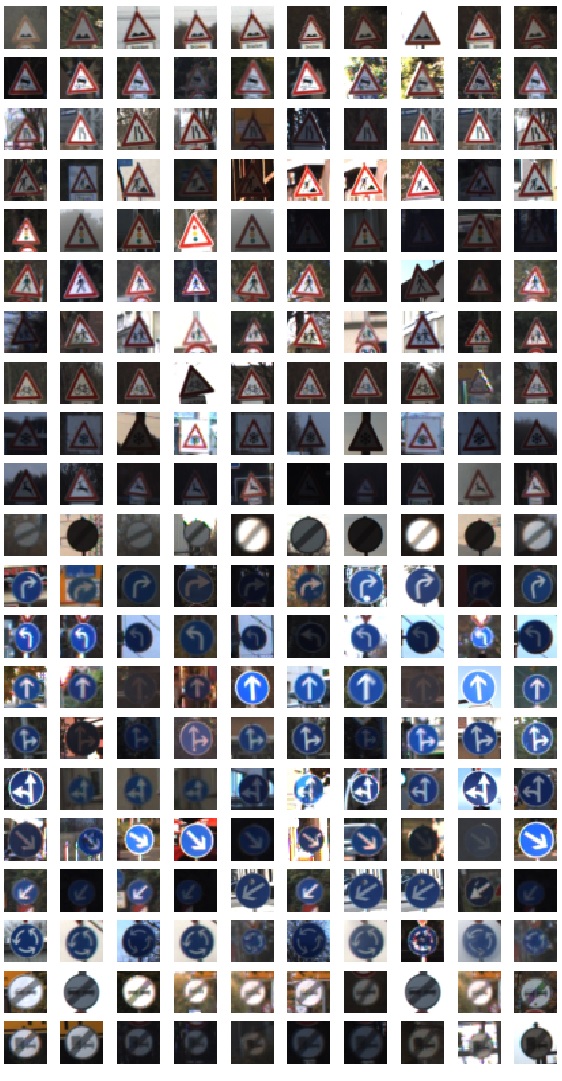
標籤有43類，RGB圖像大小32x32，

將各類的資料豐富度整理成圖表呈現:



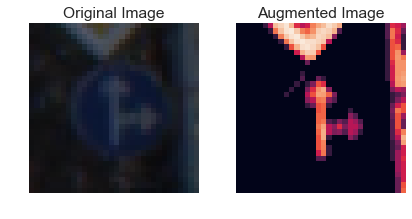
針對43類告展示10張訓練資料:



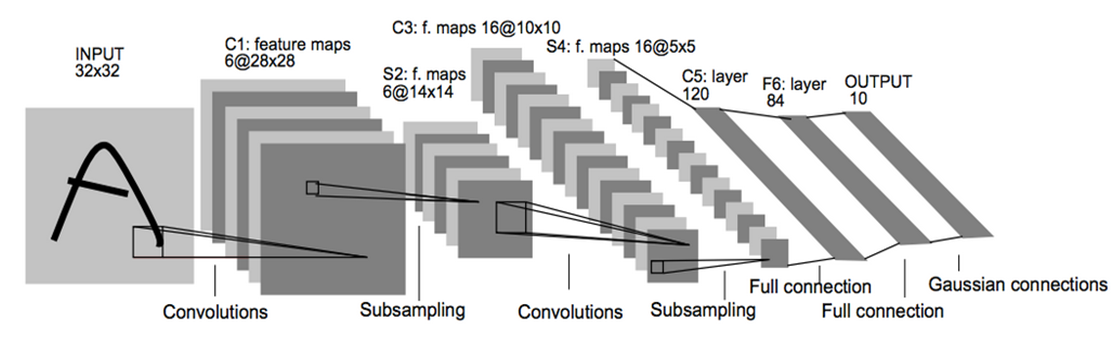


資料的前處理部分有做了幾種方法，基本轉換上包含了高斯模糊、正規化、YUV色域的轉換，其中只取的Y頻道做訓練，進階部分(下圖)還把資料做旋轉和色彩深度隨機變化，追加到訓練集當中，能提升訓練強度。





訓練的網路模型使用了LEnet五層架構:28x28



包含的池化、全連接和最後輸出層

基本流程為:

Layer 1: Convolutional. Input = 32x32x3. Output = 28x28x6 使用非線性函數relu

Pooling. Input = 28x28x6. Output = 14x14x6

Layer 2: 二次Convolutional. Output = 10x10x16 使用非線性函數relu

二次Pooling. Input = 10x10x16. Output = 5x5x16

SFlatten. Input = 5x5x16. Output = 400

Layer 3: Fully Connected. Input = 400. Output = 120 使用非線性函數relu

Layer 4: 二次Fully Connected. Input = 120. Output = 84 使用非線性函數relu

Layer 5: Fully Connected. Input = 84. Output = 43 依機率最後分成43類輸出結果

比較的部分是用softmax\_cross\_entropy來評估錯誤，使用tf.reduce\_mean從訓練資料中做cross entropy平均化，梯度下降優化使用了adaptive moment estimation 動態自適應的標準方法，透過動態超參數的調整來改善傳統隨機梯度下降的問題，主要訓練的參數值為15個 EPOCH: learning rate = 0.001每次取batch\_size = 128 ,

Dropout為0.7:用機率值去除部分計算結果，防止過度仰賴局部訓練效果，能加強泛化能力，進而避免過度擬合，

最後呈現的精準度、耗時和結果曲線等資訊就如下所示:

2020-05-16 18:27:45.647295 Training... dropout = 0.3 , batch\_size = 128 , learning rate = 0.001

2020-05-16 18:31:31.088949 EPOCH 1 - 225 sec ...

2020-05-16 18:31:31.089951 Training error = 0.233 Validation error = 0.253

2020-05-16 18:35:14.144010 EPOCH 2 - 448 sec ...

2020-05-16 18:35:14.145011 Training error = 0.135 Validation error = 0.152

2020-05-16 18:38:57.056148 EPOCH 3 - 671 sec ...

2020-05-16 18:38:57.057149 Training error = 0.097 Validation error = 0.107

2020-05-16 18:42:39.813176 EPOCH 4 - 894 sec ...

2020-05-16 18:42:39.814177 Training error = 0.084 Validation error = 0.100

2020-05-16 18:46:24.803638 EPOCH 5 - 1119 sec ...

2020-05-16 18:46:24.805640 Training error = 0.082 Validation error = 0.083

2020-05-16 18:50:22.839532 EPOCH 6 - 1357 sec ...

2020-05-16 18:50:22.840533 Training error = 0.072 Validation error = 0.075

2020-05-16 18:54:06.750374 EPOCH 7 - 1581 sec ...

2020-05-16 18:54:06.751375 Training error = 0.068 Validation error = 0.073

2020-05-16 18:57:52.767700 EPOCH 8 - 1807 sec ...

2020-05-16 18:57:52.768701 Training error = 0.066 Validation error = 0.073

2020-05-16 19:01:36.151170 EPOCH 9 - 2030 sec ...

2020-05-16 19:01:36.152171 Training error = 0.066 Validation error = 0.070

2020-05-16 19:05:20.018981 EPOCH 10 - 2254 sec ...

2020-05-16 19:05:20.018981 Training error = 0.066 Validation error = 0.078

2020-05-16 19:09:06.675804 EPOCH 11 - 2481 sec ...

2020-05-16 19:09:06.676804 Training error = 0.061 Validation error = 0.070

2020-05-16 19:12:49.641979 EPOCH 12 - 2704 sec ...

2020-05-16 19:12:49.641979 Training error = 0.071 Validation error = 0.069

2020-05-16 19:16:32.598148 EPOCH 13 - 2927 sec ...

2020-05-16 19:16:32.599149 Training error = 0.061 Validation error = 0.072

2020-05-16 19:20:18.160153 EPOCH 14 - 3152 sec ...

2020-05-16 19:20:18.160153 Training error = 0.060 Validation error = 0.068

2020-05-16 19:24:03.343893 EPOCH 15 - 3377 sec ...

2020-05-16 19:24:03.344893 Training error = 0.059 Validation error = 0.065

2020-05-16 18:27:45.647295 Training... dropout = 0.3 , batch\_size = 128 , learning rate = 0.001



Accuracy Model On Training Images: 0.94

Accuracy Model On Validation Images: 0.94

測試集的準確度為:

Accuracy Model On Test Images: 0.91

結果算是能夠接受的範圍，最後從網路上找了六張路標來做實驗，

判斷結果和預測出來的類別機率如下:



Accuracy Model On Internet Images: 0.8333333134651184

Top 5 Labels for image 'Speed limit (30km/h)':

- 'Speed limit (30km/h)' with prob = 0.97

- 'Speed limit (50km/h)' with prob = 0.02

- 'Speed limit (80km/h)' with prob = 0.00

- 'Speed limit (20km/h)' with prob = 0.00

- 'Speed limit (100km/h)' with prob = 0.00

Top 5 Labels for image 'Speed limit (60km/h)':

- 'Speed limit (60km/h)' with prob = 0.45

- 'Speed limit (30km/h)' with prob = 0.37

- 'Speed limit (50km/h)' with prob = 0.05

- 'End of speed limit (80km/h)' with prob = 0.03

- 'Keep left' with prob = 0.03

Top 5 Labels for image 'No passing':

- 'No passing' with prob = 0.86

- 'End of no passing' with prob = 0.13

- 'No passing for vehicles over 3.5 metric tons' with prob = 0.00

- 'No entry' with prob = 0.00

- 'Yield' with prob = 0.00

Top 5 Labels for image 'Road work':

- 'Road work' with prob = 1.00

- 'Ahead only' with prob = 0.00

- 'Wild animals crossing' with prob = 0.00

- 'Go straight or left' with prob = 0.00

- 'Priority road' with prob = 0.00

Top 5 Labels for image 'Ahead only':

- 'Ahead only' with prob = 1.00

- 'Go straight or right' with prob = 0.00

- 'Turn left ahead' with prob = 0.00

- 'Road work' with prob = 0.00

- 'Keep right' with prob = 0.00

Top 5 Labels for image 'Roundabout mandatory':

- 'Priority road' with prob = 0.18

- 'End of no passing' with prob = 0.10

- 'No entry' with prob = 0.09

- 'Roundabout mandatory' with prob = 0.07

- 'Turn left ahead' with prob = 0.07

準度為0.83左右，只有Roundabout mandatory迴轉判斷失敗，

我覺得可能是因為這類別的訓練資料本身就比較少的關係，

在網路架構不動的情況下能做的是異樣資料擴充、最佳化參數調整，像是前處理的部分和SGM低度下降的方式，最後出來的結果泛化程度會比較好。