POLITECNICO
DI TORINO

Corsi di Laurea Magistrale in Ingegneria Biomedica ed Ingegneria Informatica
Bioinformatics

S.N.A.C Software – Sistema Neurale di Analisi dei Carcinomi

CNNs project - Whole slides attention maps

Candidates:
Greta Berardengo, s244278
Silvia Fornara, s244216
Elisa Wan, s241854
Andrea Grippi, s236305

A.A. 2017/2018

Index

1. Introduction

The aim of this project is the creation of a software for cancer detection by using attention maps. The main goal of this software is to help and to simplify the doctors and pathologists work: these last ones have to make diagnosis based on the visual examination and search for abnormal areas in samples, taken from a possible pathogenic region of the human body. The attention map created by our software allows instead to simplify the pathologist research using multi-resolution images of the samples: the software task is the elaboration of the images and the highlight of the pathological areas, if present.
Therefore the software is able to recognize between healthy regions and pathologic ones and to mark them in different shades to allow an immediate visual identification of the pathogen region on images.
Moreover the software can make a more precise diagnosis of the disease because it is able to make an additional discrimination between adenoma, a precursory lesion which may turn into cancer, and adenocarcinoma, the already evolved cancer.

We used whole slide images (WSIs) for creating an attention map for detecting cancer. A WSI is a digitized high-resolution version of a histopathology glass slide that can be manipulated through software to mimic microscope review and diagnosis [1].

Our WSI images are in SVS format [2], a multi-layer format used by a number of medical/microscope scanner. SVS images have different layers, each one representing the slide ad different resolution.
Our software takes SVS images, crops them and converts them into a .jpg format that will be then fed to the neural network for prediction. After the prediction is done, the full image gets recombined and is ready to be further analyzed. Areas with high probability of cancer are strongly highlighted, while areas with low cancer probability are highlighted as well but with lighter shades.

2. Strategy

The chosen approach was to adapt state of the art image classification techniques, like for example Convolutional Neural Networks, to our problem.
This study is composed of the following main steps:

- Crop WSI into smaller images with reasonable dimensions to be fed into CNNs
- Remove useless crops
- Train different CNNs on 2 classes: Healthy (H) and not healthy (non-H)
- Train different CNNs on 3 classes: Healthy (H), Adenocarcinoma (AD) and Adenoma (AD)
- Create an attention map on test set images
- Analyze the results

SNAC has been developed using Python3 together with:

- Keras, a library based on TensorFlow for image classification and prediction using Neural Networks
- OpenCV for image manipulation
- OpenSlide for dealing with WSI images

3. Theory about CNNs

Neural Networks are mathematical models that represents the interconnection between elements called artificial neurons. They are built as multiple layers of neurons densely connected to each other, through a parameter called 'weight'. The information travels from layer to layer during the elaboration; the output gets compared to the input and based on the difference between the two (error), the network uses a backpropagation algorithm to modify the weights of the neurons in the different layers. This process is called training of the neural network and is the process of teaching the network how to recognize specific inputs. Usually the training is done using a specific set of input parameters called 'training set', while the evaluation of the net's performance is done on a different group of parameters called 'test set'.
When we have a large number of parameters involved, it's impossible to train the system without overfitting the model; in this case the network will recognize very well the elements of the training set, but the performance will be lacking regarding the test set.
We can solve this problem using Convolution Neural Networks (CNN), in which we train the network from scratch, using a large dataset, like for example ImageNet. This kind of NN shares the parameters between the neurons and sparse connections in convolutional layers.
The difference between NN and CNN are the following:

- NNs use a structure with 3 kinds of layer: input, hidden and output
- CNNs use a 4 layers structure:
    - the convolution layer, that uses a filter matrix over the array of image pixels and performs convolution operation to obtain a convolved feature map
    - the ReLU layer, that introduces non-linearity to the network, and whose output is a rectified feature map
    - the pooling layer, that reduces the dimension of the feature map, and gives a pooled feature map as output
    - the fully connected layer, that classify the images

Most CNN have huge memory and computation requirements, especially while training.
Two of the most used CNN for image classification are VGG16 and Inception ResNet.

3.1 Inception-resnet-v2

Inception-resnet-v2 is a neural network released in 2016 by Google as an improvement of their previous work Inception-V3.
Tested using the ILSVRC image classification benchmark [3], it obtained the following results:

- 80.4% Top-1 Accuracy
- 95.3% Top-5 Accuracy

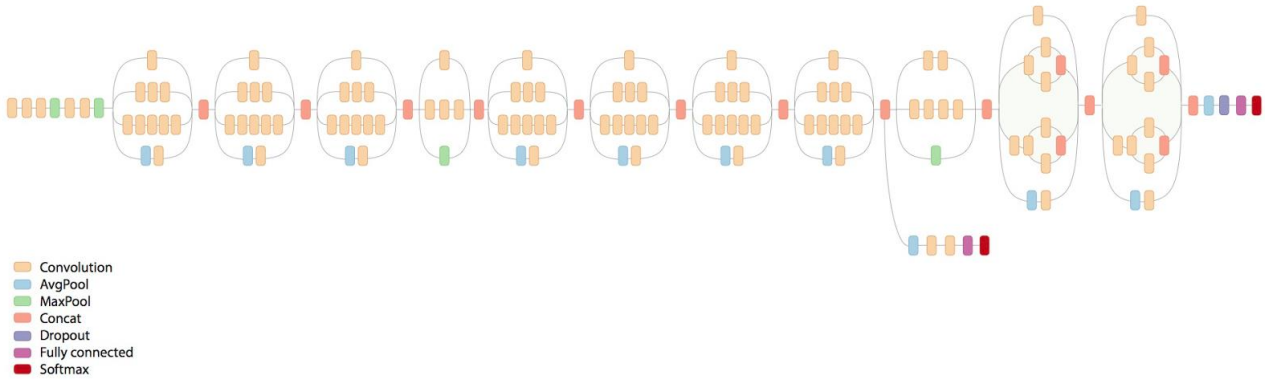

Figure 1: structure of Inception net [4]

The Inception nets builds on the idea that most of the activations in a deep network are either unnecessary (value of zero) or redundant because of correlations between them, therefore the most efficient architecture of a deep network will have a sparse connection between the activations.
The Inception architecture was also designed to perform well even under strict constraints on memory and computational budget. This has made it feasible to utilize Inception networks in big-data scenarios.
The problem with Inception architecture it's that its complexity makes it more difficult to adapt to new use-cases while maintaining its efficiency.
The inception module uses 1X1 convolutions before applying larger sized kernels to reduce the dimension of the input channels, before feeding into those convolutions. Doing so, in the first inception module, the input to the module is first fed into 1X1 convolutions, with just 16 filters before it is fed into 5X5 convolutions. Then we have a global average pooling, which averages out the channel values across the 2D feature map, after the last convolutional layer. This drastically reduces the total number of parameters and the computations, making it much faster than VGG, even though is much deeper.
Increasing the depth should increase the accuracy of the network; but the problem with increased depth is that is required to change the weights, since prediction

becomes very small at the earlier layers. In conclusion adding layers leads to higher training error.

Residual networks allow training of such deep networks by constructing the network through modules called residual models. It achieves better accuracy than while being computationally more efficient.
The architecture is similar to the VGGNet consisting mostly of 3X3 filters.
Inception-ResNet-v2 is similar to an Inception-v4 network. However, the step time of Inception-v4 proved to be significantly slower in practice, probably because of the larger number of layers.
While the Inception-v4 is a pure Inception variant, the ResNet uses residual connections, that add the output of the convolution operation of the inception module, to the input. This leads to dramatically improved training speed for the Inception architecture.
Compared to VGG networks, Inception-ResNet models were able to achieve higher accuracies at a lower epoch, thanks to the training with residual connections.

3.2 VGG-16

Tested using the ILSVRC image classification benchmark [3], it obtained the following results:

- 91.2% Top-5 Accuracy

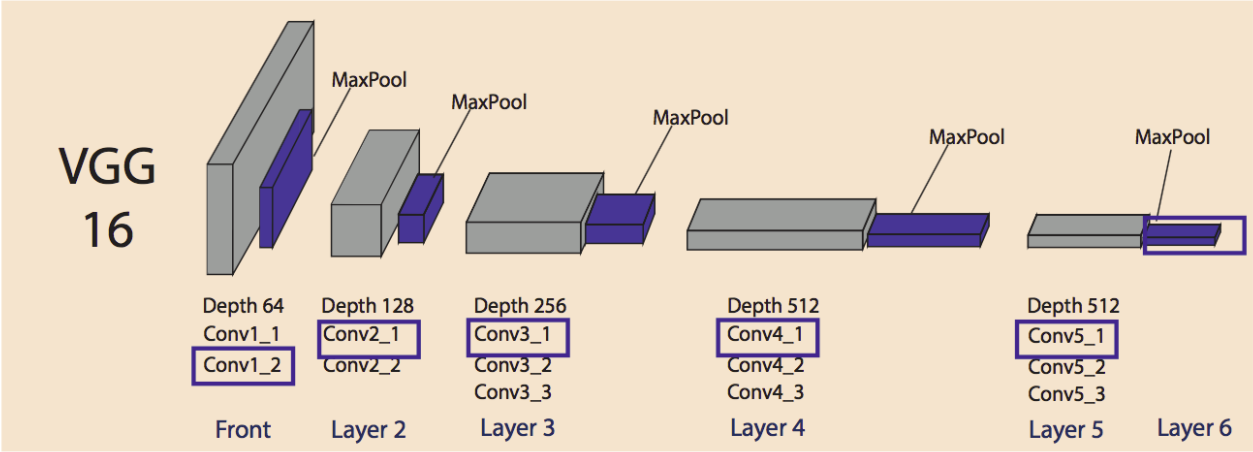

Figure2: structure of VGG-16 net

The VGG network architecture was introduced by Simonyan and Zisserman in their 2014 paper, Very Deep Convolutional Networks for Large Scale Image Recognition [5]. This network is characterized by its simplicity, replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3X3 convolutional layers stacked on top of each other in increasing depth.
Two fully-connected layers, each with 4,096 nodes are then followed by a softmax classifier.
It achieves a phenomenal accuracy on ImageNet dataset, with a 92.7% top-5 test accuracy in a dataset of over 14 million images belonging to 1000 classes, but there are two major drawbacks:

- it is very slow to train
- the network architecture weights themselves are quite large (in terms of disk/bandwidth).

The VGG16 in particular has 16 convolutional layers, only 3x3 convolutions, but lots of filters. It is the most preferred choice in the community for extracting features from images.

3.3 Basic-NN

The network that we called “Basic-NN” is a standard neural network that can be used for basic image recognition.
It’s made using standard keras libraries.
We wanted to use a basic network in order to compare its performances to the ones of state of the art neural networks such as Inception Resnet and VGG-16.

We show below the structure of the network:

Layer (type)	Output Shape	Param #
conv2d_7 (Conv2D)	(None, 224, 224, 32)	896
conv2d_8 (Conv2D)	(None, 222, 222, 32)	9248
max_pooling2d_4 (MaxPooling2	(None, 111, 111, 32)	0
dropout_5 (Dropout)	(None, 111, 111, 32)	0
conv2d_9 (Conv2D)	(None, 111, 111, 64)	18496
conv2d_10 (Conv2D)	(None, 109, 109, 64)	36928
max_pooling2d_5 (MaxPooling2	(None, 54, 54, 64)	0
dropout_6 (Dropout)	(None, 54, 54, 64)	0
conv2d_11 (Conv2D)	(None, 54, 54, 64)	36928
conv2d_12 (Conv2D)	(None, 52, 52, 64)	36928
max_pooling2d_6 (MaxPooling2	(None, 26, 26, 64)	0
dropout_7 (Dropout)	(None, 26, 26, 64)	0
flatten_2 (Flatten)	(None, 43264)	0
dense_3 (Dense)	(None, 512)	22151680
dropout_8 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 2)	1026

Total params: 22,292,130
Trainable params: 22,292,130
Non-trainable params: 0

4. Dataset

Our initial dataset is composed by 109 SVS images of 27 different patients.
In literature, neural networks for image classification are trained on 224x224 pixel images, instead of the SVS default format of 240x240 pixel. Except for our custom built neural network, we will use state of the art models that were built with this standard format in mind so we will keep this parameter unchanged.

As we already said our input WSIs have multiple resolutions, as shown below:

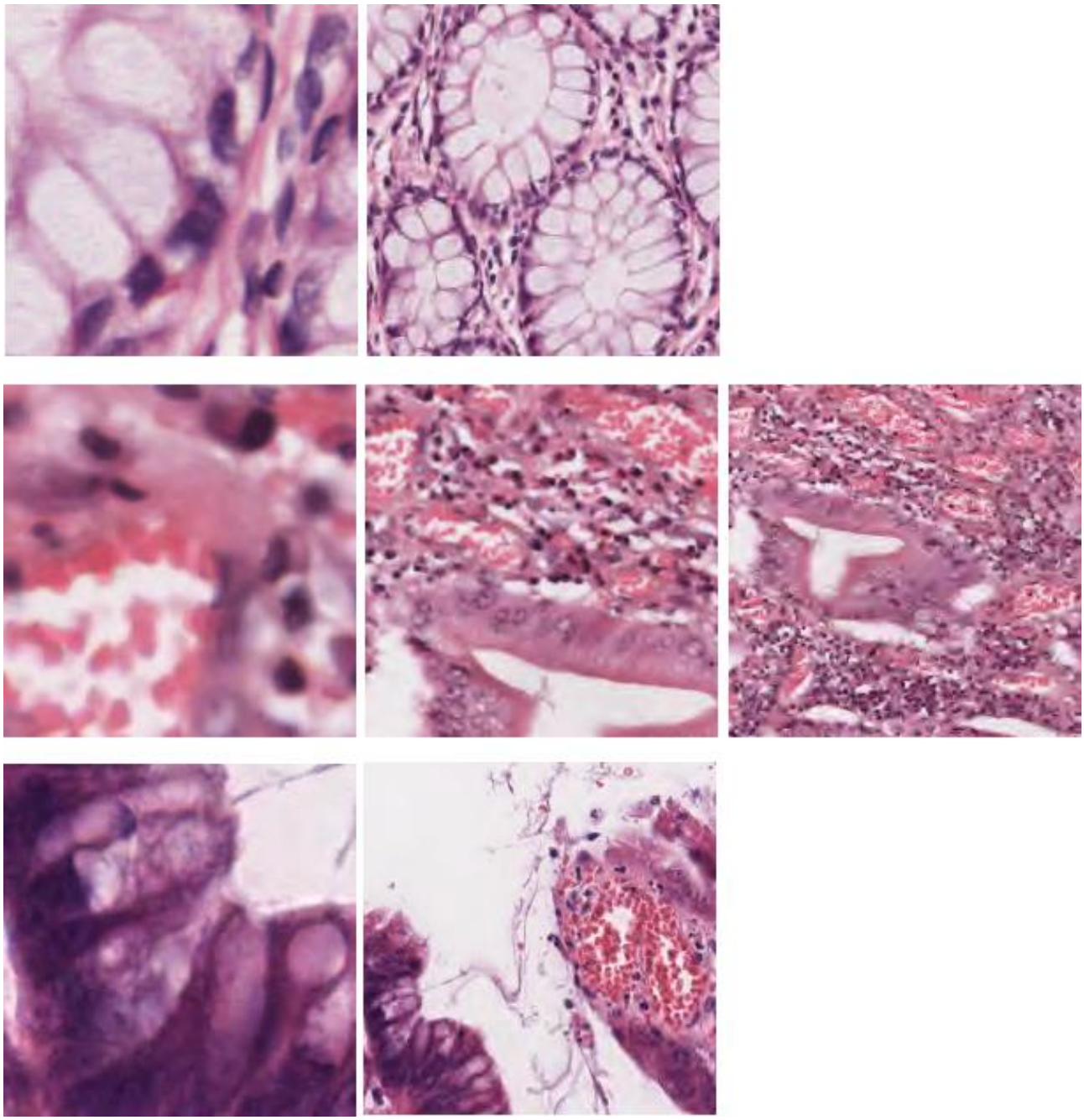

Figure 3: image crops with different resolution

At first level, images are not always in focus and not all images have a third level.
We decided to work only on the second level, as it gave us the best tradeoff in terms of quality and performances.
While cropping the images, we decided to overlap them by 50 pixels which is half of the average size of a healthy cell, in order not to miss important information between cells, and also because we wanted to build an attention map as smooth as possible.
We worked on histological samples from different patients for training and validation purposes.

4.1 Training set

The crops with too many white parts, like the ones in fig.[4] are removed automatically from the training set, because not useful for the purposes of the training.

Figure4: example of not useful crops

Also crops containing only stoma, like fig.[5] are removed because not useful, but with a manual inspection work:

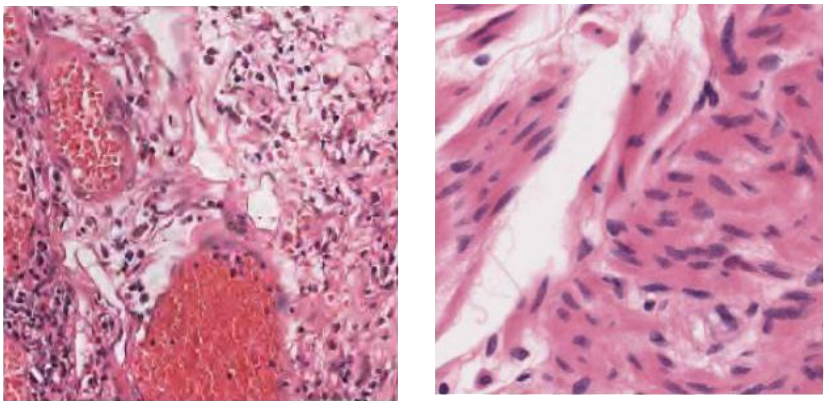

Figure 5: example of stroma crops

From the remaining crops, some have been selected for the validation set and some have been removed to balance out the number of crops for each class.

4.2 Validation set

We selected three patients (21, 38, 48) to create a validation set of 1073 crops with the following structure:

- 371 Healthy
- 368 Adenocarcinoma
- 334 Adenoma

which is about 15% of the total.

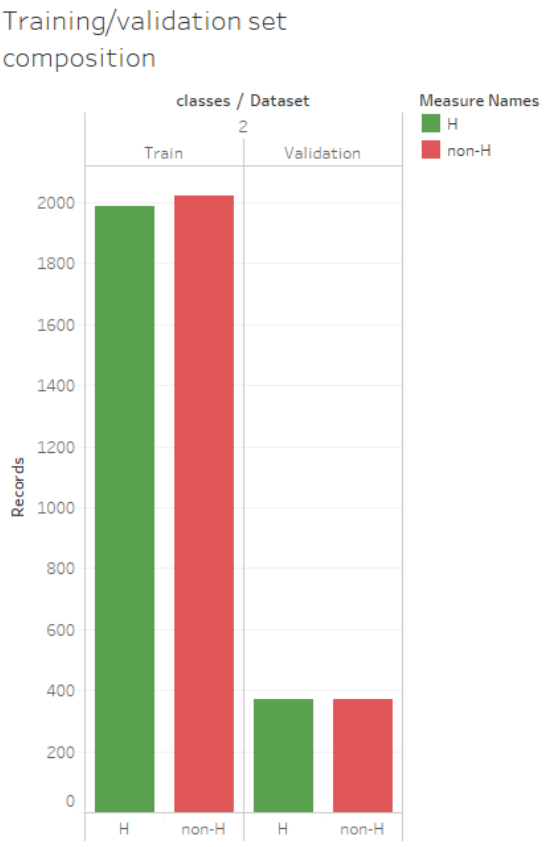

Figura 2:composition of training and validation set with two classes.

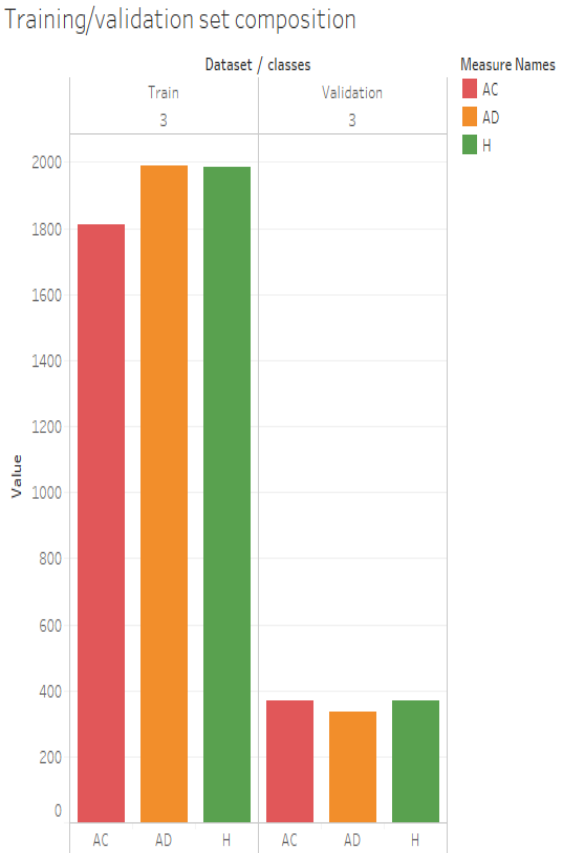

Figura 2: composition of training and validation set with three classes

5. Prediction

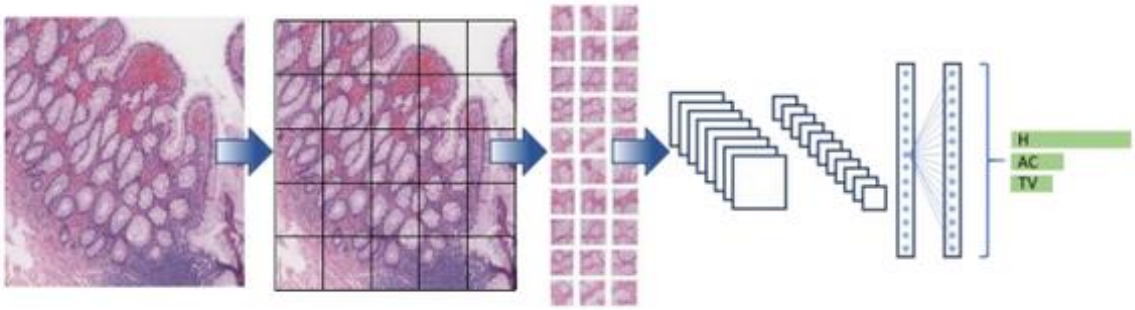

Figure 3: Steps performed by the software

The CNN's implemented in this project have been trained at first to predict the crops class between healthy or non-healthy: the healthy tissues are detected because of the regular and well defined cell profile while the non healthy ones are characterized by cells with abnormal and too elongated profile. In a second time networks have been trained to make a further classification in non healthy tissues between adenoma, a precursory lesion which may turn into cancer, and adenocarcinoma, the already evolved cancer.

classes	epoches	NN	train_acc	val_acc
2	10	basic-NN	0,7197	0,7281
		inception	0,9968	0,9870
		vgg	0,7207	0,8451
	20	inception	0,9982	0,9762
3	10	basic-NN	0,9561	0,9326
		inception	0,9927	0,9631
		vgg	0,7440	0,8227

Figura 4:Accuracy of models trained on the unbalanced dataset.

classes	NN	epochs	days	train_acc	val_acc
2	basic	10	1	0,504	0,654
		20	2	0,504	0,654
		100	8	7,949	0,654
	inception	10	2	0,993	0,982
		20	6	0,988	0,794
	vgg	10	4	0,843	0,467
		20	timeout		
3	basic	10	1	0,974	0,686
		20	2	0,979	0,745
	inception	10	5	0,994	0,750
		20	timeout		
	vgg	10	6	0,841	0,347
		20	timeout		

Figura 5:Accuracy of models trained on the balanced dataset.

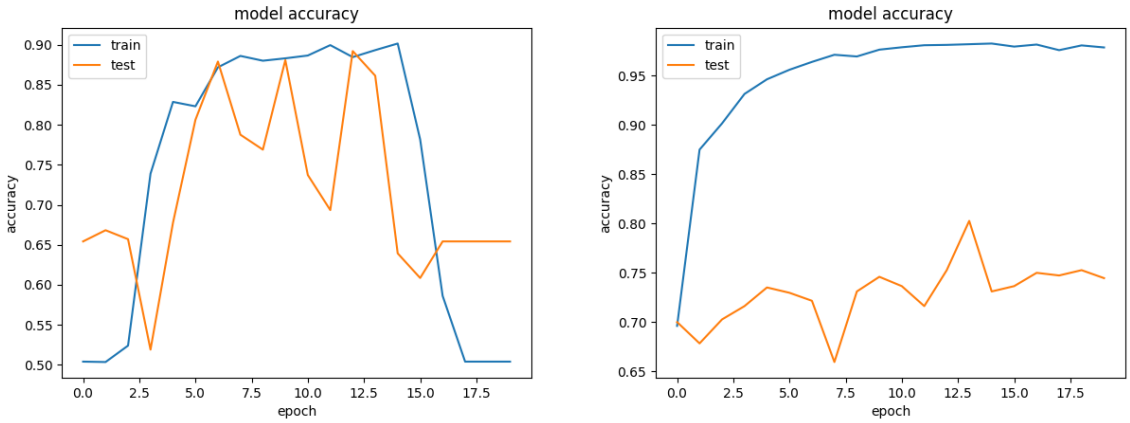

Figura 8: Training accuracy of basic-NN for 20 epochs. Training on 2 classes on le left and on 3 classes on the right.

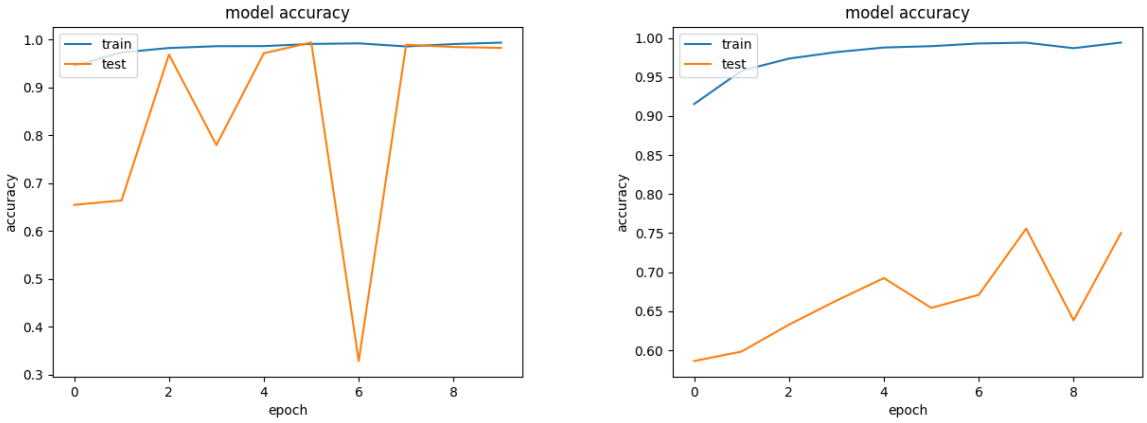

Figura 7: Training accuracy of inceptionResNetV2 for 10 epochs. Training on 2 classes on the left and on 3 classes on the right.

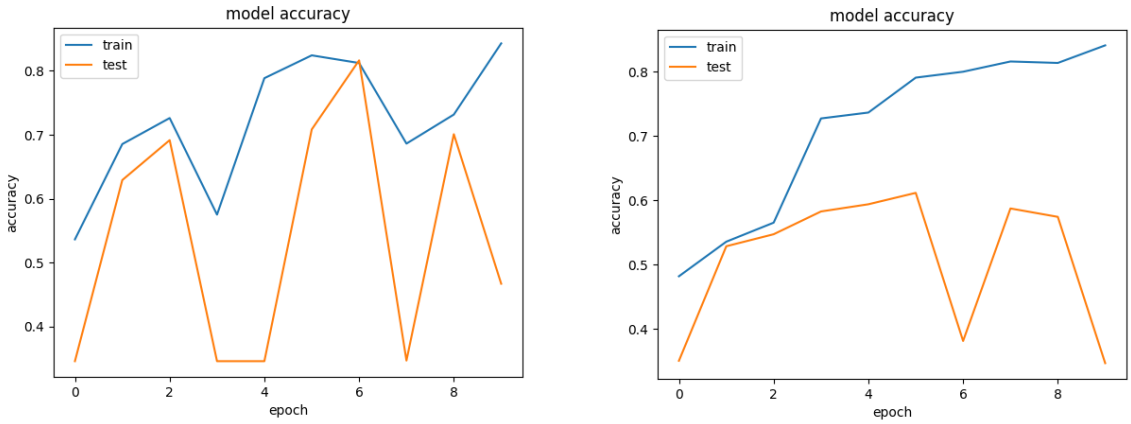

Figura 6: Training accuracy of vgg-16 for 10 epochs. Training on 2 classes on the left and on 3 classes on the right.

At training time, a VGG model accuracy increases with highly irregular pace, while an Inception model learns at constant velocity. Surprisingly, the inception model accuracy at the first epoch is extremely high, between 0.9 and 0.95, while a VGG model has a different strategy, it starts at a low value and then, rapidly increases.
The Basic NN model presents a drastical drop in accuracy at the end the the training phase on 2 classes, while it is able to improve its accuracy and reach higher values in

the 3 classes case. This behavior could be due to a high variability inside the non healthy class.
At testing time on 2 classes, the accuracy floats between very low levels and very high ones, we justify this beahavior again with the high variability inside the non healthy class. When dealing with 3 classes the validation accuracy is never able to reach acceptable values.
In general, validation accuracy is much lower that training accuracy, we investigated and found out that the helthy cells of a patient are very different from the healthy cells of another one, the same apply for carcinoma cells and adenocarcinoma cells. The networks has not been able to find a general rule, but is overfitting the training set. To overcome this problem, we should try with a larger number of patients, and balance the number of crops for each patient.

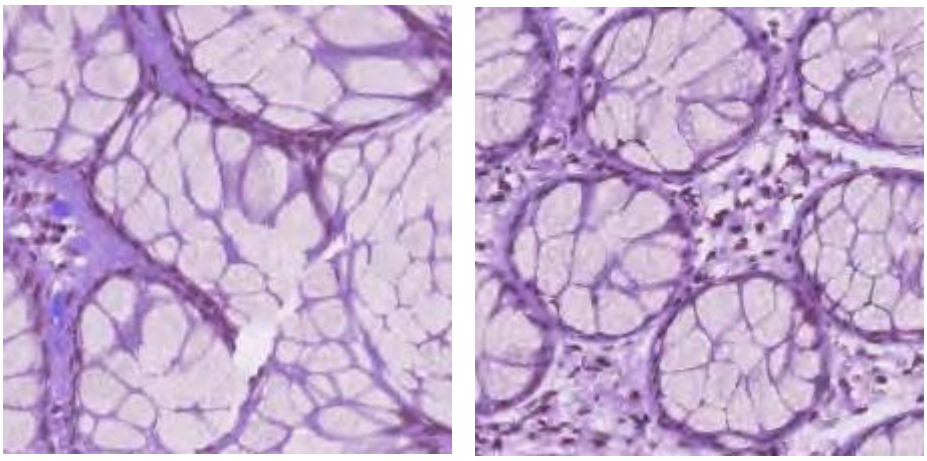

Figura 9: Healthy cells of patient n.18

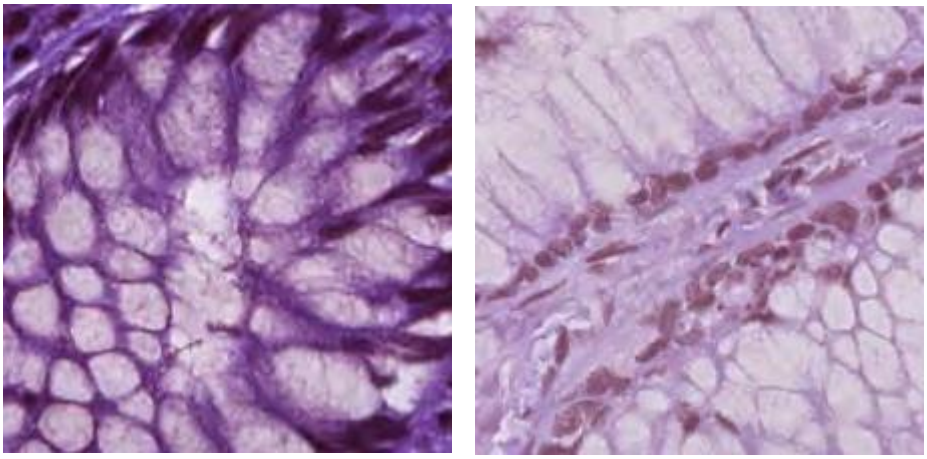

Figura 10: Healthy cells of patient n.20 on the left and patient n.48 on the right.

6. Attention map

The creation of the attention map is the ultimate goal of this project. All of the Neural Networks we used, after having classified crop by crop in the two or three different classes, recreate the whole image and, based on the previous classification of the single crops, highlight the zones that have been recognized to be pathologic, with different intensity based on the probability of cancer (AD or AC).

6.1 Test images

These are the original images used for testing our models:

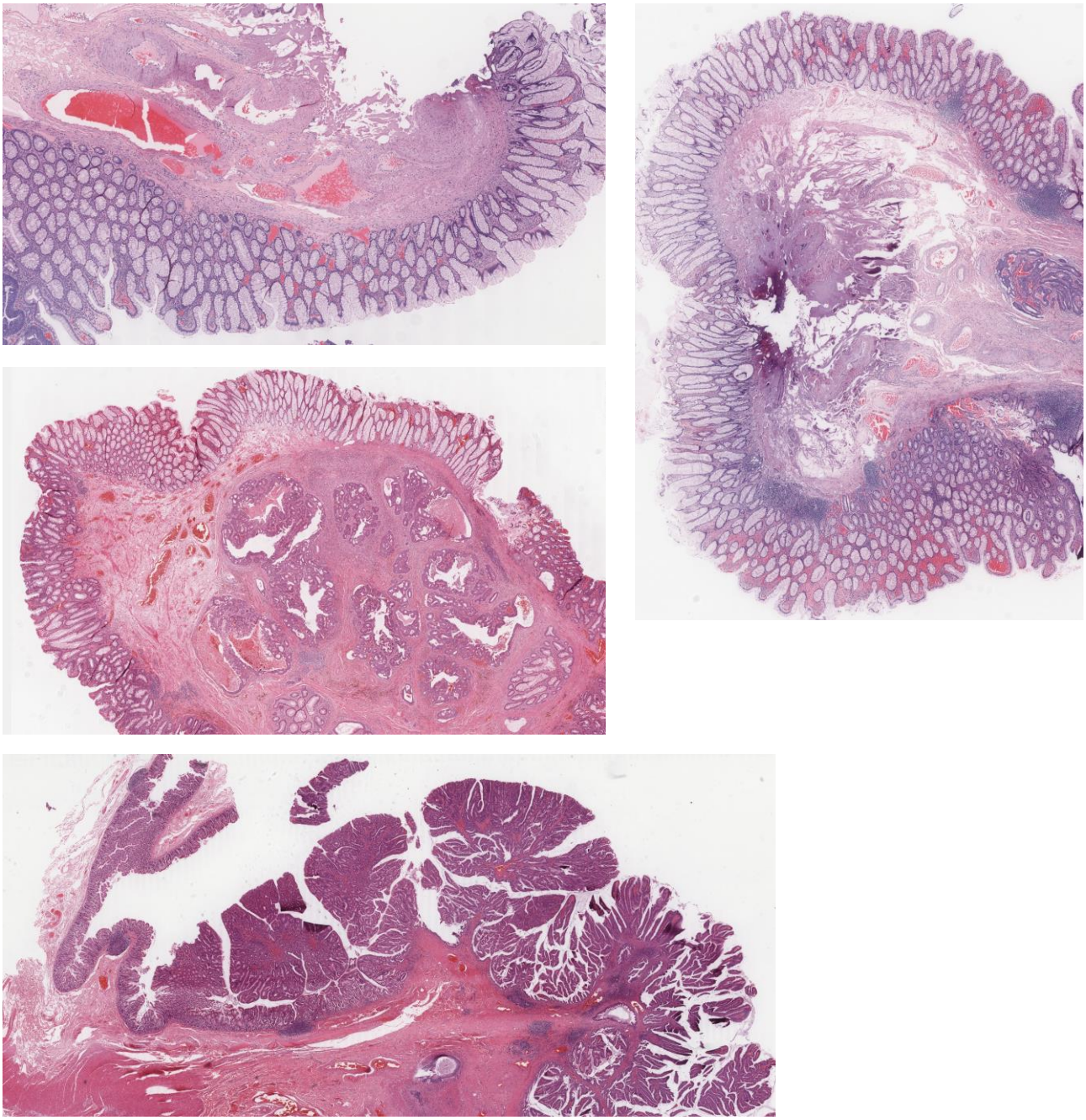

Figure 1: Test set images

6.2 Results from 2-class training

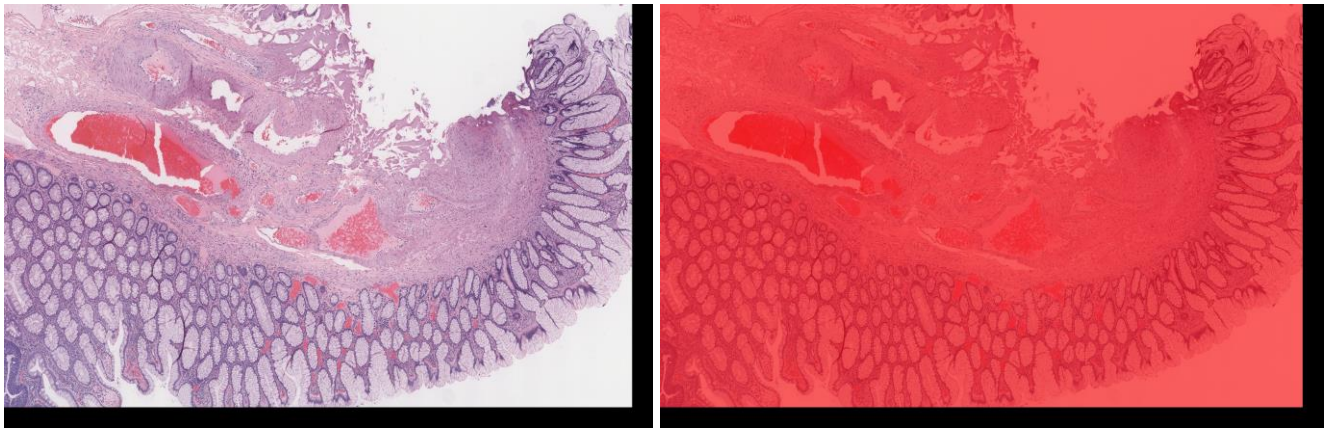

Figura 11: Target class H on the left and non-H on the right.

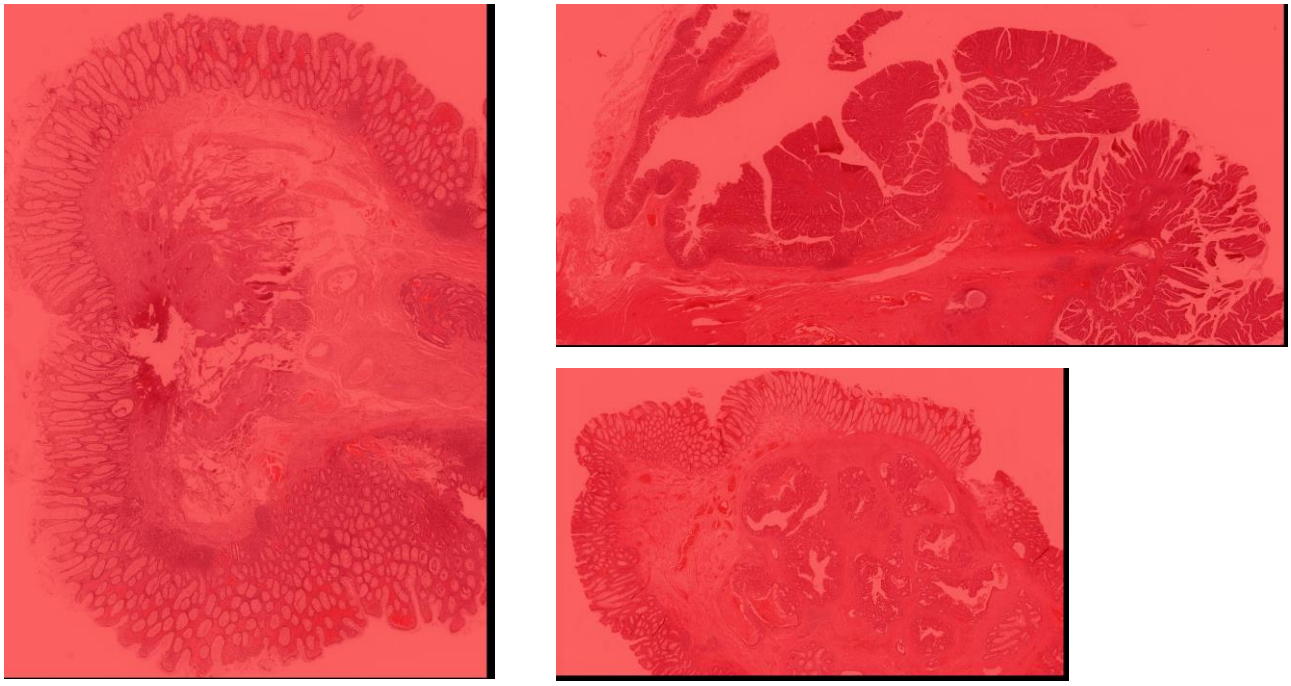

Figure 2: All test images are predicted as non-H

All three models have the same output even after we balanced the dataset to have the roughly the same number of crops for both classes.
The validation accuracy of Basic-NN and Vgg-16 is around 0.5 so we could have predicted this behavior, but the InceptionResNet's validation accuracy is 0.9. Why it isn't preforming better then the other two is still an open question.

6.3 Results from 3-class training

Basic-NN predicts all crops of all test images as belonging to class “AD” while InceptionResNetV2 predicts everything as class “AC”. Only Vgg-16 shows some variability:

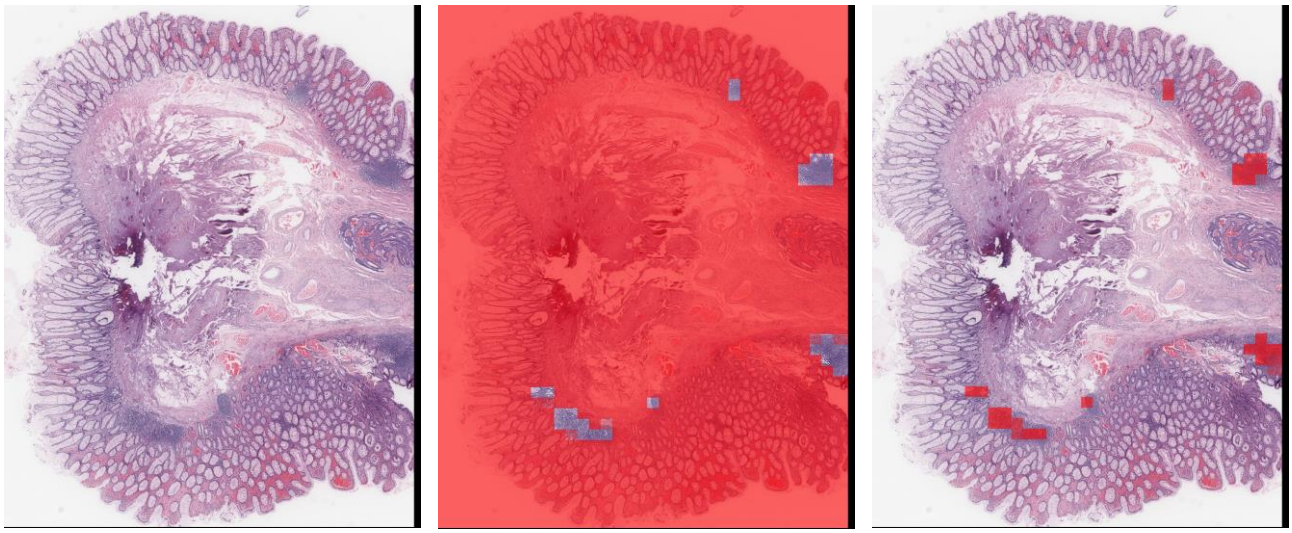

Figura 12:Attention map from vgg-16 prediction. Target class AC on the left, AD in the middle and H on the right.

7. Encountered Difficulties

Class imbalance
After most of the models have been trained we found out that, due to class imbalance, predictions had big bias toward one class. To overcome this problem, we had to re-train every model on a balanced training set.

Our trained models are not able to distinguish among H, non-H, AC and AD. This could be due to different reasons:

- High difference between patients
- Low number of epochs
- Limited time for training

8. Computational aspects

While developing SNAC we tried to keep time and space complexity in mind.
The first version involved saving not only the whole input and output (colored) images but also all the crops, before and after the classification. This process had a high memory usage and computational cost. To solve the problem, we implemented a second version of SNAC that doesn't save all crops and their colored version, in order to significantly reduce space complexity.

9. Future improvements

9.1 Parameters tuning

One aspect that could be improved from our work is definitely parameters tuning: because of time restrictions and being the training on these images extremely computationally heavy, we had to use a low number of epochs to train the network, in order to be able to gather some results in a couple of weeks.
It would be good to try to increase the number of epochs to a "standard" value, e.g. 100-200.

9.2 Image resolution level

It would be also interesting to study the differences in performances between networks trained on the different level of the SVS images and see if there can be an improvement in the accuracy of the predictions.

10. References

1. *https://digitalpathologyassociation.org/glossary-of-terms_1*
2. *http://paulbourke.net/dataformats/svs/*
3. *ILSVRC image classification benchmark*
4. *Improving Inception and Image Classification in TensorFlow*
5. *Very Deep Convolutional Networks for Large Scale Image Recognition*