Active Learning Model for Nucleus Classification

Sam Vanmassenhove
Ghent University, Faculty of Engineering and Architecture
May 2018

Supervisors: Dr. Saskia Lippens, Prof. dr. ir. Wilfried Philips
Counsellors: Dr. ir. Jan Aelterman, Dr. ir. Evelien Van Hamme, Joris Roels

***Abstract* – Manual annotation is still quite common in biology for both detection and classification problems. Often classification problems can be solved with machine learning techniques, but these typically require hundreds or even thousands of training samples to work effectively. For this reason such methods are not commonly used in practice.**

In this thesis we propose an active learning approach which drastically reduces the number of labelled training samples required to perform machine learning.

Experiments on real-world data show that the sampling effort can be reduced by a factor 6 in some cases. This makes machine learning viable where it would otherwise not be used in practice.

***Keywords* – active learning, nucleus classification, microscopy**

I. Introduction

A. Problem

Researchers at the Flanders Institute for Biotechnology (VIB) work on many different classification problems of cells and nuclei in biological microscopy images. These problems can be solved with machine learning methods which are available in most commercial software packages.

These machine learning methods are often not applicable in practice because they require large quantities of labelled samples. This manual effort by researchers is often unacceptable for uncommon cases where the labelled dataset cannot be reused later, so in many cases machine learning is not applied at all and the annotations are done completely manually.

In this thesis we propose active learning as a solution that drastically reduces the amount of samples required for machine learning.

The structure of this paper is as follows. First a general overview of the proposed framework is given and each step explained, after which the different methods of active learning are discussed. Finally, two real-world applications are laid out and the results of the experiments discussed.

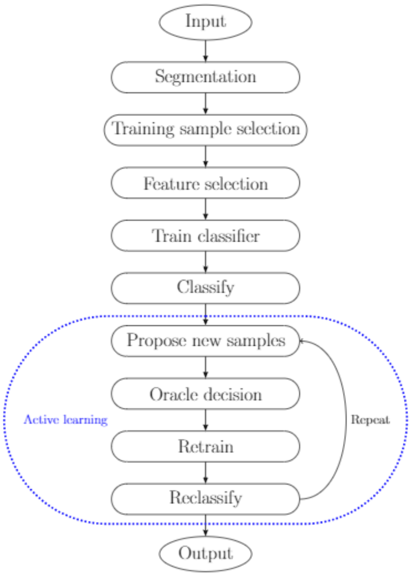


Figure 1 - Active learning workflow. The feedback loop is marked in blue.

B. Software packages

There are several software packages available for the analysis of biological microscopy images. The most popular include ImageJ and its extended version Fiji, but also CellProfiler and QuPath. The method described in this paper was implemented as a QuPath extension. The source code for this extension is available at https://github.com/savmasse/qupath-extension-tunel.

The main advantage of QuPath compared to the other packages described here is that it was made to handle large datasets and allows the user to view and process whole slide images without having to chop them into smaller pieces. While QuPath contains functionality for both brightfield and fluorescence images, it does not support analysis of 3D or 4D (3D + time) images.

II. Proposed Framework

A. Framework overview

The classification of cells or nuclei in a microscopy image is dependent on much more than the classifier itself. The complete proposed workflow is illustrated in **Figure 1**. The first step in the diagram is the segmentation of nuclei from the image. This is followed by the selection of the initial training samples and the relevant features by the user. These steps will be further explained below.

B. Segmentation

Before classification can be applied the nuclei must first be segmented and separated from each other. There are several existing techniques for this, of which the *watershed transform* is the most commonly used for biological images.

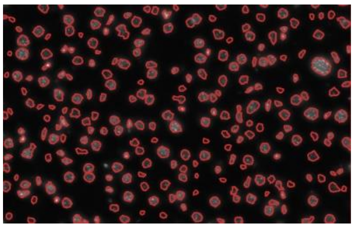


Figure 2 - Watershed segmentation in QuPath on a fluorescence image.

The watershed technique is modelled on the natural phenomenon of several bordering basins of water. The edges separating the bodies of water can be found by flooding each basin from the bottom. The place where the rising waters of two different basins meet will be the edge. Watershed segmentation tends to oversegment – it divides the image into more segments than are actually present – but this can be combatted by carefully selecting the points from which the water starts flowing, and filtering out the segments which are too small to be nuclei. See **Figure 2** for an example of a segmentation of a fluorescence image in QuPath.

C. Training sample selection

At this point in the workflow the user still has full control over the selection of training samples; this will not be the case inside the active learning loop. It is their responsibility to provide at least one sample of each class.

D. Feature selection

Another step before the classification is the selection of the correct features for the task. At present the end user is responsible for choosing good features. Popular features in software packages are *shape factors* like the area, perimeter, compactness and eccentricity of a segment, but also intensity features like the mean and extrema of the intensity values. More complex features like edges can be calculated but are unlikely to be used by end users with little knowledge of image processing.

Advances in deep learning have shown that autoencoders (AE) and convolutional neural networks (CNN) can be used to extract features from images [2]. Interestingly, this extraction can be performed fully unsupervised so that no prior manual labelling is required. Although feature extraction using autoencoders is briefly discussed in the full thesis, no conclusion was reached as to its usefulness in this context. Future work could further explore these possibilities, as the automatic extraction of features could make the classifier less dependent on the good judgement of the end user.

E. Classifier

Deep learning methods are becoming increasingly popular for classification tasks. This is especially the case for image classification problems, where CNN's have been shown to outperform all classical machine learning methods [4]. Neural networks do however require much larger annotated datasets than is the case for other techniques, and as such are not appropriate in the context of this thesis.

Random forests (RF) is an ensemble machine learning method consisting of many decision trees, and is the most commonly used in biology. Recently it has been applied to X-ray images. Compared to deep learning RF trains quickly and requires fewer training samples.

III. Active learning

A. Introduction

Active learning (AL) is a branch of machine learning that introduces a feedback loop into the process. After the initial classification step new training samples are added to add information that may be missing. This process of adding new samples, retraining and reclassifying can be repeated until the performance converges or the user (referred to as the '*oracle*') is satisfied with the performance result.

For active learning to be applicable the following requirements must be met:

1. The oracle must always be correct.
2. The oracle must always be available.
3. There is only one oracle.
4. The cost of consulting the oracle is always constant. This cost is usually computational in academic papers but in practical applications this could also be a monetary cost.

B. Sampling

Active learning requires that the next sample is one that provides new information that the classifier is currently lacking. The sampling must happen in an intelligent way so as to avoid redundancy and include outliers in the classifier. Some popular sampling methods are described below.

1. Random sampling

The simplest way of proposing new samples is to serve random samples from the training set. The only added condition is that each sample is unique, and as such has not been proposed in a previous iteration.

2. Least confidence sampling (LC)

Many classifiers will return a measure for the certainty, also called the label confidence, of the prediction. For *random forests* this is a probability based on the amount of votes each class received; for *support vector machines* the measure of confidence is the distance of the sample to the hyper-plane.

In *least confidence* (LC) sampling the next proposed sample will always that sample of which the classifier is least certain. For classification problems where the amount of classes M is greater than two the *cross-entropy* (Equation (1) is commonly used to measure label confidence. Here p_c is the predicted probability that the sample has a class label c, and y_c is the binary indicator which is one if c is the true class label.

$$-\sum_{c=1}^{M} y_c \log p_c \qquad (1)$$

3. Clustered sampling

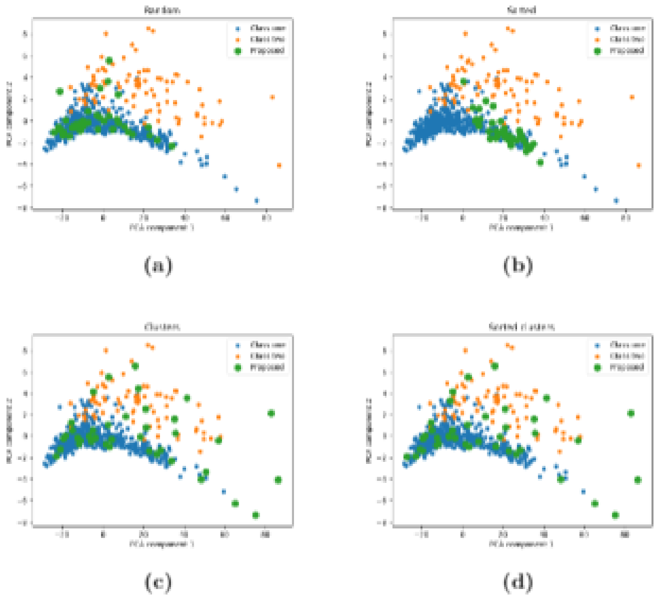
Another method of sampling is to sub-divide the feature space into clusters. This is done to force the exploration of the complete feature space. Clustering ensures that different types of samples are proposed which otherwise may be omitted.

When randomly sampling only the densest areas are well represented and outliers mostly ignored. Conversely, when applying LC only those samples on the edge of the class divide are sampled.

The clustering algorithm used here is the k-means algorithm because this is a commonly used algorithm for the subsampling of a dataset.

4. Least confidence clustered sampling (LCC)

The method used in the implementation is a combination of the two previous methods. Here the feature space is divided into clusters as was done previously, but these clusters will be individually sorted in order of increasing confidence. Then one sample is served per cluster until the final cluster has been reached, at which point all clusters are sorted again and the process is repeated.



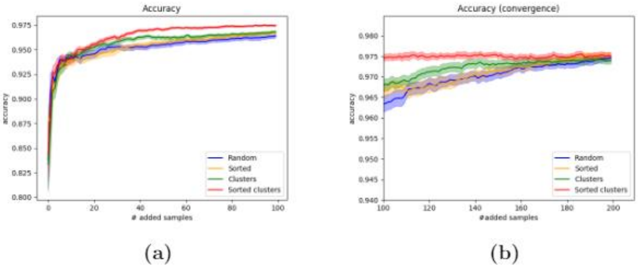
(a)

(b)

(c)

(d)

Figure 3 – Different active learning sampling methods compared for a binary classification problem where the true labels of the two classes are denoted in blue and orange dots, and the 40 first proposed samples ae denoted in green.



(a)

(b)

Figure 4 – Comparison of accuracy performance of the four different sampling methods on a synthetic dataset, where (a) shows the performance for the first 100 samples, and (b) the convergence of all four methods at the end of the training set. The error shown on these graphs is the 95% confidence interval over 100 simulations.

C. *Performance*

Figure 3 shows the first 40 samples for each of the four different methods described above. Random sampling and LC sampling do not explore the whole dataset, but only the densest and areas closest to the class divide respectively. The two clustering methods perform better in this respect, and can be seen to propose samples from across the whole dataset. The difference between the two methods is very minimal in this respect.

It is not enough for active learning to explore the complete feature space. There must be an actual difference in performance. The results in **Figure 4a** show the clustering methods achieving a higher accuracy for fewer samples than the other two methods. It is evident that clustering is greatly improved when sorting by confidence within each cluster. **Figure 4b** shows the point where all four methods eventually converge to the maximum accuracy. The sorted clustering method was used in the QuPath implementation as it has been shown to outperform the others.

IV. Evaluation and Experiments

This section will explain how the experiments are conducted and describes the method of evaluation. The biological context of both use cases will also be addressed here. The results of the experiments are explained in the next section.

A. *Evaluation of classifiers*

Classification problems in the biological and medical sciences are often greatly imbalanced. This means that the classes in the training sets are not equally distributed. It is well understood that classifiers do not perform well under such circumstances, and that the accuracy will not be a reliable measure for performance. For a great imbalance a classifier may well predict everything to be of the majority class and achieve a good accuracy value. Other performance measures which are more meaningful in this case, namely the precision, recall and f-score, are defined in Equations (2-(5).

For these equations the following terms must be defined: *True positives* (TP) are samples which were both annotated and correctly predicted by the classifier as positive, whereas *false positives* (FP) are samples which were annotated as being negative but were falsely predicted to be positive.
Conversely, *true negatives* (TN) are both annotated and correctly predicted as being negative, whereas *false negatives* (FN) were annotated as positive and wrongly predicted to be negative.

$$Accuracy = \frac{TP\ +\ TN}{TP\ +\ FP\ +\ TN\ +\ FN} \tag{2}$$

$$Precision = \frac{TP}{TP\ +\ FP} \tag{3}$$

$$Recall = \frac{TP}{TP\ +\ FN} \tag{4}$$

$$F1 = \frac{2 * Recall * Precision}{Recall + Precision} \tag{5}$$

Both experiments which shall be discussed in this section are performed in the same way. The active learning process is simulated 100 times after which the averages and confidence interval are calculated. Each simulation is performed as follows.

The dataset is split into training samples and testing samples. An initial training set is created by randomly selecting one sample for each class.

After this samples are added one by one while for each step the performance measures are computed. It should be noted that the x-axis on the performance graphs starts at zero 'added samples' but that zero does already include the initial samples.

B. *TUNEL experiment*

Terminal deoxynucleotidyl transferase dUTP nick end labeling (TUNEL) is a technique for the detection of DNA fragmentation in cells [3]. This fragmentation occurs on cell death (*apoptosis*) and can be quantified by adding a chemical marker. This allows for the detection of dead nuclei in a cell population. Another stain (DAPI) is added which lights up for both dead and living nuclei. Each of these stains forms a distinct channel of the fluorescence image.

When deciding whether a nucleus is dead researchers usually focus on the channel with the cell death marker, but there is good reason to look at both channels. As a cell dies the DAPI marker becomes less intense and the shape changes slightly. Therefore, the classifier includes intensity features for both channels and shape factors.

The segmentation for this use-case is often lacking, so that multiple nuclei are grouped together in the same segment. This is combatted here by modelling this as an extra classification problem. As such there are four possible classes for a segment:

- Multiple-Negative: multiple nuclei in the segment, all of which are alive.
- Multiple-Positive: multiple nuclei in the segment, at least one of which is dead.
- Single-Negative: only one living nucleus inside the segment.
- Single-Positive: only one dead nucleus inside the segment.

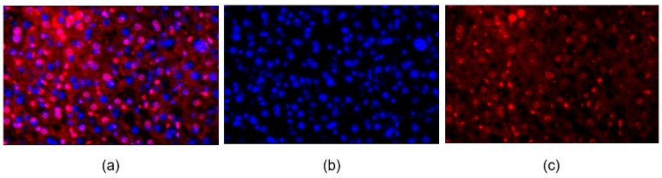


Figure 5 - Example of TUNEL image: (a) combined image; (b) DAPI channel; (c) channel with TUNEL marker.

C. *Microglia classification*

The second use case addressed in this thesis is based on a recently published paper by VIB researchers [1]. During the research specific cells called microglia had to be detected and counted. Microglia are found in the brain and spinal cord where function as immune cells.

Researchers had to resort to manually counting these cells in the tissue because the existing techniques could not be relied upon to correctly classify the microglia from other cell types in the tissue.

The main difficulty here is that there are many cells which have the same dark brown color as the microglia, but lack the characteristic appendages. **Figure 6** shows an example of how the cells should be classified. In this classification microglia are considered *positive* and other cells *negative*. Cells (1) and (2) are microglia with clear appendages, while (3) is not because it is purple and not dark brown. Cell (4) has the correct coloration but lacks the appendages so is also classified as negative.

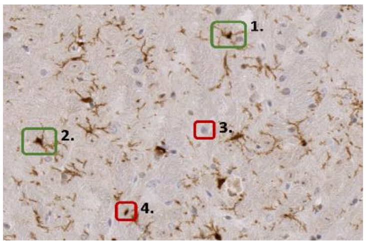


Figure 6 - Brightfield image of brain cells with microglia marked in green and other cells in red.

V. Results

A. *Hypothesis*

After identifying the best active learning method in **Figure 4** this will now be compared against random sampling on real world data. We expect the result to be comparable to the earlier simulation on synthetic data, although the difference may be less pronounced. The active learning should outperform the random sampling and converge faster to the maximum of each performance metric.

B. *TUNEL*

The results of accuracy and precision (other metrics were also computed in the full thesis) are given in **Figure 7.** The performance of LCC is clearly better than random: it takes 82 samples to reach convergence whereas random sampling requires the whole training set (500 samples).

Please note that, since we are dealing with a non-binary classification here, that the metrics displayed here are weighted averages of the metrics of the four separate classes.

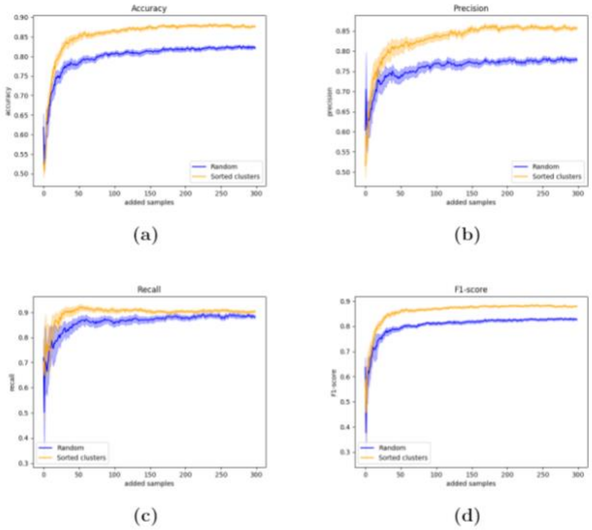
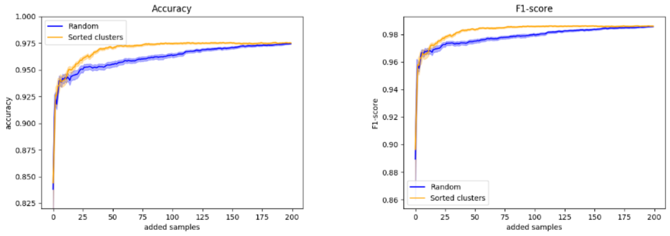


Figure 7 – TUNEL classification performance of LCC vs random sampling: (a) Accuracy, (b) precision, (c) recall, and (d) f1-score.

C. *Microglia classification*

Figure 8 illustrates the results of accuracy and f1-score for the microglia use-case. Here it is even more evident that LCC outperforms random sampling.



(a) (b)

Figure 8 – Microglia classification performance of LCC sampling vs random sampling: (a) accuracy, and (b) precision.

D. Conclusion

The active learning sampling outperforms the random sampling for both experiments. The peak performance is reached 4 times faster in the microglia experiment, and 6 times faster in the TUNEL experiment. This reduction in amount of samples results in an equal reduction in the time spent labelling by researchers.

VI. Future work

It has proven to be the case that difficult classification problems are often preceded by an equally difficult segmentation problem. As such, future work should not focus on one or the other, as this paper has, but look at both problems together.

Another requirement for a more generic framework is to remove responsibility from the end user. Autoencoders may be applied to automatically extract the best features for each specific classification problem. After such an extraction, feature selection techniques can be applied to keep only the most relevant features for the classification.

VII. Conclusion

In this thesis we have proposed active learning as a solution to the excessive manual annotation required in traditional machine learning classifiers. From the different tested sampling methods, the LCC method was found to have the best performance on both synthetic and real data.

Two different experiments on real-world use-cases have shown that active learning greatly outperforms random sampling. In one case AL requires six times fewer training samples to converge to the maximum performance.

This is a significant reduction in the time spent labelling by researchers.

References

[1] Voet, S., Guire, C. M., Hagemeyer, N., Martens, A., Schroeder, A., Wieghofer, P., . . . Loo, G. V. (2018). A20 critically controls microglia activation and inhibits inflammasome-dependent neuroinflammation. Nature Communications, 9(1). doi:10.1038/s41467-018-04376-5

[2] Rawat, W., & Wang, Z. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. Neural Computation, 29(9), 2352-2449. doi:10.1162/neco_a_00990

[3] Labat-Moleur, F., Guillermet, C., Lorimier, P., Robert, C., Lantuejoul, S., Brambilla, E., & Negoescu, A. (1998). TUNEL Apoptotic Cell Detection in Tissue Sections: Critical Evaluation and Improvement. Journal of Histochemistry & Cytochemistry, 46(3), 327-334. doi:10.1177/002215549804600306

[4] Chandra, B., & Sharma, R. K. (2015). Exploring autoencoders for unsupervised feature selection. 2015 International Joint Conference on Neural Networks (IJCNN). doi:10.1109/ijcnn.2015.7280391