Exercise-5 : Visualising how a deep CNN makes decsisions

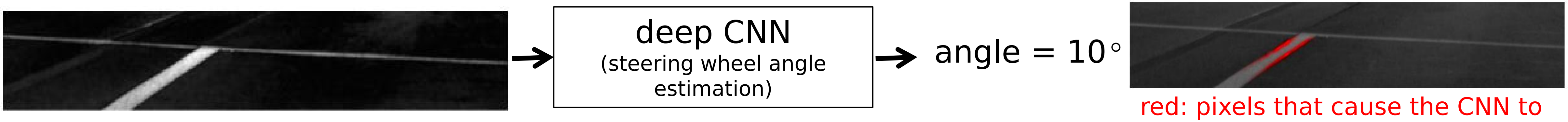
[Background]

Reference paper:

https://arxiv.org/abs/1611.05418

[1] "VisualBackProp: visualizing CNNs for autonomous driving" - Mariusz Bojarski(NVIDIA), Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Larry Jackel, Urs Muller, Karol Zieba, Arvix 2016

In this paper, the authors propose a method to determine which pixels in the image causes the final output of a deep CNN.



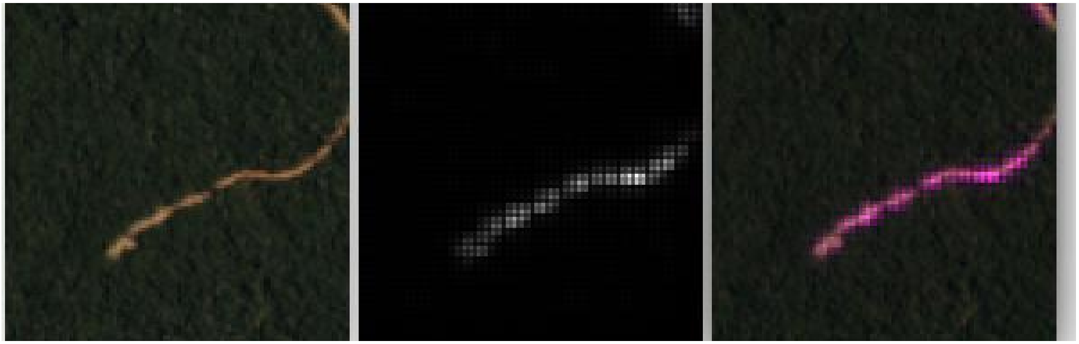
deep CNN
(steering wheel angle estimation)

angle = 10°

red: pixels that cause the CNN to make this decsision

We want to implement the visualisation method of the paper in pytorch and apply it to our satellite image classification problem. An example is:

input image classified as 'road'



pixels that cause this decsision

[tasks] Duration: 4 days

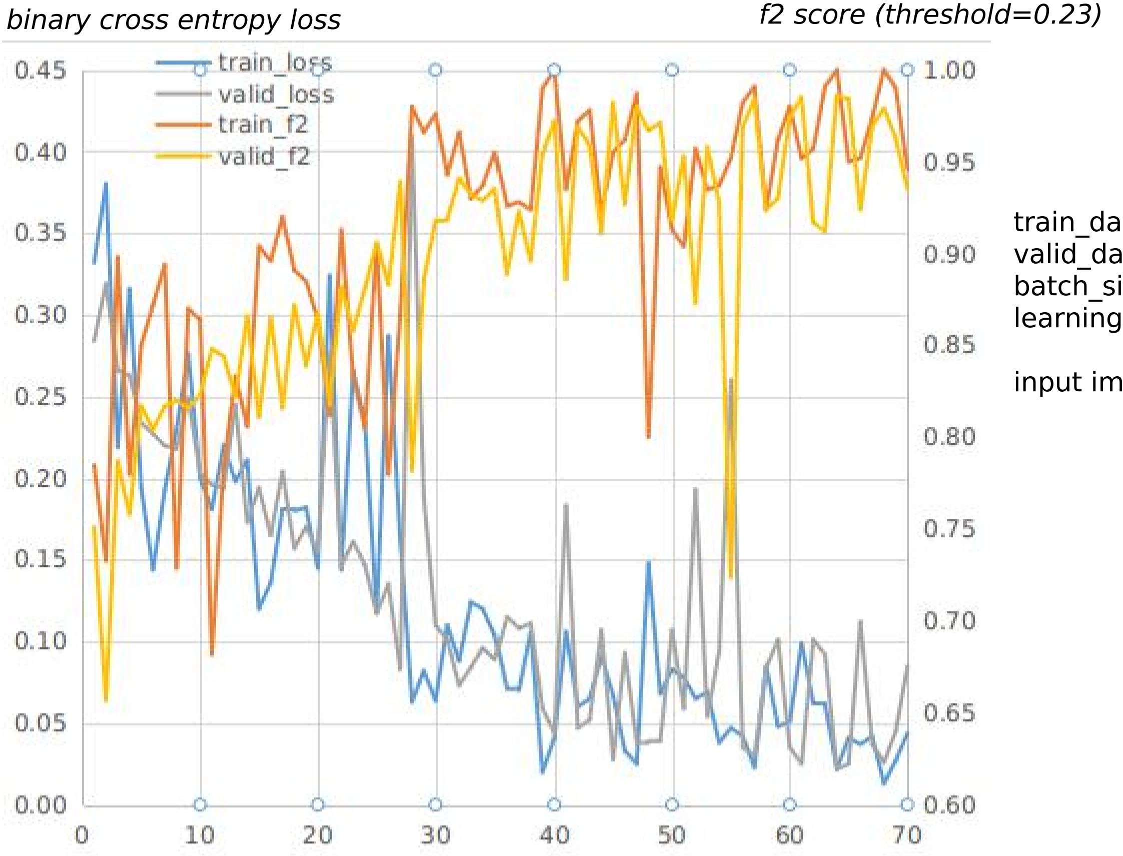
Step.1. Read the paper[1]. Make a presentation (e.g. PPT) to explain:

- the steps to compute the contribution score of each pixel to the final decision
- the mathematical reasoning for the above steps

[10 marks]

Step.2. Train a single label classifier. We use the 'road' class. Use the CNN below.

			parameters		
		feature maps	**kernel**	**strid**	**pad**
input		3x96x96			
block-0					
	conv2d	8x?x?	1x1	1	0
	batchnorm2d				
	relu				
block-1					
	conv2d	32x?x?	3x3	2	1
	batchnorm2d				
	relu				
block-2					
	conv2d	32x?x?	3x3	2	1
	batchnorm2d				
	relu				
block-3					
	conv2d	64x?x?	3x3	2	1
	batchnorm2d				
	relu				
global_maxpool		64			
block-8					
	linear	512			
	batchnorm1d				
	relu				
prob					
	linear	?			
	...				



train_dataset.num = 32384
valid_dataset.num = 8095
batch_size = 96
learning rate = 0.01

input image = 96x96

example results

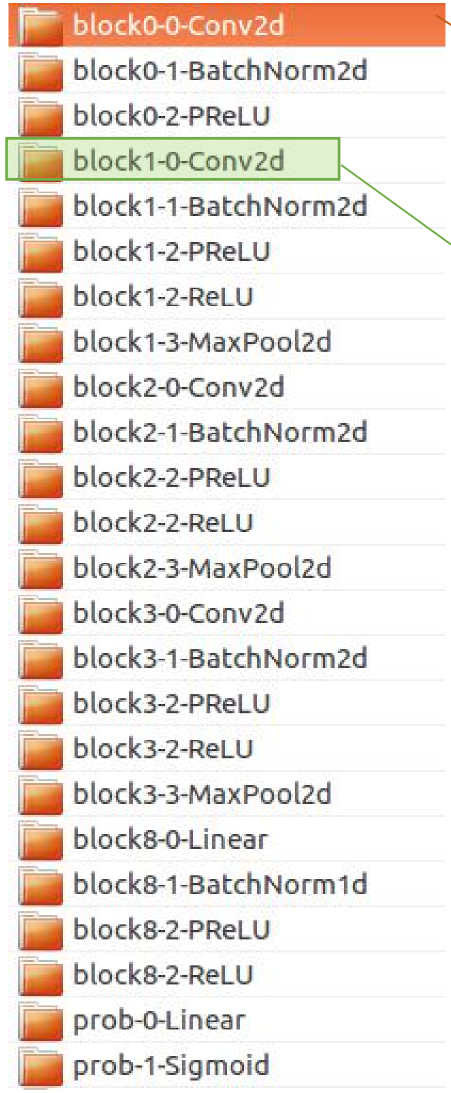
[10 marks]

Step.3. Write a function to save all feature maps (ouput of each layers) during a forward pass of a given image. *Hint: use 'register_forward_hook()'*

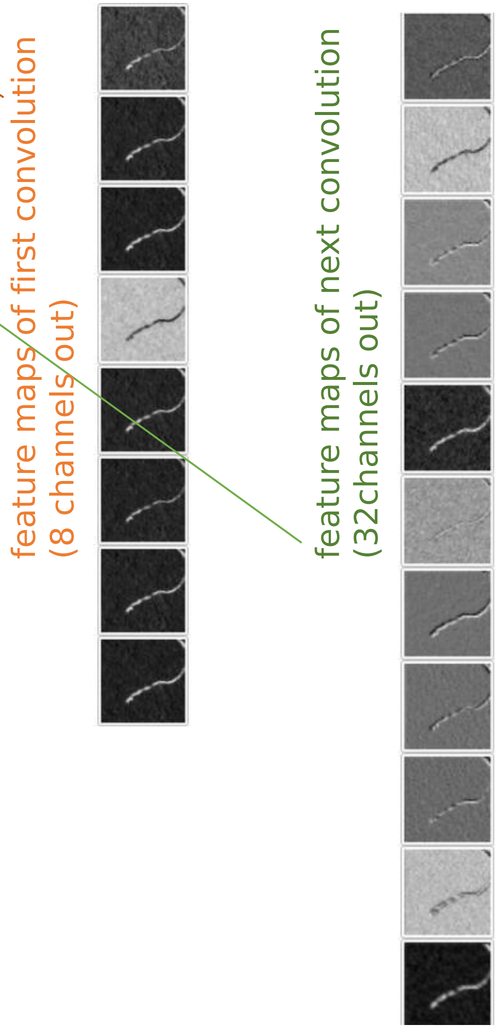
see: *http://pytorch.org/tutorials/beginner/former_torchies/nn_tutorial.html#forward-and-backward-function-hooks*
https://discuss.pytorch.org/t/how-to-extract-features-of-an-image-from-a-trained-model/119

example results

input



block0-0-Conv2d
block0-1-BatchNorm2d
block0-2-PReLU
block1-0-Conv2d
block1-1-BatchNorm2d
block1-2-PReLU
block1-2-ReLU
block1-3-MaxPool2d
block2-0-Conv2d
block2-1-BatchNorm2d
block2-2-PReLU
block2-2-ReLU
block2-3-MaxPool2d
block3-0-Conv2d
block3-1-BatchNorm2d
block3-2-PReLU
block3-2-ReLU
block3-3-MaxPool2d
block8-0-Linear
block8-1-BatchNorm1d
block8-2-PReLU
block8-2-ReLU
prob-0-Linear
prob-1-Sigmoid



feature maps of first convolution (8 chanels out)

feature maps of next convolution (32channels out)

saved feature maps

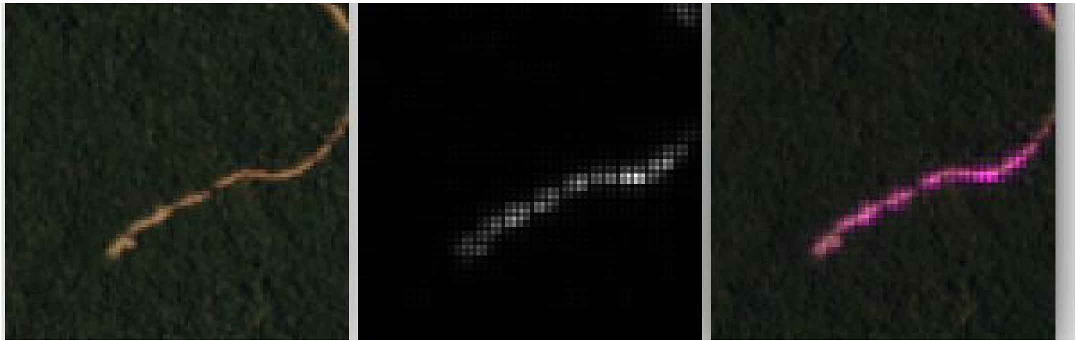
[10 marks]

Step.4. Write a function compute and visualise the contribution score of each pixel

see: https://github.com/mbojarski/VisualBackProp

example results

input image
classified as
'road'



pixels that cause
this decsision

[20 marks]

Question: Explain why is there blocky artifacts in the visualisation

[10 marks]

More results (on validation set):

image (true label)
estimated probability