Intel & MobileODT Cervical Cancer Screening: Training From Scratch vs. Transfer Learning Approaches For Deeper Architectures

Federico Vasile
Politecnico di Torino
s267175@studenti.polito.it

Mattia Tarquinio
Politecnico di Torino
s265802@studenti.polito.it

Abstract

This work deals with the study of cervical cancer prevention which can be performed optimally thanks to a large screening campaign combined with machine learning methods to recognize the individuals most at risk and therefore need to be checked more frequently, so as to allow a resolution of the disease at a very early stage, if it occurs, allowing to save a considerable number of lives. In the first part of the work we built a neural network based on the concept of Residual Network. We applied various techniques, which will be analyzed specifically, trying to achieve satisfactory results, but above all to study the behavior and fully understand the difficulties of approaching a real case. Finally, we traced the behavior of some models using transfer learning, to compare the behavior of the two types of training in an area still difficult to treat as medical.

1. Introduction

Cervical cancer is the second most common cancer affecting women worldwide and the most common in developing countries, but if it is diagnosed at an early stage the treatment becomes simple and there are high rates of recovery. The study divides the cervix into three types, two of which require more frequent and detailed investigations and checks. The distinction is not easy, wanting to make a brief summary, without wanting to replace a medical opinion for which we do not have the expertise, is based on the position of the Transformation Zone, the part where two different types of epithelial cells are touched (endocervice and ectocervice), the more this zone is exposed and therefore external and the greater the possibility that the tumor is diagnosed early, on the contrary need to keep the patient under control. We have therefore used the dataset and the challenge provided by [3] to approach the problem in the best possible way.

2. Related Works

The worldwide trend to move towards deeper and deeper networks in the medical field has had a much more moderate step, both for the intrinsic difference of medical images compared to the classic ones on which networks are trained (such as Imagenet), and in some specific areas due to the lack of availability of good quality images. Since we do not have any knowledge in our cultural background regarding the technological situation in the medical field, we have relied on the paper by Litjens et al. of February 2017 [4], in this work they give a very comprehensive overview of the techniques used and the various areas that are touched, providing the reader not only with an excellent overall picture of the state of the art, but also with a large number of ideas on which to explore the various topics. This paper also gave us a lot of ideas on what other works to look at. We were drawn to the approach used by the paper by Menegola et al. 2016 [5] that analyzes the importance of transfer learning and its effectiveness in the analysis of images for the discovery of melanoma in an extremely early state, so that it can be fought more effectively. During the course of the study more transfer learning tests are performed, enabling or not fine tuning and starting from weights already trained on different sets, linked or not linked to our area of interest. In this paper scholars distinguish three types of transfer learning applied, one on medical data, one on classical data and one combined. Taking advantage of the results of the study, we observed that the combined one, although it seemed like a good idea, is the one that works the worst; for this reason we used their results opting for a pure transfer learning based solely on ImageNet which is what turned out to give better results. Finally, a trained from scratch network is also used and the results are all compared to draw conclusions. All the networks in question are based on the VGG-M architecture, a slightly deeper CNN deeper model than AlexNet. Taking a cue from here we therefore decided to make a comparison of this kind, always in the medical field, but trying to use clearly deeper networks. For this

reason, it was necessary to read the paper by He et al. [1], with which they presented ResNet, the aim was to understand more deeply the concept behind the residual networks so as to be able to build one from scratch by understanding the various steps a little better. For the construction of ResNet we read a second paper by the same authors, with an improvement regarding the residual block [2]. Finally, one last paper influenced our work, even if we identified it during the experimentation, it is the work of Zimmerman et al. [8] that does an excellent analysis on the segmentation of the cervix by dividing the process into various phases and obtaining ever better results, from this paper was born the code written by the user Chattob of Kaggle [6], whose code allowed us to apply a segmentation, also if not optimal as the one described in the paper, but which nonetheless has significantly improved the quality of the dataset.

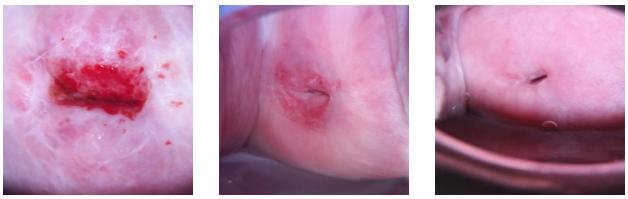
3. Dataset

The dataset used was provided directly by Kaggle for the challenge, it has a rather small size given the difficulty in obtaining good quality photos and especially representing the various types of cervix to which it was decided to assign each photo. The distribution of the dataset is as follows:

Kaggle Challenge Dataset			
Folder	$Type_1$	$Type_2$	$Type_3$
Train	250	781	450
Additional	1189	1558	1886
Test	512		

To increase the cardinality of dataset used for training Kaggle added to the train a large amount of additional images, but these images are of poor quality, the folder contains a lot of repeated photos of the same patient, but also blurred photos, very dark or with the cervix that occupies only a small part of the image, making life very difficult to the algorithm that must analyze it; in addition, in many of the photo there is the tool used to dilate the vagina and allow the shot. As far as the size is concerned, the format is very varied, you can go from the largest that exceed even 3000x4000 to the smallest that are just 480x640. After observing the dataset we decided to start our work using, at least in the first phase, only the train dataset, this because we thought that the images of the additional folder could affect the results in a negative way, even if they would raise the cardinality of the dataset. Even if at a later stage the additional will be added to the images examined, it will always be necessary to keep this difference in quality in mind and to take precautions, especially in the distinction between train and validation, so as not to compromise the study. A brief discussion on what approach could be taken will be discussed in a later section.

Since the beginning we talked about three types of cervix, we decided to inform ourselves slightly more about how to recognize the various types so that we could make more informed choices in the pre-processing phase, but especially in the data augmentation phase. To summarize, the differentiation between types is based on the endocervical skin, which in the photos has a red appearance, which can be more or less exposed. It is important to identify $Type_2$ and $Type_3$ because they belong to people whose possible lesions are not clearly visible and therefore need to be subjected to further tests. As you can see from the table there is a clear imbalance of classes, this is because Type-1 and Type-3 are the clearest situations (endocervix all exposed or not exposed at all), while Type-2 includes the whole grey scale in between, so it has a much higher number of samples.



(a) Type 1 (b) Type 2 (c) Type3

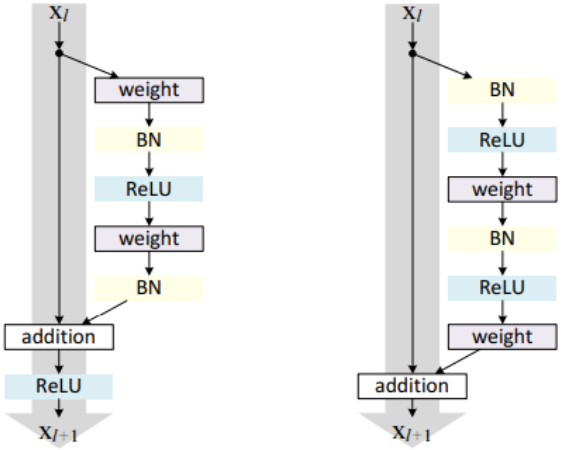
Figure 1: Here are shown the three types of cervix analyzed

4. Methods

4.1. Residual Network

With regard to computer vision, it is well established that deep networks are able to achieve better performance compared to more original CNNs such as AlexNet. The great performance increase has been achieved thanks to a better gradient treatment. More specifically, in 2015 He et al.[1] empirically showed that the extension of additional layers to a network to improve its performance remains valid only below a certain threshold, beyond which this operation does not bring any benefit but rather a degradation. Therefore, the concept of building block has been introduced where the relevant operation is the shortcut connection which allows to contrast the problem of vanishing/exploding gradient. More precisely, this connection is called identity shortcut connection because the input is taken and passed to the output through the identity function. This solution intrigued us so much that we tried to build our ResNet. The key element from which we started is the residual block, of which we have studied in depth some characteristics before starting to write code. First of all, based on what has been said in the original work we used as shortcut the identity connection instead of the projection connection, as well as the use of the bottleneck block (instead of the basic block) when the

network has more than 50 layers. This is an interesting solution, introduced mainly for computational reasons since it is well established that 3x3 convolutions are very expensive. Therefore the basic idea is to ”reduce” the image before executing the convolution in such a way that it is less expensive. This is done by applying a 1x1 convolutional layer before the 3x3 convolution and another one after, in which the first one has the purpose to reduce the input image channels while the last one takes care to bring the image back to the starting shape (note that it is therefore possible to treat a higher number of features maps which means you can deal with higher resolution images). The most compelling part of the analysis was the arrangement of the various layers inside the residual block. He et al. performed a very thorough analysis in their other work [2] which is based on the importance of identity mapping. In fact, analysizing the backpropagation phases it is shown how an alpha ending greater or less than 1 leads to the problem of gradient exploding (and therefore the loss cannot converge) or gradient vanishing (and therefore the loss stops on a plateau). This has led them to say that the shortcut connection must be left clean (which means alpha=1) and then the ReLu placed at the end of the residual block must be removed. So, several combinations have been tried, at the end the full-preactivation is the one that got better results because it has both the benefit of leaving the shortcut clean and of using the ReLu in conjunction with the Batch Normalization, so that the ReLu can take full advantage of the benefits of the latter. Finally, sticking to the following analysis, we built the residual block.



(a) Original Residual Block (b) Full pre-activation

Figure 2: Transformation of Residual Block

4.2. Architecture trained with Transfer Learning

Although we had read about the more than positive behavior of some architectures such as VGG-16, we did not use them to stay faithful to our purpose which was to observe the behavior of deeper networks, so we focused mainly on:

- Resnet50, this is the network described in the paper by He et al. [1] and therefore with the residual block characteristics described in the previous paragraph.
- InceptionV3, is a network that makes use of a particular technique, the Inception module, from which it takes its name. The main concept introduced in V1, to which small additions will be made in later versions (such as regularization tricks), is to make the networks wider rather than deeper. This is because the deeper a network is, the more likely it is go into overfitting, to avoid this behaviour a series of filters are not applied sequentially, but several filters of different sizes are applied at the same level, the results of which will then be concatenated before moving on to the next level.
- InceptionResNetV2, as the name suggests, combines the two main concepts of the networks described above. In fact, the residual block principle is joined to the inception module, often concatenated with a module that scales the residual activation to prevent the ”death” of the network.

4.3. Overfit prevention

As we said before, one of the big problems of this challenge is the small amount of images in the train set, especially when compared to the depth of the networks we wanted to use. This disparity will undoubtedly lead the networks to overfitting after some time, this will certainly further worsen the results on the set of validation and consequently on the test, so we will try to prevent this situation mainly in two ways: data augmentation and regularization. The data augmentation will allow us to increase and differentiate the images provided to the network at each iteration, dynamically modifying them randomly. Given the nature of the dataset, but above all the position and size of the uterus, which in some images is very small or in a position far from central, we will not be able to make a too stressful action of data augmentation, which therefore modifies the image too much. We will limit ourselves to applying random rotation and flips both horizontal and vertical. We would also try to work on the brightness, which we have read to be a complicated topic in this specific area, because the photos are taken with special 35mm cameras placed in close contact with the vagina, but by illuminating the part to be photographed tend to make some parts too bright, slightly distancing the result from reality. This treatment deviates from our purposes and so we will simply try random lighting changes. Another possibility we have taken is to use regularization to prevent overfitting, especially dropout. We made this choice despite the fact that in the paper by He et al. [1] describing the

Resnet, dropout is not used, but they had a large amount of data available so it is a risk that we decided to take to try to counteract overfitting that otherwise could lower our performance. We remember that the dropout is a method of regularization that allows you to simulate the training of multiple networks in parallel, because at each time you decide to turn off some neurons, in this way the fully-connected part of the network focuses at each iteration on different features, making the model more flexible and less tied to input data.

5. Experiments

In order to evaluate and learn more about our built ResNet, our general approach is to perform the first runs on both from scratch and transfer learning networks, of course we expect the latter to work much better. The results obtained will be our starting reference, then different techniques will be applied and their effectiveness on the from scratch network will be analyzed, finally we will observe the different benefit obtained on the pre-trained ones compared to the from scratch one.

5.1. Image pre-processing

From the preliminary analysis carried out on the dataset it has emerged that due to the great variance in terms of image shape, it is necessary to perform some short preprocessing operations:

- In general, larger images allow the network to learn more specific features, but increase the processing time; in our case the use of the bounding box tends to create very small crops, so we couldn't enlarge too much to avoid pixeled images.
- Pixel normalization: divide each pixel by the value 255 so that each one is in the range 0 to 1.
- Also note that centering(subtract the mean image) has not been performed since it is usually not a best practice in images.

The same transformations were also applied to the additional data.

5.1.1 Train and validation split

Given the non-trivial nature of the additional data, a more accurate treatment than normal is required. After trying to mix them with the train data and then assigning them a minus weight, we didn't get significant result and during the early epochs it was noticed that the validation scores are slightly more unstable, because the additional images introduced some noise that affects the training and validation partitions. Then we understand that the real problem was not the weight of the additional data, but the way we split train and validation. Including the additional images in the validation is wrong in our opinion, because the fact that some of them are duplicated means that one copy could end up in the training and the other in the validation, this is not good because in a real-world case it will never happen that the network sees an image that it has already seen during the training. We were not able to find a simple solution to solve this problem, so we relied on the online community, where this post [7] proposes a very simple and powerful solution, we chose to re-implement it in the same way. The solution is based on the observation that duplicate images, images related to the same patient or extremely similar images, have a strong correlation in terms of colors, so for each photo the colour histogram is calculated and on all histograms is performed k-means with k=100, 80 clusters will be the training set and 20 the validation set.
Since this is the first time that we are facing such a problem, even if it is not our solution, we thought it was appropriate to apply it and study its advantages in a way that we would be less unprepared in the future.

5.2. Baseline experiments

During the first experimental runs, several tuning operations have been performed, the most relevant are the choice of the optimizer(note that it may differ between from scratch and pre-trained networks) and the learning rate, the depth of the network, the number of epochs. The two main optimizers compared were Adam and SGD+Momentum. During the choice between these two, it was kept in mind, briefly, that Adam tends to work better for most problems and its main difference from SGD is that it performs a diversified update per parameter (it is an adaptive method), also handles saddle points better and converge much faster. SGD, on the other hand, has been seen empirically to generalize better than adaptive methods. As for the depth, we know as already said that a network too deep, working on this dataset, would tend to overfit, making a few runs we even observed that the network tended to predict only one type(Type 2, the most frequent).

Therefore, we focused only on ResNet with 34 layers and all subsequent experiments and improvements will be tested mainly on it. Transfer learning, even if its application is trivial, still requires a short analysis. There are two alternatives, using the network as a fixed feature extractor or fine tuning the whole network. The choice is dictated by the size of the dataset and the similarity with the pre-trained dataset. Obviously the size of the dataset was the most influential factor, which led us to say that the most obvious solution was the fixed feature extractor, moreover the problem of dataset differentiation does not cause a worsening of the data, as demonstrated by Menegola's work[5]. We believe that the failure of fine-tuning, in our case, is due to

the fact that the scarcity of the dataset and the low quality of some images tends to "dirty" the features extractors. All results will be discussed in the dedicated paragraph.

Keep in mind that, even if here we have chosen not to discuss some hyperparameters because they are trivial, they have been taken into account during our experiments, such as the number of epochs, the learning rate, early stopping, the reduction of the learning rate when on a plateau, for which the values differ for transfer learning and training from scratch.

Despite this extensive tuning on a variety of hyperparameters, as we have seen the results are still not very satisfactory. This behavior, as we had also previously assumed, is certainly due to the quality of the dataset, so we will describe how we tried to improve it using some techniques.

5.3. Dataset improvement

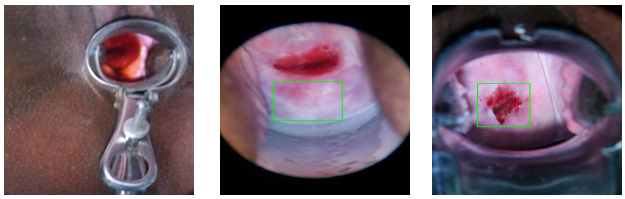
To start surely the first step was to increase the cardinality of the dataset so we applied the types of data augmentation described in the appropriate paragraph. We didn't find great results as far as the final accuracy or loss is concerned, the only real benefit is overfitting, in fact all the runs to which data augmentation was applied always showed the presence of overfit, but 15-20 epochs later than their respective counterparts, a sign that indicate that we were acting in the right direction.

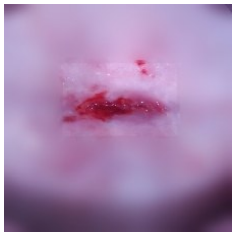
Our next observation was that in additional dataset some photos represented the cervix only in small part [fig: 3a]. In general the part on which it was interesting to make the assessment, was the one concerning the os. This part is always very narrow compared to the image as a whole, also in good quality images, so it seemed reasonable to us that it had a greater weight than the rest. So we found some annotations on the os area, for all the images inside the train folder, these have been provided by the user Paul from Kaggle, who works in the field of biology and oncology, thanks to these annotations we could manually cut out the part we are interested in of the image and paste it on top, at the same coordinates, of the original image, that was previously transformed thanks to a blur operation. The result is the one shown below [fig: 3d], the images will have a well focused rectangle, which contains the os and all the necessary traits, while the rest will be much more blurred. The problem was that these annotations were only present for the train images, so we had to find a way to perform the same operation on the internal images of the additional, so that we could perform the train on data having all the same structure. For this we found extremely useful the Chattob code [6], the user proposed to make a segmentation of the cervix, first cleaning the image from the contour due to the dilator and the camera and then, the interesting part for us, to identify the ROI (Region of Interest) for the classification. We studied the code provided and we tried to readjust it to our purposes, the algorithm tried to identify the fundamental area of the photo, i.e. the one containing the os, then performing object detection and enrolling it in a bounding box that allowed us to better identify it [fig: 3c]. The algorithm worked well in most situations on the extraction of the ROI of the cervix, less well instead was made the recognition of the os, which was often not centered in the bounding box or sometimes partially cut. We slightly modified the code to pass the coordinates of the bounding box to the previously written functions. This allowed us to do the same operation by cropping the affected area and blurring the rest of the image. Once we applied this transformation to all the images we tried again to train the network, but the results were broadly the same. We then read the approaches of the other users in the challenge and realized that the best thing could be to completely crop the identified area from the image [fig: 3e]. At this point the only thing to do was to bring all the cropped rectangles to 224x224 squares. To make this operation we had to add padding to make the image a square before making the resize, this allowed us not to crush or stretch the image, which certainly would not have had a positive effect on our network. It's important to underline how with the algorithm of bounding boxes, some images, like the one shown below [fig: 3b], are not cropped correctly, but we thought it was worth trying because the number of samples that would have lost in quality was definitely contained.

(a) Non centered cervix

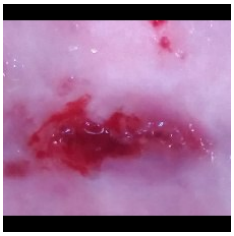
(b) Bad positioning of Bounding Box

(c) Good positioning of Bounding Box





(d) This image was blurred around the os



(e) This image has the ROI of os cropped

Figure 3: Here we can see the various transformations applied to images

Finally, after this last trick we began to see some results, objectively better than the previous sections, this is probably due to the fact that the blurred pixels did not allow the

network to learn less about that area, but simply confused it by making it learn unwanted features.

5.4. Results Evaluation

In this section a comparison will be made between the starting situation and the one we managed to arrive at after all the improvements discussed above. It was decided to write in these tables significant and synthetics values indicating the value around which the model had stabilized.

Baseline Results				
Network	Train		Validation	
	Acc	Loss	Acc	Loss
Resnet34	0.8834	2.637	*	3.952
Resnet50	0.9347	0.1934	*	1.6859
InceptionV3	0.498	1.054	*	6.033
Inception ResNetV2	0.6625	0.7812	*	11.1955

Final Results				
Network	Train		Validation	
	Acc	Loss	Acc	Loss
Resnet34	0.5285	2.436	*	2.945
Resnet50	0.5363	0.6372	0.4738	1.032

The success of the challenge was based on achieving the lowest possible loss value on the test set, so it was the parameter on which we weighed our choices the most. As we can see ResNet50 is the one that reached the best results already initially and has maintained its trend throughout the study. As far as the accuracy on the validation, in the tables there are some '*', this because the networks, above all initially, tended to predict all the values as belonging to a class, mainly to the $'Type2'$ being the one clearly more numerous (as it is possible to observe in the first confusion matrix above). Also the final accuracy of our ResNet34 was not stable, but the oscillations were much smaller and often limited to the prediction of only two classes instead of three.

We felt it was pointless to show the improvements we received on InceptionV3 and InceptionResNet, as we have not been able to produce improvements comparable to ResNet50, we attribute this behavior both to the performance of the networks, which are probably not suitable for this purpose, and to our lower experience with these networks compared to ResNet.

Talking about the Resnet50, which represents our best result, we can see how the loss on the validation set, is around values that we consider very positive even when we compare with the Leaderboard of the challenge. In particular, we also managed to eliminate almost completely the overfitting that was one of our main initial goals, you can see how the accuracy between train and validation differs by a few points. It should be noted that the accuracy has

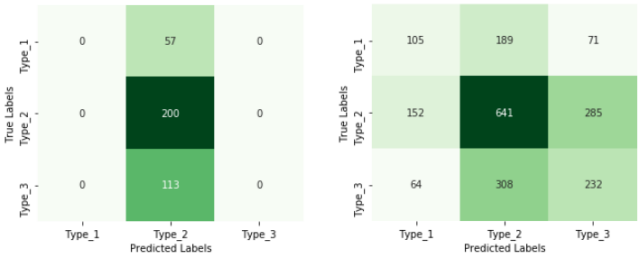


Figure 4: On left the initial ResNet50 confusion matrix, on right the final one

not suffered great increases in numerical level, but as you can guess by looking at the matrices, the 54% achieved in the case of the right is not absolutely reliable and therefore should not be considered for comparison. On the contrary, the right matrix predicts all the classes, even making mistakes, but reaching a 47% result of real predictions. In fact the loss goes down clearly, as evidenced by the graphs.

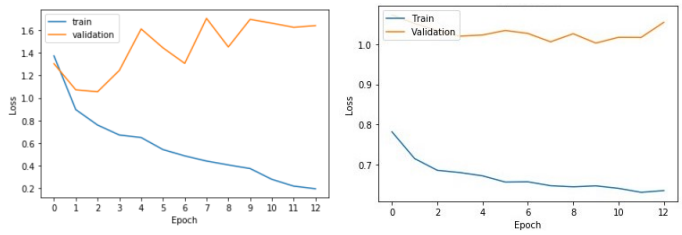


Figure 5: On left the initial ResNet50 loss graph, on right the final one

To highlight the reason why the loss has fallen we have taken some samples and the predicted probabilities between the starting and the final network were compared. Initially, the network had prediction probabilities not centered on a single label, this led the loss to rise also in case of correct prediction. With the final network instead, analyzing the same samples, for most of them this bad behavior was alleviated. We report here below a comparison on a sample taken in examination before with the initial and then with the final network.

```
[Sample  P(class_0), P(class_1), P(class_2), pred_label, truth_label,  pred==truth]
[376.jpg       0.13   0.4800001        0.39          1.           1.           1.]
[376.jpg   0.090777      0.6902        0.21          1.           1.           1.]
```

These above were the results achieved in the current state of the project, in view of the challenge, if we assume to achieve better accuracy and to export the network to the real world, we should certainly make further evaluations on

the results. We are in the medical field, where not all mistakes have the same weight, predicting a person as $Type1$ would lead her not to perform further examinations and could therefore make her believe she is healthy, leading her, for example, to discover a possible disease at a more advanced stage. So in this perspective the confusion matrix, from which through precision and recall for each class, it is possible to prefer one class over another. In the medical field, in situations of doubt it is preferable to predict a sample as sick (subjecting it to further investigation) rather than healthy without being sufficiently sure. In our case, this means ideally having a high precision for $Type1$ and at the same time a high recall for $Type2$ and $Type3$.
In conclusion, we submitted to the challenge and placed at position 224 of 848.

6. Conclusion and Future Work

Compared to when the project started, we now have a much deeper understanding of the problems encountered in a real-world situation. At first we thought it was a choice of the right network and its configuration, but this turned out to be easy to understand. What we unexpectedly clashed with is the treatment of the image, understanding its semantics so that we could transmit it to the network, performing very intensive pre-processing operations, only at this moment we were able to appreciate the great impact of this phase.
As already proved in the results, despite the great diversity between ImageNet and cervix dataset, transfer learning certainly works better, and for each improvement applied to the images the network from scratch drew only a small improvement (but for us however significant) while instead the pre-trained network improved much more. In the face of this observation, the work proposals listed below are all aimed at a continuation in the treatment of images(programmed by us but not carried out due to lack of time), because at this point of the project has proved to be the phase with more room for improvement.

1. To perform a further more intensive analysis, for example through the use of saliency maps to understand what leads the model to classification errors

2. Clean up the additional images, through the use of histogram clustering to exclude a cluster from the training to see if that cluster has influence on performance or not (and then remove it from the dataset)

3. Use Generative Adversarial Networks(GAN) so that our networks are hopefully able to learn more features

4. Take advantage of the fact that we have handmade bounding boxes to train a bounding box prediction model

References

[1] Ren S Sun J. He K, Zhang X. Deep residual learning for image recognition, 2016. https://arxiv.org/abs/1512.03385.

[2] Ren S Sun J. He K, Zhang X. Identity mappings in deep residual networks, 2016. https://arxiv.org/pdf/1603.05027v2.pdf.

[3] Kaggle. Intel mobileodt cervical cancer screening, 2017. https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening/overview.

[4] Kooi T. Ehteshami B. et al. Litjens, G. A survey on deep learning in medical image analysis, 2017. https://arxiv.org/abs/1702.05747.

[5] Fornaciali M. Pires R. Avila S. Valle E. Menegola, A. Towards automated melanoma screening: Exploring transfer learning schemes, 2016. https://arxiv.org/pdf/1609.01228.pdf.

[6] Chattob Kaggle user. Cervix segmentation (gmm), 2017. https://www.kaggle.com/chattob/cervix-segmentation-gmm/notebook.

[7] vfdev Kaggle User. Type 1 clustering, 2017. https://www.kaggle.com/vfdev5/type-1-clustering.

[8] Greenspan H. et al. Zimmerman G., Gordon S. Automatic detection of anatomical landmarks in uterine cervix images, 2008. https://ieeexplore.ieee.org/document/4663866.