Machine Learning HW4

MLTAs
ntumlta2019@gmail.com

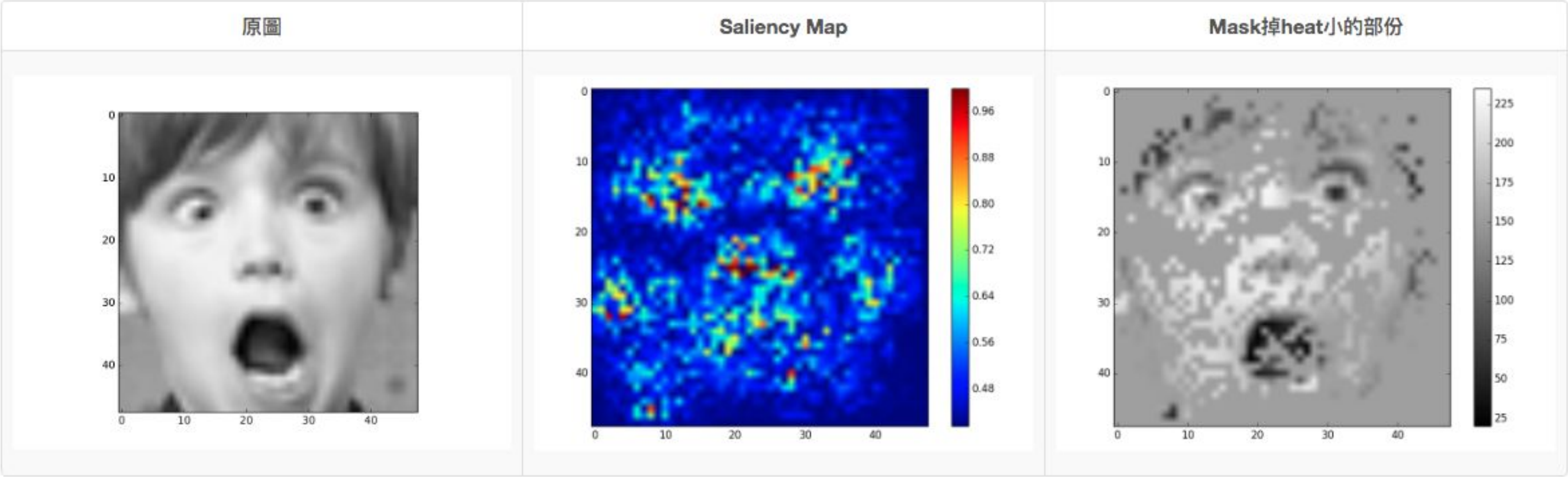
Outline

1. Task Introduction
2. Task1 - Sailency Map
3. Task2 - Filter Visualization
4. Task3 - Lime
5. FAQ

Task - Explain your model

Task1 - Saliency Map

Compute the gradient of output category with respect to input image.

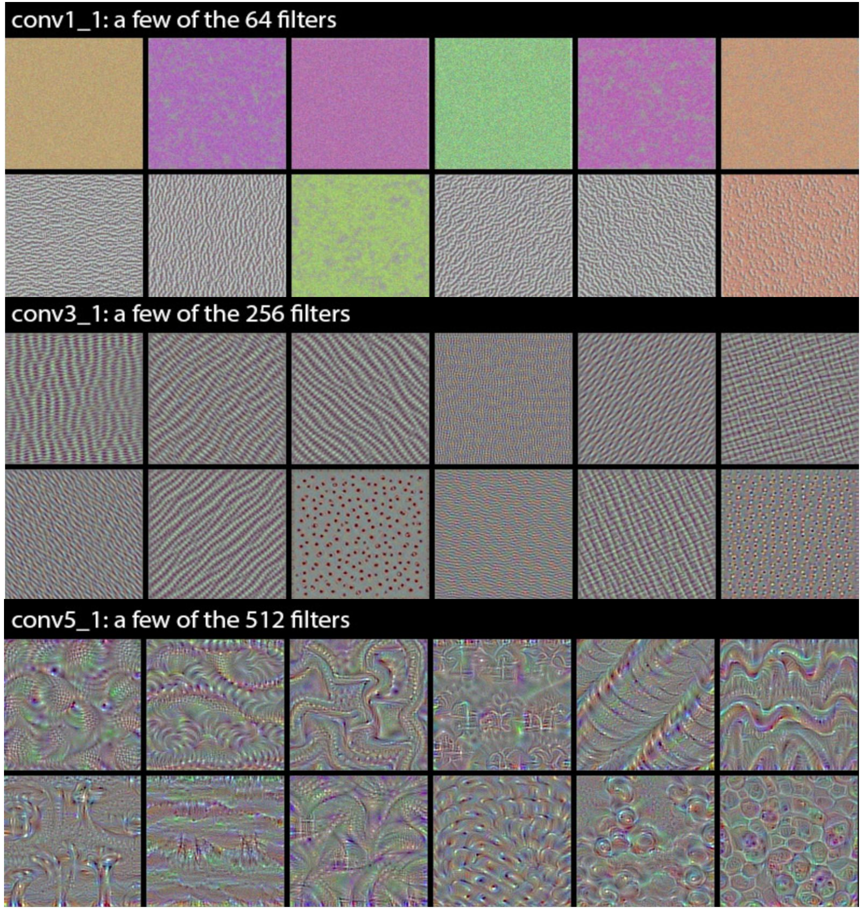


原圖 | Saliency Map | Mask掉heat小的部份

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps:
https://arxiv.org/pdf/1312.6034v2.pdf

Task2 - Filter Visualization

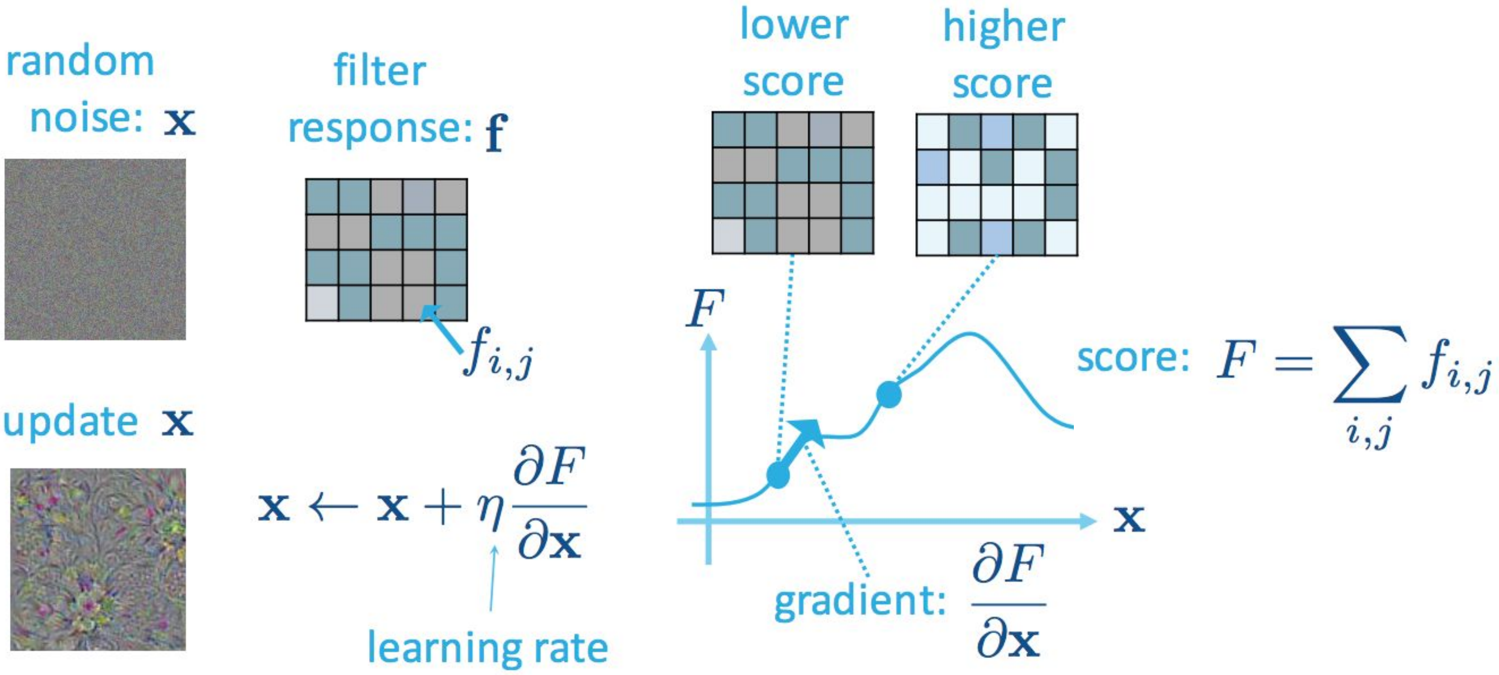
- Use **Gradient Ascent** method to find the image that activates the selected filter the most and plot them (start from white noise).


conv1_1: a few of the 64 filters
conv3_1: a few of the 256 filters
conv5_1: a few of the 512 filters

Ref:
https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html

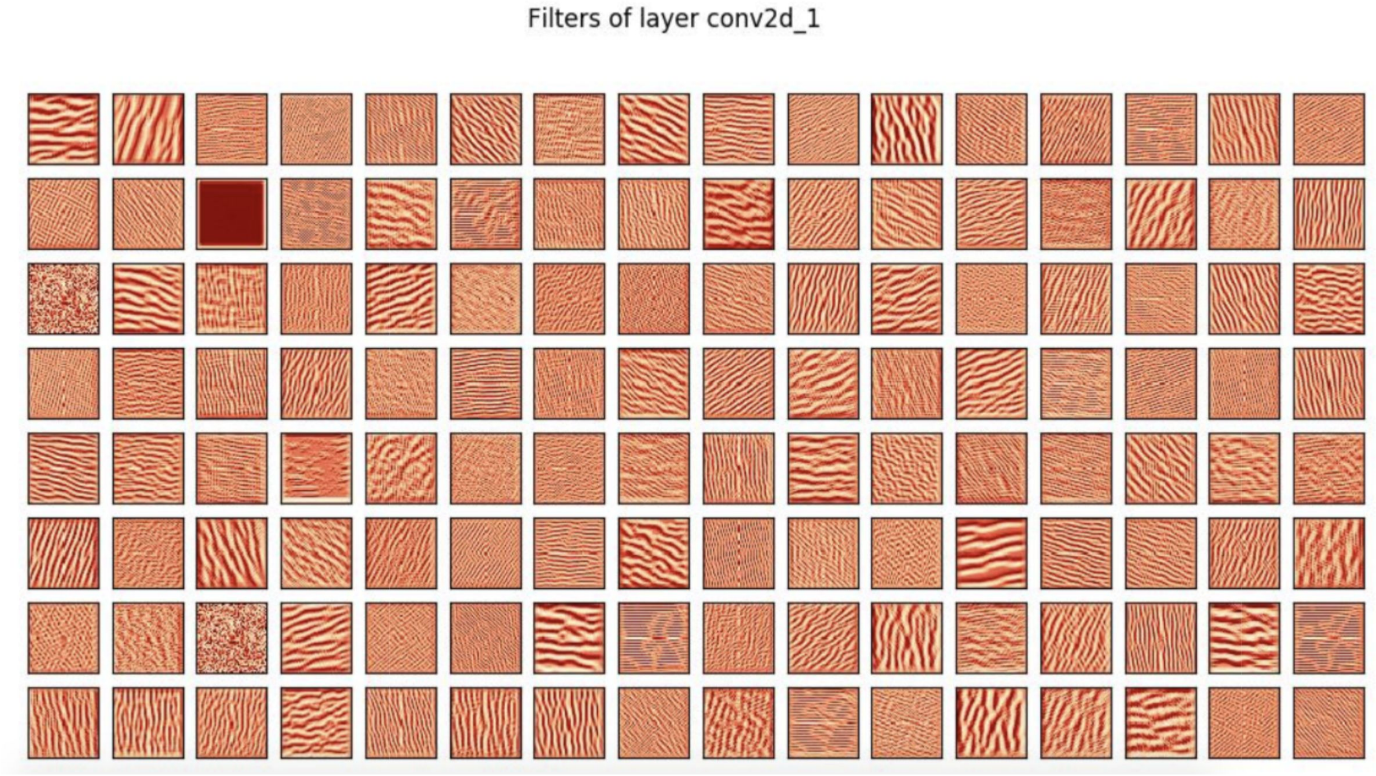
Filter Visualization

- Gradient Ascent : Magnify the filter response

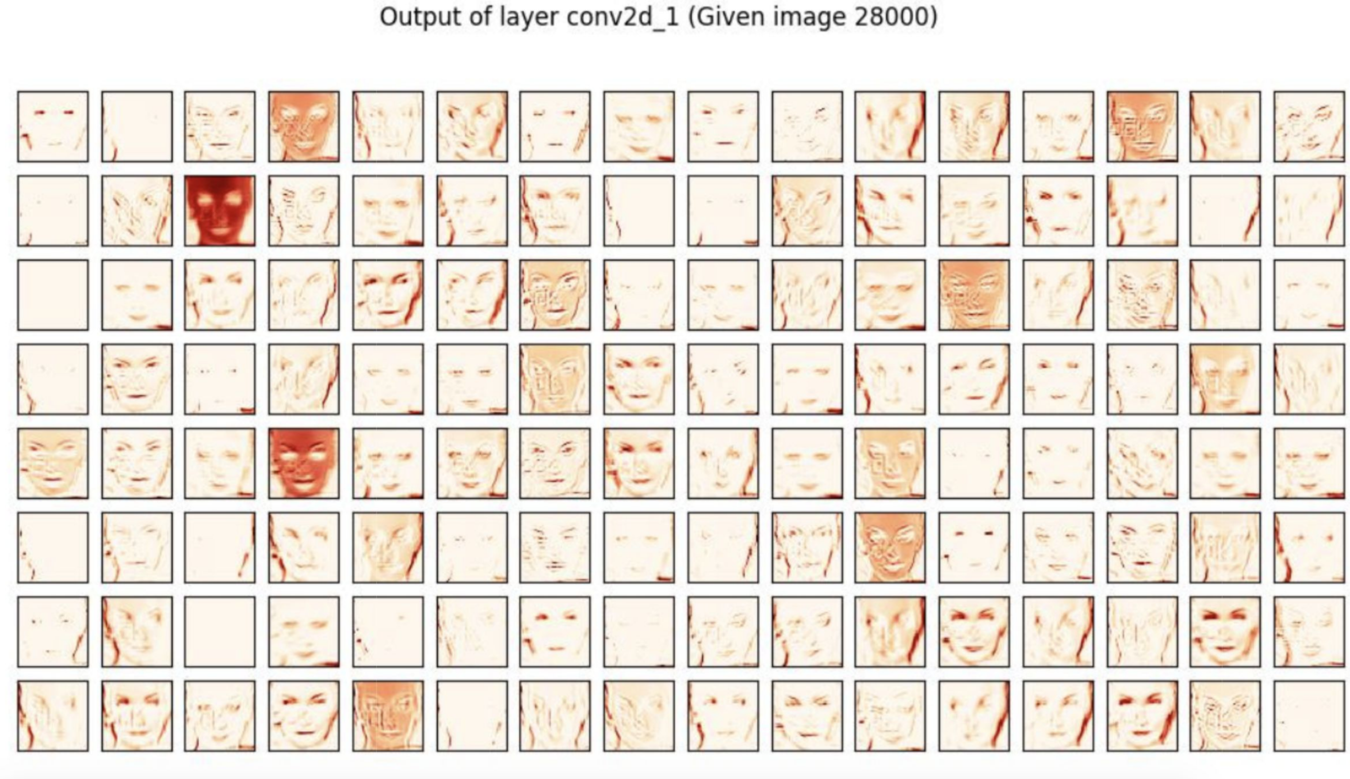

random noise: $\mathbf{x}$
filter response: $\mathbf{f}$
$f_{i,j}$
update $\mathbf{x}$
$\mathbf{x} \leftarrow \mathbf{x} + \eta \frac{\partial F}{\partial \mathbf{x}}$
learning rate
lower score
higher score
F
$\mathbf{x}$
score: $F = \sum_{i,j} f_{i,j}$
gradient: $\frac{\partial F}{\partial \mathbf{x}}$

https://www.csie.ntu.edu.tw/~yvchen/f105-adl/doc/161103_ConvolutionalNN.pdf

Filter Visualization



Filter Visualization

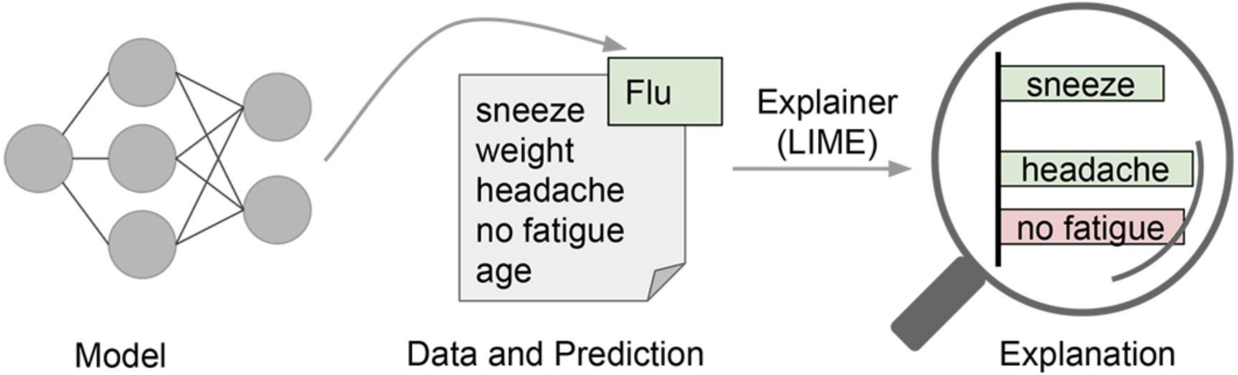


Output of layer conv2d_1 (Given image 28000)

Task-3 Lime

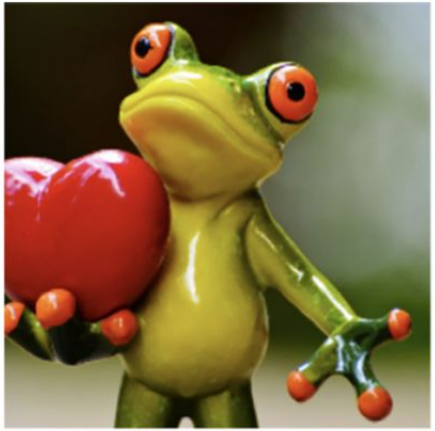
Local Interpretable Model-Agnostic Explanations

To approximate a black-box model by a simple model locally

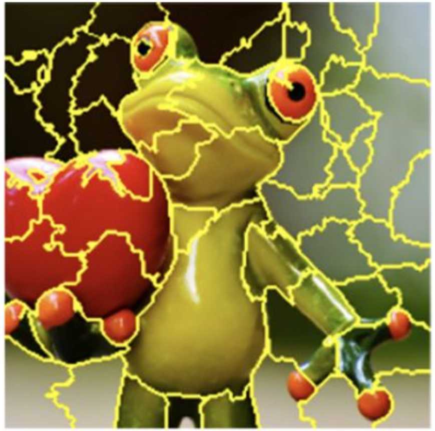


Ref: "Why Should I Trust You?": Explaining the Predictions of Any Classifier

Lime - 1/3

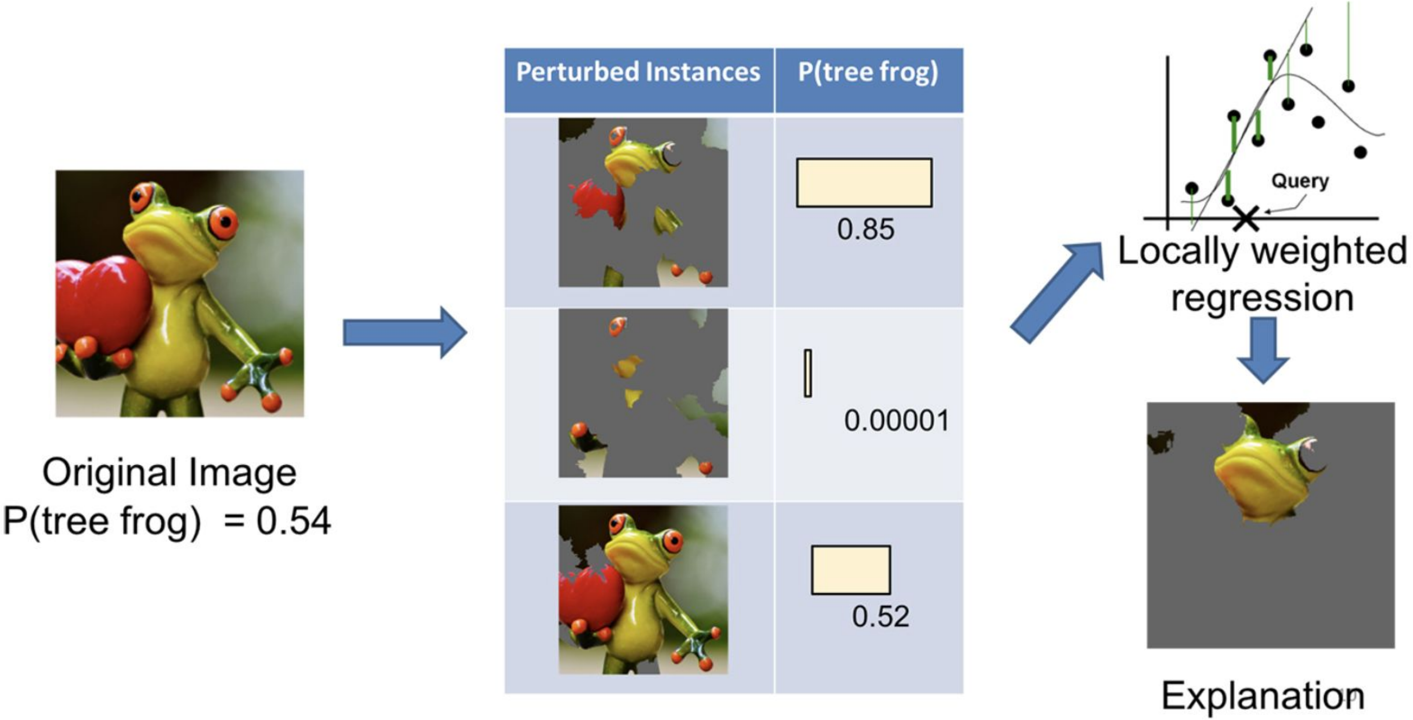


Original Image



Interpretable Components

Lime - 2/3



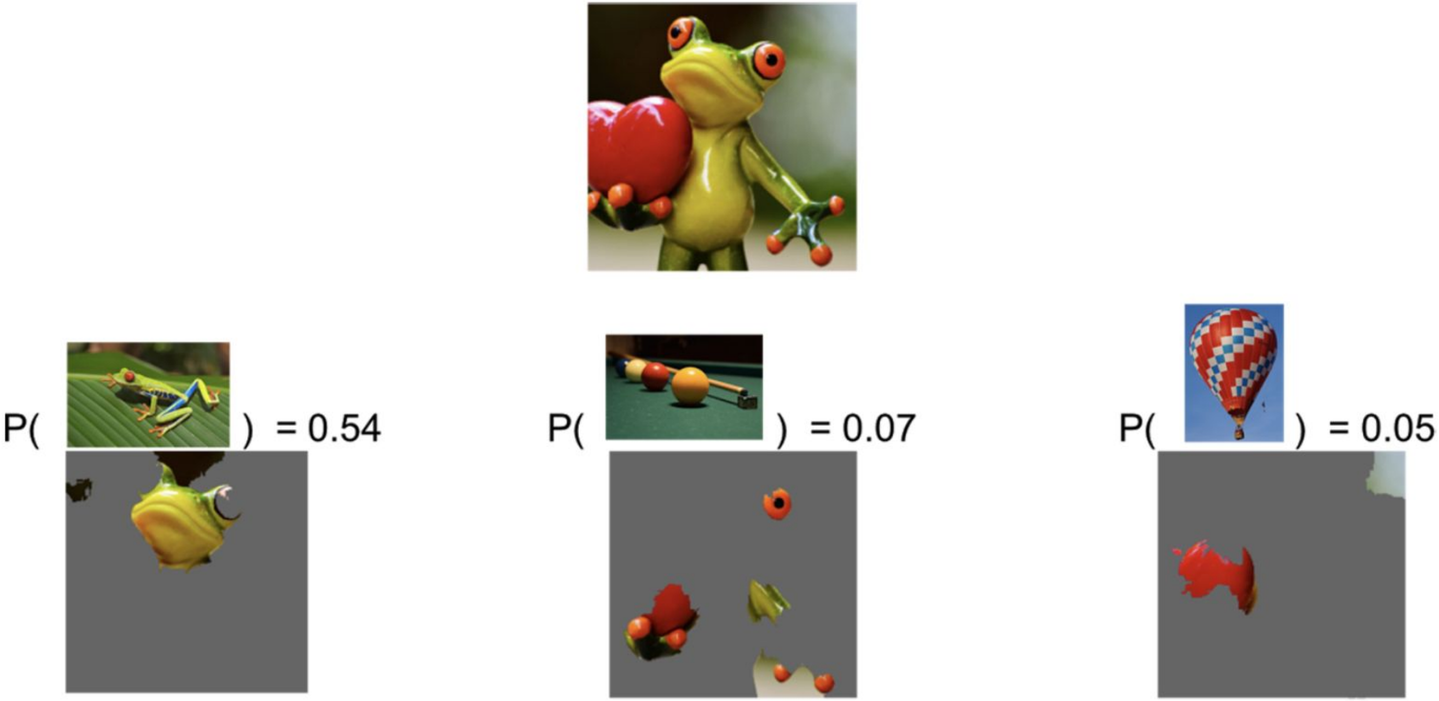
Original Image
P(tree frog) = 0.54

Perturbed Instances	P(tree frog)
	0.85
	0.00001
	0.52

Locally weighted regression

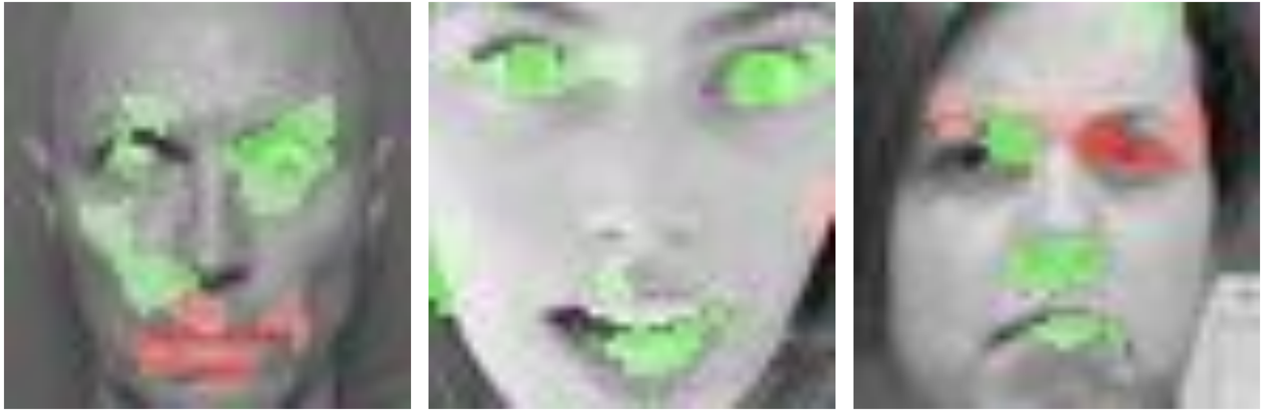
Explanation

Lime - 3/3



Lime

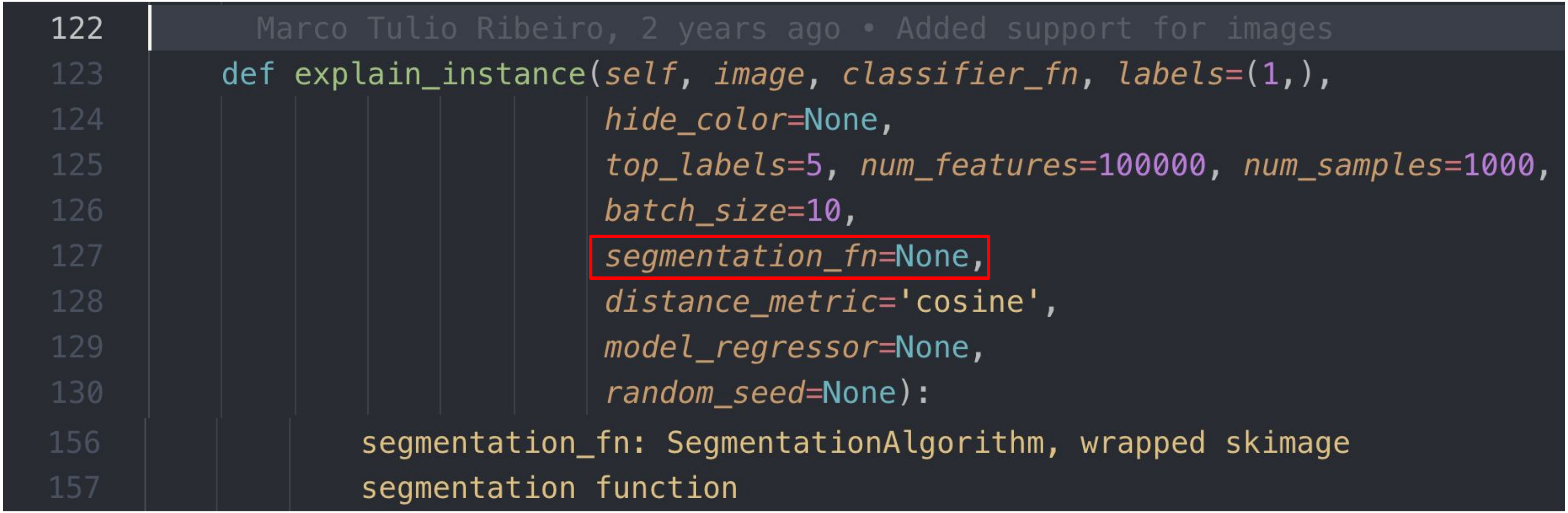
```
> pip install lime
```



Github Repo: https://github.com/marcotcr/lime

Ref: https://goo.gl/anaxvD

Lime - Hints



```
    Marco Tulio Ribeiro, 2 years ago • Added support for images
def explain_instance(self, image, classifier_fn, labels=(1,),
                     hide_color=None,
                     top_labels=5, num_features=100000, num_samples=1000,
                     batch_size=10,
                     segmentation_fn=None,
                     distance_metric='cosine',
                     model_regressor=None,
                     random_seed=None):
        segmentation_fn: SegmentationAlgorithm, wrapped skimage
        segmentation function
```

Requirements

1. 請使用CNN實作model，建議使用HW3 train好的model
2. 不能使用額外data
3. 請附上訓練好的model (及其參數)至github release或dropbox，並於hw4.sh中寫下載的command（請參照以下網站中方法：http://slides.com/sunprinces/deck-16#/2）
4. hw4.sh要在10分鐘內跑完

Assignment Regulation

- Only Python 3.6 available !!!!
- 開放使用套件
  - numpy >=1.14
  - pandas >= 0.24.1
  - python standard library
  - pytorch == 1.0.1
  - tensorflow == 1.12.0
  - keras == 2.2.4
  - matplotlib == 2.1.0
  - scikit-image == 0.14.2
  - lime == 0.1.1.33 (Install via pip)
- 若需使用其他套件，請儘早寄信至助教信箱詢問，並請闡明原因。

Deadline

- Github: 2019/04/12 23:59:59 (GMT+8)

助教會在deadline一到就clone所有程式，並且**不再重新clone任何檔案**

Policy

github上ML2019SPRING/hw4/裡面請至少包含 :

1. report.pdf
2. hw4.sh
3. your python files
4. model參數 (Make sure it can be downloaded by your script.)
   (*請將model download到與script相同的位置)

請不要上傳dataset, 請不要上傳dataset, 請不要上傳dataset

Policy

1. 以下的路徑，助教在跑的時候會另外指定，請保留可更改的彈性，不要寫死
2. Script usage: bash hw4.sh <training data> <output path/>

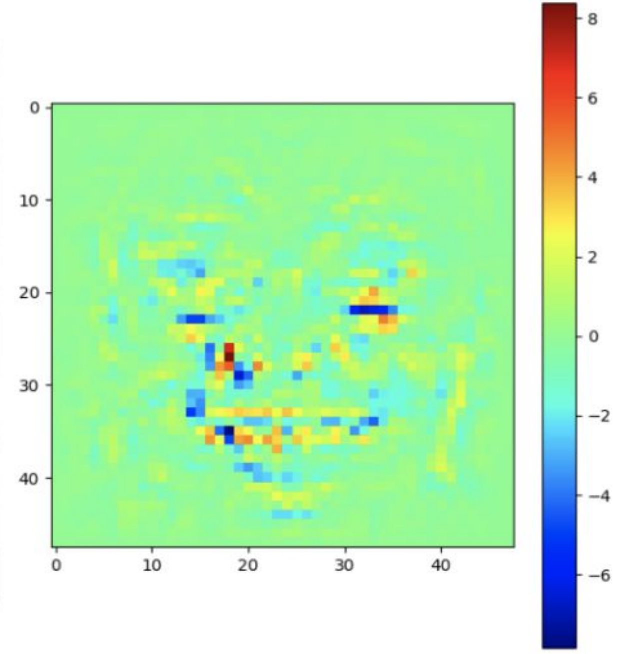
   training data: train.csv 的路徑

   After executing your script, there should be

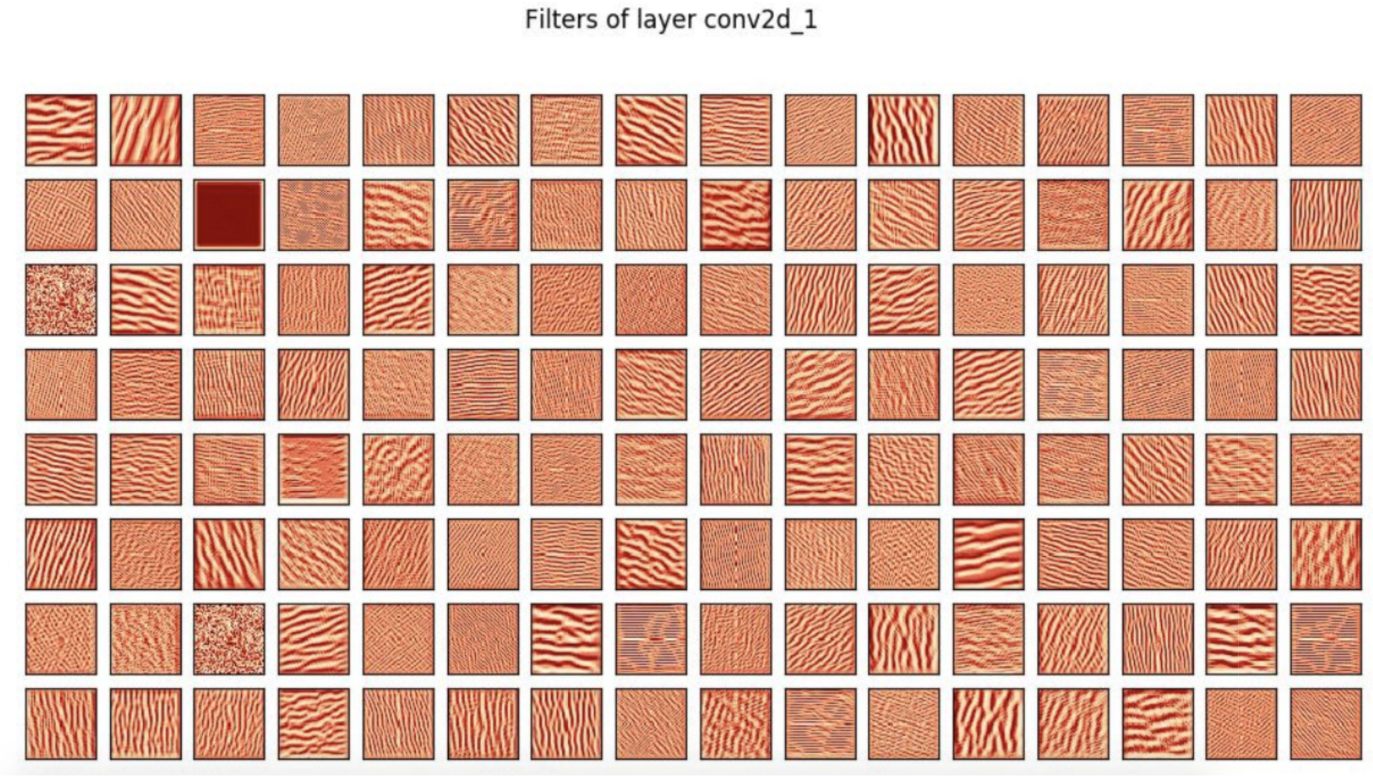
   - fig1_{0,1,2,3,4,5,6}.jpg (one image for each label class)
   - fig2_1.jpg
   - fig2_2.jpg
   - fig3_{0,1,2,3,4,5,6}.jpg (one image for each label class)

   under specified output path, these figures should be identical to the corresponding figures in your report.

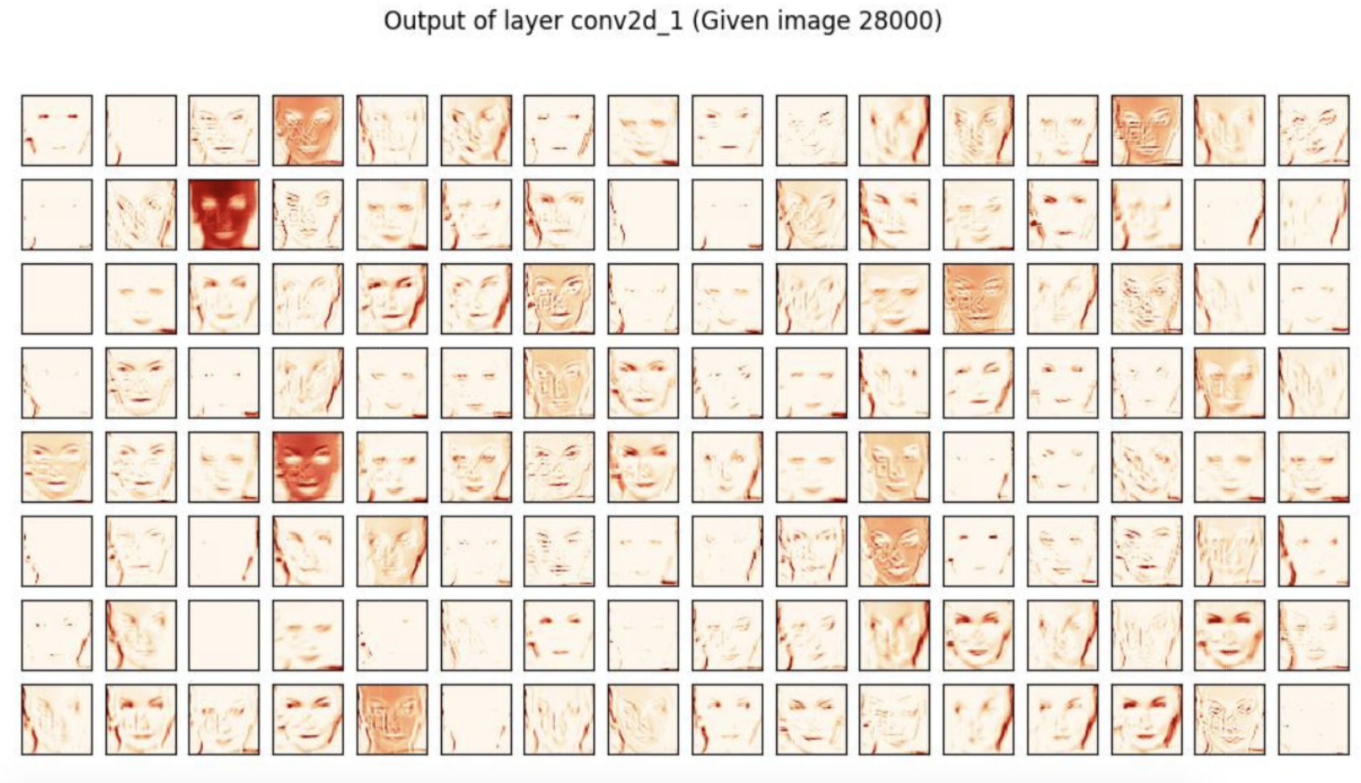
Sample Submission - fig1_3.jpg (高興)



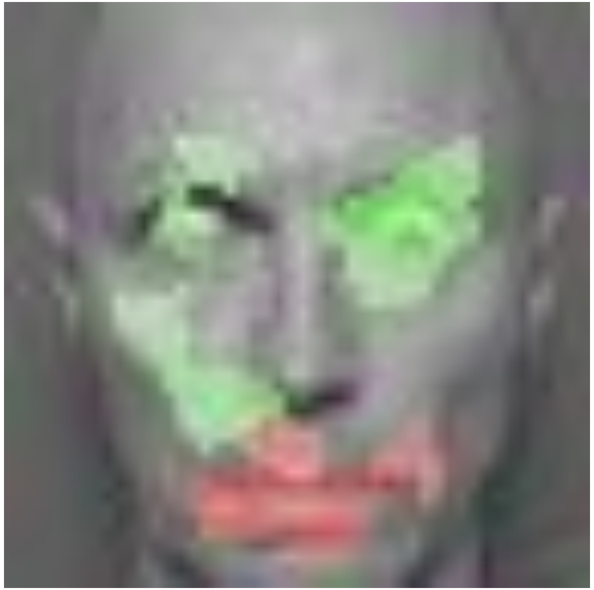
Sample Submission - fig2_1.jpg



Sample Submission - fig2_2.jpg



Sample Submission - fig3_5.jpg (驚訝)



Score - Other Policy

- 當script**格式錯誤**，造成助教無法順利執行，請在公告時間內寄信向助教說明，修好之後重新執行所得分數將x0.7。
- 可以更改的部分僅限syntax及io的部分，不得改程式邏輯或是演算法，至於其他部分由助教認定為主。
- reproduced出來的結果若與report上的圖片不相符，**該小題分數為零分**。
- Github遲交一天(*0.7)，不足一天以一天計算，不得遲交超過兩天，有特殊原因請找助教。
- Github遲交表單：https://goo.gl/forms/5ctgJjCkgYdDndOh1
  遲交請「**先上傳程式**」至Github再填表單，助教會根據表單填寫時間當作繳交時間。(**遲交才必需填寫**)

Score - report.pdf

- ➢ (2%) 從作業三可以發現，使用CNN 的確有些好處，試繪出其saliency maps，觀察模型在做classification 時，是 focus 在圖片的哪些部份？
- ➢ (3%) 承(1) 利用上課所提到的gradient ascent 方法，觀察特定層的filter最容易被哪種圖片activate 與觀察 filter 的 output。
- ➢ (3%) 請使用Lime套件分析你的模型對於各種表情的判斷方式，並解釋為何你的模型在某些label表現得特別好 (可以搭配作業三的Confusion Matrix)。
- ➢ (2%) [自由發揮] 請同學自行搜尋或參考上課曾提及的內容，實作任一種方式來觀察CNN模型的訓練，並說明你的實作方法及呈現visualization的結果。

Report template: https://goo.gl/JCijVR

Reminder

- Start early!!!
- Please do not hand in TA's sample images.
- Remember to fix the random seed in your code to insure that all the result images are reproducible. How to fix Lime's random seed
- The output results should be identical to the corresponding figures in your report.
- Please do not directly output saved images. Your script should run through the whole process.

FAQ

- 若有其他問題，請po在FB社團裡或寄信至助教信箱，**請勿直接私訊助教**。
- 助教信箱：ntumlta2019@gmail.com

Link

- 雲端使用方法:http://slides.com/sunprinces/deck-16#/2）
- 作業網址:https://ntumlta2019.github.io/ml-web-hw4/
- Report template: https://goo.gl/JCijVR
- Github遲交表單:https://goo.gl/forms/5ctgJjCkgYdDndOh1