Explain how to execute program clearly

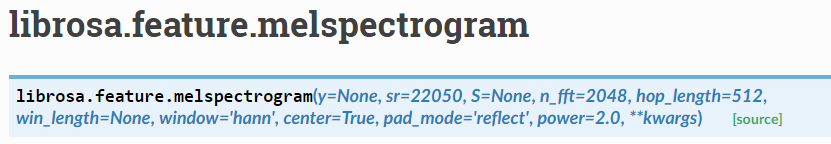
The all source code are in r08921005.ipynb, TA can just open it, and realize what I do.

The important will be described as below.The architecture of my program in folder:

DSP\_HW->train(\*.jpg, \*.npy), test(\*.jpg, \*.npy), val(\*.jpg, \*.npy) ()→is the content in folder

|  |  |
| --- | --- |
| 將npy轉到spectrogram    Take test for example,  the output will be as below | In this part, we can load the model which we trained before, the upper code just take “weight.pth” for example.  Then, we can load the model to validate the \*jpg in val to output is validation accuracy. |
| Put result in dictionary type and write in csv file | |

1. Settings for generating spectrograms (at least 2 settings)
   1. 利用librosa.feature.melspectrogram python library 將個npy檔轉成jpg檔



總共做了兩次第一次都使用default,第二次將hop\_length調為1024使Overlap

從75%變為50%

Overlap 主要被 hop\_length parameter控制

EX:The default frame length is 2048 (for STFT operations), and the default hop is 512, which results in 75% overlap.

* Exp 1. Generate a magnitude spectrum using the hanning window function with a FFT length of 2048 and overlapping of 75% between segments
* Exp 2. Generate a phase spectrum using the hanning window function with a FFT length of 2048 and overlapping of 50% between segments

在network 不變的情況下只是單純dataSource 下去train 得到不同的結果

overlapping 50% 的validation 效果比overlapping 75%原本差

|  |  |
| --- | --- |
| Exp 1 | Exp 2 |
| https://scontent-tpe1-1.xx.fbcdn.net/v/t1.15752-9/78612520_1390759347772063_2127656133341478912_n.png?_nc_cat=108&_nc_ohc=ywVO0OJYQHkAQlqN9hR-7HUuFletg-veiJuCPylgozNBDWZi0xjwXRJIw&_nc_ht=scontent-tpe1-1.xx&oh=1921165f13f2b82a41b51ec8027289e1&oe=5E8C211F |  |
| validation accuracy:0.925129 | validation accuracy:0.835215 |

1. Settings for your neural network (at least 1 setting)
   1. Please include implementation details like architecture(LeNet/VGGNet/…), optimizer(Adam/SGD/…), initialization, learning rate, etc.
   2. 全部的實驗都在LeNet架構下完成(在validate階段測其實都差不多的結果，唯一很特別是:SGD, learning rate :0.001可能overfitting了!最後accuracy只有0.088640)

|  |  |
| --- | --- |
| Exp 1 optimizer:SGD, learning rate :0.05  https://scontent-tpe1-1.xx.fbcdn.net/v/t1.15752-9/78612520_1390759347772063_2127656133341478912_n.png?_nc_cat=108&_nc_ohc=ywVO0OJYQHkAQlqN9hR-7HUuFletg-veiJuCPylgozNBDWZi0xjwXRJIw&_nc_ht=scontent-tpe1-1.xx&oh=1921165f13f2b82a41b51ec8027289e1&oe=5E8C211F | Exp 2 optimizer:SGD, learning rate :0.001 |
| Exp 3 optimizer: Adam, learning rate :0.05 | Exp 4 optimizer: Adam, learning rate :0.001 |

* 1. Report the validation accuracy based on different settings.
     + 上個part 已經對不同種dataset 得出來的validation accuracy做過比較，此part不再描述，主要針對不同optimizer，learning rate,做比較

1. What I have learned for this project.(Difficulties, interesting things, or special techniques)

我將分為以下幾點做感想整理

* 1. Signal vs. Spectrum
* 本次作業學到signal 與spectrum 之間的關係，實踐了課堂上所學所教，非常有感覺
  1. Neural Networks
* 對於Neural Networks、CNN的深入理解和實作，也認識到各種不同的Network 架構 EX: LeNet/VGGNet/…
* 因為本次作業有機會讓我去對神經網路有更深入研究，甚至去看了本次作業主要所採用的Lenet 論文:<http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf>
  1. GPU vs. CPU
* 本次作業我採用GPU 來train 然後發現其實只要改一下.cuda() 或 .cpu()其實大多的東西不太用改，滿方便的
* 上網survey一下發現GPU 比CPU 的運算速度好以外，原來他train 出來的model 也會比較好，所以validation accuracy也相較好了一點，但就真的只有一點，有可能是因為本次作業運算量比較小。
* 看了一下我電腦規格發現其實有可能是我的CPU本身就已經屬於高規格，所以可能不會效能差太多

|  |  |
| --- | --- |
|  |  |

* 1. To be Improve:或許dataset的座標軸，label，colormap 可以拿掉使判斷更精準