

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LARISSA ROZALES GONÇALVES

**Comet assay image analysis using
Convolutional Neural Networks**

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Science

Advisor: Prof. Dr. Luigi Carro

Porto Alegre
July 2018

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof^a. Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Wladimir Pinheiro do Nascimento

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Sérgio Luis Cechin

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

ABSTRACT

The comet assay is a technique used to detect DNA lesions. It involves analyzing images from individual cells, which can later be sorted into categories according to damage level. In this work we propose performing the image analysis and classification using Convolutional Neural Networks, comparing this method with other existing solutions.

Keywords: Comet assay. convolutional neural network. neural networks in python.

RESUMO

Comet assay é uma técnica utilizada para detectar lesões no DNA. Ela envolve analisar imagens de células individuais, que depois devem ser classificadas em categorias de acordo com seu nível de dano. Nesse trabalho propomos fazer a análise e classificação dessas imagens usando Redes Neurais Convolucionais, comparando esse método com outras soluções existentes.

Palavras-chave: Comet assay. redes neurais convolucionais. redes neurais em python.

LIST OF FIGURES

Figure 1.1 Comet assay image containing several comets.....	9
Figure 1.2 A single comet	10
Figure 1.3 Comets ranging from classes 0 to 4.....	11
Figure 2.1 CaspLab's user interface.....	14
Figure 2.2 OpenComet's user interface.....	15
Figure 2.3 Overview of HiComet's methodology	16
Figure 3.1 Architecture of LeNet-5.....	17
Figure 3.2 Convolutional kernels learned by AlexNet.....	18
Figure 4.1 Comets from class 2	21
Figure 5.1 The same comet, segmented with and without box expansion.....	24
Figure 5.2 Comet segmentation using Li's Minimum Cross Entropy thresholding method with and without extended boxes.....	26
Figure 5.3 A few threshold methods tested.....	27

LIST OF TABLES

Table 4.1 Accuracy results	22
Table 4.2 Accuracy results by number of examples per training class	23

LIST OF ABBREVIATIONS AND ACRONYMS

SCGE Single cell gel electrophoresis

CNN Convolutional Neural Network

ILSVRC ImageNet Large-Scale Visual Recognition Challenge

ReLU Rectified Linear Unit

CONTENTS

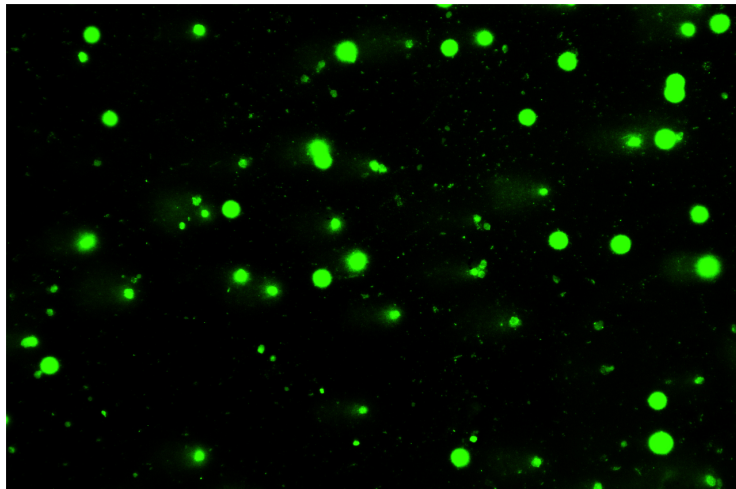
1 INTRODUCTION	9
1.1 Document Organization	11
2 RELATED WORK	13
2.1 Convolutional Neural Networks	13
2.2 Comet Assay	14
3 DETAILED REVIEW OF USED TECHNIQUES	17
3.1 Convolutional Neural Networks	17
3.2 Comet Assay Technique	19
4 IMPLEMENTATION	20
4.1 Results of Comet Classification	20
5 SEGMENTATION	24
6 CONCLUSION	28
6.1 Future Work	28
REFERENCES	30

1 INTRODUCTION

In biology, the comet assay is a technique used for detecting DNA damage in individual cells that can also be used to evaluate DNA repair (SILVA et al., 2000). Also known as single cell gel electrophoresis assay, or SCGE, it is a simple method that involves staining DNA comets with DNA-binding dye and analyzing the resulting images (GYORI et al., 2014).

Figure 1.1 shows one of those resulting images, which is made of several comets. For the classification to work, each individual comet, as shown in Figure 1.2, must be identified and then sorted into the right category, considering the severity of DNA damage: class 0 refers to undamaged cells, while class 4 represents maximum damage. Figure 1.3 shows one comet from each class. All comet images used in this work were obtained with Universidade Federal do Rio Grande do Sul's biochemistry department. More of their work is available at (SILVA et al., 2013).

Figure 1.1: Comet assay image containing several comets

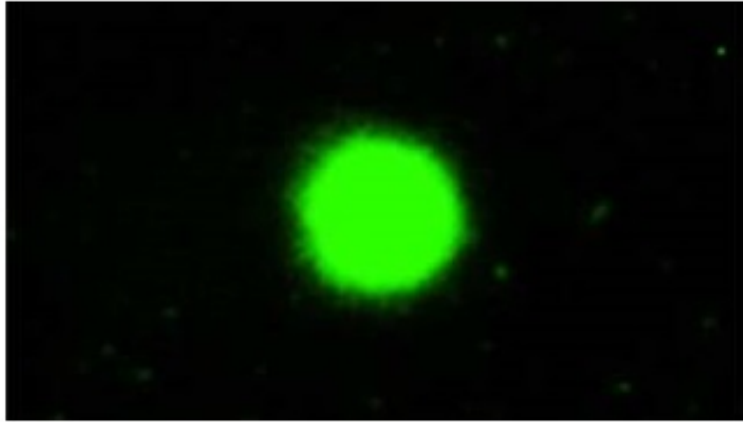


Fonte: UFRGS' biochemistry department

Currently there are several free and commercial tools available for comet image analysis; among the free ones are CaspLab (KONICA et al., 2003), CometScore (HOOVER, 2016), OpenComet (GYORI et al., 2014) and HiComet (LEE et al., 2018). The problem with all available free software is that they do not actually output the category of each comet, and instead return a variety of values such as comet area, head area and tail length, which the user can later use for manual classification. In this work we propose a method for automatic classification of comets into categories without the need for user interven-

tion.

Figure 1.2: A single comet



Source: UFRGS' biochemistry department

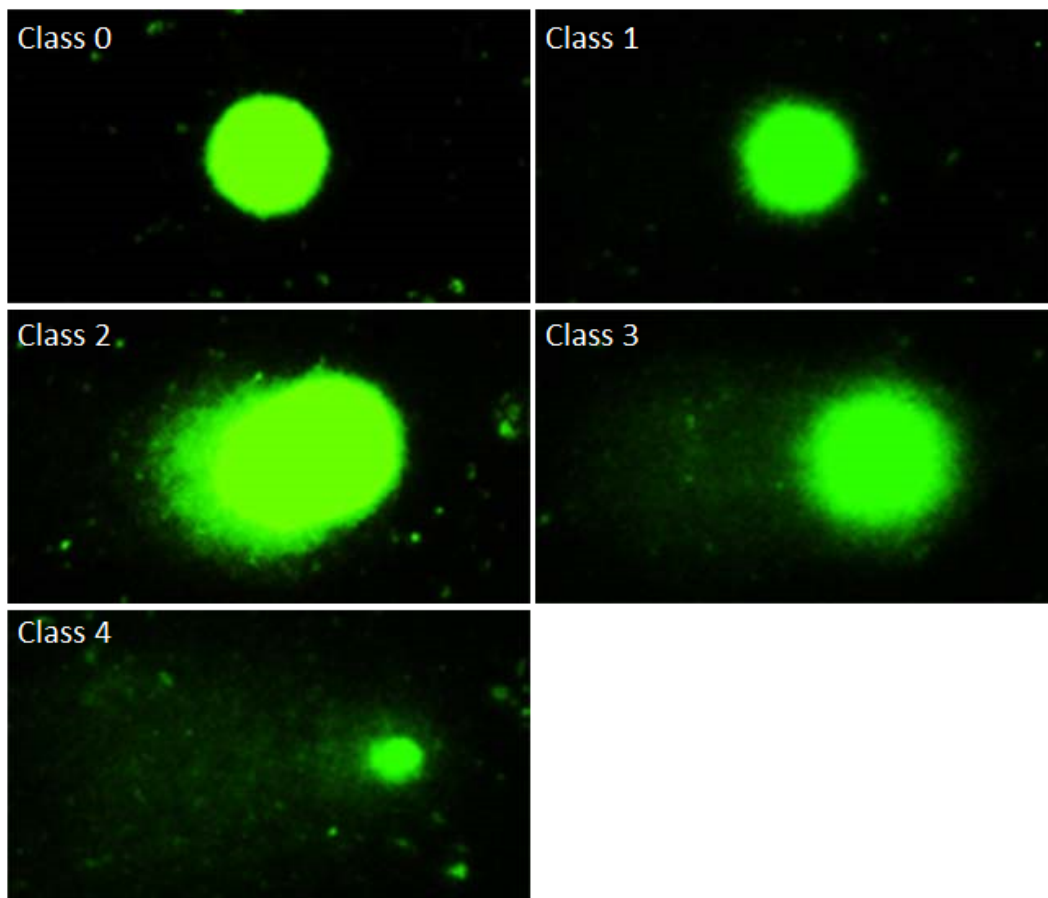
From the mentioned free comet image analysis tools, OpenComet and HiComet are the most recent ones. OpenComet had its latest version made available in 2016 and can identify individual comets on its own by using pixel intensity to separate the comets from the background. The tool cannot, however, identify comets that are too damaged, for example having the head separated from the tail, or recognize overlapping comets, leading to a quantity of valid comets being discarded. HiComet, released in 2018, overcomes this problem and can identify overlapping comets, but still cannot classify comets into categories.

In this work, we propose using a Convolutional Neural Network (CNN) to classify comet assay images. CNNs are a special kind of neural network designed to recognize patterns from images (LECUN, 2018). Image classification in CNNs works by using the image that needs to be classified as an input, and then processing this input by passing it through a sequence of pre-trained layers. The output of the network is the predicted class for the image.

A CNN architecture is usually composed by several layers of different types, such as convolutional and pooling layers. Convolutional layers are connected only locally, and are made of kernels that learn relevant features to recognize the images, such as edges. Kernels in the first layers learn basic features, while the ones in later layers learn higher-level ones. The pooling layers reduce the input size and control overfitting. After them, fully connected layers are responsible for predicting the class of the input image, based on how the network was previously trained.

We hypothesize that CNNs are an adequate method for the comet assay image analysis problem based on the fact that they are already widely used in several fields, ranging from applications that deal with face recognition in social media to video surveillance, and thus can potentially be trained for this task as well. On top of that, after training a network, running the classification for new images is done without user intervention, meaning that overall time spent by the user is reduced.

Figure 1.3: Comets ranging from classes 0 to 4



Source: UFRGS' biochemistry department

1.1 Document Organization

Following this introduction, chapter 2 of this work describes related works on both CNNs and comet assay image analysis. Chapter 3 follows with an explanation on how CNNs and the comet assay technique work. Chapter 4 shows the CNN implementation as well as its results on comet assay images, and chapter 5 introduces the segmentation

proposed to obtain individual comets from the original image. Chapter 6 presents the conclusion and future works.

2 RELATED WORK

This chapter introduces what has already been done in terms of image classification using CNNs, while also considering networks that offer solutions for the object detection problem. After that, we describe available free softwares for comet assay image analysis and the respective techniques each one of them uses for the task.

2.1 Convolutional Neural Networks

LeNet-5 was a network designed in 1998 to recognize handwritten letters and numbers (LECUN et al., 1998). Since then, several CNN architectures have been proposed for the image classification problem, and some of them, in more recent years, also offer solutions for object detection. In this section we will present several networks that have offered important contributions for solving both problems.

AlexNet was the first CNN to win an ILSVRC (ImageNet Large-Scale Visual Recognition Challenge), a competition that evaluates algorithms for object detection and image classification at large scale (RUSSAKOVSKY et al., 2015), in 2012. AlexNet has 60 million parameters and 650.000 neurons; it is composed by five convolutional layers and three fully-connected layers, followed by a 1000-way softmax function, and achieved an error rate of 15.6% compared to the error rate of 26.2% achieved by the second-best entry in that year (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

In 2014, VGG Net won second place in that year's ILSVRC, with an error rate of 7.2%. It is considerably deeper than AlexNet, with variations having 11 to 19 convolutional layers, and the number of parameters ranging from 133 to 144 million (SIMONYAN; ZISSERMAN, 2014). GoogLeNet, with a 6.67% error rate, was the winner of the 2014 ILSVRC, and has a different architecture from previous CNNs, being composed by modules that can be executed in parallel so that the entire network does not have to be run sequentially (SZEGEDY et al., 2014).

The 2015 ILSVRC winner, ResNet, has 152 layers, making it 8 times deeper than VGG Net (HE et al., 2015). The error rate achieved by ResNet in the ILSVRC is 3.57%, a better rate than humans, who usually score between 5 to 10%, can achieve (DESHPANDE, 2016).

All networks described until now are used for the image classification task, which consists on receiving an image as input and sorting it into a class. For the object detection

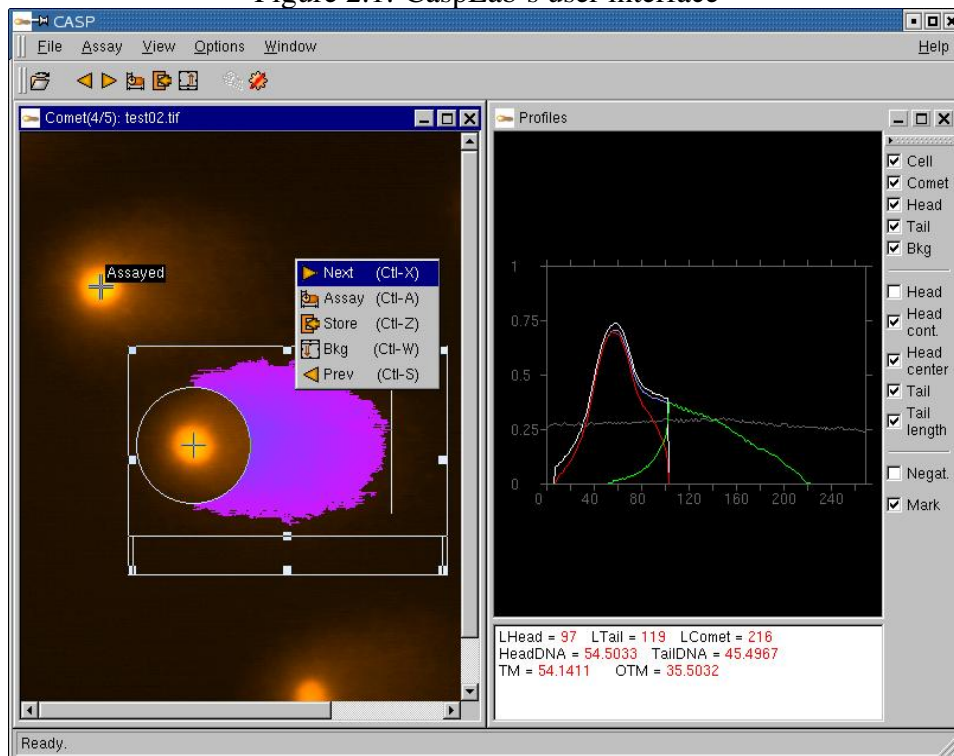
task, however, it is also important to locate separate objects in an image, meaning one image might have more than one object from different classes.

R-CNN (GIRSHICK et al., 2013), Fast R-CNN (GIRSHICK, 2015) and Faster R-CNN (REN et al., 2015) are CNNs designed to detect objects within images and then classify them into categories. Faster R-CNN, in particular, works by merging a Region Proposal Network and Fast R-CNN into a single network in order to achieve state-of-the-art object detection accuracy. YOLO, another network designed for object detection that has state-of-the-art accuracy, runs at 45 fps and has a faster, less accurate version that runs at 155 fps (REDMON et al., 2016).

2.2 Comet Assay

In this section we describe what is available in terms of free software for the analysis of comet assay images. It is relevant to remember that none of the available free software can classify comets into the desired five classes; instead they offer a variety of parameters related to each comet's tail and head.

Figure 2.1: CaspLab's user interface



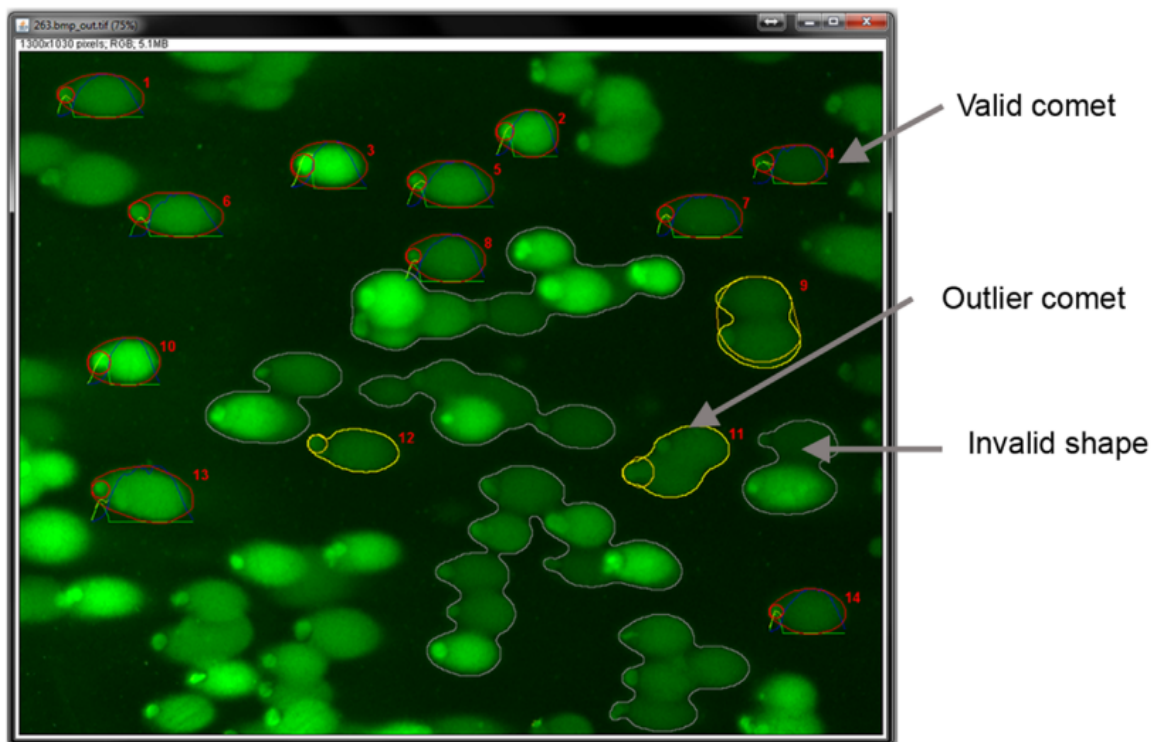
Source: (CASPLAB..., 2018)

CaspLab (KOńCA et al., 2003) works with color and gray-scale images of fluorescence-stained comets in the TIF format, with silver stained comet images needing to be converted into negatives. It calculates parameters such as head radius, tail length and tail moment. After launching the program, once the user has adjusted threshold sensitivity settings, it is estimated that 25 to 50 measurements can be performed in less than 10 minutes. CaspLab's interface can be seen in Figure 2.1.

CometScore (HOOVER, 2016) works by having the user adjust a threshold so that the comets can be separated from the background, and then drawing a rectangle around each comet that must not overlap with other comets. The program calculates several parameters, such as comet length, comet area, head area, head length, etc.

OpenComet (GYORI et al., 2014) is a plug-in for the ImageJ image processing platform. The user selects a set of images for analysis and an output folder to save the results in a spreadsheet format. It can automatically detect comet locations, but overlapping comets are considered invalid and discarded. Output parameters include comet length, head length, tail length, among others. OpenComet's interface is shown in Figure 2.2, where comets considered invalid by the software are also visible.

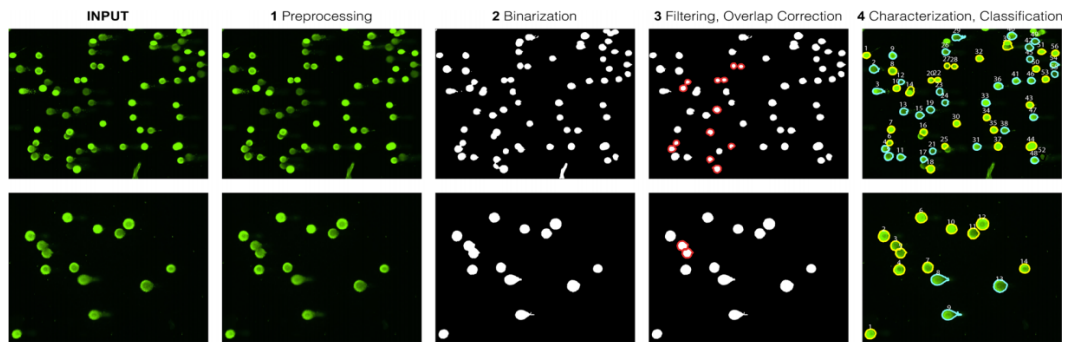
Figure 2.2: OpenComet's user interface



Source: (GYORI et al., 2014)

HiComet (LEE et al., 2018) goes through 4 steps (preprocessing, binarization, filtering and overlap correction, characterization and classification) in order to find parameters such as size, head moment, tail moment, etc. Unlike OpenComet, it can identify and analyze overlapping comets; this is an important feature, as overlapping comets are still valid for classification. Figure 2.3 shows the output image for each step of the HiComet method.

Figure 2.3: Overview of HiComet's methodology



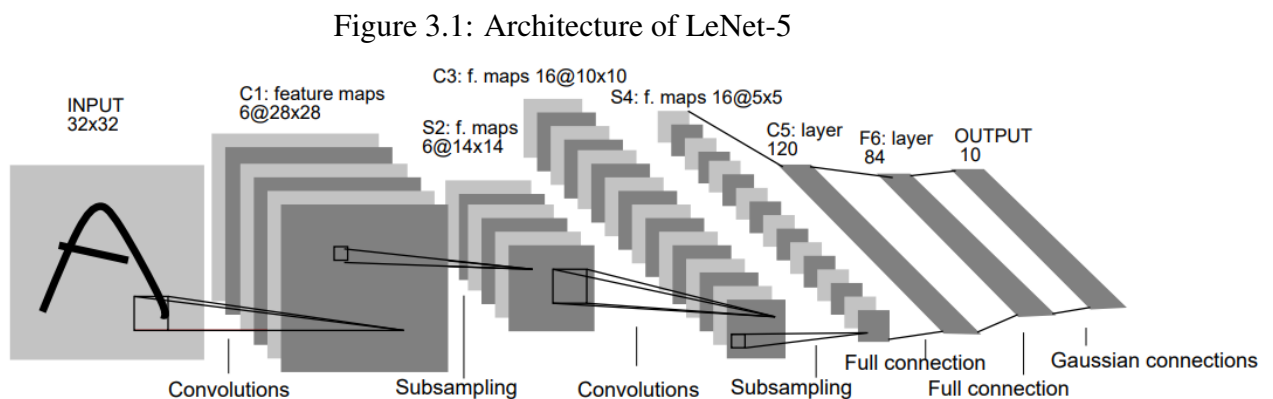
Source: (LEE et al., 2018)

3 DETAILED REVIEW OF USED TECHNIQUES

3.1 Convolutional Neural Networks

A Convolutional Neural Network is a neural network architecture inspired in biology, specifically by visual perception of living beings. It receives an input, processes it, and outputs the class predicted for the said input. When used for image processing in computer vision, for example, a CNN can receive a picture and classify its contents based on what categories the network has been trained to recognize. This section describes the components a regular CNN is made of. The next section describes how the network works for image classification.

Like other backpropagation neural networks, CNNs are composed by neurons, biases, and their interconnections. While there are architecture variations, typically CNNs are made by a combination of several convolutional, pooling (subsampling), and fully-connected layers. Figure 3.1 shows the CNN architecture of LeNet-5 with all mentioned layers, extracted from (LECUN et al., 1998).



Source: (LECUN et al., 1998)

Convolutional layers are made of kernels (also called filters), which in turn are made of neurons; neurons in CNNs share weights, introducing sparsity, reducing the number of total weights (KAFUNAH, 2016) and the model complexity, and making the network easier to train (GU et al., 2015). Those kernels learn features about the input they are fed with. Kernels in lower level layers learn simple, low-level features, while features learned by kernels in higher layers are more complex. By stacking several convolutional layers in succession, it is possible to learn high-level features that convey complex information about the input. Figure 3.2 shows convolutional kernels learned by AlexNet, with

a variety of frequency and orientation-selective kernels, extracted from (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). The kernels on the top of the image were trained by one GPU, while the ones on the bottom were trained by a second one; this figure shows an specialization exhibited by the two GPUs, with the second one learning color-specific kernels.

Figure 3.2: Convolutional kernels learned by AlexNet



Source: (KRIZHEVSKY; SUTSKEVER; HINTON, 2012)

Pooling layers usually follow convolutional layers, and are used to reduce the spatial size of the previous layer's output, also reducing the number of operations in subsequent layers, controlling overfitting. Common functions to use in the pooling layer are max-pooling and average pooling. Max-pooling, the most used from the two, consists on defining a window size, sliding it over the pooling layer's input (usually the output from the previous convolutional layer), and performing a max operation on chosen cells, passing to the next layer only the one with the highest value and effectively reducing matrix size.

Activation functions are commonly used after convolutional layers to introduce nonlinearities, making it possible for the network to learn non-linear features. One common way to do this is by using a rectified linear unit (ReLU) layer, although other functions can be used. ReLU takes the output of the convolutional layers and turns all negative values into zeros, keeping positive values the same.

After the sequence of convolutional and pooling layers, fully-connected layers can be used for high-level reasoning. As the name implies, neurons in these layers do not share weights and are normal backpropagation neural networks. When classifying an image, for example, each neuron from the last layer of the network can be interpreted to mean the probability of the image belonging to the class associated with that neuron. To choose which class the image belongs to, one method available is to pick the class

associated with the neuron who presented the highest value, usually only if that value is above a threshold. This can be done using a softmax function. A detailed review of CNN behavior and structure can be found in (GU et al., 2015).

With their structure in mind, training CNNs is very similar to training regular neural networks: the input is passed forward through all the layers, and then afterwards during the backpropagation phase the network is updated based on the output error. The problem with training this kind of network, however, is that not only a huge amount of data is needed, each input also has to be annotated with which class it belongs to. The annotation task is done by humans, and so requires time and resources. If not enough data is available, sometimes images are flipped, shifted, etc., in order to emulate more training cases.

3.2 Comet Assay Technique

The comet assay is a technique used for detecting DNA damage in individual cells. Cells are first embedded in agarose on a microscope slide, then lysed, which leaves nucleoids. Those nucleoids are electrophoresed in alkali, and DNA loops containing breaks are relaxed and extended towards the anode, forming a comet-like image, as already shown in Figure 1.2, that can be viewed by a fluorescence microscopy with a suitable stain. The extent of DNA damage can be expressed as the percentage of DNA fluorescence in the tail (LORENZO et al., 2013).

Visual scoring of the comets in the five categories is arbitrary and not strictly comparable with softwares that provide tail percentage. As it is, an approximation of each class on the visual scale can be seen as 20% of the percentage interval (COLLINS et al., 1997). Since available softwares do not output the class for each comet, it is usually done by the user; while it can, as mentioned, be done considering the tail percentage, it does not correspond perfectly to the five classes.

4 IMPLEMENTATION

While several CNN implementations are available for free, in this work we decided to focus on implementing one from scratch, in order to fully understand and be in control of all parameters. The full code is available at <https://bitbucket.org/lrzg/comet-assay-cnn/src/master/>.

The programming language chosen for the task was Python, using the Jupyter Notebook platform. This choice was based on the several libraries and packages available, such as numpy. The network follows a classic CNN structure as presented in chapter 2 of this work, with convolutional, max-pooling and ReLU layers in sequence, followed by fully-connected layers.

Several network parameters can be adjusted in the **config.txt** file. The number of convolutional, ReLU, max-pooling and fully-connected layers can be changed. The total number of filters of each convolutional layer can also be modified, as well as filter size; each layer's filter size and number of filters are independent from each other. Max-pooling layers have adjustable filter size and stride, and if the user does not want to use a max-pooling layer the filter size can be simply set to 1. It is also possible to modify the number of neurons in fully connected layers. Input images are of the **.bmp** format and their size can be changed by setting the correct **W** (width) and **H** (height) values in the first convolutional layer; those values have to be kept consistent in subsequent layer's configurations.

To ensure network correctness, we first trained our CNN to learn a set of simple geometric formats, with the following classes: circle, square, triangle. After assessing the network could learn these classes correctly, we moved on to real comet images.

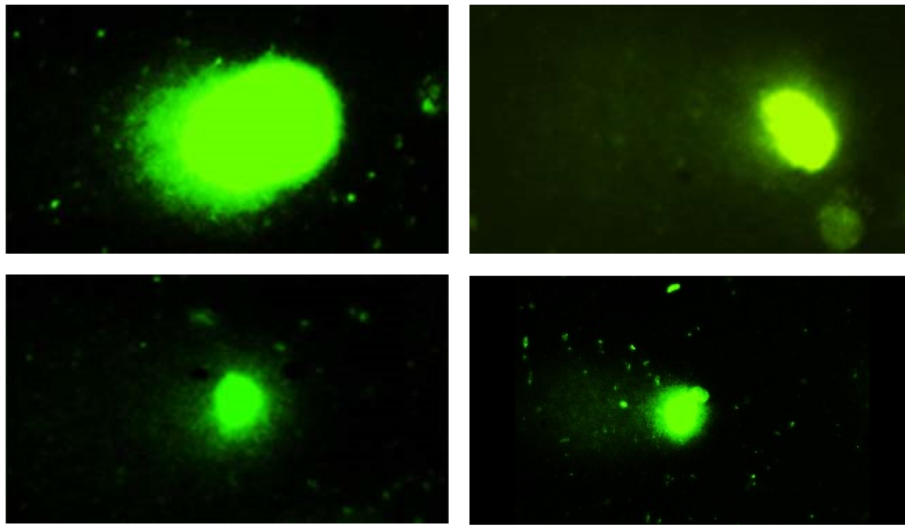
4.1 Results of Comet Classification

Training was performed using images obtained with UFRGS' biochemistry department. As we were able to obtain only few examples per class, from two microscope slides, training had to be done with a very limited training set. Each image was first labeled by one specialist, and then checked by another one to ensure it was classified in the correct class.

Due to the small number of images, comet category 2 had to be omitted from training, as it is a very heterogeneous category and most examples available from it had

a strong variation and dissimilarity. A few variations of class 2 comets can be seen in Figure 4.1, showing that some comets from this class have almost no tail while others do, and also that tail intensity severely varies between examples, making such a small training set inadequate for learning this class.

Figure 4.1: Comets from class 2



Source: UFRGS' biochemistry department

We trained the network with several configurations and a preset number of iterations. Input images are all 100x100 pixels. Accuracy was calculated by the percentage of right classifications per total images on the test set, previously separated from the training set. The test set is composed by two images per category. Accuracy results can be seen in Table 4.1

Configurations with the same number of convolutional layers where increasing iterations lowered the accuracy might mean an overfitting; considering our limited training set, the network could have learned features that are too specific of images from the training set, but do not generalize well. For the 5 convolutional layers setup, as it has more weights to train, 500 iterations might not have been sufficient for it to converge, hence the better accuracy of training with 1000 iterations. This behavior is in accordance with what has already been reported in literature (MITCHELL, 1997).

Table 4.1: Accuracy results

Number of convolutional layers	Iterations	Accuracy
2	500	75%
2	1000	62.5%
3	500	87.5%
3	1000	75%
4	500	62.5%
4	1000	62.5%
5	500	50%
5	1000	87.5%

We are aware that CNNs are usually trained with hundreds of images (GU et al., 2015), but comet image labeling by humans is a time consuming task, as it involves manually classifying each comet and may require a second person for validation. On top of that, transformations such as shifting or rotating images do not work well for this problem. Shifting a comet does not add relevant characteristics for the network to learn. Rotating a comet also cannot be done, as comets have a certain orientation, in our case with the tail extending to the left of the head; other softwares for comet assay image analysis also rely on the comets' head and tail orientations.

Due to the time factor, the total number of images we were able to obtain during this work's timespan was limited, but in the future we will be able to acquire more images with UFRGS' previously mentioned biochemistry department and, if possible, other external collaborators, such as different universities. Collaboration with other sources might also add important variations to the training and testing sets, as slides coming from other laboratories' microscopes will have different characteristics.

As it is, Table 4.2 shows that adding more images to the training set gradually increases network accuracy. The network configuration for this test was of two convolutional layers with a max-pooling layer between them, followed by three fully-connected layers. This topology has been reached after experimenting with different topologies, and obtaining a decent error in the training phase.

As we progress with this work we will obtain more images and, taking Table 4.2 as an example, increase the network accuracy more. With more examples we will also be able to cover comet variations better, including class 2, which will be added to training when possible. We also plan to train overlapping comets as separate classes; the network

Table 4.2: Accuracy results by number of examples per training class

Number of images per training class	Accuracy
1	37.5%
2	50%
3	62.5%
4	75%

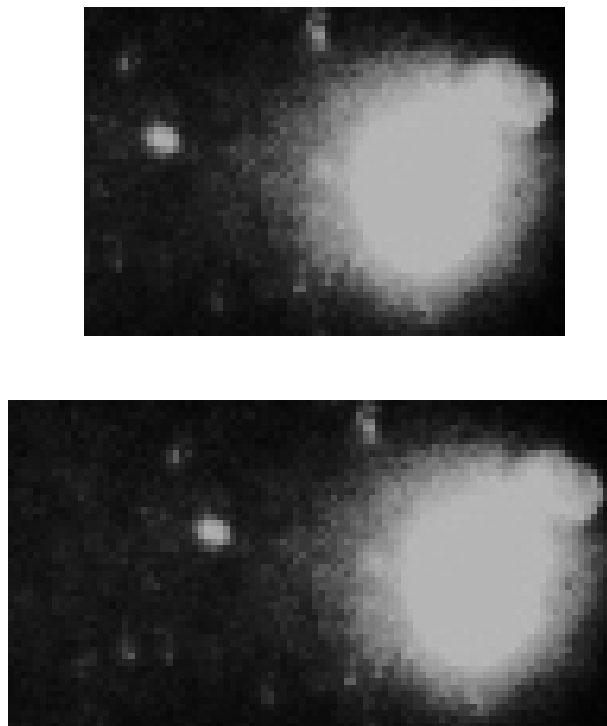
should have no problem learning them, seeing as one comet on top of another one has a distinguished shape from a single separate comet. This task, however, will require a lot of slides, as overlapping comets are not that common; collaboration with other sources is, then, essential to obtain a relevant amount of instances.

5 SEGMENTATION

Segmentation of comet assay images was also done in Python, using image labelling (LABEL..., 2018) and Li's Minimum Cross Entropy thresholding method (LI; LEE, 1993). The original code is available at http://scikit-image.org/docs/dev/auto_examples/segmentation/plot_label.html, while the modified code is available together with the CNN code.

A few changes were made in the code to better adapt it to comet segmentation. First, bounding boxes smaller than a set size were discarded, as objects that small are not considered comets. Second, we changed the original thresholding function from Otsu's method to Li's Minimum Cross Entropy method, as it better performs comet detection. Last, we expanded some boxes to cover comet tails more thoroughly.

Figure 5.1: The same comet, segmented with and without box expansion



Li's Minimum Cross Entropy was chosen as the thresholding method after comparing it to several other methods. Li's method obtained the best results in terms of being able to detect all comets. A comparison between some of the thresholding methods tested for this task can be seen in Figure 5.3. Otsu's method (OTSU, 1979) can detect the major-

ity of comet heads, but cuts off some tails that are detected by Li's method; the ISODATA method (RIDLER; CALVARD, 1978) presents similar problems. Yen's method (YEN; CHANG; CHANG, 1995) cannot detect several separate comets as different objects, and groups them in one box.

While the proposed method is able to detect all comets on test images, tails from heavily damaged comets sometimes get cut off in the middle. In order to overcome this problem, we expanded bounding boxes that have higher width than height, so bounding boxes that were reaching only part of a tail could better cover the whole comet. Results before and after bounding box expansion can be seen in Figure 5.2. Those images were not classified by our network as they need first to be evaluated by a specialist.

Figure 5.1 shows a single comet obtained from this segmentation technique. Without expanding boxes, a part of the comet's tail is lost, but box expanding fixes this issue. The box expansion, however, cannot be too big, or it might overlap with another comet. Due to image availability, we have only been able to test this method with two comet assay images, which were provided by UFRGS' biochemistry department.

In order to train the network in the future with images extracted directly from microscope slides by the segmentation step presented here, we aim to obtain not only a higher quantity of slide images, but also for those slides to have each comet classification on them. Specialists manually drawing bounding boxes on all valid comets and assigning each one a class, for example, will be one way to compare if our method can detect all comets and appropriately segment them.

Figure 5.2: Comet segmentation using Li's Minimum Cross Entropy thresholding method with and without extended boxes

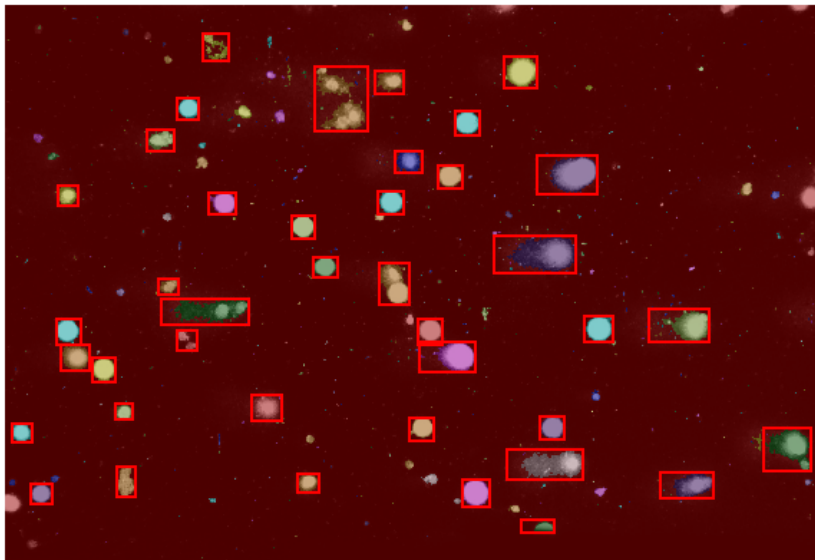
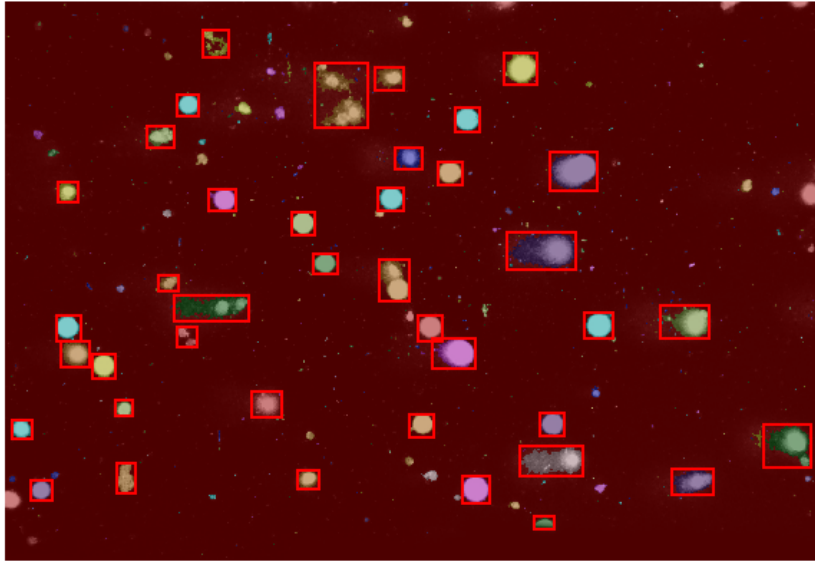
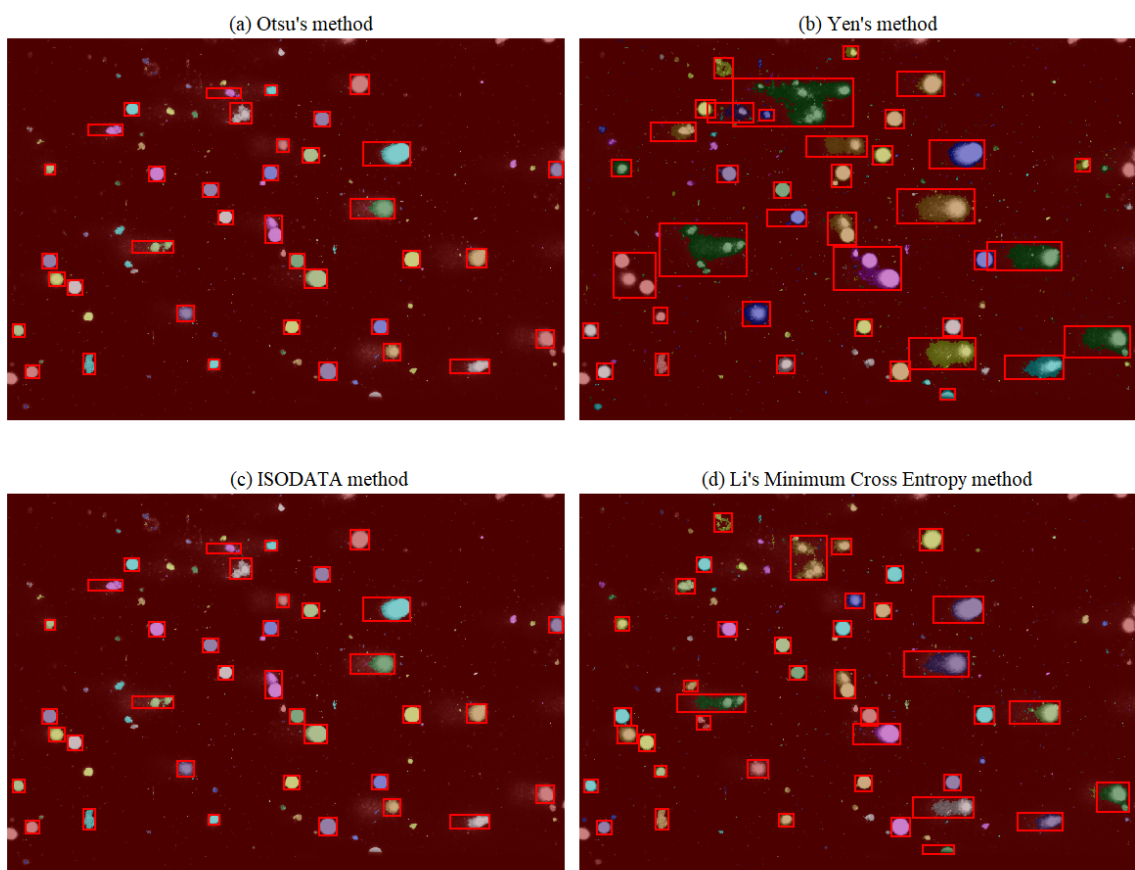


Figure 5.3: A few threshold methods tested



6 CONCLUSION

While there are many CNN implementations available as open code, the process of programming one from scratch was important for this work, as not only a deeper understanding of them was achieved, but also all relevant parameters can be set as wished. It also means that integration of CNN and segmentation codes will be trivial, as the two are already on the same platform.

Choosing Python to implement the CNN means it does not run as fast as it could if it were done in another language, such as C. However, as the running time for processing comets is measured in minutes, and since no user intervention is needed, it does not impact usability. The reduced time taken to implement the code in Python over other available options due to specific libraries and packages was enough to make up for the running time being a bit slower. Training each CNN configuration for this work, however, took approximately one hour per network, and will take more as we expand the training set and train for more iterations. Training time, if programming was done in C, could have been significantly reduced.

6.1 Future Work

In this work we demonstrated that a CNN is capable of classifying comets into four of the five comet assay categories. The next step is acquiring more pre-labeled images with area specialists. With more images in the training and testing sets, we can potentially increase accuracy, as it has already been demonstrated in Table 4.2. On top of that, more slides coming from different microscopes also mean a different variety of comets in each class; since comets, even in the same class, have varying characteristics, it is important to collect examples that encompass all of them. The best example of this behavior is class two, which could not be trained due to highly different comet images from the same class.

We also plan to make the transition from segmentation to classification automatic, and train the network directly with images from our segmentation algorithm. For this to work, we need to obtain more classified images, specifically with each comet in a full slide annotated. An specialist can, ideally, draw bounding boxes in the slide images, attaching a class to each one; this way we can also thoroughly check if our segmentation algorithm is missing any important comets as well as correctly covering all comets' area.

Collaboration with other laboratories will be important for gathering a significant

amount of examples, also adding possible comet characteristics variations between laboratories. Not only that, the green dye of the images used in this work is not the only possible option; gathering images obtained with other dye colors is important. Based on the fact that the network can be trained with grayscale images, we believe other color will not affect performance.

REFERENCES

- CASPLAB Screenshots. 2018. <<http://casplab.com/screenshots>>. Access date: 28 jun. 2018.
- COLLINS, A. et al. Comet assay in human biomonitoring studies: Reliability, validation, and applications. **Environmental and Molecular Mutagenesis**, Wiley-Liss Inc., v. 30, n. 2, p. 139–146, 1997. ISSN 0893-6692.
- DESHPANDE, A. **The 9 Deep Learning Papers You Need To Know About (Understanding CNNs Part 3)**. 2016. <<https://adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html>>. Access date: 25 jun. 2018.
- GIRSHICK, R. B. Fast R-CNN. **CoRR**, abs/1504.08083, 2015.
- GIRSHICK, R. B. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. **CoRR**, abs/1311.2524, 2013.
- GU, J. et al. Recent advances in convolutional neural networks. **CoRR**, abs/1512.07108, 2015.
- GYORI, B. M. et al. Opencomet: An automated tool for comet assay image analysis. **Redox Biology**, v. 2, p. 457 – 465, 2014. ISSN 2213-2317.
- HE, K. et al. Deep residual learning for image recognition. **CoRR**, abs/1512.03385, 2015.
- HOOVER, R. **CometScore**. 2016. <<http://rexhoover.com/index.php?id=cometscore>>. Access date: 25 jun. 2018.
- KAFUNAH, J. **Backpropagation In Convolutional Neural Networks**. 2016. <<http://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/>>. Access date: 08 mai. 2018.
- KOŃCA, K. et al. A cross-platform public domain pc image-analysis program for the comet assay. **Mutation Research/Genetic Toxicology and Environmental Mutagenesis**, v. 534, n. 1, p. 15 – 20, 2003. ISSN 1383-5718.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1**. USA: Curran Associates Inc., 2012. (NIPS'12), p. 1097–1105.
- LABEL image regions. 2018. <http://scikit-image.org/docs/dev/auto_examples/segmentation/plot_label.html>. Access date: 22 jun. 2018.
- LECUN, Y. **LeNet-5, convolutional neural networks**. 2018. <<http://yann.lecun.com/exdb/lenet/>>. Access date: 27 jun. 2018.
- LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, Nov 1998. ISSN 0018-9219.

- LEE, T. et al. Hicomet: a high-throughput comet analysis tool for large-scale dna damage assessment. **BMC Bioinformatics**, v. 19, n. 1, p. 44, Feb 2018. ISSN 1471-2105.
- LI, C.; LEE, C. Minimum cross entropy thresholding. **Pattern Recognition**, v. 26, n. 4, p. 617 – 625, 1993. ISSN 0031-3203.
- LORENZO, Y. et al. The comet assay, dna damage, dna repair and cytotoxicity: hedgehogs are not always dead. **Mutagenesis**, v. 28, n. 4, p. 427–432, 2013.
- MITCHELL, T. M. **Machine Learning**. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072.
- OTSU, N. A threshold selection method from gray-level histograms. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 9, n. 1, p. 62–66, Jan 1979. ISSN 0018-9472.
- REDMON, J. et al. You only look once: Unified, real-time object detection. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 779–788.
- REN, S. et al. Faster R-CNN: towards real-time object detection with region proposal networks. **CoRR**, abs/1506.01497, 2015.
- RIDLER, T.; CALVARD, S. Picture thresholding using an iterative selection method. v. 8, p. 630–632, 08 1978.
- RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, v. 115, n. 3, p. 211–252, 2015.
- SILVA, A. L. G. da et al. Evaluation of DNA damage in COPD patients and its correlation with polymorphisms in repair genes. **BMC Medical Genetics**, 2013.
- SILVA, J. d. et al. An alkaline single-cell gel electrophoresis (comet) assay for environmental biomonitoring with native rodents. **Genetics and Molecular Biology**, sciELO, v. 23, p. 241 – 245, 03 2000. ISSN 1415-4757.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **CoRR**, abs/1409.1556, 2014.
- SZEGEDY, C. et al. Going deeper with convolutions. **CoRR**, abs/1409.4842, 2014.
- YEN, J.-C.; CHANG, F.-J.; CHANG, S. A new criterion for automatic multilevel thresholding. **IEEE Transactions on Image Processing**, v. 4, n. 3, p. 370–378, Mar 1995. ISSN 1057-7149.