# Cells recognition in Pap smear exams

**Andrei Gonçalves Rohlfs Massaini, João Gabriel Polonio Teixeira, João Pedro Torres de Souza Silva**

Instituto de Ciências Exatas e Informática – PUC Minas
Belo Horizonte, Minas Gerais

{agrmassaini,927828,joao.silva}@sga.pucminas.br

***Abstract.*** *This article documents the application of image analysis and processing techniques in a local application according to work proposed by the Processing and Image Analysis. An intuitive graphic interface was implemented, which allows the user to load images of cell clusters obtained by Pap smears exams to individual analysis of the cells contained in the images.*

***Resumo.*** *Este artigo documenta a aplicação de técnicas de processamento e análise de imagens em uma aplicação local segundo trabalho proposto pela disciplina de Processamento e Análise de Imagens. Foi implementada uma interface gráfica intuitiva, que permite ao usuário carregar imagens de aglomerados de células obtidas em exames de Papanicolau para análise individual das células contidas nas imagens.*

## 1. Problem Description

The Pap smear exam consists of extracting cell samples from the cervix to diagnose cervical cancer if anomalies are recognized [Rezende et al. 2021]. This analysis aims to identify six classes of anomalous cells: ASC-US, ASC-H, LSIL, HSIL, AGC; and one class of healthy cells: Negative for Intraepithelial lesion.

The developed work intent to provide a graphic application [Limited 2021] able to classify images of cells obtained by Pap smear exams by using techniques of image analysis and processing, capturing the images characteristics, and using them to predict the class in which the represented cell fits.

## 2. Methodology

To make the application to serve it's purpose, was implemented analysis and description functionalities that are applied to every loaded image into it.

To analyze the tonal distribution within the images, the application generates a histogram calculated from the converted gray scale image using OpenCV [Bradski 2000], and another one calculated from the H and V channels of the converted HSV image, both being plotted by using the library Matplotlib [Hunter 2007]. To calculate the gray scale histogram, are considered the distinction between 16 tonalities of gray, and, to generate the H and V histogram, are distinguished 16 tonalities in the Hue channel, and 8 in the Value channel.

It also calculates Hu's invariant moments to capture shape features that are invariant to image transformations. This is done by splitting the input image in four channels: one for gray scale, and three others for HSV channels.

Additionally, co-occurrence matrices are generated, using Numpy [Harris et al. 2020], to analyze the spatial relationship between pixels in the images, and also to calculate it's Haralick Descriptors. The application calculates six co-occurrence matrices, mapping the adjacency values for each pair of 1, 2, 4, 8, 16 and 32 valued pixels in the gray scale converted images.

### 2.1. Classification Techniques

To implement the cell classifiers, were utilized a total of four models: two shallow models based on XGBoost [Chen and Guestrin 2016] from the Scikit-Learn [Pedregosa et al. 2011] Python library, and two pre-trained Efficient Net as the deep learning models, imported from Pytorch [Paszke et al. 2019] and Torchvision [tor 2021], that underwent fine-tuning.

The dataset used to train the four models was obtained by cutting Pap smear exams images in pictures $100 \times 100$ around the cell's nucleus contained in it. It was done by using Pandas [McKinney 2010] to read a CSV file which mapped the cells coordinates for each image, also containing the labels.

#### 2.1.1. XGBoost

For classifying the cells using XGBoost, the used dateset was composed by the values encountered in Hu's invariant moments calculus, generating a vector with 28 values (seven values for each channel) for each cell. Both, binary and multiclass, models were trained using 1000 estimators, a learning rate of 0.05, a maximum depth of 6, a subsample rate of 0.8 (to prevent overfitting on using randomly only 80% of the features), a column sample by tree rate of 0.8.

Given the dataset irregularity, which had different amount of samples for each class (341 for ASC-H, 126 for ASC-US, 321 for HSIL, 388 for LSIL, 902 for Negative for intraepithelial lesion and 43 for SCC) and was impacting in the model accuracy, it was balanced using the SMOTE technique, provided by Imbalanced-learn [Lemaître et al. 2017], on the multiclass training, getting the distribution illustrated in Figure 4. The results of both XGBoost training are represented in the Figures 1 as accuracy graphs and confusion matrices.

#### 2.1.2. EfficientNet

For classifying the cells using EfficientNet, the dataset consisted of $100 \times 100$ cell images extracted from Pap smear pictures. Each image was processed using a series of transformations including resizing to 256 pixels, center cropping to 224 pixels, and normalization to standardize the pixel values across the dataset. These steps ensure that the model receives consistent input data, which is crucial for effective training and classification.

To handle the training process, the dataset was split into training and testing sets with an 80-20 split. The training set was used to optimize the model's parameters using the Adam optimizer and the cross-entropy loss function, while the testing set evaluated the model's performance on unseen data to assess its generalization ability.

During training, performance metrics such as accuracy and loss were monitored and recorded for each epoch. Additionally, visualizations including accuracy curves and confusion matrices were generated to provide insights into the model's learning progress and its ability to distinguish between different cell types.

## 3. Results Analysis

This section presents the results of the image analysis and classification techniques, including accuracy metrics and visual examples.

### Model Performance

#### EfficientNet Multiclass

- **Accuracy:** 80.64%
- **Key Findings:**
  - Demonstrates high true positive rates for severe cases like ASC-H, HSIL, and SCC, critical for early intervention.
  - Struggles with distinguishing between ASC-US and LSIL categories, leading to notable misclassifications of benign cases as potentially pre-cancerous.
  - Overall performance indicates room for improvement in classifying less severe abnormalities and distinguishing benign lesions.

#### XGBoost Multiclass

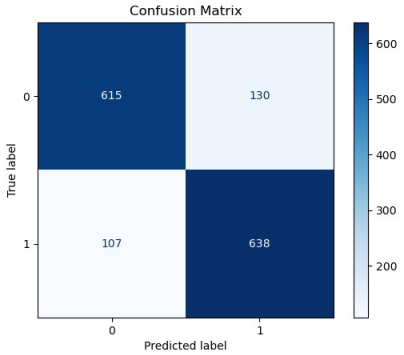
- **Accuracy:** 95.16%
- **Key Findings:**
  - Exceptional accuracy in identifying negative cases and LSIL, minimizing the risk of false negatives which are crucial in clinical settings.
  - Shows some misclassifications between ASC-US and HSIL, though at lower rates compared to EfficientNet.
  - Demonstrates robustness in distinguishing between different lesion types and benign cases, essential for accurate diagnosis.

#### XGBoost and EfficientNet Binary

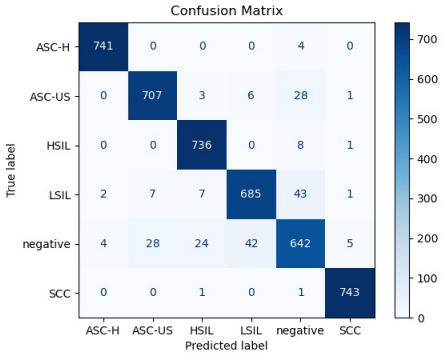
- **XGBoost Accuracy:** 84.09%
- **EfficientNet Accuracy:** 88.82%
- **Key Findings:**
  - Both models perform similarly in binary classification tasks, with XGBoost showing a slight advantage in accuracy.
  - XGBoost exhibits lower false negative rates, indicating its suitability for minimizing missed diagnoses in clinical applications.
  - EfficientNet, while performing well, could benefit from further refinement to reduce misclassifications, especially in distinguishing between benign and pre-cancerous lesions.

## 4. Conclusion

The confusion matrices indicate that while the classifier performs well in many areas, there is variability in its accuracy across different cancer cell types. The performance for common categories like ASC-H, HSIL, and SCC is strong, but there is notable room for improvement in categories like ASC-US and LSIL. Reducing the rates of false positives and false negatives will be crucial in enhancing the overall effectiveness of the classification model.

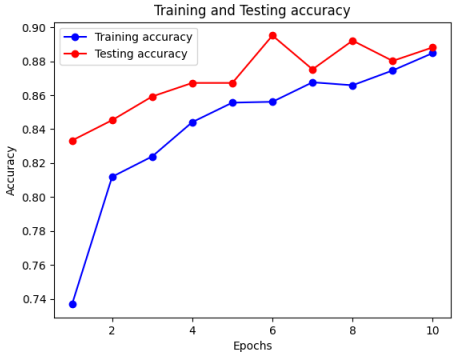



(a) Confusion matrix for the binary XGBoost.

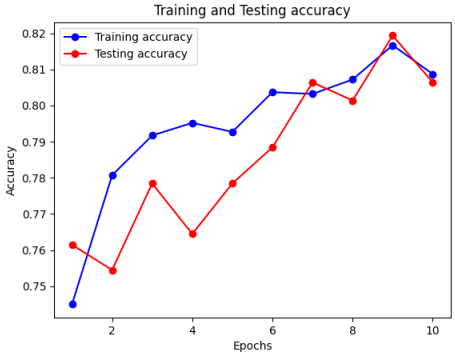



(b) Confusion matrix for the senary XGBoost.

**Figura 1. This figure represents the confusion matrices for XGBoost mentioned in Section 2.1.1**

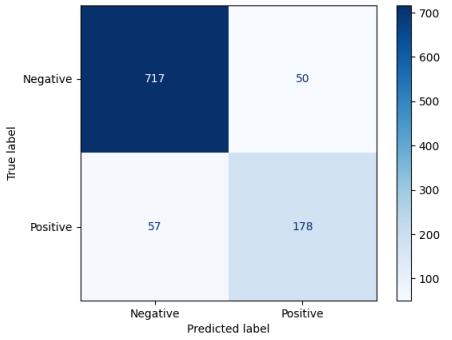



(a) Graph of accuracy through epochs for the binary EffNet mentioned in Section 2.1.2.

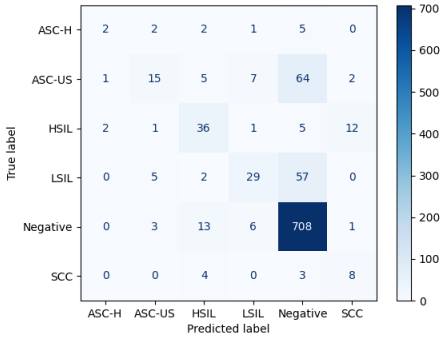



(b) Graph of accuracy through epochs for the senary EffNet mentioned in Section 2.1.2.

**Figura 2. This figure represents the accuracy through epochs graphs for EffNet**




(a) Confusion matrix for the binary EffNet mentioned in Section 2.1.2.




(b) Confusion matrix for the senary EffNet mentioned in Section 2.1.2.

**Figura 3. This figure represents the confusion matrices for EffNet**

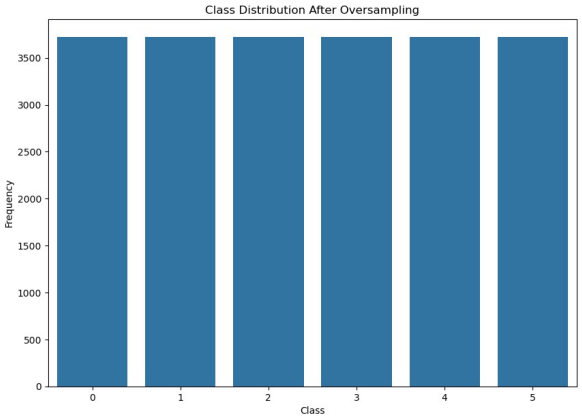



**Figura 4. This figure represents the final distribution of samples between classes for the dataset used in the XGBoost training mentioned in Section 2.1.1.**

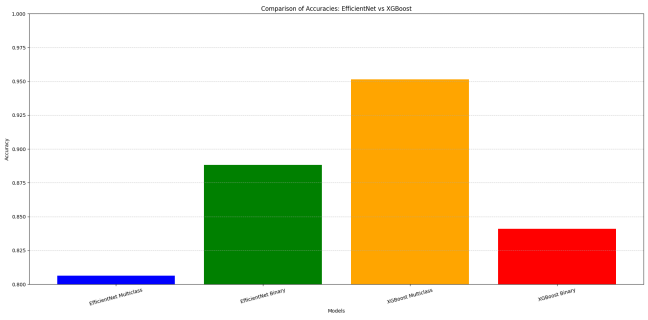



**Figura 5. This figure shows the overall accuracy comparasion**

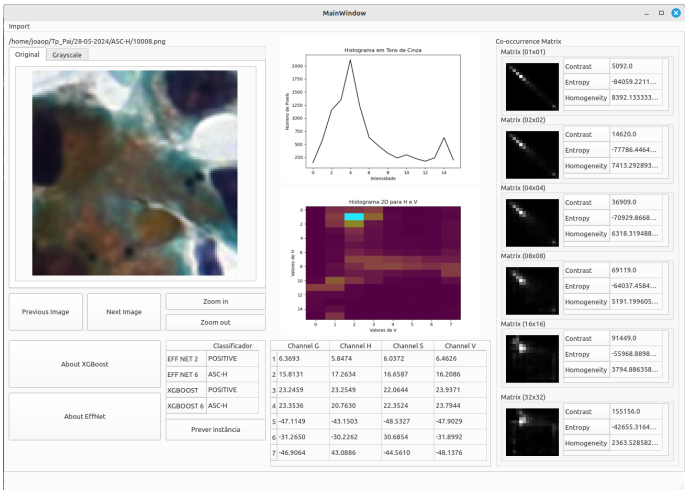



**Figura 6. This figure shows the developed application interface.**

## Referências

(2021). Torchvision: Part of the pytorch project. `https://pytorch.org/vision/stable/index.html`.

Bradski, G. (2000). The opencv library.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., et al. (2020). Array programming with numpy. *Nature*, 585(7825):357–362.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Lemaître, G., Nogueira, F., and Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

Limited, R. C. (2021). Pyqt6 documentation.

McKinney, W. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, pages 51–56.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Rezende, M., Silva, R., Bernardo, F., and et al. (2021). Cric searchable image database as a public platform for conventional pap smear cytology data. *Sci Data*, 8.