UNIVERSITAT DE BARCELONA

Deep Learning System for the Automatic Classification of Normal and Dysplastic Peripheral Blood Cells as a Support Tool for the Diagnosis

Andrea Milena Acevedo Lipes

Deep Learning System for the Automatic Classification of Normal and Dysplastic Peripheral Blood Cells as a Support Tool for the Diagnosis

Author

Andrea Milena Acevedo Lipes

Supervised by

Dr. José Rodellar

Dra. Anna Merino

Tutor

Dra. Roser Sala Llonch

This thesis is presented to obtain the degree of

Doctor in Biomedicine

Doctoral programme in Biomedicine

UNIVERSITAT DE BARCELONA

December, 2020

Acknowledgements

First, I would like to express my deepest gratitude to Dr. José Rodellar and Dra. Anna Merino for their unwavering support and invaluable advice. I want to express my sincere appreciation for their trust in me and the opportunity to develop this thesis.

My gratitude extends to all members of the research group Control, Modelling, Identification and Applications (CoDAlab). Thanks to Dr. Santiago Alférez for his friendship and support. His contributions to the research group have been a motivation to this thesis. I also thank to Laura Boldú and Ángel Molina for their insightful collaborations and continuous work.

I would like to thank to my tutors, Dr. Javier Pavia and Dra. Roser Sala Llonch from the University of Barcelona. Their feedback have helped me to successfully complete this thesis.

I wish to show my gratitude to the Hematology and Cytology Unit at the Core Laboratory in the Hospital Clinic of Barcelona, for allowing me use the facilities to obtain the images and confirmed diagnoses of the patients. I greatly appreciate the knowledge I have acquired during this experience.

I am also grateful to the Technical University of Catalonia (UPC), Barcelona East Engineering School and Department of Mathematics for providing me a unique academic environment and teaching experience in such beautiful facilities.

I wish to thank to the Administrative Department of Science, Technology and Innovation, Colciencias, for granting me the scholarship to carry out my doctoral research.

I am deeply grateful to all my family, specially to my mother Ligia, my sister Ángela and my brothers Jorge and Carlos, for their loving support and care. I specially thank Albert, for his love and precious company and to Adrià Escámez family for having welcomed me into their home.

My appreciation also goes out to my friends who directly or indirectly helped me to complete this thesis.

Any omission in this brief acknowledgement does not mean lack of gratitude.

Abstract

Clinical pathologists identify visually many morphological features to characterize the different normal cells, as well as the abnormal cell types whose presence in peripheral blood is the evidence of serious diseases. Disadvantages of visual morphological analysis are that it is time consuming, needs expertise to perform an objective review of the smears and is prone to inter-observer variability. Also, most of the morphological descriptions are given in qualitative terms and there is a lack of quantitative measures. The general objective of this thesis is the automatic recognition of normal and dysplastic cells circulating in blood in myelodysplastic syndromes using convolutional neural networks and digital image processing techniques. In order to accomplish this objective, this work starts with the design and development of a MySQL Database to store information and images from patients and the development of a first classifier of four groups of cells, using convolutional neural networks as feature extractors. Then, a high-quality dataset of around 17,000 images of normal blood cells is compiled and used for the development of a recognition system of eight groups of blood cells. In this work, we compare two transfer learning approaches to find the best to classify the different cell types. In the second part of the thesis, a new convolutional neural network model for the diagnosis of myelodysplastic syndromes is developed. This model was validated by means of a proof of concept. It is considered among the first models that have been built for diagnosis support. The final work of the thesis is the integration of two convolutional networks in a modular system for the automatic classification of normal and abnormal cells. The methodology and

models developed constitute a step forward to the implementation of a modular system to recognize automatically all cell types in a real setup in the laboratory.

Keywords: *Blood morphology; Convolutional neural networks; Blood cells automatic recognition; Myelodysplastic syndromes; Diagnosis support*

Contents

List of Figures

List of Tables

List of Abbreviations

AA Aplastic Anaemia.

AI Artificial Intelligence.

ALL Acute Lymphocytic Leukemia.

AML Acute Myeloid Leukemia.

BLASTS Blasts.

BM Bone Marrow.

CLL Chronic Lymphocytic Leukemia.

CLR Infections.

CNN Convolutional Neural Network.

CoDAlab Control, Modelling, Identification and Applications.

DB Database.

FC Fully Connected.

FL Follicular Lymphoma.

FM Feature maps.

HCL Hairy Cell Leukemia.

ig immature granulocytes.

LGL-T Large Granular Lymphocyte Leukemia.

MCL Mantle Cell Lymphoma.

MDS Myelodysplastic Syndromes.

ML Machine Learning.

MNC Mononuclear Cells.

MySQL MySQL Database.

PB Peripheral Blood.

PCA Pricipal Components Analysis.

PCL Plasma Cell Leukemia.

PL B and T Prolymphocytic Leukemia.

POC Proof of Concept.

ReLU Rectifier Linear Unit.

ROC Receiver Operating Characteristic.

SGD Stochastic Gradient Descent.

SLPC-B B-cell Chronic Lymphoproliferative Disorder.

SLPC-T T-cell Chronic Lymphoproliferative Disorder.

SMZL Splenic Marginal Zone Lymphoma.

SQL SQL Language.

SS Sézary Sindrome.

SVM Support Vector Machine.

t-SNE t-distributed Stochastic Neighbour Embedding.

TPR True Positive Rate.

UPC Technical University of Catalonia.

WHO World Health Organization.

Chapter 1

Introduction

1.1 Motivation

Peripheral Blood (PB) carries several cell types suspended in plasma, all essential for immunity and life: erythrocytes (red cells), leukocytes (white cells) and thrombocytes (platelets). Fortunately, PB is easily accessible and observable by optical microscopy, therefore visual cell inspection is very relevant in working flows of clinical laboratories. World Health Organization (WHO) considers morphology, along with other complementary tests such as immunophenotype, cytogenetic and molecular genetic studies, essential for the integral diagnosis of hematological diseases.

Clinical pathologists identify many morphological features to characterize the different normal cells, as well as the abnormal cell types whose presence in PB is the evidence of serious diseases. A drawback of visual morphological analysis is that is time consuming, needs expertise to review the PB smears in an objective and reliable way and is prone to inter-observer variability. Also, most of the morphological descriptions are given in qualitative terms and there is a lack of quantitative measures. As a result, the subtle inter-class morphological differences among abnormal cell types produce low specificity scores

in the daily work screening.

The detection and correct classification of abnormal cells is required to give a correct and prompt initial diagnostic approach. Laboratory autoanalysers show flags when the instrument detects possible blood cell abnormalities. Nevertheless, the smear review under the microscope is needed to confirm the presence of abnormal cells circulating in blood. Given all these limitations in the recognition of malignant cells, our research group Control, Modelling, Identification and Applications (CoDAlab) from Technical University of Catalonia (UPC) in collaboration with the Hematology and Cytology Unit of the Core laboratory at the Hospital Clinic of Barcelona, are interested in proposing solutions to the automatic recognition of PB cells, using Artificial Intelligence (AI), to assist in the daily work of clinical pathologists.

The rapid development of AI algorithms in recent years provides computational tools to solve the problem of the objective morphological classification of PB cells. Specifically, deep learning comprises multilayered computer models to extract features from input information to discover automatically subjective and intuitive patterns from complex data structures like the images of PB cells. The main objective of this thesis is the automatic classification of normal PB cells and dysplastic cells associated to myelodysplastic syndromes circulating in PB through the use of deep Convolutional Neural Network (CNN). This research is part of the global objectives of the group to advance in the development of PB cell classification systems using deep learning.

1.2 Objectives

The general objective of this thesis is the automatic recognition of normal and abnormal cells circulating in blood in myelodysplastic syndromes using CNNs and digital image processing techniques.

To achieve this, the following specific objectives have been proposed:

1. Configure a database of PB cell images acquired during the research using CellaVision DM96. The database will be organized by entities and patients.
2. Extract deep features from PB cell images using different architectures of CNNs described in the literature and further use these features to train a classifier to recognize the corresponding images.
3. Design and assessment of a CNN recognition system for eight groups of PB cells: neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes (myelocytes, metamyelocytes and promyelocytes), erythroblasts and platelets.
4. Design and testing of a CNN model for the automatic recognition of hypogranulated dysplastic neutrophils in PB.
5. Perform a proof of concept with cell images from MDS patients to validate the proposed system for the diagnosis support of MDS.
6. Integration of two CNN models in a system for the automatic recognition of MDS using all types of blood cells circulating in PB.

1.3 Theoretical framework

This section highlights the body of knowledge used for the development of this thesis. The framework is organized in three principal blocks: fundamental knowledge, specialized knowledge and tools.

The fundamental knowledge displayed in Figure 1.1 is the group of disciplines, represented by cubes, upon which the other two blocks are constructed. A background on

linear algebra (Chapter 2 in reference [1]) is useful to refresh basic concepts related to matrix transformations, eigenvalues and eigenvectors decomposition, which were important in issues like dimensional reduction used in data analysis. Multivariable calculus and optimization tools were the foundations for the neural networks training [2]. Probability theory background in Chapters 1 and 2 of reference [3], like algebra of events, law of total probability, Bayes theorem and probability distributions were also reviewed to apply concepts from information theory like entropy and the entropy loss used to measure the performance of a neural network.

On the other hand, peripheral blood cell morphology was studied from [4], Chapters 1, 2 and 6, to learn and understand the principal morphological features of PB cells, specifically those of the normal cells: neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes, erythroblasts and platelets and dysplastic cells: hypogranulated, hypersegmented and Pelger-Hüet neutrophils. A more detailed description of the most important morphological features of normal PB cells is presented in Chapters 3 and 4 of this thesis. The relevant features of dysplastic cells are described in Chapter 5.

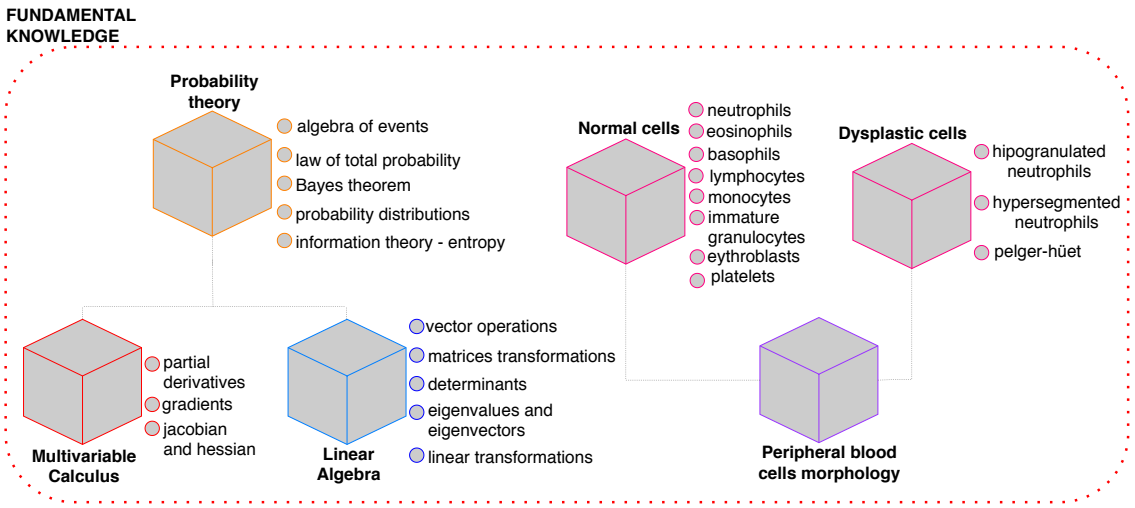


Figure 1.1: *Fundamental knowledge. Group of basic disciplines and concepts upon which the other two blocks are constructed. Cubes represent the disciplines and the dots are their respective concepts used to develop this thesis.*

The second block in Figure 1.2 represents specialized knowledge, conformed by concepts and tools related to machine learning: Support Vector Machine (SVM) and CNNs.

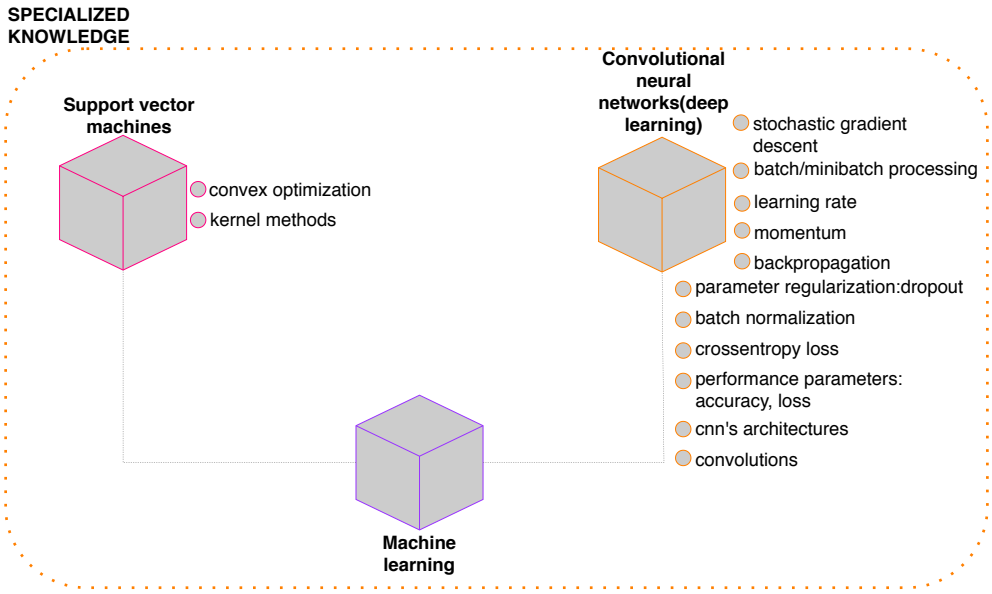


Figure 1.2: *Specialized knowledge. Specific concepts related to machine learning like the Support Vector Machine (SVM) and Convolutional Neural Network (CNN).*

SVMs more important related concepts are the convex optimization and the kernel methods, which are the basics to understand their inner functioning. These concepts were reviewed from [5], Chapter 3 and from [6], Chapters 5 and 6. SVMs were implemented in works presented in Chapters 2 and 4 of this thesis.

CNNs, the principal model implemented in this thesis, and related concepts like the Stochastic Gradient Descent (SGD), backpropagation and crossentropy loss used for training, as well as basic concepts to tune models like batch/minibatch processing, learning rate, batch normalization, momentum, and dropout were reviewed from different sources: Chapters, 7, 8 and 9 from [1], Chapter 15 from [5], Chapters 10 and 11 from [7] and Part I from [8]. The architectures implemented in our research and the different layers that conform the CNNs, as well as the training algorithms, are explained in detail in Chapters 4 and 5 of this thesis.

The last block in Figure 1.3 are the software tools used to implement the CNNs. The first work about classification presented in Chapter 2 was developed with the propietary software Matlab and a toolbox named matconvnet [9], which contains several pretrained

CNN architectures easy to implement and train. The rapid development of more powerful open source libraries in python [10], like tensorflow [11] and keras [12], allowed us to adopt them along with other numerical libraries like numpy [13], pandas [14], scikit-learn [15], matplotlib [16] and seaborn [17]. The mysql library [18] was used to design the images database explained in Chapter 2 of this thesis.

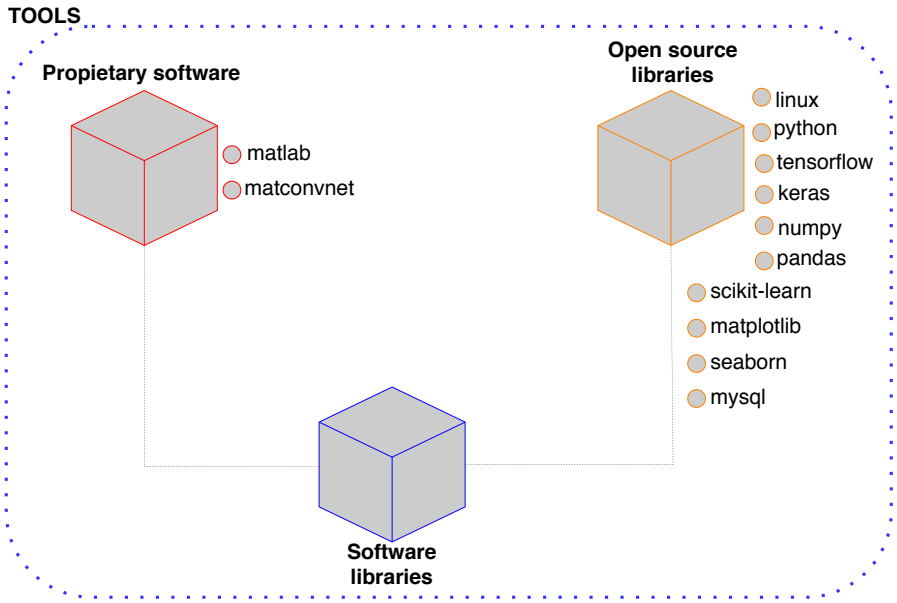


Figure 1.3: *Software tools used to to implement the models developed in this thesis.*

1.4 Research methodology

The use of CNNs for image classification requires a set of steps to obtain models that are able to extract useful information by means of convolutional layers and use it to perform the automatic classification with high accuracy. These steps conform the research methodology presented in the flowchart in Figure 1.4. Rectangles represent the principal steps in the methodology. Parallelograms are outputs resulting from some processes and rhombus represent points where a decision must be made.

The research workflow is organized in two blocks (dashed rectangles). The first part, from the acquisition of images to the data split, are data driven processes, which have to do only

with the data processing needed for training the models. The second block, from selection of model to the workflow end, are model driven processes because they are related to the model selection, design, training and testing processes.

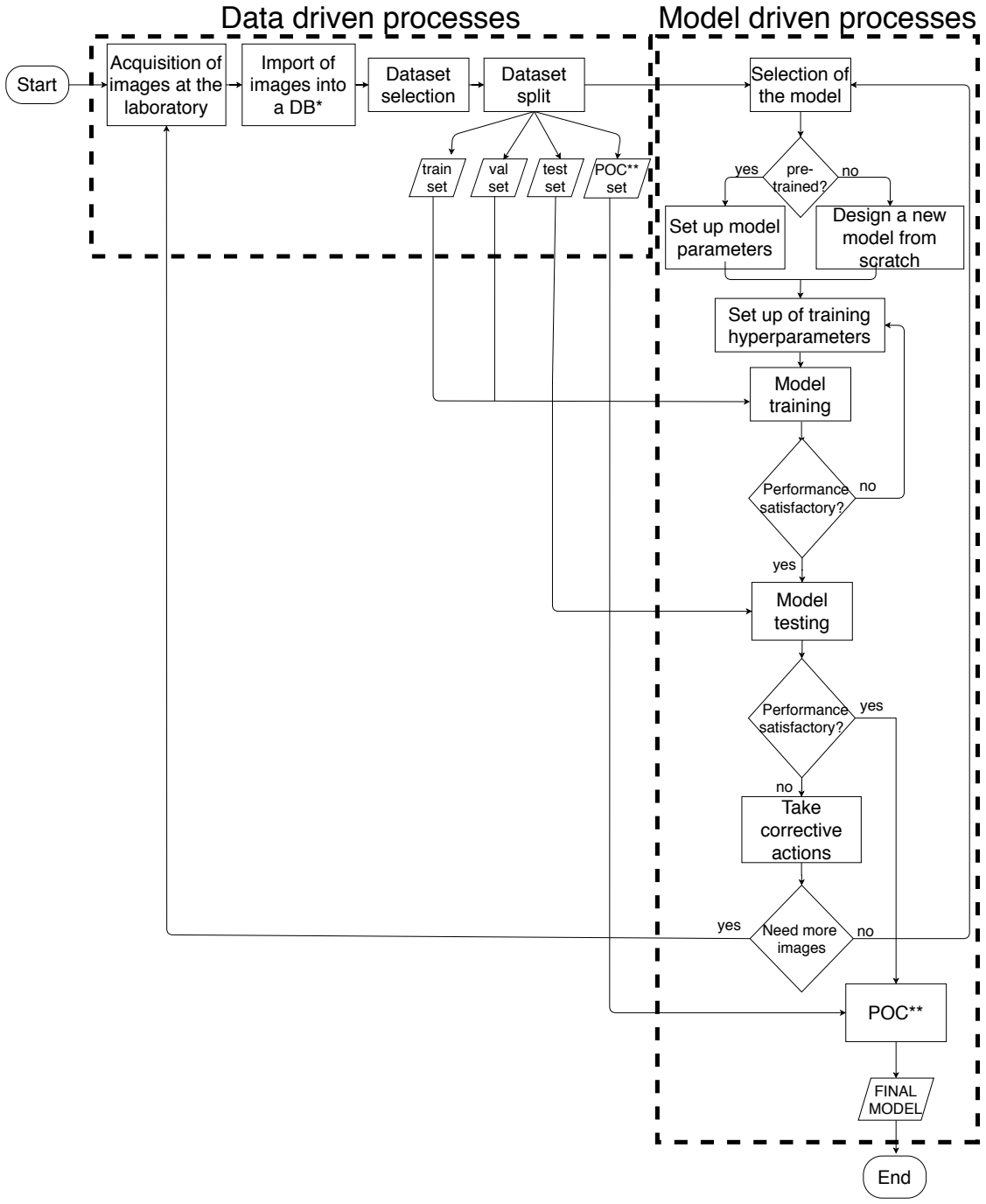


Figure 1.4: *Research methodology used in this thesis. Rectangles represent the principal steps in the methodology, parallelograms are outputs resulting from some processes and rhombus represent points where a decision must be made. (DB*=database, POC**=proof of concept).*

Acquisition of images in the laboratory: At this first stage blood samples are processed at the Core laboratory of the Hospital Clinic with a standardized quality control system to ensure equal and stable conditions for all samples. As a result, images of individual peripheral blood cells of size 360x363 pixels in *jpg* format are obtained. Each cell image is labeled by clinical pathologists.

Import of images into a Database (DB): The images obtained are anonymzed and imported into a MySQL Database. We designed an indexed the database to store in a systemic and organized manner the information about the cells like the smear date, number of cells and type of cells in each smear, location of image storage, etc. This organization enables the subsequent search for information using different criteria to conform the datasets for training the models.

Dataset selection: The images required to train a model are selected from our DB. Some criteria selection are the disease or entity and the cell type.

Dataset split: Once the dataset is set up, it is split in four groups: train, validation, test and proof of concept sets. With this step the data processing is finished.

Selection of the model: In this step we select the type of model we are going to use. Among the models in this thesis are SVMs and CNNs. If the model selected is pretrained, we have to set up the model parameters for training. If the model is not pretrained, we have to design a model from scratch and initialize its parameters randomly.

Set up of training hyperparameters: For training the models we have to set up some parameters like the number of epochs, as well as select the training algorithms and number of layers to train if the model is a CNN.

Model training: Execute the training and monitor the progress of the performance metrics, usually accuracy and loss. The training time of a model depends on the number of parameters to be trained. The more parameters and more complex the architecture, the

longer time the training will take. Several trains can be executed until the desired accuracy levels are achieved.

Model testing: The objective of this step is to test the model with new images not used previously for the training stage and measure a more real performance of the model. If the accuracy and loss results are satisfactory, we consider the model ready for the proof of concept. If the results are not satisfactory, corrective actions must be taken.

Implement corrective actions: This action is taken based on the analysis of the classification errors obtained. The corrective actions to implement can be mainly of two types: 1) Related to data: this means to go back to the start of the workflow and collect more images to improve the classification results; and 2) Related to models: go back to model selection and follow the workflow from this step.

Proof of Concept (POC): When testing results of the classification model are satisfactory, the POC is performed. The main goal with this final test is to evaluate the performance of the model in a real set up, using smears from individual patients.

1.5 Thesis organization and contributions

Figure 1.5 illustrates the thesis organization. This document can be read in two ways: 1) Following the chapters column (from top to down). This is the same order in which the thesis was developed. 2) Following the objectives column from left to right, the reader may select an specific objective, the chapter where the development and results are explained in detail and their related contribution.

The thesis is conformed by seven chapters. The first chapter is the Introduction where important aspects of the context are presented. It starts with the motivation and justification to perform this research, next, the theoretical framework used for the development of the

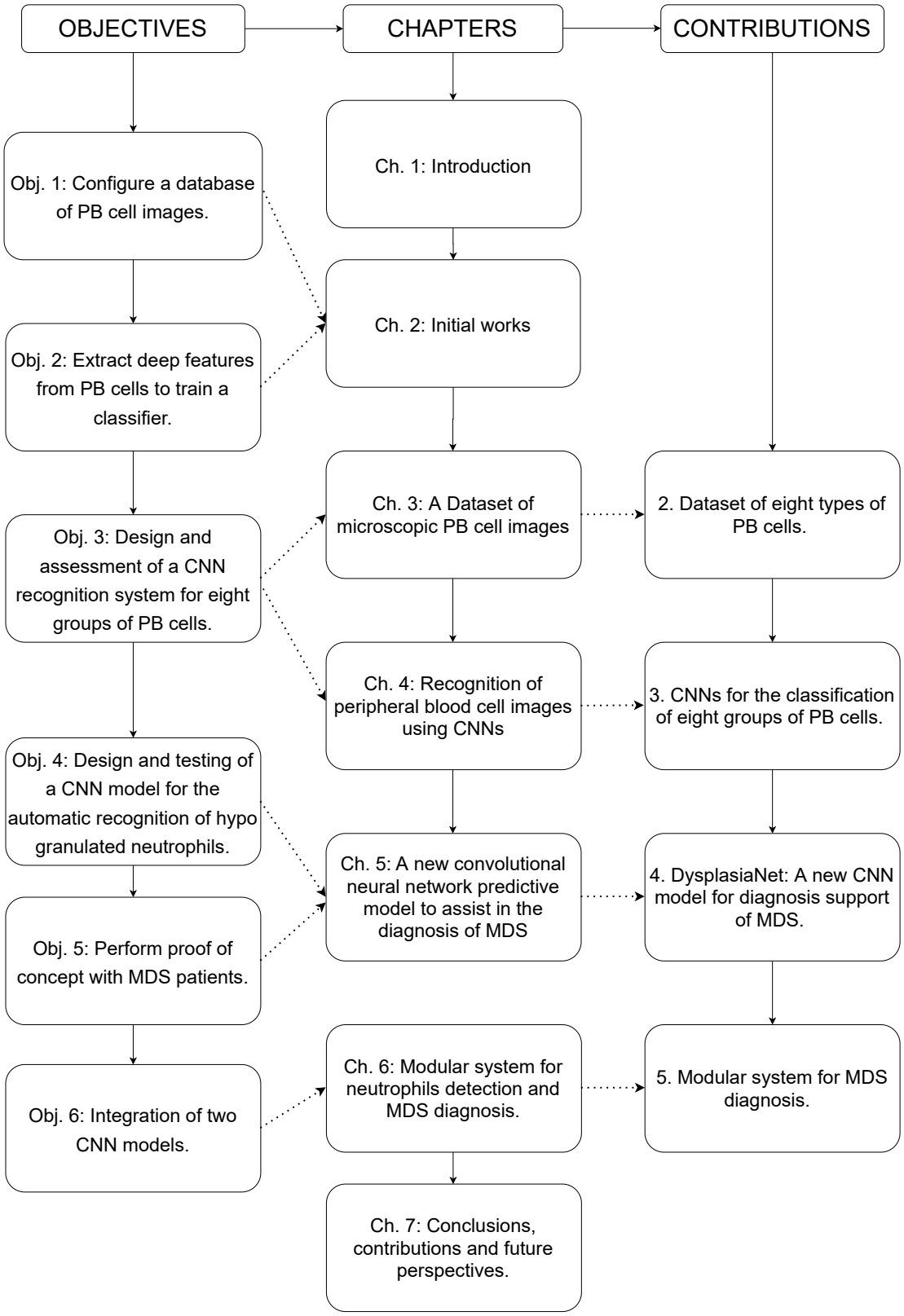


Figure 1.5: *Thesis organization and contributions.*

thesis, following with the objectives that guided the thesis development and finishing with the research methodology implemented. Chapter 2 details the initial works developed to accomplish the first two specific objectives proposed in the thesis. The first work was the design and development of an indexed MySQL Database to store the images of PB cells. The second work was the development of a first model to perform automatic classification of four groups of PB cells using several pretrained CNNs to extract image features to train SVMs. The results obtained were presented at the International Symposium of Biomedical Imaging, ISBI 2017. Chapter 3 describes the dataset of PB cell images compiled for the training and testing of CNNs [19]. The dataset was published in Data in Brief, an open access scientific journal that provides a way for researchers to share and reuse each other's datasets.

Chapter 4 presents the development of CNN models for the classification of eight groups of PB cells [20]. The results obtained were published in Computer Methods and Programs in Biomedicine, a scientific journal classified in the first quartile (Q1) and impact factor of 3.362 (2019) accordingly to the Journal of Citation Reports ranking (JCR). Chapter 5 presents the design and development of a new model for the diagnosis of MDS. The results are being evaluated by the scientific journal Computers in Biology and Medicine, which is classified in the first quartile (Q1) and has an impact factor of 3.434 (2019) according to JCR.

Chapter 6 presents a modular system for neutrophils detection and MDS diagnosis. In the same chapter, we also present a final work towards a modular system for the classification of a wide range of PB cells. Although a global system is out of the scope of this thesis, we have performed a final test to Model 1 using abnormal cells present in serious diseases as leukemia, lymphoma and infections. This is the first effort in integrating the different classification models that are being developed in the research group. Chapter 7 summarizes the conclusions, contributions and some perspectives on future works.

Chapter 2

Initial works [1]

[1] Based upon: Acevedo, A., Alférez, S., Merino, A., Puigví, L., & Rodellar, J. (2017). Automatic recognition system of peripheral blood cell images using deep features. Abstract presented at IEEE International Symposium on Biomedical Imaging – ISBI 2017.

Abstract

This chapter presents the initial works to develop the first two specific objectives proposed in this thesis. The first work, in Section 2.1, is the design of the MySQL Database developed to store information from PB cell images in a structured manner, which are obtained from healthy controls and patients. Also, the process to import the images to the database and some SQL queries to retrieve information, are explained. Section 2.2 presents the development of a first classifier to classify four groups of PB cells using CNNs as feature extractors. A dataset of 22,371 images was selected from the MySQL Database and organized in four groups: Mononuclear cells, granulocytes, platelets and smudge cells. Six pre-trained CNNs were used to extract features from the images to train a SVM to classify these four groups. Total accuracies in all cases were above 94 %.

2.1 Database design

A MySQL Database is a set of tables created to store information in units called registers. The rows of the table are the registers and the columns are the fields of the registers. All registers from the same table have the same number and type of fields. When using a MySQL Database, before storing new data, all the tables and the type of data of the columns must be configured. We use three types of data in the database: VARCHAR, to store data composed by numbers and letters of different sizes, INT to store integers and DATE to store dates.

The DB designed is conformed by three tables: "historias", "entrenamiento" and "testing". The tables, their respective column names and the data type of each column are presented in Figure 2.1. Table "historias" contains two columns. First is *nhc*, which corresponds to patient history. The type of data of this column is VARCHAR and can be of up to 255 characters. The second column is *nhc_aleatorio*, where are stored the new history

numbers of data type INT. This is a random number generated to replace the original history number. The principal objective of this table is to map the original history number to the new random history number assigned and ensure that the new generated number is unique.

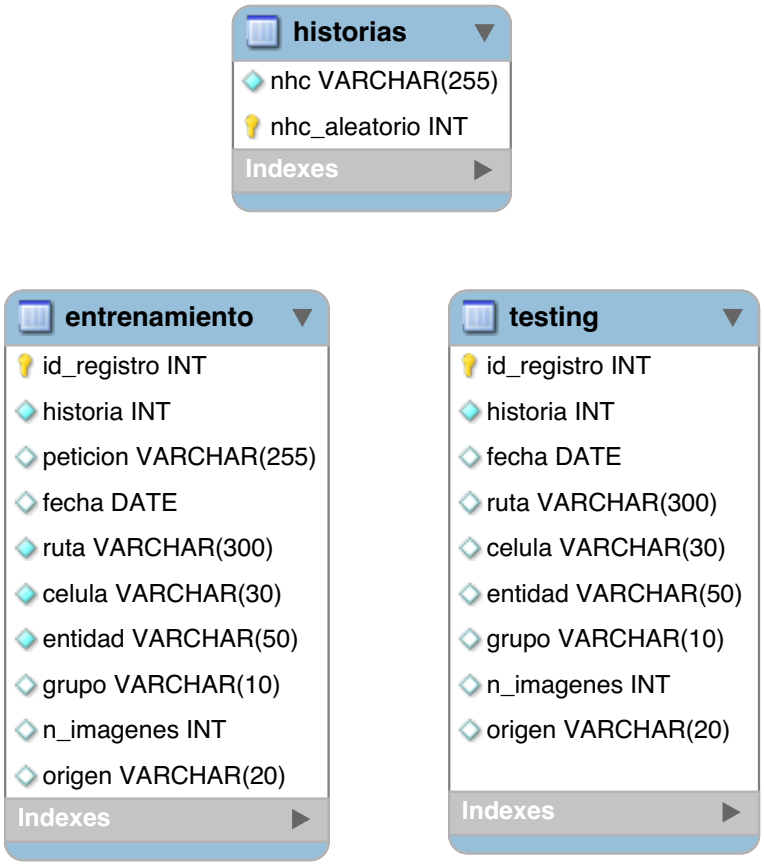


Figure 2.1: *Database tables schema. The database is conformed by three tables: historias, entrenamiento and testing.The columns of each table are also listed.*

Table "entrenamiento" stores the smears used to train Machine Learning (ML) models. There is one register for each type of cell (column celula) in a smear. This table contains 10 columns:

- *id_registro*: random number generated each time a new register is inserted in the table. It is also named as the primary_key used to identify uniquely each register.
- *historia*: random history number assigned to the smear. Corresponds to *nhc_aleatorio*

from "historias" table.

- *peticion*: number generated in the laboratory to identify each smear.
- *fecha*: the date when the smear was generated at the laboratory.
- *ruta*: directory where are stored the images that correspond to an specific smear and type of cell. The physical route to each type of cell has the following structure : grupo/entidad/historia/fecha/celula
- *celula*: cell type. There are 19 cell types in the database among normal and abnormals: monocyte, lymphocyte, eosinophil, neutrophil band, neutrophil segmented, basophil, metamyelocyte, promyelocyte, myelocyte, prolymphocyte, variant lymphocyte, smudge, giant platelets, erythroblasts, atypical lymphocytes, centroblasts, blasts, atypical promyelocyte, plasma cells.
- *entidad*: is the type of entity or disease. There are cells of 13 entities in the database: Normal, Chronic Lymphocytic Leukemia (CLL), Follicular Lymphoma (FL), Hairy Cell Leukemia (HCL), Large Granular Lymphocyte Leukemia (LGL-T), Mantle Cell Lymphoma (MCL), Plasma Cell Leukemia (PCL), B and T Prolymphocytic Leukemia (PL), Splenic Marginal Zone Lymphoma (SMZL), Sézary Sindrome (SS), Infections (CLR), Acute Lymphocytic Leukemia (ALL) and Acute Myeloid Leukemia (AML).
- *grupo*: entity group. Entities are organized in five groups: Normal, CLR , B-cell Chronic Lymphoproliferative Disorder (SLPC-B), T-cell Chronic Lymphoproliferative Disorder (SLPC-T) and Blasts (BLASTS).
- *n_imagenes*: number of images in the directory.
- *origen*: source of the images. The images can be from two sources: CELLAVISION and MICROSCOPE.

Regarding the table "testing", it contains 9 columns with the same data type as of "entrenamiento" table. In this table, smears from different patients are stored and are usually used to test the models developed.

2.1.1 Import process

The import process was designed to store the images and smear information acquired in the laboratory. It is the second step in the research methodology described in Figure 1.4. The information is imported to the MySQL Database by means of an algorithm described in the flowchart in Figure 2.2. The algorithm script was written in Perl language.

The import starts with the connection to the MySQL Database with the proper access credentials. The script reads a excel file which contains information of the smears selected in the laboratory. Each row of the excel contains the following fields with information from one smear: patient history, number of petition, date of the smear and entity or disease. The script checks that the first three fields are not empty. If any of them is empty, the row is discarded and the script proceeds to the next row in file. If these fields are not empty, the script query to "historia" table to check if the patient has been previously inserted in the DB. If not, a new register is inserted in the "historia" table and a new patient history number is generated. If the patient is already in the DB, the *nhc_aleatorio* is retrieved and kept in memory. Next, the script checks that the smear directory exists and moves it to the final location into the directory system of the DB with the following structure: /grupo/entidad/historia/fecha/celula. The number of images in each "celula" subdirectory is counted and the register is inserted in the proper table: training or testing. If the register already exists in the table, it is not inserted and the script follows with the next "celula" sudirectory in the smear. When all the directories corresponding to each type of cell in the smear have been inserted, the script writes a log file to keep track of the inserts and follows with the next smear in the excel file until all lines are processed.

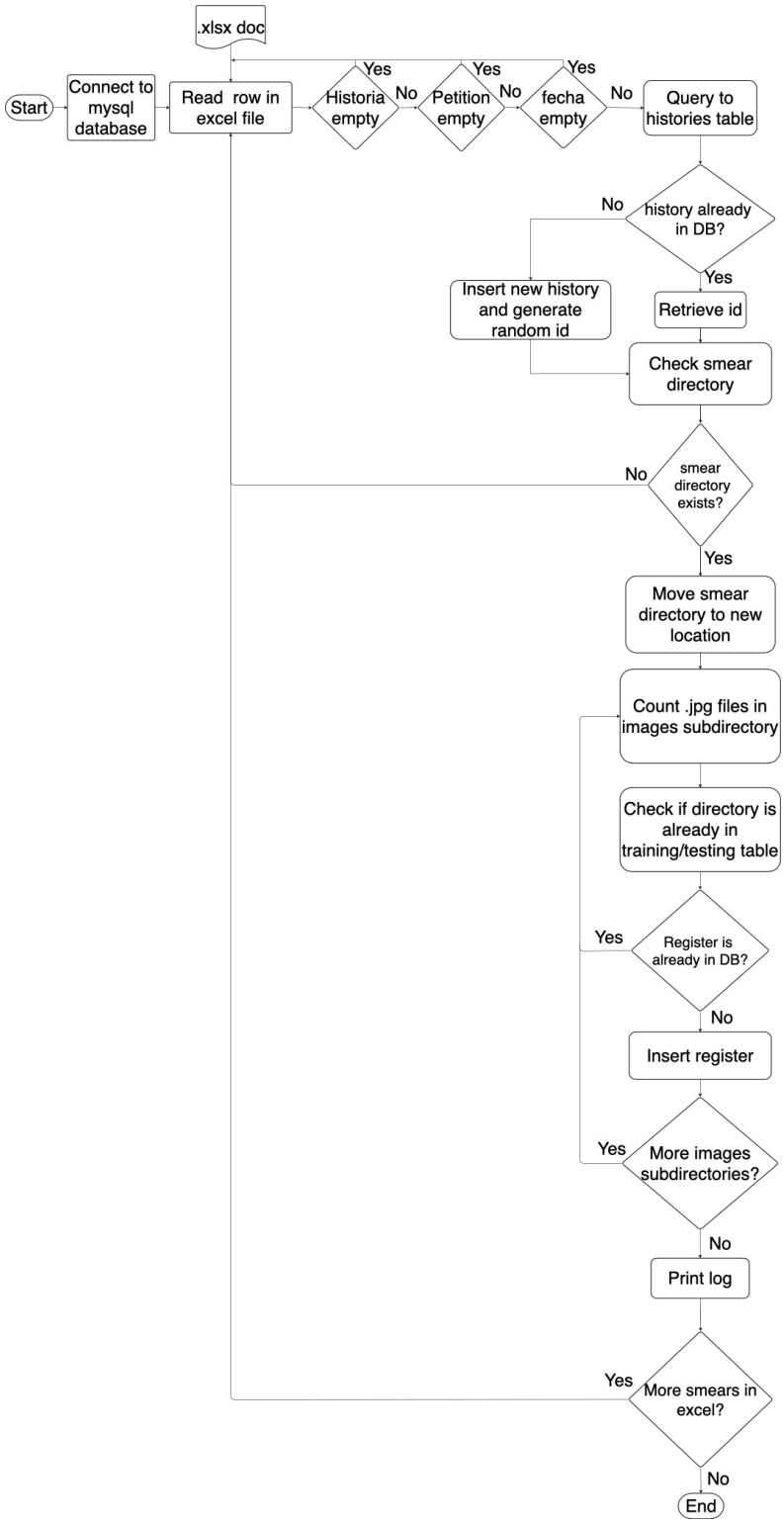


Figure 2.2: *Import algorithm developed to insert into the mysql database new smears selected in the laboratory.*

2.1.2 Database queries

The DB can be consulted using the MySQL workbench. This is a graphical interface to create and interact with MySQL Database. Figure 2.3 presents four important parts of the interface (red numbers). Number one is the query. This is the information we want to retrieve from the database. It should be written in SQL. In the example we query all the registers in "entrenamiento" table whose entity is NORMAL. The tables that conform the database are marked with number two: historias, testing and entrenamiento. The answer to the query, in this case, all the registers that correspond to NORMAL smears in the "entrenamiento" table, are marked with number three. Number four marks the summary of the action performed with information like the query executed, the total of registers consulted and duration time of the query.

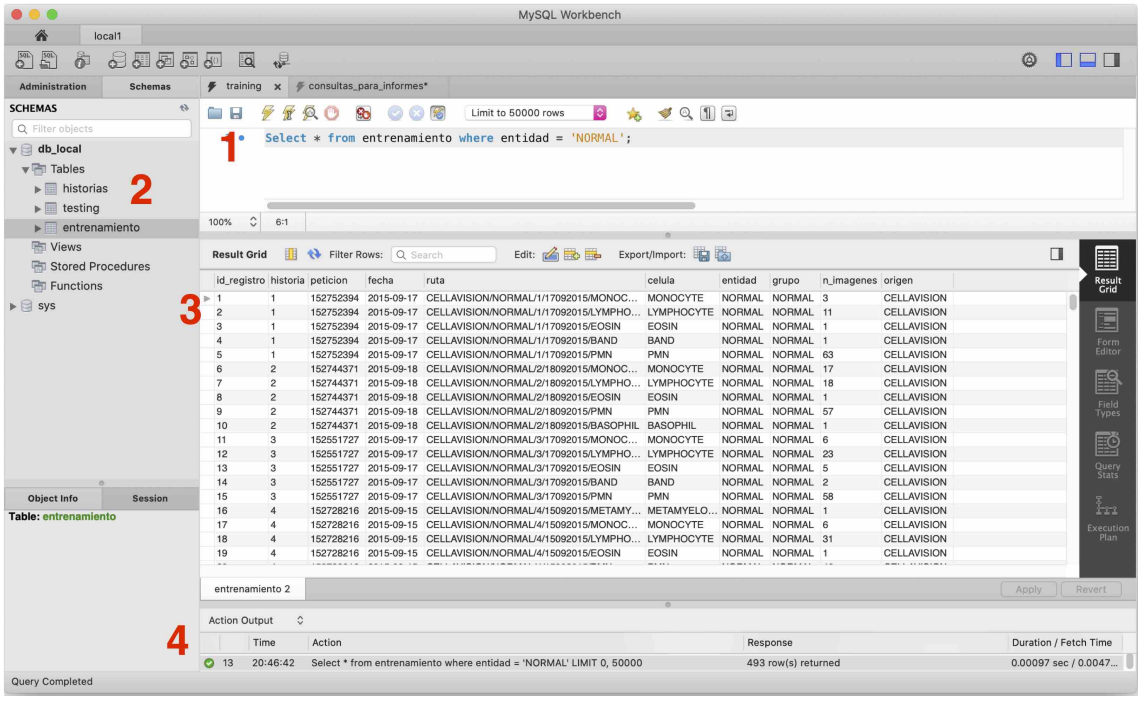


Figure 2.3: *Important features of the workbench interface.*

If it is required to retrieve only normal lymphocyte images for training, for example, we have to write the query presented in Figure 2.4, where we ask for all registers from healthy controls (entidad = NORMAL) and cell type lymphocytes (celula = LYMPHOCYTE).

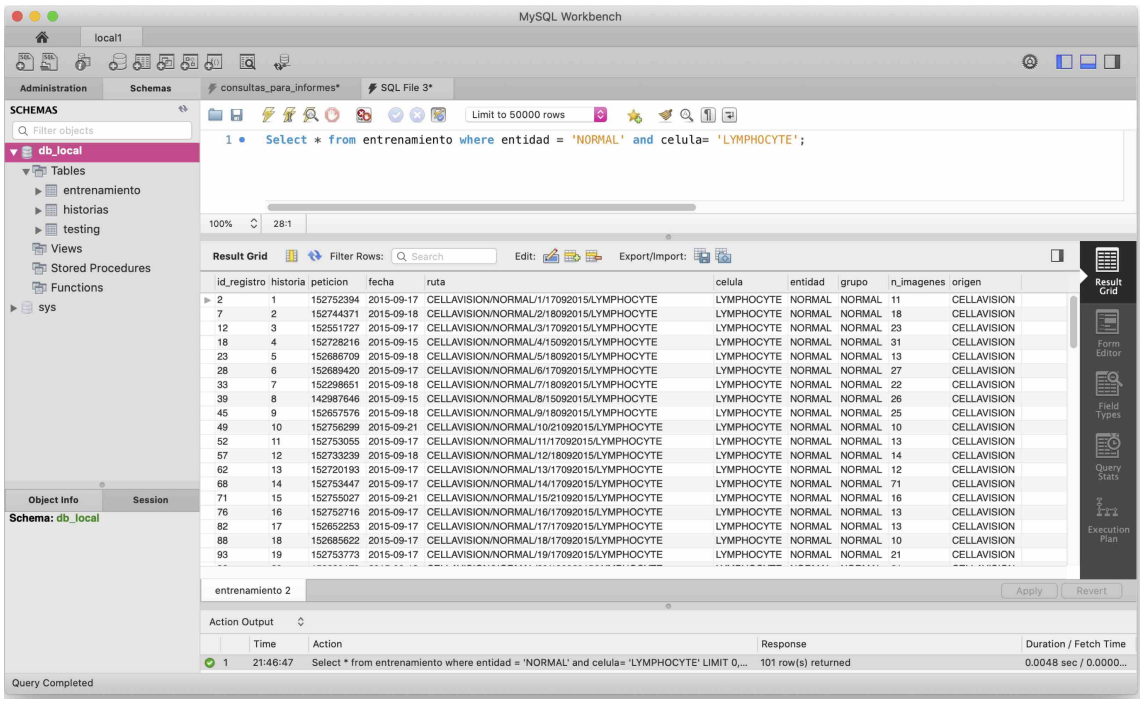


Figure 2.4: *Query example to retrieve only the normal lymphocytes from training table.*

Next, there are some examples of queries to retrieve information from the database:

1. To obtain the total of normal patients:

   Select count(distinct historia) from training where entidad = 'NORMAL';

2. To obtain the number of smears from each normal patient:

   Select historia, count(distinct fecha) from training where entidad = 'NORMAL' group by historia,fecha;

3. To obtain the total of normal lymphocytes:

   Select sum(n_imagenes) from training where entidad= 'NORMAL' and celula= 'LYMPHOCYTE';

4. Obtain the abnormal lymphocytes from follicular lymphoma disease:

   Select * from training where entidad = 'LF' and celula= 'ATYPICAL_LYMPHOCYTE';

2.2 A first classifier of PB images

The second preliminary work was the development of a first classifier for the automatic morphological detection of PB images [21]. A SVM was trained for the recognition of four cells groups: 1) Mononuclear Cells (MNC), 2) granulocytes, 3) smudge cells, and 4) platelets, using deep features extracted using different architectures of CNNs. The MNC group was conformed by abnormal lymphoid and blast cells circulating in PB in malignant diseases such as lymphoid neoplasms and acute leukemias. Granulocytes and platelets are common normal cells that can be encountered in PB. Smudge cells represent abnormally fragile cells that break during the smear preparation. Their observation is frequent in PB in chronic lymphocytic leukemia.

We acquired 22,371 digital cell images from a total of 404 patients at the routine workload of the Core Laboratory of the Hospital Clinic of Barcelona. PB films were stained automatically with May Grünwald-Giemsa and images were acquired using the CellaVision DM96 system. Images were imported to the MySQL Database. The dataset was organized in four groups: MNC (10,982 images including lymphocytes, monocytes, reactive lymphocytes, abnormal lymphocytes, plasma cells, atypical promyelocytes, blast cells and erythroblasts), granulocytes (7,186 images including basophils, eosinophils, neutrophils, band cells, myelocytes, metamyelocytes, promyelocytes), platelets (1,962 images corresponding to giant platelets and aggregates) and smudge cells (2,241 images of isolated cell nuclei). Examples of images of the different groups are shown in Figure 2.5.

We extracted a total of 4,096 features from each image using six pre-trained CNNs: Alexnet [22], vgg-s,m,f [23], vd-16 and vd-19 [24]. Then, with these features we trained a quadratic SVM in Matlab, implementing a 5-fold cross validation.

Results obtained are presented in Table 2.1. The True Positive Rate (TPR) for each group

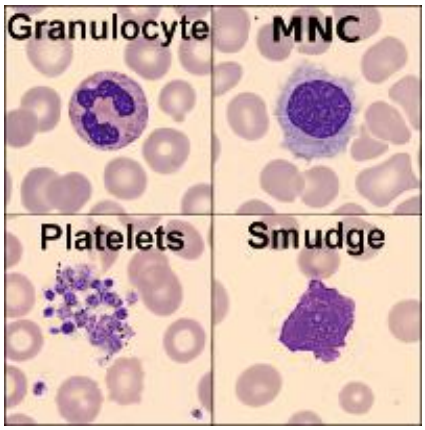


Figure 2.5: *Examples of images of the four groups classified: granulocytes,Mononuclear Cells (MNC), platelets and smudge cells.*

classified is in percentage. Total accuracies for all CNNs were above 94%. In addition, the area under the curve Receiver Operating Characteristic (ROC) was close to 1 for all the analysis.

CNN	***TPR MNC***	***TPR Granulocytes***	***TPR Platelets***	***TPR SC***
Alexnet	98	95	99	93
Vgg-s	98	96	99	94
Vgg-m	98	94	99	95
Vgg-f	98	96	98	93
Vd-16	96	92	98	91
Vd-19	96	92	97	91

CNN: Convolutional Neural Network, MNC: Mononuclear cells, SC: Smudge cells.

Table 2.1: *True Positive Rate (TPR) for each group classified with different CNNs. Total accuracies in all cases were above 94 %.*

2.3 Conclusions

We made a first approach to the classification of PB cells using pre-trained CNN to extract deep features. We obtained high accuracy percentages in the automatic classification without requiring previous segmentation of the regions of interest of the images, which was the predominant direction in the research group. These results were encouraging

towards the development of automatic recognition models for a broader family of cells and with a diagnosis perspective. With this purpose, the immediate next step was to select a larger dataset to increase the number of cell types to be classified. This is explained in Chapter 3.

Chapter 3

A dataset of microscopic peripheral blood cell images for development of automatic recognition systems[1]

[1]Based upon: Acevedo, A, Merino, A, Alférez, S, Molina, Á, Boldú, L, and Rodellar, J. (2020). A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. Data in Brief, 30. https://doi.org/10.1016/j.dib.2020.105474

Abstract

This chapter presents a dataset for the development of automatic recognition systems of PB cell images. The dataset contains a total of 17,092 images of individual normal blood cells, which were acquired using the analyzer CellaVision DM96 in the Core Laboratory at the Hospital Clinic of Barcelona. The dataset is organized in the following eight groups: neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes (promyelocytes, myelocytes, and metamyelocytes), erythroblasts and platelets or thrombocytes. The size of the images is 360 x 363 pixels, in format JPG, and they were annotated by expert clinical pathologists. The images were captured from individuals without infection, hematologic or oncologic disease and free of any pharmacologic treatment at the moment of blood collection.

This high-quality labelled dataset may be used to train and test ML and deep learning models to recognize different types of normal PB cells. To our knowledge, this is the first publicly available set with large numbers of normal PB cells, so that it is expected to be a canonical dataset for model benchmarking.

3.1 Data

The normal PB dataset contains a total of 17,092 images of individual cells, which were acquired using the analyser CellaVision DM96. All images were obtained in the colour space RGB. The format and size of the images is jpg and 360 x 363 pixels, respectively, and were labelled by clinical pathologists at the Hospital Clinic.

The dataset is organized in eight groups of different types of blood cells as indicated in Table 3.1.

Cell type		**Total of images by type**	**%**
neutrophils		3,329	19.48
eosinophils		3,117	18.24
basophils		1,218	7.13
lymphocytes		1,214	7.10
monocytes		1,420	8.31
immature granulocytes	metamyelocytes myelocytes promyelocytes	2,895	16.94
erythroblasts		1,551	9.07
platelets(thrombocytes)		2,348	13.74
Total		17,092	100

Table 3.1: *Types and number of cells in each group.*

Although the group of immature granulocytes includes myelocytes, metamyelocytes and promyelocytes, we have kept all in a single group for two main reasons: (1) the individual identification of these specific subgroups does not have special interest for diagnosis; and (2) morphological differences among these groups are subjective even for the clinical pathologist. Figure 3.1 shows examples of the ten types of normal PB leukocytes that conform the dataset.

3.2 Experimental design, materials and methods

The images were obtained during the period 2015–2019 from blood smears collected from patients without infections, hematologic or oncologic diseases and free of any pharmacologic treatment at the moment of their blood extraction. The procedure followed the daily workflow standardized in the Core Laboratory at the Hospital Clinic of Barcelona, which is illustrated in Figure 3.2.

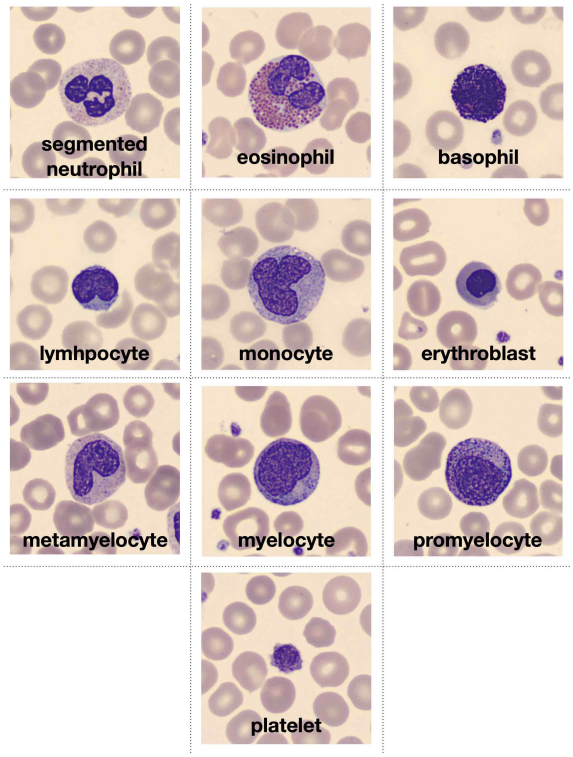


Figure 3.1: *Example images of the ten types of normal blood cells that can be found in the dataset organized in eight groups.*

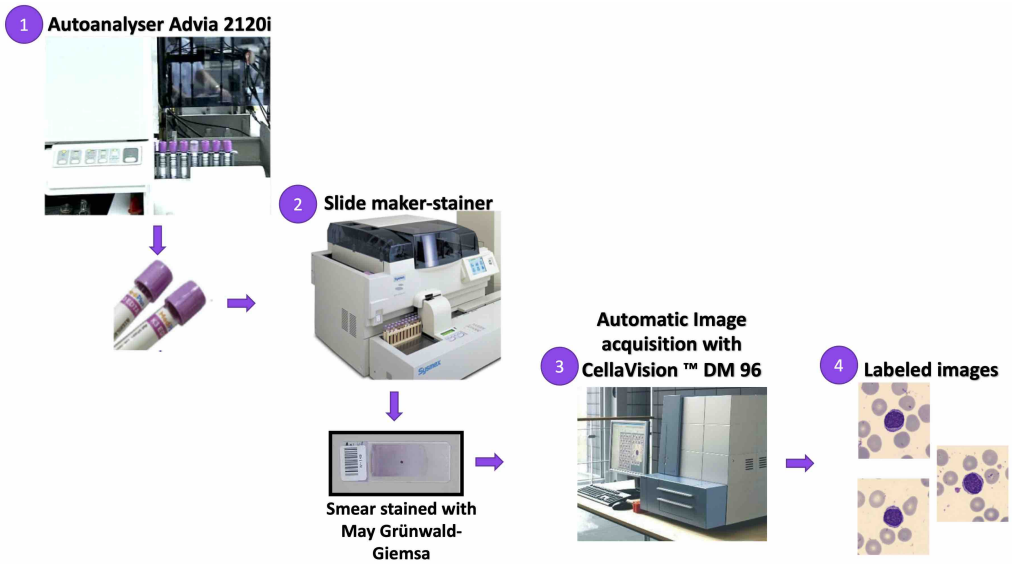


Figure 3.2: *Daily work flow at the Core Laboratory performed to obtain the blood cell images.*

The workflow starts in the Autoanalyzer Advia 2120 instrument where blood samples are processed to obtain a general blood cell count. In a second step, the blood smears were au-

tomatically stained using May Grünwald-Giemsa [25] in the autostainer Sysmex SP1000i. This automated process ensures equal and stable staining regardless of the specific user. The laboratory has a standardized quality control system to supervise the procedure. Then the resulting stained smear goes through the CellaVision DM96 where the automatic image acquisition was performed. As a result, images of individual normal blood cells, with jpg format and size 360x363 pixels, were obtained. Each cell image was annotated by the clinical pathologist and saved with a random identification number to remove any link and traceability to the patient data, resulting in an anonymized dataset. No filter and further pre-processing were performed to the images.

The above acquisition procedure has been extensively used by our research group in several works related to cell image segmentation and classification of PB cells [26]–[30]. The dataset presented in this article has been used in our more recent work to develop a CNN model for the automatic classification of eight types of normal PB cells [20].

3.3 Value of the data

- This dataset is useful in the area of microscopic image-based haematological diagnosis since the images have high-quality standards, have been annotated by expert clinical pathologists and cover a wide spectrum of normal PB cell types.
- The dataset can be useful to perform training and testing of ML and deep learning models for automatic classification of PB cells.
- This dataset can be used as a public canonical image set for model benchmarking and comparisons.
- This dataset might be used as a model weight initializer. This means to use the available images to pre-train learning models, which can be further trained to classify other types of abnormal cells.

3.4 Specifications table

Subject	Hematology
Specific subject area	Computational tools for hematological diagnosis using microscopic cell images and automatic learning methods
Type of data	Images
How data were acquired	Digital images of normal PB cells were obtained from samples collected in the Core Laboratory at the Hospital Clinic of Barcelona. In order to obtain the all blood counts, blood samples were analysed in the Advia 2120 instrument. Next, the smear was automatically prepared using the slide maker–stainer Sysmex SP1000i with May Grünwald-Giemsa staining. Then, the automatic analyser CellaVision DM96 was used to obtain individual cell images with format jpg and size 360 x 363 pixels. Images obtained were labelled and stored by the clinical pathologists.
Data format	Raw
Parameters for data collection	The dataset images were obtained from individuals without infections, hematologic or oncologic diseases and free of any pharmacologic treatment at the moment of their blood collection.

Description of data collection	The images were collected in a 4-year period (2015 to 2019) within a daily routine. Blood cell images were annotated and saved using a random number to remove any link to the individual data, resulting in an anonymized dataset.
Data source location	Institution: Hospital Clinic of Barcelona, Region: Barcelona, Catalonia Country: Spain
Data accessibility	The dataset is stored in a Mendeley repository: Repository name: "A dataset for microscopic peripheral blood cell images for development of automatic recognition systems" Data identification number: 10.17632/snkd93bnjr.1 Direct URL to data: `https://data.mendeley.com/datasets/snkd93bnjr/draft?a=d9582c71-9af0-4e59-9062-df30df05a121`
Related research article	Author's name: Andrea Acevedo, Anna Merino, Santiago Alférez, Laura Puigví, José Rodellar. Title: Recognition of peripheral blood cells images using convolutional neural networks. Journal: Computer Methods and Programs in Biomedicine. DOI: https://doi.org/10.1016/j.cmpb.2019.105020

Table 3.2: *Tables with general specifications of the dataset.*

Chapter 4

Recognition of peripheral blood cell images using convolutional neural networks[1]

[1]Based upon: Acevedo, A, Alférez, S, Merino, A, Puigví, L, and Rodellar, J. (2019). Recognition of peripheral blood cell images using convolutional neural networks. Computer Methods and Programs in Biomedicine, 180. https://doi.org/10.1016/j.cmpb.2019.105020

Abstract

Background and objectives: Morphological analysis is the starting point for the diagnostic approach of more than 80 % of hematological diseases. However, the morphological differentiation among different types of normal and abnormal peripheral blood cells is a difficult task that requires experience and skills. Therefore, in this work we develop a system for the automatic classification of eight groups of peripheral blood cells with high accuracy by means of a transfer learning approach using convolutional neural networks. With this new approach, it is not necessary to implement image segmentation, feature extraction becomes automatic and existing models can be fine-tuned to obtain specific classifiers.

Methods: A dataset of 17,092 images of eight classes of normal peripheral blood cells was acquired using the CellaVision DM96 analyzer. All images were identified by pathologists as the ground truth to train a model to classify different cell types: neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes (myelocytes, metamyelocytes and promyelocytes), erythroblasts and platelets. Two designs were performed based on two architectures of convolutional neural networks, Vgg-16 and Inceptionv3. In the first case, the networks were used as feature extractors and these features were used to train a support vector machine classifier. In the second case, the same networks were fine-tuned with our dataset to obtain two end-to-end models for classification of the eight classes of blood cells.

Results: In the first case, the experimental test accuracies obtained were 86 % and 90 % when extracting features with Vgg-16 and Inceptionv3, respectively. On the other hand, in the fine-tuning experiment, global accuracy values of 96 % and 95 % were obtained using Vgg-16 and Inceptionv3, respectively. All the models were trained and tested using Keras and Tensorflow with a Nvidia Titan XP Graphics Processing Unit.

Conclusions: The main contribution of this paper is a classification scheme involving a

convolutional neural network trained to discriminate among eight classes of cells circulating in peripheral blood. Starting from a state-of-the-art general architecture, we have established a fine-tuning procedure to develop an end-to-end classifier trained using a dataset with over 17,000 cell images obtained from clinical practice. The performance obtained when testing the system has been truly satisfactory, the values of precision, sensitivity, and specificity being excellent. To summarize, the best overall classification accuracy has been 96.2 %.

4.1 Introduction

PB is a fluid that circulates through the blood vessels and contains three main cell types suspended in plasma: erythrocytes (red blood cells), leukocytes (white blood cells) and platelets (thrombocytes). Specifically, leukocytes are nucleated cells that play an important role in defense against infections. They are classified as granulocytes (segmented and band neutrophils, eosinophils and basophils), lymphocytes and monocytes [31]. Granulocytes represent 54-62 % of the total leukocytes in PB. Its nucleus is divided into segments (from 2 to 5) joined by a thin filament of nuclear material. In the band neutrophils the nucleus is not segmented and usually the normal percentages in blood are between 0 and 6 %. Eosinophils constitute 1-3 %, basophils 0-1 %, lymphocytes 25-33 % and monocytes 3-10 % of the leukocytes circulating in PB [4]. In several situations (infections, regenerative anaemias or others) it is possible to find in PB some immature granulocytes (ig) (promyelocytes, myelocytes and metamyelocytes) or erythroid precursors, such as erythroblasts. Examples of these types of cells and some of their most important morphological characteristics are shown in Figure 4.1.

The morphological analysis of the different types of cells in PB is the starting point for the diagnostic approach of more than 80 % of the hematological diseases. Currently, in the market there are several systems for the automated pre-classification of normal leukocytes

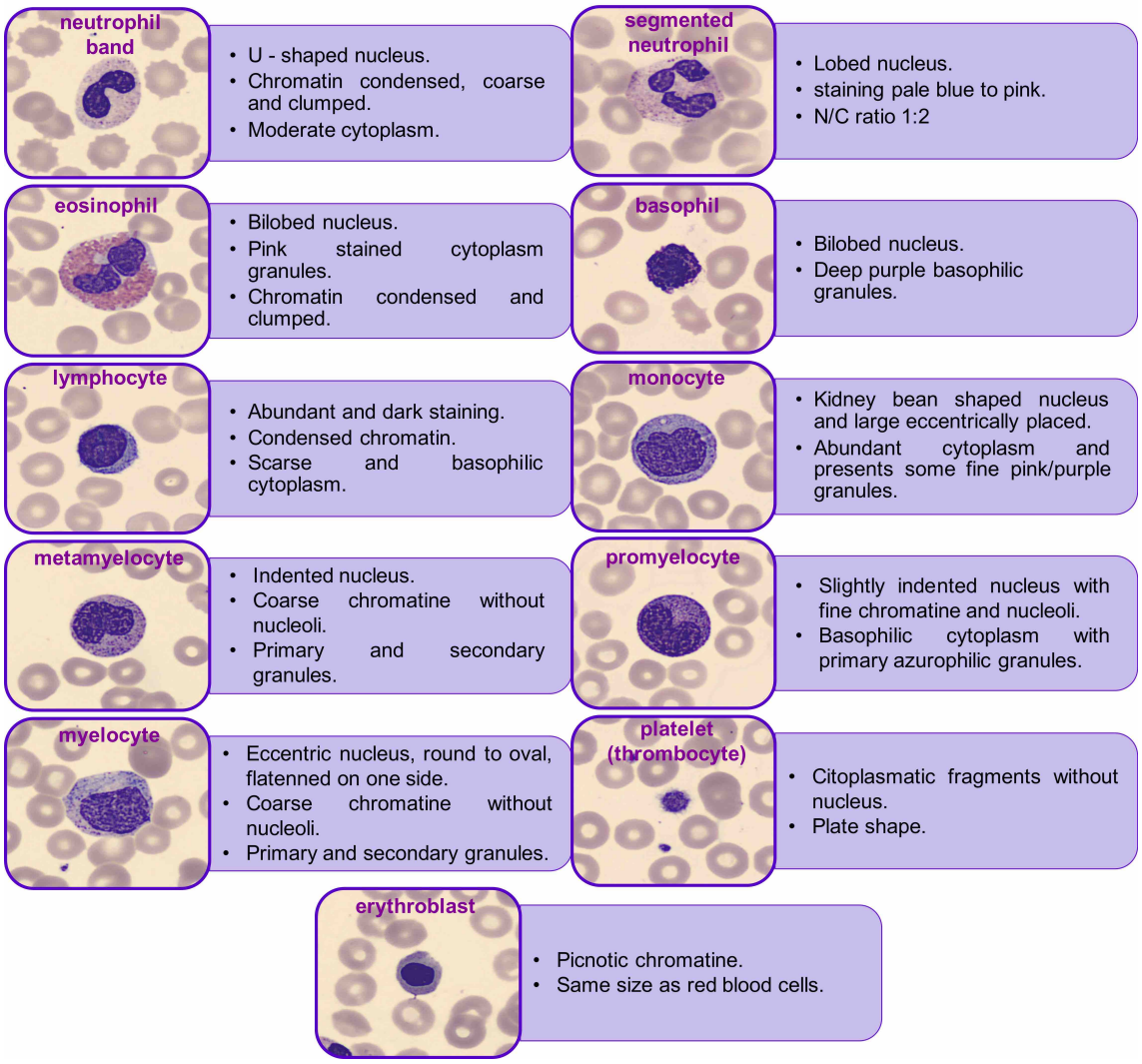


Figure 4.1: *Images of different types of leukocytes: neutrophil, eosinophil, basophil, lymphocyte, monocyte and immature granulocytes (metamyelocyte, myelocyte and promyelocyte), erythroblasts and platelets (thrombocytes) that can be found in peripheral blood and some of their most important morphological characteristics.*

[32]–[34]. These analyzers are based on a motorized microscope to acquire the images from a blood smear and algorithms that perform the pre-classification of the blood cell images. These systems have a high cost due to the complexity of the acquisition and the classification steps. Usually, the classification process using image analysis requires different steps, such as segmentation, feature extraction, selection and classification of the images using algorithms for pattern recognition [29], [30], [35]. The segmentation is considered the most critical step in the process, mainly due to the complex morphology and subtle differences in the wide class of blood cells. It has a direct effect on the accuracies

of the classification results [27], [36]. Trying to overcome some of the limitations of traditional computer vision techniques, new methods known as deep learning have emerged. This is a sub-field of machine learning, whose approach is learning representations from data using models called neural networks, which are structures arranged in layers [8] containing descriptive features. The 'depth' of the model depends on how many layers make up the network. The most important characteristic of these networks is that engineers do not design these layers of features. Conversely, they are learned from data using different learning algorithms in which the segmentation of the images is not necessary. In recent years, different studies using one type of deep neural network known as convolutional neural networks (CNN) have shown good results in image classification [37].

The main objective of this work was to design a CNN-based system for the automatic classification of eight classes of peripheral blood cells: neutrophils, eosinophils, basophils, lymphocytes, monocytes, ig (metamyelocytes, myelocytes and promyelocytes), platelets and erythoblasts. To our knowledge, a convolutional network has not been designed and implemented before for the recognition of this high number of blood cell types achieving an overall accuracy over 96% as obtained in this work. The practical clinical interest is twofold: first, they include the family of normal leukocytes and the cells more frequently observed in infections and regenerative anaemia; and second, the experience gained in this work will be the basis for further extensions of CNNs to classify the broader classes of abnormal cells related to acute leukemia and lymphoma, such as myeoloblasts, lymphoblasts, abnormal B or T lymphoid cells and dysplastic cells.

In this chapter, we present the full training process and experimental test results considering two different architectures of CNN's already pre-trained with the ImageNet dataset [38]: Vgg-16 [24] and Inceptionv3 [39]. In a first step, a transfer learning approach is followed using the networks as feature extractors [40] and then, these features are used to train a SVM classifier. In an alternative second stage, both networks are modified and tuned with the dataset presented in Chapter 3, to obtain two end-to-end models for

classification of the eight classes of blood cells.

After a summary of the most related work in Subsection 4.1.1, the chapter is organized as follows: Section 4.2 describes the data set used in the study and Section 4.3 describes the CNN methodological background. The proposed classification system is presented in Section 4.4, giving details on the architectures and the training procedures. The experimental assessment is described in Section 4.5 and their results and practical issues are then discussed in Section 4.6. Finally, Section 4.7 highlights the most relevant conclusions of this work.

4.1.1 Related work

The automatic classification of PB cells has been studied for several years in order to create tools that support the work on the clinical laboratories. The traditionally used technique is mainly based on the segmentation and extraction of handcrafted features that are then used together with pattern recognition tools for the classification of the different blood cells. In [41], [42] the authors present a review of the state-of-the-art methods of leukocyte segmentation, feature extraction and classification published in the last two decades. As a result, the authors identified several aspects of improvement with respect to the classical approach, such as, the accuracy percentages obtained so far, and the lack of datasets containing a greater number of images to consider sufficiently reliable the classification models obtained. Due to the development of more efficient algorithms and methods related to machine learning, their use have become popular, more specifically those of deep learning and neural networks in the medical field [43]–[45]. In [46] the authors performed the classification of five types of normal PB cells using two different approaches: first, the traditional segmentation, extraction and classification steps using a SVM and Pricipal Components Analysis (PCA) for feature selection; and second, the novel approach using a CNN and the entire image as input. Better performances (85 %)

were obtained with the CNN. In [47] Shahin et al. also proposed the use of CNN for the classification of five types of PB cells. They performed fine-tuning of Alexnet and LeNet-5 networks achieving accuracies of 91.2 % and 84.9 % respectively. They also proposed a trained model with a dataset of 2,551 images for the classification of five types of PB cells and reported a maximum testing accuracy of 96 %.

4.2 Materials

The dataset used in this work is explained in detail in Chapter 3. Table 4.1 details the number and percentages of the different types of blood cell images.

Cell type		**Total of images by class**	**%**
neutrophils		3,329	19.48
eosinophils		3,117	18.24
basophils		1,218	7.13
lymphocytes		1,214	7.10
monocytes		1,420	8.31
immature granulocytes	metamyelocytes myelocytes promyelocytes	2,895	16.94
erythroblasts		1,551	9.07
platelets(thrombocytes)		2,348	13.74
Total		17,092	100

Table 4.1: *Dataset organized in eight classes: neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes, erythroblasts and platelets.*

In order to balance the dataset, we performed data augmenting to increment the number of images and obtain the same number per group. Individual images were modified applying randomly transformations as vertical and horizontal flips, rotation from 0 to 60 degrees and pixel value standardization [12]. Up to four different versions of each of the original images were obtained from the groups with less number of images (basophils and lymphocytes) as shown in Figure 4.2.

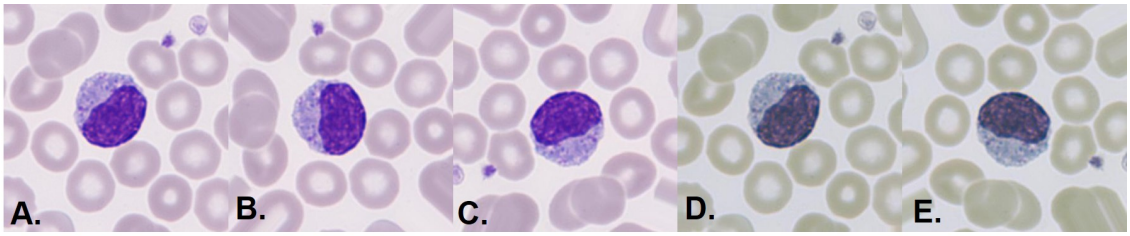


Figure 4.2: *Examples of some of the transformations applied to augment the number of training images. The lymphocyte (A) is the original image, (B) is a rotated version, (C) is a flipped and rotated image, (D) is a version with pixel values standardized and (E) is pixel standardized and a rotated image.*

Regarding the network architectures, we selected two different architectures of CNNs pre-trained with other images different from blood cells, named Vgg-16 [24] and Inceptionv3 [39]. The networks were trained using Keras and Tensorflow software libraries [12] and a Nvidia Titan XP GPU.

4.3 Methodological background

4.3.1 Overview

The main objective of this work is to design a system for the automatic classification of eight classes of peripheral blood cells using CNN's as the background tool for processing and extracting information. Images of PB cells are the input to the system and the output should be the correct label indicating the class of each cell image. Figure 4.3 shows its general structure. We considered two alternative approaches to design the classifier. In the first case, the CNN is used as a feature extractor and then a SVM is trained with these features. This serves as an exploratory search to find out if the use of convolutional neural networks is suitable to solve our classification problem. The second approach is based on the concept of fine tuning, where the CNN is re-trained with PB cell images to obtain an end-to-end classifier. Our hypothesis is that if the results of the classification are good with the approach one, they may improve when implementing the second approach. In

any case, the system's output is a set of eight probabilities (one per class), such that the highest probability indicates the group which the input image belongs to.

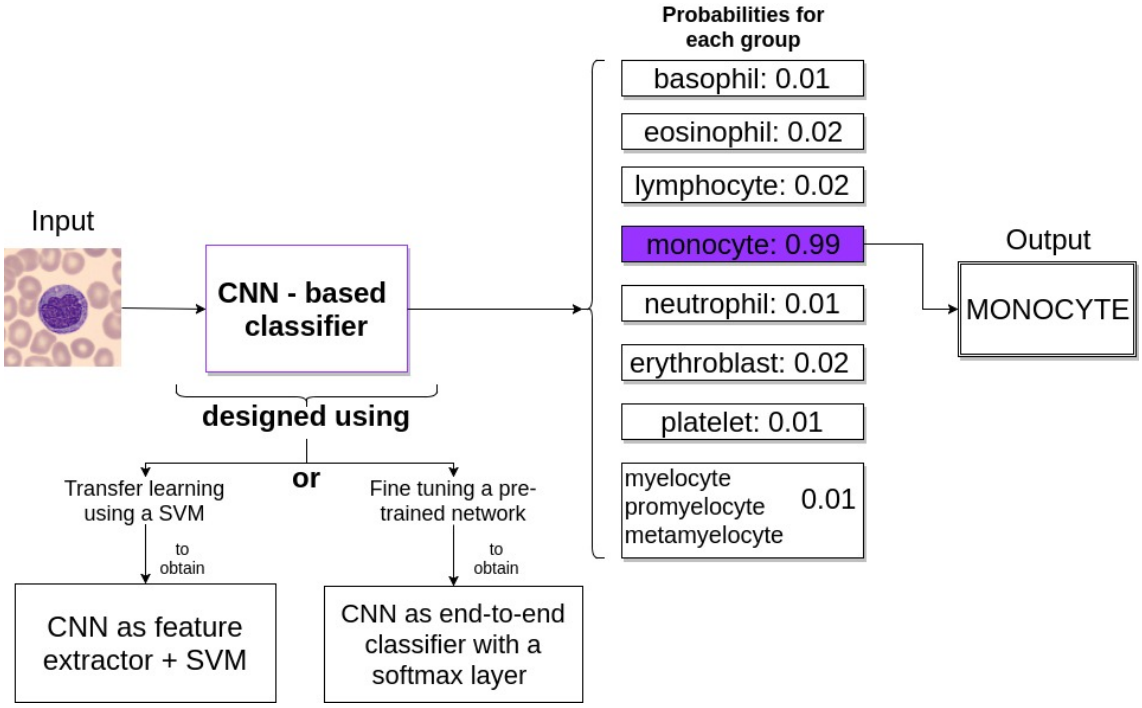


Figure 4.3: *Overall structure of the system for the classification of eight types of peripheral blood cells using convolutional neural networks as a principal tool for processing and extracting information from the images. *SVM: support vector machine. *CNN: Convolutional neural network.*

The purpose of using these two approaches is to take advantage of the knowledge of a CNN previously trained with a large dataset in a specific domain and use it for the classification of images in a different domain [48]. This helps to overcome the deficit of training examples and serve as an effective weight initialization technique for these deep neural networks [49].

CNNs are multilayered networks that can learn complex and high-dimensional features from a large collection of examples. Unlike traditional computer vision models where hand-crafted features are extracted and used to train a classifier, in this approach the model can learn to extract relevant features and use a fully connected structure for classification purposes. The use of CNNs for image classification presents many advantages, such as the efficient use of memory and the capacity of processing large images due to the use of

connections of single neurons to specific portions of an image. In addition, these trained networks present invariance with respect to translation or distortions of the input image and can take advantage of the high correlation between nearby pixels of an image and combine them to identify simple patterns as edges and corners or more complex features. Based on the model presented by Lecun et al in [50], Figure 4.4 shows the basic layered structure of a CNN. The first layer is the input, which reads the pixels of the image. The following is the convolutional layer, where a number of convolution operations are performed over the image to extract the features. Its output is a set of feature maps. A CNN may contain one convolutional layer or a group of consecutive convolutional layers called convolutional blocks. After a convolutional layer or block, a pooling layer is usually placed to reduce the dimensions of the feature maps. Subsequent layers are combinations of convolutional and pooling layers to extract more complex features. After the convolutions and poolings, a set of fully connected layers is placed to combine all the features. Finally, the output is calculated in the last layer by applying a softmax operation to obtain the result of the classification as probability values.

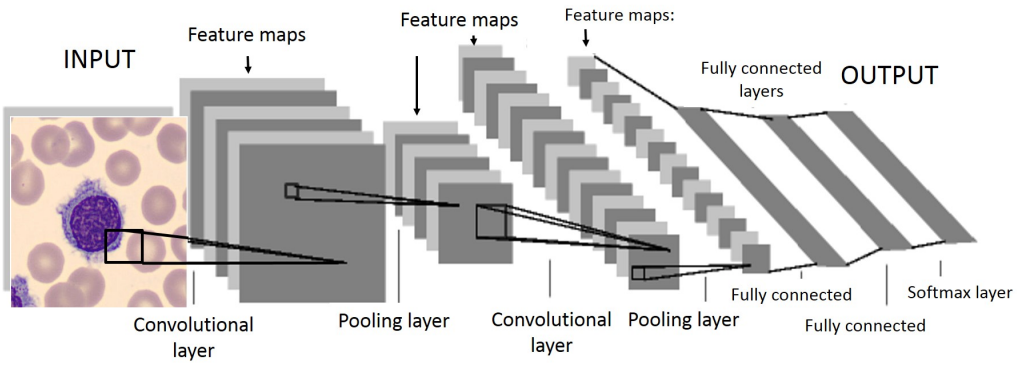


Figure 4.4: *Basic structure of a convolutional neural network. [50]*

4.3.2 Architectures for peripheral blood cells classification

As it was stated before, we implemented two CNNs as a base for the design of our classification system: Vgg-16 and Inceptionv3. The Vgg-16 is a deep convolutional neural network for object recognition designed and trained by the Visual Geometry Group (Vgg)

of the Oxford University in 2014 [24]. Despite its around 160 million of parameters, this network is known for its good performance and simple architecture. On the other hand, the network Inceptionv3 was developed by the Google research team in 2015 and overcomes the high computational cost of convolutional networks as Vgg, introducing more complexity to the model by implementing convolutions in parallel, organized in modules called Inception [39]. This resulted into a more complex architecture but with less parameters, diminishing the execution time and making possible to implement the Inception network in cases where memory or computational capacity is limited.

Next, we describe the main components of the system that we trained for PB cell classification.

The convolutional layer

The convolutional layer is the most important component in a CNN. It is a structure that transforms an input to an output through a convolution-based procedure. Both input and output are seen as a volume structure. The input volume is composed of planes with fixed height and width. The number of planes defines the depth (D). As illustrated in Figure 4.5 (A), when the input is a RGB image, we have $D = 3$ and each plane contains the corresponding component pixel matrix.

The convolutional layer has also a volume structure with a number N of planes, and each plane is defined by a number of filters (kernels) of small size (F). A typical case is, for instance, $F = 3$ so that the filter is an 3×3 matrix with weights. Let us consider the kth input plane with value matrix X_k. Consider also the corresponding filter in the jth plane of the convolutional layer, whose parameters (weights) are arranged in a matrix W_{kj}. The convolution of this filter sliding over the kth input plane produces a two-dimensional matrix $W_{kj} * X_k$. Figure 4.5 (B) illustrates an example of the convolution of a filter of size 3×3 with the input image. These matrices are summed for all the input planes and biased

by a matrix B_j and the result is transformed by an activation function f as follows [51]:

$$Y_j = f\left(B_j + \sum_{k=1}^{D} W_{kj} * X_k\right); \quad j = 1,\ldots,N \tag{4.1}$$

The resulting matrix Y_j is called feature map. The final output of the convolutional layer is a volume whose planes are the N feature maps.

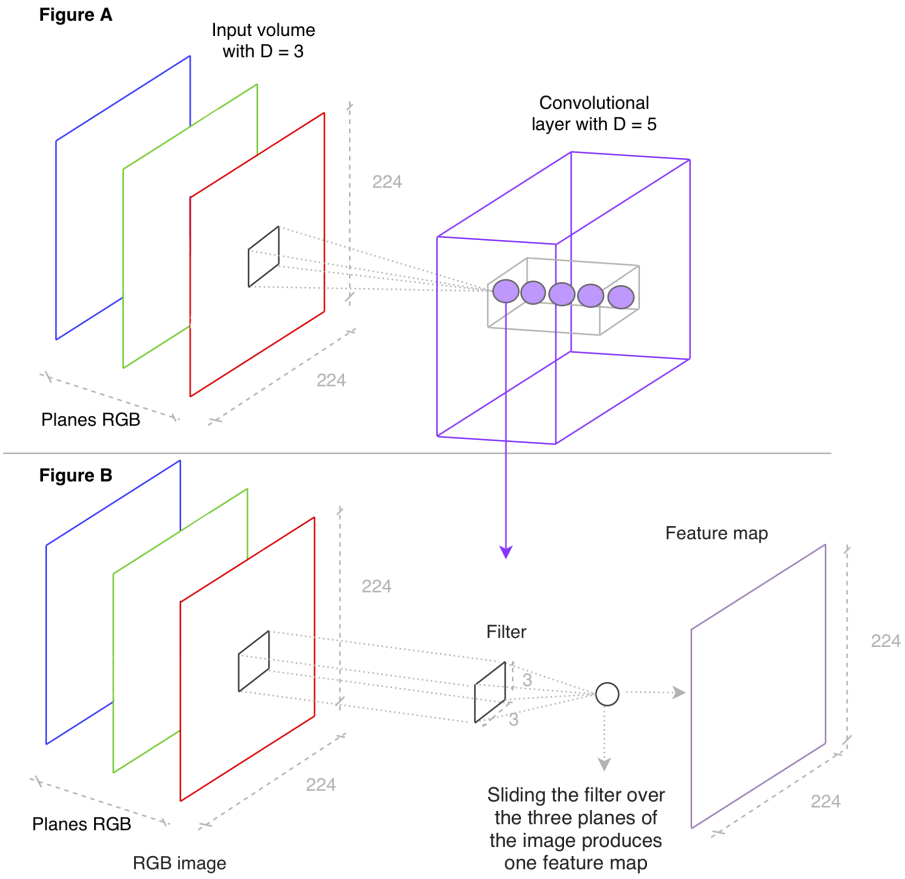


Figure 4.5: *Convolutional layer structure. Figure A presents the local connectivity of a neuron with a portion of the input image. Figure B is equivalent to Figure A but planes contain filters which are convoluted with the input image. The result of the convolution of each filter is a feature map [52].*

The Rectifier Linear Unit (ReLU) is the activation function mostly used in the current CNN architectures. It converts all negative input values to zero while keeping all positive inputs the same, being defined as follows:

$$f(x) = max(0, x) \tag{4.2}$$

This ReLu function is applied to the output of a convolutional layer, and it helps to increase the learning speed of the network and the classification accuracy.

The pooling layer

After each convolutional layer or block of layers, a pooling layer is usually placed to reduce the feature map size. This reduction preserves important information while eliminating irrelevant details about the position of the features within the input images, reducing the sensitivity of the model to shifts and distortions.

Pooling is defined essentially by two parameters: (1) the size (F) of the squared portion of the feature map whose pixels are pooled in a single value, and (2) the stride (S), which indicates the steps along both width and height that pooling moves to cover the full map.

The Vgg-16 has in total five pooling layers with parameters $F = 2$ and $S = 2$. This means that the pooling operation is applied every 2×2 pixels with steps of two pixels over the whole feature map, keeping the maximum value of each frame (max pooling). The Inceptionv3 case is different because it has many more pooling layers. This network has five max pooling layers, four of them with $F = 3$ and $S = 2$ and one with $F = 3$ and $S = 1$. In addition, this network has eight average pooling layers with F=3 and S=1 and a last average pooling layer with F=8 and S=1. Unlike max pooling, the average pooling substitutes a frame of pixels by its average value.

Table 4.2 presents a summary of the implemented architectures of Vgg-16 (left) and Inceptionv3 (right). The Vgg-16 network is composed by 13 convolutional layers organized in blocks. Each block may contain two or three convolutional layers, whereas the Inceptionv3 is composed by five convolutional layers and 9 Inception modules [39], which

contain between four and 10 convolutional layers each. The last layers of both networks are fully connected with 1000 nodes for the 1000 class classification of the Imagenet contest.

Layers	Filter size	Number N of planes (filters)
2xconv2D	3x3	64
Max pooling: F=2 and S=2		
2xconv2D	3x3	128
Max pooling: F=2 and S=2		
3xconv2D	3x3	256
Max pooling: F=2 and S=2		
3xconv2D	3x3	512
Max pooling: F=2 and S=2		
3xconv2D	3x3	512
Max pooling: F=2 and S=2		
4096 nodes fully connected		
4096 nodes fully connected		
1000 nodes fully connected with softmax activation		

Layers	Filter size	Number N of planes (filters)
conv2D	3x3	32
conv2D	3x3	32
conv2D	3x3	64
Max pooling: F=3 and S=2		
conv2D	1x1	80
conv2D	3x3	192
Max pooling: F=3 and S=2		
3x inception modules		
4x inception modules		
2x inception modules		
Average pooling: F= 8		
1000 nodes fully connected with softmax activation		

Table 4.2: *Architecture of Vgg-16 (left) and Inceptionv3 (right).*

The fully connected and output layers

After the successive convolutions and pooling layers, a set of feature maps is obtained. They are a useful representation of the input images. At this point, it is possible to take two approaches to use these features. The first is to use them to train a different classifier such as SVM, k-NN, decision trees and others. The second possibility is to couple a block of fully connected layers to learn non-linear combinations of the features and to perform the final classification. Each neuron of a fully connected layer performs an input-output operation defined as follows:

$$Z = f\left(b + \sum_{i}^{m} \omega_i x_i\right) \tag{4.3}$$

where x_i are the input features, ω_i are the neuron weights, m is the number of features, and b is a bias parameter. Function f is the same activation function described in equation

4.2. The output Z is a new feature that serves as input to the next fully connected layer. The output layer is the last layer of the network. It is also a fully connected configuration but the activation function used is a softmax expressed as follows:

$$f(y_j) = \text{softmax}(y_j) = \frac{e^{(y_j)}}{\sum_{k=1}^{8} e^{(y_k)}} \quad for\ j = 1, \ldots, 8 \tag{4.4}$$

This function is applied to the result of each node of the output layer to obtain the results as a probability distribution. In our design, we have eight output neurons because we aim to classify eight classes of cell images.

4.4 Design and training of the proposed system

In the next sections we describe the two approaches taken to design and train our classifier: the transfer learning using a pre-trained CNN as a feature extractor and the fine tuning of a pre-trained CNN to build an end-to-end classifier.

4.4.1 Transfer learning using a CNN as feature extractor

In this approach the dataset was split into two subsets: training and testing. We selected 15,173 (88 %) images for the training set and 1,919 (12 %) images for the test set. The training set was augmented by applying random transformations, as explained in Section 4.2, to obtain a new training set with a total of 4,608 images per class. To extract the features of the training set, we used the Vgg-16 and Inceptionv3 with their original weights of the Imagenet contest. From Vgg-16, the fully connected layer and the output layer were removed in order to extract $4,096$ features for each image. From Inceptionv3, the fully connected and the output layers were removed to extract $2,048$ features per image.

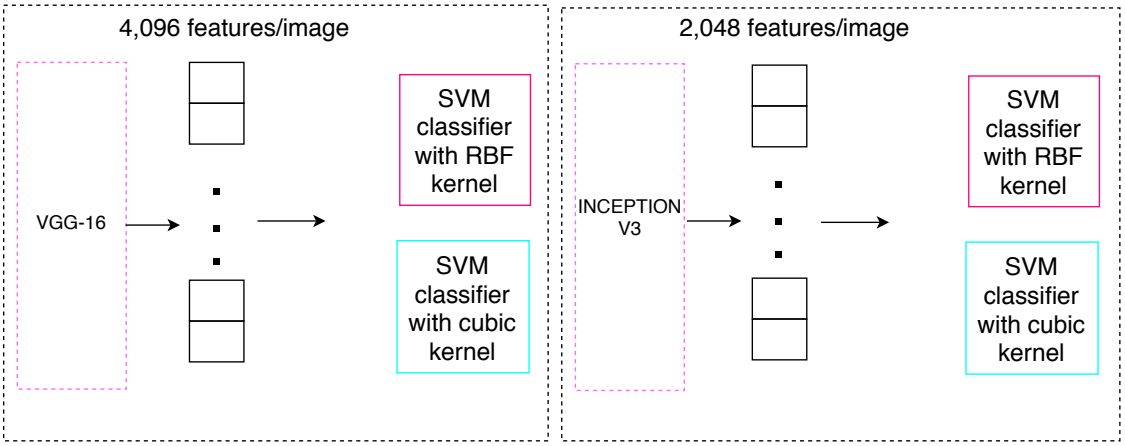


Figure 4.6: *Transfer learning.* $4,096$ *features per image from the Vgg-16 and* $2,048$ *features per image from the Inceptionv3 were extracted.*

We selected two types of SVM kernels : 1) a radial basis function (RBF) kernel, which we have implemented in previous works with satisfactory results [27]; and 2) a cubic polinomial kernel. See Figure 4.6.

To train the SVMs, we implemented a 5-fold cross validation, in which the training is performed through five iterations where the training set is divided automatically into five subsets. In each iteration, one subset is used as a validation set and the other four are used for training. The accuracy of the model is averaged over all the iterations. The training was implemented using Keras and Scikit-learn software libraries [53].

In Figure 4.7 we present the training and validation accuracies for each iteration of the resulting four models. In view of these results, we decided to select the SVM with cubic kernel trained with both Vgg-16 and Inceptionv3 features. Validation average accuracies were 0.87 and 0.88, respectively. The resulting models will be used later on in the experimental assessment.

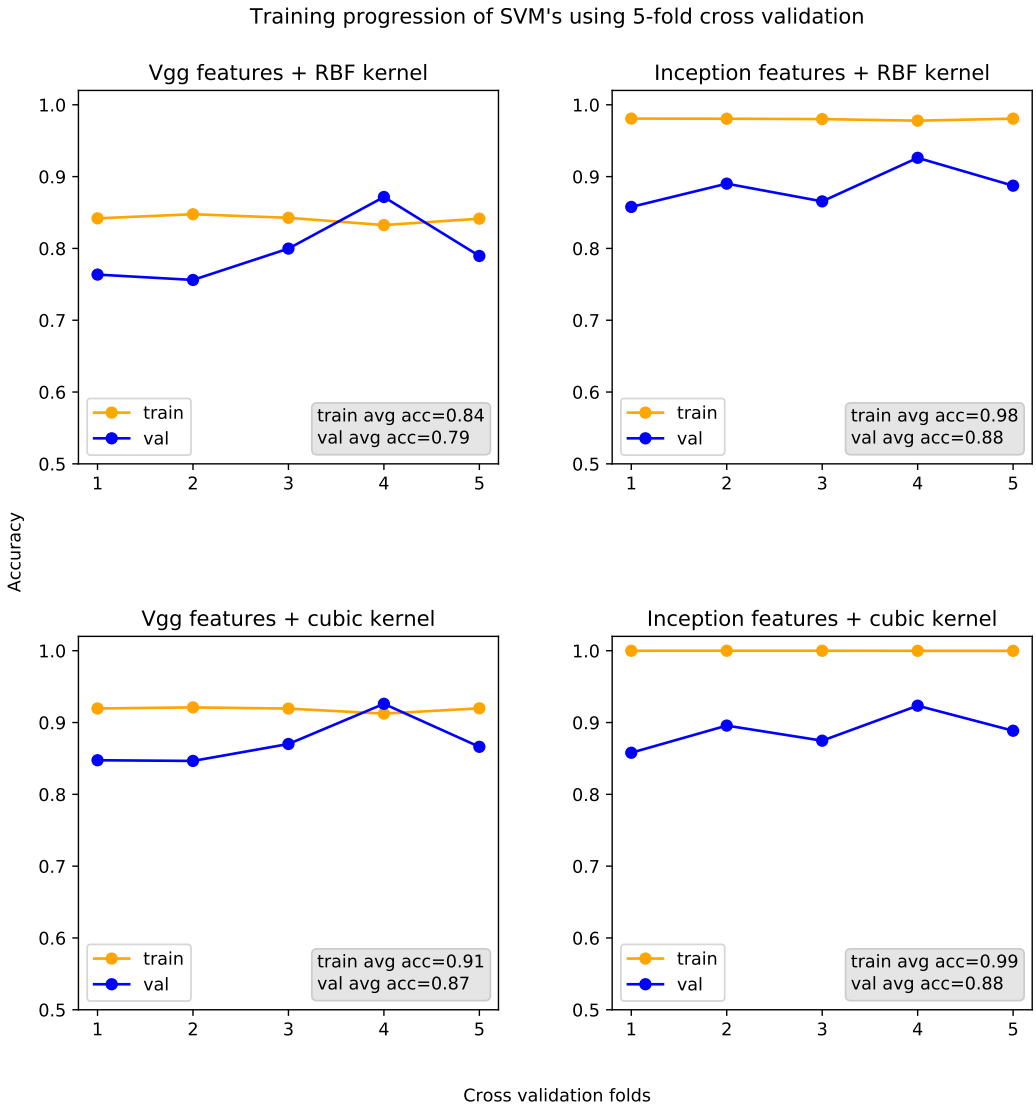


Figure 4.7: *Training and validation accuracies progression for the training of four clasifiers: 1) use of Vgg-16 features to train a SVM with RBF kernel, 2) use of Inception features to train a SVM with RBF kernel, 3) use of Vgg-16 features to train a SVM with cubic kernel and 4) use of Inception features to train a SVM with cubic kernel.*

4.4.2 Transfer learning using fine-tuning

The second approach, fine-tuning, is a type of transfer learning in which a CNN that has been trained for another task is modified, re-training some layers and adapting them to new data [54]. This approach is usually implemented with CNNs like Vgg-16 and Inceptionv3, which have large quantity of parameters because training them from scratch (weights initialized randomly) can affect their ability to generalize and may result into

low accuracies or overfitting [7]. As the deeper layers of a CNN extract more dataset-specific features [55], the fine-tuning is usually performed over these layers to modify the representations accordingly to the new classes.

In this second approach we split the former training set, used for the first approach in Section 4.4.1, into a training set with 11,077 images (64 %) and a validation set with 4,096 images (24 %). The test set with 1,919 images (12 %) remained the same.

A total of 512 feature maps were obtained from the Vgg-16 and converted to an array of $25,088$ features as illustrated in Figure 4.8. These features are the input to the fully connected layers, the first layer with $4,096$ nodes and the second with $1,024$ nodes. To obtain the final classification, a third fully connected layer with eight nodes was configured, one node for each class.

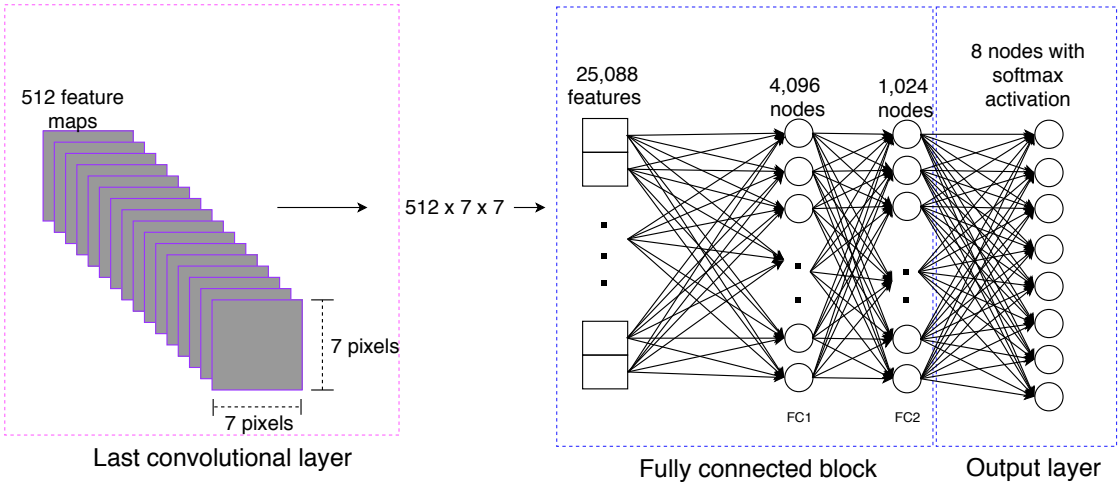


Figure 4.8: *Fully connected block and output layer designed for the Vgg-16.*

The design of the fully connected layer for the Inceptionv3 is shown in Figure 4.9. A number of $2,048$ feature maps of 8×8 pixels were obtained from Inceptionv3 convolutional blocks and were converted to an array of $2,048$ features using average pooling. These features are fed to a fully connected layer of $1,024$ nodes and the output goes to a final layer of eight nodes, one by each class.

The design of the fully connected layers of both networks was done empirically, starting

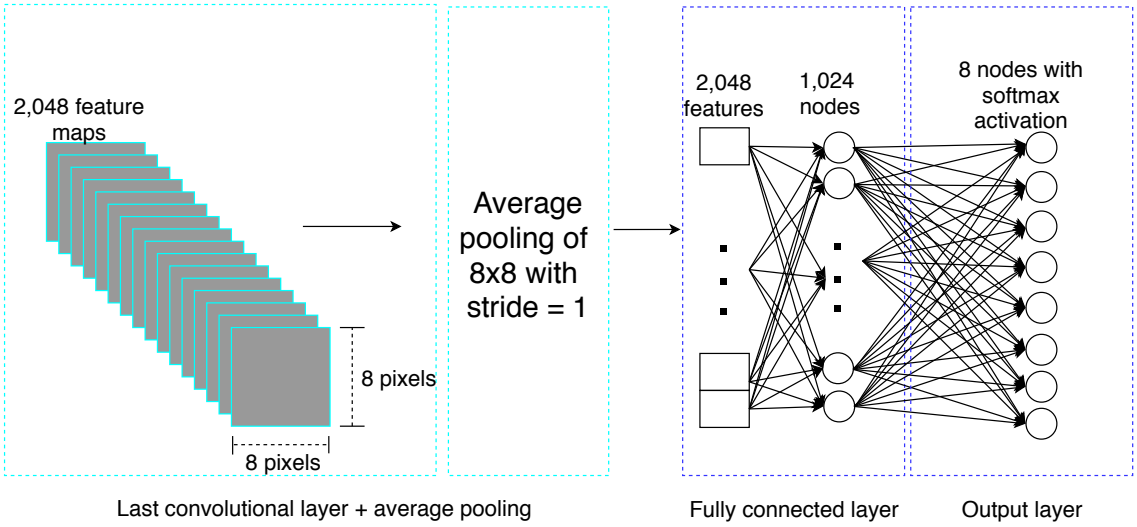


Figure 4.9: *Fully connected block and output layers designed for the Inceptionv3.*

from the original design and changing the number of layers and nodes to obtain better accuracies. The objective was to design a customized fully connected block maintaining a balance between performance and training time.

Training using the backpropagation algorithm

During the training stage, the neural network classifies each training image based on the features extracted. After each image is classified, the difference between the manual label assigned by the pathologists (ground truth) and the prediction made by the network is measured using a loss function. For this case, it was implemented a categorical cross-entropy function defined as follows:

$$L = -\sum_{j=1}^{n} y_j \log(\hat{y}_j) \tag{4.5}$$

where $\hat{y}_j$ is the probability of the predicted label and depends of the weights (ω) and biases of the network, y_j is the ground truth and n is the number of classes. During the training, the goal is to minimize the loss function calculating its gradient (∇L) at certain point and updating the weights and biases of the network iteratively. The gradient is

calculated backwards through the network, computing first the gradient of the weights of the final layer and finishing with the gradients of the weights of the first layer. The computations from one layer are reused in the computation of the gradient of the previous layer. The weights are updated according to a hyperparameter called learning rate (α) with the following relationship:

$$\omega := \omega - \alpha \nabla L \tag{4.6}$$

The learning rate is an empirical parameter that is selected based on experience. The lower the learning rate, the slower is found the minimum of the loss function. We selected a small value of learning rate ($\alpha = 10^{-4}$) to ensure the result convergence and train the model in an acceptable time. The training was performed through 100 iterations (epochs). In each epoch, all the images from the training and validation set are passed through the network and the training validation accuracy and loss are obtained. Although different CNNs architectures may require different number of training epochs, we selected a value of 100 because both models reached accuracy plateau for this value. Our target performance parameter was the accuracy, so that we saved the models that obtained the highest accuracy during the training. This is the number of images correctly classified over the total number of images.

Two important hyperparameters were selected: the number of training images and the number of convolutional blocks to train. To do this, we carried out several tests in which we varied: 1) the number of training images per group: 512, 1,024, 2,048 and 4,096 images; and 2) the number of convolutional blocks fine tuned: last block, last two, last four, last six and last eight blocks, respectively. The validation accuracies are presented in Table 4.3. In the case of Vgg-16, the last 1, 2 and 4 blocks were trained (values in italic). From the results, it can be observed that the best models, with higher accuracies, are those whose last four and last six blocks were trained with 2,040 and 4,096 images per class.

Based on the results in Table 4.3, we selected the models with the last four convolutional

images/group \ blocks trained	**1**	**2**	**4**	**6**	**8**
512	*92*	*93.7*	*93.5*		
	92.7	92	91.8	92.7	91
1,024	*93*	*95*	*95*		
	92.3	93	93.2	94	94
2,048	*93.2*	*95*	*95.5*		
	93	94	94	95	95
4,096	*94.5*	*95*	***95.4***		
	93	94.7	**95.2**	96	96

Table 4.3: *Validation accuracies of Vgg-16 (values in italic) and Inceptionv3 changing two parameters: 1) The number of training images per group and 2) the number of convolutional blocks trained. The accuracies values in bold represent the models selected for testing.*

blocks tuned using 4,096 images per class (accuracies in bold). Figure 4.10 shows the training and validation progression of the accuracy and loss for these two models. In addition, the training and validation maximum accuracy values (train max acc, val max acc) and training and validation minimum loss values (train min loss, val min loss) are presented.

For Vgg-16, the maximum validation accuracy (0.96) was obtained in epoch 80. For Inceptionv3, the maximum accuracy (0.95) was obtained in epoch 78. The weights obtained in these epochs determined the final models. It should be noted that the training and validation curves are close and present a similar performance. This means that it can be assured that the tuned models do not present overfitting.

The fine-tuning and all tests were performed using Keras and Tensorflow software libraries.

In the next section, we present the results of a test assessment of the selected models, including the confusion matrices and some performance parameters.

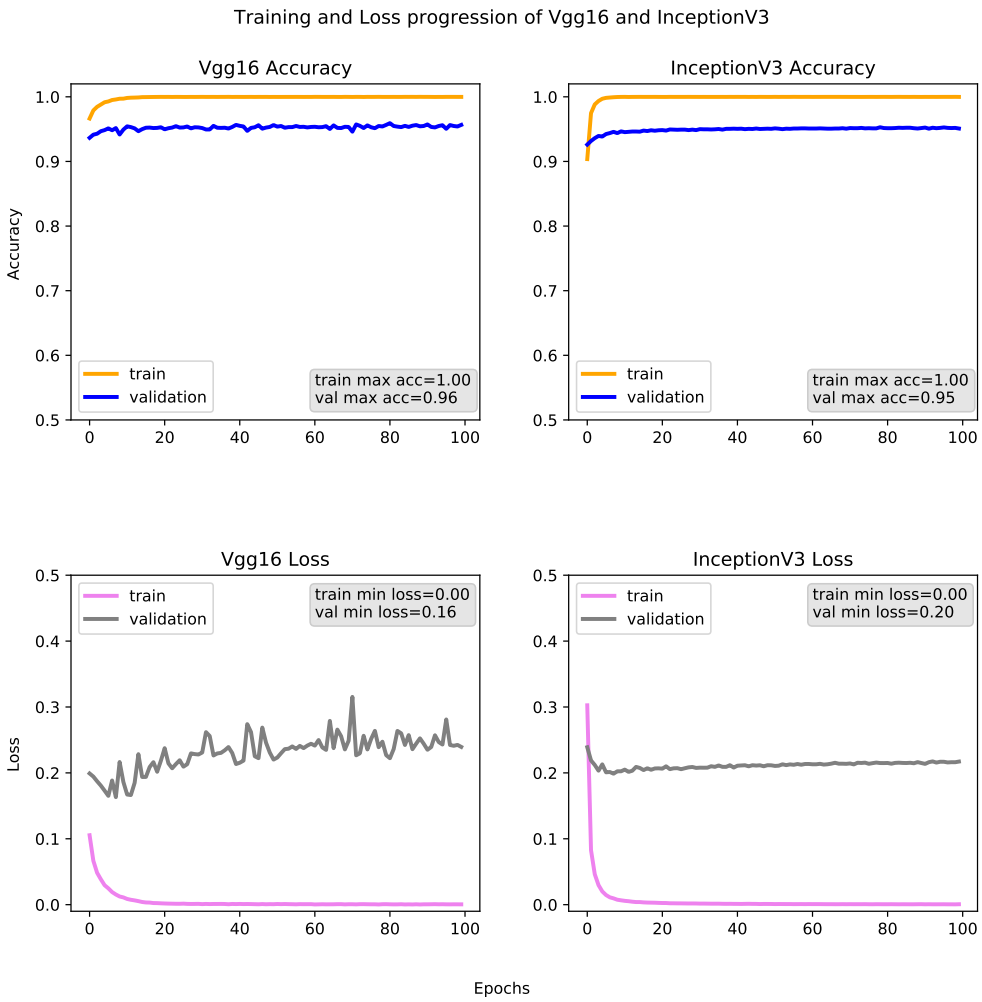


Figure 4.10: *Progression of the accuracy and loss during the fine tuning of the last four convolutional blocks of Inceptionv3 and Vgg-16. Each network was trained for 100 epochs.*

4.5 Experimental assessment

The objective of this work was the classification of eight classes of PB cells with high accuracy. For this assessment, we used the test set of images that were not used for training. For each experiment, a confusion matrix was calculated to obtain performance parameters as follows:

$$Sensitivity = \frac{TP}{TP+FN}; \qquad Specificity = \frac{TN}{TN+FP}; \qquad Precision = \frac{TP}{TP+FP}$$

where for each class:

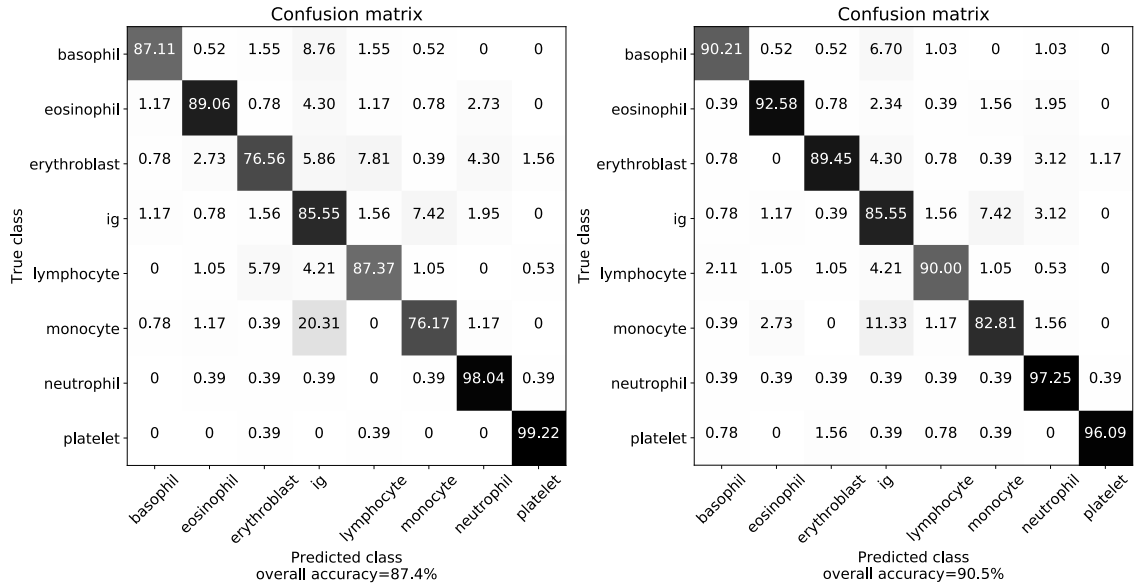
- TP = True positive. Value in the principal diagonal of the target class in the confusion matrix.
- TN = True negative. Is the sum of all rows and columns excluding the target class column and row.
- FP = False positive. Is the sum of the column values of the target class, excluding the true positive.
- FN = False negative. Is the sum of the row values of the target class, excluding the true positive.

Since this is a multi-class classification, we took a "one vs all" approach where the performance parameter is calculated for each class, considering the target class as the positive and the rest of the classes as negative We also calculated the sensitivity, specificity and precision average for each model and presented them in a bar plot along with the standard deviation for a general overview of each model performance.

4.5.1 Results of transfer learning using a CNN as feature extractor

Figure 4.11 presents the confusion matrices for the classification of the test set using the SVM classifiers with cubic kernel and trained with Vgg-16 features (model a) and Inceptionv3 features (model b).

The overall test accuracies were 87.4 % for model (a) and 90.5 % for model (b). The principal diagonal of the matrix contains the TPR for each group (% of images correctly classified). The rows of the matrices are true values and the columns are the predicted values. For example, the confusion matrix for the results using a SVM trained with Vgg-16 features shows that 99.22 % of the platelet images were correctly classified.



(a) *SVM trained with Vgg-16 features.* **(b)** *SVM trained with Inceptionv3 features.*

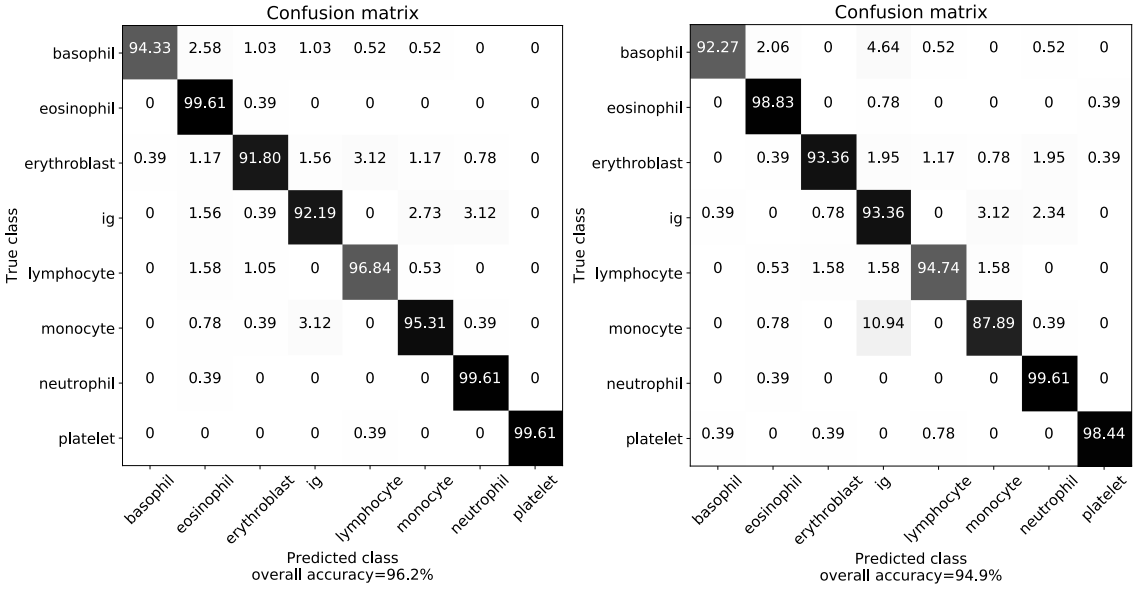
Figure 4.11: *Test assessment confusion matrices of SVM classifiers with cubic kernel. Values are in percentage.*

With respect to the TPR for each class using the model a, we obtained the highest values for platelet and neutrophil classes (>98 %). Other classes as eosinophil, lymphocyte, basophil, ig present values greater than 85 %. Monocyte and erythroblast present the lower TPRs with 76.17 % and 76.56 % respectively. The overall accuracy with model (a) is 87.4 %. Model (b) presents similar performance to model (a) but, in general, with higher true positive rates and consequently with higher overall accuracy (90.5 %).

4.5.2 Results of fine tuning

The confusion matrices in Figure 4.12 show the results of the test assessment of the best models based on Vgg-16 and Inceptionv3 networks selected at the training stage. The model based on Vgg-16 presents very high TPRs for neutrophils, eosinophils and platelets (99.61 %) classes. TPR for the rest of classes is over 90 %. The Inceptionv3 based model presents a similar performance to Vgg-16, with the highest TPRs for the same three groups: neutrophils (99.61 %), eosinophils (98.83 %) and platelets (98.44 %). An

important difference between the two models lies in the results for the monocyte class (95.31 % for Vgg-16 and 87.89 % for Inceptionv3). Regarding the overall accuracies, the fine-tuned Vgg-16 is 96.2 % and Inception 95 %.



(a) *Vgg-16 based model* **(b)** *Inceptionv3 based model*

Figure 4.12: *Confusion matrices of the test assessment of tuned models based on Vgg-16 and Inceptionv3. Rows represent the true values and columns represent the predicted values. The principal diagonal (grey) are the true positive rates. All values are in percentages.*

The bar plots in Figure 4.13 present average sensitivity, specificity and precision for the four models tested. We obtained higher sensitivity and precision values with the fine-tuning approach in comparison with the first experiment of transfer learning. Specificity values are similar for all models.

From these results, we can state that by implementing fine tuning to pre-trained models it is possible to obtain new models for classification with very high accuracies. This is particularly noticeable in this case of classification of PB cell images, where characteristics of the different classes are very subtle, such as the chromatin clumping in neutrophils and eosinophils, and the variations in the nucleus shape: spherical, oval, u-shaped, kidney bean shaped, bilobed and others.

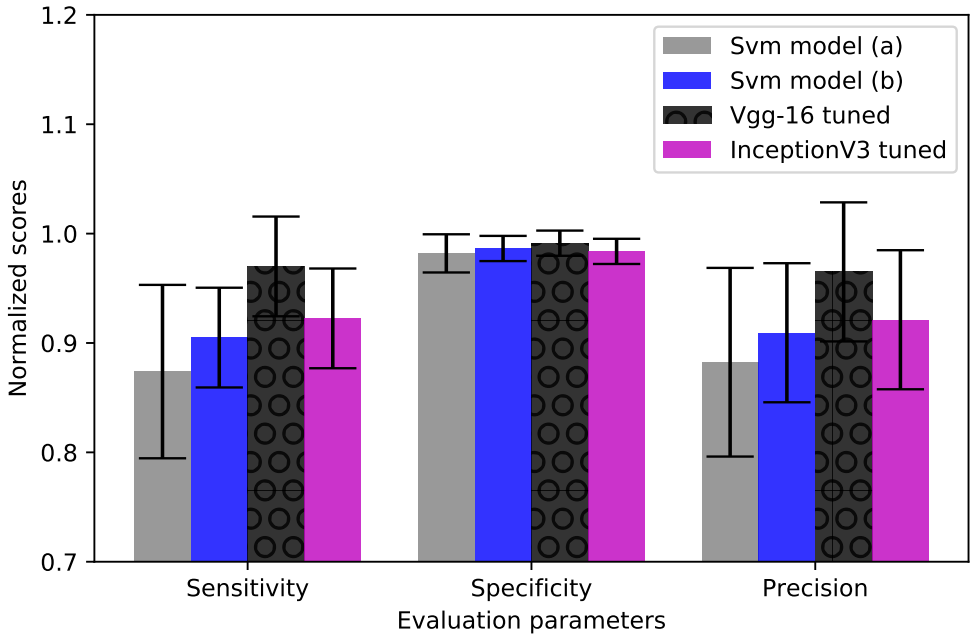


Figure 4.13: *Average sensitivity, specificity and precision for the four classification models tested.*

4.6 Discussion

The problem of automatic classification of the different cells circulating in peripheral blood has been addressed for some years now within the usual machine learning framework with the following steps: preprocessing, segmentation, feature extraction, selection and classification. Specifically, for the classification of leukocytes, different approaches have been used such as Fuzzy logic [56], Learning Vector Quantization [57], k-nearest neighbors algorithms (k-NN), Bayes classifiers [58], multi class support vector machines [59], multilayer perceptrons [59]–[61] among others.

The traditional framework requires to segment and extract features manually, this means that all possible variations in the morphology of normal and abnormal cells circulating in blood cells should be considered, which is truly difficult [62]. Specially, segmentation presents many challenges due to the complex morphology of the cells, the variability of the blood smear staining and the general conditions of the images acquisition [57]. All

of these aspects affect the results of the accuracies of the classification models [56], [57], [60], [63]. In addition, the small amount and low quality of the blood cell images available also affect the robustness of the models [58], [61].

Our group has been working in developing image-based screening methods for automatic classification of abnormal lymphoid cells using the traditional framework [21], [26], [64]. In this work, we have addressed the use of new emerging deep learning techniques for the recognition of other groups of cells that are also circulating in peripheral blood, such as neutrophils, eosinophils, basophils, lymphocytes and monocytes, immature granulocytes (metamyelocytes, myelocytes and promyelocytes) and others as erythoblasts and platelets. As training a CNN from scratch requires large amounts of data and a lot of expertise to ensure convergence, fine-tuning arises as an alternative to train a model that can outperform a CNN trained from scratch [65].

In this work, two pre-trained networks of different architectures, Vgg-16 and Inceptionv3 have been used. They were trained for the ImageNet contest with images of very different objects with respect to blood cells. The test accuracy of 96 % obtained with our fine-tuned Vgg-16 is higher than the one previously reported [46], [47].

To perform a more detailed comparison with these works, Figure 4.14 shows the true positive rate for each individual cell group. Our fine-tuned Vgg-16 used as an end-to-end classifier outperforms the results of the other models for all the five normal types of cells. In addition, our model classifies with high accuracy other types of cells circulating in PB such as platelets, erythroblast and immature granulocytes.

To accomplish this, two strategies have been implemented with the aim to improve the performance of the classifiers. The first one was the application of data augmentation to the training set to increase its size and balance the number of images per class. This strategy allows to obtain a more robust model [66]. The second strategy was the use of open source software that allows easy implementation of the classifier and the reproduc-

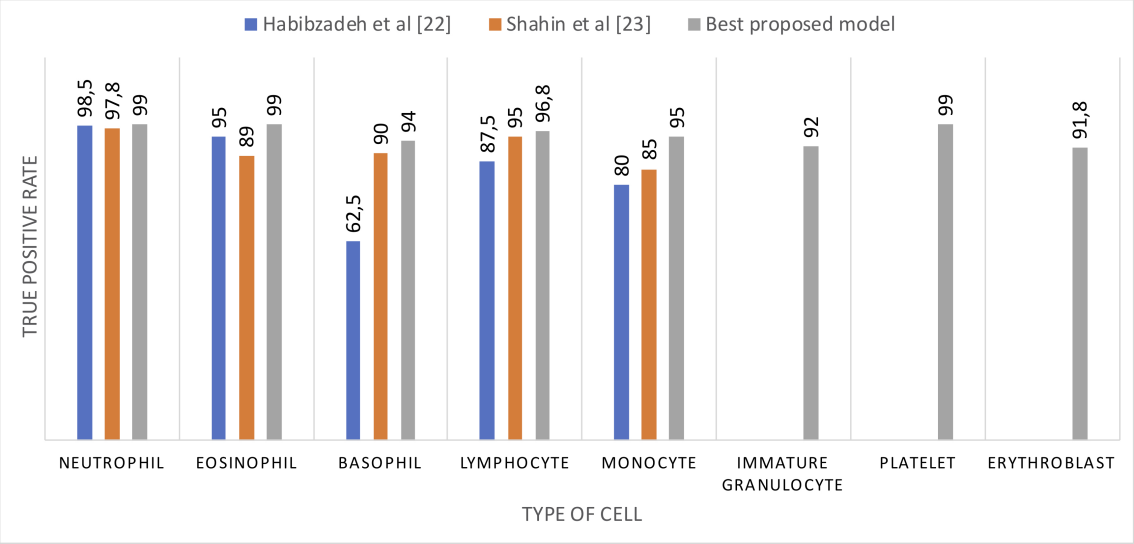


Figure 4.14: *True positive rates for each individual blood cell type, comparing the proposed Vgg-16 best model with two other previous published works.*

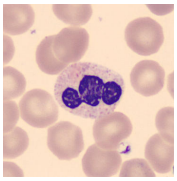
tion of the results presented here. Another important aspect to remark is the number of images and their high quality used to perform this work. The dataset acquired, with more than 17,000 images, is among the largest used so far to train a blood cell classifier. This allowed to obtain a robust model, avoiding overfitting [46].

An additional difference of this work with previous publications [46], [47], [63], and one of the main contributions, is that our tuned models succeeded in classifying eight types of blood cells. To our knowledge, this high number of blood cell groups automatically recognized in this work using CNNs has not been reported in the literature. The practical interest is twofold: first, they include the family of normal leukocytes and the cells more frequently observed in infections and regenerative anaemia; and second, the lessons learned in this work will be the basis for further extensions of CNN to classify the broader classes of leukemic and dysplastic cells.

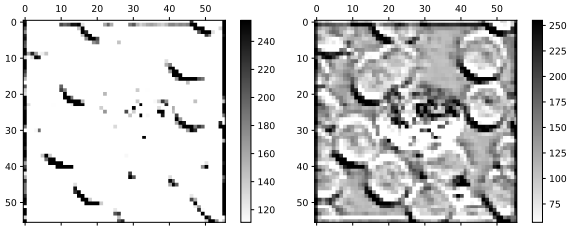
The remaining subsections discuss some insights on the convolutional features, some interpretations about the classification results and a few practical implementation issues.

4.6.1 Visualization of intermediate feature maps

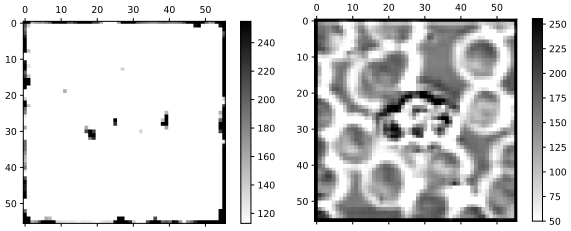
Displaying the feature maps generated by intermediate convolutional layers of a CNN may be useful to get an idea of what features the network extracts. With this purpose, we fed both our trained Vgg-16 model and the original Vgg-16 Imagenet model with a cell image selected from the test set. For comparison, we extracted the same feature maps from the same convolutional layers as illustrated in Figure 4.15. Part (a) shows the original image (neutrophil). Figure 4.15(b) contains the feature map 38 of 256 from the convolutional block 3, layer 1. From the feature map generated by the original CNN (left), it can be noted that at this deepness of the network, it barely detects diagonal lines of the input image. This was expected since, at this depth, the network still detects dataset-specific features and the original Vgg-16 model was trained with images of everyday objects such as animals, plants, people, which are very different from the images of blood cells. Instead, our own fine-tuned network is able to detect information about the cell, its nucleus, its texture and the red cells around. In our problem of blood cell classification, subtle morphological differences, such as the roundness of the nucleus, the chromatin texture and the presence or absence of granules in the cytoplasm, can make the difference between cell types. In Figure 4.15(c), the feature map 27 of 256 from the convolutional block 3, layer 2, shows that at this stage our network is capable of detecting the red cells from the background, while the feature map from the original network does not capture almost any information from the image. Figure 4.15(c) shows another case where the improvement in the detection of important characteristics of the cells is visually interpretable. In each case, the network detects different types of characteristics assigning different degrees of importance depending on the training performed. It is important to notice that, although all layers of the network extract important information about the input images, not all the feature maps can be visually interpreted. Deeper layers contain more abstract information about the specific class to which the image belongs and further iterations produce smaller maps with features more difficult to observe and interpret.



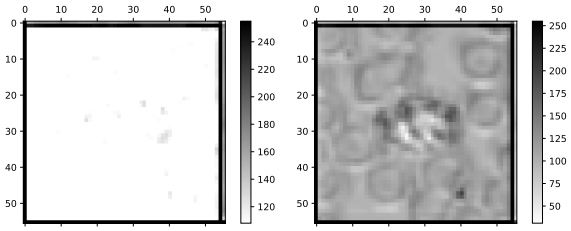
(a) *Input image selected (Neutrophil)*



(b) *Feature map 38 from convolutional block 3, layer 1.*



(c) *Feature map 27 from convolutional block 3, layer 2.*



(d) *Feature map 54 from convolutional block 3, layer 3.*

Figure 4.15: *Feature maps from Vgg-16 with Imagenet weights (left) compared with feature maps from Vgg-16 fine tuned for the classification of eight classes of peripheral blood cells.*

In the daily work, pathologists perform a visual qualitative analysis of the cell features based on their experience and knowledge. These qualitative features have direct relationship with quantitative features that can be extracted directly from the images using classical machine learning pipelines and mathematical formulations. For example, shape characteristics expressed qualitatively by the pathologist as the size of the cell or the size of the nucleus may be expressed quantitatively using nucleus/cytoplasm ratio, the perimeter of the nucleus, the perimeter of the cell and cytoplasmic external profile. Regarding

texture, qualitative features like chromatin clumping or cytoplasmic granulation among others are quantitatively described using formulations based on granulometry [21], [30], [35]. On the other hand, convolutional neural networks can extract features without image segmentation and are robust models that can achieve state-of-the-art accuracies as we achieved in this paper, but the features extracted have not direct relationship with mathematical formulations and are of limited understanding. Further developments would be worth to explore to translate them into visually interpretable patterns [67].

4.6.2 Detailed analysis of the confusion matrix

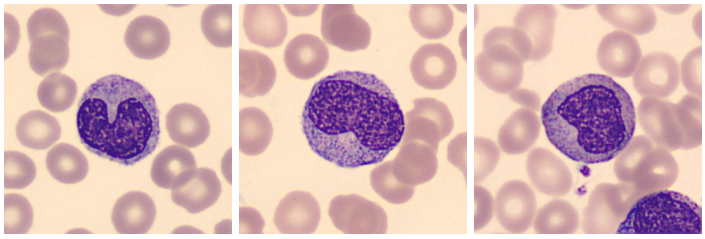
Another way to understand the behavior of the final models developed in this work is to perform an analysis of the results presented in the confusion matrix of Figure 4.12. Both models Vgg-16 and Inceptionv3 present the highest TPR for the neutrophils, eosinophils and platelets. The remaining blood cells also present high TPR but with some classification errors, which are worth to discuss in light of the morphological similarities between cell types to understand the model behavior and establish future lines of improvement.

For example, the Vgg-16 model classifies 3.12 % of monocytes as immature granulocytes as observed in Figure 4.12(a). These two classes come from the same precursor cell, the myeloblast and therefore, have common morphological characteristics such as their size: monocytes are large cells (12-20 μm) and have fine chromatin as well as the promyelocytes (15-25 μm). In addition, the nucleus of the monocyte can be kidney-shaped or be indented as well as the nucleus of the metamyelocyte. A clear difference between monocytes and immature granulocytes is the texture of the cytoplasm, since the monocyte has some granulation but not as basophilic as that of the immature group (promyelocytes, myelocytes and metamyelocytes). Examples of these cells are presented in Figure 4.16 (a).

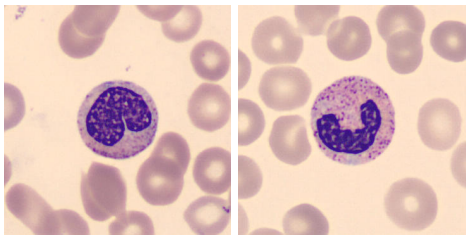
On the other hand, 3.12 % of immature granulocytes were classified as neutrophils, as

seen in Figure 4.12(a). This may be due to morphological similarities, mainly in the nucleus shape, between the metamyelocytes and the neutrophils band since the latter are an intermediate state in the process of maturation of the metamyelocyte to segmented neutrophil. See Figure 4.16 (b).

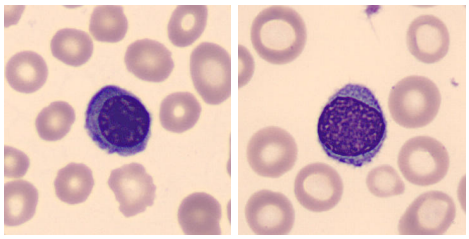
Regarding the erythroblasts, 3.12 % of their images were classified as lymphocytes, see Figure 4.12(a). Both cells exhibit common patterns. For instance, they have similar size, round nucleus and condensed chromatin as displayed in Figure 4.16 (c).



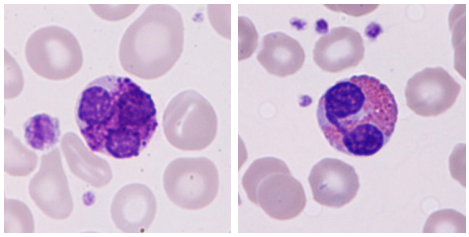
(a) *Examples of monocyte (left), promyelocyte (center) and metamyelocyte (right).*



(b) *Examples of metamyelocyte (left) and neutrophil(right).*



(c) *Examples of erythroblast (left) and lymphocyte (right).*



(d) *Examples of basophil (left) and eosinophil (right).*

Figure 4.16: *Examples of cell types where the final models present classification errors*

Finally, 2.58 % of basophils were classified as eosinophils. These two groups share some morphological characteristics such as the similar size and the lobed shape of the nucleus. In addition, the basophil granules are larger and darker and cover the entire nucleus and cytoplasm. See Figure 4.16 (d).

4.6.3 Practical implementation issues

There are no mathematical expressions or well-established rules to define the number of images to properly train a CNN, how many epochs should be required or how many layers should be tuned. Therefore, most of the model hyperparameters in this work were selected empirically. We performed different tests varying these hyperparameters to know how they affect the final classification results. From our experience, we concluded that using a minimum of 1,000 images per group, a number of epochs around 80 and tuning the last two or four blocks of our proposed CNNs it was possible to obtain satisfactorily high accuracies (96 % for Vgg-16).

It is interesting to analyze how the performance of the models change depending on the balance of the training set. In this respect, we performed fine-tuning of both CNNs, Vgg-16 and InceptionV3, with the original unbalanced dataset to compare the results with the balanced dataset presented in this paper. Figure 4.17 shows the validation accuracies during the training in both cases. They are higher when the networks are trained with the balanced dataset (>0.95), as expected. Another observable feature is that training with the unbalanced dataset produces many fluctuations in the Vgg-16 validation accuracy. This effect may be due to the fact that the unbalanced validation set is too small. By increasing the number of images and balancing the dataset, it is observed that these fluctuations decrease.

We also performed a testing of the networks when trained with the unbalanced data set. In comparison to the balanced data, the overall test accuracy reduces from 96.2 % to 93.9 %

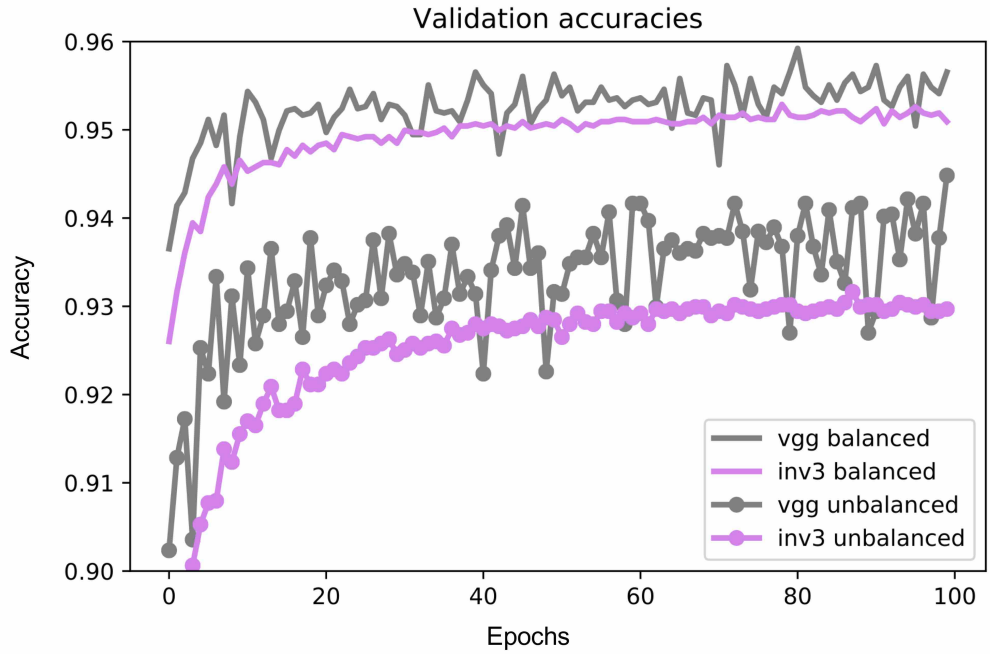


Figure 4.17: *Validation accuracies of Vgg-16 and Inceptionv3 trained with balanced and unbalanced (original) dataset.*

for Vgg-16, while reduces from 94.9 % to 92.5 % for Inceptionv3. To give more details, Figure 4.18 shows the true positive rate values for each type of cell. Two main details may be highlighted:

1. The TPR of the groups with higher number of images, eosinophil, neutrophil and platelet, remain at very high values in both training cases, balanced and unbalanced dataset.

2. The TPRs for the rest of the groups with smaller number of images (lymphocyte, monocyte, immature granulocyte, basophil and erythroblast) decrease for both networks when are trained with the unbalanced dataset.

To summarize, the dataset quality is essential. It should not only have a sufficient number of images, but these images should be of good quality and be properly labeled to be able to observe morphological characteristics that can lead towards a diagnosis. In general, medical data is scarce and difficult to label. In our interdisciplinary research group we

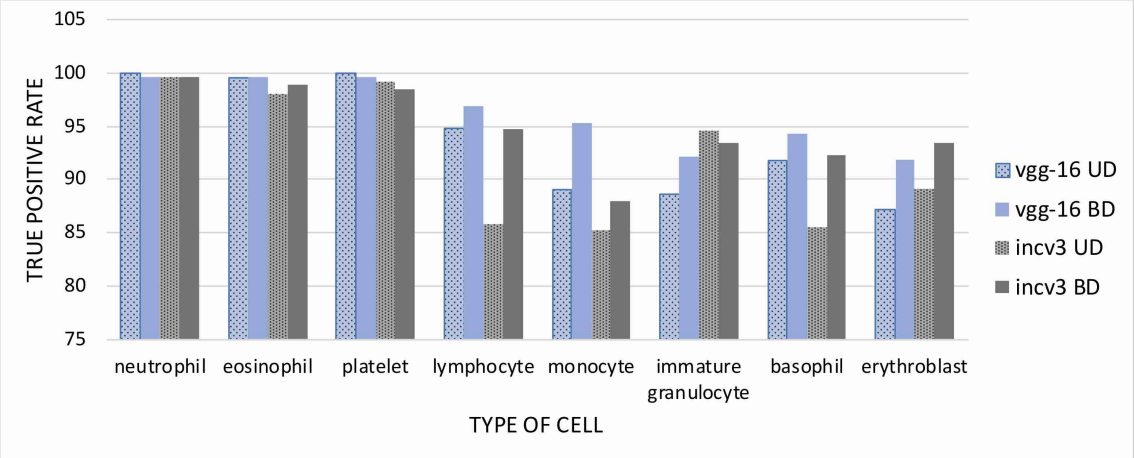


Figure 4.18: *True positive rate by type of cell obtained with Vgg-16 and Inceptionv3 (incv3) trained with the unbalanced dataset (UD) and the balanced dataset (BD).*

have experienced pathologists able to perform a manual labelling that is also confirmed through other complementary tests if needed. On the other hand, images were all stained with the same standard May Grünwald-Giemsa technique used in the clinical practice.

4.7 Conclusion

The main contribution of this paper is a classification scheme involving a convolutional neural network trained to discriminate among eight classes of cells circulating in peripheral blood. Starting from a state-of-the art general architecture, we have established a fine tuning procedure to develop an end-to-end classifier trained using a dataset with over 17,000 cell images obtained from clinical practice.

The performance obtained when testing the system has been truly satisfactory, the values of precision, sensitivity and specificity being excellent. To summarize, the overall classification accuracy has been 96.2 %.

The focus of this research has been on the group of normal leukocytes and the cells more prevalent in regenerative anaemia and infections, which cover a clinically relevant spectrum of blood cells. This is a necessary first step to develop a broader CNN-based clas-

sification system covering other types of blood cells circulating in peripheral blood in different acute leukemias and lymphomas, such as myeloblasts, lymphoblasts, abnormal B or T lymphoid cells and dysplastic cells.

Chapter 5

A new convolutional neural network predictive model for the automatic recognition of hypogranulated neutrophils in myelodysplastic syndromes[1]

[1]Based upon: Acevedo, A, Merino, A, Boldú, L, Molina, A, Alférez, S, and Rodellar, J. (2020). A new convolutional neural network predictive model for the automatic recognition of hypogranulated neutrophils in myelodysplastic syndromes. Computers in Biology and Medicine (paper in revision)

Abstract

Background: Morphological review of peripheral blood cells has a relevant role in the initial diagnosis of MDS since dysplasia is observed in one or more of the major myeloid lineages. Dysplastic neutrophils commonly show at least 2/3 reduction of the content of cytoplasmic granules by morphologic examination. Recognition of less granulated dysplastic neutrophils by human eyes is difficult and prone to inter-observer variability. To tackle this problem, we propose a new deep learning model (DysplasiaNet) able to automatically recognize the presence of hypogranulated dysplastic neutrophils in PB.

Methods: A total of eight models were generated by varying convolutional blocks, the number of nodes in layers and the fully connected layers. For training and testing we used the hold-out method, splitting the dataset in train, validation and test. Each model was trained for 20 epochs and the train set was augmented up to 3000 images by class. The five models with highest accuracies were selected for a second stage, being trained until they reached a validation accuracy of 99.5 % or a maximum number of 100 epochs. After training, for each model, quantitative cut-off values were calculated for a granularity score that discerns between normal and dysplastic neutrophils. Furthermore, a threshold value was obtained to quantify the minimum proportion of dysplastic neutrophils in the smear to consider that the patient might have MDS. With all these parameters set, classification true positive rates and total accuracies were calculated. The final selected model (DysplasiaNet) was the one with the highest accuracy (95.5 %). Additional 5 and 10-fold cross-validation was performed, which confirmed the accuracy obtained using hold-out.

Results: We performed a final proof of concept with 66 smears from 32 MDS patients and 70 smears from 40 healthy controls, none of them involved in previous steps. All smears were classified by DysplasiaNet according to whether they belong to a patient with dysplasia or not indicated by the established threshold. As result, we report a sensitivity of 95.5 %, specificity of 94.3 %, precision of 94 %, and a global accuracy of 94.85 %.

Conclusions: The primary contribution of this work is a predictive model for the automatic recognition in an objective way of hypogranulated neutrophils in peripheral blood smears. We envision the utility of the model implemented as an evaluation tool for MDS diagnosis integrated in the clinical laboratory workflow.

5.1 Introduction

MDS comprise several morphological, immunophenotypic, functional and genomic alterations that take place in the different haematopoietic lineages, from stem cells to more mature elements. MDS constitute a heterogeneous group of clonal haematological diseases having in common an ineffective haematopoiesis beside recurrent genetic abnormalities, characterized by cytopenia and variable risk of transformation to acute myeloid leukaemia [68].

Initial diagnosis of MDS is mainly supported by clinical information, blood count parameters, detection of morphological abnormalities in peripheral blood (PB) and Bone Marrow (BM) cells, flow cytometry and genetic studies. Blood smear microscopic examination is a low-expensive and easily accessible tool for assessment of cell abnormalities. It has a relevant role to support the initial MDS diagnosis, since dysplasia is observed in one or more of major myeloid lineages [69]. In MDS, dyserythropoiesis and megakaryocyte morphological alterations are more apparent in BM. In contrast, dysgranulopoiesis, which includes nuclear hyposegmentation (pseudo Pelger-Huët) or hypersegmentation, giant neutrophils and cytoplasmic hypogranularity, are morphological changes usually detected in PB.

Visual morphology shows that dysplastic hypogranulated neutrophils commonly exhibit at least 2/3 reduction of the content of cytoplasmic granules [70]. However, differential diagnosis should be done since dysgranulopoiesis is not itself a definitive evidence of a

clonal process. Differential MDS diagnosis includes other conditions that may also cause myelodysplastic changes in neutrophils, such as megaloblastic anaemia due to deficiency of vitamin B12 or folic acid, HIV infection, granulocyte colony-stimulating factor administration, the effect of drugs including chemotherapeutic and immunosuppressive agents, or copper deficiency[71].

Modern automated analysers based on digital image analysis enable faster blood smear reviews. They present satisfactory results for normal white blood cells pre-classification but do not identify the presence of abnormal blood cells [72]. The presence of less granulated dysplastic neutrophils in PB is the most difficult abnormality to detect by human eyes. This difficulty causes a considerable inter-observer variance in their manual classification. Therefore, the identification of cytoplasmic hypo-granularity becomes inconsistent among pathologists [73]. There is a clear need for further standardization of morphological cell classification and the development of computational classification models for the automatic detection of cytoplasmic hypogranulation for MDS diagnosis.

Research objective

The objective of this work was the development of a new CNN model to automatically recognize the presence of hypogranulated dysplastic neutrophils in peripheral blood. The model was designed having in mind its potential use as a secondary diagnosis support tool when MDS is suspected by the clinical pathologist based on the morphological PB smear review as well as haematological parameters such as anaemia or high mean corpuscular volume (MCV). With this purpose, the CNN was trained to identify individual dysplastic neutrophil images and used to supply a diagnosis prediction when using an individual full patient smear. A proof of concept using 136 PB smears from 72 individuals (including healthy controls and MDS patients) showed sensitivity, specificity and precision values of 95.5 %, 94.3 % and 94.0 %, respectively. To the authors knowledge, this is the first CNN model that was built and validated for MDS diagnostic support, with potential to be integrated in the workflow of the clinical laboratory. Since the recognition of hypo

or hypersegmented nuclei, giant neutrophils, pseudo Pelger-Huët anomaly or dysplastic abnormalities in other cell types are quite evident and easily detected by visual inspection [72], we decided to focus our research to the design of a predictive model for the automatic recognition of cytoplasmic hypogranularity in dysplastic neutrophils.

After a summary of the most related work, the organization of the chapter is as follows. Section 5.2 describes the different image sets used in this work. Section 5.3 describes the step by step methodological development of the CNN model, including architecture, training and evaluations to obtain a final model ready for a proof of concept. Section 5.4 describes the results of such proof of concept and interpretability techniques. Results and practical aspects are discussed in Section 5.5. Finally, Section 5.6 highlights the main conclusions of this work.

Related work

Machine Learning (ML) methodologies have been applied in hematopathology as decision support systems in different tasks, such as differential diagnosis of malignant haematological diseases, therapy selection or risk predictions [73]. Recently, several works on cells classification have been focused on using segmentation and hand-crafted feature extraction with the objective to classify among normal and reactive lymphocytes, blasts and abnormal lymphoid cells [28], [29].

Since the advent of deep learning, more studies presenting new methodologies and models applied to malignant diseases detection have been published. Deep learning is a class of ML that uses multiple layers to extract high-level features from the input, which increasingly is present in diagnosis support tools based on image analysis. Among deep learning techniques, CNNs are frequently used for medical image analysis in haematology [74] because they allow to define systems not relying on hand-crafted features and obtain better classification results [75], [76]. Nowadays, several CNN-based models can discriminate among different types of leukocytes in PB [20], [77], [78]. Recent efforts

are noticeable for accurate classification of acute leukaemia and other haematological malignancies using pre-trained CNN models [79]–[81].

However, little has been published on the automatic detection of dysplastic cells present in MDS. The assessment of dysplasia in Bone Marrow (BM) has been approached using a pre-trained CNN like Resnet-152 for the automatic classification of hypo-granulated cells [82]. The dataset used was small, conformed by 1,797 labelled images, what may be due to the difficulty of obtaining BM cell images. The model performance was evaluated using a five-fold cross-validation algorithm, obtaining sensitivity and specificity of 91 % and 97 %, respectively, as well as a total accuracy of 97.2 %. Regarding MDS in PB, an ensemble of CNN and Extreme Gradient Boosting (XGBoost) was trained to differentiate MDS from Aplastic Anaemia (AA). The system was developed in two parts, first the CNN was trained to extract features from the cell images. Then the XGBoost was configured with these features and trained using smears from 75 MDS and 36 AA patients. The system was tested with 26 MDS and 11 AA samples. The sensitivity and specificity reported was 96.2 % and 100 %, respectively [83].

5.2 Patients and dataset selection

Blood samples, collected in EDTA, were automatically prepared using the slide maker-stainer SP10i (Sysmex, Kobe, Japan) and stained with May Grünwald-Giemsa. Digital images of individual cells were acquired with CellaVision®DM96 (CellaVision, Lund, Sweden) (363x360 pixels) from smears obtained during daily work in the Core Laboratory of Hospital Clinic of Barcelona. Cell images were selected by pathologists according to their morphological characteristics. Patients diagnosis (ground truth) were confirmed by integrating all supplementary test results: clinical data, PB and BM morphology, flow cytometry, cytogenetics and molecular biology following WHO 2016 classification of tumours of haematopoietic tissues [84].

A total of 20,670 images were collected: hypogranulated neutrophils from patients with MDS and normal neutrophils from healthy controls (HC). Images were imported to our MySQL Database described in Chapter 2 and organized into four subsets summarized in Table 5.1.

1. Training set. Split into two subsets: *train* and *validation*. The train subset contains 1,887 images of dysplastic neutrophils and 2,923 normal neutrophils. The validation subset contains 500 dysplastic and 500 normal neutrophils.

2. Test set by cell. It contains 1,000 images, 500 dysplastic and 500 normal neutrophils. It was selected to assess the performance of the model and calculate a cut-off value, as explained later.

3. Test set by smear. Arranged with 32 MDS patients (49 blood smears), it involves 2,718 dysplastic neutrophils images. It also includes 64 smears from 40 HC (3,834 images). This set was selected to assess the system for classification of individual smears and to determine a threshold value for MDS prediction.

4. Proof of concept set. It includes 66 smears from 32 new MDS patients and 70 smears from 40 new HC. It contains 7,308 images. This set was used to evaluate the final system performance in a diagnosis setting.

		IMAGE SUBSETS				
		Training				
		Train subset	**Validation subset**	**Test by cell**	**Test by smear**	**Proof of concept**
CELL IMAGES	**Dysplastic**	1,887	500	500	2,718 (49 smears)	3,071 (66 smears)
	Normal	2,923	500	500	3,834 (64 smears)	4,237 (70 smears)
TOTAL IMAGES		5,810		1,000	6,552 (113 smears from 72 patients)	7,308 (136 smears from 72 patients)

Table 5.1: *Dataset for training and testing DysplasiaNet CNN.*

5.3 Methodology

5.3.1 Overview and problem statement

The ultimate goal of this work is to build a system where the input is a set of images of neutrophils from a patient blood smear on suspicion of dysplasia. The output is a diagnosis prediction whether the patient has MDS or not. Figure 5.1 shows an overview of the workflow followed to develop and evaluate the system. Several processes are enumerated and represented in rectangles, while the input datasets and the processes outputs are represented in parallelograms.

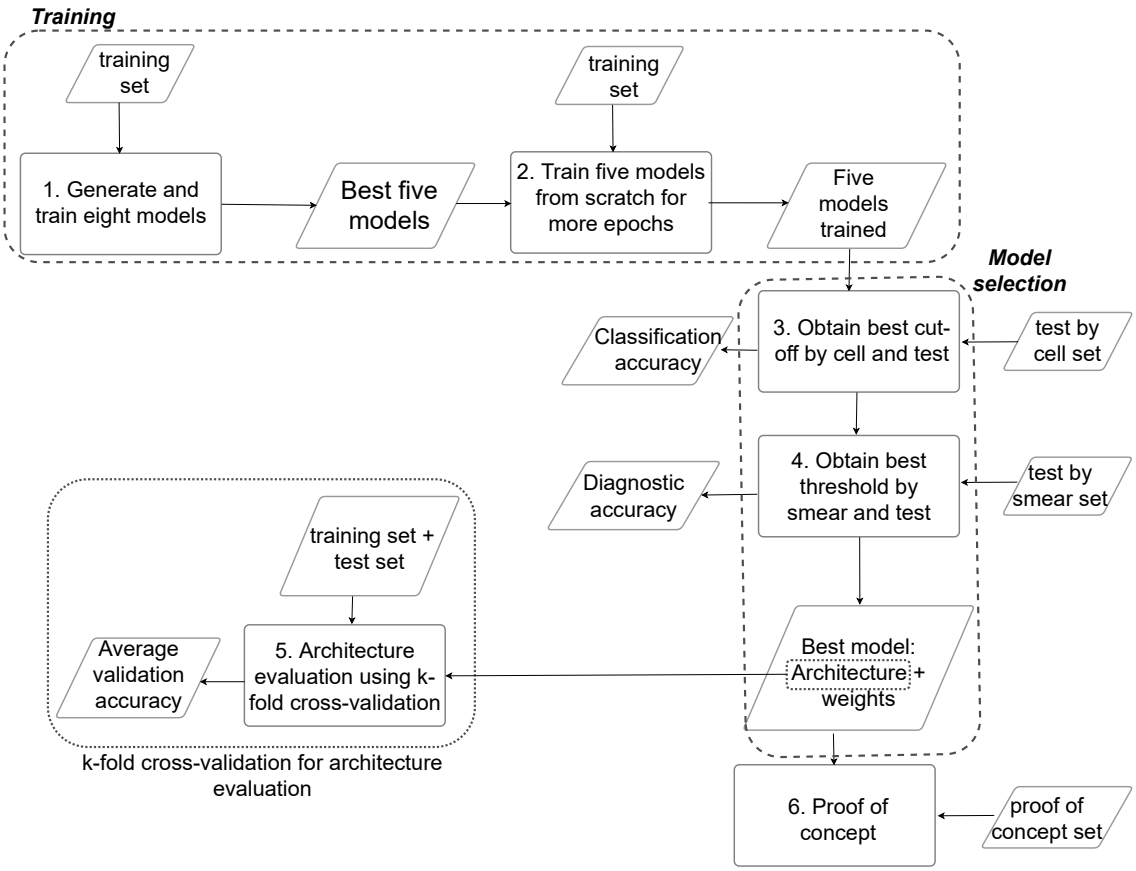


Figure 5.1: *Methodology used to develop the system (DysplasiaNet). Processes are enumerated and represented in rectangles. Input data sets and processes outputs are represented in parallelograms.*

The development started from eight CNN models generated by varying their number of

convolutional blocks, filters and fully connected layers. These models were trained for 20 epochs to observe their performance trend, selecting those that reached higher accuracies. In a second step, we trained these models from scratch for more epochs until they were ready for testing. These two processes conform the training phase represented on top of Figure 5.1.

In a visual inspection of blood cell morphology, pathologists consider neutrophils to be dysplastic if they exhibit "sufficiently low" granularity. Therefore, our next specific objective was to select the model with the best performance in terms of this specification, following the steps shown from top to bottom in Figure 5.1 (right part). In step 3, all five models were tested using the test by cell set to determine the best quantitative cut-off for each model to separate dysplastic from non-dysplastic neutrophils. The classification accuracies were used as performance measures.

As a fourth step, to adjust the model to give an objective prediction for MDS diagnosis, all five selected models were tested using full smears from healthy controls and patients in which MDS diagnosis was confirmed by integration of clinical information and all supplementary tests. The best threshold for smear was obtained and diagnostic accuracies were calculated. The result of this step was the selection of the model with the highest diagnostic accuracy, which was named DysplasiaNet. In addition, we decided to perform an alternative evaluation (step 5 in Figure 5.1) of the selected architecture by implementing five and ten-fold cross-validation techniques. The objective here was to check if such robust performance evaluation, using the whole cell dataset (training and test sets), could improve the model selected by comparing the average accuracy with the one obtained in the hold-out procedure followed so far.

The final step 6 was the proof of concept of DysplasiaNet to obtain the performance of the model using a set of images not used in any previous step. Details of each step of the methodology are explained in next subsections.

In developing a CNN classifier for eight normal blood cell categories in [20], we used fine-tuning, where already established pre-trained architectures were re-trained with our own image datasets. In this work, all models were built, trained and tested from scratch using our own datasets. We took this approach based on what was reported in [85], where authors performed a fine-grained study on transfer learning for medical images. They found that transfer learning does not significantly help to improve performance in the classification of medical images. Smaller and simpler convolutional architectures perform comparably to standard Imagenet models even when there is little data. This is the case of our medical setting, where large models might restrain a real integration in the laboratory workflow.

5.3.2 CNNs general architecture

We will use DysplasiaNet to explain the general architecture of a CNN (See Figure 5.2). DysplasiaNet consists of two sequential parts: 1) four blocks, whose end output is a set of quantitative features learned from input images (Figure 5.2, sub-figure A); 2) two fully connected layers, which supply the image classification using the learned features (Figure 5.2, sub-figure B).

Convolutional layers are the most important components in the CNNs. They are composed by N planes, and each plane is defined by a filter (kernel) with small size (F). For all CNNs tested, the size of convolutional filters is $F = 3$, so that each is a 3×3 matrix with numerical values known as weights, which conform the trainable parameters of the network.

Convolutional filters are the main components in a CNN because they learn to efficiently extract features from data. When processing the input image, these filters detect simple shapes and spots like edges or circles. When used in subsequent blocks, different convolutional filters are used to identify increasingly complex objects. Globally, the whole

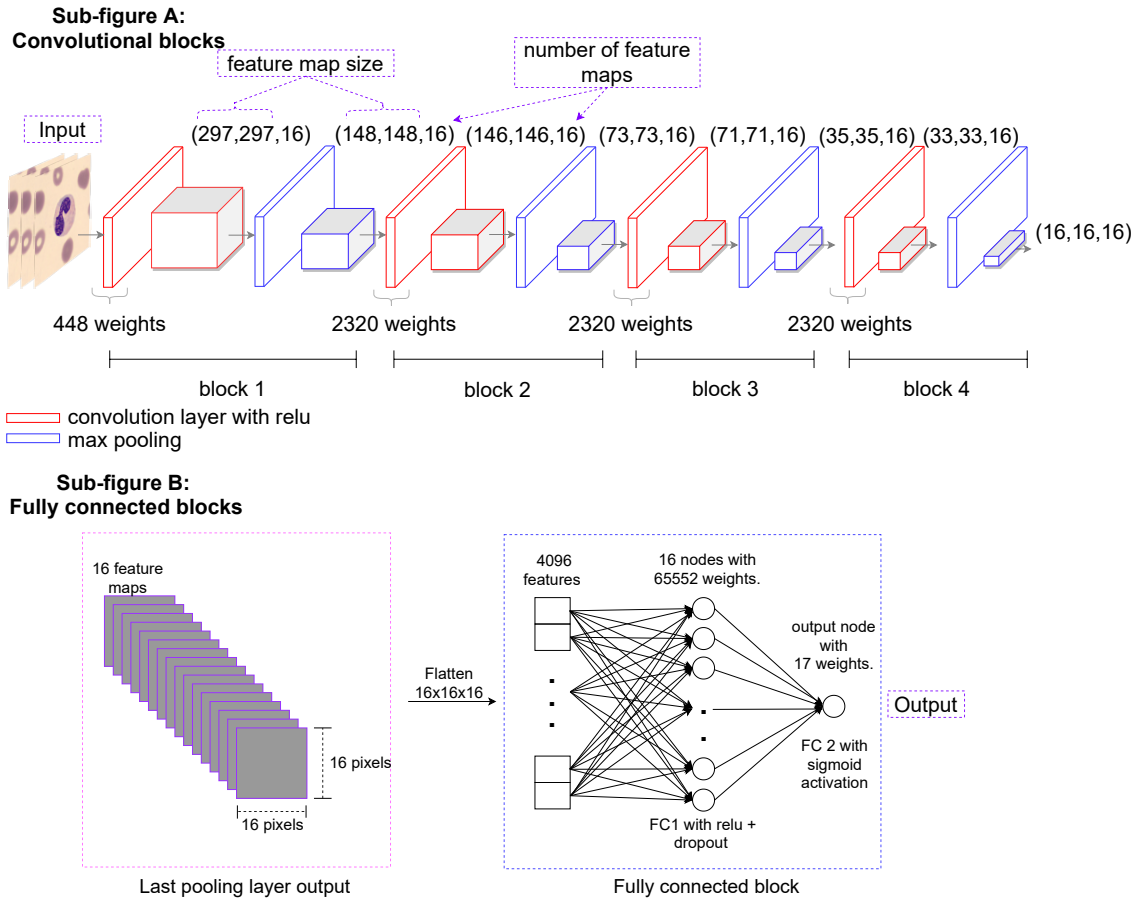


Figure 5.2: *DysplasiaNet CNN designed for the detection of myelodysplasia. It is conformed in two sequential parts: 1) four convolutional blocks (sub-figure A) and 2) two fully connected layers (sub-figure B). Images of neutrophils are the input. The output is a single score to allow the binary classification of the cell image as dysplastic or normal.*

network uses a hierarchy of convolutional filters that allows the final classification of the image.

The input image is composed by a number of planes with fixed width (im_w) and height (im_h). Specifically, this work deals with RGB images, so that we have three planes. Each plane is a gray scale image represented by a matrix of $im_w \times im_h$ pixels. Each convolutional filter slides over the images one pixel at a time in such a way that, through the convolution operation, it generates a new matrix. The obtained matrix is passed through a *ReLU* (rectified linear unit) activation function. It simply converts all negative values to zero, while keeping positive values unchanged. It is extensively used in the current CNN architectures and helps to accelerate its learning process [86]. The result is named feature

map. This way, the output of the convolutional layer is a number of N *feature maps* (FM). All FM together can be seen as a three-dimensional volume whose size can be expressed in the form:

$$FM\ output\ size_{conv} = (im_w - 2, im_h - 2, N) \tag{5.1}$$

The convolutional layers keep the sizes of FM almost the same, they only remove two positions in each dimension. This is a direct result of the filter size $F = 3$ and the sliding over the whole image. We configured the size of input images as (299,299,3). Therefore, the size of the resulting FM volume from the first convolutional layer is (297, 297, 16).

An important parameter is the amount of weights that need to be trained. This is relevant to control the CNN size and to know the necessary computational resources. The number of weights depends on the number (N) and size (F) of the filters and the number of input planes (D). This parameter is simply $(F \times F \times D + 1) \times N$.

It is important to remark that the reduced size of the filters is a key feature of CNN architectures, since it makes possible to train and store a number of parameters significantly small in comparison to the classic fully connected networks. This reduces memory requirements and computing time, and improves statistical efficiency [1].

After each convolutional layer, we placed a *max-pooling layer*, which takes the maximum value from a 2×2 frame of pixels from each feature map. It is used to introduce local invariances and generate features that are most robust to noise in the input data [87]. As a result, this layer reduces the FM size to half of the original size but maintaining the same number of FM. This structure is repeated for the subsequent blocks.

Figure 5.2 shows the output size and the number of trainable parameters of each convolutional layer as well as the output size of the pooling layers of DysplasiaNet. Note that the output of a pooling layer is the input for the next block.

The output of the last pooling layer is flattened into a one-dimensional vector, which

feeds the last part of the network conformed by fully connected layers FC . These layers are typical in the classical feed-forward neural network architectures, which are defined by their number of nodes. These node units compute a weighted sum of their inputs from the previous layer and pass the result through an activation function like the *ReLU* used before. The objective is to process the features so that classes become linearly separable by the output layer [37].

After the first FC, we place a dropout layer, which works as follows: during the CNN training, a fraction of the FC1 nodes is randomly dropped at every iteration with probability $P_{drop} = 0.5$. The objective is to force the network to learn more general patterns from the data and to prevent from overfitting. Dropout is applied only for the training phase [88]. The output layer (FC2) is conformed by a single node because we want the output of the network to be a score between 0 and 1, which indicates if the neutrophil image processed by the system shows dysplastic features or not. To accomplish this, we use a sigmoid activation function instead of the *ReLU* used in previous layers.

5.3.3 Training

With the general parameters described, as a first step (see Figure 5.1), we generated a total of eight models considering different choices of three hyper-parameters: 1) number of convolutional blocks; 2) number of nodes in layers; and 3) number of fully connected layers. Table 5.2 summarizes the models generated.

For training and testing we used the hold-out method, splitting the dataset in train, validation and test as presented in Table 5.1. Hold-out was initially preferred because the CNNs were trained from scratch and this method does not require as many computational resources as cross-validation. Each model was trained for 20 epochs where all images in the train and validation sets were used. The train set was augmented up to 3000 images by class using image augmentation techniques. This means generate new images by applying

Model name	Conv. Blocks	Nodes in layers	Fully connected layers	Weights
M1	4	16	1	72,977
M2	6	16	1	13,105
M3	4	32	1	290,849
M4	6	32	1	51,297
M5	4	16	2	73,249
M6	6	16	2	13,377
M7	4	32	2	291,905
M8	6	32	2	52,353

Table 5.2: *CNN models generated by varying the number of convolutional blocks, nodes in each layer, fully connected layers and number of weights to be trained.*

transformations to the original ones following the same procedure as in [20].

In each training iteration (epoch), all images from the train set pass forward through the network. The classification of each image is compared to the ground truth assigned by the pathologists and a loss function is calculated to quantify the classification error. In a second step, the error is propagated backwards through the network using the gradient descent algorithm to update all the weights involved in the network [5]. The training objective is twofold: find the weights that minimize the loss function and to obtain a high rate of images correctly classified (accuracy). Using the updated weights, in the end of each epoch, the validation set is passed through the network to provide an assessment of the model fit using the validation accuracy and the loss function as performance metrics. For learning weights, we used the binary cross entropy as the loss function, the ADAM optimizer [89] and the validation accuracy as the performance metric. In Figure 5.3 we present the validation accuracy obtained for each model during the training. The training and validation time of each model was around 8 minutes. Tensorboard was used for visualizing the training progression.

As it is shown in Figure 5.3 five models (M1, M2, M3, M4 and M7) reached high validation accuracy (> 95 %), while three models (M5, M6 and M8) did not improve 50 % accuracy (random accuracy). These three models were discarded and the best five were

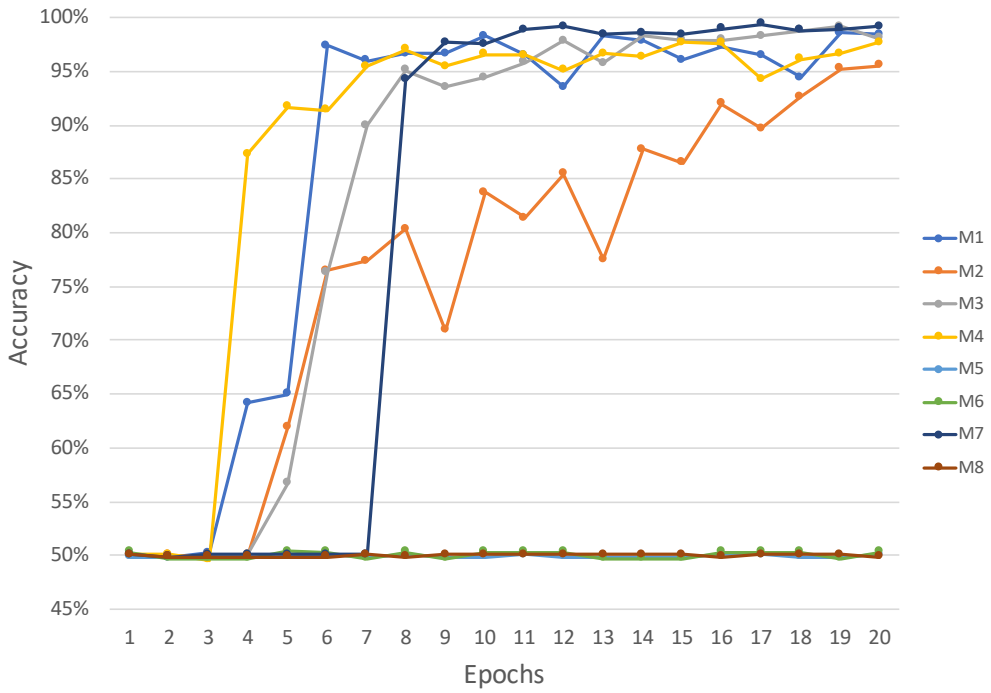


Figure 5.3: *Validation accuracy for the training of the eight models generated. All models were trained for 20 epochs.*

selected to perform further training.

In a second step (see Figure 5.1), all five models were trained from scratch until they reached a validation accuracy of 99.5 % or a maximum number of 100 epochs. Figure 5.4 shows the training progression of the five models. It can be observed that M1, M2, M4 y M7 were trained for 100 epochs, while M3 training stopped at epoch 12 when reached a validation accuracy of 99.5 %.

5.3.4 Model selection

The model selection starts with the third step presented in Figure 5.1. The objective is to obtain a CNN such that the output score quantifies granularity in the neutrophil cytoplasm. As stated before, lower granularity is associated to dysplasia. Therefore, higher values of the score (close to one) should indicate normal granularity, while values closer to zero should indicate lower granularity. Since we aim to use the network as an objective

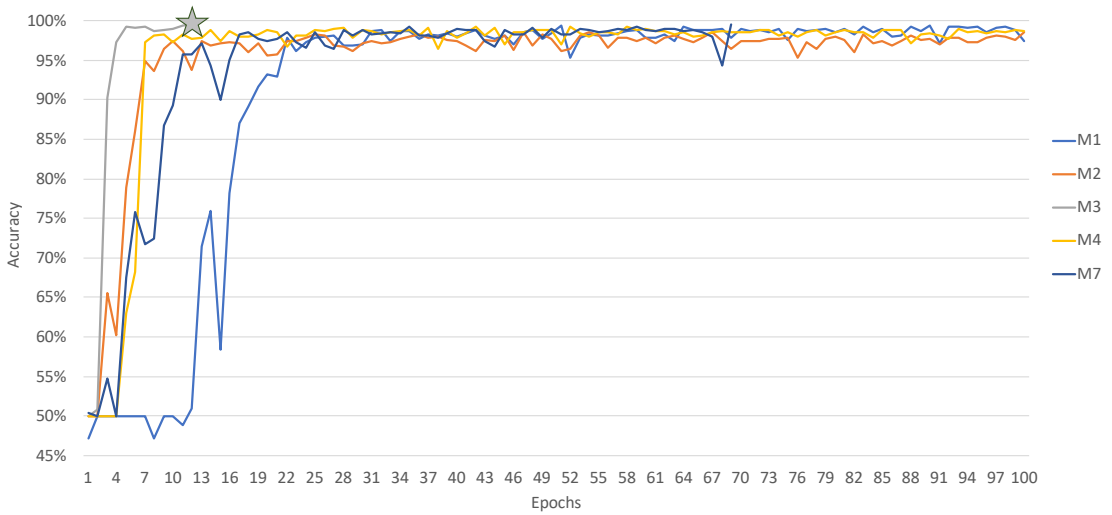


Figure 5.4: *Training progression of the best five models in a second step. All models were configured to be trained from scratch until they reached a validation accuracy of 99.5 % or a maximum of 100 epochs*

predictor, we need to determine a quantitative cut-off for the required granularity score to discern between normal and dysplastic neutrophils. The optimal cut-off represents the best trade-off between sensitivity and specificity. This means that if the score assigned by the network is above the cut-off value, the neutrophil is predicted as normal; and dysplastic otherwise.

To determine the best classification cut-off, we used the test set by cell (Table 5.1). We used the scores predicted by the network to plot true positive rate (sensitivity) versus false positive rate (1 - specificity) for different cut-off values. This resulted in the Receiver Operating Characteristic (ROC) curve, commonly used to measure the capability of a model to predict the presenceabsence of a disease, independently of its prevalence [90]. From the optimal cut-off for each model, we calculated the TPRs and the total accuracy. Table 5.3 presents the results obtained for each model.

To select the final best model, it is necessary a fourth step to test the candidate models in a more real set up to give the clinical pathologist an objective support tool for MDS diagnosis. For this purpose, it is required to establish a threshold value to quantify the minimum

Model name	Optimal cut-off	ROC - auc	TPR (%)		Total accuracy (%)
			dysplastic	normal	
M1	0.979	0.998	97.6	97.8	97.7
M2	0.875	0.996	97.2	97.6	97.4
M3	0.606	0.998	98.2	98	98.1
M4	0.443	0.996	98.6	97.2	98.2
M7	0.631	0.996	97.2	97.8	97.5

Table 5.3: *Optimal cut-off and area under the ROC curve for each model used to indicate the degree of granularity of a neutrophil. With each cut-off, TPR percentages and total accuracies were calculated.*

proportion of dysplastic neutrophils in the smear to consider that the patient might have MDS. We used the test set by smear (see Table 5.1) with smears from 40 healthy controls and 32 patients, in which the MDS diagnosis was confirmed by integration of the clinical information and all supplementary test results. All neutrophil images from the smears were classified. A ROC curve was calculated to obtain the optimal threshold and use it to obtain TPRs and diagnostic accuracy for each model. This accuracy measures the percentage of smears correctly classified. Consequently, MDS is the predicted diagnosis when the number of hypogranulated neutrophils detected in a single smear is above the optimal threshold. The results are presented in Table 5.4. M1 was the model selected because it reached the best diagnostic accuracy (95.5 %).

Model name	Optimal threshold	ROC - auc	TPR (%)		Diagnostic accuracy (%)
			dysplastic	normal	
M1	0.123	0.953	89.7	100	95.5
M2	0.089	0.959	89.7	98.4	94.7
M3	0.215	0.965	87.8	98.4	93.8
M4	0.138	0.980	89.8	98.4	94.7
M7	0.483	0.915	79.6	98.4	90.2

Table 5.4: *Optimal threshold and area under the ROC curve (auc) for each model used to indicate if a patient presents MDS or not. With each threshold, true positive rate (TPR) percentages and total accuracies were calculated. The model with highest diagnostic accuracy (M1) was selected as the final model.*

To further evaluate the selected architecture, we implemented 5 and 10-fold cross-validation using the whole dataset: training and test by cell sets. Results are presented in Table 5.5, showing the accuracy for each fold and the average accuracy and standard deviation for

	5-fold cross-validation	10-fold cross-validation
Fold	**Accuracy/fold (%)**	**Accuracy/fold (%)**
1	96.8124	99.6249
2	98.3124	97.1249
3	96.875	96.875
4	94.8125	98.3749
5	93.3125	93.8750
6	-	98.5000
7	-	98.6249
8	-	98.7500
9	-	97.6249
10	-	97.6249
Average accuracy	**96.0249**	**97.6999**
Standard dev	**1.7551**	**1.4970**

Table 5.5: *Accuracies obtained from 5-fold and 10-fold cross-validation performed to the best selected architecture.*

all folds. The average accuracies using 5 and 10-fold cross-validation were high (96.02 % and 97.69 %, respectively), and very close to the accuracies obtained in the test by cell using hold out (97.7 % in Table 5.3). These results confirm that the architecture and model selections performed before using hold out method were correct.

5.4 Results

Once the final DysplasiaNet model was selected, the next step was to establish a real scenario for detecting MDS and perform a proof of concept. Besides presenting the outcome of this assessment, in this section we show results trying to get some interpretable insight on how the CNN model achieves the classification of the neutrophil images.

5.4.1 Proof of concept

Suspecting that a patient may have MDS on the basis of some parameters provided by the haematological analyser (low blood cell counts) and in the case that MDS is suspected, neutrophil images are selected and analysed by DysplasiaNet to quantify the dysplastic neutrophils in the smear.

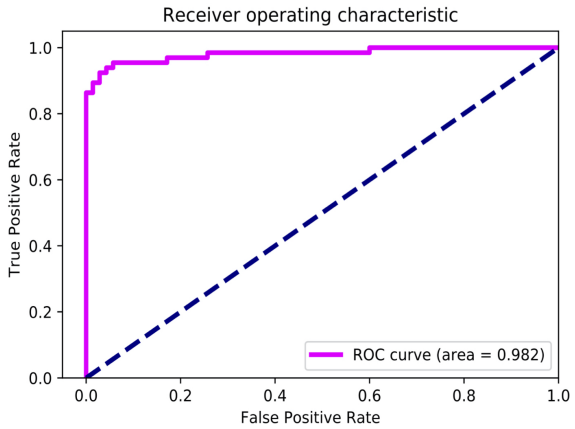
We performed this final test using the proof of concept set detailed in Table 5.1. This set included 66 smears from 32 MDS patients and 70 smears from 40 HC, remarking that none of them were involved in previous training and testing steps. All smears were classified by DysplasiaNet according to whether they belong to a patient with dysplasia or not indicated by the established threshold. Figure 5.5 shows the confusion matrix, where the classification results are compared with the ground truth (confirmed diagnosis). From this matrix, we calculated the sensitivity or true positive rate (TPR), specificity or true negative rate (TNR), precision or positive predictive value (PPV) and overall accuracy as follows:

$$Sensitivity(TPR) = \frac{TP}{TP+FN} = 95.5\% \quad Specificity(TNR) = \frac{TN}{TN+FP} = 94.3\%$$

$$Precision(PPV) = \frac{TP}{TP+FP} = 94\%$$

$$Overall\ accuracy = \frac{TP+TN}{TP+TN+FP+FN} = 94.85\%$$

As it is shown in Figure 5.5, the area under the ROC curve was 0.982.



		Predicted: Dysplastic	Predicted: Normal
True	Dysplastic	63 (95.5 %)	3 (4.5 %)
	Normal	4 (5.7 %)	66 (94.3 %)

Figure 5.5: *ROC curve (left) and confusion matrix results of the proof of concept set classification using DysplasiaNet (right). Sensitivity was 95.5 %, specificity 94.3 %, precision 94 % and overall accuracy 94.85 %.*

5.4.2 Model interpretability

In order to get insight into the inner functioning of DysplasiaNet we implemented three techniques: 1) t-distributed Stochastic Neighbour Embedding (t-SNE) [91] feature reduction to visualize and compare the features extracted by the network versus the original images; 2) visualization of activation maps from the last convolutional block to understand which parts of the images are contributing the most to the classification; and 3) feature map visualization to interpret which features are being extracted by the CNN.

t-SNE feature reduction

We implemented this technique using all *test by cell set* images projecting 268,203 pixels from each image onto two dimensions for visualization, see Figure 5.6 (left). Purple and red dots represent dysplastic and normal neutrophils, respectively. This visualization may help us to understand how difficult is the differentiation between both groups. Alternatively, Figure 5.6 (right) represents the two-dimensional reduction of the 4,096 internal features extracted from the images by the last convolutional block of DysplasiaNet. As shown in the figure, the images are clearly separated in two clusters.

Activation maps visualization

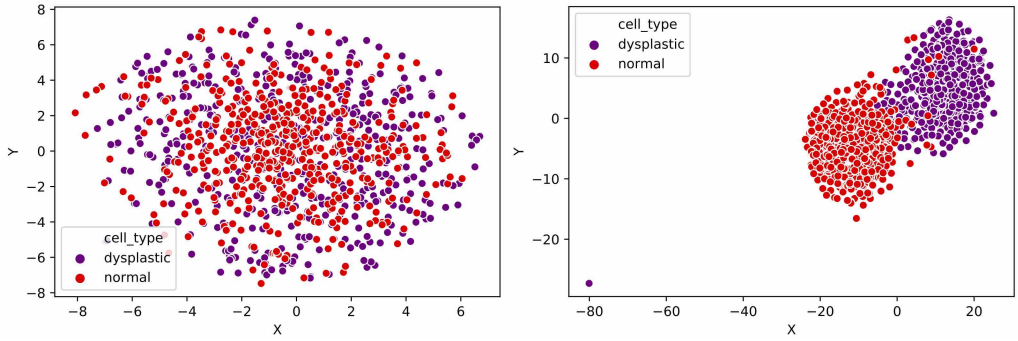


Figure 5.6: *Dimensionality reduction using t-SNE over the test by cell images. Reduction from the pixel values of the original images (left) and reduction from the features extracted from the last convolutional block of DysplasiaNet (right). Purple dots represent the dysplastic neutrophils and red dots the normal ones. DysplasiaNet features result in two clusters that can be easily separated with few overlapping.*

Activation maps were extracted using Gradient-weighted Class Activation Mapping (Grad-CAM) technique [92]. Figure 5.7 presents six neutrophil images from the test by cell set that were correctly classified as dysplastic and normal neutrophils, respectively. It also shows their corresponding activation maps extracted from the last convolutional block of DysplasiaNet (Block 4), as well as activation maps overlaid on the original images. The yellow colour in the activation map indicates higher activations, while purple indicates lower activations. In all cases, stronger activations are observed in the cells, thus showing that the network uses cell pixels for its recognition. In dysplastic neutrophils (left), we see that the network is also using some information of the background. The score is the CNN output indicating the classification result.

Feature maps visualization

In this respect, it is interesting to visualize how granularity is characterized by the network. Figure 5.8 displays two examples of dysplastic and normal neutrophils, respectively. Left side shows the original images, while right side displays the respective feature maps given by the third convolutional layer. Recall that feature maps are the quantitative representations of the original image processed through the subsequent convolutional filters. Figure 5.8 (right) shows how one of the deep CNN layers assigned high values to the pixels corresponding to the cytoplasmic granules in the normal neutrophil image (>200),

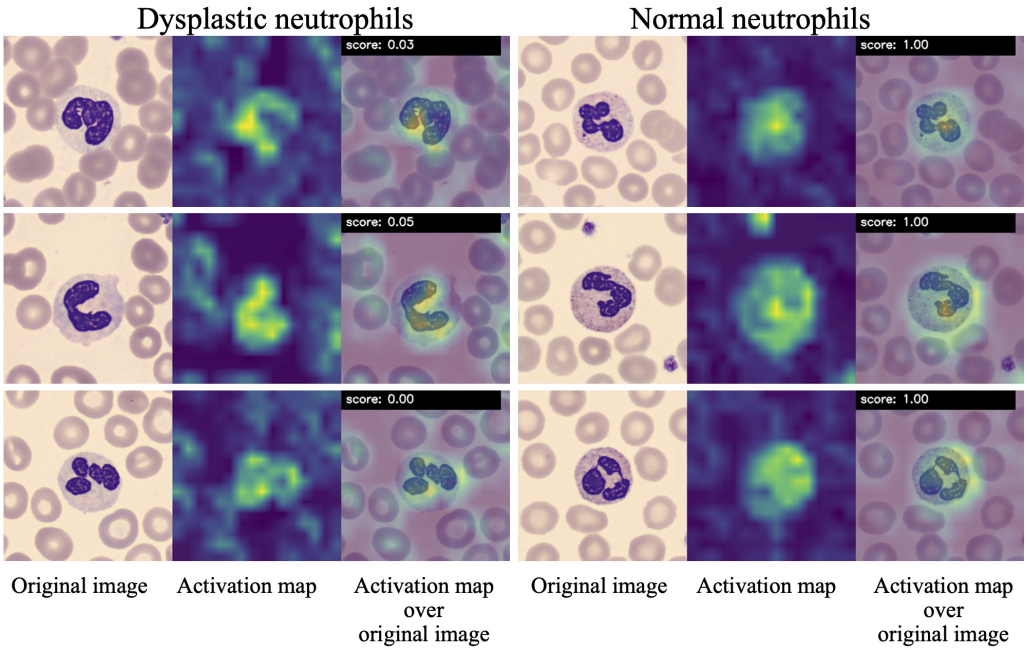


Figure 5.7: *Dysplastic and normal neutrophils, their corresponding activation maps and the activation maps overlaid on the original image. The score value is the CNN output.*

while these values were < 175 in the hypogranulated neutrophil.

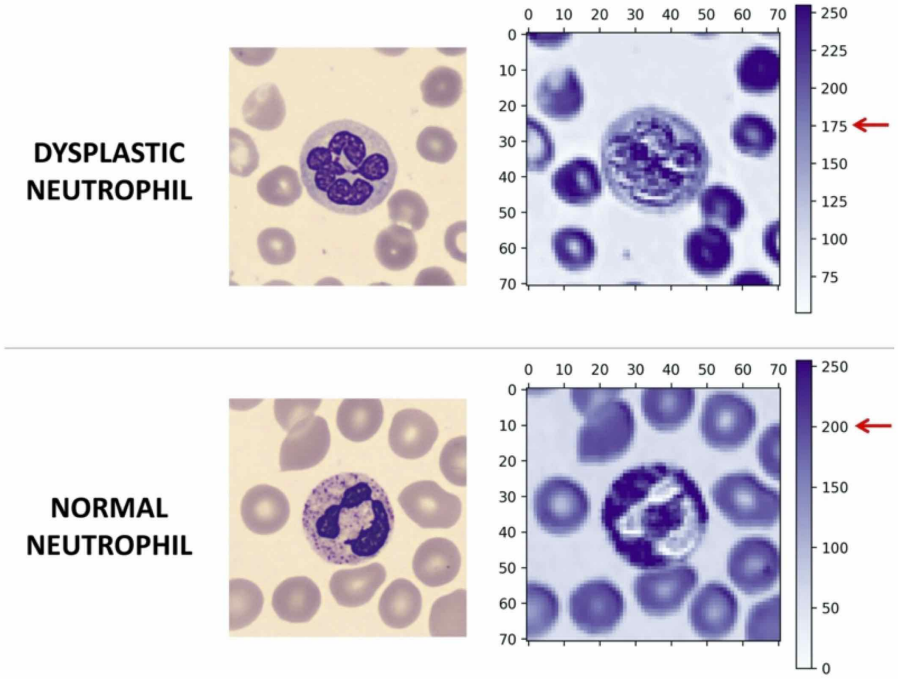


Figure 5.8: *Examples of feature maps extracted from DysplasiaNet. The original images of a dysplastic and normal neutrophils (left) are compared to their correspondent feature maps extracted from the third convolutional layer of DysplasiaNet. The model assigns higher values to the pixels when it detects granules in the neutrophil cytoplasm.*

5.5 Discussion

Our group has been working in developing tools for the automatic classification of normal leukocytes and abnormal blood cells in acute leukaemia, lymphoma or malaria using machine learning [27], [30], [93] and, more recently, deep learning techniques [94]. In this work, we have proposed a tool for the automatic detection of dysplastic cells in peripheral blood present in patients with MDS, which is a problem scarcely addressed in the literature [83].

Cytoplasmic hypogranulation in neutrophils associated to MDS may undergo high rates of interobserver variability. Predictions based on deep learning might provide additional objective tools to address this issue [95]. This work presents the development of a CNN model (DysplasiaNet) designed and trained from scratch for the automatic recognition of the cytoplasmic hypogranularity in dysplastic neutrophils. Other MDS morphological features were considered out of the scope of this work because they are more evident and easily detected by visual inspection of the PB smear [72]. DysplasiaNet was designed by our multidisciplinary research group having in mind its deployment in a real set up. The CNN was trained to predict if a single neutrophil image is dysplastic or not depending on a single quantitative score, which was related to the cytoplasmic granularity level, in a similar way to how clinical pathologists apply their skills and consider that a neutrophil is dysplastic if it shows sufficiently low granularity. To the authors knowledge, this is the first CNN model that has been built and validated for MDS support tool to assist in the diagnosis, with potential to be integrated in the workflow of clinical laboratory as a secondary evaluation under MDS suspicion.

Compared to the previously published paper using deep CNNs to differentiate MDS and aplastic anaemia in [83], we remark some relevant differences in our work. The first is that we focused in the automatic detection of the dysplastic neutrophils showing hypogranularity circulating in blood in patients with MDS. We selected hypogranularity because, based

on practical experience in the clinical laboratory, this is the most difficult feature to detect under the microscope by visual inspection, being one of the most frequent morphological abnormalities observed in blood in MDS patients. Moreover, it might be challenging to differentiate dysplastic from normal neutrophils when the cytoplasmic granulation is fine and scarce. Another significant difference to consider is that DysplasiaNet is a simpler and lightweight CNN model that performs feature extraction and classification with high accuracy (94.85 %).

Training a deep CNN from scratch requires a large amount of high quality-data. For this reason, we compiled a dataset with 20,670 images obtained from healthy controls and patients with MDS and they were annotated by expert clinical pathologists. This high number of images was sufficient to obtain a model with high accuracy using the hold out method. This was further verified with 5 and 10-fold cross-validation performed for the architecture evaluation. Accuracies obtained with both methods were similar. Another aspect is related to the CNN parameter initialization. Given the random nature of the initial parameters, it is possible to face problems like the vanishing/exploding gradients, where the gradients become zero or too high, respectively [1]. This produces low accuracies (< 50 %) during training as occurred in the first step of the training, where models M5, M6 y M8 did not improve their 50 % accuracy (see Figure 5.3). These three models were discarded and we kept only the models that reached high accuracies.

Model interpretation techniques implemented are visual guided approaches to identify the features that influence the decisions of deep learning models. Nowadays, these techniques are being applied to reduce the black box nature and understand the functioning of models for medical image diagnosis [96]. With the first technique applied, t-SNE, image features are embedded into two dimensions in order to visualize if classes are separable by linear classifiers [97]. With the comparison of the embedding of the original image pixels versus the embedding of features extracted from the last DysplasiaNet convolutional block (see Figure 5.6), we highlight the importance of the correct training of the network

for the extraction of relevant characteristics for the subsequent image classification. The subtle morphological differences between normal and dysplastic neutrophils represent a challenge when training a model for diagnosis.

The second technique applied, Grad-CAM, was used to generate heatmaps to highlight the important regions in the image for predicting a class. The result is called activation map. In the field of medical images, its application in images for diabetic retinopathy classification is particularly relevant [98]. In our image classification work, we obtained higher activations (yellow colour) in the area where the cells are located (see Figure 5.7). This shows that the model is 'looking' at the cell to extract important information, in the same way that the clinical pathologist focuses his analysis on the cell and not on other portions of the image. It is important to clarify that this is a coarse localization technique, and as result, it is not possible to see clear differences among normal or dysplastic cell activation maps. However, as a future work, it is possible to implement more advanced techniques of guided gradients to achieve more fine-grained localization. As an alternative, we extracted the feature maps from an intermediate convolutional layer of DysplasiaNet. This was the third technique implemented (see Figure 5.8). Feature maps show that in deep layers of the network, the cytoplasm granulation was detected by assigning high values to the pixels corresponding to the normal cytoplasmic granules, while these values were lower in the hypogranulated dysplastic neutrophil. Both techniques together, Grad-CAM and feature map extraction, show that DysplasiaNet works extracting features from the cell and detecting the cytoplasm granularity.

For the practical use of the model in MDS diagnosis, we adjusted a threshold for the prediction of a patient status based on the smear review. We envision that DysplasiaNet would be used by the clinical pathologist as follows. In the case of a new patient presents to the hospital with anaemia and/or leukopenia and/or thrombocytopenia, high mean corpuscular volume and blast flag provided by the automated haematological analyser, an assessment of the PB smear morphology is required. If MDS is suspected in the PB re-

view, the clinical pathologist might select the neutrophil images from the smear and enter them to DysplasiaNet. This predictive model is in fact and algorithm that would be easily implemented in any computer device as a secondary evaluation tool.

A practical concern for the implementation may be related to the quality of the smear preparations. It is well known that the stain is particularly critical in assessing neutrophil granulation. In the routine workflow of clinical laboratories, sometimes the smears could be understained due to fluctuations of the reagents in the automated stainer. In this circumstance, all the neutrophils are seen as hypogranulated no mater they are normal or dysplastic. In this case, the clinical pathologist would detect that the smear is not appropriate and would revise the staining process. AI models are trained and assessed with smears prepared using a standardized staining. It is expected that the pathologist, before entering the images to DysplasiaNet for their automatic recognition, will follow the same criteria than when doing visual inspection.

In addition to the high performance of DysplasiaNet, there are technical aspects that may facilitate its real implementation. Conformed by four convolutional blocks and two fully-connected layers, DysplasiaNet is a simple and lightweight CNN model compared to other more complex architectures like Inception [39] and Resnet [99]. Therefore, the proposed model does not require graphics processing units (GPUs) to be trained and tested. On the other hand, DysplasiaNet was developed using open source software TensorFlow [11]. A big advantage of open software is the reduction of development, implementation and maintenance costs because no license is needed and software libraries can be freely updated. As a next step, it is possible to implement DysplasiaNet in a web browser to make it easily accessible and usable, taking advantage of open source libraries such as TensorFlowjs [100].

5.6 Conclusion

The main contribution of this work is a CNN-based model for the automatic detection of hypogranulated neutrophils in peripheral blood smears. We have proposed a methodology to design a simple and lightweight architecture that was trained from scratch and tested with our dataset of 20,670 images. It was compiled from the clinical practice and annotated by expert pathologists. The model was configured to obtain the best balance between sensitivity and specificity. We envision the utility of the model implemented as a secondary evaluation tool for MDS diagnosis integrated in the clinical laboratory workflow.

Chapter 6

Modular system for neutrophils detection and MDS diagnosis

Abstract

This chapter presents a modular system for the classification of MDS conformed by two CNNs designed and trained from scratch. The complete system was developed in two steps: 1) Development and test of Model 1; and 2) Integration of Model 1 with Model 2 and proof of concept of the complete system. Model 1 is a pre-classifier to detect eight types of PB cells: neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes, erythroblasts and platelets. This new CNN has as reference the pre-trained Vgg-16 adopted in Chapter 4. However, in this work Model 1 is designed from scratch to obtain a customized classifier with less layers and parameters. The images classified as neutrophils are selected automatically to be the input to Model 2 (DysplasiaNet). The output of the system is an indicator of weather the patient presents MDS or not, which can used as a diagnostic assistance. As a final test of the system, we performed a proof of concept where the input are neutrophils from PB smears and the output is the detection or absence of MDS. The total accuracy obtained was 94.8 %.

Although a global system is out of the scope of this thesis, we performed an additional experiment to test Model 1 and evaluate its performance for the classification of different abnormal cells for which the model was not trained. It was found that abnormal mononuclear cells, important for the detection of serious diseases, are classified mainly in four classes: lymphocyte, monocyte, immature granulocyte and erythroblast. Based on these results, further steps could be done to integrate into the system other models to detect leukemia and lymphoma.

6.1 Introduction

Chapter 5 of this thesis is focused on the diagnosis of myelodysplastic syndromes (MDS) by means of the DysplasiaNet CNN. The input is a set of images of neutrophils selected

by the pathologist from a patient smear. In this chapter, our aim is to mimic the selecting role of the pathologist by a model that performs an automatic pre-classification among all cell types present in the patient's smear. With this purpose, we developed the modular system presented in Figure 6.1. The system is conformed by two CNNs: Model 1, the pre-classifier, and Model 2, DysplasiaNet. The input is a set with all the smear images (normal and abnormal) and the output from Model 1 is the classification in eight groups: neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes (myelocytes, promyelocytes, metamyelocytes), erythroblasts and platelets. The images classified as neutrophils by Model 1 are selected automatically and are the input for DysplasiaNet, which finally indicates if the patient presents MDS or not, as described in Chapter 5.

In a general system, the other groups of abnormal cells could be the input of other specialized classifiers to detect specific diseases such as acute leukaemia or lymphoma. In this work, we just discard such cells and follow the track of neutrophils through DysplasiaNet.

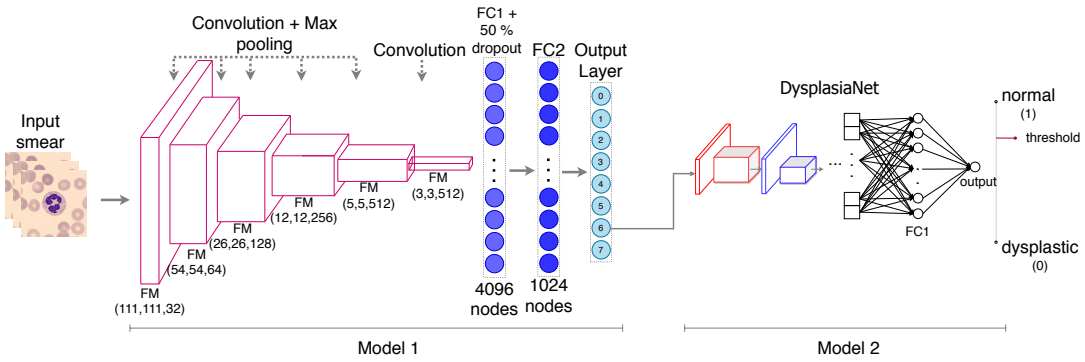


Figure 6.1: *Integrated system for the classification of dysplastic cells. Model 1 receives as input all the smear images (normal and abnormal) and performs the classification in eight groups: neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes (myelocytes, promyelocytes, metamyelocytes), erythroblasts and platelets. Images classified as neutrophils are selected automatically as the input for Model 2 (DysplasiaNet) to finally indicate if a patient presents MDS or not.*

The modular system was developed in two steps: 1) Development and test of Model 1; and 2) Integration of Model 1 with Model 2 and proof of concept of the complete system.

These steps are described in Sections 6.2 and 6.3, respectively. Section 6.4 presents one additional test to evaluate the Model 1 capability for the classification of other types of abnormal cells.

6.2 Development and test of Model 1

6.2.1 Architecture design and training

In order to design Model 1 we have the reference of the CNN classifier developed in Chapter 4 [20] for the classification of the eight cell groups circulating in peripheral blood. In that case, we started with a pre-trained Vgg-16, which was fine tuned with our dataset. However, Model 1 is a new CNN designed from scratch, following a similar approach to the one adopted in Chapter 5. Based on what was reported in [85] about the effects of transfer learning and our experience with Chapter 5, we wanted to obtain a customized classifier with less layers and parameters trained with the dataset presented in Chapter 3.

Details about the number and type of layers and parameters are presented in Table 6.1.

Model 1 is conformed by five blocks of convolutional and max pooling layers, a sixth block with one convolutional layer and a final fully connected block. The number of filters in each convolutional layer (conv2D) increases in powers of two, for example, for block 1 there are $2^5 = 32$ filters, for block 2 there are $2^6 = 64$ filters and so on. Block 5 and 6 have the same number of filters: $2^9 = 512$. The size of the filters of all convolutional layers is 3x3 pixels. Regarding the fully connected block, it is conformed by three layers: the first fully connected (FC1) with 4,096 nodes where a dropout of 50 % is applied at the output, the second (FC2) conformed by 1,024 nodes and the output layer (FC3) with eight nodes, one for each group to classify. The rightmost column contains the number of parameters (NPB_i) required to train in each block $i = 1, 2, \cdots 6$. This value depends on the

	Layers	Filter size	Number of filters	Number of parameters
Block 1	conv2D	3x3	32	896
	max pooling : F = 2 and S= 2 output shape: (111, 111, 32)			
Block 2	conv2D	3x3	64	18,496
	max pooling : F = 2 and S= 2 output shape: (54, 54, 64)			
Block 3	conv2D	3x3	128	73,856
	max pooling : F = 2 and S= 2 output shape: (26, 26, 128)			
Block 4	conv2D	3x3	256	295,168
	max pooling : F = 2 and S= 2 output shape: (12, 12, 256)			
Block 5	conv 2D	3x3	512	1,180,160
	max pooling : F = 2 and S= 2 output shape: (5, 5, 512)			
Block 6	conv2D	3x3	512	2,359,808
	output shape: (3, 3, 512)			
Fully connected block	FC1+ 50 % dropout: 4,096 nodes			18,878,464
	FC2 : 1,024 nodes			4,195,328
	FC3 : 8 nodes			8,200

Table 6.1: *Model 1 architecture description*

filter size (FS_i), the number of filters (NF_i) and the number of feature maps that process each layer (NFM_i). The following expression can be used to compute this parameter for the convolutional layers:

$$NPB_i = (FS_i \times NF_i \times NFM_i) + NB_i$$

where NB_i is the number of bias terms of each filter, so that it is equal to the number of filters NF_i.

As an example, for Block 1 we have:

- Filter size $FS_1 = 3 \times 3 = 9$
- Number of filters $NF_1 = 32$

- Number of feature maps, the number of channels of the input images, $NFM_1 = 3$

- Number of bias $NB_1 = 32$

Consequently, the number of parameters in Block 1 is $NPB_1 = (9 \times 32 \times 3) + 32 = 896$.

For Block 2 the number of parameters is $NPB_2 = (9 \times 64 \times 32) + 64 = 18,496$. In this case, $NFM_2 = 32$ because the output from Block 1, which is the input to Block 2, is a volume of 32 feature maps of size 111×111 pixels. Similarly, the parameter numbers are obtained for the subsequent convolutional blocks.

The fully connected layers have many more parameters than the convolutional layers and are calculated differently. The number of parameters of FC1 is calculated having into account the output volume from Block 6 of size (3, 3, 512), which contains a total of $3 \times 3 \times 512 = 4,608$ features. These features are the input to the 4,096 nodes, so in this layer we have $(4,608 \times 4,096) + 4,096 = 18,878,464$ parameters.

For FC2 we have $(4,096 \times 1,024) + 1,024$ *biases* $= 4,195,328$ parameters. Finally, for the output layer (FC3) we have $(1,024 \times 8) + 8$ *biases* $= 8,200$ parameters. Model 1 has a total of 27,010,376 parameters, which are much less than for Vgg-16 model with around 133 millions of parameters.

Model 1 was trained using the dataset published in [19] to classify eight groups of PB cells: neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes (myelocytes, promyelocytes, metamyelocytes), erythroblasts and platelets. The neutrophils subset was modified replacing 1,500 normal neutrophils by hypogranulated neutrophils. The dataset was split in three subsets: train, validation and test. The train set was balanced up to 3,000 images/group applying data augmenting as explained in Chapter 4. For validation and test we selected 200 images/group. Model 1 was trained for 30 epochs. The training and validation accuracy progression are presented in Figure 6.2.

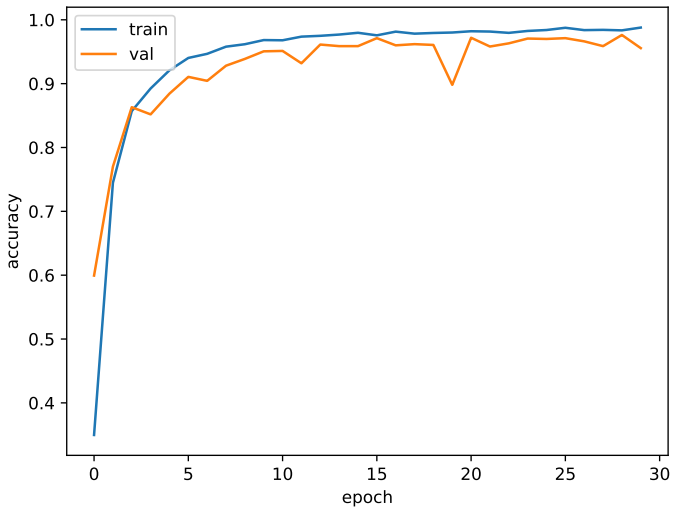


Figure 6.2: *Model 1 training and validation accuracies progression for 30 epochs.*

6.2.2 Model 1 testing

After training, the model performance was tested using the test set. The results of classification are presented in the confusion matrix in Figure 6.3. True positive rates (TPR) for basophil, eosinophil, erythroblast, lymphocyte, monocyte and platelet groups are above 97 %. Neutrophils TPR was 89.5 % due to the confusion of 4.5 % of them as ig (immature granulocytes) and 3.5 % as erythroblasts. The ig class obtained the lower TPR with 83 %. A 10 % of these cells were missclassified in the monocyte group. This behavior may be because these two types of cells share morphological features. The overall accuracy obtained was 95.4 %.

Figure 6.4 presents the ROC curves calculated for this test. The ROC curve for each class was obtained considering a "*one versus all*" approach. Class 0 to 7 correspond to classes basophil, eosinophil, erythroblast, ig, lymphocyte, monocyte, neutrophil and platelet, respectively. Areas under each curve are very close to one, indicating the accurate performance of the model. The macro-average ROC curve is the average of all class curves.

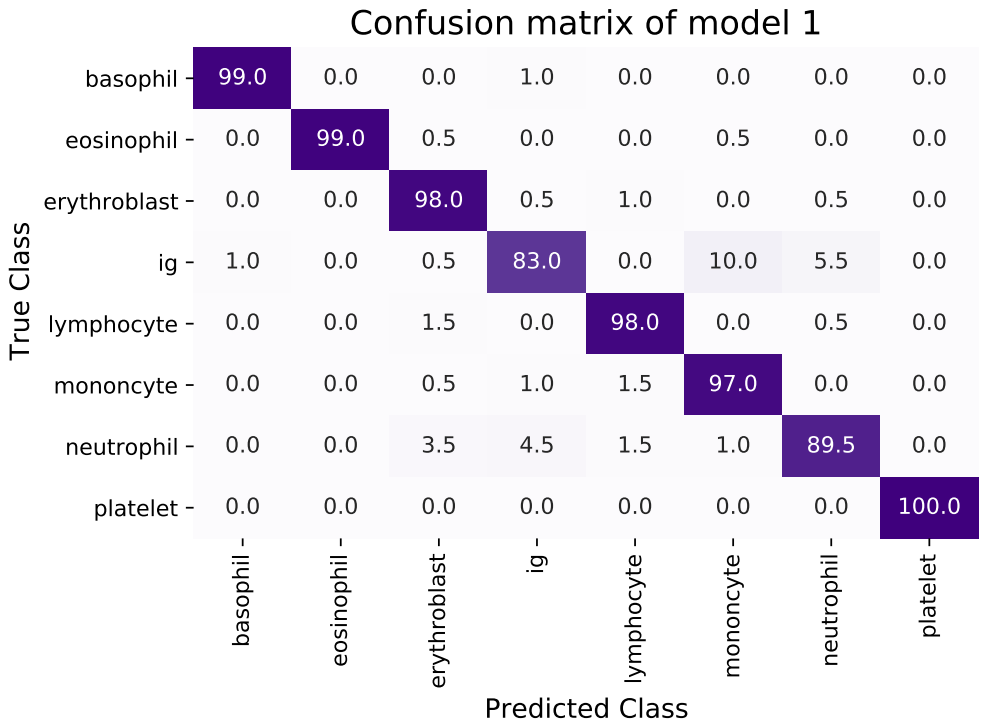


Figure 6.3: *Confusion matrix of the results of testing Model 1. Rows represent the true label (True Class) and the columns the label predicted by the network (Predicted Class). True positive rates are values in the principal diagonal. The overall accuracy obtained was 95.4%. The neutrophil subset contains both normal and hypogranulated classes. ig: immature granulocytes.*

Another visual representation of the performance of Model 1 is presented in Figure 6.5. The left plot is a reduction of the $299 \times 299 \times 3 = 268,203$ pixels of each input cell image to a two-dimensional map to visualize them using t-SNE reduction. The result is a set of data points corresponding to each image. The data points are concentrated and very close, making difficult to separate them by cell type. In fact, this was expected due to the similar morphological characteristics existing between the cells. In contrast to this observation, the right plot presents the reduction of the $3 \times 3 \times 512 = 4,608$ features extracted from Block 6 of Model 1 to a two-dimensional map. The resulting data points apear clearly differentiated and separated. This may indicate that the features extracted from the network are allowing the classification of the eight types of cells. The same t-SNE reduction was applied in the development of DysplasiaNet in Chapter 5.

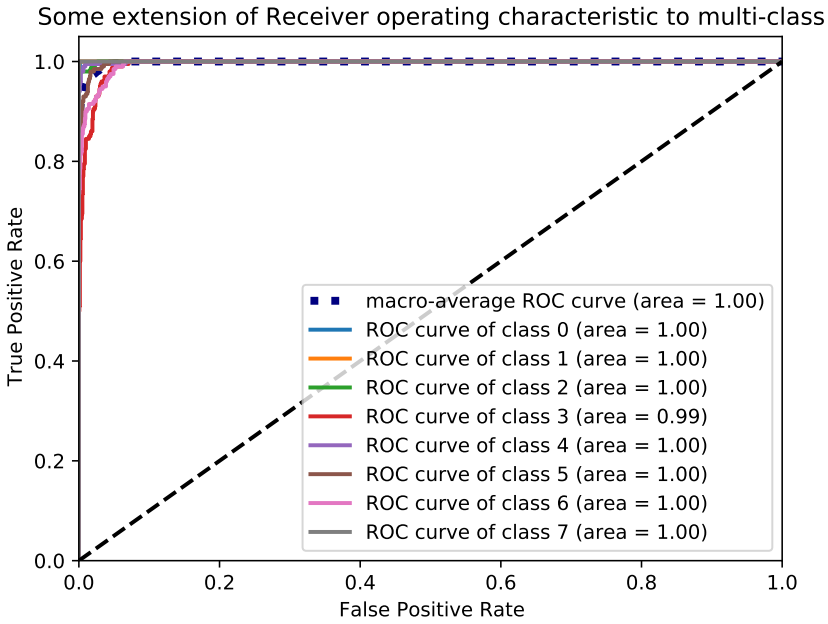


Figure 6.4: *ROC Curves obtained from testing Model 1. Class 0 to 7 correspond to classes basophil, eosinophil, erythroblast, ig, lymphocyte, monocyte, neutrophil and platelet, respectively. The macro-average roc curve is the average of all class curves.*

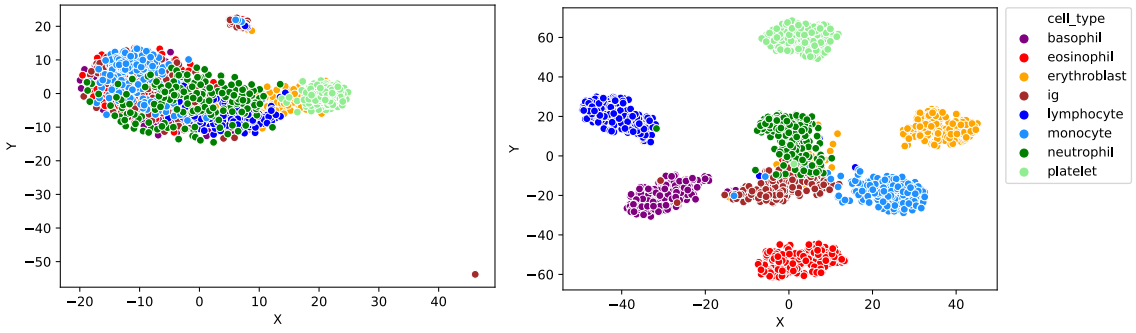


Figure 6.5: *Feature reduction into a two-dimensional map using the t-SNE technique. Left plot are data points corresponding to the reduction of 268,203 pixels of each image. The right plot corresponds to the reduction of 4,608 features extracted from the block 6 of Model 1.*

6.3 Model integration and proof of concept

In this step the output of the identified neutrophils is connected to the input of DysplasiaNet to obtain the final system (see Figure 6.1). The input is a set of images from an individual PB smear and the output is the detection or absence of MDS. Using this setup, the pathologist in search of dysplastic neutrophils does not need to select the neutrophils

to feed DysplasiaNet. Instead, all leukocytes from the PB smear are given to the integrated system. Model 1 will identify all neutrophils and automatically will send them to DysplasiaNet.

In order to test the performance of the complete system, we used the proof of concept set described in Chapter 5, which includes a total of 136 smears (see Table 5.1). Figure 6.6 presents the classification results in a confusion matrix. We see that a number of 63 smears obtained from MDS patients and 66 smears obtained from healthy controls were correctly classified, which means a global accuracy of 94.8 %. It is important to remark that these results are the same obtained for DysplasiaNet in Figure 5.5. This means that Model 1 is effective in filtering the neutrophils correctly and redirecting them to DysplasiaNet.

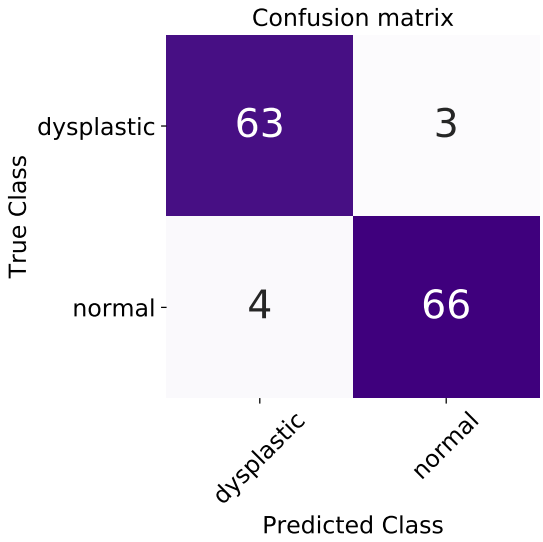


Figure 6.6: *Confusion matrix of the results obtained in the proof of concept of the modular system. A total of 63 smears from patients with MDS and 66 smears from healthy controls were correctly classified to obtain a global accuracy of 94.8 %.*

6.4 Towards a modular system for the classification of blood cells

This thesis is framed within the global objectives of the research group to advance in the development of computational methodologies to assist the clinical pathologist to reach a feasible, objective and reliable diagnosis in myelodysplastic syndromes, acute leukemia or lymphoma by means of the automatic classification of abnormal PB cells using convolutional neural networks.

The idea of performing a pre-classification to identify the cells of interest for a subsequent specialized classifier could be extended to other diseases beyond MDS. Although a global system is out of the scope of this thesis, we decided to perform an additional experiment to test Model 1 and evaluate its performance in the classification of abnormal cells, present in serious diseases, but for which the system was not trained.

Our hypothesis is that, even though Model 1 was not trained to classify cells like blasts and abnormal lymphocytes, it might be used with no further re-training to classify these "unknown" cell types into the groups of cells with which they share some morphological features. Different abnormal cell types were selected and classified with Model 1.

To carry out this experiment, we selected the following four cell groups:

1. Abnormal lymphoid cells. Conformed by nine groups of 200 cells each: CLL (chronic lymphocytic leukemia), FL (follicular lymphoma), HCL (hairy cell leukemia), LGLL-T (large granular lymphocyte leukemia), MCL (mantle cell lymphoma), PCL (plasma cell leukemia), PL (B and T prolymphocytic leukemia), SMZL (splenic marginal zone lymphoma) and SS (Sézary syndrome).

2. Blast cells (BLASTS). Conformed by two types of cells: 629 blasts of acute lymphocytic leukemia and 2,968 blasts corresponding to acute myeloid leukemia. A

total of 3,597 blasts.

3. Reactive lymphoid cells (RL): 2,324 variant lymphocytes corresponding to patients with infections.

4. Normal lymphocytes (LY): 2,498 lymphocytes.

For this test, a new group named "*other*" was configured in order to avoid false positives when classifying other cell types for which the model was not trained. Model 1 output for each image is a set of eight probabilities (one for each cell group) of the image to belong to a certain group. The sum of the eight probabilities is equal to 1 due to the softmax function at the output layer, as explained in [20]. Usually, the cell is classified in the group with the highest probability. In this case, we defined that the highest probability must be greater than 0.5. If this criterion is not met, the image will be classified in the group *other*, indicating that Model 1 did not clearly identify the input cell class.

The results are presented in the confusion matrix of Figure 6.7. Rows are the true classes of the input images and columns are the predicted labels. As expected, most of the abnormal lymphoid cells were classified into lymphocyte and monocyte groups. Specifically, most of the cells from CLL (96 %), FL (88.5 %), LGL-T (90 %), MCL (80.5 %) and SS (71.5 %) were classified as lymphocytes. On the other hand, most of the reactive lymphocytes (67.7 %), blasts (67.5 %) and SMZL cells (65.5 %) were classified in the monocyte group. Prolymphocytes were distributed mainly in two groups: lymphocytes (48.5 %) and monocytes (46 %). Most of plasma cells were also classified in two groups: lymphocytes (61.5 %) and erythroblasts (24 %). Regarding normal lymphocytes, 99 % were correctly classified.

Cells classified by Model 1 as lymphocytes, monocytes, immature granulocytes (ig) and erythroblasts can be redirected to other models specialized in detecting different types of leukemia [28] and lymphoma [29].

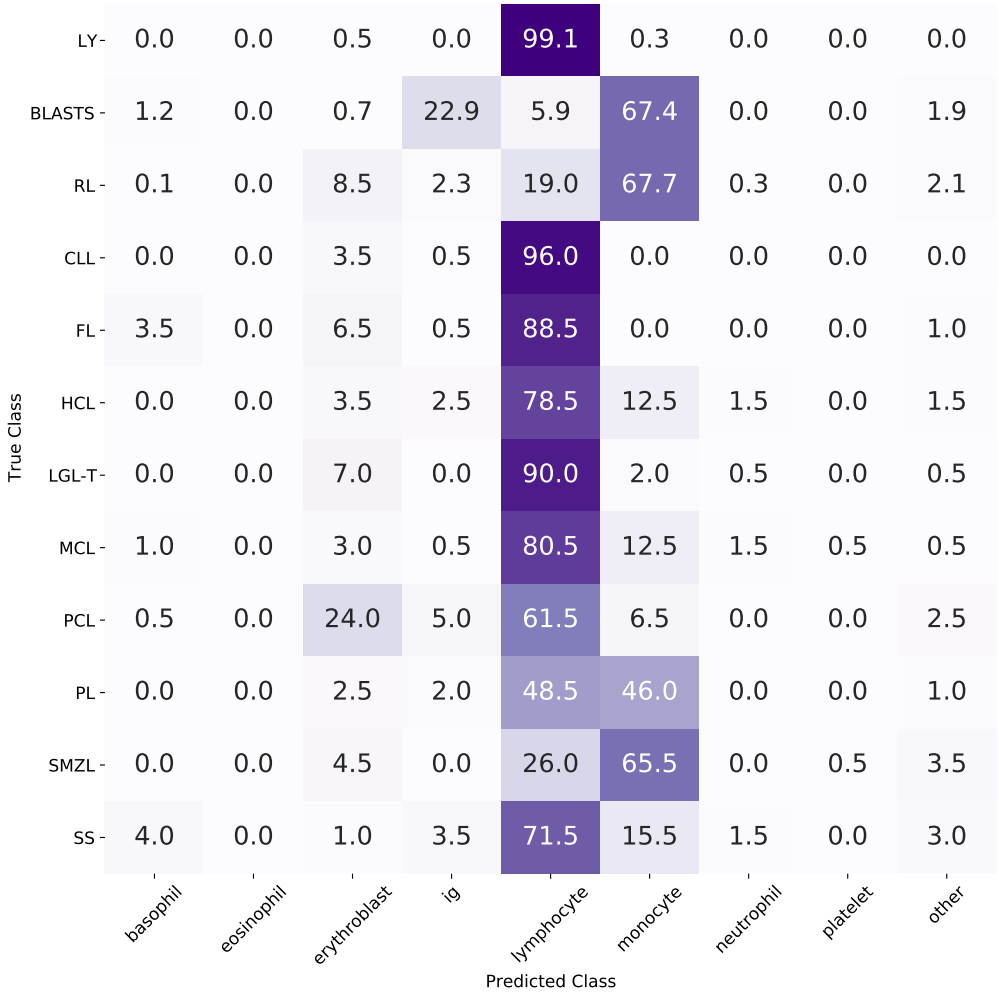


Figure 6.7: *Confusion matrix obtained from testing Model 1 using other types of abnormal cells. Rows are the true classes or labels and columns are the predicted labels. The class "other" gathers the images that do not obtain the maximum probability value greater than 0.5. LY: lymphocytes, RL: reactive lymphocytes, CLL: chronic lymphocytic leukemia, FL: follicular lymphoma, HCL: hairy cell leukemia, LGL-T: T-large granular lymphocytes, MCL: mantle cell lyphoma, PCL: plasma cell leukemia, PL: prolymphocytes, SMZL: splenic marginal zone lymphoma and SS: Sézary syndrome.*

6.5 Conclusion

This chapter presents the development of an integrated system for the detection of MDS. The system is conformed by two CNNs trained from scratch and customized for the classification of PB cells. The inputs to the system are all types of cells observable in pe-

ripheral blood. Cells classified as neutrophils by Model 1 are redirected automatically to Model 2, DysplasiaNet, to detect dysplastic neutrophils. The overall accuracy of the test of Model 1 (95.4 %), obtained when training from scratch, is comparable to the obtained performing fine tuning in [20]. The high accuracy (94.8 %) in the proof of concept of the integrated system for MDS diagnosis shows that Model 1 is performing pre-classification as expected.

These satisfactory results suggested us to evaluate the utility of Model 1 as a pre-classifier of other types of abnormal blood cells, besides dysplastic ones, for which it was not trained. It was found that reactive and abnormal mononuclear cells (blasts and abnormal lymphocytes) were classified mainly in four classes: lymphocytes, monocytes, immature granulocytes and erythroblasts. This is encouraging for further works where other models could be integrated to detect malignancies like leukemia and lymphoma.

Chapter 7

Conclusions

This thesis has contributed with a methodology for the automatic classification of peripheral blood cells circulating in blood in healthy controls and in patients with infections, regenerative anaemia and Myelodysplastic Syndromes (MDS) using Convolutional Neural Network (CNN) models.

This thesis has grown through the evolution of different works involving CNNs models with the following targets:

- The automatic recognition of eight groups of cells circulating in blood.
- The automatic recognition of normal and dysplastic neutrophils.
- The design of a new model (DysplasiaNet) built and validated for the MDS diagnosis in individual patients.
- The development of a modular system to predict if the patient shows dysplastic cells or not using all cells circulating in blood.

The conclusions derived along this research are outlined in Section 7.1. The main contributions of this thesis are highlighted in Section 7.2. Finally, some future perspectives are

discussed in Section 7.3.

7.1 Conclusions

Chapter 2 describes the design of an indexed database to organize all the images acquired in the laboratory to facilitate our research, according to the first objective proposed in this thesis. Two relevant conclusions can be highlighted:

- The acquisition and import algorithms developed facilitated the selection of datasets for training and testing the different classification models throughout this thesis.
- We demonstrated that with the use of high-quality datasets of images, labelled by clinical pathologists, it is possible to design high performance and customized CNN models without requiring pre-trained models.

As it is detailed in Chapters 2 and 4, one objective of this thesis was the extraction of deep features from blood cell images using different architectures of CNNs described in the literature. These features were used to train a machine learning classifier to recognize the corresponding images. From the obtained results the following conclusion is derived:

- Pre-trained CNNs were useful to extract features from different types of blood cells, which were used to train a SVM model to classify four and eight groups of blood cell images, obtaining accuracy values of 94 % and 90.5 %, respectively.

The main results presented in Chapter 4 correspond to the next objective of this thesis, the design and assessment of a CNN recognition system for eight groups of PB cells. The conclusions are the following:

- Starting from a state-of-the art general architecture, a fine-tuning procedure was established to develop an end-to end classifier trained using a dataset over 17,000 blood cell images obtained from clinical practice in the daily workflow of the laboratory.

- A new classification system was designed involving CNNs (Vgg-16 and Inceptionv3) to discriminate among eight classes of blood cells: five-types of normal leukocytes, erythroblasts (cells circulating in blood in regenerative anaemia) and metamyelocytes, myelocytes and promyelocytes (circulating in blood in infections).

- Most of the model hyperparameters in this work were selected empirically. From our experience, we concluded that using a minimum of 1,000 images per group, a number of epochs around 80 and tuning the last two four blocks of our proposed CNNs, it was possible to obtain satisfactorily high accuracies.

- The performance obtained when testing the system was truly satisfactory, showing excellent precision, sensitivity and specificity values. To summarize, the overall classification accuracy for Vgg-16 was 96.2 %. Automatic recognition of this number of groups of normal blood cells using deep learning models has not been reported previously in the literature.

- It was proved that validation accuracies were higher when the networks were trained with the balanced dataset. Testing performed when the networks were trained with the unbalanced dataset showed a reduction of the overall test accuracy from 96.2 % to 93.9 % for Vgg-16 and from 94.9 % to 92.5 % for Inceptionv3. Training with the unbalanced dataset produced many fluctuations in the Vgg-16 validation accuracy. By increasing the number of images and balancing the dataset, it was observed that these fluctuations decreased.

The work detailed in Chapter 5 corresponds to the objective centred in the design and testing of a CNN model for the automatic recognition of hypogranulated dysplastic neu-

trophil images from peripheral blood. This chapter also includes the objective of a proof of concept to validate the model to support the diagnosis of MDS. The following conclusions were obtained:

- A number of eight CNN models were generated from scratch by varying convolutional blocks, number of nodes in layers and fully connected layers. The CNN model showing the highest overall accuracy (95.5 %) in the automatic recognition of hypogranulated neutrophils in peripheral blood was selected. This new deep learning model was named DysplasiaNet.

- DysplasiaNet was trained from scratch for the automatic recognition of hypogranulated neutrophils using 20,670 images from our database. Model training and selection were performed using the hold-out method. In addition, the final architecture was evaluated using five and ten-fold cross-validation. Accuracies obtained with these two methods were similar which indicates that hold out was a correct choice for model development.

- Conformed by four convolutional blocks and two fully-connected layers, DysplasiaNet, with 72,977 parameters, is a simple and lightweight CNN model compared to other more complex architectures like Inception or ResNet.

- It was possible to calculate a single quantitative score to relate the amount of cytoplasmic granularity with the recognition of normal or dysplastic neutrophil by the CNN.

- Experimental testing gave an optimal threshold value (12 %) for the minimum proportion of dysplastic neutrophils in the patient's smear for the MDS prediction by the CNN. It is remarkable that this low value is good enough to make the CNN achieve excellent performance metrics.

- Model interpretability techniques implemented gave an insight into the inner functioning of DysplasiaNet. t-SNE allowed the comparison of the embedding of the

original image pixels versus the embedding of features extracted from DysplasiaNet, confirming that subtle morphological differences between normal and dysplastic neutrophils represent a challenge when training a model for diagnosis. Using Grad-CAM, we visualized the parts of the images that DysplasiaNet is using to classify normal and dysplastic neutrophils. Finally, feature maps visualization showed that the CNN provided different numerical values to the pixels corresponding to the cytoplasmic granules of the images. These values were lower in the hypogranulated neutrophils with respect to normal.

- The proof of concept to validate DysplasiaNet using a total of 70 smears from 40 healthy controls and 66 smears from 32 MDS patients showed sensitivity, specificity and precision values of 95.5 %, 94.3 % and 94.0 %, respectively, for the automatic detection of MDS.

- AI models should be trained and assessed with smears stained using standardized methods as the one used in this thesis. Understained smears may affect the automatic classification of DysplasiaNet.

- This thesis presents the first CNN model that was built and validated for MDS to assist in the diagnosis.

The next objective proposed in this thesis, as detailed in Chapter 6, was the integration of two CNN models in a system for the automatic recognition of MDS using images of all type of cells circulating in blood. The conclusions obtained are detailed below:

- A modular sequential system was designed for the automatic selection of neutrophils and the further recognition of those showing dysplasia. The system detected MDS using a set of images obtained from individual blood smears, showing accuracy values of 94.8 % in the proof of concept.

- Blast cell images in patients with different leukaemia subtypes and abnormal lymphocytes in patients with different B or T-lymphomas were recognized by the first model as mononuclear cells (lymphocytes, monocytes, immature granulocytes or erythroblasts). Therefore, this model could be a preclassification system to be integrated with other CNNs models specialized in leukaemia or lymphoma recognition.

7.2 Contributions

We have proposed CNN models that make feasible its integration in the workflow of the clinical laboratory for the automatic classification of cells circulating in blood. The more relevant contributions of this thesis are the following:

1. Generation of a published dataset. As it is explained in Chapter 3, a high-quality dataset of 17,092 blood cell images has been acquired, labelled, published and made available at the Mendeley repository to contribute to the research community.

2. CNN models for the automatic recognition of normal and abnormal blood cell images:

    - A classification system has been developed involving a CNN (Vgg-16) trained to discriminate among eight classes of cells circulating in PB, as it is described in Chapter 4.

    - A CNN-based model (DysplasiaNet) has been designed with a simple and lightweight architecture for the automatic detection of hypogranulated neutrophils in PB smears, as it is detailed in Chapter 5. The model can be used as a secondary evaluation tool for MDS diagnosis integrated in the clinical laboratory workflow. It may be easily implemented and made accessible and usable in any computer device.

- A modular system for MDS diagnosis, as explained in Chapter 6, has been built conformed by two CNN models, the first to recognize neutrophils and the second to discriminate among normal and dysplastic ones.

7.3 Future perspectives

The results obtained in this thesis open some issues that could be further developed. Some of them are outlined below:

- Analysis of the feature maps generated for the intermediate convolutional layers of DysplasiaNet showed that the CNN provided different numerical values to the pixels corresponding to a specific morphological characteristic of neutrophils: the cytoplasmic granularity. It would be interesting the study of the feature maps generated for the intermediate convolutional layers of the CNN models used for the classification of other abnormal cells circulating in blood, as blast cells in acute leukaemia or abnormal lymphocytes in different types of lymphoma. In general, representation analysis [101], [102] and explicability [96] techniques could be used to obtain a step by step explanation of the model decision processes and determine which input features contribute the most to the classification. This is particularly important in a medical diagnosis scenario.

- The modular structure used in the last part of this thesis could be implemented in different problems, where a preclassification could be linked to other CNN models specialized in the recognition of different malignancies.

- Techniques named "few-shot learning" [103] could be explored to train CNN models with few images to generalize to unfamiliar categories without extensive retraining. This would be very useful for the automatic detection of critical morphological

findings, such as blue-green cytoplasmic inclusions of death in PB smears in patients suffering from acute liver damage.

- A practical concern when using automatic blood cell image classifiers is related to staining, particularly affecting when using images from different laboratories. It may be interesting to explore the use of CNN and deep generative models [104] to match certain colour normalization and build "normalized" training and testing sets.

- The final diagnosis of hematological diseases is performed based on the integration of clinical signs and results of several complementary tests, including blood morphology. It would be interesting to train a CNN not only with images, but also with numerical data obtained from other complementary tests to improve the classification accuracy and contribute to the integrated diagnosis.

7.4 Publications derived from this thesis

7.4.1 Conferences

1. Acevedo, A., Merino, A., Boldú, L., Molina, Á., Alférez, S., and Rodellar, J. (2020). A new model for the automatic detection of dysplastic cells in peripheral blood: dysplasianet. In International Journal of Laboratory Hematology. Abstract Proceedings of the 2020 Annual Meeting of the International Society for Laboratory Hematology.

2. Acevedo, A., Merino, A., Alférez, S., Boldú, L., Molina, Á., and Rodellar, J. (2019). Automatic detection of dysplastic cells using deep learning. (Vol. 41, pp. i–ii). Abstract Proceedings of the 2019 Annual Meeting of the International Society for Laboratory Hematology. doi:https://doi.org/10.1111/ijlh.13104

3. Acevedo, A., Merino, A., Alférez, S., Puigví, L., and Rodellar, J. (2018). Training a convolutional neural network for automatic classification of peripheral blood cells. (Vol. 40, pp. 1–165). Abstract Proceedings of the 2018 Annual Meeting of the International Society for Laboratory Hematology. doi:10.1111/ijlh.12923.

4. Acevedo, A., Alférez, S., Merino, A., Puigví, L., and Rodellar, J. (2017). Automatic recognition system of peripheral blood cell images using deep features. Abstract presented at IEEE International Symposium on Biomedical Imaging – ISBI 2017.

5. Acevedo, A., Alférez, S., Merino, A., Puigví, L., and Rodellar, J. (2016). Automatic recognition system of nucleated peripheral blood cell images. In International Journal of Laboratory Hematology (Vol. 38, pp. i–xi). Abstract Proceedings of the 2016 Annual Meeting of the International Society for Laboratory Hematology. doi:https://doi.org/10.1111/ijlh.12568

7.4.2 Journal papers

1. Acevedo, A., Merino, A., Boldú, L., Molina, Á., Alférez, S., & Rodellar, J. (2021). A new convolutional neural network predictive model for the automatic recognition of hypogranulated neutrophils in myelodysplastic syndromes. Computers in Biology and Medicine. (Impact factor: 3.434). Document in revision.

2. Acevedo, A., Merino, A., Alférez, S., Molina, Á., Boldú, L., & Rodellar, J. (2020). A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. Data in Brief, 30. doi:`https://doi.org/10.1016/j.dib.2020.105474`. (Impact factor: 0.970).

3. Acevedo, A., Alférez, S., Merino, A., Puigví, L., & Rodellar, J. (2019). Recognition of peripheral blood cell images using convolutional neural networks. Computer Methods and Programs in Biomedicine, 180. doi:`https://doi.org/10.1016/j.cmpb.2019.105020`. (Impact factor: 3.632).

7.4.3 Awards

1. ISLH 2020 Trainee Travel Award. Work acknowledged: A new model for the automatic detection of dysplastic cells in peripheral blood. International Society for Laboratory Hematology.

2. ISLH 2019 Trainee Travel Award. Work acknowledged: Automatic detection of dysplastic cells using deep learning. International Society for Laboratory Hematology.

3. ISLH 2018 Berend Houwen Travel Award. Work acknowledged: Training a Convolutional Neural Network for Automatic Classification of Peripheral Blood Cells. International Society for Laboratory Hematology.

4. ISLH 2016 Trainee Travel Award. Work acknowledged: Automatic Recognition System of Nucleated Peripheral Blood Cell Images. International Society for Laboratory Hematology.

7.5 Other publications

With the experience acquired in machine learning, CNN and classification of blood cell images, the author of this thesis could help in other works conducted by other members of the research group.

1. Merino, A., Vlagea, A., Molina, A., Egri, N., Laguna, J., Barrera, K., Boldú, L., **Acevedo, A.**, Diaz, M., Sibina, F., Bascón, F., Sibila, O., Juan, M. and Rodellar, Jose. (2020) Atypical lymphoid cells circulating in blood in COVID-19 infection: morphology, immunophenotype and prognosis value. Journal of Clinical Pathology [Internet]. jclinpath-2020-207087. (Impact factor: 2.460).

2. Molina, A., Alférez, S., Boldú, L., **Acevedo, A.**, Rodellar, J., and Merino, A. (2020). Sequential classification system for recognition of malaria infection using peripheral blood cell images. Journal of Clinical Pathology,73(10), 665–670. doi:10.1136/jclinpath-2019-206419. (Impact factor: 2.460).

3. Alférez, S., Merino, A., **Acevedo, A.** et al. (2019). Color clustering segmentation framework for image analysis of malignant lymphoid cells in peripheral blood. Med Biol Eng Comput 57, 1265–1283 (2019). https://doi.org/10.1007/s11517-019-01954-7. (Impact factor: 2.116).

4. Boldú, L., Merino, A., Alférez, S., Molina, A., **Acevedo, A.** and Rodellar, J. (2019). Automatic recognition of different types of acute leukaemia in peripheral blood by image analysis. Journal of Clinical Pathology, 72:755-761. doi: 10.1136/jclinpath-2019-205949 (Impact factor: 2.460).

5. Puigví, L., Merino, A., Alférez, S., Boldú, L., **Acevedo, A.** and Rodellar, J. (2019). Quantitative cytologic descriptors to differentiate CLL, Sézary, granular, and villous lymphocytes through image analysis. American Journal of Clinical Pathology, Volume 152, Issue 1, July 2019, Pages 74–85, https://doi.org/10.1093/ajcp/aqz025. (Impact factor: 2.094).

6. Rodellar, J., Alférez, S., **Acevedo, A.**, Molina, A. and Merino, A. (2018). Image processing and machine learning in the morphological analysis of blood cells. Int J Lab Hem. 2018; 40 (Suppl. 1): 46- 53. https://doi.org/10.1111/ijlh.12818. (Impact factor: 2.141).

7. Puigví, L., Merino, A., Alférez, S., **Acevedo, A.** and Rodellar, J. (2017). New quantitative features for the morphological differentiation of abnormal lymphoid cell images from peripheral blood. Journal of Clinical Pathology 2017;70:1038-1048. doi: 0.1136/jclinpath-2017-204389. (Impact factor: 2.460).

Bibliography

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[2] J. Stewart, *Multivariable Calculus*, 8th ed. Cengage Learning, 2015, ISBN: 9781305266643.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning*. 2006, ISBN: 978-0-387-31073-2. DOI: 10.1117/1.2819119.

[4] A. Merino, *Manual de Citología de Sangre Periférica y Líquidos Biológicos*, 2nd ed. Madrid: Editorial panamericana, 2019, ISBN: 978-84-9110-262-5.

[5] S. Raschka and V. Mirjalili, *Python Machine Learning*, Third Edit. Packt, 2019, p. 725, ISBN: 978-1-78995-575-0.

[6] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, 2nd ed. The MIT Press, 2018, ISBN: 9780262039406.

[7] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Incorporated, 2019, ISBN: 9781492032649.

[8] F. Chollet, *Deep Learning With Python*, 2nd ed. Manning Publications Company, 2018, ISBN: 9781617294433.

[9] A. Vedaldi, *Matconvnet*, https://www.vlfeat.org/matconvnet/, 2014.

[10] P. S. Foundation, *Python language reference*, https://www.python.org/doc/, 2020.

[11] Martın Abadi, Ashish Agarwal, Paul Barham, *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: http://tensorflow.org/.

[12] F. Chollet *et al.*, *Keras*, https://keras.io, 2015.

[13] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2.

[14] W. McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[16] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

[17] M. Waskom and the seaborn development team, *Mwaskom/seaborn*, version latest, Sep. 2020. DOI: 10.5281/zenodo.592845. [Online]. Available: https://doi.org/10.5281/zenodo.592845.

[18] Oracle, *Mysql reference manual*, https://dev.mysql.com/doc/refman/8.0/en/, 2020.

[19] A. Acevedo, A. Merino, S. Alférez, Á. Molina, L. Boldú, and J. Rodellar, "A dataset of microscopic peripheral blood cell images for development of automatic recognition systems," *Data in Brief*, vol. 30, p. 105 474, Jun. 2020, ISSN: 23523409. DOI: 10.1016/j.dib.2020.105474. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2352340920303681.

[20] A. Acevedo, S. Alférez, A. Merino, L. Puigví, and J. Rodellar, "Recognition of peripheral blood cell images using convolutional neural networks," *Computer Methods and Programs in Biomedicine*, vol. 180, p. 105 020, 2019, ISSN: 01692607. DOI: 10.1016/j.cmpb.2019.105020. [Online]. Available: https://doi.org/10.1016/j.cmpb.2019.105020.

[21] S. Alférez, A. Merino, L. Bigorra, and J. Rodellar, "Characterization and automatic screening of reactive and abnormal neoplastic b lymphoid cells from peripheral blood," *International Journal of Laboratory Hematology*, vol. 38, no. 2, pp. 209–219, 2016, ISSN: 1751553X. DOI: 10.1111/ijlh.12473.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 1097–1105, 2012. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[23] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014*, pp. 1–11, 2014. DOI: 10.5244/c.28.6. arXiv: 1405.3531.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations (ICRL)*, pp. 1–14, 2015, ISSN: 09505849. DOI: 10.1016/j.infsof.2008.09.005. [Online]. Available: http://arxiv.org/abs/1409.1556.

[25] E. Piaton, M. Fabre, I. Goubin-Versini, *et al.*, "Recommandations techniques et règles de bonne pratique pour la coloration de May-Grünwald-Giemsa : revue de la littérature et apport de l'assurance qualité," *Annales de Pathologie*, vol. 35, no. 4, pp. 294–305, Aug. 2015, ISSN: 0242-6498. DOI: 10.1016/J.ANNPAT.2015.05.019. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0242649815001194?via%7B%5C%%7D3Dihub.

[26] S. Alférez, A. Merino, L. Bigorra, L. Mujica, M. Ruiz, and J. Rodellar, "Automatic recognition of atypical lymphoid cells from peripheral blood by digital image analysis," *American Journal of Clinical Pathology*, vol. 143, no. 2, pp. 168–176, Jan. 2015, ISSN: 0002-9173. DOI: 10.1309/AJCP78IFSTOGZZJN.

[27] S. Alférez, A. Merino, A. Acevedo, L. Puigví, and J. Rodellar, "Color clustering segmentation framework for image analysis of malignant lymphoid cells in peripheral blood," *Medical & Biological Engineering & Computing*, Feb. 2019, ISSN: 1741-0444. DOI: 10.1007/s11517-019-01954-7. [Online]. Available: https://doi.org/10.1007/s11517-019-01954-7.

[28] L. Boldú, A. Merino, S. Alférez, A. Molina, A. Acevedo, and J. Rodellar, "Automatic recognition of different types of acute leukaemia in peripheral blood by image analysis," *Journal of Clinical Pathology*, vol. 72, no. 11, pp. 755–761, Nov. 2019, ISSN: 0021-9746. DOI: 10.1136/jclinpath-2019-205949. [Online]. Available: http://dx.doi.org/10.1136/jclinpath-2019-205949.

[29] J. Rodellar, S. Alférez, A. Acevedo, A. Molina, and A. Merino, "Image processing and machine learning in the morphological analysis of blood cells," *International journal of laboratory hematology*, vol. 40, pp. 46–53, 2018. DOI: 10.111/ijlh.12818.

[30] A. Merino, L. Puigví, L. Boldú, S. Alférez, and J. Rodellar, "Optimizing morphology through blood cell image analysis," *International Journal of Laboratory Hematology*, vol. 40, no. S1, pp. 54–61, 2018. DOI: 10.1111/ijlh.12832. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/ijlh.12832.

[31] H. Walker, W. Hall, and J. Hurst, *Clinical Methods: The History, Physical, and Laboratory Examinations*. Butterworths, 1990, ISBN: 9780409900774. [Online]. Available: https://books.google.es/books?id=qtz3JRBuwRIC.

[32] *The easyCell assistant*, Medicacorp, 2019. [Online]. Available: http://www.medicacorp.com/.

[33] *Vision Hema*, West Medica, 2019. [Online]. Available: http://hema.wm-vision.com/.

[34] Cellavision, *Cellavision DM9600*, 2019. [Online]. Available: http://www.cellavision.com/.

[35] L. Puigví, A. Merino, S. Alférez, A. Acevedo, and J. Rodellar, "New quantitative features for the morphological differentiation of abnormal lymphoid cell images from peripheral blood," *J Clin Pathol*, vol. 70, no. 12, pp. 1038–1048, 2017. DOI: 10.1136/jclinpath-2017-204389.

[36] X. Zheng, Y. Wang, G. Wang, and J. Liu, "Fast and robust segmentation of white blood cell images by self-supervised learning," *Micron*, vol. 107, pp. 55–71, 2018, ISSN: 0968-4328. DOI: https://doi.org/10.1016/j.micron.2018.01.010.

[37] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, ISSN: 14764687. DOI: 10.1038/nature14539.

[38] O. Russakovsky, J. Deng, H. Su, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015, ISSN: 15731405. DOI: 10.1007/s11263-015-0816-y. [Online]. Available: http://dx.doi.org/10.1007/s11263-015-0816-y.

[39] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2818–2826, ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.308. [Online]. Available: http://arxiv.org/abs/1512.00567.

[40] S. Hoo-Chang, H. R. Roth, M. Gao, *et al.*, "Deep convolutional neural networks for computer-aided detection : Cnn architectures , dataset characteristics and transfer," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016, ISSN: 0278-0062. DOI: 10.1109/TMI.2016.2528162.

[41] M. Saraswat and K. Arya, "Automated microscopic image analysis for leukocytes identification: A survey," *Micron*, vol. 65, pp. 20–33, 2014.

[42] J. Rawat, H. Bhadauria, A. Singh, and J. Virmani, "Review of leukocyte classification techniques for microscopic blood images," in *Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on*, IEEE, 2015, pp. 1948–1954.

[43] A. Kumar, J. Kim, D. Lyndon, M. Fulham, and D. Feng, "An ensemble of fine-tuned convolutional neural networks for medical image classification," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 31–40, 2017, ISSN: 2168-2194. DOI: 10.1109/JBHI.2016.2635663.

[44] D. Raví, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G.-Z. Yang, "Deep learning for health informatics," *IEEE Journal of Biomedical and Health Informatics*, 2017, ISSN: 2168-2194. DOI: 10.1109/JBHI.2016.2636665.

[45] F. Cabitza and G. Banfi, "Machine learning in laboratory medicine: Waiting for the flood?" *Clinical Chemistry and Laboratory Medicine (CCLM)*, vol. 56, no. 4, pp. 516–524, 2018. DOI: 10.1515/cclm-2017-0287. [Online]. Available: https://doi.org/10.1515/cclm-2017-0287.

[46] M. Habibzadeh, A. Krzyżak, and T. Fevens, "White blood cell differential counts using convolutional neural networks for low resolution images," in *International Conference on Artificial Intelligence and Soft Computing*, Springer, Springer Berlin Heidelberg, 2013, pp. 263–274, ISBN: 978-3-642-38610-7.

[47] A. I. Shahin, Y. Guo, K. M. Amin, and A. A. Sharawi, "White blood cells identification system based on convolutional deep neural learning networks," *Computer Methods and Programs in Biomedicine*, vol. 0, pp. 1–12, 2017, ISSN: 18727565. DOI: 10.1016/j.cmpb.2017.11.015. eprint: https://doi.org/10.1016/j.cmpb.2017.11.015.

[48] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724, ISBN: 9781479951178. DOI: 10.1109/CVPR.2014.222.

[49] H. Chougrad, H. Zouaki, and O. Alheyane, "Deep convolutional neural networks for breast cancer screening," *Computer Methods and Programs in Biomedicine*, vol. 157, pp. 19–30, 2018, ISSN: 18727565. DOI: 10.1016/j.cmpb.2018.01.011. [Online]. Available: https://doi.org/10.1016/j.cmpb.2018.01.011.

[50] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[51] Y. LeCun, K. Kavukcuoglu, C. Farabet, *et al.*, "Convolutional networks and applications in vision.," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, vol. 2010, 2010, pp. 253–256. DOI: 10.1109/ISCAS.2010.5537907.

[52] *Cs231n: Convolutional neural networks for visual recognition*, http://cs231n.github.io/convolutional-networks/, Accessed: 2019-01-12, 2019.

[53] L. Buitinck, G. Louppe, M. Blondel, *et al.*, "Api design for machine learning software: Experiences from the scikit-learn project," *arXiv preprint arXiv:1309.0238*, 2013.

[54] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017, ISSN: 00189219. DOI: 10.1109/JPROC.2017.2761740.

[55] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" In *Advances in neural information processing systems*, vol. 27, 2014, pp. 3320–3328, ISBN: 978-1-5090-0620-5. DOI: 10.1109/IJCNN.2016.7727519. [Online]. Available: http://arxiv.org/abs/1411.1792.

[56] Q. Wang, L. Chang, M. Zhou, Q. Li, H. Liu, and F. Guo, "A spectral and morphologic method for white blood cell classification," *Optics & Laser Technology*, vol. 84, pp. 144–148, 2016. DOI: https://doi.org/10.1016/j.optlastec.2016.05.013.

[57] N. Ramesh, B. Dangott, M. E. Salama, and T. Tasdizen, "Isolation and two-step classification of normal white blood cells in peripheral blood smears," *Journal of pathology informatics*, vol. 3, 2012. DOI: 10.4103/2153-3539.93895. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3327044/.

[58] J. Prinyakupt and C. Pluempitiwiriyawej, "Segmentation of white blood cells and comparison of cell morphology by linear and naıve bayes classifiers," *BioMedical Engineering Online*, vol. 14, no. 1, pp. 1–19, 2015, ISSN: 1475925X. DOI: 10.1186/s12938-015-0037-1.

[59] S. H. Rezatofighi and H. Soltanian-Zadeh, "Automatic recognition of five types of white blood cells in peripheral blood," *Computerized Medical Imaging and Graphics*, vol. 35, no. 4, pp. 333–343, 2011. DOI: 10.1016/j.compmedimag.2011.01.003.

[60] S. Nazlibilek, D. Karacor, T. Ercan, M. H. Sazli, O. Kalender, and Y. Ege, "Automatic segmentation, counting, size determination and classification of white blood cells," *Measurement: Journal of the International Measurement Confederation*, vol. 55, pp. 58–65, 2014, ISSN: 02632241. DOI: 10.1016/j.measurement.2014.04.008. [Online]. Available: http://dx.doi.org/10.1016/j.measurement.2014.04.008.

[61] M. C. Su, C. Y. Cheng, and P. C. Wang, "A neural-network-based approach to white blood cell classification," *The Scientific World Journal*, vol. 2014, no. 1, 2014, ISSN: 1537744X. DOI: 10.1155/2014/796371.

[62] Meiyin Wu and Li Chen, "Image recognition based on deep learning," *2015 Chinese Automation Congress (CAC)*, pp. 542–546, 2015. DOI: 10.1109/CAC.2015.7382560. [Online]. Available: http://ieeexplore.ieee.org/document/7382560/.

[63] A. Mathur, A. Tripathi, and M. Kuse, "Scalable system for classification of white blood cells from leishman stained blood stain images," *Journal of Pathology Informatics*, vol. 4, no. 2, p. 15, 2013, ISSN: 2153-3539. DOI: 10.4103/2153-3539.109883. [Online]. Available: http://www.jpathinformatics.org/text.asp?2013/4/2/15/109883.

[64] L. Bigorra, S. Alférez, A. Merino, and J. Rodellar, "Feature analysis and automatic identification of leukemic lineage blast cells and reactive lymphoid cells from peripheral blood cell images," *Journal of Clinical Laboratory Analysis*, vol. 31, no. 2, e22024, 2017. DOI: 10.1002/jcla.22024.

[65] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, 2016, ISSN: 1558254X. DOI: 10.1109/TMI.2016.2535302.

[66] A. Mikolajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, May 2018, pp. 117–122. DOI: 10.1109/IIPHDW.2018.8388338.

[67] Z. Qin, F. Yu, C. Liu, and X. Chen, "How convolutional neural networks see the world — a survey of convolutional neural network visualization methods," *Mathematical Foundations of Computing*, vol. 1, no. 2, pp. 149–180, 2018, ISSN: 2577-8838. DOI: 10.3934/mfc.2018008. [Online]. Available: http://dx.doi.org/10.3934/mfc.2018008.

[68] M. Cazzola and L. Malcovati, "Myelodysplastic syndromes - Coping with ineffective hematopoiesis," *New England Journal of Medicine*, vol. 352, no. 6, pp. 536–538, 2005, ISSN: 00284793. DOI: 10.1056/NEJMp048266.

[69] S. Swerdlow, E. Campo, H. NL, J. ES, P. SA, S. H, and T. J, *WHO Classification of Tumours of Haematopoietic and Lymphoid Tissues*, 4th ed. International Agency for Research on Cancer, 2017, ISBN: 978-9-28-324494-3.

[70] J. E. Goasguen, J. M. Bennett, B. J. Bain, *et al.*, "Proposal for refining the definition of dysgranulopoiesis in acute myeloid leukemia and myelodysplastic syndromes," *Leukemia research*, vol. 38, no. 4, pp. 447–453, Apr. 2014, ISSN: 0145-2126. DOI: 10.1016/j.leukres.2013.12.020. [Online]. Available: https://doi.org/10.1016/j.leukres.2013.12.020.

[71] D. Bowen, D. Culligan, S. Jowitt, S. Kelsey, G. Mufti, D. Oscier, and J. Parker, "Guidelines for the diagnosis and therapy of adult myelodysplastic syndromes.," eng, *British journal of haematology*, vol. 120, no. 2, pp. 187–200, Jan. 2003, ISSN: 0007-1048 (Print). DOI: 10.1046/j.1365-2141.2003.03907.x.

[72] G. Zini, "How I investigate difficult cells at the optical microscope," *International Journal of Laboratory Hematology*, no. August, pp. 1–8, 2020, ISSN: 1751553X. DOI: 10.1111/ijlh.13437.

[73] R. Shouval, J. A. Fein, B. Savani, M. Mohty, and A. Nagler, "Machine learning and artificial intelligence in haematology," *British Journal of Haematology*, 2020, ISSN: 13652141. DOI: 10.1111/bjh.16915.

[74] N. Radakovich, M. Nagy, and A. Nazha, "Artificial Intelligence in Hematology: Current Challenges and Opportunities," *Current Hematologic Malignancy Reports*, 2020, ISSN: 1558822X. DOI: 10.1007/s11899-020-00575-4.

[75] A. Rehman, N. Abbas, T. Saba, S. I. ur Rahman, Z. Mehmood, and H. Kolivand, "Classification of acute lymphoblastic leukemia using deep learning," *Microscopy Research and Technique*, vol. 81, no. 11, pp. 1310–1317, 2018, ISSN: 10970029. DOI: 10.1002/jemt.23139.

[76] T. Pansombut, S. Wikaisuksakul, K. Khongkraphan, and A. Phon-On, "Convolutional neural networks for recognition of lymphoblast cell images," *Computational Intelligence and Neuroscience*, vol. 2019, 2019, ISSN: 16875273. DOI: 10.1155/2019/7519603.

[77] S. Muhammad, A. Muhammad, M. Adnan, Q. Muhammad, A. Majdi, and M. K. Khan, "Medical Image Analysis using Convolutional Neural Networks A Review," *Journal of Medical Systems*, pp. 1–13, 2018.

[78] N. Radakovich, M. Nagy, and A. Nazha, "Machine learning in haematological malignancies," *The Lancet Haematology*, vol. 7, no. 7, e541–e550, 2020, ISSN: 23523026. DOI: 10.1016/S2352-3026(20)30121-6. [Online]. Available: http://dx.doi.org/10.1016/S2352-3026(20)30121-6.

[79] S. Shafique and S. Tehsin, "Acute Lymphoblastic Leukemia Detection and Classification of Its Subtypes Using Pretrained Deep Convolutional Neural Networks.," eng, *Technology in cancer research & treatment*, vol. 17, p. 1 533 033 818 802 789, Jan. 2018, ISSN: 1533-0338 (Electronic). DOI: 10.1177/1533033818802789.

[80] A. Ghosh, S. Singh, and D. Sheet, "Simultaneous localization and classification of acute lymphoblastic leukemic cells in peripheral blood smears using a deep convolutional network with average pooling layer," in *Industrial and Information Systems (ICIIS), 2017 IEEE International Conference on*, IEEE, 2017, pp. 1–6.

[81] A. T. Sahlol, P. Kollmannsberger, and A. A. Ewees, "Efficient Classification of White Blood Cell Leukemia with Improved Swarm Optimization of Deep Features," *Scientific Reports*, vol. 10, no. 1, pp. 1–12, 2020, ISSN: 20452322. DOI: 10.1038/s41598-020-59215-9.

[82] J. Mori, S. Kaji, H. Kawai, *et al.*, "Assessment of dysplasia in bone marrow smear with convolutional neural network," *Scientific Reports*, vol. 10, no. 1, pp. 1–8, 2020, ISSN: 20452322. DOI: 10.1038/s41598-020-71752-x. [Online]. Available: https://doi.org/10.1038/s41598-020-71752-x.

[83] K. Kimura, Y. Tabe, T. Ai, *et al.*, "A novel automated image analysis system using deep convolutional neural networks can assist to differentiate MDS and AA," *Scientific Reports*, vol. 9, no. 1, pp. 1–9, 2019, ISSN: 20452322. DOI: 10.1038/s41598-019-49942-z. [Online]. Available: http://dx.doi.org/10.1038/s41598-019-49942-z.

[84] D. A. Arber, A. Orazi, R. Hasserjian, *et al.*, "The 2016 revision to the World Health Organization classification of myeloid neoplasms and acute leukemia," *Blood*, vol. 127, no. 20, pp. 2391–2405, 2016.

[85] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, "Transfusion: Understanding transfer learning for medical imaging," in *Advances in neural information processing systems*, 2019, pp. 3347–3357.

[86] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, PMLR 15:315-323, 2011*, vol. 15, pp. 315–323, 2011, ISSN: 15208532. [Online]. Available: http://proceedings.mlr.press/v15/glorot11a.html.

[87] C. Aggarwal, *Neural Networks and Deep Learning*, 1st ed. New York: Springer Nature, 2018, p. 497, ISBN: ISBN 978-3-319-94463-0. DOI: https://doi.org/10.1007/978-3-319-94463-0.

[88] 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*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html.

[89] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2015. arXiv: 1412.6980.

[90] T. A. Lasko, J. G. Bhagwat, K. H. Zou, and L. Ohno-Machado, "The use of receiver operating characteristic curves in biomedical informatics"," *Journal of Biomedical Informatics*, vol. 38, no. 5, pp. 404–415, 2005. DOI: https://doi.org/10.1016/j.jbi.2005.02.008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1532046405000171.

[91] L. Vandermaaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 1, pp. 1–48, 2008, ISSN: 1532-4435. DOI: 10.1007/s10479-011-0841-3. arXiv: 1307.1662. [Online]. Available: http://www.cs.toronto.edu/%7B~%7Dhinton/absps/tsne.pdf.

[92] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, 2020, ISSN: 15731405. DOI: 10.1007/s11263-019-01228-7. arXiv: 1610.02391.

[93] A. Molina, S. Alférez, L. Boldú, A. Acevedo, J. Rodellar, and A. Merino, "Sequential classification system for recognition of malaria infection using peripheral blood cell images," *Journal of Clinical Pathology*, 2020, ISSN: 14724146. DOI: 10.1136/jclinpath-2019-206419.

[94] A. Merino, A. Vlagea, A. Molina, *et al.*, "Atypical lymphoid cells circulating in blood in COVID-19 infection: morphology, immunophenotype and prognosis value," *Journal of Clinical Pathology*, jclinpath–2020–207 087, Dec. 2020. DOI: 10.1136/jclinpath-2020-207087. [Online]. Available: http://jcp.bmj.com/content/early/2020/12/10/jclinpath-2020-207087.abstract.

[95] K. Naqvi, E. Jabbour, C. Bueso-Ramos, *et al.*, "Implications of discrepancy in morphologic diagnosis of myelodysplastic syndrome between referral and tertiary care centers," *Blood*, vol. 118, no. 17, pp. 4690–4693, Oct. 2011, ISSN: 0006-4971. DOI: 10.1182/blood-2011-03-342642. [Online]. Available: https://doi.org/10.1182/blood-2011-03-342642.

[96] A. Singh, S. Sengupta, and V. Lakshminarayanan, "Explainable deep learning models in medical image analysis," pp. 1–18, 2020. arXiv: arXiv:2005.13799v1.

[97] S. K. Vuppala, M. Behera, H. Jack, and N. Bussa, "Explainable Deep Learning Methods for Medical Imaging Applications," *2020 IEEE 5th International Conference on Computing Communication and Automation, ICCCA 2020*, pp. 334–339, 2020. DOI: 10.1109/ICCCA49541.2020.9250820.

[98] H. Jiang, J. Xu, R. Shi, *et al.*, "A Multi-Label Deep Learning Model with Interpretable Grad-CAM for Diabetic Retinopathy Classification," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2020-July, pp. 1560–1563, 2020, ISSN: 1557170X. DOI: 10.1109/EMBC44109.2020.9175884.

[99] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, 2016, ISSN: 10636919. DOI: 10.1109/CVPR.2016.90. arXiv: 1512.03385.

[100] D. Smilkov, N. Thorat, Y. Assogba, *et al.*, "Tensorflow.JS: Machine learning for the web and beyond," *arXiv*, 2019, ISSN: 23318422. arXiv: 1901.05350.

[101] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of Neural Network Representations Revisited," 2019. arXiv: arXiv:1905.00414v4.

[102] T. Nguyen, M. Raghu, and S. Kornblith, "Do Wide And Deep Networks Learn The Same Things ? Uncovering How Neural Network Vary With Width And Depth," 2019. arXiv: arXiv:2010.15327v1.

[103] S. Jadon, "An overview of deep learning architectures in few-shot learning domain," *ArXiv*, vol. abs/2008.06365, 2020.

[104] D. Tellez, G. Litjens, P. Bándi, W. Bulten, J. M. Bokhorst, F. Ciompi, and J. van der Laak, "Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology," *Medical Image Analysis*, vol. 58, 2019, ISSN: 13618423. DOI: 10.1016/j.media.2019.101544. arXiv: 1902.06543.