Capstone Project

Intel & MobileODT Cervical Cancer Screening

Daniele Galloni

---

1 Definition

Project Overview

Cervical cancer is highly treatable if caught in its early stages. Therefore, large scale cervical cancer screening programs have been implemented in many countries, to detect early signs of cervical abnormalities and thus begin early treatment. Such procedures are often simple enough that for many women it is possible to be screened and treated in a single medical visit. However, different cervix types require different treatments, and not all types can be treated in a single visit. Especially in rural settings, there is a lack of expertise in identifying the various types of cervices, which in turn results in incorrectly-chosen treatments. Not only are such treatments ineffective, thus unnecessarily delaying adequate care, but they can also hide future signs of cancerous growth on the cervix. The correct identification of the cervix type is crucial to determine the correct treatment option, and to ascertain whether more advanced treatment is required.

For this reason, MobileODT offers remote medical support to healthcare providers, by aiding in the identification of cervix types. This is done through an Enhanced Visual Assessment (EVA) System, a biodiagnostic tool embedded in specially-designed devices, which combines internal algorithmic tools with external information provided through the mobile-phone network, to support cervical screening visits. The work-flow required at MobileODT can be greatly aided by a more advanced automated classification of cervix types.

Some automation of cervical cancer screening and detection already exists. For example, an algorithmic triage for screening programs has been proposed in [1]. While such a triage encompasses many factors of the patients, it does not perform a careful analysis of the actual visual cervix, which is the focus of this project. Neural networks have already been applied to some success in the detection of cancerous cells; in [2] they were used to detect cancer at a cellular level, while [3] focused more on detecting cancer from larger images containing both tissue and cancerous growths. However, the classification of (usually healthy) cervix types has not been performed to my knowledge.

This problem and its data has been made public through a Kaggle competition. Intel has partnered with MobileODT to provide computational tools and resources in order to aid exploration of deep learning methods.

Problem Statement

The principle task in this project is the classification of cervices based on images. Each cervix can be classified as one of three types, which differ by the location of the "transformation zone"[1]:

- Type 1 has an entirely external and visible transformation zone;
- Type 2 has parts of the transformation zone which are endocervical, i.e. in the interior part of the cervix, but through inspection are declared fully visible;
- Type 3 has parts of the transformation zone which are endocervical and are not visible upon inspection. These cases especially require more advanced diagnostic screening.

The classification task is supervised: human labels have been assigned to each of the images. The machine learning model designed during the course of this project should *assign probabilities* to each of the three types and compared to the true labels of the images. Model evaluation is described below.

Metrics

The evaluation metric is dictated by Kaggle: the score is determined by the average log-loss of the probabilities given to each label, i.e. for each image the log-loss ℓ is

$$\ell = \sum_{j=1}^{3} y_j \ln(p_j)$$

where y_j equals 1 for the correct Type j and is zero otherwise, and p_j is the probability we assign to each cervix Type. The total score will be the average log-loss ℓ_i on all testing-set images i:

$$\text{Loss} = -\frac{1}{N} \sum_{i} \ell_i \quad .$$

Furthermore, all probabilities smaller than 10^{-15} are rounded up to 10^{-15}; this is also done for probabilities closer to 1 than 10^{-15}, which are rounded down to $1 - 10^{-15}$.

This metric is also known as the *cross-entropy*, and provides a good information-theoretic metric on the quantity of information shared between the probabilities assigned by the model and the true labels.

[1] The transformation zone is the area of the cervix where columnar cells, which are located more interior in the cervix, have over time converted to squamous cells, located in the exterior regions of the cervix. The transformation zone is particularly prone to cancer.

Note. In this project, if we were to assign equal probability $\frac{1}{3}$ to all cervix types, and the test set is balanced among the various categories, we would expect an average log-loss of $\ln(\frac{1}{3}) \approx 1.099$. This will be called an *agnostic probability* assignment. In the event of no information on the training set, the log-loss favors the agnostic probability to other assignment choices; this has to do with the asymmetry of the logarithm, which near zero quickly grows to negative infinity. For example, assigning probabilities $p_1 = 0.4$, $p_2 = 0.3$, $p_3 = 0.3$ (where p_i is the probability we assign to Type i) to all test-set examples will perform worse; this is because on Type 1 examples we get a slightly higher reward when correct, since $\ln(0.4) > \ln(\frac{1}{3})$, but on the aggregate of Type 2 and Type 3 examples we get a considerably larger punishment. This has an important consequence: even in the event of better-than-agnostic probability assignments, if the prediction is overconfident we are likely to get punished more severely for the overconfidence than rewarded for a better accuracy. Therefore, we must take precautions to not make overconfident predictions.

2 Analysis

Data Exploration

The data is provided by Kaggle [4], which in turn was provided by MobileODT; it consists of 35 GB of color-image files for training. These are divided into two parts: primary images for training, which have been carefully selected, and additional images, which are near-duplicates of the primary images or of low quality. The near-duplicates are photos that were often taken during the same medical session but were not chosen as the best photo of the cervix. The primary images are comprised of 251 Type 1, 782 Type 2 and 451 Type 3 cervices. The additional images are comprised of 1 191 Type 1, 3 567 Type 2 and 1 976 Type 3 cervices. The test set contains 512 images to classify. The image resolution varies, but most images are around 3000 pixels tall and 2000 pixels wide.

There are a number of data-quality concerns:

- Duplicate images have been found [5][2], some of which appear with differing labels. This puts into doubt the accuracy of the human labels assigned to the images.
- The photographed area of the cervix is often out of focus; an example of this is seen in Figure 1.
- The portion of the image constituting the cervix varies widely, from filling the entire image to only a small circular portion (in images which have not cropped away the inspection instruments and similar irrelevant portions of the image). Figure 2 exemplifies some of the diversity of zoom used in the images.

[2] These have been highlighted in Kaggle "Kernels", i.e. code-sharing and discussion platforms on the Kaggle website.

Figure 1. Some images are out of focus.

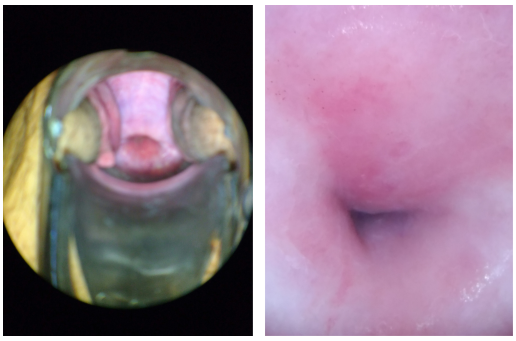

Figure 2. The images use widely different ranges of optical zoom.

- Some images in the additional datasets are not of cervices [6]; there is a Motorola logo, a hand, plastic instruments, a face, and various white fabrics. These images form an extremely small portion of all images however, and are not known to occur in the primary training set.
- Early versions of the data were claimed to contain test-set images which also appeared in the training data. There is a new version of the data which may have addressed this issue.

Despite the rather large number of data-quality concerns, the vast majority of the images are clear images which can be used effectively for training; the contamination of the datasets is small.

While the data provided by Kaggle cannot be made public, any model constructed during this project should be distributed with the MIT license.

Exploratory Visualization

Cervices are typically colored white or pink, unless they've been daubed with specific dyes to highlight various features. Only a small number of images have received this color treatment. The visible columnar cells inside the cervix, on the other hand, are bright red. It is therefore reasonable to explore the color space of the images of the three types to identify any potential differences. Figure 3 shows the RGB pixel-color distribution for Type 1, Type 2 and Type 3 cervix images, where each image has been collapsed to its average RGB colors. From the

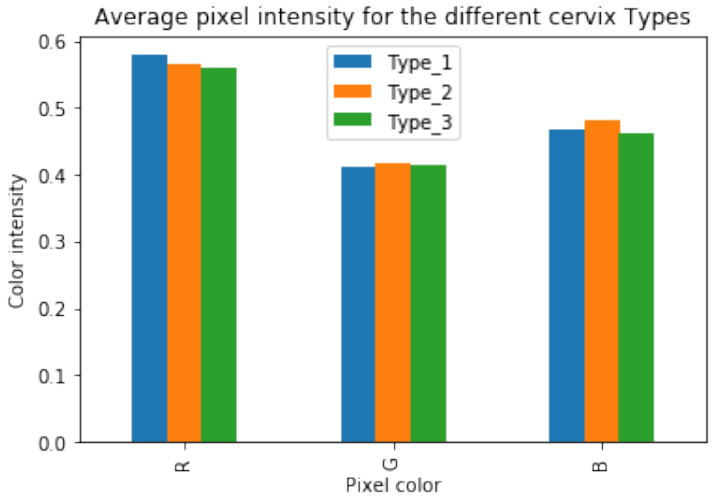

Figure 3.

image it is clear that while the differences on the mean RGBs are only slight, Type 1 cervices tend to have slightly higher values of red in them. It is interesting to plot the average number of "bright red pixels" in each image, whose RGB values are constrained to have at least the value 160 on the R channel and at most the value 60 on the G and B channels. Figure 4 illustrates the average percentage of bright-red pixels for images of each cervix type. Is is clear that Type 1 cervices have many more red pixels than Type 2 and Type 3 images do.

Finally, it is also helpful to simultaneously consider all three RGB colors. For this, I plot each image in the 3-dimensional RGB space, where each image is collapsed to its mean RGB values. Figure 5 illustrates this scatterplot, where the three types of cervix-images are labeled with separate colors. As is clear from the scatterplot, there is no particular clustering around RGB values for the various cervix types; as already intuited from Figure 3, the average colors of the different types of images are very similar to each other. It is still interesting to note, however, that the variance of the images is rather large, especially for Type 1 image, which is consistent with the fact that these are the only ones which sometimes contain a high proportion of red pixels.

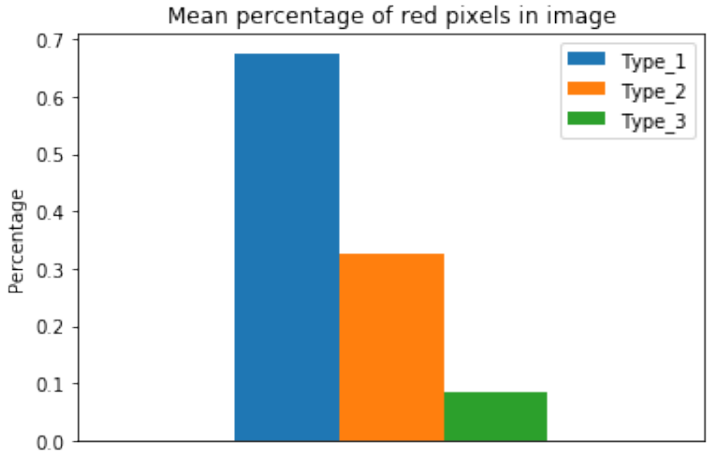

Figure 4. Type 1 has a higher proportion of bright red pixels than Type 2 or Type 3.

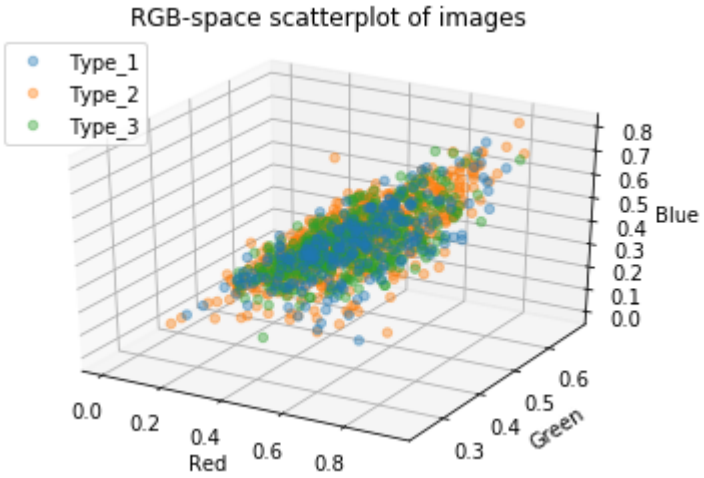

Figure 5. Each image is collapsed to its mean RGB values, which are then plotted in RGB space.

Algorithms and Techniques

The strategy adopted in this project is to use convolutional neural networks, trained on the images of cervices taken by medical professionals. Since it is the probabilities we are chiefly interested in obtaining correctly, neural networks are well suited as their output is easily translated into probabilities by using the softmax function. *Convolutional* neural networks are well suited to this problem, since they are translation invariant and are particularly apt at capturing a very wide range of different features in the data. They require, however, a large amount of data to be trained adequately. The primary training data provided constitutes a little over 1000 images, which is a somewhat low but feasible quantity of data to train the

neural network. The additional data largely contains images which are nearly duplicates of images in the primary training set, and is not expected to significantly improve the model performance.

Cervix images contain a large amount of information that is not helpful for identifying the cervix type; for example, often a large portion of the image is dominated by the speculum used to visualize the cervix, as is also seen in Figure 2. This project will therefore attempt to create a *diagrammatic representation* of the cervix image, to simplify the training of the neural network.

Benchmark

Currently we are not provided any solutions already in place by MobileODT. Hence, there is no method currently available which performs the task of classifying cervix types. The benchmark model chosen in this project will be a random forest classifier trained on the percentage of bright-red pixels in each image. As was shown in Figure 4, red pixels are much more prevalent in Type 1 and Type 2 images; the random forest classifier should be able to make a moderately decent prediction based on this. In particular, this should outperform any model that does not have a large amount of translational invariance, such as SVM used when classifying the MNIST dataset (which is particularly simple due to the numbers appearing in the center of each image).[3]

3 Methodology

Data Preprocessing

I will consider as training data the *primary* set of images, without inclusion of the additional images. This is because the additional images are nearly duplicates of the primary training set and can only provide limited benefit; they are also often of poor quality. Limiting ourselves to this set automatically filters out those images that are not cervices and most of those images that are out of focus. However, it is possible to expand the training set by considering mirrored versions of each image. In this project, each image will be flipped left-to-right as well as upside-down, thus effectively quadrupling the number of training images. The mirroring will be done on-the-fly, batch by batch, as described in the Implementation section below.

While there is not much that can be done about duplicate images with differing labels, it is possible to address the issue of having a widely different angle of zoom. Our preprocessing will constitute of three stages:

- Turn the original images into a common size. After manually inspecting sample images, it seems that turning images into the size of 150×150 pixels is suitable for vastly reducing the image sizes while keeping the visually important features of each image intact. I will get a value for the performance of a simple convolutional neural network trained on these images.

[3] Moreover, SVM does not yield required probabilities, which random forests can provide.

- Automatically crop the images to only contain the cervical component of the image, thus removing as much of the instruments as possible. For this task there is a pre-created python script available [7]. When making square images from rectangular ones, the images can get distorted. Therefore, the cropping script is combined with an additional set of black bands on the images, which turn it square. The images can then be resized without creating distortions. I will then run the same convolutional neural network on this dataset and note down the performance improvement, if any.

- Much of the cervix image is not necessary for classifying its Type. In fact, it is primarily the size and shape of the central part of the cervix which plays the most important role in performing the classification. Therefore, each cropped image is turned into a *diagram*. This will be achieved in a two-stage process. First, the colors of the image are quantized into two, by clustering the pixels in RGB space with two clusters, and assigning each pixel to its cluster-mean[4]. This will yield a light region centrally located in each image, and a dark region in the periphery of each image. The dark region will be used as a *mask* on the original photograph, effectively eliminating any instruments or areas of the photograph that are not the cervix. After applying the mask, each photograph will be identical to its original state, but with large portions colored over in black. If the colors are now quantized into five separate groups using the same clustering technique[5], the resulting image will look much more like a *diagram* of a cervix. Since diagrams are much simpler objects, they can be resized to a smaller size; visual inspection suggests 80×80 pixels is appropriate for keeping the diagrammatic properties of the images intact, as shown in Figure 6. A neural network is then trained on the diagram.

Implementation

The project involves an element of exploration to find the most suitable architecture for the neural network[6], as well as its combination with a good set of preprocessed images. In order to facilitate this exploration, it was critical to set up a work-flow pipeline that would allow minor modifications to be tested and re-run without requiring many changes in the code. To this end, this project was set up by creating a number of python modules for the various tasks, with the aim to eliminate any duplicated code and keep a clean logical flow, from preprocessing to generating probabilities for the images. Moreover, this approach allows us to make frequent Kaggle submissions with incremental improvements to the model, effectively adopting an AGILE-like approach to the work-flow. The various modules are controlled by a single IPython notebook, which clearly outlines the entire work-flow from image to submission.

[4]The code to achieve this was completed by a Kaggle-competition collaborator, Sam Playle, who in turn heavily based his code on [8].

[5]Note that one of the cluster-centers will sit on the large number of black pixels, so this step is effectively only breaking up the colors into four separate groups.

[6]Since the final assessment metric is the log-loss, and neural networks are usually optimized with precisely this in mind, the evaluation metric needs no refinement.

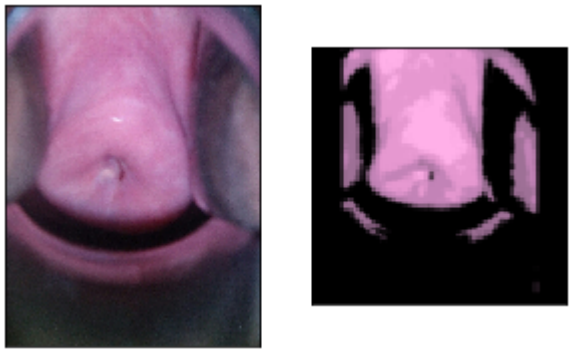

Figure 6. Original image and its diagrammatic version, complete with black bands on the left-and-right sides in order to make it square while maintaining the image's aspect ratio. The image has been resized to 80×80 pixels, which as shown preserved the important features of the image, such as the black central area at the opening of the cervix.

Preprocessing. The first step was to create the three sets of preprocessed images as described in the previous section: the original photographs were resized to 150×150 pixels, the cropped images were turned square and also resized to 150×150 pixels, and the diagrams of each image were resized to 80×80 pixels. To facilitate this process, a DataPreprocessor class was created, which in turn makes use of additional python modules for achieving the required tasks.

Benchmark. Subsequently, the benchmark model was set up and its performance was evaluated. This was done using a BenchmarkModel class. In this specific case, the benchmark random forest did not beat the agnostic probabilities, mostly due to the model's overconfidence in assigning probabilities (cf. the remarks in the Metrics section). This is true regardless of which preprocessing was performed on the images. Therefore, instead of setting the benchmark to the score of agnostic probabilities, i.e. a loss of 1.099, I set it to Kaggle's benchmark score, i.e. a score of **1.00575**.

Neural Networks on Images. The convolutional neural networks creation, training and testing is completely performed using a ConvNet class. The iterative work-flow was the following:

- Set up a simple LeNet-like convolutional neural network [9] to train on images resized to 150×150 pixels. The chosen network architecture was composed of a convolutional layer with filter size 4×4 and step-size 1 ("same padding"), followed by a 2×2 max pooling with step-size 2. The output channels of this layer are 20. This is followed by another convolutional layer of identical size but with 40 output channels, and the same size of max pooling. Finally, the 3-dimensional image array is flattened to one dimension and applied to two fully-connected layers with sizes of 100 neurons and 30 neurons, respectively. All layers use the ReLU activation function and a dropout rate

of 0.5. The last step is the output layer, which generates the logits. The logits are then converted into probabilities with the softmax function.

- The training data was broken up into batches, where each batch was at most of size 256.
- The network described above was then also trained on cropped images, also of size 150×150 pixels (see preprocessing section above).
- The same network architecture was then trained on diagrams of cervices of size 80× 80 pixels.
- For each of the above training runs, the first batch was the validation batch and the remaining batches were used for training. The learning rate was set to 0.001 for the images with no cropping; for the cropped images and the diagrams it was set to 0.01 for the first 70 epochs, followed by 0.001 for the subsequent 100 epochs. The training happened on minibatches of size 64. In each batch (not minibatch), all images were additionally flipped upside-down and left-to-right, effectively quadrupling the number of training images. The images were then *resampled* so that there were equal numbers of images from each cervix type; this was done by repeating multiple times images of infrequently-occurring Types until there were as many of them as the most frequently-occurring type. The training was performed for as many epochs as was necessary to show clear signs of overfitting. This was 100 epochs, 170 epochs, and 170 epochs when training on the full images, the cropped images, and diagrams, respectively. An example of a training run is illustrated in Figure 7, which shows that up to the 70 epoch the validation-set loss decreases, and beyond that it is only the training set loss which decreases, while the validation set loss increases. 70 epochs was also the ideal training stopping point for the cropped images and the diagrams.
- The best-performing of the above training runs was then run on the test set to generate probabilities to be submitted to Kaggle.

Refinement

The LeNet-like neural network above has a huge number of hyper-parameters which determine its architecture. Here I discuss some of the most important hyper-parameters, why they were chosen to their final values, and which exploration, if any, they were subjected to.

As already explained in the previous section, the learning rate was initially set to a large value and decreased when reaching a certain number of epochs: for cropped images and diagrams, the first 70 epochs had a learning rate of 0.01 and the subsequent 100 epochs had a learning rate of 0.001.[7]. The following hyper-parameters were not tuned, except in individual practice runs:

[7]The learning rate 0.001 is also the default value in TensorFlow.

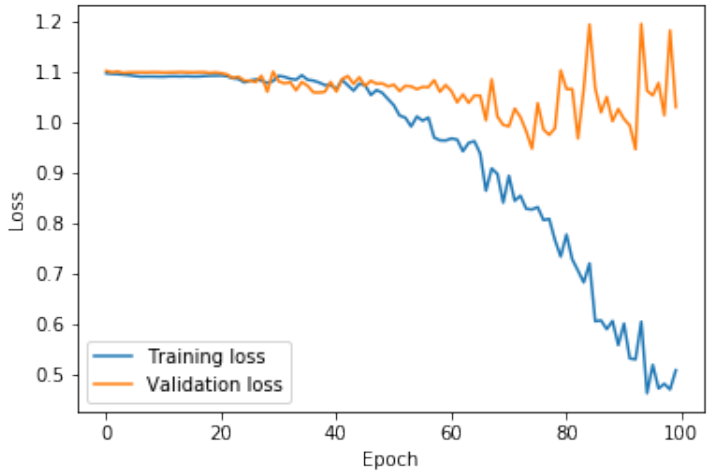

Figure 7. Training on non-cropped images resized to 150×150 pixels, with learning rate 0.001 and minibatch size 64. At around 70 epochs signs of overfitting become manifest.

- The size of the minibatches was fixed to 64. This is because the minibatch size is related to the learning rate and number of epochs required to train, and this was found to be a reasonable (and commonly-chosen) value to train on.
- All layers had ReLU activation functions, which is known to be efficient for learning, due to it not suffering from vanishing-gradient problems.[8]
- Training was done using TensorFlow's AdamOptimizer, set to minimize the log-loss described in the Metrics section, above.

The network architecture, on the other hand, was tuned to obtain decent predictions. Different sizes for the convolutional layers as well as the fully-connected layers were tried, as well as making use of convolutional layers with filter-size of 1×1. Denoting [20, 60] as two consecutive convolutional layers with 20 and 40 output channels, respectively, and {100, 30} as two fully-connected layers with 100 neurons and 30 neurons, respectively, the following architectures were attempted on the best-performing of the preprocessing types:

- $[20, 40] \times \{100, 30\}$. This is the original architecture described in the previous section. This was trained using a dropout "keep-probability value" of 0.5, which is a commonly-chosen and recommended value [10].
- $[10, 20] \times \{100, 30\}$. This was also trained using a dropout of 0.5
- $[20, 40] \times \{200, 60\}$. This was trained using a dropout of 0.75 to ensure convergence in a reasonable time frame. This dropout values means that 25% of the neurons were turned off in each training iteration.

[8] It can, however, irreversibly turn off neurons.

- $[10, 20] \times \{200, 60\}$. This was also trained using a dropout of 0.75.
- $[5, 2, 20, 40] \times \{200, 60\}$, where the first two convolutional layers are of size 1×1 (without max pooling) and the last two are of size 4×4 (with max pooling as described above). The details and motivation for these additional layers is below. Here again a dropout of 0.75 was used.

Out of the above, the original architecture [20, 40] × {100, 30} and the slightly larger architecture [20, 40] × {200, 60} seemed to give the best validation-set results.

As mentioned above, convolutional layers with filter-size 1×1 (and step-size 1) were also explored. The rationale for this is that not all colors are equally important; columnar cells are typically bright red (except on night-vision photographs), and squamous cells a very uniform pink. Being able to detect where in the image these two types are is very important. Therefore, as a method to tease out the relevant information of each pixel, a 1×1 convolution was also performed on the color space, with the hope of turning off pixels which do not aid in the classification. The convolution which was tested was set up with size 1×1 and 5 separate channels (and step size of 1), followed by another 1×1 convolution which transforms the 5 channels into 2 channels. After this, the architecture [20, 40] × {200, 60} was implemented. Unfortunately, the results were no better using the 1×1 convolutional layers compared to simply using the [20, 40] × {200, 60} architecture directly.

4 Results

Model Evaluation and Validation

The final model was chosen based on the validation-set scores and the Kaggle submission scores. Rather surprisingly, the best scores were achieved when training on the full (resized) images, i.e. those images that were not cropped and not turned into diagrams. This shows that the additional preprocessing was inadequate for the task at hand, and washed out relevant information used for training, or complicated the training process.[9]

The scores for the various models were as follows:

- $[20, 40] \times \{100, 30\}$: validation score 0.94577, submission score 0.99060.
- $[10, 20] \times \{100, 30\}$: validation score 0.99214, submission score 1.00499.
- $[20, 40] \times \{200, 60\}$: validation score **0.90432**, submission score **0.99195**.
- $[10, 20] \times \{200, 60\}$: validation score 0.92112, submission score 1.00052.

[9] It is also important to consider the possibility that the training set and validation sets may be correlated, i.e. that there ma be systematic differences between the photographs taken on Type 1 images to those taken on Type 2 and Type 3, such that the neural network is learning these systematic differences rather than training on the relevant details of the images. Confirming whether this is the case, however, is beyond the scope of this project.

- $[5, 2, 20, 40] \times \{200, 60\}$: validation score 0.90288, submission score 1.01872. (The first two convolutional layers here were of size 1×1).

This shows that the slightly larger models are more suitable for this sort of task, but that the 1×1 convolutions seem to not help, at least given the amount of available training data. In order to check whether the complexity of the model was saturated by the amount of data, it is useful to check how the performance changes with increasing amounts of training data. For the model $[20, 40] \times \{200, 60\}$, this is plotted in Figure 8, where it is clear that the best validation loss continues to decrease with increasing amounts of data. Hence, we have not saturated the amount of complexity which can be captured by the model.

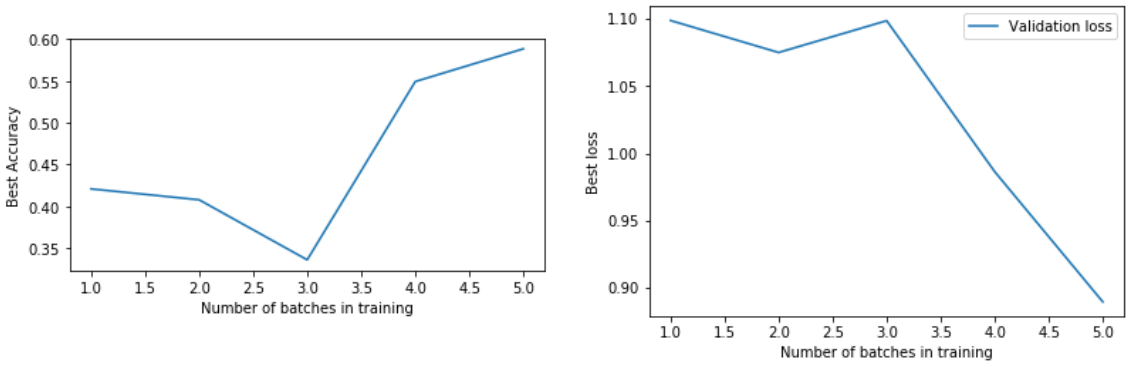

Figure 8. Validation-set accuracy and log-loss as a function of the number of training batches available when training. As more data is made available, the model accuracy improves while the log-loss decreases, showing that the complexity which can be captured by the model is not saturated by the training data.

It is interesting to note that the test-set results are generally inferior to the validation-set results. This hints at the possibility of a systematic difference between the training photographs and the testing photographs. It is also possible that the performance of the model is very sensitive to the precise epoch we stop the training at. The difference between validation-set scores and testing-set scores implies that results given by the model should be interpreted with caution.

Justification

The benchmark score that was chosen is a log-loss value of 1.00575. The best validation-set score obtained by our model is 0.90432. While it is definitely an improvement, and shows that the model captures non-trivial features of the data, it is not a sufficiently large improvement to consider the problem solved. Especially in the field of cancer screening, high accuracy is extremely important as mistakes can be highly punitive. The accuracy obtained by the model

is simply not good enough to be considered automated. However, it can be useful as an aid to flag up unusual diagnoses.

The differences between Type 1 and Type 2 are often very subtle and the line between them is blurred; even medical professionals often disagree on the correct labeling. However, between Type 1 and Type 3 there is a significant difference, Type 3 being the most dangerous; if a healthcare professional determines a cervix to be Type 1 while the model assigns a high probability to Type 3, this should be flagged as a case requiring a third opinion.

5 Conclusion

Free-Form Visualization

The problem of classifying cervices based on images of varying quality was solved using convolutional neural networks. Figure 9 illustrates the process of starting from images representing cervices of Type 1, Type 2 or Type 3, and arriving at the desired probabilities for each cervix type. The figure also displays the model's predictions on a sample of test-set images.

The particular challenge of this problem was to find a suitable architecture that would perform a decent classification, by tuning the number of convolutional and fully-connected layers, and choosing the size of the convolutional and max-pooling filters as well as the size of the fully-connected layers. These choices are critical to the success of the classification.

As we can see in Figure 9, the chosen model assigns probabilities decently to the test-set images. Those images with a clear transformation zone, i.e. the zone around the entrance to the cervix which has a clearly different color to the rest of the cervix, are assigned probabilities more weighted towards Type 1 and Type 2 cervices; those cervices apparently missing a visible transformation zone, i.e. appearing as a homogeneous color up to and including the entrance to the cervix, have higher probabilities assigned to the Type 3 label.

Reflection

The project's goal is to classify cervices based on images taken during cervical cancer screenings, often in rural settings. This classification is very important to determine further treatment options; if incorrect treatments are applied, they can dangerously hide future cancerous growth in the cervix. Since the ability to distinguish between the cervix types and their correct treatments is often lacking, in particular in rural communities, MobileODT provides a service that supports healthcare professionals in this classification. The aim is to automate or semi-automate the supporting role offered by MobileODT.

The problem had several challenges. The first, and possibly most important, involves data quality. The images are provided by Kaggle, and are split into training images, testing images, and additional labeled images which are near-duplicates of the training images. These additional images are often very blurry, do not capture the cervix at all, or are fundamentally identical to other images already in the training set. For these reasons, the additional images were not used for training in this project. Even in the training set, however, there were several

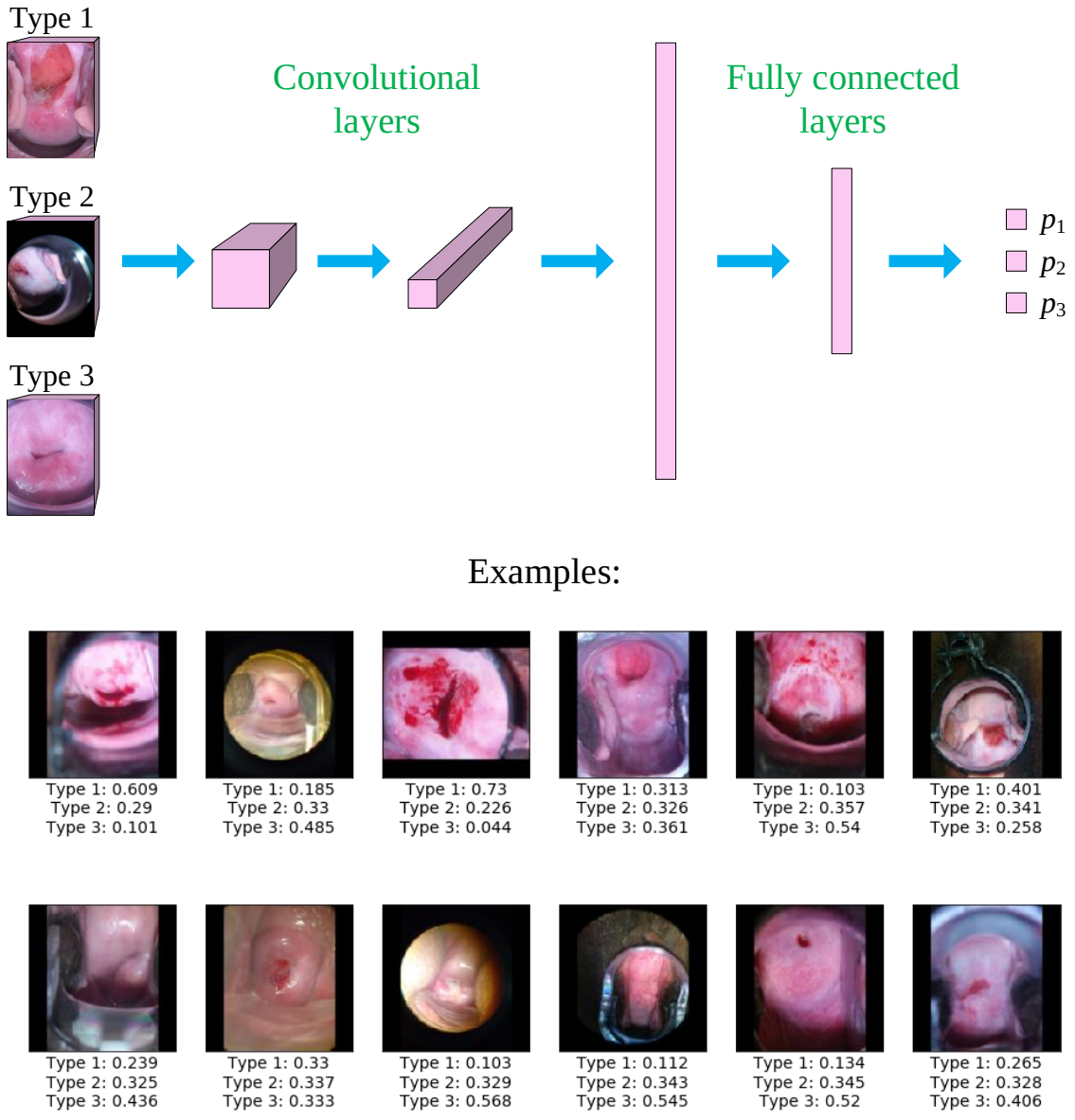

Figure 9. Schematic procedure for classifying images of cervices into three types. We begin with two convolutional layers, followed by two fully-connected layers which decrease in size, to obtain a probability for each cervix type. Test-set examples are shown, where it is clear that images with a very visible transformation zone have probabilities more weighted towards Type 1 and Type 2, whereas probabilities for images with no visible transformation zone are more weighted towards Type 3.

complications; this set also contains many images that are out-of-focus, and different images

use widely differing levels of zoom, from photographing only the cervix to capturing large portions of instruments and unrelated anatomy. Moreover, the supervised labels given to the training images appear in cases to be inconsistent, presumably due to differing opinions on the correct labeling assigned by human inspection.

The first step in the project explored the color-space of the images. It was particularly challenging to find notable differences between images of the three types of cervices. In the end, the clearest difference was in the percentage of red pixels present in each image, which was largest for Type 1 cervices and smallest for Type 3 cervices.

The next step was to preprocess the images, in order to facilitate training. Since it was unknown how much preprocessing was ideal, this step was divided into several stages. First, the images were normalized to a common size. Secondly, they were cropped automatically, centered around the area likeliest to constitute the center of the cervix. Last, the images were turned into diagrams, by first blackening out large portions of the images that were unlikely to depict the cervix, followed by discretizing the color space of the remaining image.

The resulting images and diagrams were fed to a convolutional neural network with 2 convolutional layers and 2 fully-connected layers. The precise size of these networks was explored, demonstrating that while slightly larger networks achieved better results than smaller networks, 1×1 convolutional layers did not improve the results. Ultimately, the best network among those explored was found by starting with an image resizing to 150×150 pixels, applying a 4×4 convolution (with "same padding") with 20 color channels followed by a 2×2 max pooling, applying another similar convolution and max pooling but now with 40 color channels, flattening the image to one dimension, applying a fully-connected layer of size 200, applying another fully-connected layer of size 60, and completing the task with an output layer consisting of a softmax function giving three outputs. Each output was then interpreted as the probability for Types 1, 2 and 3.

A surprising discovery of this project was that training on cropped images and diagrammatic versions of the images (i.e. those which had been color-discretized) had a much worse validation-set performance than training directly onto the original resized images. This may be because the cropping is automated, and only adequately crops certain images, effectively splitting the training set into images only containing the cervix and images where the cropping failed and hence contained non-cervix portions. This split may have created a less homogeneous training set than the original training set. Furthermore, the color discretization may have obscured important features which are useful for the neural network to determine the cervix type.

Crucial to the completion of this project was setting up a work-flow which facilitated exploration of different types of preprocessing and different neural network architectures. To this end, a large amount of code was set up in order to allow the user, through a simple interface in a jupyter notebook, to flexibly tune the various parameters subject to exploration. Very little was hard-coded and custom made; the code thus created can easily be applied to other, possibly very different, image-classification tasks.

The final result obtained by the chosen preprocessing and neural network architecture

was a log-loss score of 0.90432 on the validation set and 0.99195 on the test set, as verified by a Kaggle submission.[10] This result is an improvement over the benchmark score of 1.00575, but is not vastly better. The small amount of data, approximately 1 000 images, combined with its variability and poor quality posed a huge challenge when training the neural network. Therefore, the model created in this project should not be used to completely automate the classification of cervices. It is instead advisable that it be used as a confirmation, or guide, for a healthcare professional when making human classifications. In particular, it can be of particular help when two human inspections result in different classifications, where the model can arbitrate between the two. Additionally, the model chosen in this project can serve as a benchmark for the incremental improvement by next-generation classification models of cervical images.

Improvement

The task in this project was particularly challenging, and a large portion of the time devoted to the project was spent setting up an automated work-flow and exploring possibilities, in order to tease out differences between the images depicting the three different cervix types. There is a significant amount of improvement that can still be achieved, but which fell beyond the scope of this project:

- The neural network architectures which were explored were nearly all similar to a LeNet network, with simple convolutional layers and max pooling followed by fully-connected networks. It would be very interesting to explore fundamentally different architectures, such as the Inception architecture [11].

- The size of the neural networks chosen in this project is small compared to many state-of-the-art neural networks; the primary reason for this was a lack of high-end computing resources: the neural networks were trained on a desktop with 8-threads of 3.6 GHz CPUs and 16 GB of RAM. Ideally the network would be trained on a GPU with 16 GB of memory, as accessed by AWS services. The limited budget for the project placed AWS GPUs out of reach for long training sessions on large and complex architectures, and thus went unexplored.

- The preprocessing used to crop images and turn them into diagrams was largely unfruitful. It is very likely that there are other preprocessing methods which are more suitable to this classification task; such exploration is likely to yield vast improvements on the classification scores, since it could effectively filter out much of the difficulties posed by the variability and quality of the training data.

- It is possible to obtain probabilities using very different methods, e.g. by using convolutional networks while separately counting the number of red pixels, as done in the exploratory analysis. When fundamentally different methods are simultaneously used,

[10] I remind the reader that log-loss scores are to be *minimized*.

they can then be *combined* to provide a more accurate and robust probability assignation to the images. There are various methods for combining the probabilities; for example, they could simply be averaged; another way of combining them could be to multiply the probabilities for each classification type (followed by an overall normalization to ensure that they sum to 1). The second method yields more confident probability assignations when two methods independently agree.

- The log-loss scoring used in this project was dictated by Kaggle, and is what the neural network was trained on. However, mistaking a Type 1 cervix for a Type 3 cervix is much less severe than the converse. It would be useful to use an asymmetric loss score, which punishes mistaking Type 3 images more severely. Such a scoring is complex and would need to take into account factors such as the risk posed to the patient by a misclassification, and the additional medical cost of remedying such a mistake.

As already mentioned in the previous section, the model developed in this project can serve as a useful benchmark for further exploration and evolution of automated tools for the classification needed in cervical cancer screening.

References

[1] S. Sen, M. Horta Ribeiro, R. C. de Melo Minardi, W. Meira, Jr. and M. Nigard, *Portinari: A Data Exploration Tool to Personalize Cervical Cancer Screening*, *ArXiv e-prints* (Apr., 2017) , [1704.00172].

[2] M. A. Devi, S. Ravi, J. Vaishnavi and S. Punitha, *Classification of cervical cancer using artificial neural networks*, *Procedia Computer Science* **89** (2016) 465 – 472.

[3] K. J. Geras, S. Wolfson, S. G. Kim, L. Moy and K. Cho, *High-Resolution Breast Cancer Screening with Multi-View Deep Convolutional Neural Networks*, *ArXiv e-prints* (Mar., 2017) , [1703.07047].

[4] Mobile and ODT. https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening/data.

[5] Kaggle user "amaia". https://www.kaggle.com/aamaia/leeak.

[6] Kaggle user "thinkski". https://www.kaggle.com/chiszpanski/non-cervix-images.

[7] Kaggle user "chattob". https://www.kaggle.com/chattob/cervix-segmentation-gmm.

[8] Scikit-learn documentation pages. http://scikit-learn.org/stable/auto_examples/cluster/plot_color_quantization.html#sphx-glr-auto-examples-cluster-plot-color-quantization-py.

[9] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, *Gradient-based learning applied to document recognition*, *Proceedings of the IEEE* **86** (November, 1998) 2278–2324.

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, *Dropout: A simple way to prevent neural networks from overfitting*, *Journal of Machine Learning Research* **15** (2014) 1929–1958.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov et al., *Going Deeper with Convolutions*, *ArXiv e-prints* (Sept., 2014) , [1409.4842].