ML2018FALL Final Proposal

1. 隊名及隊員

隊名 ：

NTU_r06922134_台北大冒險

隊員 ：

葉沛陽 r06922134
湯梵平 r06922120
廖彥綸 b05902001
林文炤 b05901157

2. 所選擇的題目

Human Protein Atlas Image Classification

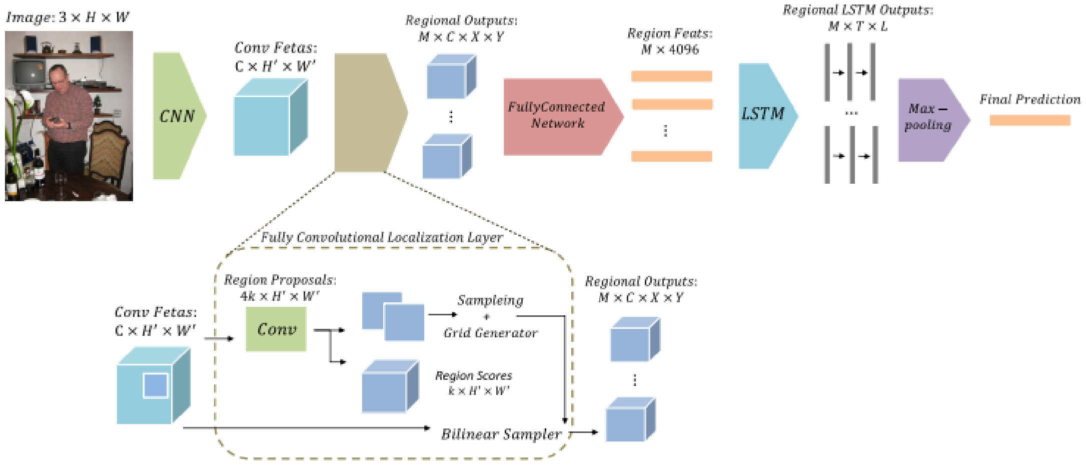
3. problem study:

第一篇 paper 題目:

Multilabel Image Classification With Regional Latent Semantic Dependencies

1. Regional Latent Semantic Dependencies (RLSD) model

使用 Regional Latent Semantic Dependencies (RLSD) model，考慮到除了 label visual 會影響分類 semantic dependencies 也會影響，所以這篇 paper 另外使用了 RNN 來抓 labels 之間的關聯。



2. Convolution Feature 部分:

使用 VGGNet 包含 13 個 convolution layers(3*3 kernel sizes) 和 五個 (2*2) max-pooling layer,然後使用 ReLU 跟 下圖的 loss function。

$$L(b,g) = \sum_{i \in x,y,w,h} \mathrm{Smooth}_{L_1}(b_i, v_i)$$

$$\mathrm{Smooth}_{L_1} = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise.} \end{cases}$$

3. Box sampling and bilinear interpolation :

使用 subsampling 來降低資料大小再傳進 LSTM. 在 test 階段，使用 non-maximum suppression 來挑選 the top M highest ranked proposals.

4. Fully-Connected Network:

用兩層 4096 維 fully-connected layers 跟 dropout.

5. Max-Pooling:

使用 Max-Pooling 因為較 average-Pooling 適合用來消除預測噪音， the fusion layer 的 output 被餵進 a multi-way softmax layer with the squared loss as the cost function, which is defined as:

$$J = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{L}\left(p_i^j - \hat{y}_i^j\right)^2$$

where y is ground truth and p is prediction.

6. Pre-train 方面:

a. localization layer is pre-trained on the Visual Genome region caption dataset
b. the LSTM is first pre-trained on the global image without region proposals, where every time step has the global image label as ground truth to compute loss as the initial of RLSD.

7. 參數方面:

label embedding size is 64.
one-layer LSTM(memory cell size is 512).
Optimization 用 SGD 以及 learning rate 為 0.00001.

8. 結論:

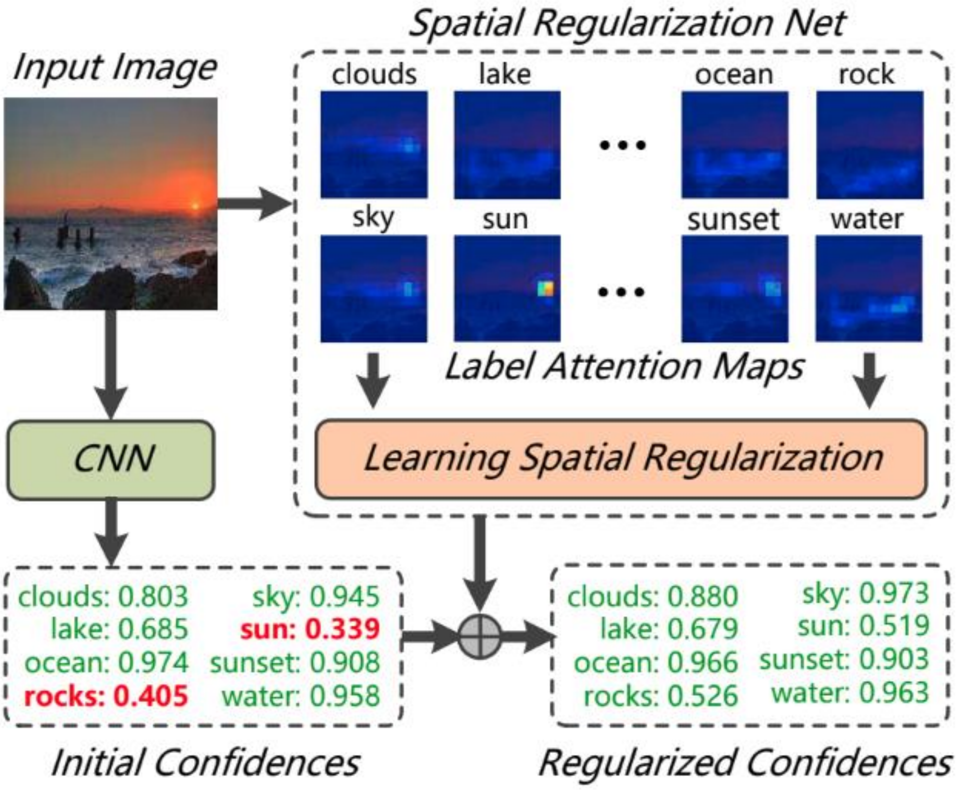
RLSD 非常適合預測小物品以及場景中物品有緊密關聯的影像。

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8310600&fbclid=IwAR3TOZVVsQphImkUPBcIZSgp3EU224RVebJvdfqOtLNLHkYywtyf2OACsgU&tag=1

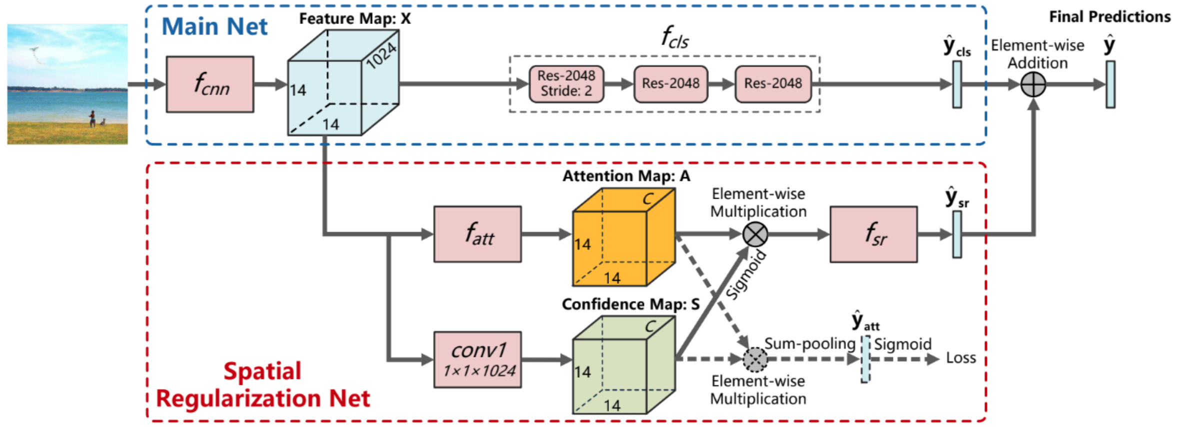
第二篇 paper 題目:

Learning Spatial Regularization with Image-level Supervisions for Multi-label Image Classification

[summary]
In this paper, They propose Spatial Regularization Network (SRN) that can generate attention maps for all labels and models the latent relations between them.Generally, spatial annotations of the labels are unavailable, thus traditional approaches are unable to model the relations in terms of spatiality.
[method keypoint]

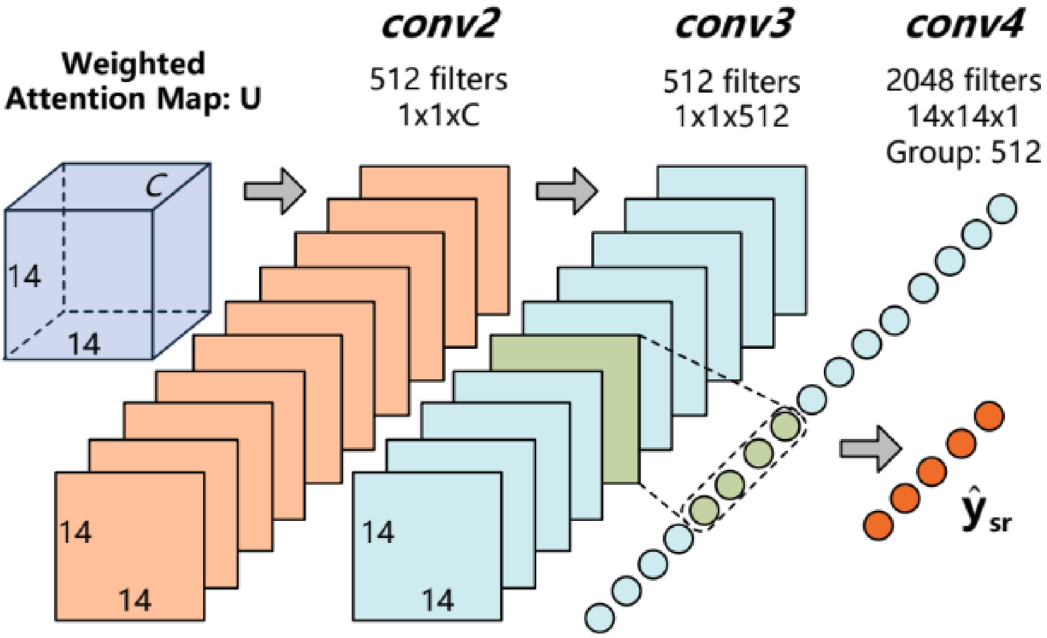


Spatial Regularization Net they proposed learns relations between semantics and spatial labels from label attention maps with only image-level supervisions.



Their SRN mainly based on Resnet-101 and learns each independent classfier for each label.The proposed SRN captures spatial and semantic relations of labels with attention mechanism.

- When there exists a label in image, it focuses more on related area.
- Label attention maps encode rich spatial information oflabels.They can be used to generate more robust spatial regularizations for labels.
- Since the weighted attention maps for all labels are spatially aligned, it is easy to capture their relative relations with stacked convolution operations. The convolutions should have large enough receptive fields to capture the complex spatial relations between the labels.
- They propose to decouple semantic relation learning and spatial relation learning in different convolution layers.The intuition is that one label may only semantically relate to a small number of other labels, and measuring spatial relations with those unrelated attention maps is unnecessary.



Detailed information for attention maps, conv4 is composed of single-channel filters.

fcnn - Resnet

- input: image, 224*224*3
- output: feature map X, 14*14*1024

fcls - multi-label classification

- Res-2048(stride=2) - Res-2048 - Res-2048
- intput: feature map X, 14x14x1024
- output: confidence interval of predicted label

fatt - Attention Maps

- Conv(512, kenel 1x1) - Conv(512, kenel 3x3) - ReLU - Conv(C, kernel 1x1) - Softmax
- input: feature map X, 14x14x1024
- output: label attention values Z, 14x14xC；final label attention maps A, 14x14xC. A=Softmax(Z)

fsr - Spatial Regularizations

- input: label attention maps A, 14x14xC.confidence map S, 14x14xC (from feature map X and Conv(C, kernel1x1))
- output: weighted attention maps U, 14x14xC

[training]

They choose cross-entropy loss.

four stages:

1.only train main network, based on Resnet and pretrained on ImageNet, fcnn and fcls.

2.Fix fcnn and fcls, but train fatt.

3.Fix fcnn, fcls and fatt, but train fsr.

4.Train whole network.

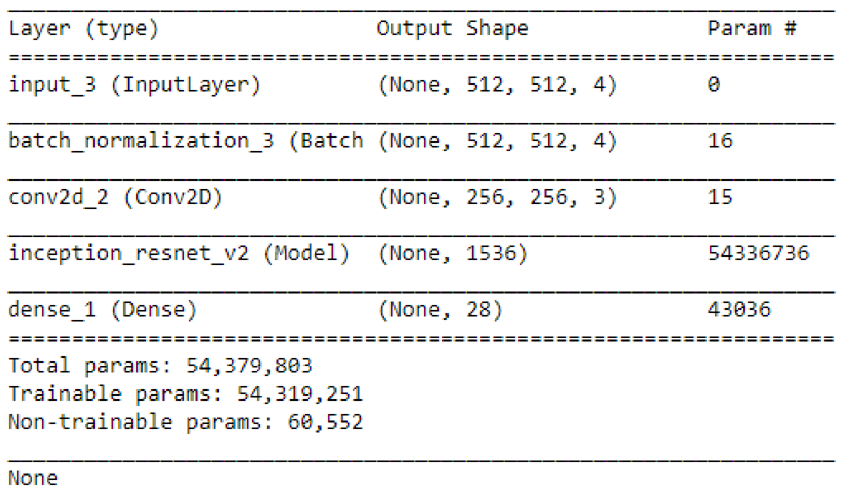
reference:

https://arxiv.org/pdf/1702.05891.pdf

4. proposed method:

目前初步將 RGBY 四個檔案當作四個 channels，直接讀圖檔 512*512 沒有 resize。

下圖為目前使用的 model 架構，主要使用一層 CNN 加上 keras 上 pretrained 好的 Inception_resnet_v2 架構，最後再加上一層 28 output 的 Dense，分類是採用 softmax，threshold 取 0.05。



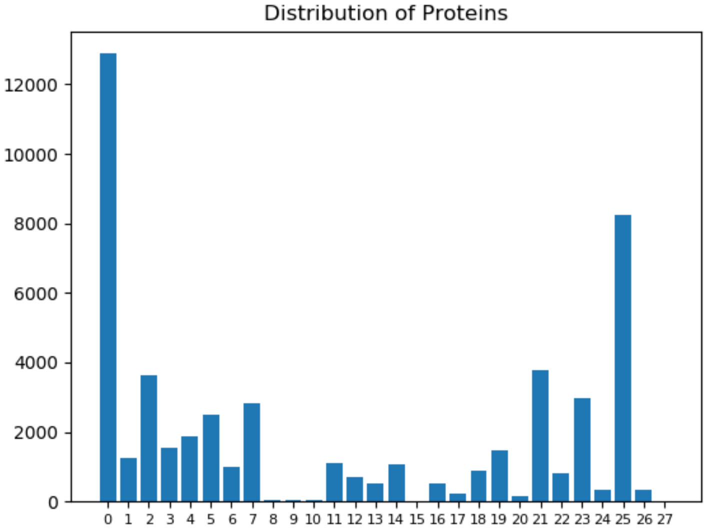
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_3 (InputLayer)         (None, 512, 512, 4)       0         
_________________________________________________________________
batch_normalization_3 (Batch (None, 512, 512, 4)       16        
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 256, 256, 3)       15        
_________________________________________________________________
inception_resnet_v2 (Model)  (None, 1536)              54336736  
_________________________________________________________________
dense_1 (Dense)              (None, 28)                43036     
=================================================================
Total params: 54,379,803
Trainable params: 54,319,251
Non-trainable params: 60,552
_________________________________________________________________
None
```

目前對於 RGBY 四個檔案直接使用，之後可能嘗試對他們作前處理，transformation 和 normalization。

目前採用 softmax 之後也打算改為 sigmoid 試看看有無改善。

目前也有發現資料有很嚴重的 Imbalanced 問題，之後可能要想辦法處理，目前想到的是針對每個 label 給予不同 class weight。

也打算針對每個 label 給予不同的 threshold，看看結果有無改善。


Distribution of Proteins
12000
10000
8000
6000
4000
2000
0
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27