Cervix Screening Classification for Cancer Preventive Treatment

Sebastiano Bea
Stanford University
sbea@stanford.edu

Kevin Poulet
Stanford University
pouletk@stanford.edu

Dan Zylberglejd
Stanford University
dzylber@stanford.edu

1. Abstract

This project is trying to address the cervix screening classification in order to improve cancer preventive treatment. This is a big challenge, as better classification means better treatment and lower cost, and this task is very hard to perform, even by specialized doctors. The team used Inception Resnet and VGG 16 networks to classify the cervix type based on an image. The data was provided in the context of the Kaggle competition. Using transfer learning, and re-training the last two layers of VGG 16 (pre-trained on COCO) enabled the team to reach 58% of test accuracy, and to rank mid-table on the Kaggle leaderboard.

2. Introduction

We will investigate the problem of cervix type classification, defined by Intel through a Kaggle competition (Kaggle Competition [5]). It is a project with a potentially large impact in helping the prevention of cancer in early stages on women all over the world. The project scope is very straightforward, and consists on classifying a specific picture of a cervix into 3 main types of cervix (each of which has its own risks and characteristics, as well as different ways to treat it).

We plan to use standard CNN architectures (VGG, Inception, ResNet...), pretrained on a standard database (for instance, Imagenet), but fine tuned to our specific dataset in smart ways (for example, by testing to decide how many of the last layers should be fine tuned). Our approach is similar to the one described in a paper on the use of convolutional neural networks for medical image analysis [10], except that the methods implemented there are applied to object detection, whereas we focus on classification.

We will evaluate our accuracy with log-loss error (cross-entropy), and compare our scores to other Kaggle teams. We expect to achieve accuracy better than random guess by a large margin. Since the data is very noisy, we are very skeptical about initial results, but given the large size of the database, we hope to be able to identify the main patterns.

The input consist of jpg images obtained from the Kaggle competition website. The images are not pre-processed and all processing and preparation will be performed by the algorithm. For each image, the output consists of log predictions for each of the possible classes. The highest prediction represents the class which the image should be assigned to.

3. Problem Statement

3.1. Motivation

Cervix cancer treatment's effectiveness varies significantly patient to patient. Even in low resource settings such as developing countries, it is possible to easily identify high risk patients and start preventive treatment of cervix cancer in a single visit. However, due to the lack of expertise and the difference in the cervix position of each woman,

defining the correct treatment for a specific patient is a challenge.

This places doctors in a difficult position, as they can easily identify subjects which are at high risk of developing cervix cancer, but cannot prescribe the correct preventive treatment. In addition, prescribing the wrong treatment has a high health care and human cost, as the wrong cure can actually mask the growth of a cancer in a woman, reducing the risk of success for further treatment in the case of a cancer effectively arising in the patient.

Successfully defining the cervix type of a patient would provide valuable information to doctors in verifying patient eligibility to specific treatments, reducing the prescription of wrong cures and, in the long run, increasing the success rate of cervix cancer prevention. This would be particularly helpful to rural doctors that do not have access to advanced medical equipment that aids in the identification of the cervix type.

The aim of this project is to correctly classify the cervix type of a woman based solely on images of the cervix, something which even rural doctors can obtain given their restricted resources.

3.2. Challenges

The problem presents a number of challenges.

First of all, cervices are very similar across types, and the distinction is not apparent to people without a specialized background in the subject. The types mostly differ by transformation zone, and the main characteristics of each type are[7]:

1. Ectocervical, fully visible, small or large transformation zone

2. Has an endocervical component and may have a small or large ectocervical component. The transformation zone is fully visible

3. Has an endocervical component and may have a small or large ectocervical component; the transformation zone is not fully visible

A brief summary on the interpretation of cervix images is provided here [1]. The differences between the cervix types are subtle and not immediately apparent to the human eye. This underlines the importance, but also the difficulty, of this project.

Secondly, the images have been gathered using different camera / tools and this generates very different images of the data. This will reduce the probability of overfitting, but will also make the model more difficult to train. According to the type of tool used, some images have a black border and are circular. Every image has a different orientation and zoom factor, so we will have to take that into account too.

4. Related Work

The main inspiration for our work is contained in the paper "Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?" [10] by Nima Tajbakhsh et al. The paper addresses the issue of how the features of medical images of a specific kind, characterized by unique borders, colors and shapes, might undermine the effectiveness of the common practice of using pre-trained nets for classification and detection, given that the features present in medical imaging are very different from those found in common datasets used for pre-training (Imagenet, COCO...). The conclusions of the paper state that pre-trained nets do not underperformed compared to nets fully trained on medical imaging. This pointed us in the direction of using transfer learning. We will also use the technique of retraining different layers and comparing results on the validation set in order to identify the optimal number of layers to retrain. In order to understand more about the subject we are dealing with, we used the book "the Cervix" [7] by Joseph Jordan et al. Even after the consultation of the book and appropriate material, we struggled with the identification of the type of a cervix based on a raw image.

5. Methods

5.1. Initial Testing

We will use the Tensorflow framework [2] to conduct our analysis.

5.2. Inception Resnet

Our initial implementation considered using a Inception Resnet given the higher efficiency and performance [8]. We utilized the Tensorflow implementation of Resnet [3] with SGD optimizer and a decaying learning rate and pre-loaded the weights of the parameters, given our intention of using transfer learning. As a starting point we retrained only the last fully connected layer of the net. We did not load the weights associated to the last layer and seeked to establish them through 10 epochs of training just on the last layer. After that we trained the whole net for 10 epochs to fine tune it.

5.3. VGG 16

This solution occupied a lot of training time and, as described in the results section, the hyperparameters needed fine tuning. In order to be able to experiment in a more iterative way, we also utilized a VGG 16 net according to the implementation of Simonyan and Zisserman [9]. This VGG net is made up of five 3x3 convolutional layers each followed by a 2x2 max pooling layer, then two fully connected layers each followed by a dropout layer and a final fully connected layer. We also loaded the weights of a VGG pre-trained on COCO. A great part of our coding work is based on the fine-tuning algorithm provided by the CS231N staff [6].

By taking inspiration from the medical paper, and seeking to understand the optimal number of layers to be retrained, we iteratively re-trained the last layer, the last two layers, the last three layers... recording training and validation accuracy each time. The objective for this stage is to identify the number of re-trained layers that optimizes the accuracy of the VGG net, then further fine-tune that and use the obtained net to obtain an estimation of the test error on the test set

6. Data

6.1. Data Origin and Availability

Two labeled datasets have been provided by Kaggle [4]. The first dataset (from now on data1) is approximately of 1500 images (5.5 GB) and each image is from a unique patient. The second dataset (from now on data2) was provided at a later date and is a lot bigger (approximately 26 GB), but includes multiple images from the same patient.

Kaggle also provides an unlabeled test set, which they use to assess the participant's models. Given that the data is not labeled, we do not use this data for the scope of this project, but we will submit a prediction to Kaggle once we have ultimated and fine tuned our model.

Data1 does not provide enough images, especially after a train - val - test split. However, each image comes from a unique patient, thus enabling the data to be split randomly without increasing the risk of overfitting. Data2 instead includes multiple images from the same patient and these images are not tagged in a way that it is possible to group the images coming from the same patient. If images from a same subject were to end up in both the training and validation set, then there would be a high probability of overfitting, given that similar images would be encountered in the training set and therefore be classified correctly easily once seen in the validation set.

6.2. Data Management

Due to the risk of overfitting using data2, we will use solely data1 the for the finetuning of the hyperparameters and during the identification of the optimal numbers of layers to re-train. This keeps our computation time reasonable and ensures that we take hyperparameter and level re-training decisions by using a viable dataset. We will then use data2 as a block to train the final model, which will then be validated and tested on data1. For the purpose of Kaggle submission, we will train on data1 + data2 after having tuned the main hyperparameters.

6.3. Data Preprocessing

Regarding augmentation, we apply a random cropping and horizontal flipping every time an image is taken to be part of a mini-batch and go through training. Since most images are already very blurred, we do not add gaussian noise.

Concerning preprocessing, we had to uniform the images in order to be able to run them through

an algorithm. The input images are first resized to have the smaller side of maximum 256 pixels long, then randomly cropped to obtain a 224x224 image. The result is then horizontally flipped half of the times (only for training data) and the pixel value of the mean image of the Imagenet data used to pre-train the net is subtracted.

6.4. Data Details

Data1 is the dataset that we used during the fine-tuning of the hyperparameters and the retraining of the layers. All our initial results are obtained from it.

It includes a total of 1481 images divided in the following way between classes:

- Type 1: 250 images
- Type 2: 781 images
- Type 3: 450 images

As we can see, type 2 is the most prevalent class, followed by type 3. On the other hand, type 1 is a lot less frequent. We will have to take into account this unequal split when evaluating result, as the algorithm could end up predicting type 2 more often only because it is prevalent in the training set.

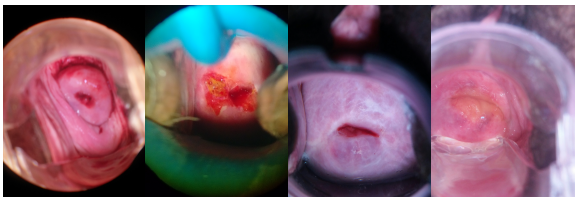
Data 1 was randomly split into train, validation and test set with the following proportions (but maintaining Type 1 - Type 2 - Type 3 ratio):

- 60% Training: 888 images
- 20% Validation: 296 images
- 20% Test: 297 images

The training set will be used to train the net, the validation set will be used to benchmark the accuracy and loss for each of the hyperparameter or layer re-training iteration and the test set will be used to obtain an estimation of the accuracy of the model after all hyperparameter and layer re-training decisions have been made. We will use the results on the validation set to pick the number of layers to retrain from scratch on the final model and to choose the hyperparameters.

6.5. Sample Data

For better understanding, we present some examples of cervix images as provided by Kaggle. Note how different all images look between each other, even for images from the same class. That makes the challenge even harder, since there seems to be very little patterns for images in a specific class.

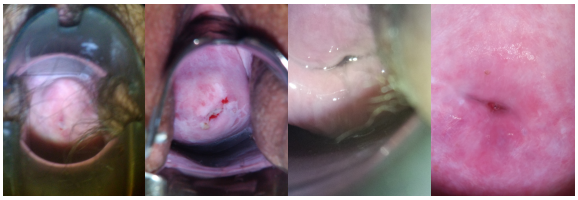


(a) (b) (c) (d)

Figure 1: Type 1 cervix images

(a) (b) (c) (d)

Figure 2: Type 2 cervix images



(a) (b) (c) (d)

Figure 3: Type 3 cervix images

7. Results

7.1. Resnet

We started by training the last layer of the Resnet from scratch with 10 epochs, then train the entire model for 10 epochs. The hyperparameters were:

- Learning rate: 1e-3. Each 3 epochs the learning rate is decreased by an order of magnitude (*0.1)
- Dropout keep probability: 0.5
- Weight decay: 5e-4

With this setup we did not achieve any result. The net was always predicting the same class (type 2) which is also the most frequent. In order to understand where we were going wrong we passed to training a VGG 16 net.

7.2. VGG

For the VGG we tweaked the hyperparameters differently for each number of layers we were retraining. We started off by trying the following parameters for all the possible values for the number of retrained layers:

- Number of epochs: 12
- Learning rate: 1e-3. Each 3 epochs the learning rate is decreased by an order of magnitude (*0.1)
- Dropout keep probability: 0.5
- Weight decay: 5e-4

We obtained successful results for the retraining of the last one, two and three layers. In those cases the net was learning and we obtained accuracies higher than 52% for both training and validation. The accuracy threshold of around 52% is significant as that is the point that corresponds to the algorithm predicting always type 2. While retraining 4 or more of the last layers, the accuracy was always 52%, equivalent to always predicting type 2.

We identified the reason of the failure to be the retraining of the convolutional layers (which start from the 4th last layer).

By tweaking the parameters to the following, we managed to obtain results different to always predicting type 2:

- Number of epochs: 12
- Learning rate: 1e-4. Each 3 epochs the learning rate is decreased by an order of magnitude (*0.1)
- Dropout keep probability: 0.7
- Weight decay: 5e-5

The comparison of the results per epoch for the training of a different number of last layers is graphed below:

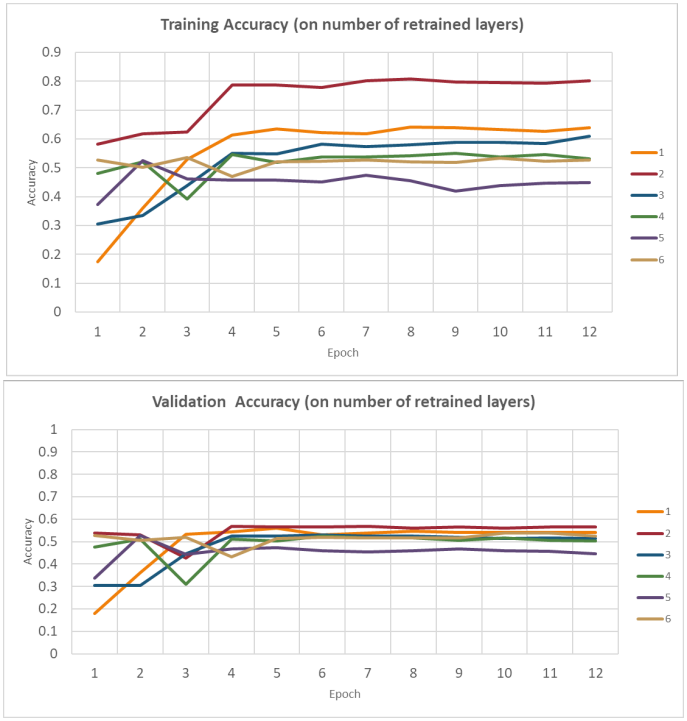


Figure 4: Training and validation accuracy per epoch, for each number of retrained layers

According to our results, the best number of layers to train from scratch is two, as that achieves a

better accuracy both on the training and on the validation set. Another result from the previous graph is the fact that after 5 epochs, most of the learning has been performed.

8. Conclusion

After re-training the last two layers of the model on the training and validation sets combined, we evaluated its performance on the left-out test set. This would be a good estimation of the performance of the model in case of use on a new patient. Our model had a test accuracy of 58.45%.

This performance seemed pretty low to us, and it is probably caused by two main factors. First of all, as said in the introduction, this classification problem is very hard as the images are very similar. The VGG 16 is probably too simplistic for a net structure to be able to capture the inherent feature distinctions between the different classes representing the cervix types. The second reason is that this model has been trained on data1 only, not on all of the data available for this competition. If we would train it on the bigger dataset, we would expect the accuracy to increase by a significant amount.

After evaluating the test accuracy of our model, we decide to retrain it on the whole labeled dataset data1, and use the resulting model to predict the cervix type for images from the unlabeled test set provided by Kaggle. We then submitted our predictions to the public leaderboard on Kaggle website, and we are currently ranked at a mid-table position, with a cross entropy loss of 0.94841. The best teams at the time of this paper had a loss of 0.4 approximately (one had a loss of 0.07, and 3 teams managed to reach a zero loss).

References

[1] https://kaggle2.blob.core.windows.net/competitions/kaggle/6243/media/Cervix%20types%20clasification.pdf, title = Competition Cervix Type Explanation,.

[2] https://www.tensorflow.org/.

[3] https://github.com/tensorflow/models/blob/master/slim/nets/inception_resnet_v2.py.

[4] Data source. https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening/data.

[5] Kaggle competition page. https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening.

[6] J. Johnson. Project starter code. https://gist.github.com/omoindrot/dedc857cdc0e680dfb1be99762990c9c.

[7] J. Jordan and A. Singer. *The Cervix*. Wiley-Blackwell, Apr. 2009.

[8] F. Li, J. Johnson, and S. Yeung. Cs231n course material.

[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image reognition. *arXiv*, 2015. https://arxiv.org/pdf/1409.1556.pdf.

[10] N. Tajbakhsh, J. Y. Shin, uryakanth R. Gurudu, R. T. Hurst, C. B. Kendal, M. B. Gotway, and J. Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299 – 1312, May 2016.