UNIVERSITÀ DEGLI STUDI DI PALERMO

Information and Communication Technologies
Dipartimento di Ingegneria
ING-INF/05 ING-INF/03

DEVELOPMENT AND IMPLEMENTATION OF MACHINE LEARNING METHODS FOR THE IIF IMAGES ANALYSIS

IL DOTTORE
ING. VINCENZO TAORMINA

IL COORDINATORE
PROF.SSA ILENIA TINNIRELLO

IL TUTOR
PROF.SSA ILENIA TINNIRELLO

IL CO TUTOR
PROF. DONATO CASCIO

CICLO XXXIII
ANNO CONSEGUIMENTO TITOLO 2021

Index

Introduction

Autoimmune Diseases (AD) are a range of pathologies characterized and determined by a common anomaly, an inexplicable reaction of the *immune system*, which instead of defending our body from external aggressions, damages some parts or tissues as it "mistakes" them for potentially harmful external agents. In short, the body's immune system attacks its own antigens, a sort of *self-aggression* mechanism.

These diseases can occur in patients of all ages and both sexes with a higher incidence among women of childbearing age. Autoimmune Diseases are collectively among the most prevalent diseases in the word, affecting at least 7% of the population. They are a family of more than 80 chronic illnesses and the third major pathological process after cardiovascular disease and cancer. Some examples of autoimmune diseases are type 1 diabetes, rheumatoid arthritis, multiple sclerosis.

Laboratory techniques, capable of indicating and confirming the diagnosis of autoimmune diseases, are based on the research and identification of autoantibodies, in the patient's serum, revealed by the presence of specific complex antigen-antibody reactions [1]. In particular, the recommended method to identify any autoimmune diseases in patients is the IIF (Indirect ImmunoFluorescence) test performed on the substrate of HEp-2 cells (Humane Epithelial cell line) [2][3][4][5][6].

The Humane Epithelial cell line test is carried out by the expert through the visual inspection of a slide, using a fluorescence microscope. The test is mainly based on two steps:

1. estimating fluorescent intensity; this can be positive or negative;
2. identification of the staining patterns; these can be numerous.

Figure I shows some examples of HEp-2 images.

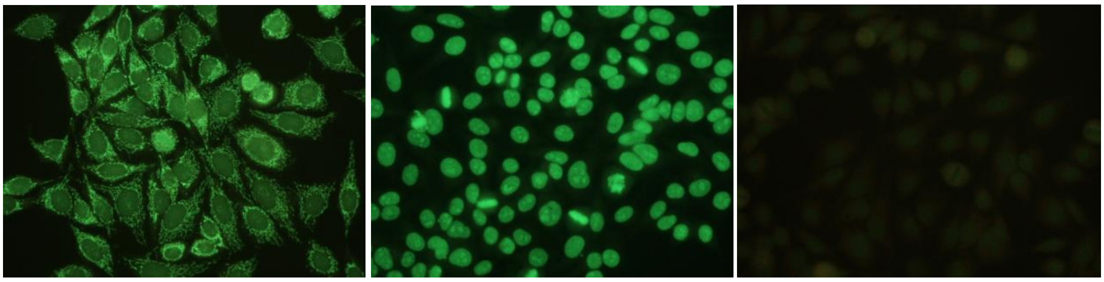


Figure I: IIF images with different fluorescence intensity: on the left is a positive cyctoplasmatic image, in the center there is a positive nuclear image and, on the right, a negative image is shown.

This technique, used for many years as a screening test, offers correct predictive values, however, as for other methods, a serious drawback is the subjectivity in the interpretation of the results, highly dependent on the operator's experience.

The introduction of new and modern approaches based on computer systems represents a cheaper support for the diagnosis of autoimmune diseases. In general, IT systems for diagnostic, prognostic and therapeutic support have increasingly become a necessity especially in hospital centers. The complexity and quantity of the data produced by *medical imaging systems* makes the use of Expert Information Systems more necessary for image analysis and for the management of increasing size databases.

The development of a CAD system (Computer Aided Detection) can therefore represent a valid support for diagnosis [3][7][8]. A CAD expert in the field of indirect immunoflourescence can decrease the variability of the methodology, increasing the level of standardization and helping the specialist.

This PhD thesis involves the design and implementation of a CAD system to support the diagnosis of autoimmune diseases. The first objective of an automatic system applied to the support of diagnosys of autoimmune diseases is to improve diagnostic accuracy, proposing itself as a "second reader". The design of a CAD to support the diagnosis of autoimmune diseases must conceptually follow the guidelines recommended by experts in the sector, in particular the two main requirements concern the identification of the fluorescence intensity and if this is positive, the identification of the staining patterns.

For that reason, the realization of the CAD has been divided into three macro-areas:

1. segmentation of ROIs (Regions Of Interest) that is cells and their cytoplasm;
2. binary classification of the fluorescence intensity of HEp-2 images;
3. multiclass classification of the fluorescence patterns in HEp-2 images.

Referring to the state of the art it is clear that, the implementation and validation of CAD for the computerized support in IIF images analisys is not an easy problem. The main objective of this thesis work is the development and the implementation of a CAD that can be a valid diagnostic support for autoimmune diseases using the most innovative computer vision techniques and artificial intelligence algorithms.

The goodness of the CAD produced was assessed using quantitative numerical merit measures such as AUC (Area Under the ROC Curve), Accuracy (ACC), etc and also exploiting the public DataBase of HEp-2 images made available to the scientific community; this allows the reproducibility of the research shown and the scientific comparison with other international scientific publications.

The content of this PhD thesis is articulated as follows:

- *Chapter 1* gives an overview on autoimmune diseases, the guidelines for the diagnosis, and discusses about the motivations to use an automatic system. Public databases of HEp-2 images available to the scientific community are also described;
- *Chapter 2* presents the components on which a CAD is based: the image processing, segmentation, feature extraction, classification and evaluation;
- *Chapter 3* presents the HEp-2 cells segmentation with different techniques and it discusses the goodness of the approaches used through figures of merit, such as the Dice index and the Jaccard index with the segmentation masks produced by experienced staff considered as the ground truth;
- *Chapter 4* discusses the binary classification in positive or negative fluorescence intensity. Various strategies are compared and evaluated with quantitative figures of merit such as AUC and ACC on a public database;
- *Chapter 5* presents the multiclass classification necessary for a correct association of the different types of patterns. Also, for this problem various strategies are highlighted, some of which are based on deep learning, by comparing them quantitatively on public databases.

Chapter 1 - Automatic system to support the diagnosis of autoimmune diseases

The Antinuclear Antibody (ANA) test is widely used for screening, diagnosing, and monitoring of autoimmune diseases. The most common method to determine ANA is Indirect Immune Fluorescence (IIF), performed by Human Epithelial type 2 (HEp-2) cells, as substrate antigen. The evaluation of ANA consists in a visual analysis of fluorescence intensity and staining patterns.
This chapter describe the Autoimmune Diseases (AD), the guidelines recommended for the diagnosis and the motivation to use a CAD system (Computer Aided Detection) to support the diagnosis of autoimmune diseases. Public databases of HEp-2 images available to the scientific community are also described.

1.1 Autoimmune diseases and laboratory diagnosis

Autoimmune diseases are due to reaction of the immune system to self antigens, occurring through tolerance breakage. The targeted antigens could be common to all kinds of cells or organ specific, and their recognition by humoral or cellular immune effectors could lead to diversified symptoms, depending on pathology [9][10][11]. Autoimmunity is the phenomenon for which the immune system activates its mechanisms towards molecules, cells, and structures of the same organism to which it belongs. The diseases caused by this phenomenon, which are defined as “autoimmune”, are becoming increasingly widespread and include some of the most serious and penalizing conditions for the quality of life of those affected [12].
AD are a family of more than 80 chronic, and often disabling, illnesses that develop when underlying defects in the immune system lead the body to attack its own organs, tissues, and cells. It is important to note that Autoimmune Diseases are collectively among the most prevalent diseases in the word, affecting at least 7% of the population. Because most AD are chronic and incurable, from a public health perspective they constitute a major health problem which, besides causing individual suffering, has high societal costs [12]. These diseases can affect people of all ages and both sexes, with a higher frequency in women of child-bearing age. A autoimmune diseases are multifactorial, and their risk factors are genetic and environmental. The combination of risk factors may vary from one population to another, generating different epidemiological profiles. AD includes celiac disease, Sjogren’s syndrome, systemic lupus erythematosus, multiple sclerosis, diabetes mellitus type 1, rheumatoid arthritis, etc.
Diagnosis of autoimmune pathologies is based on research and identification of Antinuclear Antibodies. In effect, a variety of ANA are found in serum samples obtained from patients with autoimmune diseases. ANA are a group of antibodies produced by the immune system that can mistakenly recognize the structures of the

organism they belong to (autoantibodies). It is observed how ANA attack the cells in the nucleus' structures and or in the cytoplasm. The ANA test identifies the presence of these autoantibodies in the blood. The search of autoantibodies in sera is based on a routine technique performed by immunologists and on Indirect ImmunoFluorescence method [13]. The IIF test is conducted on the HEp-2 (Humane Epithelial cell line) cultured cells used as a substrate [2][3][4]. The binding of ANA on HEp2 cells is revealed by fluorescent antibodies to human immunoglobulin. Highly qualified physicians identified the fluorescence and specific patterns that characterize ANA, through visual inspection of slides using a fluorescence microscope. The fluorescence pattern observed under microscope is characteristic of the nature of the self antigen and of its location in the cell.

ANA are clinically useful markers and IIF which was described in 1958 has become the standard method for the detection of several autoantibodies including ANA [14]. Despite the development of several assays like ELISA or microarrays, IIF on HEp-2 cells substrate, obtained from human laryngeal carcinoma, remains the "gold standard" for ANA screening [15]. HEp-2 cells show a very high nucleus/cytoplasm ratio and, by virtue of their neoplastic nature, present numerous mitotic figures allowing the operator to identify antibodies directed against the cellular antigens expressed during the mitotic phase. Presence of autoantibodies in patient sera has in itself a value of diagnosis, and the ascertaining of their titer and specificity helps to confirm the autoimmune disease and its follow-up.

The IIF test is the gold standard for the diagnosis of autoimmune diseases. Indeed, this test has high sensitivity, but only analytical and not diagnostic specificity, since the positivity for ANA does not automatically confirm the presence of autoimmune disease. Furthermore, the quality of the response is strongly influenced by reader's experience, by the quality of reagents used for testing (characteristics of the cell substrate or fluorochrome-labeled anti-human immunoglobulins used), and by other local factors. As regards the methods immunochemical alternatives, they have the major advantage of being more easily automated and do not require great expertise in interpretation of the results. By contrast, the number of antigenic specificities reportable in the test is certainly lower than that detectable on Hep-2 cells and also, the integrity of the antigenic epitopes theoretically detectable is not always preserved [16]. Hep-2 cells allow for recognition of over 30 different nuclear and cytoplasmic patterns, which are given by upwards of 100 different autoantibodies.

IIF test is examined by the expert with a fluorescence microscope which allows the visual inspection of a slide, properly prepared in the laboratory. Generally, these types of microscopes are equipped with cameras that capture portions of the wells contained in the slides. These images are archived and analyzed by experts who, based on their experience, recognize the fluorescence patterns emitted by HEp-2 cells. In [5] and [6] the guidelines and recommendations for the standardized processing and interpretation of the HEp-2 images and for the diagnosis of autoimmune diseases are described. These publications provide requirements regarding the diagnostic tests used, instructions for laboratory procedures and evaluation, and recommendations for interpretation.

Laboratory techniques to diagnose autoimmune pathologiesis are based on two main parameters: fluorescence intensity and fluorescence staining pattern. Fluorescence intensity analysis is very often complex, and depending on the capabilities of the operator, the association with incorrect classes is statistically easy. The type and the localization of fluorescence define the IIF pattern, indeed the pattern depends on the distribution of the antigen inside the nucleus [17]: antibodies against antigens in different location give different patterns of fluorescence and therefore allow the identification of the different diseases.

Positivity to the ANA test, performed by analyzing patterns and fluorescence intensity [13], is associated with multiple autoimmune diseases. Some laboratories report the fluorescence intensity as numerical (0, 1, 2, 3, 4) or symbols (-, +, ++, +++, ++++) representing a semi-quantitative approximation of the fluorescence intensity. In the clinical practice, IIF samples are categorized into these aforementioned levels based on the visual assessment of their fluorescent intensity compared to a set of negative and positive controls.

For the fluorescence intensity the "titer" must be taken into consideration. Titer testing employs serial dilution to evaluate positivity or negativity of the sera. The titer corresponds to the highest dilution factor that still yields a positive reading [18]. For economic and practical reasons, many laboratories dilute only up to a given dilution. In general, the fluorescence intensity is assessed by patient serum dilution from 1/40 to 1/320, or fixed dilution 1/80.

In summary, the IIF technique is performed on HEp2 cells, using sera from patients referred to immunology laboratories for the detection of autoantibodies. After incubation of serum dilution, generally at 1/80, the antibodies linked to human immunoglobulin are detected by fluorescent antibodies. The negativity or positivity the serum is established along with the fluorescence pattern that is indicative of autoantibodies specificity. Therefore, IIF diagnosis requires estimating fluorescent intensity and pattern description of the sample at the fluorescence microscope.

1.2 ANA pattern classification tree

Antinuclear antibodies are significant biomarkers in the diagnosis of autoimmune diseases which is done by mean of IIF test with HEp-2 cells. The evaluation of ANA consists in the analysis of the fluorescence intensity and the staining patterns. The International Consensum on ANA Patterns (ICAP) discusses and promotes consensus regarding the richness in nuances of morphological patterns observed in the indirect immunofluorescence assay on HEp-2 cells [19]. In the official web site (www.anapatterns.org) representative images of ANA patterns and accurate description are displayed, but also the nomenclature and the classification tree. Figure 1.1 shows the nomenclature and classification tree for Hep-2 patterns.

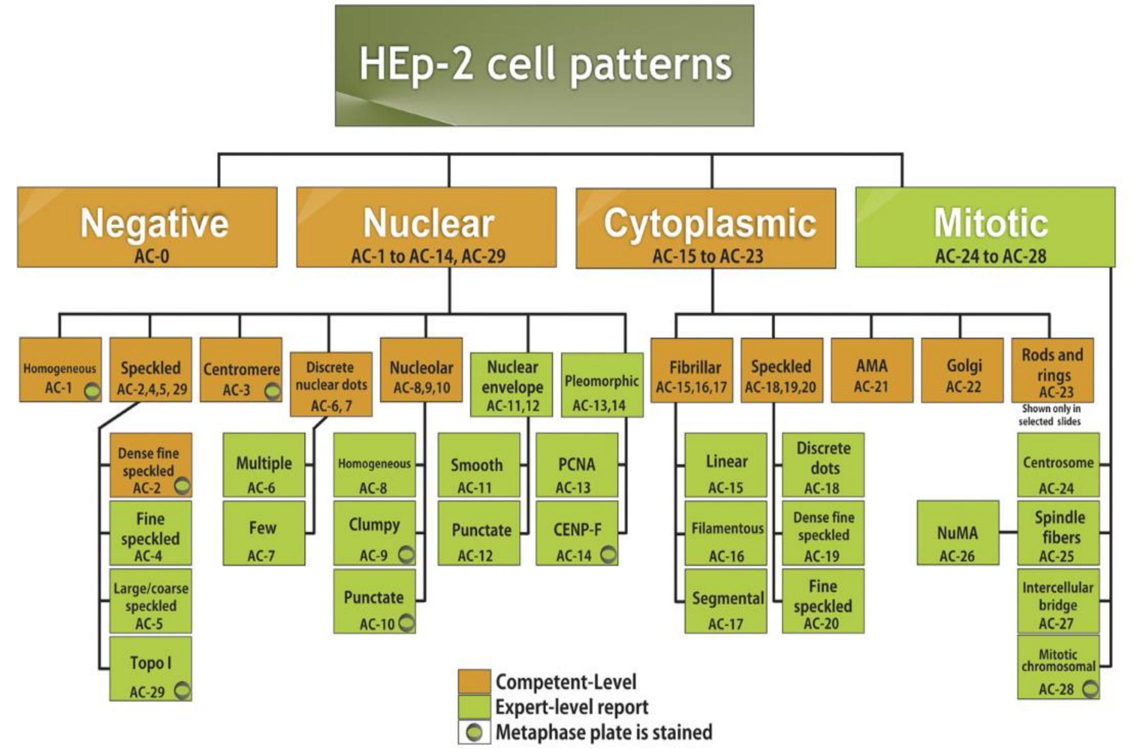
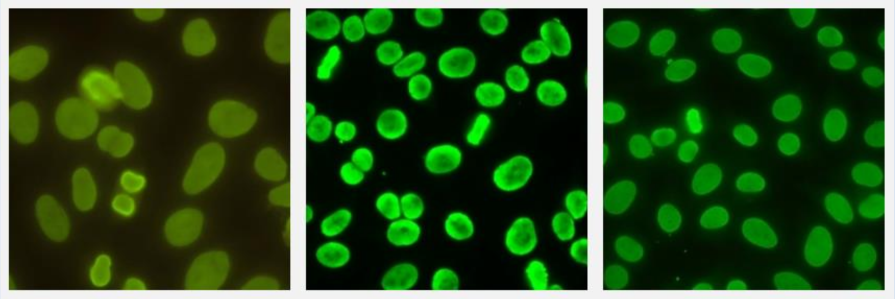
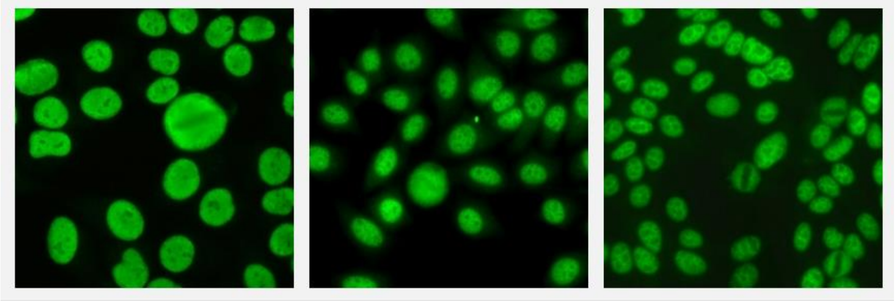
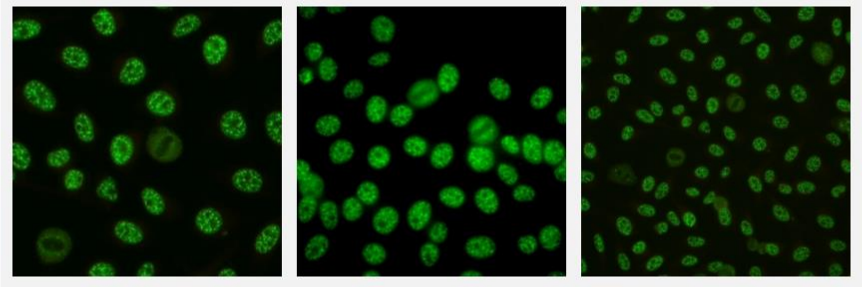
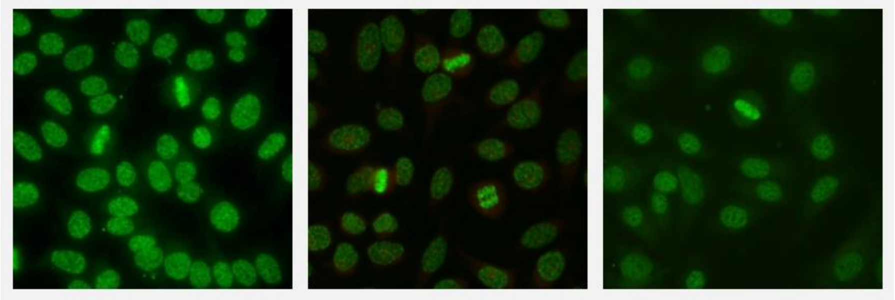
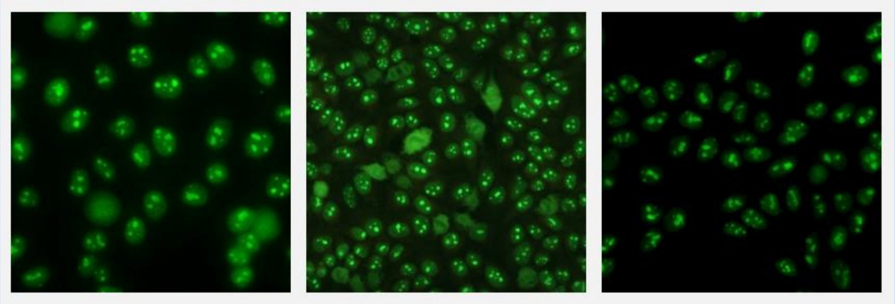


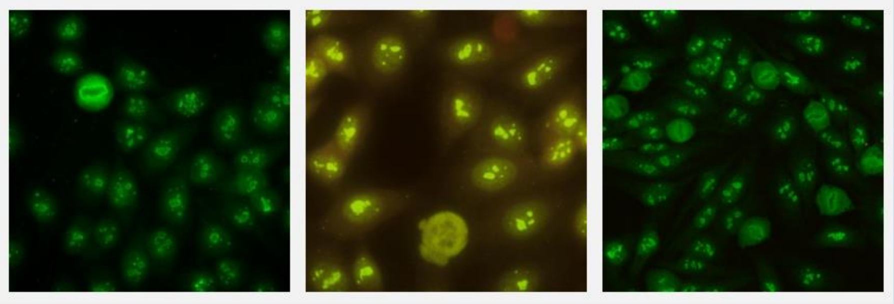
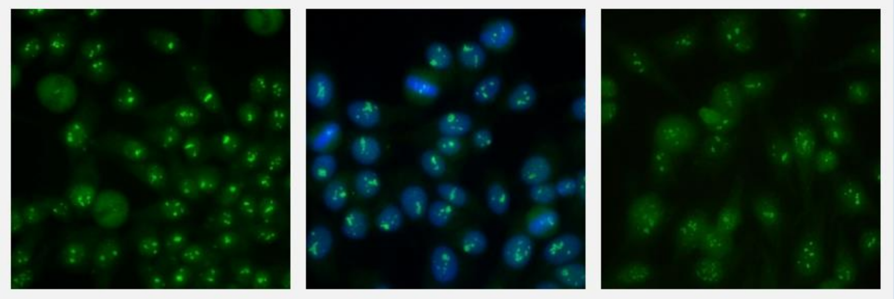
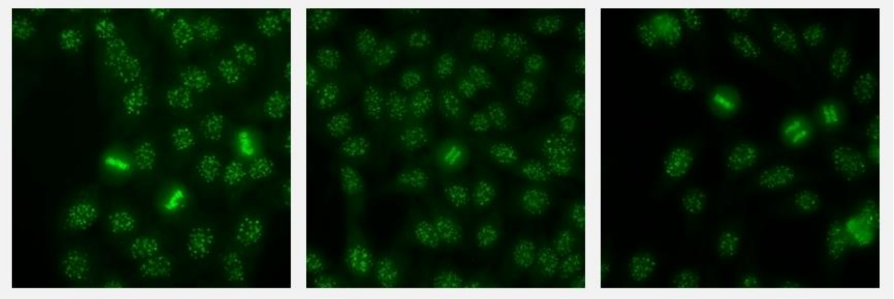
Figure 1.1: The nomenclature and classification tree for all HEp-2 cell [19].

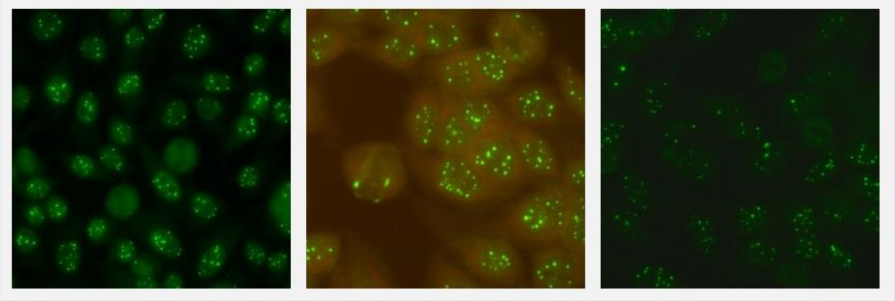
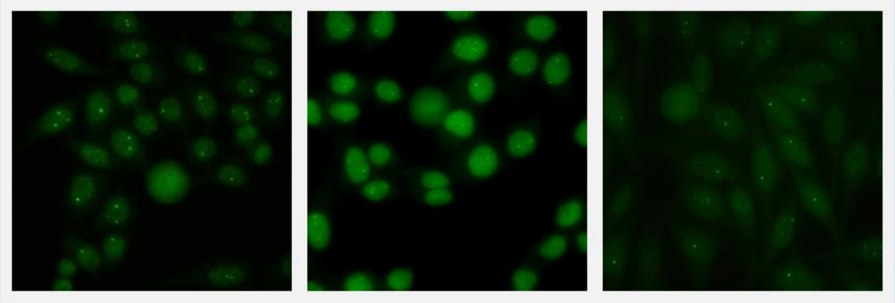
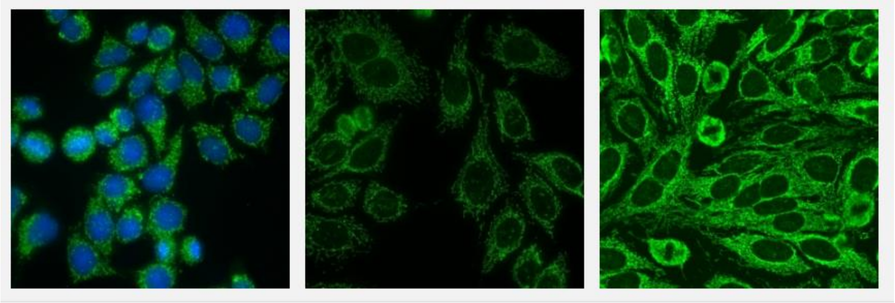
All ICAP patterns (designated with alphanumeric AC codes from AC-1 to AC-28), about thirty, are shown in the classification tree. Boxes with amber background are recommended as “competent-level” reporting, whereas those with olive green background are considered for “expert-level” reporting. Competent-level patterns are those that should be readily recognized versus patterns that would be more challenging and distinguishable only when observers or technologists have attained the expert-level [19][20]. For each pattern it is possible to view the AC (Anti Cell) code, the name of the pattern, other common names in use (synonym), a general description, additional information on antigen association, disease association and two small IIF icons of full representative images [20].

The first branching of the tree considers the negative and positive patterns, the latter are represented by nuclear, cytoplasmic and mitotic patterns. A pattern is considered negative if it is characterized by the absence of a clear-cut staining in any given subcellular structure. This definition is both subjective and semi-quantitative at best [21]. In the literature, the more relevant staining patterns are classified into one of the following groups: Homogeneous, Speckled, Nucleolar, Centromere, Discrete nuclear dots and Cytoplasmic. Table 1.1 shows a description and representative images from www.anapatterns.org for each of these main patterns.

Patterns	Description and representative images
Homogeneous	Homogeneous and regular fluorescence across all nucleoplasm. The nucleoli may be stained or not stained depending on cell substrate. Mitotic cells (metaphase, anaphase, and telophase) have the chromatin mass intensely stained in a homogeneous hyaline fashion. Clinical relevance: SLE, chronic autoimmune hepatitis, juvenile idiopathic arthritis. 
Speckled	It is divided into fine speckled, coarse speckled, dense fine speckled and topo I. The first three, generally more relevant, are shown below. **Fine speckled:** Fine tiny speckles across all nucleoplasm. The nucleoli may be stained or not stained. Mitotic cells (metaphase, anaphase, and telophase) have the chromatin mass not stained. Clinical relevance: present to a varying degree in distinct SARD. In particular, SjS, SLE, subacute cutaneous lupus erythematosus, neonatal lupus erythematosus, congenital heart block, DM, SSc, and SSc-AIM overlap syndrome [22].  **Coarse speckled:** Coarse speckles across all nucleoplasm. The nucleoli may be stained or not stained. Mitotic cells (metaphase, anaphase, and telophase) have the chromatin mass not stained. Clinical relevance: present to a varying degree in distinct SARD. In particular SLE, SSc, MCTD, SSc-AIM overlap syndrome, and UCTD [23].

	 **Dense fine speckled:** Speckled pattern distributed throughout the interphase nucleus with characteristic heterogeneity in the size, brightness and distribution of the speckles. Throughout the interphase nucleus, there are some denser and looser areas of speckles (very characteristic feature). The metaphase plate depicts strong speckled pattern with some coarse speckles standing out. Clinical relevance: negative association with SARD. 
Nucleolar	It is divided into three group: nucleolar homogeneous, nucleolar speckled e nucleolar clumpy. **Nucleolar homogeneous:** Diffuse fluorescence of the entire nucleolus, while the metaphase plate shows no staining. Clinical relevance: found in patients with SSc, SSc-AIM overlap syndrome, and patients with clinical manifestations of other SARD.  **Nucleolar clumpy:** Irregular staining of the nucleoli and Cajal bodies with a peri-chromosomal staining at the metaphase plates. Clinical relevance: found in patients with SSc.

	 **Nucleolar speckled:** Densely distributed but distinct grains seen in the nucleoli of interphase cells. In metaphase cells, up to 5 bright pairs of the nucleolar organizer regions can be seen within the chromatin body. The cytoplasm of mitotic cells may be slightly positive. Clinical relevance: SSc, Raynaud's phenomenon, SjS, cancer [24]. 
Centromere	Discrete coarse speckles (40-80/cell) scattered in interphase cells and aligned at the chromatin mass on mitotic cells. Clinical relevance: found in patients with limited cutaneous SSc and in a subset of patients with SjS. 
Discrete nuclear dots	It is divided into multiple nuclear dots and few nuclear dots. **Multiple nuclear dots:** Countable discrete nuclear speckles (6 to 20 nuclear dots/cell). Clinical relevance: found in a broad spectrum of autoimmune diseases, including PBC, AIM (DM).

	 **Few nuclear dots:** Countable discrete speckles (1 to 6 nuclear dots/cell in most cells). These are known as Cajal bodies or coiled bodies. Clinical relevance: rarely in SLE, (localized linear) SSc, and SjS. 
Cytoplasmic	It is divided into five categories: fibrillary, speckled, AMA, golgi, road & rings. Only the AMA and the golgi patterns are shown below to represent cytoplasmic, the reader is referred to the www.anapatterns.org site for the complete list. **AMA:** Coarse granular filamentous staining extending throughout the cytoplasm. Clinical relevance: found in PBC, but also detected in SSc, including PBC-SSc overlap syndrome and PBC-SjS overlap syndrome.  **Golgi:** Discontinuous speckled or granular perinuclear ribbon-like staining with polar distribution in the cytoplasm. Clinical relevance: found in small numbers of patients with a variety of conditions, including SjS, SLE, RA, MCTD, GPA, idiopathic cerebellar ataxia, paraneoplastic cerebellar degeneration, adult Still's disease, and viral infections including HIV and EBV.

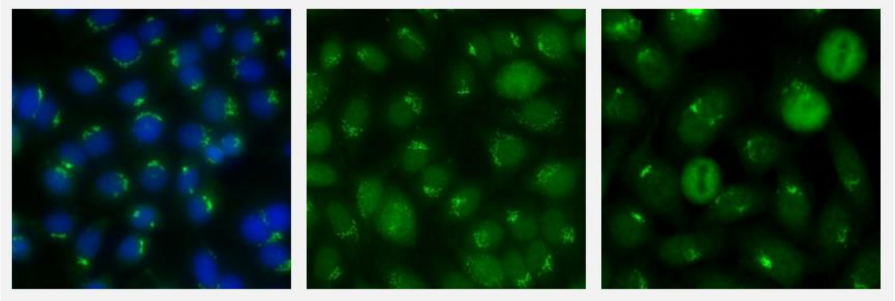
	

Table 1.1: The following Table shows a description and representative images from www.anapatterns.org forHomogeneous, Speckled, Nucleolar, Centromere, Discrete nuclear dots and Cytoplasmic patterns.

1.3 Motivation of a CAD system applied to AD

Diagnosing autoimmune diseases can be particularly difficult because these disorders can affect any organ or tissue in the body, and produce highly diverse clinical manifestations, depending on the site of autoimmune attack. Moreover, disease symptoms are often not manifest until the disease has reached a quite advanced stage. Laboratory techniques, able to point out and confirm diagnosis of autoimmune pathologies, are based on research and identification of autoantibodies, revealed by the presence of specific antigen - antibody complexes. In particular, identification of ANA through IIF method is an important part of clinical medicine and clinical immunology. Although the IIF techniques have progressively increased since they were firstly used in 1957 to demonstrate antinuclear antibodies, there are still various disadvantages in these techniques, among them the lack of an automatic procedure which could make easier, faster and more reliable the tests execution and hence lower the costs. The fluorescence pattern observed with the microscope (example: homogeneous, speckled, nucleolar, nentromere, etc.) is specific according to the nature of the self antigen and of its location in the cell.

Following the recent statement made by the American College of Rheumatology that the IIF technique should be considered as the standard screening method for the detection of ANA, the biomedical industry has proposed technological solutions which significantly improve the automation of the procedure, mostly in the preparation of substrates and slides with robotic devices performing dilution, dispensation and washing operations. Furthermore, much has been done for the digitalisation of microscope fluoroscopic images. These systems are based on the use of automated microscopes, robotised slide trays, high-sensitivity video cameras, and software dedicated to the acquisition and analysis of digital images [25].

Now what attracts most interest is the classification of patterns using standardised approaches: automated positive/negative screening and pattern interpretation. Such slides are typically examined by pathologists, however, due to the difficulty of the task, a CAD system is desirable [3][25]. It is in fact known that the interpretation of the IIF test is strongly influenced by reader's experience, by the quality of reagents used for testing (characteristics of the cell substrate or fluorochrome-labeled antihumanimmunoglobulins used) and by other local factors.

In general, CAD systems have been widely proposed in different areas of medicine and with different objectives, such as second-reading, improving the speed of the diagnostic processes, training physicians for special tasks, etc.

The first objective of an expert system applied to the diagnosys of autoimmune diseases is to improve diagnostic accuracy, proposing itself as a "second reader". CAD systems may use several automatic classification techniques to recognize the type of IIF patterns contained into an image. A properly trained CAD system in the field of indirect immunoflourescence can decrease the variability of the methodology, increasing the level of standardization and helping the specialist. The design of a CAD to support the diagnosis of autoimmune diseases must conceptually follow the guidelines recommended by experts in the sector, in particular the two main requirements concern the identification of the fluorescence intensity and if this is positive, the identification of the staining patterns. The main reasons for the development of a CAD system applied to the diagnosis of autoimmune disease are derived from multiple considerations. These considerations are listed below:

- the readings in IIF tests are subjected to interobserver variability that limits the reproducibility of the method and induces subjectivity to the results [7];
- the vast amount of image data that is generated by some imaging devices makes the detection of potential disease a burdensome task and may cause oversight errors [8];
- the lack of resources and adequately trained highly specialized personnel that are not always available [26];
- the low level of standardization intensifies the limitations of human ability to detect and diagnose a disease during image interpretation due to their non systematic search patterns, to the presence of noise and technical issues such as the photobleaching effect, which bleaches significantly the tissues in a few seconds [27][28];
- the fluorescence intensity and the fluorescence pattern analysis are particularly difficult, due to the similarity between different classes, and in any case is linked to the operator’s experience [16].

Besides, inter-laboratories and intra-laboratory discrepancies in terms of negative or positive aspects and in terms of pattern recognition depends on:

- the variability inherent to the used kit of HEp-2 cells;
- the reading system: microscope sensitivity, and the kind of the used lamp (HBO or LED);
- the threshold of positivity and the pattern identification, in fact usually several patterns overlap. Therefore, the skill of the observer is a key point.

The main disadvantage of IIF technique is its subjectivity in the interpretation of results, highly depending on the experience of the operator. For that reason, two Senior

Immunologists (double reading) with strong experience in fluorescent image interpretation are quite often needed. However, this condition is not respected in all immunology laboratories involved in diagnosis.

As already mentioned, the definition of fluorescence positivity and negativity is both subjective and semi-quantitative at best [21]. There should be a discussion regarding how ANA-positive vs ANA-negative cut-off is determined. There is general consensus that such cut-offs should be determined experimentally and locally with normal population controls [19]. The cut-off is highly dependent on the HEp-2 substrates used by individual laboratories, including factors specific to HEp-2 slide manufacturers and lot-to-lot variations, fluorochrome conjugated secondary antibody reagents, microscope and camera settings, serum dilutions, and other variables. Each laboratory should have a working set of positive and negative human serum sample controls for this specific purpose to ensure that the microscope and camera settings, but also the preparation of substrates and slides, is appropriate. It is to be considered that the NC (Negative Control) and the PC (Positive Control) as well as being prepared, occupy at least two slide wells used in the microscope, that is to say that, these are a cost. Another point to be considered is the fluorescent conjugate. Considering the variability in the microscope settings worldwide it is unrealistic to assume that the ready-to-use conjugate provided with the HEp-2 slide kits will fit all customers. Moreover, HEp-2 cutoff should be determined within the local population using the specific reagents and microscope setting [19].

These considerations related to fluorescence intensity are also valid for the fluorescence pattern recognition. Many patterns have similarities that often lead to error. A typical example is the speckled pattern that may be reported as nuclear homogeneous. Furthermore, patterns can occur in multiple ways, making the recognition problem very complex. These observations clearly suggested developing an automatic analysis of HEp-2 patterns. The introduction of new modern systems is an economic and effective support for the diagnosis of autoimmune diseases [29].

The expected advantages of an automatic system are:

- to minimise intra- and inter-laboratory variability;
- to improve the performance of the positive/negative classification, that is the reduction in frequency of false negative and false positive results;
- to minimize the use of positive and negative controls;
- to enhance the correlation of staining patterns with corresponding autoantibody reactivity;
- higher throughput in the laboratory workflow [25].

Referring to the state of the art it is clear that the implementation and validation of an Expert System CAD for the computerized support for the recognition of fluorescent patterns on IIF images is not an easy problem but at the same time the great and growing interest can also be inferred in being able to have a support that exploits the most innovative goals of computer vision and artificial intelligence.

1.4 Public HEp-2 database

The need of the scientific community for a large database of IIF images reported out by medical experts is increasing. Its use could be related to various purposes: training of young immunologists, epidemiological studies, diagnosis, etc.

The first public database of HEp-2 images is the "MIVIA" database or “ICPR 2012” [30]. MIVIA database is used during the first ‘Contest on HEp-2 Cells Classification’ at ICPR 2012 [31]. This database is the outcome of a research project jointly conducted by the Mivia Lab of the University of Salerno and the University Campus Biomedico of Rome with the financial support of ‘Regione Campania’ within the project ‘Classification of Immunofluorescence Images for the Diagnosis of Autoimmune Diseases’. “MIVIA” is an annotated database of IIF images, acquired using slides of HEp-2 substrate at the fixed dilution of 1:80, as recommended by the guidelines. The composition of the database is described in Table 1.2 and in Figure 1.2 some representative images of the database are shown.

PATTERN	Number of images
Centromere	6
Homogeneous	5
Fine speckled	4
Coarse speckled	5
Nucleolar	4
Cytoplasmatic	4

Table 1.2: List of images of public database “MIVIA”.

IIF slides are examined at the fluorescence microscope and belong to six patterns: homogeneous, fine speckled, coarse speckled, nucleolar, cytoplasmatic, centromere.

In addition to the description of staining pattern, the estimation of fluorescence intensity in intermediate (positive) and positive is provided. There are no negative intensity images in this database. Specialists took HEp-2 images with an acquisition unit consisting of the fluorescence microscope (40-fold magnification) coupled with a 50W mercury vapor lamp and with a digital camera (SLIM system by Das srl). The camera has a CCD sensor with square pixel of side equal to 6.45 µm. The images have a resolution of 1388×1038 pixels, a colour depth of 24 bits and they are stored in bitmap format. Specialists manually segment and annotate each cell at a workstation monitor and report data on fluorescence intensity (only intermediate and positive), pattern (according to the six classes reported above) and mitosis phase. Firstly, a biomedical engineer segmented the cells by the use of a tablet. Subsequently, each image was reviewed and annotated by an immunologist.

For each image, the database provided a description file containing:

- image’s pattern and intensity;
- objects seed points;

- objects class: cell, mitotic cell, artifact (due to slides preparation process);
- objects pattern (if is a cell): homogeneous, fine speckled, coarse speckled, nucleolar, cytoplasmatic and centromere.

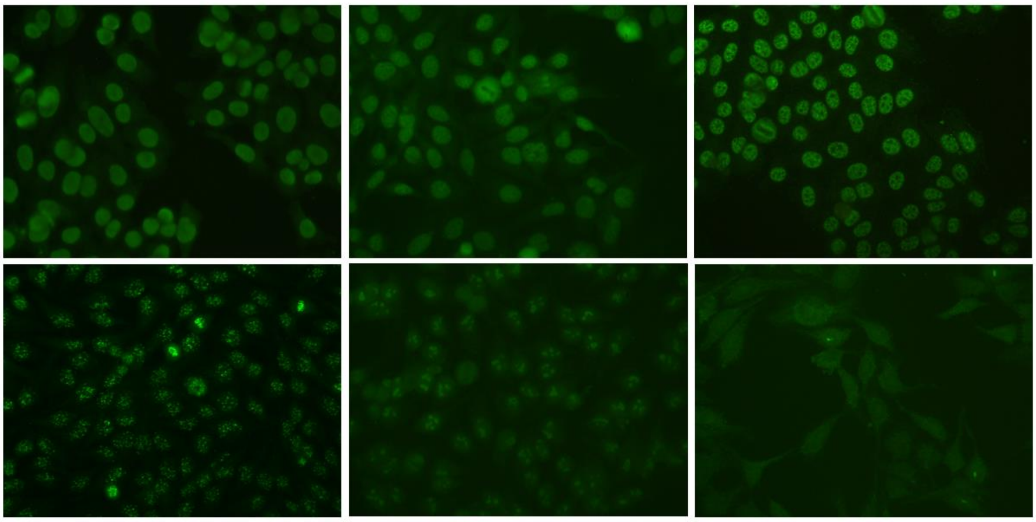


Figure 1.2: Representative images of the MIVIA database: left to right, top-down, homogeneous, fine speckled, coarse speckled, centromere, nucleolar, cytoplasmic.

The second public database is the “I3A” [32] provided for the participation to the ‘competition on cells classification by fluorescent image analysis‘ held at ICIP 2013 (International Conference on Image Processing) [33] and to the ‘Contest on Performance Evaluation on Indirect ImmunoFluorescence Image Analysis Systems‘ hosted by the 22th International Conference on Pattern Recognition (ICPR 2014) [34] and ICPR 2016 [35]. I3A database have two versions, “Task1” and “Task2”. Both are composed of a public part and a private part used exclusively to test the methods proposed for the competition. In the contest with Task1 the goal is the design and implementation of a IIF pattern recognition system able to classify the HEp-2 pre-segmented cells belonging to HEp-2 images in one of the following six pattern classes: homogeneous, speckled, nucleolar, centromere, Golgi and nuclear membrane. The total number of pre-segmented cells was 13596 extracted from 83 specimens. Table 1.3 shows the pattern distribution of the images provided for task-1.

	Homogeneous	**Speckled**	**Nucleolar**	**Centromere**	**Golgi**	**Nuclear Membrane**
Cells number	2494	2831	2598	2741	724	2208

Table 1.3: Staining patterns distribution in Task1 dataset.

In task2, the goal is the classification of the entire well (specimen) in seven patterns: homogeneous, speckled, nucleolar, centromere, Golgi, nuclear membrane and mitotic spindle. Task2 has 252 wells where four images for well are captured with a total of 1008 images.

The dataset task1 has been collected between 2011 and 2013 at Sullivan Nicolaides Pathology laboratory, Australia. It utilizes 419 patient positive sera, which were prepared on the 18-well slide of HEP-2000 IIF assay from Immuno Concepts N.A. Ltd. with screening 1:80 dilution. The specimens were then automatically photographed using a monochrome high dynamic range cooled microscopy camera which was fitted on a microscope with a plan-Apochromat 20x/0.8 objective lens and an LED illumination source. Approximately 100-200 cell images were extracted from each patient serum. The labelling process involved at least two scientists who read each patient specimen under a microscope. A third expert's opinion was sought to adjudicate any discrepancy between the two opinions. It used each specimen label for the ground-truth of cells extracted from it. Furthermore, all the labels were validated by using secondary tests such as ENA and anti-ds-DNA in order to confirm the presence and/absence of specific patterns. Each cell image contained in the database is annotated with the following information: cell pattern (one of the patterns above defined), cell intensity, cell mask, ID of the image which the cell belongs to. A characterization of the dataset is reported in Figure 1.3 and Figure 1.4.

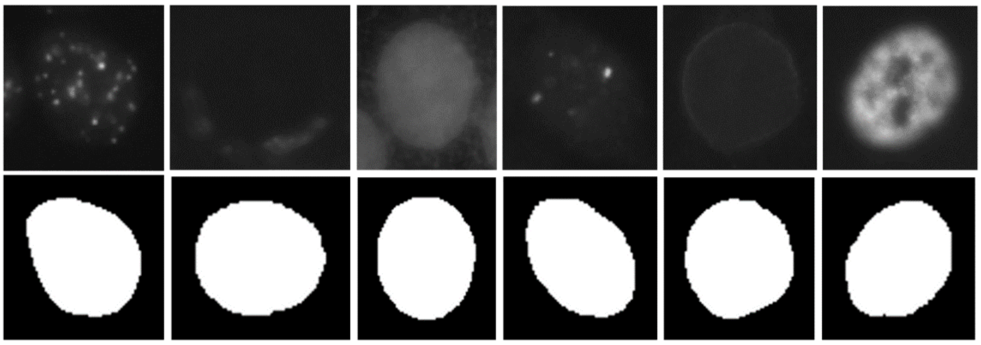


Figure 1.3: Representative cells images of the I3A Task1 database (left to right) centromere, golgi, homogeneous, nucleolar, nuclear membrane, speckled. In the second row the respective segmentation masks.

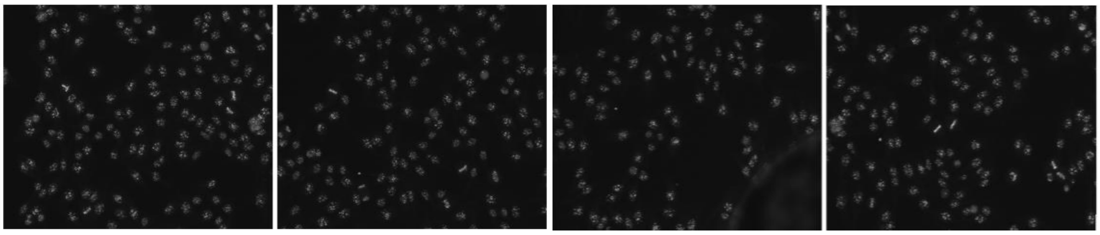


Figure 1.4: Four centromere images from a well from I3A task2 database.

The third public database is AIDA [36]. Started in November 2012, the AIDA project [37] (AutoImmunité, Diagnostic Assisté par ordinateur) is an international strategic project funded by the EU in the context of ENPI Italy-Tunisia cross-border cooperation in which one of the objectives was to collect a large database available to the scientific community. The project concerns particularly the application of ICTs for the diagnosis of autoimmune diseases, reading by computer the images of test IIF. Using a standard approach, seven immunology services have collected images of the IIF test on HEp-2 cells accompanied by the report of the senior immunologists.

The database acquired during the AIDA project consisted of two parts: one part was public, the other was private. The public AIDA database was a subset of the full AIDA database, where three physician experts (independently) have expressed a unanimous opinion when reporting. This is available to the scientific community and, it is the largest HEp-2 images public database that is most representative of real cases, including a variety of single and multiple patterns. The AIDA public database consists of 2080 images relating to 998 patients (261 males e 737 females); of these images 582 are negative while 1498 show a positive fluorescence intensity. The total number of wells are 1000: 530 positive wells and 470 negative wells. Table 1.4 summarizes the database numbers. The AIDA database is the public HEp-2 image database containing both images with positive fluorescence intensity and negative images; the other public databases are essentially composed of positive and weak positive fluorescence images, but not negative cases. Besides being “numerous”, the database is extremely varied, containing fluorescence positive sera with a variety of more than twenty staining patterns. In each image a single or multiple pattern can be present. The patterns terminology is in accordance with the “International Consensus on ANA Patterns” (ICAP) (http://www.anapatterns.org) [19] (see subsection 1.2 “ANA pattern classification tree”). Moreover, manufacturers of kits and instruments employed for the ANA testing were different site-to-site, as well as, different automated systems solutions for the processing of Indirect Immunofluorescence tests have been used: IF Sprinter Euroimmun, NOVA from INOVA diagnostic, and Helios from Aesku. HEp-2 images have been acquired, after incubation of the 1/80 serum dilution, by means of a unit consisting of a fluorescence microscope (40-fold magnification) coupled with a 50W mercury vapor lamp and a digital camera. The camera has a CCD sensor equipped with pixel size that equals 3.2 μm. The images have 24 bits color-depth and were stored in different common image file formats, as "jpg", "png", "bmp" and "tif". The public database can be downloaded, after registration, from the download section of the site (http://www.aidaproject.net/downloads). The dataset consists of 1000 wells records and for each of them are reported the following information: image/s from well were patient sera was prepared; the negativity or positivity fluorescence intensity of the patient sera; the staining pattern (single or multiple) for the patient positive sera; the age and sex of patient (if available); the image acquisition instrument. The private part of the AIDA database is structured in the same way as the public part, and it has about 20,000 positive and negative images. However, among all these images only about 3000 have triple concordance of reports. This part of the AIDA database is only accessible to the partners who participated in the project.

Positive fluorescence intensity wells	**Negative fluorescence intensity wells**	**Total wells**	**Total images**
530	470	1000	2080

Table 1.4: Wells distribution in AIDA dataset.

Some representative images of the AIDA database are shown in the Figure 1.5.

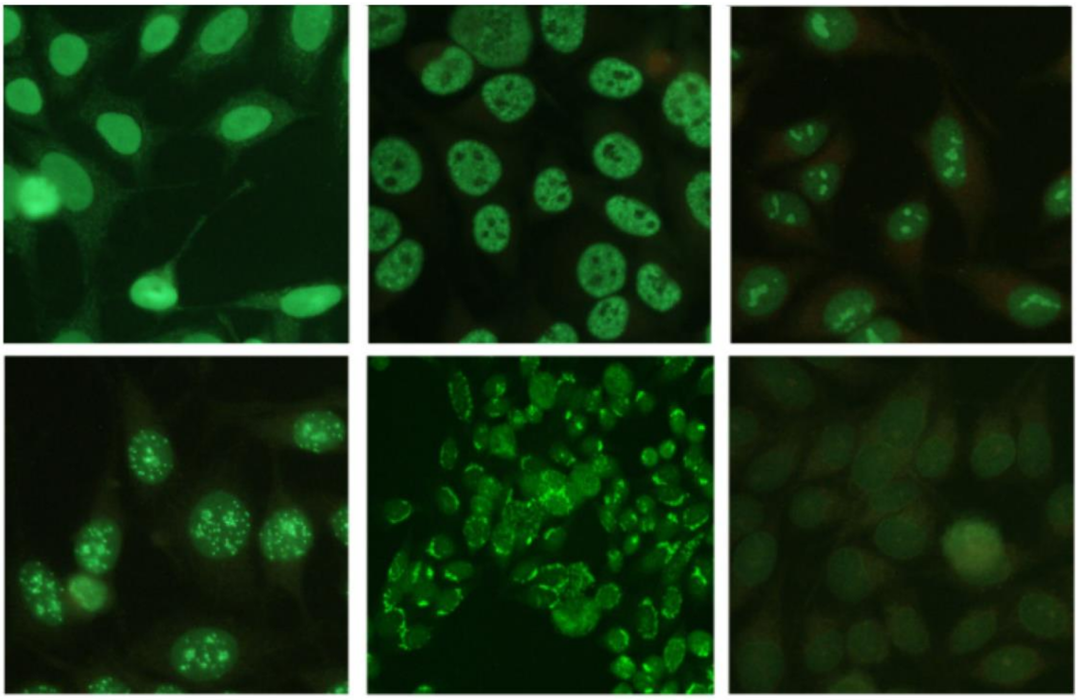


Figure 1.5: Examples of fluorescent patterns (from left to right and for up to down: homogeneous, speckled, nucleolar, centromere, golgi and nuclear membrane).

Table 1.5 summarizes the information characterizing the public databases described.

Database	Fluorescence Intensity	List of patterns	Pattern number	Cell number	Image number	Well number
MIVIA	Intermediate positive and positive	{homogeneous, coarse speckled, fine speckled, centromere, nucleolar, cytoplasmic}	6	1455	28	28
I3A task1	Intermediate positive and positive	{homogeneous, speckled, centromere, nucleolar, nuclear membrane, golgi}	6	13596	83	83
I3A task2	Intermediate positive and positive	{homogeneous, speckled, centromere, nucleolar, nuclear membrane, golgi, mitotic spindle}	7	-	1008	252
AIDA	Negative and positive	More than twenty single patterns and various multiple patterns	22	-	2080	1000

Table 1.5: Salient data from public databases.

Chapter 2 - The CAD system

The main objective of this thesis work is the implementation of a CAD that can be a valid diagnostic support for autoimmune diseases. This goal has been achieved by using the most innovative computer vision techniques and artificial intelligence algorithms.

In this Chapter is shows the distinctive peculiarity of a CAD system applied to the diagnosis of autoimmune disease, and the two fundamental pillars on which it is based: image processing and machine learning.

2.1 CAD system peculiarities

The design and implementation of an expert CAD system to support the diagnosis of autoimmune diseases must conceptually follow the guidelines recommended by experts in the sector, in particular the two main requirements concern the identification of the fluorescence intensity and if this is positive, the identification of the staining patterns.

In general, a CAD is a system that assists physicians in the interpretation of medical images. The main functionality of a CAD regards the automatic classification of the images. These systems have become increasingly widespread as a result of the growing development of digital imaging systems for medical imaging. Just think of digital mammography, X-ray tomography (CT), nuclear magnetic resonance imaging (MRI), and so on. Furthermore, the growing amount of data and medical images that the radiologist or other medical professional has to analyze and evaluate comprehensively in a short time, has certainly stimulated the development of these automatic CAD systems.

In general, CAD systems process digital images for typical appearances and to highlight conspicuous sections, such as possible diseases, in order to offer input to support a decision taken by the professional. The first CAD systems used flow-charts, statistical pattern-matching, probability theory or knowledge bases to drive their decision-making process [38]. Historically, the first CAD systems were developed in the field of mammography and in the detection of lung cancer in radiographic images of the chest. The importance of the development of CAD systems for the detection of breast and lung tumors is mainly due to two factors: breast and lung cancer are the cancer forms with the highest mortality rate; moreover, CAD systems can be a valid support for radiologists engaged in population screening programs. Subsequently, the CAD systems spread to the most varied areas of medical imaging, not least the digital pathology derived from the possibility of acquiring digital images from the microscope. Glass slides are converted into digital slides that can be analyzed with the CAD system.

The development of a CAD is an interdisciplinary activity that combines medical knowledge and innovative computer vision and artificial intelligence techniques. It should be emphasized that the CAD does not substitute the physician or other professional, but rather plays a supporting role. The professional is generally responsible for the final interpretation of a medical image. The role of the CAD is to help in the interpretation, highlighting possible suspicious areas, and above all, often takes the role of second reader.

In literature there is a distinction between Computer-aided detection (CADe) systems that are usually confined to marking conspicuous structures and sections and Computer-aided diagnosis (CADx) systems evaluate the conspicuous structures. As will be shown in this thesis, the CAD developed and applied to autoimmune diseases, is oriented to detection, in fact the segmentation and recognition of the patterns of Hep-2 cells is a detection activity.

In the field of CAD system development, a large number of techniques (more or less standard) can be used, however, it is possible to group them, based on the objective they set, in the following families:

- image acquisition: it is the process of acquiring images through sensors of any nature (for example: electron microscopes, digital mammograms, digital radiographic equipment, CT scanners, etc.);
- preprocessing: it is a preliminary treatment of the images and has the purpose of preparing the images for subsequent manipulations. Examples of preprocessing are: the reduction of the noise introduced during the acquisition process and dependent on its modalities, the elimination of artifacts such as the "plates" in a mammographic image, the improvement of the image quality and the image filtering;
- segmentation: it has the task of dividing the image into its constituent parts.It can be carried out at various levels in order to identify different structures in the image generally called regions of interest, such as cells, potentially pathological lesions, and so on according to the specific field;
- feature extraction: the techniques for the extraction of characteristic parameters, features, have the purpose of obtaining analytical quantities in order to characterize the visual information of the ROIs;
- classification: it is the procedure used for the treatment of the regions identified in the segmentation phase that will allow the association of a generic ROI to a class, for example "positive" and "negative".

The research proposed here is the implementation, training and optimization of a CAD that can be a valid diagnostic support for autoimmune diseases. Being the diagnosis of autoimmune diseases based on the recognition of fluoroscopic patterns on the HEp-2 images acquired under the microscope, it is clear that the realization of the aforesaid CAD must exploit the most innovative computer vision techniques and artificial intelligence algorithms.

The main role of computer vision is the “image preprocessing” through advanced digital image processing techniques in the space and frequency domain. The medical images preprocessing has the goal of reducing noice and artifacts, improving image quality for example through contrast normalization, and filter the images to emphasize certain characteristics. The segmentation phase, where the Regions of Interest are extracted, and feature extraction from these ROI are of great importance. Image segmentation is defined as a partitioning of an image into regions that are meaningful for a specific task. A segmented ROI may be characterized by different type of attributes that make up the feature space. These features must be sufficiently discriminative and appropriately chosen for the application since they fundamentally impact the resulting quality of the classifier [39].

Artificial intelligence algorithms have the main role in the last phase: once ROI are obtained, discrimination between various patterns will be addressed. The ROI evaluation and classification belong to the field of pattern recognition and all modern approaches are based on machine learning. In recent years, the world of pattern recognition has been overtaken by deep learning, thanks to which machine learning has achieved impressive results.

2.2 Medical images processing and segmentation

Medical image processing deals with the development of problem-specific approaches to the enhancement of raw medical image data for the purposes of selective visualization as well as further analysis. Automatic medical imaging analysis techniques have been in continuous development since their inception with the discovery of X-rays over a hundred years ago. Through bioimaging, the physician extends the functionality of his sense organs by being able to explore the anatomy and function of internal organs, in search of the initial signs of disease. Images are not self-explanatory. Their interpretation requires professional skill that has to grow with the number of different imaging techniques [40]. However, both objective and subjective factors heavily condition the correct interpretation of biomedical images and consequently reduce their diagnostic accuracy.

There are many advantages of digital medical imaging. Unlike analog images, which can only be examined on the support and in the format in which they are produced, the digital images, being in electronic form, can be viewed on various devices, stored on electronic storage media and easily transferred also via the network. The type of management is certainly simpler, safer and less expensive: a digital image can be sent remotely both inside and outside the laboratory or in general in the hospital, even to several health professionals at the same time, without being physically moved from the electronic archive that contains it. But the most important characteristic of digital images is, without doubt, the possibility of modifying their characteristics, according to the needs, by means of processing algorithms.

These algorithms, developed to enhance certain components of an image, can significantly improve the visualization capabilities of the human eye and allow the identification of details that might otherwise be lost. Image processing is generally referred to as the discipline in which both the input and the output of the processing is an image, main purpose is to reduce noise, increase contrast, etc. The image analysis takes a further step in the complexity of the processing, as it concerns the extraction of characteristics and their analysis. Typical operations are the division of an image into regions or objects (segmentation) and the subsequent description for further processing.

The preprocessing methods can be varied, and it is important to point out that in a CAD system it is not said that a single preprocessing phase is sufficient, for example segmentation can take advantage of a preprocessing methodology different from that used to extract the features from ROI. In this research it will see the use of different preprocessing techniques will be applied to the segmentation phase, binary classification and multi pattern classification.

If preprocessing has the goal of reducing noice and artifacts, improving image quality and filtering the images to emphasize certain characteristics, segmentation is the process of dividing an image into distinct regions and in particular to find the ROIs in which the features will be extracted. The segmentation process is a fundamental step in the area of image analysis and pattern recognition, the purpose of which is to break down an image into parts that are significant with respect to a particular application.

Segmentation is a critical phase in the analysis of the image: the precision and quality of the result of the segmentation can very heavily influence the subsequent phases. Many segmentation techniques are known, as regards both grayscale images and as regards color images. These techniques differ in defining the homogeneity criterion between regions and in the algorithm used to build these regions. The approaches for segmenting monochromatic images are based on measures of discontinuity or homogeneity in the gray levels of the image.

Methods based on discontinuity detection partition the image by detecting isolated points, edges, lines and contours, while homogeneity-based approaches include histogram thresholding, clustering, region splitting and merging and region growing. The latter can be further divided into grouping techniques with respect only to the characteristics of the pixels (which only take into account the values of the individual pixels) and techniques based on the detection of regions (and therefore combining information on the values of the pixels with information space).

In this work, the segmentation was addressed with different techniques and considering both methods proposed in the literature and their appropriate variants, and ad-hoc methods. The evaluation of this objective has been carried out through measures of merit such as the "Dice index" and the "Jaccard index" aimed at evaluating the segmentation process both in terms of foreground and background, by overlapping with masks of segmentation produced by experienced staff consider the ground truth.

After the ROI segmentation, the characteristics are extracted on which the classification is based. The feature extraction procedure provides for the representation of the object under study by means of a numerical vector whose components are derived from the object itself and as it will see later, the classification is strictly linked to the choice of these measures.

2.3 Machine learning classifiers

Classifier systems are divided into binary or multi-class and are generally made with *machine learning* techniques. For the problem of discrimination between positive and negative fluorescence intensity, the most appropriate classifier is obviously the binary one, while for the classification of the patterns it is necessary to use multiclassification.

The classification scheme is usually based on the availability of a set of examples already classified. This set of patterns is called a *training set* and the consequent learning strategy is called *supervised*. Learning can also be *unsupervised*, in the sense that to the system is not provided a priori patterns whose correct classification is known and therefore the system establishes classes basing on the statistical regularity of the patterns. Generally, where examples with relative labels can be used, it is advisable to use supervised systems since they achieve performances that are clearly superior to unsupervised systems.

In supervised training mode, classifiers are trained using a set of examples already classified by experts. All supervised learning algorithms start from the assumption that, if a set of *representative* and *complete* examples of the problem to be classified is provided, the resulting classifier will be able to distinguish the new specimens correctly. To train the classifiers, it is necessary to extract appropriate features from the image dataset, therefore the choice of features is a fundamental requirement as they must possess the information necessary to allow discrimination between the various patterns to be recognized.

Among the most used classifiers are the *Support Vector Machines* (SVM), the *Artificial Neural Networks* (ANN) and in particular its modern descendant, namely the *Convolutional Neural Network* (CNN). These classifiers are detailed in Appendix A.

Even if compared to *Linear Discrimination Analysis* (LDA), support vector machines and Neural Networks are more complex to implement, however, they offer a greater degree of precision in classification. A linear classifier makes a classification decision based on the value of a linear combination of the extracted features. The characteristics are presented to the classifier system through an n-dimensional vector, called *feature vector*. For a binary classification problem, the operation of the classifier can be represented by dividing an n-dimensional space with a hyperplane. In this space, the feature vectors of the various elements to be discriminated are projected in the form of points. Given a hyperplane, the points on one side are classified belonging to a category, the other points that lie on the other side of the hyperplane are associated with the other category. Training consists in optimizing the position of the hyperplane so that the dispersion in the class is reduced to a minimum, while the dispersion between classes is maximized. In other words, the classifier look for the surface that better separates the two classes to be discriminated. This implies that only a few examples near the hyperplane are important for learning, the so-called support vectors. The others can be ignored. The substantial difference between the methods indicated lies in the fact that the linear discriminants use a linear hyperplane in the space where the characteristics are projected, while the SVM and the Neural Networks allow a more flexible and non-linear classification surface. Figure 2.1 shows a graphic example of a linear and non-linear separation hypersurface.

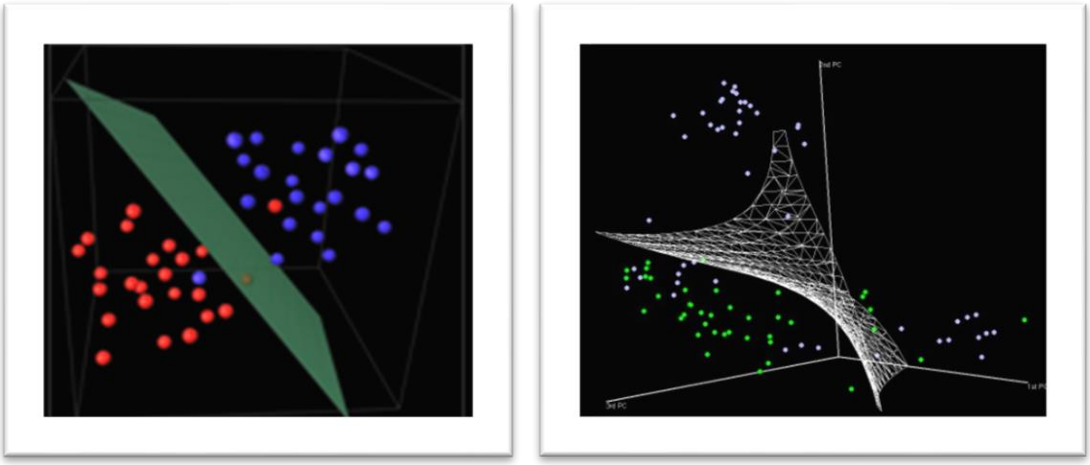


Figure 2.1: Example of binary classification using linear (left) and non-linear (right) hypersurface.

When using methods that are capable of developing a non-linear classification surface, even if the results are usually better than methods with linear hyperplanes, in addition to the greater computational complexity, so-called *overfitting* must be taken into account. This phenomenon occurs when an over-adaptation of the classifier separation surface with respect to the data used for training is achieved. Especially in cases where learning has been carried out too long or where there is a small number of examples of training, the model could adapt to characteristics that are specific only to the training set, but which are not reflected in the rest of the cases. In this circumstance, even a wrong and absurd model exactly classifies the training set, but it will not generalize correctly on the new specimens. To avoid overfitting, therefore, the law of *Occam's razor* and technical details such as *cross-validation* and *early arrest* must be kept in mind.

In recent years, the methods of deep learning have overcome the other machine learning techniques in their effectiveness and robustness, and now they prevail in artificial intelligence studies. In this context, CNNs have played a significant role especially in the biomedical field. Several CNN have been proposed in various fields of application obtaining new state-of-the-art performances. The main reason behind this success is that effective task-dependent image features can be directly or intrinsically learned through the hierarchy of convolutional kernels inside CNN [41].

While the traditional classification was based on the chain "ROIs segmentation → feature extraction → classification", the use of CNN allows to overcome the problem of searching for performing features by shortening the chain in "ROI segmentation → CNN classification". With CNN, the concept of finding the most suitable features for the problem being analyzed is skipped, the search for the so-called hand-crafted features of the traditional classification pipeline is replaced by CNN's architecture research.

Given millions of parameters to fit during model training (much more than previous traditional pipelines), CNN representation empowers and enables computerized image recognition models, with a good possibility to be able to handle more challenging imaging problems [41]. There are currently three major techniques that successfully employ CNNs to medical image classification: training the CNN from scratch, using off-the-shelf pretrained CNN features, and transfer learning, i.e., fine-tuning CNN models pretrained from natural image dataset (such as large-scale annotated natural image database: ImageNet) to medical image tasks [42].

In this work for the classification of Hep-2 images, both the traditional classification pipeline and the recent CNN-based classification chain will be compared.

2.4 CAD system evaluation

The goodness of the CAD produced was assessed using quantitative numerical measures of merit such as sensitivity and specificity, AUC (Area Under the ROC Curve), Accuracy, etc. In addition, the public databases available to the scientific community were used and allowed comparison with other publications.

If a HEp-2 image with positive fluorescence intensity is classified as positive, a true positive (TP) occurs, while if it is classified as negative, a false negative (FN) occurs. On the contrary, if a Hep-2 image with negative fluorescence intensity is classified as such, then it is a true negative (TN), while if it is classified as positive there is a false positive (FP). The total number of these four categories must be 100% of the dataset. It is possible to represent this type of situation using a 2x2 contingency table (see Table 2.1), where the columns represent the distinction between positive and negative images; the rows instead represent the result of the classifier on the examples.

		Reference value		
		Positive	**Negative**	
Predicted value	**Positive**	**True positive**	**False positive**	**Total positive**
	Negative	**False negative**	**True negative**	**Total negative**
		Total positive	**Total negative**	

Table 2.1: Contingency table for a two class prediction problem.

Starting from these elements, *specificity* or true negative rate (TNR) can be defined as the percentage of negatives found by the classifier, exactly TN among all the classified negatives (TN + FP). *Sensitivity* or true positive rate (TPR), on the other hand, is the percentage of positive images (TP) among all images classified as real (TP + FN).

The two cases incorrectly classified are false negative and false positive, the most serious one is the false negative, because it causes a delay in the diagnosis and treatment of the disease that could irreparably compromise the patient's health.

The evaluation of the performance of a diagnostic system is generally expressed by the aforementioned pair of indices: sensitivity and specificity.

$$Sensitivity = TP / (TP + FN) \qquad (1)$$

$$Specificity = TN / (TN + FP) \qquad (2)$$

Another commonly used measure is the Accuracy (ACC) and the Balanced Accuracy (BAC) defined in the following equations:

$$Accuracy = ACC = (TP + TN) / (TP + TN + FP + FN) \qquad (3)$$

$$Balanced\ Accuracy = BAC = (TPR + TNR) / 2 \qquad (4)$$

The BAC is an index that is used in place of the ACC especially in the presence of imbalanced data. In fact, considering an example with an extremely unbalanced dataset, it can be seen that a wrong classifier who always classifies the class with the highest number of presences obtains a high ACC. The BAC metric does not suffer from this problem by averaging the correct identification results of both classes.

The classifier performance can be studied and displayed graphically using the ROC (Receiver Operating Characteristic) curve. The coordinates of each point of the curve are, respectively, the true positive rate (sensitivity) and the false positive rate (1-specificity), in correspondence with a certain cut-off (threshold value on which the class is decided).

The main problem that generates uncertainty in the interpretation of a test lies in the fact that, in the vast majority of cases, there is an area of overlap between the distributions of the results of the test applied in positive and negative populations respectively. In fact, if the two populations returned separate values then it would be easy to identify the cut off value on the abscissa axis capable of discriminating with absolute precision the two populations. Unfortunately, however, in practice there is always a wider overlap of the two distributions and it is therefore impossible to identify on the abscissa axis a cut off value that allows a perfect classification, that is such as to reset both the false positives than false negatives.

Referring to Figure 2.2, when the chosen threshold value changes, different values will be obtained for sensitivity and for specificity. The pairs of values false positive rate (FPR), true positive rate (TPR), obtained at the various thresholds (XC), give rise to a ROC curve [43].

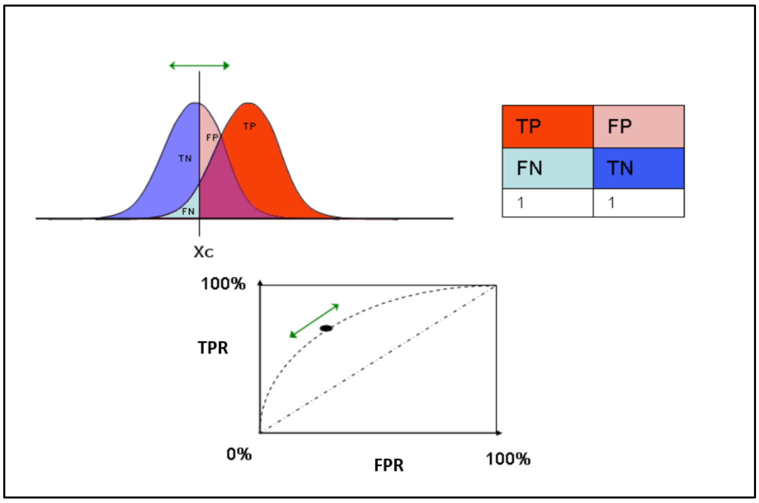


Figure 2.2: Explanatory model of the construction of the ROC curve; as the threshold Xc changes, the TNR and FPR indices vary.

A ROC curve of a binary classifier is usually between two limit curves: the first cuts the graph at 45° passing through the origin, the other is represented by the set of segments that rises from the origin to the point (0,1) and from that which joins point (0,1) to (1,1). The first line represents the case of the random classifier with an underlying area equal to 0.5. The second curve has an underlying area equal to 1, which is the perfect classifier. In consideration of this, another measure normally used, to describe the performance of a system, is the area below the ROC curve, generally indicated by Az [44].

If the above measures are the most recognized for evaluating a binary classifier, to evaluate a multiclass classification a known measure is the Mean Class Accuracy.

Let CCR_k be the correct classification rate for class k determined as follows:

$$CCR_k = T_k / N_k \qquad (5)$$

where T_k is the number of correct identifications of class *k*, while N_k is the total number of elements of class *k*. The Mean Class Accuracy is determined by:

$$MCA = (1/k) \sum_k CCR_k \qquad (6)$$

Chapter 3 - Hep-2 image segmentation

This chapter exposes the segmentation of HEp-2 cells which is a fundamental step of the development of CAD applied to AD diagnosis. It describes the strategies taken into consideration and finally the results obtained.

3.1 Introduction and related work

The image segmentation is defined as a partitioning of an image into regions that are meaningful for a specific task. These regions are called regions of interest. In the case of Hep-2 images, the regions of interest that contain the salient information on which to carry out the recognition of the fluoroscopic patterns are obviously the cells and their cytoplasm.

Segmentation is a critical phase in the analysis of the image: the precision and quality of the result of the segmentation can very heavily influence the subsequent phases. Such task is one of the most challenging of automated IIF analysis, it is not an easy problem because the segmentation algorithm has to cope with a large heterogeneity of shapes and textures due to various types of patterns that can be found in HEp-2 images.

Some methods of image segmentation have been proposed in the literature, however due to the great variety with which cell patterns can be presented, there is consensus that the problem of HEp-2 image segmentation is not easy to solve [39]. For this reason, in this thesis, numerous techniques and their combinations have been evaluated, in a process aimed at maximizing figures of merit such as the Dice and Jaccard index. In order to address this optimization strategy, the automatic segmentation is compared with a segmentation mask produced by experienced staff and considered the ground truth.

There are not many publications of articles that address the segmentation of HEp-2 cells in IIF images, probably due to the large number of patterns and the great diversity in which they appear within the IIF images. One of the first significant advances in the field of IIF images was proposed by Perner et al. [45][46] who presented an early attempt on developing an automated HEp-2 cell classification system. With reference to segmentation, the cells are localized by applying a global threshold using the Otsu algorithm. The authors argue that with the Otsu algorithm it is possible to localize cells with their cytoplasmic structure, but not the nuclear membrane itself. To overcome these limitations, they use morphological filters and the overlapping cells have been eliminated by a simple size-based heuristic. In particular, in [46] before thresholding the equalization of the histogram is used to eliminate the influence of the different patterns.

Also in [47] no pre-filter is used and the authors segmented the cells using a histogram-based mixture model threshold algorithm, which models the background intensity, in conjunction with the watershed transform. The watershed is done on the binary image (background objects) and resulted in the segmentation of the round objects, the cells in the image. Defective cells are eliminated based on knowledge on the specific cell type, expressed as of holes, size of nucleus (8-15 um), core shape (round / oval).

In [48] a specialization of the segmentation is carried out through a pre-classification of the image. The pre-classification is make on the number of segmented objects with the Otsu method: based on a given threshold, the image is classified as "sparse region cells" or "mass region cells". In the first case, the segmentation process applies an Anisotropic diffusion filter on channel B of the HSB model, applies Canny edge detection and dilation morphological filters (3x3 diamond). In the second case the Canny method is replaced with the Otsu method and the morphological dilation filters are of size (17x17 diamond).

In [49] the authors propose two stage watershed segmentation. In the first stage, a median filter (3x3) is applied to the G channel of the RGB model to reduce noise and the contrast is increased. Subsequently, the watershed transform is used, and the number of regions is counted which allow to determine whether the image must follow the second segmentation stage or not. In the second stage, channel C of the CMY color model is used, and the anisotropic filter is used as pre-processing. Then Otsu thresholding is used and watershed on the binary image. The performance results of these methods were not satisfactory because they could not detect two staining patterns (centromere and cytoplasmic).

Also in [50] a segmentation based on the watershed transform is proposed. In particular, they use the Otsu method to extract the ROI in order to pre-classify and apply methods to reduce noise. Then they transform the image based on the gradient and implement the segmentation with watershed with parameters tailored to the pre-classification.

In [51] the authors propose a variant of the adaptive and iterative OTSU method as segmentation: they divide the image into 16 blocks and apply the OTSU thresholding. After selecting the ROIs, the Otsu method is applied to each ROI increased by two pixels at each iteration, the iteration ends when the new segmented pixels are less than 1% compared to the previous iteration.

In some works, to reduce its complexity, it has been decided to identify the autoantibody patterns that are simpler from a visual point of view and in a very limited number. Roy et al [52] addressed the problem of HEp-2 image segmentation using a rough-fuzzy clustering algorithm which, using spatial information, assigns labels to the pixels.

Cheng et al [53] proposed a system adopting foreground watershed, background watershed, and foreground marker-controlled watershed for the segmentation of cells.

Percanella et al [54] proposed a method which adopts image reconstruction for a preliminary image segmentation and, then, it employs a sort of classifier-controlled dilation for better determining the structure of the cells.

Tonti et al [55] presented an adaptive marker-controlled watershed approach. The main feature of this algorithm is an improved pipeline for the watershed marker selection, which takes advantage of domain-specific knowledge about the textural and geometrical characteristics.

3.2 The methods implemented

Some segmentation techniques are known in the literature, both as regards grayscale images and as regards color images. In this thesis the Hep-2 image segmentation is carried out on grayscale. In fact, despite the fact that the images of the AIDA and MIVIA datasets are in RGB color, the channel on which the information is concentrated is the green G since the fluorescence is concentrated on this length wave. Figure 3.1 shows the histograms of the three RGB components of a HEp-2 image. It is evident that the red and blue channels, with respect to the green channel, lose part of the information referring mostly to the cells as visible in the histogram of the green channel in the hill following the first peak. The first peak, on the other hand, is mostly due to the background. Furthermore, by analyzing the patches extracted in the background, a greater presence of noise in the red and blue channels is noted.

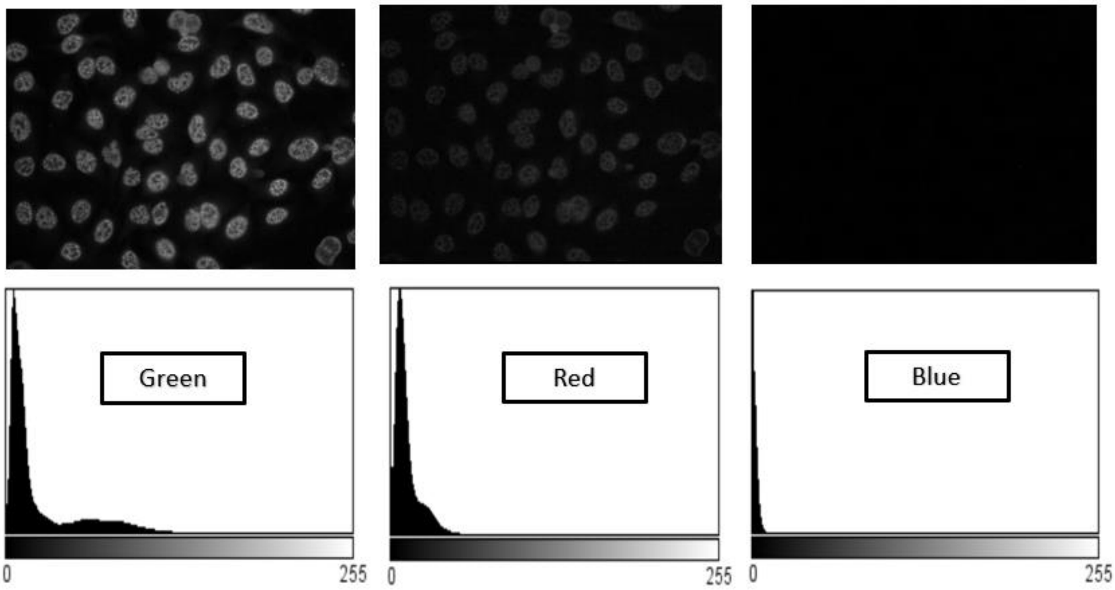


Figure 3.1: Comparison of RGB channel histograms of a color HEp-2 image.

Despite these considerations, the blue and red channels of the RGB model have been analized. The results obtained using the blue and red channels were significantly worse than the green channel. A conversion of the RGB channel into a gray channel was also experimented. In particular a weighted sum of the R, G and B components defined below was used (from "Recommendation ITU-R BT.601-7"):

$$Gray = 0.2989 * R + 0{,}5879 * G + 0{,}1140 * B \qquad (7)$$

This unbalanced conversion on the green channel has given results comparable to (but still not better than) the exclusive use of the green channel.

The approaches for segmenting monochrome images are generally based on measures of discontinuity or homogeneity in the gray levels of the image. The methods based on the detection of discontinuities apply preprocessing that allow to identify edges as for example the well-known Sobel filter does. Starting from the edges found, techniques are used to form the contours. Given the variety of patterns present on HEp-2 images, the application of discontinuity-based techniques does not produce performing results. The same can be said for other renowned techniques such as the watershed. As will be seen in the following, starting from a first coarse segmentation it is possible to use contour completion techniques in a refinement process.

The approach based on homogeneity measures sees the histogram threshold as one of the main techniques. This approach assumes that the image is composed of regions that differ from each other in the range of values of the pixels that compose them, so that the histogram of an image has peaks corresponding to these values. The values that correspond to valleys in the histogram are used as thresholds to distinguish the regions from each other.

In this thesis two main automated methods for the segmentation of HEp-2 IIF images were developed. The first method is based on a classic segmentation pipeline consisting of three main phases:

1 image pre-processing;
2 application of binarization method;
3 post-processing of the binary mask.

The second method also consists of three main phases:

1 image pre-segmentation (derived from the first method);
2 Hough trasform for ellipse;
3 active contour model.

This method uses an adaptive threshold segmentation process, derived from the first segmentation method, followed by the randomized Hough transform and a phase of active contours to achieve a robust and performing segmentation of cells. The first binarization step can be defined as a pre-segmentation to identify the cells, the last two steps refined the contour of the cells. This refinement allows to better follow the contours of the cells but also to avoid the problem of partial cell overlap that can occur. It should be emphasized that the pre-segmentation method is derived from the results obtained from the first segmentation method. Figure 3.2 shows the flowcharts of the two methods.

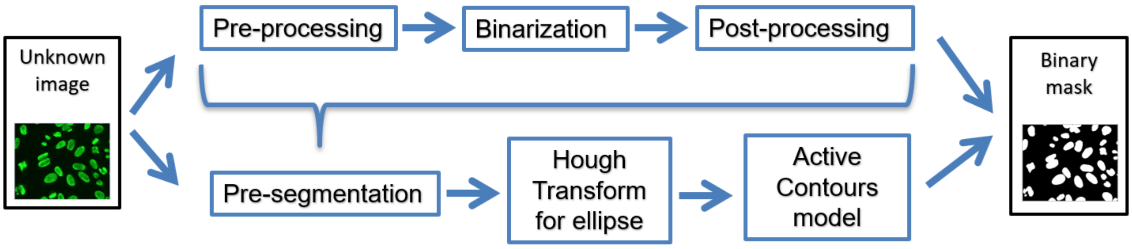


Figure 3.2: Flowchart of two segmentation methods implemented.

Both segmentation methods have as final output a binary mask, that allows the extraction of the ROIs on which the classification phase will be carried out. Since the ROIs will subsequently be used for the extraction of the features, it should be emphasized that these can be extracted with the binary mask from the HEp-2 image preprocessed in a different way than the pre-processing used for segmentation. Binary masks are also used for the evaluation of the segmentation process, both in terms of foreground and background, by overlapping with masks of segmentation produced by experienced staff considered the ground truth.

As a measure of the goodness of segmentation, the Dice index [56] has been chosen. The Dice index, also called the overlap index, is the most used metric in validating image segmentations, this index was used for a direct comparison between ground truth and automatic segmentations. More generally, the Dice measures the spatial overlap between two segmentations, A and B target regions, and is defined as:

$$Dice(A, B) = 2(A \cap B) / [(A \cup B) + (A \cap B)] \qquad (8)$$

where the symbol ∩ represents the intersection, and the symbol ∪ is the union.

The Dice index is calculated considering for A and B the two segmentation masks: the one produced automatically by the method and the one produced manually by Specialists in the sector (the latter is considered the ground truth).

Another widely used index is the Jaccard index defined below:

$$Jaccard(A, B) = (A \cap B) / (A \cup B) \qquad (9)$$

Both indices measure the similarity between finite sample sets, in this case between the segmentation mask considered ground truth and the segmentation mask obtained from the automatic process. The value of these indices is between zero and one, where one represents the exact overlap of the two masks.

The segmentation methods were tested using the Mivia [30] and the AIDA [36] datasets (see subsection 1.4 “Public HEp-2 database”). MIVIA have 28 Hep-2 images and for each of them specialists manually segmented each cell. The public AIDA database does not provide segmentation masks, but on a subset of 95 images of the private database the ground truth segmentation was obtained with manual segmentation performed by an expert in the field. Their distribution is shown below:

- 15 homogeneous;
- 16 speckled (of which 8 fine speckled and 8 coarse speckled);
- 19 nucleolar (of which 9 nucleolar clampy, 7 nucleolar homogeneous and 3 nucleolar speckled);
- 15 centromere;
- 23 nuclear dots (of which 14 few nuclear dots and 9 multi nuclear dots);
- 7 nuclear membrane.

3.3 Automatic method for identifying the best segmentation configuration

Starting from monochrome images, a classic segmentation pipeline consists of a first phase of image pre-processing followed by binarization techniques and finally a third phase of post-processing. The pre-processing aims to improve the signal-to-noise ratio, even excluding, from the image regions that are not of interest for the subsequent CAD phases. Coupling a type of pre-processing to a binarization technique can significantly increase the goodness of segmentation. Furthermore, the post-processing of binarization can refined the overall segmentation. Due to the variety of HEp-2 patterns, a segmentation technique that works well on a given pattern often does not perform well on other patterns. Figure 3.3 shows an example of a segmentation result, obtained with the standard method, on two types of patterns; as it is easy to see, the method has a good homogeneous pattern segmentation capability, but a poor centromeric pattern segmentation capability.

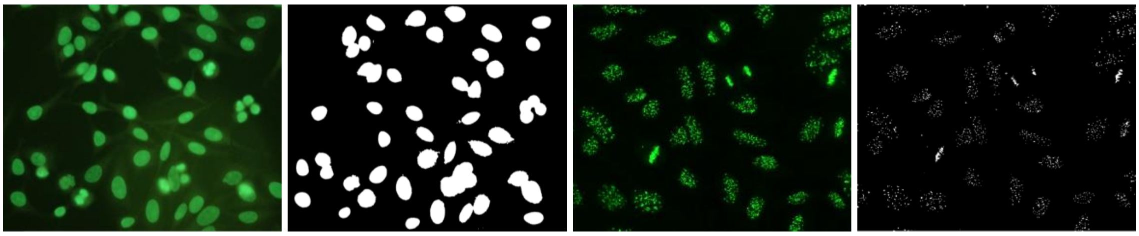


Figure 3.3: From left to right, the first image is a pattern homogeneous, the second image is the segmentation mask of the first image obtained with adaptive Otsu threshold, the third image is a centromere pattern and the fourth image is the segmentation mask relative to the third image obtained with the same segmentation technique.

In this work, to find a good segmentation process, various combinations of pre-processing, binarization techniques and post-processing were analyzed. The idea was to perform an automatic search for the best segmentation configuration based on the three phases of pre-processing, binarization and post-processing by optimizing the indexes of Dice and Jaccard. The analysis is conducted considering three lists of functions:

- the first list contains pre-processing methods (the list is shown in Table 3.1);
- the second list refers to the binarization methods (the list is shown in Table 3.2);
- the third list contains post processing methods (the list is shown in Table 3.3);

Conceptually, the more diversified techniques are present in the three lists, the higher the possibility of finding the sequence of operations that maximizes the performance of segmentation in terms of Dice and Jaccard indexes. Obviously, the more techniques are present in the lists, the more combinations to be analyzed and the calculation times could make an exhaustive search impractical.

With regard to pre-processing, the best known techniques have been considered, both those that operate on the modification of the histogram, and techniques that operate on the spatial domain locally with the application of convolution masks. See Appendix B for more detailed descriptions of the digital image processing used. The Table 3.1 summarizes the types of pre-processing functions.

Pre-processing	Abbreviation	Description
Nothing	Nt	It does not apply any image processing.
Contrast normalization	Cn	Operates on histogram, linearly remapping the intensity values so that 1% of data is saturated at low and high intensities.
Equalization	Eq	Operates on histogram, remapping the intensity values to increase the global contrast by distributing the pixel values uniformly over the entire range of possible values.
CLAHE	Ch	Contrast Limited Adaptive Histogram Equalization, operate on small regions histograms rather than on the entire image, each processed region is combined with adjacents ones using bilinear interpolation.
Gaussian filter	Gs	Applies convolution filtering with Gaussian kernel.
Median filter	Md	Applies convolution filtering with Median kernel.
Morphological filters	Dl / Er / Op / Cl / Fs	Applies a morphological operation of dilation / erosion / opening / closing / FAS (Filter Alternate Sequential) on the grayscale images with a chosen structuring element.
Anisotropic diffusion filter	An	Applies the anisotropic diffusion filter which, based on the local image content, reduces image noise without blurring the edges.
Bilateral filter	Bl	Applies the bilateral filter which, based on the local image content, reduces image noise without blurring the edges.

Table 3.1: Preprocessing functions analyzed.

Three types of operations at the histogram level were chosen to enhance contrast. Two of these are the well-known operations of contrast normalization and equalization of the histogram. The third, CLAHE (Contrast Limited Adaptive Histogram Equalization) is a block-based processing, and it can overcome the over amplification of the noise problem in the homogeneous region of the image with standard histogram equalization. It operates on small regions in the image, called tiles, rather than on the entire image. Each tile contrast is enhanced, the neighbouring tiles are then combined using bilinear interpolation to eliminate artificially induced boundaries.

As to the filters, the common Gaussian filters and the median type were chosen, known above all as means for reducing noise. Since the latter filters can lead to smoothing problems, advanced filters such as the anisotropic diffusion filter and the bilateral filter have also been considered which manage to decrease noise without blurring the edges. Finally, the group of morphological filters on grayscale were also evaluated.

Many of the functions in the Table 3.1, have been analyzed using several parameters (e.g., median kernel of size 3x3, 5x5, and 7x7 have been analyzed), and furthermore pre-processing pairs were evaluated. Obviously, considering all the possible pairs, computational analysis becomes unsustainable, while it is possible (and reasonable) to couple a contrast normalization technique with a filtering type.

Binarization	Abbreviation	Description
Otsu thresholding	Ot	Histogram thresholding using the well-known Otsu algorithm.
Max entropy thresholding	Me	Histogram thresholding using the max entropy method.
Adaptive Otsu thresholding	Ao	Adaptive histogram thresholding using the Otsu algorithm. The algorithm works on small regions and gives different thresholds.

Table 3.2: Binarization functions analyzed.

Table 3.2 summarizes the binarization methods used. These are based on the histogram threshold which are well suited for a preliminary identification of HEp-2 cells. In addition to the two well known methods such as the Otsu threshold [57] and the threshold based on max entropy [58], the adaptive variant of the Otsu algorithm was considered. This algorithm determines the threshold for a pixel based on a small region around it. So, it gets different thresholds for different regions of the same image which gives better results for images with varying illumination. See Appendix B for more details.

Post-processing	Abbreviation	Description
Remove boundary cells	Rb	Eliminate the ROIs that are in the boundary of the image.
Filling	Fl	Fill ROIs where holes are present.
Morphological filters	Dl / Er / Op / Cl	Applies a morphological operation of dilation / erosion / opening / closing on the binary image with a chosen structuring element.

Table 3.3: Post-processing functions analyzed.

Table 3.3 lists the post-processing functions and their brief description. The post-processing phase is carried out on the binary image obtained in the binarization with the aim of perfecting it. In HEp-2 images, for example, there may be cells partially visible on the edge, these must be discarded because they carry little information or, depending on the features that will be extracted from the ROIs, they could penalize the classification phase. Furthermore, it may happen that the segmentation of some cells have holes, these in the post-pocessing phase are filled with the filling technique based on morphological filtering. The application of other morphological filters can make it possible to refined the segmentation. In particular, the four common morphological filters (dilation, erosion, opening and closing) were coupled to the remove boundary and the filling. The Figure 3.4 shows an example of segmentation highlighting the three phases of the segmentation process.

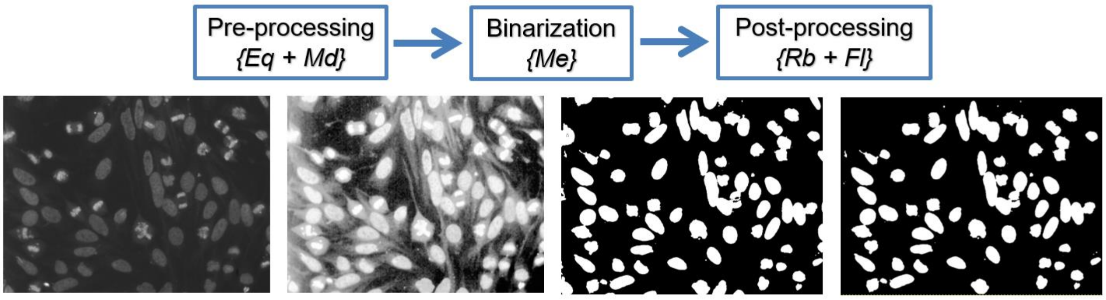


Figure 3.4: Segmentation based on the three phases: pre-processing (equalization and median filter), binarization (max entropy threshold) and post-processing (remove boudary cells and filling).

The automatic search of the segmentation process consists in carrying out an intensive analysis of pre-processing coupled to binarization and post-processing on image datasets provided by manual segmentation masks considered as ground true. Each combination (or pipeline of segmentation) is performed on the images to obtain the relative automatic segmentation masks. Finally, the Dice and Jaccard indexes are calculated for each image thanks to the reference mask obtained manually. As a result, the combination with the maximum index will be the one chosen. In order to analyze the results of the various combinations of segmentation, the indices are maximized both on the entire dataset but also on subsets relating to HEp-2 patterns. Figure 3.5 shows the flowchart of the iterative automatic search of the best segmentation pipeline.

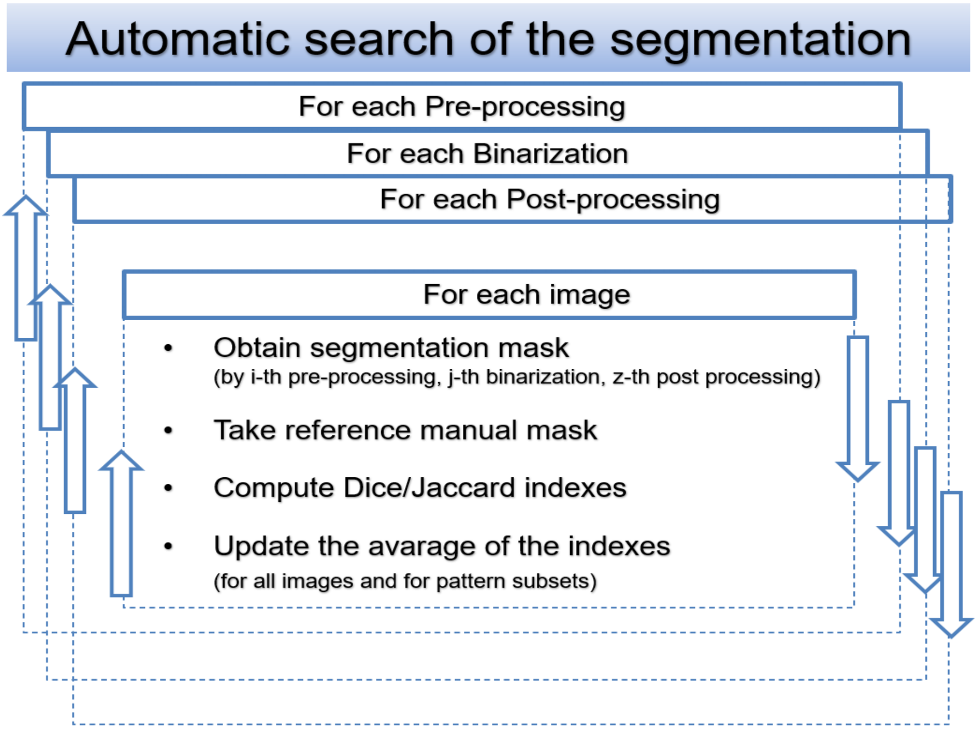

Automatic search of the segmentation
For each Pre-processing
For each Binarization
For each Post-processing
For each image
Obtain segmentation mask
(by i-th pre-processing, j-th binarization, z-th post processing)
Take reference manual mask
Compute Dice/Jaccard indexes
Update the avarage of the indexes
(for all images and for pattern subsets)

Figure 3.5: Flowchart of the iterative automatic search of best segmentation pipeline.

3.4 Segmentation with Hough trasfom and Active countourn

Very often, within HEp-2 images the cells are partially overlapped resulting in a separation problem that a threshold-based segmentation cannot overcome. To overcome this problem, a refined segmentation method based on the identification of cell contours was implemented. Furthermore, this method can better define the contours of a cell than a threshold-based method. This refined segmentation was carried out by developing the following three phases:

1 pre-segmentation: aimed at identifying regions of interest (derived from the first method of threshold-based segmentation);
2 randomized Hough transform for ellipse detection: aimed at identifying the ellipse that best characterizes the generic cell;
3 active contour model: starting from an elliptic curve, evolve expanding towards the cellular contour.

Table 3.4 shows some examples of this segmentation process.
The proposed method, for a greater definition of cellular contours, uses the active contours in the last phase of the process. The purpose of the active contours, within the segmentation process, is to allow the separation of connected regions, such as two overlapping cells, in order to obtain a better definition of the the cells to be analyzed. The initial conditions, center position and initial curve of the active contour, were obtained using the pre-segmentation and the randomized Hough transform for ellipses. The pre–segmentation method (see Table 3.4 second row) is composed of the following operations:

1 selection of the green channel;
2 anisotropic filter;
3 adaptive Otsu thresholding;
4 removal of boudary and small ROIs.

The foreground regions identified in the pre-segmentation may either contain one individual cell or multiple touching cells. The identification of the equivalent ellipse allows the definition of the center and dimensions for the generic cell. Many methods have been developed for the identification of geometric figures, certainly among the most performing is the Hough transform. Hough transforms are techniques commonly used to detect lines, circles, ellipses, etc. The basic idea of the Hough transform is to implement a voting procedure for all potential lines or curves in the image, and at the termination of the algorithm, curves that do exist in the image will have relatively high voting scores. Due to the many parameters involved in the detection of ellipses, the various methods developed often limit their recognition dimensions, or search for sub-images. A complete search would be computationally not feasible. The traditional approach for ellipse detection using the Hough technique is similar to a line or circle detection. This approach is not only memory expensive but also computationally intensive.

In this thesis, for an ellipse detection, the randomized Hough Transform with result clustering is used [59]; the method uses only a one dimensional accumulator for ellipse voting, and reducing algorithm complexity. Specifically, the Randomized Hough transform is a probabilistic variant of classical Hough transform and it takes advantage of the fact that some analytical curves can be fully determined by a certain number of points on the curve. For example, a straight line can be determined by two points, and an ellipse (or a circle) can be determined by three points.

Conceptually, for each ROI, a dilated boundary box is cut out and the contour of the related pre-segmentation mask is extracted. The randomized Hough transform for ellipse is applied to this (see Table 3.4 third row).

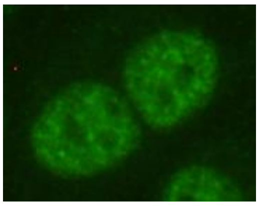
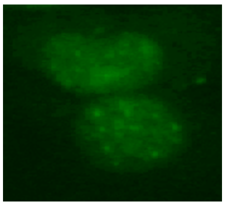
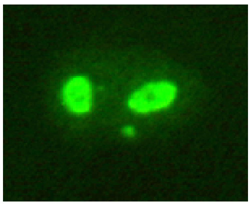
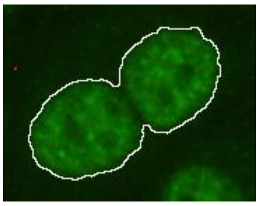
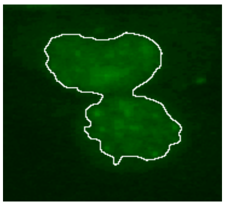
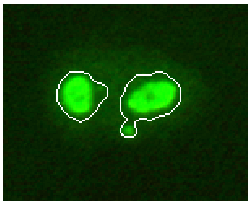
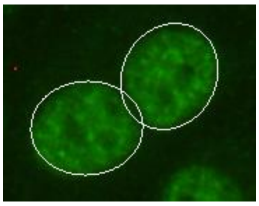
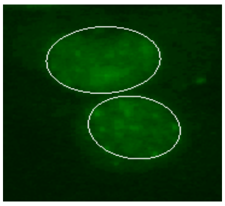
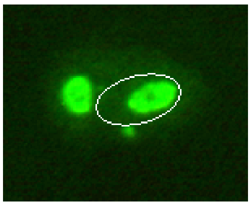
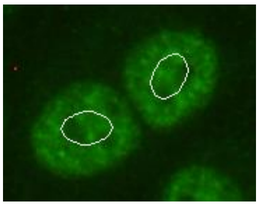
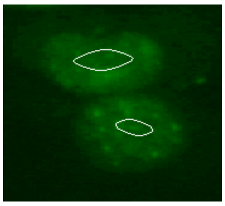
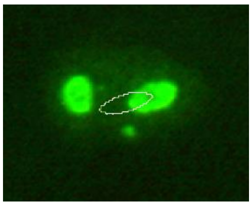
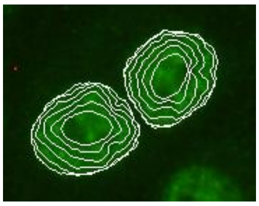
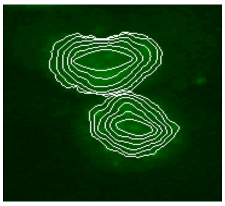
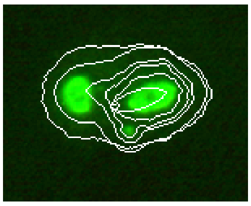
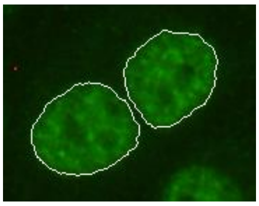
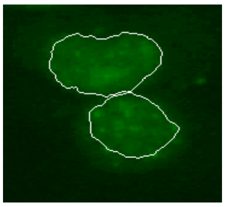
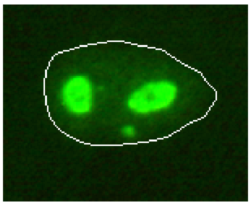
Step1: Dilated boundary box of the related pre-segmentation mask			
Step2: Contour of the pre-segmentation mask			
Step3: Randomized Hough Transform for ellipse detection			
Step4: Initialization of the active contour with reduced ellipse			
Step5: Evolution of active contour			
Step6: Final contour of the new ROI/s			

Table 3.4: Schematic procedure of refined segmentation based on the pre-segmentation, randomized Hough transform for ellipses, and active contours.

The problem of the identification of the contours for patterns that are visually very different and need to be separated from overlapping cells inside the Hep-2 image, has been addressed using an active contour model. This family of algorithms has the advantage that, if properly initialized, they can converge on the correct contour of objects and since the evolution involves a non-punctual analysis of the image, they allow the separation of contiguous but visually distinct objects. The characteristics of the active contour model algorithms allow to address the problem of cells overlapping. In particular, the method used is a model of active contours based on techniques of curve evolution, Mumford–Shah functional for segmentation and level sets [60]. This model can detect objects whose boundaries are not necessarily defined by gradient. The method evolves minimizing an energetic functional which can be seen as a particular case of the minimal partition problem; the method is not based on an edge-function to stop the evolving curve on the desired boundary. In the level set formulation, the problem becomes a "mean curvature flow" like evolving the active contour, which will stop on the desired boundary. However, the stopping term does not depend on the gradient of the image, as in the classical active contour models, but is instead related to a particular segmentation of the image. The method is particularly stable and allows good convergence even if the initial conditions are not optimal. Also, this model does not require image smoothing and in this way, the locations of boundaries are very well detected and preserved.

The result of the Hough transform was used to initialize the active contour. In particular, the ellipse identified on the generic cell has been reduced in size, halving the axle shafts, in order to obtain an evolution of contour, which in expansion tends to the desired cell boundary. Figure 3.6 shows some examples of initialization and evolution of the active contour obtained on a pattern which is particularly difficult to segment: the cytoplasmic pattern. The average iterations number for convergence of proposed method is around 80, while The max numbere of iteration is set to 100 but the process also stops the evolution if the contour position in the current iteration is the same as the contour position in one of the most recent five iterations. When the final contour is obtained, the filling is done and the refined segmentation mask is returned.

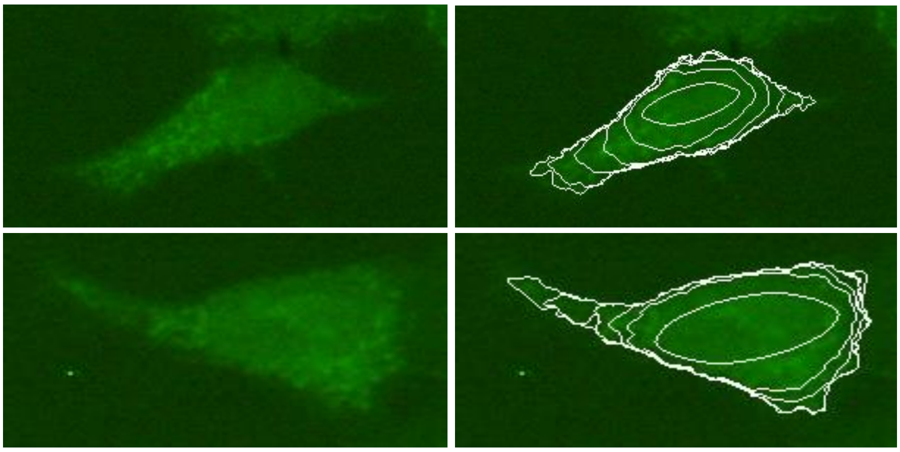


Figure 3.6: Two examples of cytoplasmic cell and the respective active contour evolution (with a step of 25 iterations).

Figure 3.7 and Figure 3.8 emphasize the difference between the manual segmentation mask (ground truth) and the two automatic segmentation masks obtained with pre-segmentation and with refined segmentation based on the active contours.

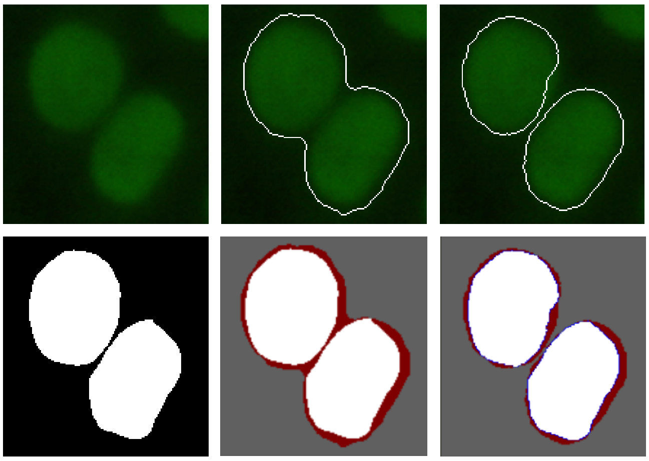


Figure 3.7: From left to right, the first row shows two homogeneous cells attached and their associate pre-segmentation and refined segmentation contour; the second row shows the manual segmentation mask and the difference with the pre-segmentation and refined mask.

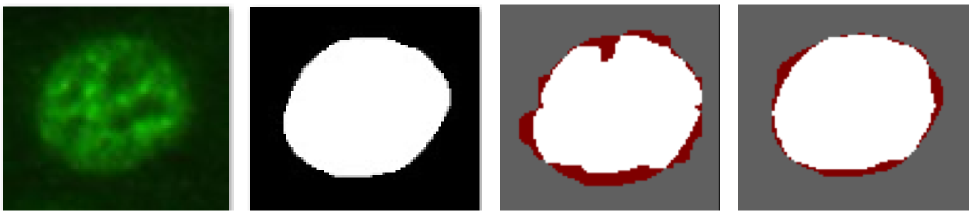


Figure 3.8: From left to right, a single HEp-2 cell, its manual segmentation mask and the difference with the pre-segmentation and the refined segmentation mask.

For both the ellipses and the contours a simple check on the size and overlap with the pre-segmented mask is carried out in order to avoid errors. In this case, the process is iterated again and if the error is not exceeded, the presegmentation mask is chosen.

3.5 Experimental results

This section shows the results obtained. For the first segmentation method based on pre-processing, binarization and post-processing 13 different types of pre-processing (see Table 3.1), 3 types of binarization (see Table 3.2) and 6 types of post-processing (see Table 3.3) were considered. Several different configurations where evaluated since some functions can be used with different parameters (e.g. for the median filter the convolution masks, sized 3x3, 5x5 and 7x7 were analyzed). Furthermore, some pre-processing pairs were evaluated.

Therefore, the automatic segmentation search analyzed 60 different pre-processing, 7 different binarizations and 14 different post-processing for a total of 5040 segmentation processes. Four PC with a 3,4 GHz Intel i7 CPU were used for the analysis. Each segmentation process was conducted on the 95 images of the AIDA dataset and on the 28 images MIVIA. The execution time on an image varies significantly with the variation of the segmentation process and in part with the variation of the image size. To optimize the execution time, the implementation was performed in C/C++ code with the IDE visual Studio 2010 and using high-performance libraries such as OpenCV (http://opencv.org). The execution was also parallelized by considering four executions for each computer. The choice of splitting into 4 executions is due to the fact that most of the image processing functions used are implemented using the OpenCV library which exploits optimized algorithms and which implicitly use parallelization. As evidence of this, the median filtering is given as an example, which for each convolution of the mask must order the elements and take the median. The implementation in C/C++ language of the classic algorithm without parallelization takes on average 7.7 sec. on an image compared to 2.3 sec. using the median filter of the OpenCV library. The average calculation time of the 5040 segmentation processes on an image is about 9 seconds. The overall computation time is about:

*computation time = 123 images * 9 sec. * 5040 processes = 5579280 sec. = ~ 65 days*

Regarding the second method of refined segmentation based on the pre-segmentation, randomized Hough transform for ellipse, and active contour, the implementation was done with Matlab 2017 (MathWorks, Natick, Massachusetts, USA). In this case the parameterization of the randomized Hough transform for ellipse function and of the active contours function was heuristically carried out on a small subset of images to find consistent parameters. In this way, only 12 processes (with different parameters) were iterated over the datasets. The average calculation time for a whole image is about 137 sec. The main parameters are included in the Table 3.5.

Functions	Parameters & description	Values used
Hough trasform	Initialization	pre-segmented contour
	Minimal length of ellipse major axis	50 (in pixel)
	Maximal length of ellipse major axis	150 (in pixel)
	Range angle of the major axis in degrees	no restrictions
	Minimal aspect ratio of an ellipse	0.5
	Randomize subsampling of all possible point pairs. Instead of examining all N*N pairs, runs only on N*randomize	[1, 2] in the range [0, N] {2 is the best value used}
Active contour	Initial contour	eroded Hough ellipse
	Maximum iterations	100
	Degree of smoothness Higher values produce smoother region boundaries but can also smooth out finer details.	[0.2, 0.3] in the range [0, 1] {0.2 is the best value used}
	Contraction bias negative values bias the contour to grow outwards (expand) while positive to shrink inwards (contract)	[-0.3, -0.5, -0.8] in the range [-1, 1] {-0.5 is the best value used}

Table 3.5: Description of main parameters and values used for randomized Hough transform and active contours.

The performance of the two segmentation methods differentiated for each patterns is shown in Table 3.6 for the MIVIA dataset and in Table 3.7 for AIDA dataset.

MIVIA Patterns	**Pre-segmentation** (pre-processing/binarization/post-processing)			Pre-segmentation **Dice & Jaccard**	Refine-segmentation **Dice & Jaccard**
Homogenous	Nt	Ao	Fl+Rb	0,88 / 0,78	0,93 / 0,84
Fine speckled	An	Ao	Fl+Rb	0,79 / 0,67	0,89 / 0,78
Coarse speckled	Eq	Me	Fl+Rb	0,76 / 0,63	0,87 / 0,74
Nucleolar	St+Md	Ao	Fl+Rb	0,73 / 0,58	0,86 / 0,74
Centromere	An	Ao	Dl+Fl+Rb	0,82 / 0,70	0,84 / 0,71
Cytoplasmic	St+Md	Ot	Er+Fl+Rb	0,72 / 0,56	0,80 / 0,68
ALL (28 images)	**St+Md**	**Ao**	**Fl+Rb**	**0,74 / 0,59**	**0,87 / 0,73**

Table 3.6: Performance of the pre-segmentation abbreviation and refined segmentation based on active contours on MIVIA dataset. Abbreviations from tables 3.1-3.3 are used to indicate the algorithms.

AIDA Patterns	**Pre-segmentation** (pre-processing/binarization/post-processing)			Pre-segmentation **Dice & Jaccard**	Refine-segmentation **Dice & Jaccard**
Homogenous	Nt	Ao	Fl+Rb	0,85 / 0,75	0,89 / 0,77
Fine speckled	St	Me	Fl+Rb	0,86 / 0,76	0,91 / 0,80
Coarse speckled	St+Md	Ao	Fl+Rb	0,85 / 0,75	0,87 / 0,76
Nucleolar hom.	Md	Ao	Fl+Rb	0,85 / 0,75	0,88 / 0,77
Nucleolar cl.	Er	Ao	Dl+Fl+Rb	0,75 / 0,60	0,81/ 0,70
Nucleolar sp.	Md	Ao	Fl+Rb	0,70 / 0,54	0,79 / 0,68
Centromere	An	Ao	Dl+Fl+Rb	0,80 / 0,68	0,84 / 0,74
Few nuclear dots	St+Md	Ot	Fl+Rb	0,84 / 0,73	0,86 / 0,75
Multiple n. dots	An	Ao	Fl+Rb	0,87 / 0,77	0,88 / 0,78
Membranous	Md	Ao	Dl+Fl+Rb	0,85 / 0,74	0,91 / 0,79
ALL (95 images)	**An**	**Ao**	**Fl+Rb**	**0,83 / 0,72**	**0,85 / 0,74**

Table 3.7: Performance of the pre-segmentation abbreviation and refined segmentation based on active contours on AIDA dataset. Abbreviations from tables 3.1-3.3 are used to indicate the algorithms.

It can be seen how different types of pre-segmentation adapt to various types of patterns, but this presupposes knowing the belonging of a given class. From the pre-segmentation performance Table it is easy to deduce that, some patterns, such as the homogeneous one, are more easily segmentable than others that have many irregularities. However, this difference remains limited. Considering the pre-processing it can be seen that only in the homogeneous case using preprocessing (Nt) does not bring improvements. In all other cases, pre-processing is present and the two most useful are the median and the anisotropic. The median filter is often coupled with contrast normalization and its presence is probably due to its ability to lower noise while avoiding major smoothing problems. The anisotropic filter in its complexity is also able to reduce noise while preserving the edges. Although the bilateral filter does not appear among the highest results, it achieves excellent results in a similar way to the anisotropic filter. Among the binarization methods the dominant one is the adaptive Otsu threshold. This is due to the fact that, on small portions of the HE-p2 image it is easier to find an optimal separation threshold. As for the post-processing what is done by default is the remove boundary cells and the filling. Only a few times a morphological filter is coupled.

It is evident that the refined segmentation, based on randomized Hough transform and active contours, improves the goodness of segmentation of all analyzed patterns. It can be noted that, the improvement is limited for the homogeneous pattern and for the patterns that have intense luminous dots inside the nucleus, such as the centromeric pattern and the two nuclear dosts patterns. In the homogeneous case, the slight improvement is due to the fact that the pattern has a homogeneous aspect inside the nucleus and this allows a good segmentation even with the pre-segmentation technique based on histogram thresholding, therefore the refinement does not produce an important improvement. However, for the centromeric and nuclear dots patterns the refinement remains conditioned by these intensity peaks. On the other hand, among the patterns that take the most advantage there is the cytoplasmic one characterized by an irregular shape and the membranous whose contour is better identified by the increasing progression of the active contour. Another pattern that is difficult to segment is the nucleolar one, this is due to the presence of more intense masses present within the nucleus whose contours are more marked than the contour of the cell. In spite of the remarkable diversity of the patterns analyzed, the refined segmentation achieves very similar segmentation results for the different patterns, demonstrating a good robustness.

The cell segmentation method presented in this thesis has been compared with other segmentation methods of HEp-2 images that have been proposed in recent years. The performance comparison with other state-of-the-art methods, in terms of the Dice index, is shown in Table 3.8. It was possible to make a comparison with those studies that presented the Dice index as a figure of merit, or for which it was possible, from the data presented, to obtain the aforementioned index.

	Images	Images dataset	Dice index
Percannella et al [54]	28	MIVIA	56,8%
Tonti et al [55]	28	MIVIA	62,1%
Roy et al [52]	22	MIVIA (Less centromere and cytoplasmic patterns)	86,8%
Cheng et al [53]	196	Private	88,9%
Proposed method	28	MIVIA	87,0%
	95	AIDA	85,2%

Table 3.8: Comparative performance of segmentation methods.

Table 3.8 shows how many works have been developed using a particularly small number of images. The performance comparison presented in the Table clearly shows the quality of the proposed system.

Chapter 4 - Fluorescence intensity classification

This chapter exposes the classification of HEp-2 images in positive or negative fluorescence. The fluorescence intensity classification is the first step in the diagnostic workflow. In this chapter the different classification methods developed and compared are presented. Finally, the results obtained are compared with other methods at the state of the art.

4.1 Introduction and related work

The classification of HEp-2 images, conducted through Indirect Immuno Fluorescence gold standard method, in the positive / negative classes, is the first step in the diagnosis of autoimmune diseases. This binary classification determines (in case of positive output) whether the analysis will be performed to identify the staining patterns present in the image. Since the test is often difficult to interpret, it is subjective and highly dependent of many factors as Hep-2 slide manufacturers, fluorochrome conjugated, microscope, camera settings and other variables, the research world has been looking for technological solutions to this problem.

Figure 4.1 highlights the difficulty of interpretation between positive and negative Hep-2 images.

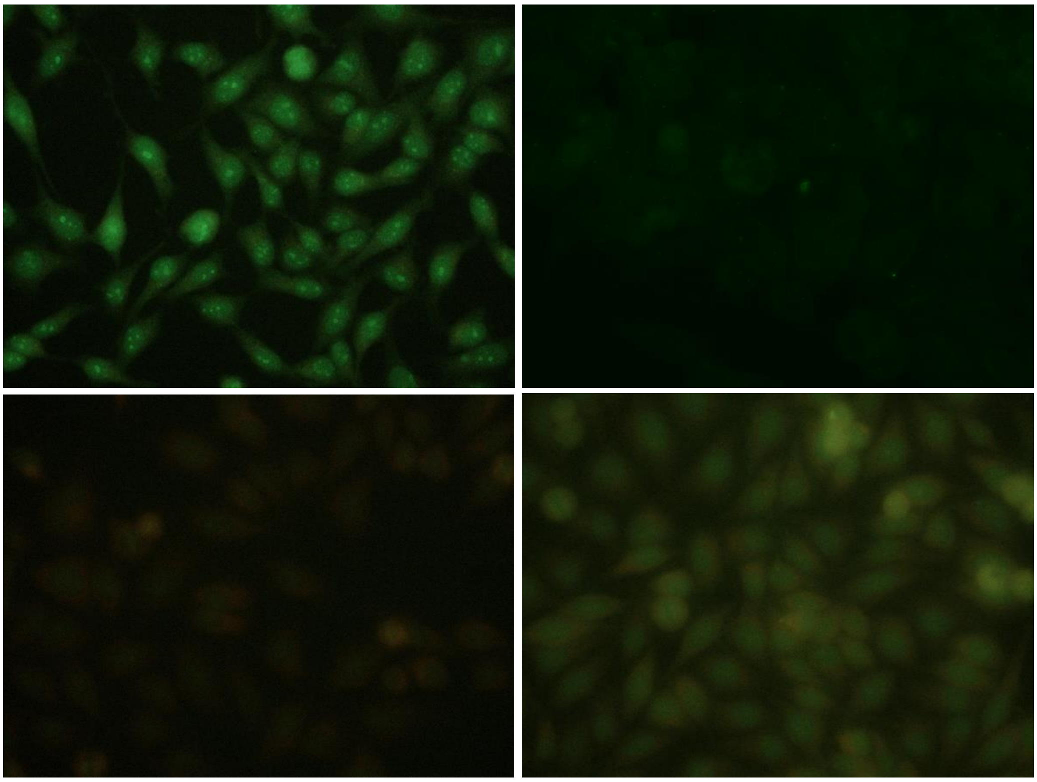


Figure 4.1: IIF images with different fluorescence intensity: at the top two positive examples, below two negative examples.

Since deep learning and in particular CNNs (see appendix A) have overcome the traditional machine learning methods, in this thesis it was decided to tackle the classification in positive and negative intensity both with the traditional classification method and with CNNs. Performance analysis was conducted in terms of ROC (Receiver Operating Characteristic) curve using the public database AIDA.

The problem of identifying the fluorescence intensity is still little addressed in the literature. The reason is probably due to the lack of public databases containing both positive and negative images; to date, it seems that the only public database of HEp-2 images with these characteristics is AIDA [36].

In Merone et al [61] the problem of fluorescence intensity analysis is addressed but the authors do not make a classification between positive and negative, rather between positive, negative and weak positive. The authors extracted features through an Invariant Scattering Convolutional Network and used SVM classifiers with a Gaussian kernel. The network was based on multiple wavelet module operators and was used as a texture description. To classify the three-class the authors applied the one-on-one approach and trained the tree binary classifier (negative vs positive / negative vs weak positive / negative vs weak positive). Their method was trained on a private database of 570 wells for a total of 1771 images in which the fluorescence intensity was blindly classified by two physicians. The accuracy reported in the private database was 89.5%.

A classification of fluorescence intensity into positive vs. weak positive was also carried out by Di Cataldo et al. [62]. The authors used local contrast features at multiple scales and a KNN classifier to characterize the image, thereby achieving an accuracy of 85% in fluorescent intensity discrimination.

In Benammar Elgaaied et al [36] the authors have implemented a method based on SVM to classify the HEp-2 images in positive or negative intensity. They get an accuracy of 85.5% using traditional features based on intensity, geometry and shape. The same set of tests was analyzed by two young immunologists verifying a greater ability of the automatic system (85.5% vs 66%).

Other authors addressed the classification in positive / negative intensity on private database. Iannello et al [63] used a private database with 914 images to train a classifier able to discriminate between positive and negative intensity. Some areas of interest called patches were extracted from the training set with the aid of the SIFT algorithm, then 19 features were extracted from these. The features were extracted from first and second order gray level histograms. Two feature reduction methods were applied, the PCA (principal component analysis) and LDA (linear discriminant analysis). Finally, the classification was based on the Gaussian mixture model and reached an accuracy of 89.49%.

In the work of Zhou et al [64] the authors presented a fluorescence intensity classification method in which private databases are analyzed. The method makes use of the fusion of global and local type features. For simple cases they use the SVM classifier with global features, while for doubtful cases they propose a further classification based on local features combined with another SVM. The results show an accuracy of 98.68%. However, as the analysis was conducted on a private database, the work does not allow easy performance comparison.

4.2 The methods implemented

Historically, the binary classification with supervised modality has been widely discussed in the literature in the field of biomedical images and beyond. The widely used pipeline includes a pre-processing phase of the images, a possible process of segmentation of the ROI, the extraction of features, a possible features selection and finally the use of a classifier. Image processing, in addition to any noise reduction, allows it to modify the image to extract some peculiar characteristics. The choice of features is a fundamental requirement as they must possess the information necessary to allow discrimination between the various patterns to be recognized. In fact, only if the characteristics are able to extrapolate the discriminating traits, it is possible to classify the two categories with appreciable results. Feature selection is the process of selecting an optimum subset of features from a set of potentially available features in a given problem domain. Generally, only some features contain significant information, features that have an insignificant contribution to classification can be eliminated to reduce the dimensionality of the feature vector and to maximize the performance of the classifier. After the determination of the characteristics, the classifier is trained on the basis of a dataset of images already classified by experts. All supervised learning algorithms start from the assumption that, if a number of *representative* and *complete* examples of the problem to be classified is provided, the resulting classifier will be able to distinguish the new specimens correctly.

Traditionally, the feature extractor has been manually designed by experts of specific areas. Therefore, it required a significant amount of cost and time. In recent years CNNs have moved beyond the traditional pipeline, they include the feature extractor in the training process rather than designing it manually. The CNNs feature extractor is composed of special kinds of neural networks, of which the weights are determined via the training process. The fact that CNN turned the manual feature extraction design into the automated process is its primary feature and advantage.

In this thesis both the traditional classification pipeline and the recent CNNs are developed and compared for the classification of the fluorescence intensity of HEp-2 images.

To train the classifier, the traditional method is consisting of five phases:

1 image pre-processing;
2 ROIs extraction;
3 feature extraction;
4 feature reduction;
5 classifier training.

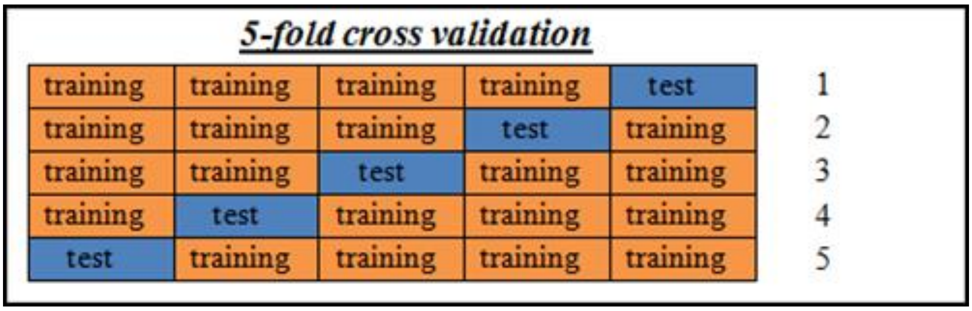
The method with CNNs uses the most popular "pre-trained" CNNs such as "AlexNet", "Squeezenet", "Resnet18", "Googlenet" rather than creating ad-hoc networks. The pre-trained networks have aroused great interest in the scientific community thanks to the "ImageNet" competition, classifying 1000 classes of objects and the possibility of performing transfer learning. In fact, pre-trained CNNs are well suited to the fine-tuning strategy, where they can be re-trained for a specific classification problem using an appropriate database. The pre-trained CNNs were used in the following two ways:

1 train a new classifier with the features extracted from the pre-trained CNN layers; CNNs, being trained on databases containing millions of images, are able to model very performing generic features sets;

2 perform pre-trained CNN transfer learning; in this case the last layer must be suitably replaced according to the classes to be discriminated and a fine-tuning carried out using the image database to be classified.

These methods have been quantitatively assessed considering AUC and Accuracy as the main figure of merit (see subsection 2.4 “CAD system evaluation”). The assessments of the binary classification has been carried out considering the AIDA public dataset [36]. This database is the only one to have both positive and negative cases, it consists of 2080 images, composed as follows: 998 patients (261 males, 737 females), 1498 images show positive fluorescence intensity, 582 show negative intensity.

In the training–tuning phase, in order to make the best use of the data, the leave-one-specimen-out (LOSO) cross-validation technique avoids bias problems. This method consists of leaving out one specimen, rather than leaving out a single image (or a single cell) for the construction of the training set; images of the same specimen, belonging to the same patient, are similar (in terms of the average intensity and contrast) and introduce bias. The specimen left out is used for validation. A variant of the LOSO is the k-fold validation considering the specimens. A k = 5 have been used, so the DB was divided into 5 folds. With this strategy, 5 trainings and related tests are performed. The Figure 4.2 shows the concept of 5-fold cross-validation. Approximately at each iteration 20% of the dataset was used for the test, the remaining 80% divided into training and validation to the extent of approximately 64% and 16% of the dataset.



5-fold cross validation

training	training	training	training	test	1
training	training	training	test	training	2
training	training	test	training	training	3
training	test	training	training	training	4
test	training	training	training	training	5

Figure 4.2: In 5-fold cross-validation the dataset is divided into 5 subsets. Sequentially a subset is used as a test, while the remaining 4 subsets are used as a training set.

As discussed for the segmentation phase in Chapter 3, the observation of the spatial distribution of the three components of the RGB images allows to verify that the information about the fluorescence intensity is mainly contained in the Green component, therefore the methods developed make use of this channel only.

4.3 Intensity classification with traditional approach

Considering the classic pipeline, the Hep-2 images classification in positive and negative fluorescence was conducted through the analysis of image preprocessing techniques both to reduce noise and to highlight certain features. Subsequently, the feature extraction phase considered both sets proposed in the literature and sets of features proposed ad-hoc.

A first classification analysis with the features extracted from the whole image did not give the very performing results compared to using a segmentation and extracting the features from the ROIs. The classification strategy, however, concerns the full slide and not the individual HEp-2 cells, the segmentation of the ROIs allows to exclude the background which is obviously not discriminating for assessing the positivity or negativity of the image. Considering all the ROIs, a significant number of features was extracted, and used by an SVM classifier (see appendix A) in order to associate the generic image with the positive / negative classes. SVM has been widely used in biomedical research and this is certainly linked to the results of the classification, but also to the advantage that this type of classifier depends on a few parameters. To select the best features and reduce the complexity of the classification problem, LDA (Linear Discriminant Analysis) technique for selecting the features was considered.

The classifier training uses a set of images with the relative label, these undergo the pre-processing and segmentation of the ROIs, then feature extraction and reduction, finally the tuning of the classifier parameters. Once the configuration that has optimized the results has been chosen, then the model built can be used on unknown images. In the test phase obviously only the subset of features selected in the training phase are extracted and these are submitted to the classifier. Figure 4.3 highlights the training phases of the classifier and the use phase of the tuned classifier.

For the segmentation, in a preliminary phase both the segmentation based on traditional pipeline, and the one based on active contours were used (see chapter 3). The comparison led to the conclusion that a particularly accurate segmentation did not bring particular benefits. This can be explained by the fact that the classification on intensity concerns the rough appearance of the foreground (cells and their cytoplasm), not in particular the structure of a single cell. For this reason, in order to decrease the calculation time, the method consisting of anisotropic filtering, binarization with adaptive otsu and post processing with filling and removal of the cells present at the edges of the image was chosen.

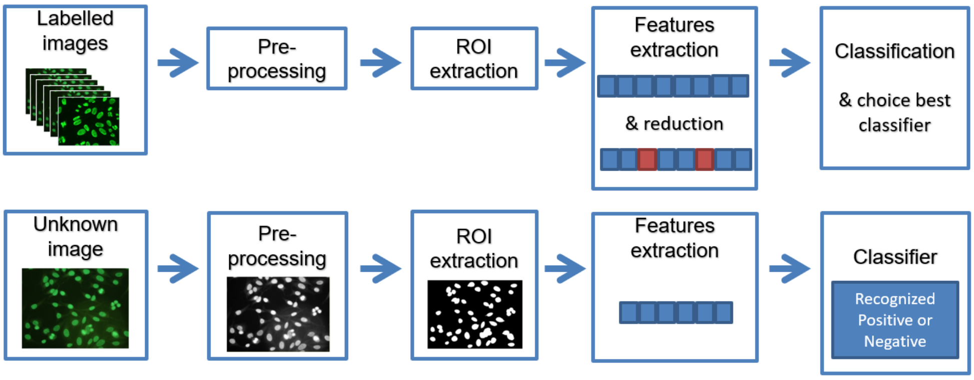


Figure 4.3: In the first row, scheme of the classifier training process. In the second row, the operating scheme of a classification system for fluorescence intensity (after classifier training).

As regards the pre-processing, it is emphasized that this does not necessarily coincide with that used for the segmentation phase, on the contrary a pre-processing that helps segmentation could be very different from the pre-processing used for the classification able to highlight peculiar characteristics. Therefore, the image is pre-processed, then the segmentation mask (obtained with anisotropic filtering) is applied to the pre-processed image. The features are then extracted on these ROIs.

An intensive analysis of the pre-processing function combinations was conducted, aimed at maximizing training performance. In this case, only the techniques that operate on the spatial domain locally with the application of convolution masks were taken into consideration. The techniques that operate on the modification of the histogram were discarded after a preliminary analysis because the results worsened. For example, the simple contrast normalization can invalidate the features based on the intensity of the pixels. Table 4.1 summarizes the types of pre-processing functions used in the training. See Appendix B for more detailed descriptions.

Pre-processing	Abbreviation	Description
Nothing	Nt	It does not apply any image processing.
Gaussian filter	Gs	Applies convolution filtering with Gaussian kernel.
Median filter	Md	Applies convolution filtering with Median kernel.
Morphological filters	Dl / Er / Op / Cl / Fs	Applies a morphological operation of dilation / erosion / opening / closing / FAS (Filter Alternate Sequential) on the grayscale images with a chosen structuring element.
Anisotropic diffusion filter	An	Applies the anisotropic diffusion filter which, based on the local image content, reduces image noise without blurring the edges.
Bilateral filter	Bl	Applies the bilateral filter which, based on the local image content, reduces image noise without blurring the edges.

Table 4.1: Pre-processing functions analyzed in the training of intensity classification.

The characteristics chosen in this work for the classification of fluorescence intensity (the same features have also been used for the classification of the fluorescence patterns as will be discussed in the next chapter) belong to three families of characteristics based on the analysis of the intensity of pixels, ROI geometry and texture (see appendix C for the list of features). Table 4.2 lists the 27 features considered in this thesis.

Features type	Features number	Features name
Intensity	6	mean value, standard deviation, ratio of the standard deviation to the mean value, entropy, skewness, and kurtosis
Shape	12	Area, perimeter, convex area, mean radius, standard deviation of radius, ratio of the standard deviation to the mean value, maximum radius, anisotropy, entropy of the contours gradient, fractal index, eccentricity, and circularity
Texture	9	Contrast, convex deficiency, roundness, compactness, solidity, inertia of co-occurrence matrix, entropy of histogram of oriented gradients (HOG), entropy of histogram of amplitude gradients (HAG), and Euler's number

Table 4.2: List of features divided by type.

Specifically, four different quantization intensity levels were analyzed. The quantizations explored were: 256, 128, 64, and 32 gray levels. The intensity quantization, as is known, affects the quality of the representation of the image. Reducing the number of bits to represent intensity, compresses the storage space, but causes the image quality to deteriorate. The different quantizations aim to highlight the different aspects of ROIs; those with more bits have more details, while those with fewer bits show the shapes more clearly. Overall, the twenty-seven features obtained at four different quantization levels, form a total of one hundred and eight features.

Since not all features fit discrimination in positive and negative, feature reduction is performed to eliminate redundant features and features having poor discriminative power. The main idea of feature subset selection is to remove redundant or irrelevant features, as they can negatively influence the classification accuracy and lead to an unnecessary increase of computational cost. There are many methods proposed in the literature for the decrease in dimensionality, used in supervised or unsupervised classification problems, such as sequential forward search, random forest algorithm, ecc. Discriminant analysis algorithms have been used for dimensionality reduction and feature extraction in many applications of computer vision. Linear Discriminant Analysis (LDA) is probably the most well-known discriminant analysis technique. This method assumes that the C classes to which the data belong, are homoscedastic, that is, their underlying distributions are Gaussian with common variance and different means. The LDA method provides the (C-1)-dimensional subspace that maximizes the between-class variance and minimizes the within-class variance, in any particular data set. In other words, it guarantees maximal class separability and, possibly, optimizes the accuracy in later classifications.

The feature selection is then used by the SVM classifier with a Gaussian RBF (Radial Basic Function) kernel. The two parameters for tuning the RBF SVM are C and γ and were obtained with the "grid-search" method. A practical method to identify good parameters makes use of the exponentially growing sequences. The analyzed values for the C e γ were:

$$C = 2^{-5}, 2^{-4}, \dots , 2^{-10} \qquad \gamma = 2^{-10}, 2^{-9}, \dots , 2^{-2}$$

the analyzed grid had sizes equal to 16 * 13, for a total of 208 grid-points.

In order to identify the optimal functions and parameters an iterative method of configuration analysis was implemented, aimed at maximizing the AUC figure of merit. In Figure 4.4, the flow chart of the iterative method used is shown.

The procedure is iterated on the various pre-processing functions listed in Table 4.1. After the pre-processing phase, the ROIs are extracted with the segmentation method set a priori. The features are extracted from the ROIs and then selected via LDA. Finally, the SVM is tuned through grid seach and the evaluation of the classification is performed with the AUC.

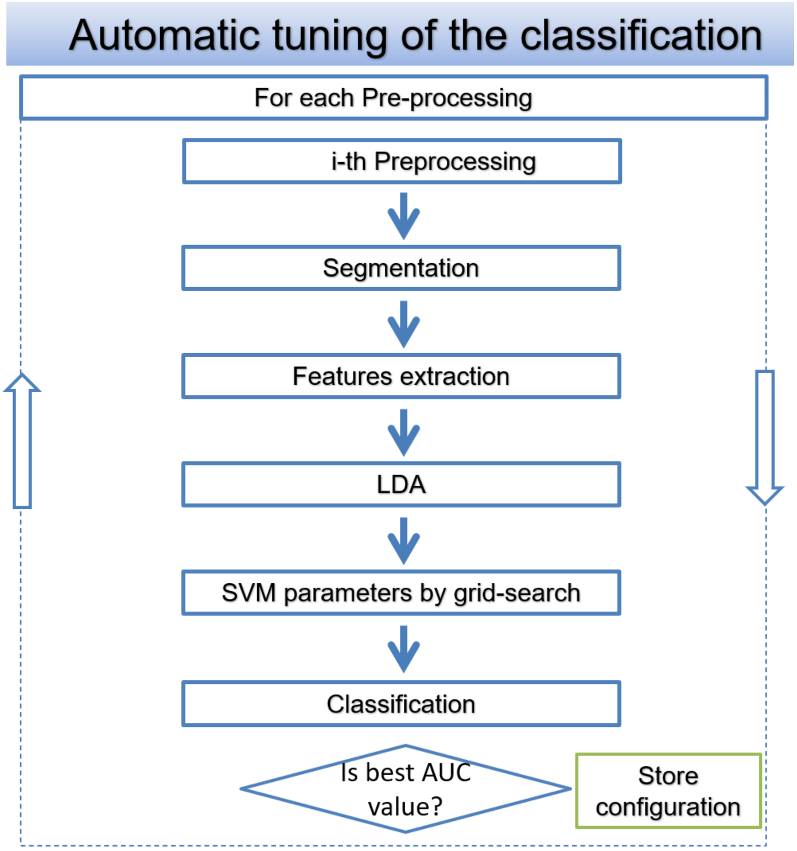


Figure 4.4: Flow chart of the iterative method used for the optimization of binary classification.

4.4 Intensity classification with CNNs

Convolutional Neural Networks (CNNs), have demonstrated their effectiveness in the classification of biomedical images, so the efficacy of the CNN fine-tuning method applied to the classification of fluorescence intensity was investigated. For this purpose, four of the best known pre-trained networks were analyzed: AlexNet, SqueezeNet, ResNet18, GoogLeNet (see appendix A for details). Thanks to ImageNet competition, these CNNs have been trained on over a million images to model generic feature rich representations. In this work therefore, the classifying power of pre-trained CNN was investigated with the two scientifically recognized strategies: feature extraction in combination with linear SVM and fine-tuning with different training modalities. The advantage of the feature extraction strategy is the simplicity of implementation (no retraining of the pre-trained networks must be carried out), the disadvantage is usually relatively lower performances than those obtained from the pre-trained networks with the fine-tuning method

For both strategies, a preliminary analysis of the intensity classification via CNN, showed much higher performance if the entire image is used compared to using segmented ROIs. Furthermore, no pre-processing analysis was necessary, as this does not bring any improvement since CNNs implicitly have different types of convolution. The analized pre-trained CNNs need RGB images in input, but since in the case of HEp-2 images the main information is contained in the green channel, this has also been duplicated in the red and blue channel. In the latter case, the performance exceeds those using the original RGB image.

Feature extraction can be the fastest way to use CNNs pre-trained. Since, as is known, the problem of finding the best set of discriminating features for a given classification problem is very complex, CNNs eliminate the need to identify good features used for image classification. In fact, it is possible to use the power of the pre-trained networks, without investing time and effort in training, to implement the extraction phase of the features. A CNN accepts an image directly as input and applies a hierarchy of different convolution kernels to it. Figure 4.5 shows the general scheme of a CNN. The first layers allow extraction of elementary visual features, such as oriented edges, end-points, and corners, and are gradually combined with subsequent layers in order to detect higher-order features.

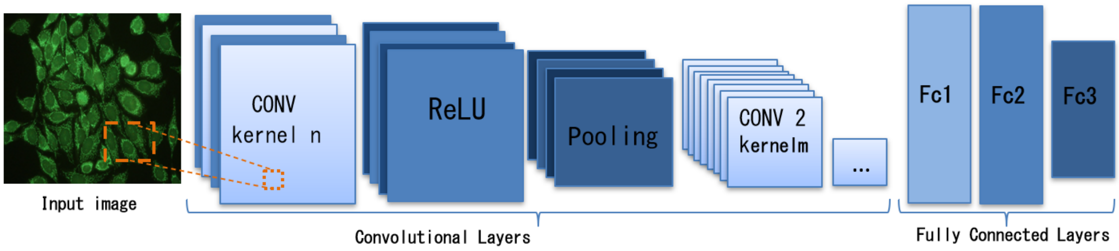


Figure 4.5: General scheme of the architecture of a Convolutional Neural Network.

The effectiveness of the feature extracted from the different CNNs has been used in order to associate the generic image with positive/negative classes. For each CNN different layers were analyzed as feature extractor, each of these vectors was then input to an SVM with linear kernel. As the size of the feature vector is large, it has chosen a linear SVM with only one parameter to tune: the penalty parameter “C” of the error term. The search for linear kernel parameter is carried out in the range [10^{-6}, $10^{-2.5}$] where twenty equidistant values on a logarithmic scale were analyzed. The linear SVM training phase was carried out by extracting the features, for the various layers, from all the training images. The SVM parameter C has been optimized on the validation set. The validated model was subsequently used for the test.

The classifier training uses a set of images with the relative label, these are processed in the CNNs and the feature vectors are extracted from different layers. The parameters of the classifier are set up for each vector. Once the configuration that has optimized the results has been chosen, then the model built can be used on unknown images.

Fine-tuning is a transfer learning technique that focuses on storing knowledge gained while solving one problem and applying it to a different problem. This method consists in the possibility of using an Artificial Neural Network, pre-trained on a large database, through a further training phase with another database, even a small one. The output level is replaced with a new softmax output level, adjusting the number of classes to the classification problem being faced. The initial values of the weights used are those of the pre-trained network, except for the connections between the penultimate and last level whose weights are randomly initialized.

New training iterations are performed to optimize the weights with respect to the peculiarities of the new dataset (it does not need to be large). Fine-tuning can be done in two ways. One way is to freeze the weights of some layers and carry out new training cycles to modify the weights of the remaining layers. The concept of fixing the weights of the layers is defined as freezing of the layers. Generally, they are the first layers to be frozen as the first layers capture low level features. The greater the number of frozen layers, the smaller the required fine-tuning effort in terms of time and resources. In this case, the weights of the first CNN levels are frozen and the remaining parameters/weights are trained. The other way is to have the architecture re-train entirely on the new database. This method is called training from scratch. It is intuitive that the greater the number of frozen layers the lower the computational cost of training, so training from scratch is the most expensive form of computational training.

The pre-trained CNNs have a consolidated and optimized architecture for the database for which they were originally trained. Usually, the many values to be optimized are referred to as hyperparameter optimization. The process of optimizing the parameters during the training phase is certainly not trivial. Among the most recognized methods, there is the Stochastic Gradient Descent with Momentum (SGDM) that minimizes the loss function at each iteration considering the gradient of the loss function on the entire training dataset. The momentum term reduces the oscillations of the SGDM algorithm along the path of steepest descent towards the optimum. The momentum is responsible for reducing some noise and oscillations in the high curvature regions of the loss function generated by the SGD. A variant of the SGD uses training subsets called mini-batches, in this case a different mini-batch is used at each iteration. Simply put, the mini-batch specifies how many images to use in each iteration. The full pass of the training algorithm over the entire training set using mini-batches is one epoch. SGDM with mini-batch was used in this thesis. Another fundamental parameter of the training process is the learning rate that allows to set the learning speed of the training process with the level of improvement of the network weights. Conceptually, a high learning rate increases the speed of training execution by sacrificing the performance of the trained network, while a low learning rate will increase training times by optimizing the network weights with an increase in performance.. This parameter defines the level of adjustments of weight connections and network topology applied at each training cycle. A small learning rate permits a surgical fine-tune of the model to the training data, at the cost of a greater number of training cycles and longer processing times. A high learning rate permits the model to learn more quickly, but may sacrifice its accuracy caused by the lack of precision over the adjustments. This parameter is generally set to 0.01, but in some cases, it is interesting to be fine-tuned, especially when it is necessary to improve the runtime when using SGD. Table 4.3 shows the search space analyzed for the optimization of the parameters.

Parameter	Configurations
Training mode	Stocastic Gradient Descent with Momentum with mini-batch
Mini-batch size	{4, 8, 16, 32, 64, 128, 256}
Learning Rate	{0.01, 0.001, 0.0001}
Momentum coefficient	0.9
Epoch	Max 10 epoch if freeze some layers max 30 epoch if training CNN from scratch

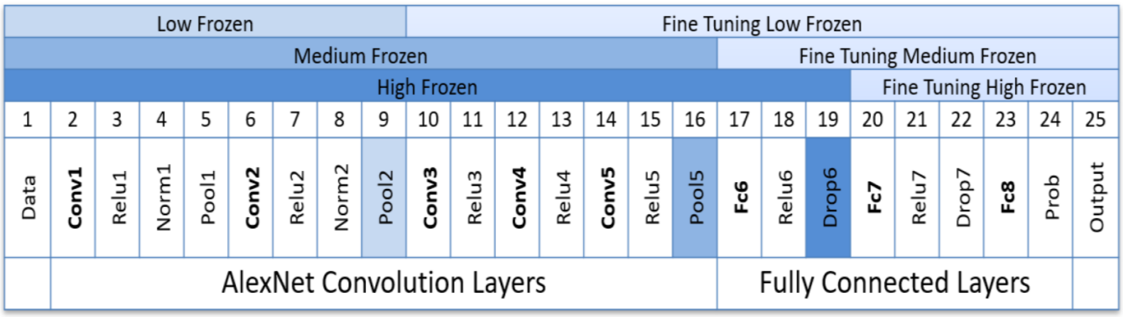
Table 4.3: Hyperparameters grid search.

The fine-tuning approach was analyzed for each CNN network, according to which, starting from the generic pre-trained CNN, the parameters are optimized by carrying out a training using the new database. In general, to implement fine-tuning, the last layer must be replaced to correctly define the number of classes to be discriminated. In this case the problem analyzed turns out to be binary. The four CNNs AlexNet, SqueezeNet, ResNet18 and GoogleNet, have been analyzed, training them both in scratch mode and with fine-tuning considering three different depths of freeze. Table 4.4 shows the three freeze levels chosen for each CNN.

CNN name	Total Layers	Low Frozen	Medium Frozen	High Frozen
AlexNet	25	9	16	19
SqueezeNet	68	11	34	62
ResNet18	72	12	52	67
GoogleNet	144	11	110	139

Table 4.4: Number of frozen layers at different levels and for the CNN analyzed.

As an example, Figure 4.6 shows the 25 layers that make up the AlexNet CNN with the graphic overlay of the three frozen levels chosen to perform the fine-tuning. The first level called low frozen indicates that the weights of the first nine layers of CNN AlexNet are fixed with the values of the network pre-trained on the original ImageNet database. The fine-tuning in this case consists in applying training cycles to CNN by changing the weights of the remaining layers at each iteration, i.e., from layer 10 to the last. In a very similar way to the first level, the medium frozen and high frozen levels were taken into consideration, which, respectively, fix the weights of the pre-trained AlexNet network up to layer 16 and layer 19; the fine-tuning in these two cases is carried out by modifying the weights of the last 9 layers and 6 layers. As described, the three selected fine-tuning levels are analyzed by iterating on the learning rate values {0.01, 0.001, 0.0001} and on the various batch sizes considering a maximum of 10 epochs.



Low Frozen									Fine Tuning Low Frozen															
Medium Frozen																Fine Tuning Medium Frozen								
High Frozen																			Fine Tuning High Frozen					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Data	**Conv1**	Relu1	Norm1	Pool1	**Conv2**	Relu2	Norm2	Pool2	**Conv3**	Relu3	**Conv4**	Relu4	**Conv5**	Relu5	Pool5	**Fc6**	Relu6	Drop6	**Fc7**	Relu7	Drop7	**Fc8**	Prob	Output
	AlexNet Convolution Layers															Fully Connected Layers								

Figure 4.6: Example diagram of the three levels of freezing of the weights referred to the AlexNet layers.

4.5 Experimental results

This section reports the results obtained for the classification of fluorescence intensity in the positive/negative classes of the HEp-2 images. The traditional binary classification strategy is pitted against recent pre-trained CNNs.

The binary classification with traditional pipeline analyzed different pre-processing (see Table 4.1) in order to emphasize the feature extracted (see Table 4.2). The latter were chosen from three families (intensity, shape, texture) and considering four different quantization levels for a total of 108 features. The LDA technique allowed to select the most suitable features for the problem of discrimination between positive and negative HEp-2 images. The intensive analysis of the preprocessing functions was conducted, aimed at maximizing the AUC performance. Several different configurations have been evaluated since some functions can be used with different parameters (e.g. for the median filter the convolution mask, sized 5x5, 7x7, 9x9, 11x11 have been analyzed). Furthermore some pre-processing pairs were evaluated. Therefore, the process analyzed 40 different pre-processing. The classification is done by SVM with a Gaussian kernel tuned by "grid-search" with 208 grid-points. The division of the images was carried out with a 5-fold cross validation and for each of the 5 iterations, training, validation, and testing were performed. As mentioned, images belonging to a specific well cannot appear in training and in test for each iteration, in order to avoid bias problems.

Four PC with a 3,4 GHz Intel i7 CPU were used for the analysis; the overall computation time is about 13 days. To optimize the execution time, the implementation was performed in C/C++ code with the IDE visual Studio 2010 and using high-performance libraries such as OpenCV (http://opencv.org) and LibSVM (https://www.csie.ntu.edu.tw/~cjlin/libsvm/).

In terms of performance, the system showed a sensitivity in the recognition of positive images equal to 92.9%, while with regard to the ability to identify the negatives, the system showed a specificity of 70.5%. The area under the curve value obtained was 0.914 and the accuracy value obtained was 87%. These results are compared with the CNN-based strategy and with other methods developed in the literature in the Table 4.10. The best performance with the traditional pipeline was obtained with the median filtering with 7x7 kernel among the forty preprocessing while the selected features were twelve and are listed in Table 4.5. The best SVM parameters are C = 64 and γ = 0.25.

#	Features name	Quantization
1 - 2	Entropy of HOG	256 & 128
3 - 4	Ratio of the standard deviation to the mean value	64 & 32
5 - 6	Entropy	256 & 128
7	Compactness	256
8 – 9 - 10	Mean value	128 & 64 & 32
11 - 12	Inertia of co-occurrence matrix	256 & 128

Table 4.5: List of selected features with indication of the quantization of gray levels.

As expected, the selected feaures are mostly based on intensity, no feature is of the "shape" type precisely because the discrimination does not occur between the different patterns but between positives and negatives.

Regarding the second method the classifying power of pre-trained CNNs was investigated with different training modalities; three levels of freezing weights and scratch. Additionally, the use of pre-trained CNNs layers to extract features coupled with a linear SVM classifier was also evaluated. The strategy used for the training-validation-test chain was the 5-fold validation considering the specimens. In this way, as mentioned before, performance bias problems are avoided. To increase the number of training examples, data augmentation was applied. In particular, an increase for image rotation at angles of 20°was achieved; overall, a multiplication of the data by a factor of 18 was obtained. Data augmentation is a very effective practice in deep learning optimization. The effect of this data augmentation was valued quantitatively in terms of performance.

The training phase was optimized considering the AUC as a measure of merit. HEp-2 images have been resized to 227 x 227 (for AlexNet and SqueezeNet networks) and 224 x 224 (for GoogLeNet and ResNet18 networks) to be provided as input to CNN; no

preprocessing has been applied to the image. CNN networks want a 3-channel image input, for this reason, it has evaluated the results using both RGB IIF images and only the green channel and replicating it on R and B channels. The results favored the second choice, in this configuration all the analyses reported below were carried out.
The training that provided the best performances required a calculation time of approximately 28 h for all 5-folds (using a 3.4 GHz Intel i7 CPU). The implementation for both CNNs and linear SVM) was performed in Matlab 2019 and Matlab 2020 (MathWorks, Natick, Massachusetts, USA). Table 4.6 shows the calculation times, in terms of intervals obtained by varying the configuration hyperparameters (mini-batch, learning rate, epoch, etc.) necessary for the training of the various CNNs. The configurations in which the CNNs were analyzed as feature extractors coupled with the SVM classifier required training times within the interval (0.57–13.6) hours.

CNN name	Training time in hours (min – max)			
	High frozen	Medium frozen	Low frozen	Scratch
AlexNet	(3.42 – 9.57)	(4.33 – 12.58)	(4.69 – 17.24)	(5.65 – 26.3)
SqueezeNet	(6.26 – 10.22)	(7.43 – 14.15)	(11.93 – 27.96)	(12.56 – 36.57)
ResNet18	(4.28 – 9.94)	(4.51 – 10.51)	(4.91 – 11.52)	(5.86 – 18.12)
GoogleNet	(4.02 – 8.26)	(4.13 – 16.32)	(4.27 – 18.54)	(5.37 – 24.4)

Table 4.6: CNNs trainings times.

Table 4.7 shows the best results obtained, in terms of AUC, from the four CNNs analyzed. The results are shown, considering the three levels of freeze of the pre-trained networks' weights and considering the retraining from scratch. The last two columns of Table 4.7 show the results obtained by performing a data augmentation, respectively, with the low frozen and with the scratch.

CNN name	High frozen	Medium frozen	Low frozen	Scratch	Low frozen + Data Augm.	Scratch + Data Augm.
AlexNet	97.20%	97.93%	97.82%	98.00%	98.08%	98.02%
SqueezeNet	97.96%	98.39%	98.55%	98.38%	98.63%	98.46%
ResNet18	97.27%	97.88%	98.11%	98.24%	98.33%	98.32%
GoogleNet	96.30%	96.78%	98.00%	97.96%	98.37%	98.20%

Table 4.7: Best AUC results of fine-tuning of pre-trained CNNs.

As expected, the training with less freezing allows a better adaptation of the CNN to the classification problem. As for the data augmentation procedure, the results show that the application of this procedure leads, in all cases, to a slight improvement in performance. For each of the four CNN networks, Table 4.8 shows the parametric configuration that obtained the best result. Figure 4.7 shows the ROC curve relating to the best configuration obtained, that is SqueezeNet with low frozen level and with data augmentation.

CNN name	AUC	Learning rate	Mini-batch	Epoch
AlexNet	98.08%	0.001	16	4
SqueezeNet	98.63%	0.001	16	6
ResNet18	98.33%	0.001	16	8
GoogleNet	98.37%	0.01	32	7

Table 4.8: Best Hyperparameters.

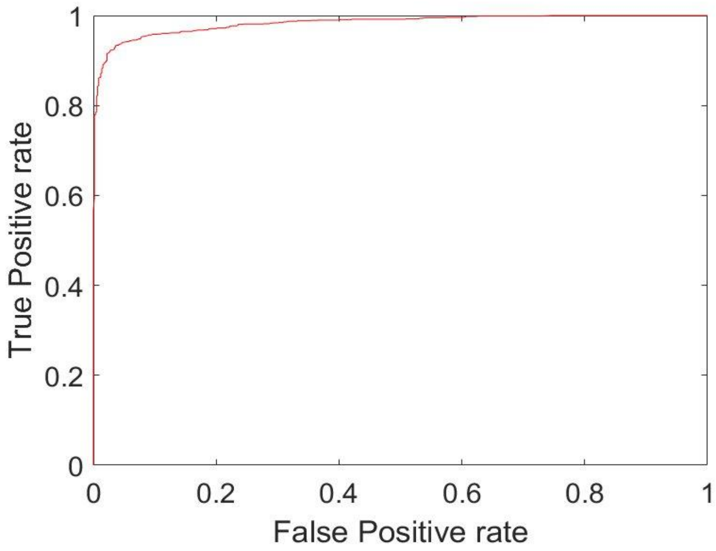


Figure 4.7: ROC curve obtained from the best fine-tuning configuration.

The method that uses the pre-trained CNNs to extract features and a linear SVM were evaluated. The linear SVM training phase was carried out by extracting the features, for the various layers, from all the training images. The SVM parameter C has been optimized on the validation set. The validated model was subsequently used for the test. Table 4.9 shows the best results obtained, in terms of AUC, for each pre-trained network analyzed and reports the layers that provided the best results.

CNN name	AUC	Best Layers
AlexNet	95.52%	‘Conv 5’
SqueezeNet	95.50%	‘Pool 10’
ResNet18	97.80%	‘Fc 1000’
GoogleNet	95.76%	‘Inception 3a output’

Table 4.9: Best AUC results obtained by CNN used as feature extractors.

The best result obtained with the CNNs used as a feature extractor coupled to the SVM classifier reached an AUC equal to 97.8%. This result was obtained with the ResNet18 CNN. The SVM parameter C associated with the best configuration was C = 0.0298. The results in Table 4.8, when compared with those of Table 4.6, show that, with the same CNN used, the fine-tuning method achieves better performances than the other method. Moreover, the results obtained with the various strategies were compared with other state-of-the-art works in Table 4.10.

Method	Images Dataset	Accuracy	AUC
Iannello [63]	914	89.5%	-
Bennamar [36]	1006	85.5%	-
Zhou [64]	1290	98.7%	-
Traditional strategy proposed	2080	87,0%	91,4%
CNN as feature extractor (best configuration)	2080	92.5%	97.8%
CNN with fine tuning (best configuration)	2080	93.9%	98.6%

Table 4.10: Performance comparison for HEp-2 intensity classification.

Unfortunately, not all of the authors of these works in Table 4.9 calculated the AUC value. While, from the comparison in accuracy it is verified that the pre-trained CNN fine-tuning method proposed turns out to be the second best. However, it should be noted that while Zhou et al. [64] aimed to maximize accuracy, the AUC was maximized in this thesis. For a more direct comparison with other methods that based their optimization on accuracy, it repeated the analysis using the latter index as a figure of merit, obtaining for the best configuration (also, this time with the SqueezeNet network), the result of 94.32% accuracy and an AUC equal to 98.34%. It should also be noted that the AIDA database used in this work is varied, both in terms of patterns contained (more than 20, both in single and multiple forms) and in terms of instruments and methods of acquisition (manufacturers of kits and instruments employed were different site-to-site); this makes it particularly difficult to classify. It must be said that the AIDA database used here is public while the other authors use a private database, so a more direct comparison on the same data is not possible.

The effectiveness of the fine-tuning technique was verified by comparing the performance of the same CNNs used as feature extractors (and coupled to SVM-type classifiers). With the same CNN used, fine-tuning has always given better results than the other method. The best performing network for the classification of fluorescence intensity in HEp-2 images was SqueezeNet.

It should be noted that the CNNs tested in the various configurations obtained performances all contained in a very small range (about 3% of AUC). This denotes very similar classifying abilities (at least on the specific problem) of the CNN analyzed.

Chapter 5 - ANA Pattern recognition

This Chapter presents the multiclass classification of the different staining patterns in HEp-2 images. The multiclass classification has been carried out on the positive Hep-2 images using public HEp-2 databases reported by expert personnel. The different methods developed are compared and then the best results are presented.

5.1 Introduction and related work

The second step in IIF image analysis is represented by the classification of the staining patterns (see subsection 1.2 “ANA pattern classification tree”). This second classification is multiclass type since different staining patterns can be found.

In accordance with what was presented in the previous chapter, both the traditional classification chain (with the analysis of preprocessing techniques, the extraction and selection of features, and supervised classification) and the CNNs were compared. In addition, for this multiclass classification, the aggregation techniques of multiple classifiers were also analyzed. The figure of merit used to evaluate the result was the Mean Class Accuracy. The results were analyzed on three available public databases and compared with the works published by the scientific community. Figure 5.1 shows some examples of staining patterns.

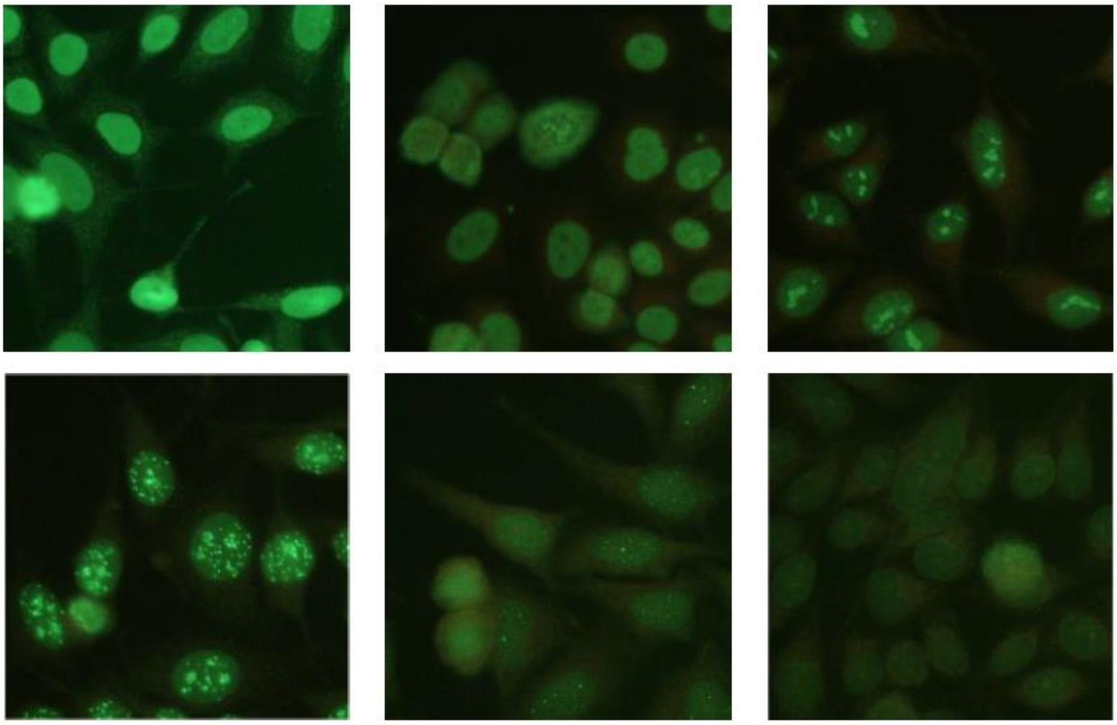


Figure 5.1: IIF images with different staining patterns (from left to right and from up to down: homogeneous, speckled, nucleolar, centromere, nuclear dots and nuclear membrane).

The problem of pattern classification attracted major attention among researchers with the contests ICPR 2012 [31] and I3A contest [34][35]. Ensafi et al [65] proposed a classification method where the SIFT and SURF features are extracted as the input features to learn a dictionary followed by Spatial Pyramid Matching (SPM) to provide the sparse representation of the input cell images. Then a Support Vector Machine (SVM) has been trained to classify the test images.

Manivann et al [66] presented a system to recognize such patterns, at cellular and specimen levels, in images of HEp-2 cells. The authors extracted sets of local features that are aggregated through sparse encoding. They used a pyramidal decomposition of the cell that consists in the central part and in the crown that contains the cell membrane. Linear SVMs are the classifiers used on the learned dictionary; specifically, they used 4 SVM the first trained on the orientation of the original images and the remaining three on the images rotated 90, 180 and 270 degrees respectively. Mean class accuracies for cell classification obtained on used test data sets were 87.1%. These were the highest achieved in the competition hosted by ICPR2014.

Nosaka et al [67] proposed Co-occurrence of Adjacent Local Binary Patterns (CoALBP) to extract textural features. Using linear Support Vector Machine (SVM), their method won the first prize in the contest ICPR2012.

One of the first researchers to use CNN in the classification of HEp-2 images was GAO [68]. The authors used a CNN with data augmentation. In particular, the network used had eight layers. Among them, the first six layers are convolutional layers alternated with pooling layers, and the remaining two are fully-connected layers for classification. They compared the method with traditional methods such as BoF and FV fisher vector. The DB used for the test is I3A Task 1 and MIVIA. The authors obtained 96.76% of mean class accuracy and 97.24% of accuracy.

Also, in the work of Li et al [69] the authors proposed CNNs for the solution of the classification problem. The method consists in the use of a CNN to construct a Histogram Pattern and through this a linear SVM is trained. CNN used was composed of 10 layers of which the first seven are convolutional layers while the last is a softamax layer for classification. The system has been trained and tested on data from the i3A Task-2 public database. In their work the authors showed that the strategy that uses SVM outperforms that of cell prevalence. Oraibi et al [70] used the well-known pre-trained CNN VGG-19 network to extract features and combined them with local features such as RIC-LBP (Rotation Invariant Cooccurence Local Binary Pattern) and JML (Joint Motif Labels). The combination of features was used to train RF classifiers (random forest).

Xu et al. [71] presented a method based on linear local distance coding in which, starting from local features, a local distance vector transformation was used by Euclidean distance. Finally, linear coding and max pooling were used, both on the local distance vector and on the local features. The concatenations were provided as examples to a linear SVM.

5.2 The methods implemented

The classification of Hep-2 patterns is faced with the classic classification chain and with the use of CNNs. In both cases, the classification is performed on the cells and not on the entire image. The whole image approach was experimentally tried with CNNs, but the results were unsatisfactory compared to the classification based on single cells. The cells are extracted thanks to the segmentation process seen in Chapter 3. In the case of the I3A task1 database, the entire images are not provided but only the boundary boxes of the cells. The classification of the entire image is therefore carried out by prevalence of the classified cells.

In line with the scientific community, the classification is conducted on six or seven patterns and evaluated with the Mean Class Accuracy (see subsection 2.4 “CAD system evaluation”) using the public databases described in Chapter 1.

The leave-one-specimen-out (LOSO) cell cross-validation technique is used to exploit the highest possible number of patterns during the training phase, without invalidate the results. The method consists in leaving out all cells belonging to the same specimen, rather than leaving out a single cell for the construction of the training set; cells of the same specimen are similar in terms of the average intensity and contrast, and introduce bias.

The classification of each segmented cell is analyzed with the traditional pipeline based on five phases:

1 Image pre-processing;
2 feature extraction;
3 feature reduction;
4 classifier training.

Similarly to the analysis conducted in the previous Chapter, the traditional process is compared with the use of recent CNNs. In particular, the method with CNNs uses the most popular "pre-trained" CNNs such as "AlexNet", "Squeezenet", "Resnet18", "Googlenet” applying the feature extraction strategy from the pre-trained CNN layers. In this way the classification pipeline is made up of two phases:

1 feature extraction from CNN layer;
2 classifier training.

The fine-tuning strategy is not explored as there is not enough data for some classes. The green component of the RGB is the one that makes the most sense to use, in the case of pre-trained CNNs, the green channel is duplicated in the R and B channels.

The conventional way to deal with a multi-class problem is to decompose an *N*-class problem into a series of two-class problems: binary approach [72][73]. The two approaches commonly used are the One-Against-One (OAO) and One-Against-All (OAA) techniques. Figures 5.2 and 5.3 show the diagrams of the two approaches OAA and OAO. The OAA approach represents the earliest and most commonly used multiclass approach and involves the division of an *N* class dataset into *N* two-class cases; OAA uses N classifiers with the i-th one separating class i from all the remaining classes. In order to assign a class to a new sample, it is necessary to evaluate the output for all binary classifiers and, usually, choosing the pattern relative to the classifier that returned the highest classification value. The problem with the participating to the decision is assumed to be equally reliable, which is rarely the case. The OAO approach on the other hand involves constructing a classifier for each pair of classes resulting in $N(N-1)/2$ classifier; the OAO approach is more computationally intensive. When applied to a test point, each classification gives one vote to the winning class and the point is labeled with the class having most votes. One drawback of OAO method, however, arises is that when the results from the multiple classifiers are counted for the final decision without considering the competence of the classifiers.

To overcome the limitation of OAA and OAO schemes, a third approach concerns the use of a K-Nearest-Neighbords (KNN) classifier that exploits all the output of binary classifiers developed in OAA and OAO. In KNN classification, an object is classified by a majority vote of its neighbors, with the object being assigned to the most voted class among its K nearest neighbors (K is a positive integer). Obviously the K parameter can not be greater than the number of examples M of a class. The best choice of K depends upon the data; generally, larger values of K reduce the effect of noise on the classification, but make boundaries between classes less distinct. Usually the order of magnitude to be assigned to K is $M^{1/2}$.

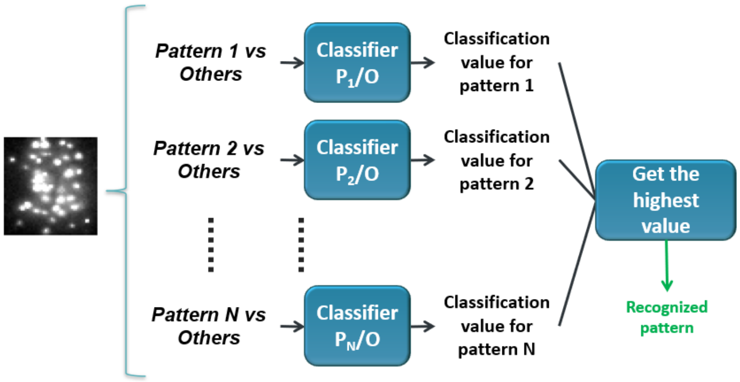


Figure 5.2: Diagram of the OAA approach classification with N binary classifiers. This approach associates the pattern with the class whose classifier returned the highest value.

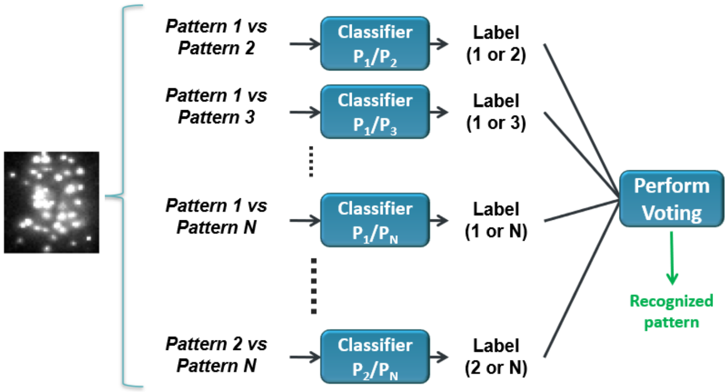


Figure 5.3: Diagram of the OAO approach classification with N(N-1)/2 binary classifiers. This approach associates the pattern with the class that got the most votes.

Figures 5.4 shows the diagrams of the KNN approach.

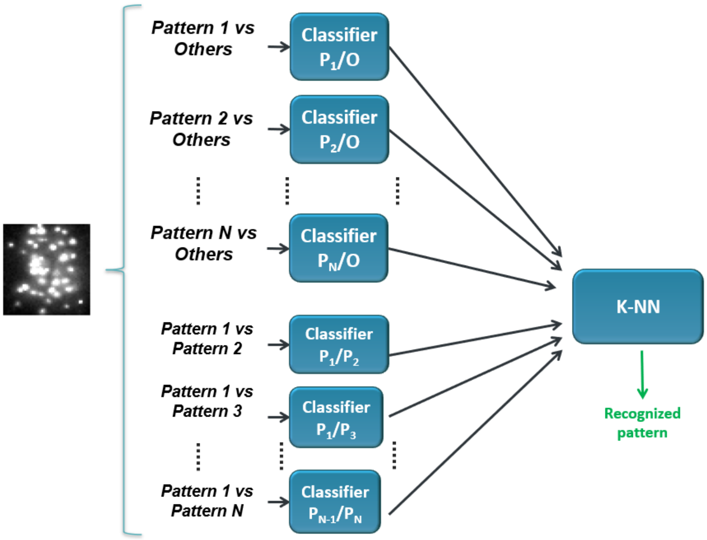


Figure 5.4: Diagram of the KNN classifier using the output of both the OAA and OAO approaches.

Summarize, three different strategies for multiclass classification are implemented and compared:

1 One-Against-All (OAA);

2 One-Against-One (OAO);

3 K-Nearest-Neighbords (KNN) classifier using OAA and OAO schemes.

5.3 Pattern classification with traditional approach

The classification of six or seven Hep-2 staining patterns is faced with the decomposition into six or seven dichotomous classifiers in the case of the OAA strategy, and in fifteen or twenty-one binary classifiers in OAO. For each of these binary classifiers, in a similar way to the intensity classification seen in section (4.3), the traditional classification chain is implemented. All set of processes consists of pre-processing, segmentation of the ROIs (except for I3A task1 dataset), feature extraction and reduction, and classification.

The segmentation process is the one implemented with the Hough transform and active countourn. In the case of the I3A task1 database, the segmentation is not necessary as the cells are provided via boundary box.

The preprocessing phase in a pattern recognition task is very important, and surely affects the next phase of feature extraction. In this thesis the preprocessing phase has been differentiated for each dichotomous classifier implemented. For this purpose, starting from a set of functions, analyzed and reported in Table 5.1, many combinations of different types of pre-processing have been analysed. The pre-processing functions analyzed are the same as those proposed in the automatic search of the segmentation seen in paragraph 3.3. Sixty different pre-processing have been evaluated since some functions can be used with different parameters (e.g. for the median filter the convolution mask, sized 3x3, 5x5 and 7x7 have been analyzed) and some pre-processing pairs were evaluated.

Pre-processing	Abbreviation	Description
Nothing	Nt	It does not apply any image processing.
Contrast normalization	Cn	Operate on histogram, linearly remapping the intensity values so that 1% of data is saturated at low and high intensities.
Equalization	Eq	Operate on histogram, remapping the intensity values to increase the global contrast by distributing the pixel values uniformly over the entire range of possible values.
CLAHE	Ch	Contrast Limited Adaptive Histogram Equalization, operate on small regions histograms rather than on the entire image, each processed region is combined with adjacents ones using bilinear interpolation.
Gaussian filter	Gs	Applies convolution filtering with Gaussian kernel.
Median filter	Md	Applies convolution filtering with Median kernel.
Morphological filters	Dl / Er / Op / Cl / Fs	Applies a morphological operation of dilation / erosion / opening / closing / FAS (Filter Alternate Sequential) on the grayscale images with a chosen structuring element.
Anisotropic diffusion filter	An	Applies the anisotropic diffusion filter which, based on the local image content, reduces image noise without blurring the edges.
Bilateral filter	Bl	Applies the bilateral filter which, based on the local image content, reduces image noise without blurring the edges.

Table 5.1: Preprocessing functions analyzed.

The feature extracted were chosen from intensity, shape and texture families (see appendix C for the features list) and considering four different quantization levels for a total of 108 features. This feature vector is the same as described in section 4.3 obtained with intensity quantizations at 256, 128, 64 and 32 gray levels. Table 5.2 lists the 27 features divided by category. To reduce the dimesion of features vector and to choose the subset of features that most discriminate certain patterns, the Linear Discriminant Analysis (LDA) is used. In fact by analyzing the cells patterns, it appears clearly that differences between classes are mainly based on the presence and distribution of bright/dark structures. Therefore, some feature sets may be more discriminating for some patterns than others.

The subset of selected features is then used with the SVM classifier with Gaussian kernel. The two parameters for tuning the RBF SVM are C and γ and were obtained with the “grid-search” method. The analyzed values for the C e γ were:

$$C = 2^{-5}, 2^{-4}, \dots , 2^{-10} \qquad \gamma = 2^{-10}, 2^{-9}, \dots , 2^{-2}$$

the analyzed grid had sizes equal to 16 * 13, for a total of 208 grid-points.

Features type	Features number	Features name
Intensity	6	mean value, standard deviation, ratio of the standard deviation to the mean value, entropy, skewness, and kurtosis
Shape	12	Area, perimeter, convex area, mean radius, standard deviation of radius, ratio of the standard deviation to the mean value, maximum radius, anisotropy, entropy of the contours gradient, fractal index, eccentricity, and circularity
Texture	9	Contrast, convex deficiency, roundness, compactness, solidity, inertia of co-occurrence matrix, entropy of histogram of oriented gradients (HOG), entropy of histogram of amplitude gradients (HAG), and Euler's number

Table 5.2: List of features divided by type.

For all pipelines (six or seven for OAA and fifteen or twenty-one for OAO) of binary classification, the choice of pre-processing, features and SVM parameters was performed automatically, using the AUC as a figure of merit.

The basic intuition behind this decomposition approach is that, instead of using a single pre-processing and a set of features to discriminate each class, it is better to combine a set of different and complementary processes. Each pipeline can have a pre-processing and a set of specialized features different from the other pipelines.

Once all the binary classification chains are trained, the generic ROI of the specimen to be tested is simultaneously processed by six/seven (OAA) or fifteen/twenty-one (OAO) classification pipeline obtaining as many outputs. Figure 5.5 shows the flow of operation in OAA strategies with six patterns to classify. Each pipeline can conceptually have a different pre-processing, a different set of features and different values of the SVM parameters from the other pipelines.

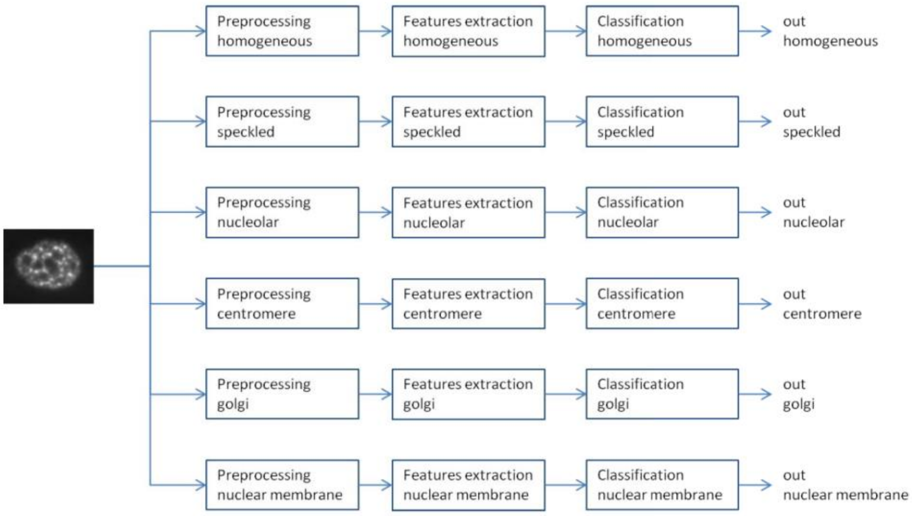

Preprocessing homogeneous
Features extraction homogeneous
Classification homogeneous
out homogeneous
Preprocessing speckled
Features extraction speckled
Classification speckled
out speckled
Preprocessing nucleolar
Features extraction nucleolar
Classification nucleolar
out nucleolar
Preprocessing centromere
Features extraction centromere
Classification centromere
out centromere
Preprocessing golgi
Features extraction golgi
Classification golgi
out golgi
Preprocessing nuclear membrane
Features extraction nuclear membrane
Classification nuclear membrane
out nuclear membrane

Figure 5.5: Flow of operations in OAA strategie with six patterns to classify.

5.4 Pattern classification with CNNs

In this work, the classification of Hep-2 cells is also addressed with the use of pre-trained CNNs (see appendix A for details). It has been demonstrated that pre-trained CNN architectures can play an important role as feature extractors and allow high classification performance. As explained in Chapter 4, the classification of fluorescence intensity using CNNs as feature extractors has achieved excellent results and the difference with the fine-tuning strategy in terms of AUC was just under one percentage point.

The effectiveness of feature extraction both from the segmented cells (internal) and from the boundary boxes containing the cells was evaluated. Performance improvement in using the segmentation mask rather than the bounding box has been verified.

In oder to increase the number of training examples, a data augmentation was made. In particular, an increase for image rotation at angles of 20° was achieved; overall, a multiplication of the data by a factor of 18 was obtained.

In order to reduce the intensity variability of the ROIs a contrast normalization was performed. Then, the sub-images containing the cells have been appropriately rescaled to acquire the correct dimensionality for the network entrance (227 x 227 for AlexNet and SqueezeNet CNNs and 224 x 224 for GoogLeNet and ResNet18). The vector of feature extracted from the pre-trained CNN are used in the training of linear support vector machine. It has chosen linear SVM, that has only one parameter to tune, because the size of the feature vector is large. The search for the penalty parameter “C” of the error term is carried out in the range [10^{-6}, $10^{-2.5}$] where twenty equidistant values on a logarithmic scale were analyzed.

Different layers of the CNNs have been evaluated as feature extractors and the best configuration has been identified using AUC figure of merit for each binary classifier. In accordance with the OAA and OAO strategy six or seven and fifteen or twenty-one binary SVMs are trained with the feature extracted from the different layers (belonging to the four pre-trained CNNs).

Figure 5.6 shows the flow of training adopted for cellular classification in the explanatory case OAA. The generic segmented cell is decomposed by the multilayer neural network to obtain the features used as inputs of the six linear SVMs. Figure 5.7 highlights the iterative strategy of the training-tuning phase. The procedure is iterated on the various CNNs and related layers, then on the all binary classifiers derived from strategies OAA and OAO. Finally, the SVM is tuned through grid seach and the evaluation of the classification is performed with the AUC.

Once all the binary classification chains are trained, the generic ROI of the specimen to be tested is simultaneously processed by six/seven (OAA) or fifteen/twenty-one (OAO) classification pipeline obtaining as many outputs. Each pipeline can conceptually have a different set of features (from different layers or pre-trained CNN) and different values of the SVM parameters from the other pipelines.

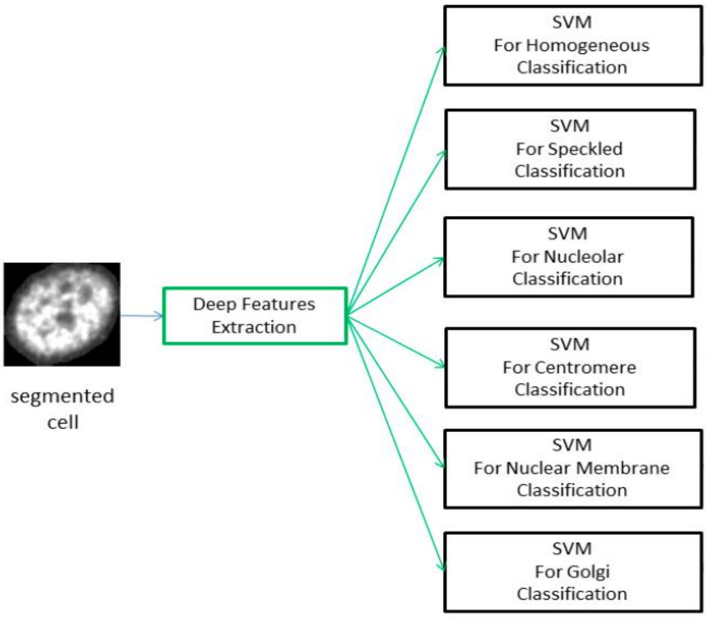

segmented cell
Deep Features Extraction
SVM For Homogeneous Classification
SVM For Speckled Classification
SVM For Nucleolar Classification
SVM For Centromere Classification
SVM For Nuclear Membrane Classification
SVM For Golgi Classification

Figure 5.6: Flow of operations in OAA strategie with six patterns to classify.

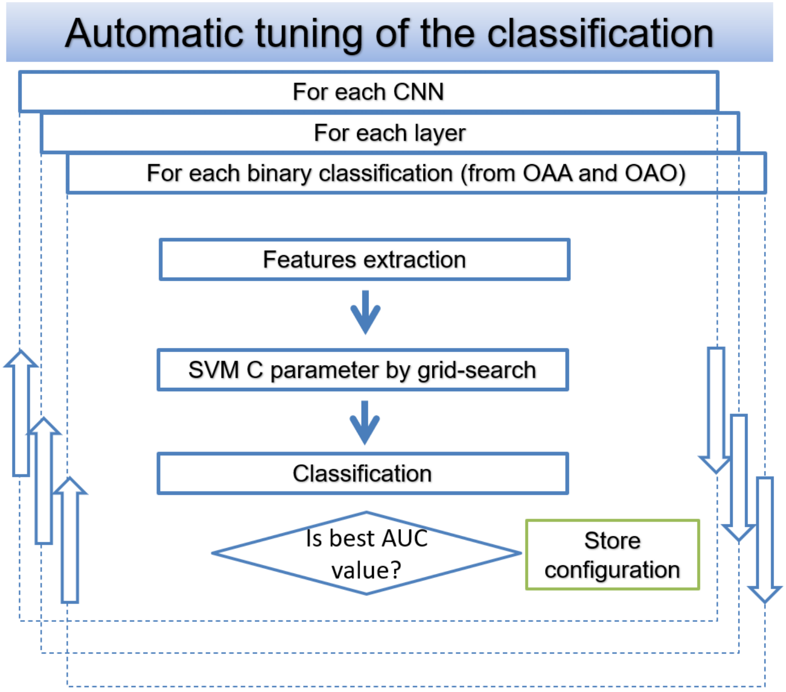

Automatic tuning of the classification
For each CNN
For each layer
For each binary classification (from OAA and OAO)
Features extraction
SVM C parameter by grid-search
Classification
Is best AUC value?
Store configuration

Figure 5.7: Flow chart of the iterative method used for the training-tuning phase in CNN feature extraction strategie.

5.5 Experimental results

In this section the experimental results of the discussed approaches for the automatic classification of staining patterns in Hep-2 images are presented. The image classification is achieved by means of the cell classification. The rate of presence of the individual patterns, within the image, is evaluated and the generic image is associated with the pattern that has a higher rate. The segmentation applied is the one based on active contours discussed in Chapter 3. The LOSO procedure was used in this work. In the LOSO strategy, each time all cell images (and the relative images obtained by data augmentation) from one of the specimens available are used for testing, the rest are used for training. Since cells belonging to the same image/specimen have a very similar informative contribution, in order not to distort the performance result on the test, if cells of an image are used in training, no cell of the same image should be present in the test. The multi-pattern classification has been faced with the decomposition in dichotomous processes and in particular with the OAA and OAO strategies. A third approach concerns the use of a K-NN classifier that has both the outputs of the OAA and OAO as inputs.

All binary classifiers are training with two approaches. The first approach mirrors the classical supervised classification chain in which a pre-processing and a set of features is searched and finally the classifier is tuned. Each classification pipeline is optimized by looking for the best pre-processing (among sixty taken into consideration), selecting a sub-set of features (among one hundred and eight extracted) with the LDA features selection, and tuning an SVM with RBF kernel with the grid-search. The optimization takes place using the AUC as a figure of merit and the final result of all the pipelines is a differentiated system both in terms of pre-processing and features for each binary classifiers (resulting from the OAA and OAO strategies).

The second approach investigated the use of pre-trained CNNs layers to extract features coupled with a linear SVM classifier. Here too, the optimization is carried out for each binary process by maximizing the AUC and the result is a differentiated system in terms of set of features (chosen from different layers of different CNNs) for each binary classifier (resulting from the OAA and OAO strategies).

Four PC with a 3,4 GHz Intel i7 CPU were used for the analysis. In the first approach, to optimize the execution time, the implementation was performed in C/C++ code with the IDE visual Studio 2010 and using high-performance libraries such as OpenCV (http://opencv.org) and LibSVM (https://www.csie.ntu.edu.tw/~cjlin/libsvm/). In the second, the implementation for both CNNs and linear SVM was performed in Matlab 2019 and Matlab 2020 (MathWorks, Natick, Massachusetts, USA). Table 5.3 summarizes the data of image database used.

Regarding the use of the AIDA database, a subset of HEp-2 images was taken, considering six patterns: homogeneous, speackled, centromere, nucleolar, nuclear dots and nuclear membrane. The total number of images in the sub-set are 220 from 71 wells. The patterns chosen are in line with those proposed in the other databases, furthermore the nuclear dots pattern has been taken into consideration to complete the list of patterns of relevant nuclear types (see subsection 1.2).

Database	List of patterns	Pattern number	Image number	Well number	OAA classifiers	OAO classifiers
MIVIA	{homogeneous, coarse speckled, fine speckled, centromere, nucleolar, cytoplasmic}	6	28	28	6	15
I3A task1	{homogeneous, speckled, centromere, nucleolar, nuclear membrane, golgi}	6	83	83	6	15
I3A task2	{homogeneous, speckled, centromere, nucleolar, nuclear membrane, golgi, mitotic spindle}	7	1008	252	7	21
AIDA (sub-set)	{homogeneous, speckled, centromere, nucleolar, nuclear dots, nuclear membrane}	6	220	71	6	15

Table 5.3: Salient data from public databases.

The sixty pre-processing and related segmentations are conducted for the MIVIA database, I3A task2 and AIDA (sub-set). The results of these sixty operations are the ROI extracted and saved with the boudary box. In the case of the I3A task1 dataset it is not necessary to perform the segmentation because the dataset is made up directly of the ROIs in the form of boundary boxes. In this case, a cell segmentation mask is applied to segment the cell with respect to the boudary box.

The average segmentation time for MIVIA, AIDA and I3A task2 images is just over 2 minutes, while the average pre-processing time for an image is about 3-4 seconds.

This involves an overall time to extract the ROIs:

*computation time to extract all ROIs = 1256 (28 + 220 + 1008) images * 2,2 min. =*

=~ 2763 min. =~ 46 hours

For all sixty pre-processing each database is subjected to the LOSO strategy to train and test the twenty-one/twenty-eight classification pipelines (six or seven OAA and fifteen or twenty-one OAO). The time to search for the parameters of the SVM with the grid search (208 points) is more expensive for the binary classifiers of the OAA method than those of the OAO method because the number of examples are greater considering that each pattern must be discriminated with all the others. But in the OAO case the binary classifiers to train are many more. With the LOSO method, for each of the sixty pre-processing, 28 training cycles are performed for the MIVIA database, 71 in the AIDA case, 83 for I3A task1 and 252 for I3A task2. Calculation times are shown in Table 5. 4, the optimization flow with the LOSO strategy, considering the sixty pre-processing, the LDA features selection and the SVM tuning with grid search is very expensive especially for the two databases with the highest number of images. In the case of I3A task2, to lower the computational calculation in each of the 252 training cycles only one image of the 4 available in the well was taken into consideration (the 252 tests instead consider the 4 images of the well).

Database	LOSO training time for traditional pipeline approach (in days)		
	Average time	Min time	Max time
MIVIA	2	2	3
I3A task1	11	6	15
I3A task2	79	52	83
AIDA (sub-set)	23	18	27

Table 5.4: Computation time for the traditional pipeline training with LOSO strategy.

The computation time for the second method based on linear SVM and feature extraction from pre-trained CNNs layers is shows in Table 5.5. The considerations already made for the first method they are also valid for the second, except that the layers of the four pre-trained CNNs are iterated instead of the preprocessing iterations. The iterations are in this case fourty, having chosen ten layers for each CNNs. The reduction of the computation time is mainly due to the fact that the chosen classifier is a linear SVM in this case with only the C parametron to search. SVM with a Gaussian kernel would cause the computation time to become excessive.

Database	LOSO training time for pipeline with CNN layer as feature extractor (in days)		
	Average time	Min time	Max time
MIVIA	1	0.5	2
I3A task1	4	3	6
I3A task2	14	12	17
AIDA (sub-set)	9	8	11

Table 5.5: Computation time for the pipeline training with CNN layer as feature extractor and LOSO strategy.

Parallelization is done explicitly by running the twenty-one or twenty-eight classification pipelines divided for the 4 PCs.

For each test resulting from the LOSO strategy, each cell is classified with the three strategies OAA, OAO e KNN. In the OAA scheme the classifier that produced the maximum output value is identified and the final association of the generic ROI is assigned to the relative class. The OAO approach chooses the most voted pattern. While the classifier KNN has been chosen, as it allows a simple multi-class implementation; this classifier, using examples belonging to the classes to be analyzed, associates the generic element with the class having the most examples close to it. The concept is to use all the outputs produced by the OAA and OAO strategies and transform them into a new feature vector for the KNN.

Tables 5.6 and 5.7 show the comparative performance obtained in terms of ACC and MCA. The first Table refers to the results obtained with the traditional process and the results are divided for each database analyzed for the three strategies OAA, OAO and KNN. Despite the high number of classificators used, the OAO strategy always has a lower performance than the OAA and KNN strategies. Another important observation is that the OAA method generally achieves the highest performances in terms of ACC, while the KNN method obtains the best performances in terms of MCA on three of four databases. This is an important point, in fact the ACC can be affected by the distribution of the various patterns while the MCA is more stable as it calculates the accuracy of each pattern and averages it. In this way, even if the classes are numerically unbalanced, the MCA gives a good estimate considering all the classes involved. Therefore, the method with the KNN is to be preferred over the OAA method. It must be said that the difference between the two OAA and KNN methods is however contained within a few percentage points. The database on which the OAA method has the highest performance is MIVIA, this database is the one with the fewest images available and therefore a mistake in the classification of an image results in a considerable leap in performance.

Database	Performance at image level for traditional pipeline approach (ACC / MCA)		
	OAA	OAO	KNN
MIVIA	**82.1% / 82.8%**	78.6% / 77.2%	78.6% / 79.4%
I3A task1	**80.4%** / 78.9%	78.2% / 77.1%	79.1% / **79.5%**
I3A task2	73.8% / 70.5%	71.5% / 67.4%	**79.4% / 71.9%**
AIDA (sub-set)	**72.4%** / 71.8%	69.4% / 67.2%	71.2% **/ 74.7%**

Table 5.6: Comparative performance at image level for traditional pipeline approach for OAA, OAO and KNN strategies.

Database	Performance at image level for CNN layer as feature extractor approach (ACC / MCA)		
	OAA	OAO	KNN
MIVIA	85.7% / 86.4%	82.1% / 82.2%	**89.3% / 89.7%**
I3A task1	94.0% / 91.5%	92.8% / 87.5%	**96.4% / 93.8%**
I3A task2	85.0% / **82.4%**	85.4% / 78.3%	**86.1%** / 80.9%
AIDA (sub-set)	89.1% / 88.3%	87.7% / 86.6%	**90.5% / 90.2%**

Table 5.7: Comparative performance at image level for traditional pipeline approach for OAA, OAO and KNN strategies.

Considering the second method based on the extraction of features through the layers of pre-trained CNNs, it can be noted that the OAO method did not achieve good results compared to the OAA and KNN strategies. It is the KNN method that almost alway achieves the greatest results in terms of ACC than MCA. Certainly to note is the high difference between ACC and MCA which is found in the I3A task2 database in both Table 5.6 and 5.7. This is due to one of the seven patterns to be classified: the mitotic pattern. This pattern is characterized by a fluorescence in the cells in mitosis and not in the other phases of the cell. Therefore, only a small percentage of all cells in an image are in the mitotic phase. This reason does not allow to identify a good classifier with the classification at the cell level. For this pattern, an approach based on the identification of cells in mitosis or an approach based on large regions of a Hep-2 image should be developed.

The comparison between the traditional method and the method based on the extraction of features with CNN is very much in favor of the second method, with performances that go beyond 10% improvement.

Table 5.8 shows the confusion matrices referred to each database, taking into consideration the best classification strategy (CNN as feature extractor and KNN).

Database	Confusion matrix									ACC	MCA
MIVIA		HO	FS	CS	CE	NU	CY			89.3%	89.7%
	HO	4	1	0	0	0	0		80%		
	FS	0	4	0	0	0	0		100%		
	CS	0	0	5	0	0	0		100%		
	CE	0	0	1	5	0	0		83,3%		
	NU	0	0	0	0	4	0		100%		
	CY	1	0	0	0	0	3		75%		
I3A task1		HO	SP	NU	CE	GO	ME			96.4%	93.8%
	HO	16	0	0	0	0	0		100%		
	SP	0	15	0	1	0	0		93.7%		
	NU	0	0	16	0	0	0		100%		
	CE	0	1	0	15	0	0		93.7%		
	GO	0	0	0	0	3	1		75%		
	ME	0	0	0	0	0	15		100%		
I3A task2		HO	SP	NU	CE	GO	ME	MI		86.1%	80.9%
	HO	47	2	0	0	0	2	2	88.7%		
	SP	2	46	0	2	0	0	2	88.5%		
	NU	0	0	45	0	4	0	1	90%		
	CE	0	5	0	46	0	0	0	90.2%		
	GO	0	1	2	0	7	0	0	70%		
	ME	2	0	0	0	0	18	1	85.7%		
	MI	5	1	0	0	0	1	8	53.3%		
AIDA (sub-set)		HO	SP	CE	NU	DO	ME			90.5%	90.2%
	HO	19	1	0	0	0	1		90.5%		
	SP	0	38	2	2	0	0		90.5%		
	CE	0	1	24	0	1	0		92.3%		
	NU	0	4	1	58	0	0		92.1%		
	DO	0	2	3	0	40	0		88.9%		
	ME	2	1	0	0	0	20		87.0%		

Table 5.8: Confusion matrices for each database with the best classification strategy.

Continuing the analysis of the results shown in Table 5.8, it sees that the highest performance values are achieved for the I3A task1 database. This is probably due to the fact that the segmentation is not performed as the cells are already supplied with the boundary box. In the other three databases, on the other hand, the segmentation is operated automatically and could lower the classification effectiveness compared to manual segmentations. Looking at the two databases MIVIA and AIDA, the effectiveness of the proposed method can be estimated at around 90% in both ACC and MCA. Even the I3A task2 database without considering the mitotic pattern would be close to 90%.

As for the patterns, it is clear from the confusion matrices that the homogeneous pattern is sometimes classified as speckled and membranous. In fact, especially for the fine speckled, the cells appear in a similar way to the homogeneous as happens for the membranous. Again, the speckled and in particular the coarse speckled have traits that resemble centromeric and nuclear dots.

The Table 5.9 shows the MCA values of the proposed method by comparing it with other methods proposed in the literature.

Database	Image number	Method	Training split	MCA
MIVIA	28	Nosaka [67]	Leave-One-Out	85.7%
		Best proposed method	LOSO	**89.7%**
I3A task1	83	Xu [71]	Train from 42 img. Test from 41 img.	85.4%
		Li [69]	LOSO	85.6%
		Best proposed method	LOSO	**93.8%**
I3A task2	1008	Oraibi [70]	LOSO	92.1%
		Manivann [66]	LOSO	89.9%
		Best proposed method	LOSO	**80.9%**
AIDA (sub-set)	220	Best proposed method	LOSO	**90.2%**

Table 5.9: Performance comparison for HEp-2 staining pattern classification.

As already explained, the proposed method is not competitive in the analysis of the I3A task2 database compared to other methods that address the problem not from the point of view of the single cell but considering regions of the image to identify cells in mitosis. For the MIVIA and I3A task1 database, on the other hand, the method proposed achieves excellent performance compared to the other state-of-art works. Finally, on the AIDA database there are no methods for comparison, but the performance remains in line with that obtained for MIVIA and I3A task1.

Conclusions

In recent years, the need to automate the analysis of IIF HEp-2 specimens has been established, in order to obtain a fundamental tool for the diagnosis of autoimmune diseases, and to avoid the subjectivity of human interpretation. For this purpose, in this thesis a complete and fully automatic CAD system has been implemented, which is able to classify the IIF images, in terms of fluorescent intensity and fluorescent pattern.
The system combines the following steps:

1 fluorescence intensity classification: this phase performs a categorization of the fluorescent intensity into positive/negative classes;
2 cells segmentation: this phase decomposes the input image, looking for the cells contained in it, without any a-priori knowledge about its intensity level or pattern;
3 pattern classification: this phase classifies the regions of interest identified in the segmentation phase by associating them with a class of autoantibody patterns.

In particular, the analysis of fluorescence intensity, for the positive/negative detection, is carried out using two main approaches:

1 traditional based on the supervised classification chain with pre-processing, extraction and selection of features and an SVM classifier with Gaussian kernel;
2 based on the performing pre-trained CNNs, both exploiting transfer learning with fine-tuning and using them as extractors of robust features coupled with linear SVM classifier.

The experimental results showed how the second approach has achieved excellent results, exceeding the best performance of the other approach by eight percentage points. In terms of AUC, the result obtained on the public AIDA database, by performing the fine-tuning with CNN "SqueezeNet", was 98.6%.
In order to address the segmentation, two other approaches were compared, the first took into consideration different pre-processing strategies, different thresholding techniques, and some post-processing operations. The second approach consists of three steps: the first step performs a pre-segmentation, the second uses the randomized Hough transform to find ellipses around the pre-segmented ROI, and finally the third step applies the active contours algorithm.
The performance of the proposed methods is analyzed with the DICE index, thanks to the use of manual segmentation masks provided for the MIVIA database and a subset of the AIDA database. In spite of the remarkable diversity of the patterns analyzed, the method based on the active contours achieves very similar segmentation results for the different patterns, demonstrating a good robustness. In AIDA the best result achieves an average index of 85.2%, while in MIVIA 87% one of the highest measures among the methods proposed in the literature.
Finally, the third step of the CAD developed concerns the pattern classification. This is a multiclassification problem, and involved the classification of six or seven patterns with the use of public databases (MIVIA, I3A task1, I3A task2, AIDA). In many

multiclass problems, it is more efficient to use a binary approach, implementing a classifier for each class for the discrimination process. In this thesis the binary decompositions with OAA and OAO strategies have been analyzed. Moreover a third approach based on the KNN classifier has been developed to act as a collector of all the outputs of the OAA and OAO strategies. For each binary classifier, in a similar way to what was done for the classification of fluorescence intensity, both the traditional classification chain (with pre-processing, extraction and selection of the characteristics and classification with SVM with Gaussian kernel), and CNNs pre-trained as feature extractors coupled to a linear SVM, have been implemented. In this classification process, the method based on pre-trained CNN has exceeded the traditional method by about 10% in terms of MCA. The best results were obtained with the KNN which synthesized all the outputs produced by the binary SVMs built with the OAA and OAO strategies.

The performances obtained on the four public databases were compared with other works in the literature. On the MIVIA database an MCA of 89.7% was reached, in I3A task1 of 93.8%, in I3A task2 of 80.9% and finally on AIDA 90.2%. while the highest result was achieved in the I3A task1 database where it was not necessary to implement the segmentation, the lowest result was found in I3A task2. In this database, a particular mitotic pattern challenges the strategy based on the predominance of cells in an image. In fact, the mitotic pattern in question occurs only in cells in the mitotic phase and therefore only on a small subset of the cells that constitute an image. For this reason, a strategy that differentiates cells in mitosis or that takes into account large frames of the entire image should be investigated. In the other three databases the MCA values obtained are certainly comparable with the state of the art.

The high performance obtained from the various phases of the CAD developed in this thesis allow to state that this system can effectively support the diagnosis of autoimmune diseases, for example as a "second reader" or by applying it to the "screening of autoimmune diseases".

Scientific Results

Articles in international journals:

1. “V. Taormina et al”. **Performance of Fine-Tuning Convolutional Neural Networks for Hep-2 Image Classification.** Applied Sciences, 10, 6940, DOI: 10.3390/app10196940, 2020.

2. “D. Cascio, V. Taormina, G. Raso”. **Deep CNN for IIF Images Classification in Autoimmune Diagnostics**. MDPI AG APPLIED SCIENCES, Vol. 9, Pag. 1618, DOI: 10.3390/app9081618, 2019

3. “D. Cascio, V. Taormina, G. Raso”. **An automatic HEp-2 specimen analysis system based on an active contours model and an SVM classification**. MDPI AG APPLIED SCIENCES, Vol. 9, Pag. 307, DOI: 10.3390/app9020307, 2019

4. “D. Cascio, V. Taormina, G. Raso”. **Deep Convolutional Neural Network for HEp-2 fluorescence intensity classification**. MDPI AG APPLIED SCIENCES, Vol. 9, Pag. 408, DOI: 10.3390/app9030408, 2019

5. “G. Cavallaro et al”. **Sedimentation of halloysite nanotubes from different deposits in aqueous media at variable ionic strengths.** Elsevier B.V. Colloids and Surfaces. A, Physicochemical and Engineering Aspects, Vol. 576, Pag. 22-28, DOI: 10.1016/j.colsurfa.2019.05.038, 2019

6. “L. Vivona, D. Cascio, V. Taormina, G. Raso”. **Automated approach for indirect immunofluorescence images classification based on unsupervised clustering method.** IET Computer Vision, Institution of Engineering and Technology, Pag. 989-995, DOI: 10.1049/iet-cvi.2018.5271, 2018

Proceedings of international congresses:

7. “V. Taormina et al". **Hep-2 intensity classification based on deep fine-tuning**, 7th International Conference on Bioimaging, BIOSTEC BIOIMAGING 2020, Pag. 143-149, ISBN: 978-989-758-398-8, 24-26 Feb.2020, Valletta, Malta.

8. 11. “S. Sorce et al”. **A REST-based Framework to support Non-Invasive and Early Coeliac Disease Diagnosis**, 20th International CompSysTech '19, Pag. 207-212, ISBN: 978-1-4503-7149-0, 21-22 June 2019, Ruse, Bulgaria.

9. 12. “D. Cascio et al”. **A Microcalcification Detection System in Mammograms based on ANN Clustering,** 2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC), Pag. 1-4, ISBN: 978-1-5386-8494-8, 10-17 Nov. 2018, Sydney, Australia.

10. “D. Cascio, V. Taormina, G. Raso”. **Automatic Segmentation of HEp-2 Cells Based on Active Contours Model**, 3rd International Conference on Biomedical Imaging, Signal Processing, ICBSP 2018, Pag. 41-45, ISBN: 978-1-4503-6477-5, October 2018, Bari, Italia.

Abstract in conference:

11. “D. Cascio, V. Taormina”. **Analisi di test di Immunofluorescenza indiretta per il supporto alla diagnosi di malattie autoimmuni basata su deep learning**, 105° Congresso Nazionale della Società di Fisica, 2019.

Appendix A - Classifiers

A.1 Support Vector Machine

One of the classification systems used in this work is the SVM (Support Vector Machine) classifier proposed in the 90s by the authors cortes and Vapnik [74]. SVM is one of the most used methods in supervised machine learning, the algorithm has been widely used in the literature for both classification and regression, representing one of the most robust prediction method [75] [76] [77].

In binary classification, the goal is the automatic differentiation between two classes of belonging, in this context the problem mainly consists in finding a criterion to distinguish the elements of two disjoint sets of sample points, defined as patterns or training data, in the space of the features.

Since the patterns are generally represented by points in the space R^n, the problem becomes discriminating between two finite sets of points, A and B, in the n-dimensional space, R^n, through a hyperplane or a non-linear separating surface. In the case of using a hyperplane, the problem is defined as linear separability and can be summarized as follows:

are the set A consisting of m points, $a_i \epsilon R^n$, $i = 1, \dots m$ and represented by the matrix A ϵR^{nm}, with $A_i = a_i^T$ i-th row; the set B consisting of k points, $b_l \epsilon R^n$, $l = 1, \dots k$ and represented by the matrix B ϵR^{kn}, con $B_l = b_l^T$ l-th row; with $A \cap B = \emptyset$.

The hyperplane

$$P \triangleq \{x \, \epsilon R^n \mid w^T x = \xi \}$$

represents a separation hyperplane between sets A and B if:

- $A_i w > \xi \qquad \forall \, i = 1, \dots m;$
- $B_l w < \xi \qquad \forall \, l = 1, \dots k.$

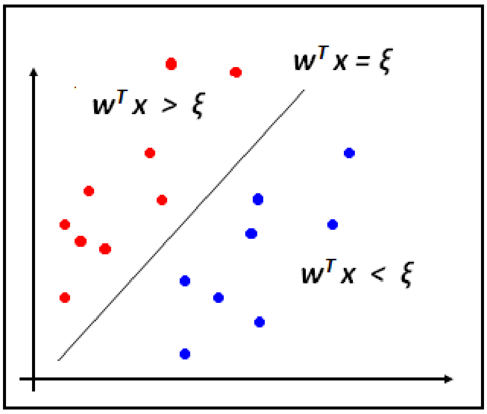


Figure A.1: Example graph of a linear separability problem.

For non-linearly separable problems, the classifier must look for that linear surface that separates the two classes as best as possible, or the possibility of exploiting non-linear surfaces allows for better results. There are different formulations of SVMs: binary or multi-class classification, or regressions, the so-called one-class SVMs etc; for the purpose of the discussion the SVM for binary classification is taken into consideration.

In binary classification, a linear SVM classifier looks for a hyperplane that separates the training data into two classes with the largest classification margin, which allows for a good degree of generalization. When the data set is not separable through a linear hyperplane, the SVM finds that hyperplane that represents a good compromise between the classification error and the ability to generalize. However, the real potential of SVM lies in its non-linear extension, in which it can find a non-linear classification surface to separate classes.

Linear SVM and linearly separable sets

Taking up the linear separability problem from the point of view of the SVM, starting from the points $\{a_i\}$, i = 1, … *m*, of A e $\{b_l\}$, l = 1, … *k*, of B, these can be represented as follows:

$\{x_i, y_i\}$, i = 1, … $m + k$ with $x_i \epsilon R^n$ and $y_i \epsilon \{-1,1\}$

Such that

$$\begin{cases} x_i \,\epsilon\, \text{A} & \text{if} \quad y_i = 1 \\ x_i \,\epsilon\, \text{B} & \text{if} \quad y_i = -1 \end{cases}$$

A separation hyperplane $\text{v}^T\text{x} - \gamma = 0$ for such data satisfies the conditions

$$\begin{cases} \text{v}^T x_i - \gamma \geq 1 & \text{if} \quad y_i = 1 \\ \text{v}^T x_i - \gamma \leq -1 & \text{if} \quad y_i = -1 \end{cases}$$

Which can still be written as

$$y_i\,[\text{v}^T x_i - \gamma] \geq 1 \qquad \text{i} = 1, \dots m + k$$

This separation hyperplane, however, may not be the best, in the sense that some points of A and / or B may be very close to the hyperplane itself. It is therefore preferable to find a hyperplane with a large separation margin or even with the largest possible margin. The hyperplane with the maximum margin is known as the optimal separation hyperplane.

Given the sets A and B, linearly separable, represented by the samples $\{x_i, y_i\}$, i = 1, … $m + k$ *with* $x_i \,\epsilon\, R^n$ and $y_i \,\epsilon\, \{-1,1\}$, suppose to be able to identify a separation hyperplane $\text{v}^T\text{x} - \gamma = 0$

where $v \in R^n$ is normal with respect to the hyperplane, $|y|/||v||$ is the perpendicular distance from the hyperplane to the origin and $||.||$ is the Euclidean norm.

Indicate with d_+ and d_- the smallest distance from the hyperplane of separation of the points of A and B, respectively, closest to it.

For the case of linearly separable sets, the SVM approach simply looks for the separation hyperplane with the largest margin, where the margin is given by the sum of d_+ and d_- and is equal to $2/||v||$.

The goal is, therefore, to find the pair of hyperplanes that give the maximum margin and therefore minimize $||v||^2$

$$\min_{v,\gamma} \frac{1}{2} ||v||^2$$

under the constraints $y_i\ [v^T x_i - \gamma] - 1 \geq 0 \forall\ i = 1, \dots m + k$

The points for which the constraints are satisfied by equality, the removal of which would change the solution found, are called support vectors.

A Lagrangian formulation of the problem is now given so that the above constraints will be replaced by constraints on Lagrange multipliers which are easier to deal with, and since in the Lagrangian reformulation of the problem the data will appear only in the form of scalar products between vectors. The latter is an important property that allows the generalization of the proposed procedure to the non-linear case.

It then introduces the Lagrange multipliers (not negative): α_i for i = 1, … m+k

it gets the Lagrangian function:

$$L_P \equiv \tfrac{1}{2} ||v||^2 - \sum_{l=1}^{m+k} \alpha_i y_i (v^T x_i - \gamma) + \sum_{l=1}^{m+k} \alpha_i$$

being the problem $\min_{v,\gamma} \frac{1}{2} ||v||^2$ a convex quadratic programming problem, it is possible to solve equivalently the following dual problem: to maximize L_P subject to the constraints that the gradient of L_P with respect to v and γ is equal to the null vector

$$v = \sum_{i=1}^{m+k} \alpha_i y_i x_i \sum_{l=1}^{m+k} \alpha_i y_i = 0$$

and that the multipliers α_i are non-negative ($\alpha_i \geq 0$).

By replacing the equality constraints just written in the expression of the objective L_P, the following problem is obtained:

$$\max_{\alpha} L_D = \sum_{i=1}^{m+k} \alpha_i - \frac{1}{2} \sum_{i=1}^{m+k} \sum_{j=1}^{m+k} \alpha_i \alpha_j y_i\, y_j x_i^T x_j$$

Under the constraints $\sum_{l=1}^{m+k} \alpha_i y_i = 0$ and $\alpha_i \geq 0$.

For linearly separable sets, therefore, the training in terms of SVM consists in solving the above problem with solution v obtained from the equality constraint $v = \sum_{i=1}^{m+k} \alpha_i y_i x_i$.

Those sample points for which $\alpha_i > 0$ in the solution are called support vectors. To all the other points correspond multipliers $\alpha_i = 0$ which do not contribute to the determination of v. For these machines (SVM) the support vectors are the critical elements of the training set, they are, in fact, the points closest to the decision boundary and are those that determine the optimal separation hyperplane.

The Karush-Kuhn-Tucker (KKT) conditions for the primal problem $\min_{v,\gamma} \frac{1}{2} ||v||^2$ under the constraints $y_i\ [v^T x_i - \gamma] - 1 \geq 0\ \forall\, i = 1, \dots m + k$ are:

$$\frac{\partial}{\partial v_j} L_P = v_j - \sum_{i=1}^{m+k} \alpha_i y_i x_{ij} = 0 \quad j = 1, \dots n$$

$$\frac{\partial}{\partial v_j} L_P = \sum_{i=1}^{m+k} \alpha_i y_i = 0$$

$$y_i(v^T x_i - \gamma) - 1 \geq 0 \qquad i = 1, \dots m+k$$

$$\alpha_i \geq 0 \qquad i = 1, \dots m+k$$

These conditions in the case under analysis are necessary and sufficient for v, γ and α to be the solution of the SVM problem. Solving, therefore, the SVM problem is equivalent to finding a solution to the KKT conditions. As an immediate application it is possible to explicitly determine v from the training procedure ($v = \sum_{i=1}^{m+k} \alpha_i y_i x_i$), while the threshold γ can be easily calculated from the complementarity KKT condition by choosing some point x_i for which $\alpha_i > 0$ (it would be, however, numerically safer to take the mean value of γ resulting from all these equations).

Once the SVM has been trained, given a new test point x, it is determined on which side of the decision boundary the point itself lies and the label of the corresponding class is assigned, i.e. the class of x will be:

$$sgm(v^T x - \gamma).$$

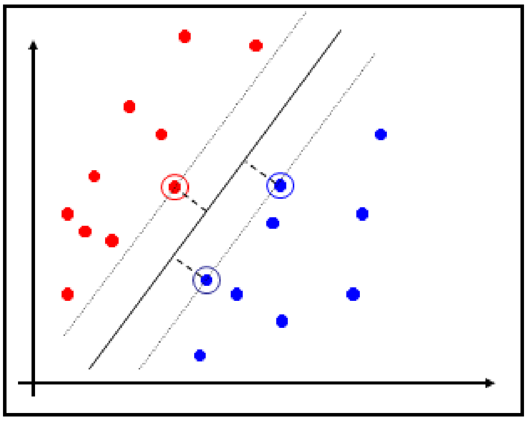


Figure A.2: The examples closest to the separation plane are called support vectors and are the most important vectors for maximizing the separation margin between classes.

Linear SVM and non-linearly separable sets

One way to extend the above ideas to the non-separable case could be to relax the separability constraints

$$\begin{cases} \mathrm{v}^{\mathrm{T}} x_{\mathrm{i}} - \gamma \ \geq \ 1 & \text{if} \quad y_{\mathrm{i}} = 1 \\ \mathrm{v}^{\mathrm{T}} x_{\mathrm{i}} - \gamma \ \leq \ -1 & \text{if} \quad y_{\mathrm{i}} = -1 \end{cases}$$

but only when necessary, thus adding an additional cost to the objective function $\min_{v,\gamma} \frac{1}{2} ||v||^2$. This can be done by introducing positive slack variables ξ_{i}, i = 1, … m+k, the constraints then become:

$$\begin{cases} \mathrm{v}^{\mathrm{T}} x_{\mathrm{i}} - \gamma \ \geq \ 1 - \xi\mathrm{i} & \text{if} \quad y_{\mathrm{i}} = 1 \\ \mathrm{v}^{\mathrm{T}} x_{\mathrm{i}} - \gamma \ \leq \ -1 + \xi\mathrm{i} & \text{if} \quad y_{\mathrm{i}} = -1 \end{cases}$$

$\xi_{\mathrm{i}} \geq 0 \qquad \forall \ \mathrm{i} = 1, \ldots \mathrm{m+k}.$

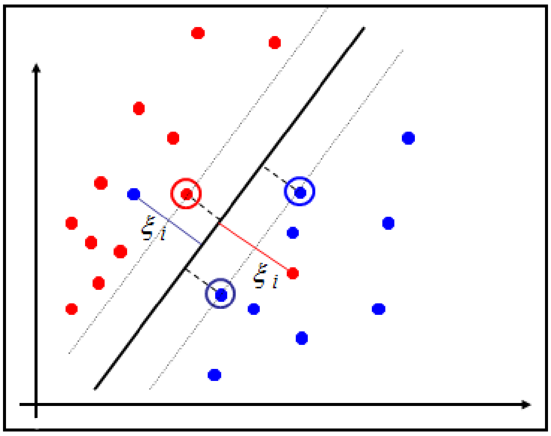


Figure A.3: If the training set is not linearly separable, the slack variables are introduced that allow the wrong classification of some points.

When there is an error (badly classified point), ξ_{i} is greater than or equal to one and, thus, $\sum_{i=1}^{m+k} \xi\mathrm{i}$ represents an upper bound on the number of badly classified points. A natural way to assign additional cost to the errors made by a separation hyperplane is to transform the objective function to be minimized from ½ $||v||^2$ in

$$½ \, ||v||^2 + \mathrm{C} \, (\textstyle\sum_{i=1}^{m+k} \xi\mathrm{i})^{\mathrm{q}}$$

Where C is a positive parameter to be fixed (the larger C the greater the penalty assigned to classification errors). For any choice of q positive integer, the problem

$$\min_{v,\gamma,\xi} \frac{1}{2} ||v||^2 + \mathrm{C} \, (\textstyle\sum_{i=1}^{m+k} \xi\mathrm{i})^{\mathrm{q}}$$

under the constraints

$y_i\ [v^T x_i - \gamma] - 1 + \xi i \geq\ 0 \forall\ i\ = 1, \dots m + k$

$$\xi_i \geq 0 \qquad \forall\ i = 1, \dots m+k.$$

It is convex and, for q = 2 and q = 1, it is also quadratic. The choice of q = 1 has as a further advantage the fact that neither the variables ξi nor their Lagrange multipliers appear in the dual problem which becomes:

$$\max_{\alpha} L_D = \sum_{i=1}^{m+k} \alpha_i - \frac{1}{2} \sum_{i=1}^{m+k} \sum_{j=1}^{m+k} \alpha_i \alpha_j y_i\, y_j x_i^T x_j$$

under the constraints $\sum_{l=1}^{m+k} \alpha_i y_i = 0$ and $C \geq \alpha_i \geq 0$.

with $v = \sum_{i=1}^{m+k} \alpha_i y_i x_i$

The only difference in the linearly separable case is therefore the upper bound C on the multipliers α_i. The Lagrangian of the primal problem turns out to be

$L_P = ½\ ||v||^2 + C\sum_{i=1}^{m+k} \xi i - \sum_{l=1}^{m+k} \alpha_i [y_i (v^T\ x_i - \gamma) - 1 + \xi i] - \sum_{i=1}^{m+k} \mu_i\ \xi i$

where the μ_i are the Lagrange multipliers introduced to reinforce the positivity of the variables ξi. KKT's conditions for the primal problem are:

$\frac{\partial}{\partial v_j} L_P = v_j$ - $\sum_{i=1}^{m+k} \alpha_i y_i\, x_{ij} = 0$ j = 1, … n

$\frac{\partial}{\partial v_j} L_P = \sum_{i=1}^{m+k} \alpha_i y_i\ = 0$

$\frac{\partial}{\partial v_j} L_P = C - \alpha_i - \mu_i = 0$ i = 1, … $m+k$

$y_i(v^T x_i - \gamma) - 1 + \xi i \geq 0$ i = 1, … $m+k$

$\alpha_i \geq 0$; $\xi i \geq 0$; $\mu_i \geq 0$ i = 1, … $m+k$

$\alpha_i\ [y_i(v^T x_i - \gamma) - 1 + \xi i] = 0$ i = 1, … $m+k$

$\mu_i \xi i = 0$ i = 1, … $m+k$

As for the separable case, the complementarity conditions can be used to determine the γ threshold. From the conditions of KKT just described, in fact, it has

$\xi i = 0$ when $\alpha_i < C$

It follows, therefore, that the computation of γ can be obtained from the complementarity relation relative to any sample point for which it results $0 < \alpha_i < C$ (as in the previous case it would be numerically safer to take the mean value of γ resulting from all these equations).

Non-linear SVM and non-linearly separable sets

The linear separation methods based on the SVM theory, which aim to determine an optimal separation hyperplane, can be generalized to the case of non-linear separation as follows: a transformation of the input space into a dimensionally larger space is carried out through a non-linear function, optimal separation hyperplanes are constructed in the augmented space, which correspond to non-linear surfaces in the original input space.

Returning to the linear case, it is possible to note that in the training problem the data comes into play only in the form of scalar products, $x_i^T x_j$:

$$\max_{\alpha} \sum_{i=1}^{m+k} \alpha_i - \frac{1}{2} \sum_{i=1}^{m+k} \sum_{j=1}^{m+k} \alpha_i \alpha_j y_i \, y_j x_i^T x_j$$

under the constraints $\sum_{l=1}^{m+k} \alpha_i y_i = 0$ and $C \geq \alpha_i \geq 0$.

Now supposing to transform the input space R^n into a generic Euclidean space H, through the transformation Φ :

$$\Phi: R^n \rightarrow H,$$

the training algorithm will depend only on the scalar products of the data in H (the transforms of the sample points), i.e. on functions of the form $\Phi(x_i)^T\Phi(x_j)$. If there exists a "kernel" function K, such that

$$K(x_i, x_j) = \Phi(x_i)^T \, \Phi(x_j),$$

the training algorithm will only use K without ever having the need to know explicitly Φ. Of course, all the considerations made for the linear case continue to hold true, as it still continues to make a linear separation, but in a new space:

$$\max_{\alpha} \sum_{i=1}^{m+k} \alpha_i - \frac{1}{2} \sum_{i=1}^{m+k} \sum_{j=1}^{m+k} \alpha_i \alpha_j y_i \, y_j K(x_i^T x_j)$$

under the constraints $\sum_{l=1}^{m+k} \alpha_i y_i = 0$ and $C \geq \alpha_i \geq 0$.

As for the vector v, which characterizes the separation hyperplane, it will belong to the space H and for its calculation it must be explicitly known Φ:

$$v = \sum_{i=1}^{Ns} \alpha_i y_i \, \Phi(x_i).$$

However, it can be observed that in the test phase it is only necessary to calculate the scalar products of a given test point x with v, or more specifically to calculate the sign of $(v^T\Phi(x) - \gamma)$, it is thus obtained:

$sgn(\sum_{i=1}^{m+k} \alpha_i y_i \, \Phi(x_i)^T \, \Phi(x) - \gamma) = sgn(\sum_{i=1}^{Ns} \alpha_i y_i \;\; K(x_{i,} x) - \gamma)$,
Avoiding, again, the calculation of $\Phi(x)$ and using the kernel function instead.

Among the most popular Kernel functions are:

- Polynomial: $K(x, y) = (\gamma x^T y + 1)^P$, $\gamma > 0$.
- Radial Basis Function (RBF): $K(x, y) = \exp(-\gamma \|x_i - x_j\|^2)$, $\gamma > 0$.
- Sigmoid: $K(x, y) = \tanh(\gamma x_i^T x_j + r)$.

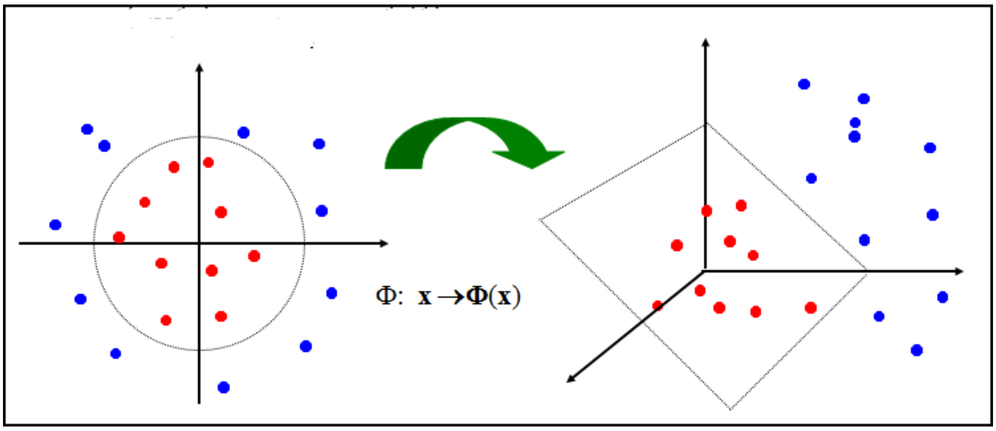


Figure A.4: Exemplary example of projecting the problem into a larger space.

In order to train an SVM, therefore, one must solve a quadratic optimization problem with bound constraints and a linear equality constraint. This problem put in the form of minimization becomes:

$$\min_{\alpha} - \sum_{i=1}^{m+k} \alpha_i + \frac{1}{2} \sum_{i=1}^{m+k} \sum_{j=1}^{m+k} \alpha_i \alpha_j y_i\, y_j K(x_i, x_j)$$

under the constraints $\sum_{i=1}^{m+k} \alpha_i y_i = 0$; $C \geq \alpha_i \geq 0$; $\forall$ i = 1, … m+k

Where it is recalled that with m + k are the number of sample points, which are used for training, and with α a vector with m + k components, one for each point (x_i, y_i).

By defining the Q matrix as

$$(Q)_{ij} = y_i y_j K(x_i, x_j)$$

The problem then can be equivalently written as:

$$\min_{\alpha} - e^T \alpha + \frac{1}{2} \alpha^T Q \alpha$$

under the constraints $\alpha^T y = 0$; $Ce \geq \alpha \geq 0$.

The size of this optimization problem depends on the number of samples, m + k. If this number is very large, since the size of the matrix Q is equal to $(m+k)^2$, it will be impossible to keep Q in memory.

A.2 Neural Network

Neural networks are made up of simple elements called artificial neurons that operate in parallel. These elements, as the name suggests, are inspired by the biological nervous system in which multiple neurons are activated or not depending on the intensity and activation of their synaptic connections [78].

In the human nervous system there are 10^{11} neurons (nerve cells) of different types each consisting of a cell body and many branched extensions, called dendrites, through which the neuron receives electrical signals from other neurons. Each neuron also has a filamentous extension called axon which branches out at its extremity, forming terminals through which electrical signals are transmitted to other cells (for example the dendrites of other neurons). Between a terminal of an axon and the receiving cell there is a space that signals pass through by means of chemicals called neurotransmitters; the point of connection between terminal and dendrite is called synapse.

A neuron "activates", that is, it transmits an electrical impulse along its axon, when there is an electrical potential difference between the inside and the outside of the cell and this electrical impulse causes the release of a neurotransmitter from the synaptic terminals of the cell. The impulse from the Axon can for example affect other neurons. Biological neurons are 5 to 6 orders of magnitude slower than conventional electronic components: an event in a chip occurs in a few nanoseconds while a neural event occurs in a few milliseconds. Nevertheless the human brain, despite being made up of very simple processing elements (the neurons), is able to perform complex computations (such as recognition, perception and control of movement), many times faster than the fastest of current computers and is also able to modify the connections between neurons based on experience: it is able to learn. Finally, the brain is fault tolerant, that is, whether a neuron or one of its connections are damaged, it continues to work, albeit with slightly performance minors that gradually degrade as more and more neurons are destroyed (graceful degradation). Therefore, in order to artificially reproduce the human brain, it is necessary to create a network of very simple elements that is distributed, strongly parallelized, capable of learning and therefore of generalizing (that is, producing outputs at inputs not encountered during training).

As in nature, the function of the artificial neural network is largely determined by the connections between the elements. One is able to train a neural network to perform a particular function by adjusting the values or weights of the connections between elements. Commonly, neural networks are trained so that a particular input configuration produces an output that satisfies a given objective. This situation is schematized in the Figure A.5.

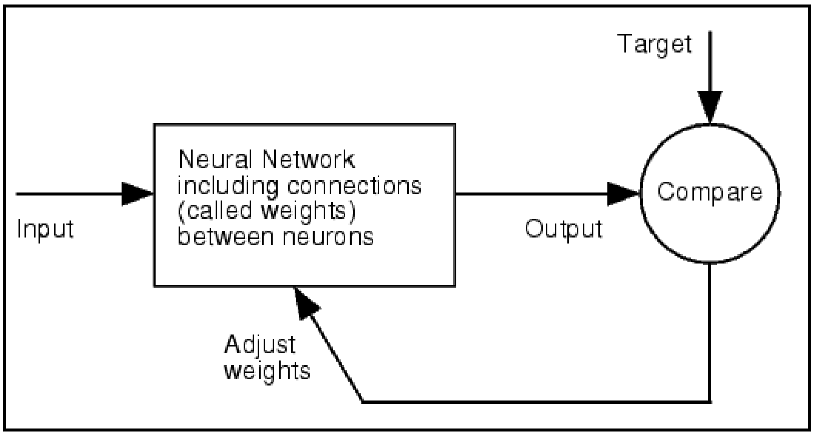

Target
Neural Network including connections (called weights) between neurons
Input
Output
Compare
Adjust weights

Figure A.5: Neural network training scheme: given the inputs to the network, the output is compared with a target and the weights between the neuronal interconnections are changed until the target is satisfied [79].

The field of neural networks has a history that dates back to more than half a century ago, and in recent years has found solid applications thanks to deep neural networks and CNNs. The fields of application have developed rapidly and neural networks have been trained to perform complex functions in various disciplines, including pattern recognition, identification, classification, speech, vision, control systems, etc. The supervised training method is commonly used, but unsupervised techniques can also be used, for example, for identifying data groups.

The neuron model and the architecture of a neural network describe how a neural network transforms inputs into outputs, and represent the limitation on what a particular neural network can calculate.

Artificial neuron model

A simple neuron model with a single input is shown in the Figure A.6.

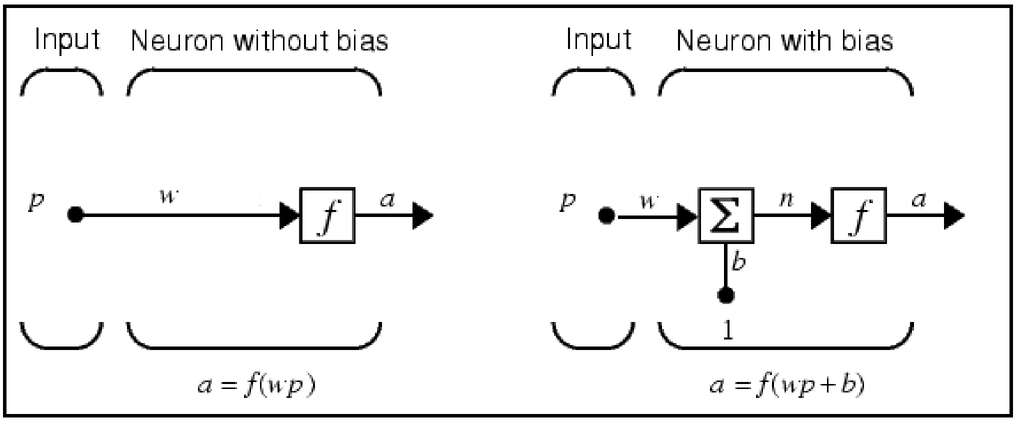


Figure A.6: Graphic example of an artificial neuron with a single input (left) and with sum of the bias (right) [79].

The scalar input is transmitted through a connection that multiplies its strength by the weight w also scalar, to form the product wp. This product is subjected to the transfer function f which produces a scalar output. As an alternative to the simplified model described, it is also possible to add a scalar bias b to the product pw. The bias is very similar to a weight, except for the fact that it has a constant input equal to 1. In this case the input n of the transfer function f is the sum of the weighted input with w plus the bias b.

The transfer function is typically a step function or a sigmoid which, taken as an argument, produces the output a; w and b therefore represent adjustable parameters of the neuron. The central idea of neural networks is that these parameters can be adjusted so that the network exhibits a desired behavior. Then it can train the network to do a particular job by adjusting the weight and bias to achieve a desired end. The transfer functions can be of different types, four of the most used are shown in the Figure A.7.

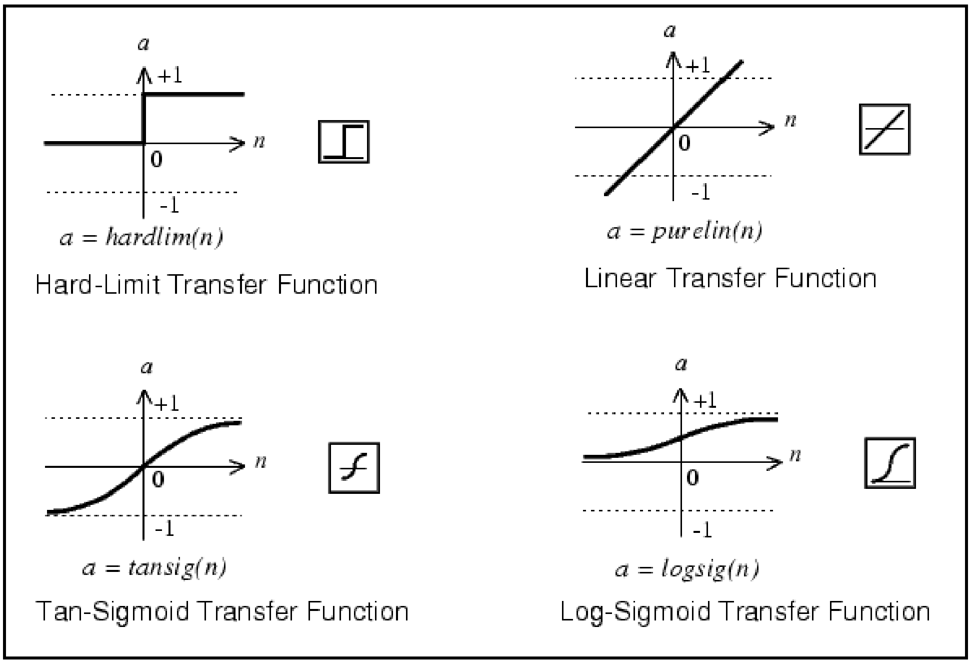


Figure A.7: Examples of transfer functions [79].

For example, the "hard-limit" shows that the neuron's output can be zero, if the input argument n is less than zero, or one, if n greater than zero.

Expanding the discussion, a neuron with an input vector with R elements is shown in Figure A.8.

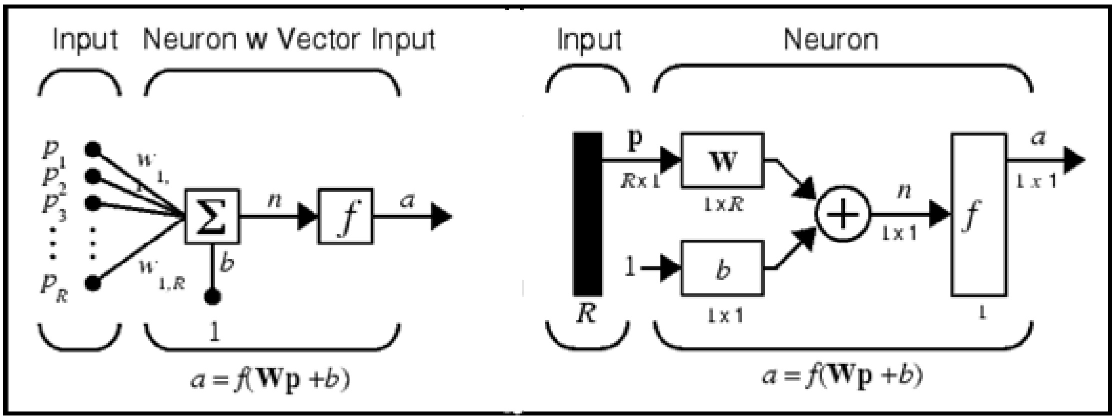


Figure A.8: Graph of an artificial neuron with R inputs [79].

The R input elements are represented by the vector $p = p_1, p_2, \dots p_R$

and are multiplied by the weight vector $W = w_1, w_2, \dots w_R$

this time the argument n in input to f is: $n = Wp + b = w_1p_1 + w_2p_2 + .. w_Rp_R + b$

If the neuron has the “hardlim” as a transfer function, shown above, it takes the name of perceptor or adaline (adaptive linear element). A perceptron is a threshold logic unit (ULS) that computes a weighted sum of the inputs and returns one or zero based on the threshold or transfer function [80].

A ULS separates the space of the input vectors that produce a response beyond the threshold from those that produce a response below the threshold through a linear surface called an R-dimensional hyperplane. For this reason the functions that can be implemented by a perceptron are called linearly separable functions. Learning is achieved by adapting the weights until performance is acceptable given a certain goal.

One method of approaching the problem of training a perceptor, to ensure that it responds appropriately to the training vectors, is to define an error function that can be minimized by changing the values of the weights [80]. A commonly used function is the quadratic error:

$$\varepsilon = \sum_{P_k} (d_k - f_k)^2$$

where f_k is the ULS response for the input vector P_k, and d_k is the desired output. The summation is calculated on all vectors $P_k = p_1, p_2, \dots p_k$ of the training set.

If the set of data is linearly separable, the error can be canceled, otherwise, in the case of non-linearly separable sets, it can be minimized. The error depends on the weights, once the training set and the desired values have been fixed. A minimum of the error can then be found by carrying out a descent process along the gradient.

A first problem in calculating the gradient is represented by the definition of the quadratic error which depends on all the k input vectors of the training set. An incremental approach is preferable, that is to modify the weights for each vector of the training set, the results can only approach those of the version that uses all the vectors together but it is a rather effective approximation. For a single input vector p the quadratic error is given by ε = (d - f)^2 and the gradient with respect to the weights is:

$$\frac{\partial \varepsilon}{\partial W} \triangleq \left[\frac{\partial \varepsilon}{\partial w_1}, \frac{\partial \varepsilon}{\partial w_2}, \cdots \frac{\partial \varepsilon}{\partial w_R}\right]$$

If the transfer function is of the step type (non-differentiable) it is clear that the gradient calculation must be faced with particular procedures such as the Widrow-Hoff procedure and the generalized delta procedure.

A perceptron has many limitations, first the output can be only 0 or 1, second, it can classify only linearly separable vector sets. To overcome these problems, more neurons can be aggregated to form an interconnected network. Neural networks can learn the appropriate behavior through training sets even for unseen inputs.

Learning is generally obtained by adapting the net weights until its performance in calculating the action is not acceptable. In the training phase, the network is adjusted on the basis of a comparison between the output and the target until the network output matches or resembles the target. Usually many input / objective pairs are used for training.

Neural network architecture

Many stimulus-responsive behaviors cannot be learned from a single ULS. This is the case where non-linearly separable sets come into play. In such a situation, correct answers can be obtained from a network of neurons. Several neurons connected in a certain way constitute an architecture of neurons.

Specifically, two or more neurons can be combined into one neuron layer, and multiple neuron layers can be combined to form a given neural network.

The function implemented by a network of neurons depends on its topology as well as on the weights of the individual neurons. Networks with forward connections (feed-forward) do not have cycles, that is no input to a neuron depends on its own output (even if the output passes through several intermediate neurons). Networks that do not have only forward connections are called feed-back networks.

If the neurons of a network with forward connections are organized in layers, with elements of the j-th layer receiving inputs only from the neurons of the j-1 layer, then the network is defined as multilayer with forward connections [80].

Considering a single layer consisting of S neurons it must specify:

the vector p containing the R inputs of the network: $p = p_1, p_2, \ldots p_R$,

the bias vector b: $b = b_1, b_2, \ldots b_S$,

the vector f oft he transfer functions: $f = f_1, f_2, \ldots f_S$,

and the matrix W of the weights:

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ & & & \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

Each of the R inputs is connected to each neuron and is weighted appropriately by the corresponding weight of the matrix W of dimensions SxR. The indices of the row of the W matrix indicate the neuron (from one to S) to which the weights are associated, the index inherent in the column indicates the input (from one to R).
The i-th neuron among the total S, provides its transfer function with the scalar n_i given by the sum of the products of the inputs for the relative weights and the i-th bias. This time the argument n_i in input to the f_i is:

$$n_i = w_{i,1}p_1 + w_{i,2}p_2 + \ldots\ w_{i,R}p_R + b_i$$

The output is a vector composed of the i elements that is the outputs generated by the S neurons:

$$a = a_1,\ a_2,\ \ldots\ a_S$$

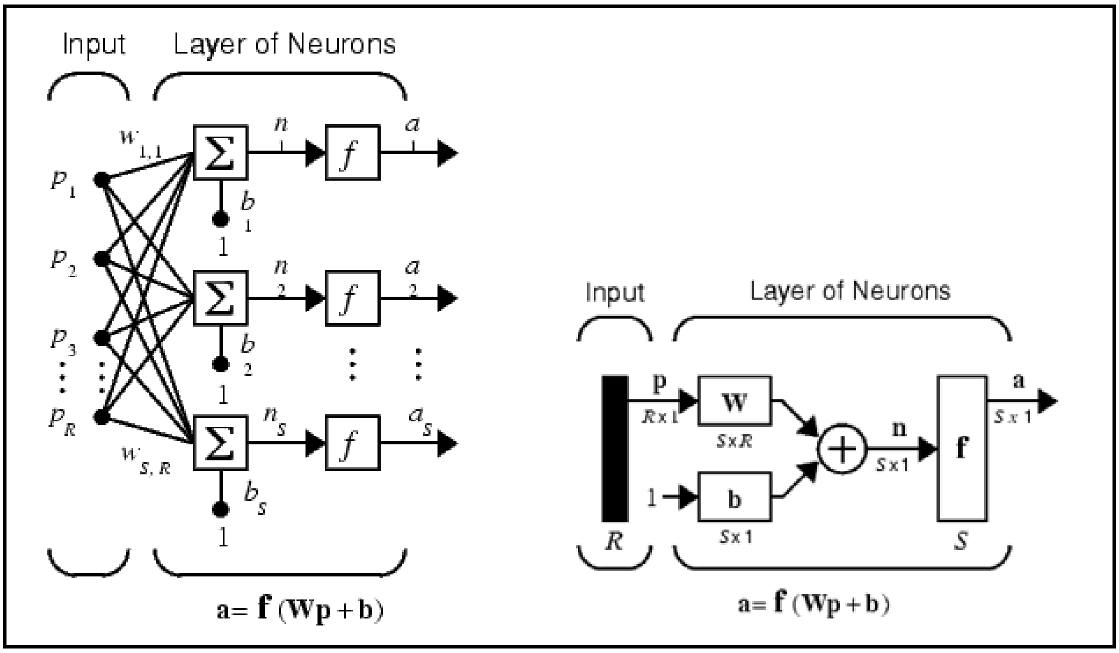


Figure A.9: Layer of S artificial neurons with R inputs and S outputs [79].

To talk about multilayer networks it is necessary to extend the notation used previously and distinguish between the weights related to the inputs and the weights associated with the connections between the layers. In this regard, with a network consisting of j layers, it has for each j-th layer an input vector p^j, a bias vector b^j, a vector of transfer functions f^j, an output vector a^j and a matrix W^j that is a matrix for each layer similar to the one discussed for the case of a single layer.

In essence, the index j serves to associate the corresponding layer of the network to which they are used between all the j weight matrices and between all the input vectors, biases, transfer and output functions.

Thus, p^1 and S^1 will respectively indicate the inputs and the number of neurons of the first layer, p^2 and S^2 those of the second layer and so on. It is common to have different quantities of neurons per layer, moreover, the outputs of a given layer represent the inputs of the next layer that is

$$a^j = p^{j+1}$$

So the input to layer 2 is a^1. Having made these observations, for each layer j the same considerations made for a single layer apply.

The layer that produces the final outputs is called an output layer, the others are called hidden layers.

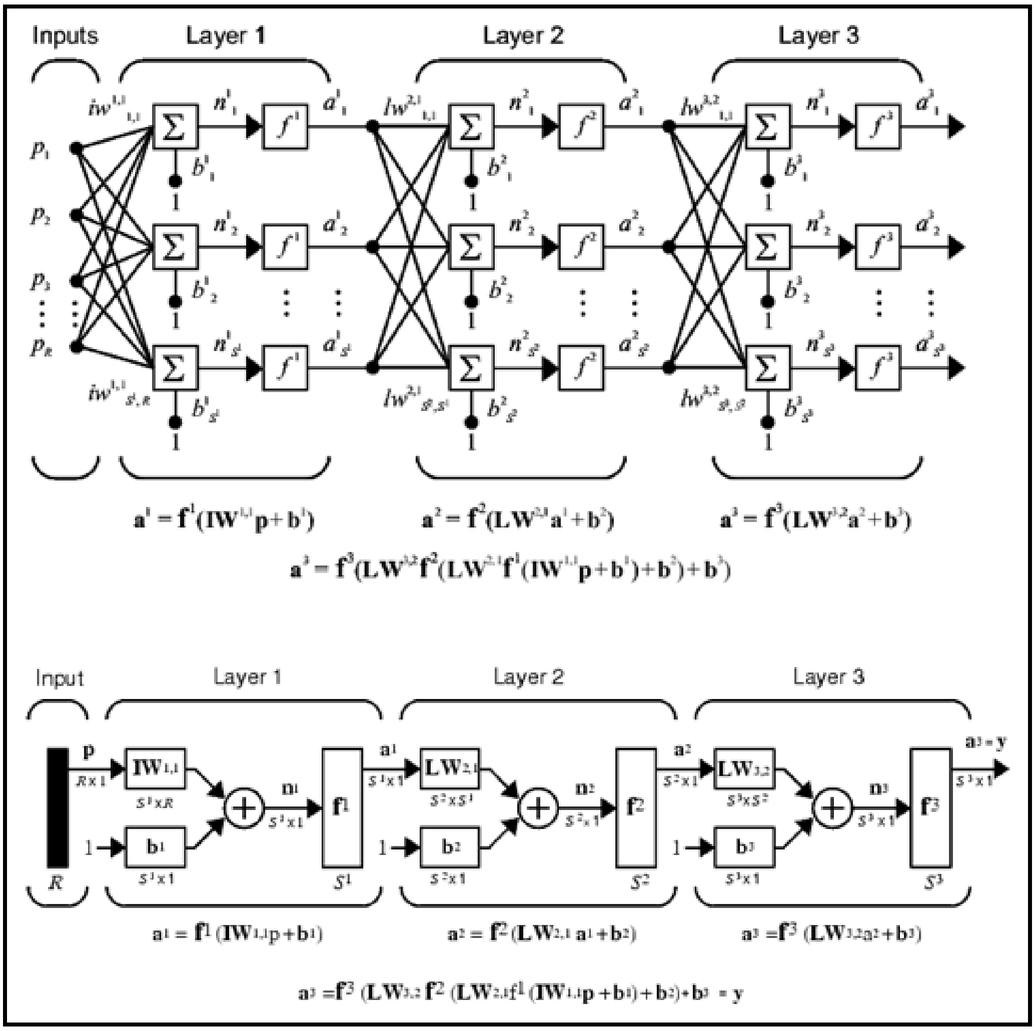


Figure A.10: Multi-layer neural network [79].

The design of a network architecture consists of a description of how many layers a network has, the number of neurons in each layer, the transfer functions of each layer and how the layers are connected to each other. The architecture depends on the type of problem and the inputs to the network can be presented in parallel or rarely sequentially. Using an arbitrarily large number of neurons, any finite function in a finite domain can be approximated, but in general there is no a priori rule to determine the number of neurons for a given problem.

The training, that is the modification of the weights of the networks, can be carried out with the method of descent along the gradient as seen for the single perceptor using a procedure called backpropagation.

Backpropagation and generalization

The learning algorithm of a neural network with backpropagation is based on the gradient descent method that allows to find a local minimum of a function even in a large space. The weights associated with the connections between the layers of neurons are initialized to small and random values, then the learning rule is applied by presenting sample patterns to the network.

The training of a network with the backpropagation method takes place in two different stages: forward-pass and backward-pass. In the first phase, the input vectors are applied to the input neurons with forward propagation of the signals through each level of the network until the output is obtained. During this phase all the weights are fixed. In the second phase, the response of the network is compared with the desired output obtaining an error. The calculated error is propagated in the reverse direction with respect to the connections and the weights are modified in order to minimize the error or the difference between the current output and the desired output. A forward-pass backward-pass loop is called a learning epoch.

Well trained backpropagation networks tend to give reasonable answers even on unseen inputs. This generalization property allows to train a network starting from input output pairs and obtain good results even on data never seen before.

An error function that can be used is the quadratic error seen above. In this case the weight vector against which the gradient is calculated should include all the weights of the network. However, it is advisable to calculate the partial derivatives with respect to the weights of the individual neurons.

$$\frac{\partial \varepsilon}{\partial w_i^j} \triangleq \left[\frac{\partial \varepsilon}{\partial w_{1,i}^j}, \frac{\partial \varepsilon}{\partial w_{2,i}^j}, \cdots \frac{\partial \varepsilon}{\partial w_{L,i}^j}\right]$$

where W_i^j represents the matrix of the weights of the i-th neuron of the J layer, and has L weights one for each output of the previous layer or R weights in the case of the first layer. In back propagation it is very important to be able to calculate the derivative of each transfer function.

The modification of the weights occurs recursively starting from the neurons of the last layer up to those of the first. For this reason the method takes the name of back propagation. The backward propagation algorithm, trying to minimize the difference between each desired output and the output actually calculated by the network, distributes the contribution of the error on each weight.

The number of inputs to a network is usually large and the complexity becomes so exponential that it is not possible to provide a training set that considers all the possible cases of a given function. A network is said to be able to generalize when it appropriately responds or classifies vectors not included in the training set.

The concept of generalization of a neural network is similar to the adaptation of geometric curves [80]. When trying to fit a straight line or polynomial curve to some data, it captures some underlying relationship to the data. If the fit to the data is very good and if there is a lot of data, the fit line or curve can then be used to determine (with reasonable reliability) the values for the new data not used before in the adaptation process. The same can be said for neural networks that compute a complex and non-linear function of their inputs. The adaptation of a trained network with a non-small set of data and representative of the domain of the problem being treated, results in a good chance of calculating the correct behavior even for never seen inputs. Behind this, there is the assumption that training data adeguately reflects the characteristics of the field data.

A problem that can occur in the training phase is called overfitting. This happens when the trained network over-adapts to the training set and doesn't generalize properly to new data. In this sense, the error on the training set tends to cancel itself out, but when the new data is presented to the network, a high error occurs. To avoid this problem, it is common practice to use a validation dataset separate from the training set on which to estimate the degree of generalization. A further set of tests is used to estimate the accuracy of a neural network in achieving the given objective for which it was designed. The use of validation set to stop training when the network begins to overfit the data is called early stopping.

To improve generalization, another solution is the regularization that modifies the network’s performance function (the measure of error that the training process minimizes). By including the sizes of the weights and biases, regularization produces a network that performs well with the training data and exhibits smoother behavior when presented with new data. Regularization attemps to construct a model structure as simple as possible. In keeping with Occam’s razor principle, the simplified model can avoid the effects of overfitting while a complex model tends to be overfitting.

Finally, to avoid overfitting in deep networks with many hidden layers the solution is the dropout, which trains only some of the randomly selected nodes rather than the entire network. Some node are randomly selected at a certain percentage and their outputs are set to be zero to deactivate the nodes. The dropout effectively prevents overfitting as it continuously alters the nodes and weights in the training process.

A.3 Convolutional Neural Network

Over the past decade, the popularity of methods that exploit deep learning techniques has considerably increased, evidently as deep learning has improved the state of the art in research fields such as speech recognition and computer vision [81]. Although neural networks had their scientific boom already in the 80s, their recent success can be attributed to an increased availability of data, improvements in hardware / software [82] and it is also due to new algorithms capable of both speeding up learning in the training phase and improving the generalization of new data [83].

In the field of computer vision, deep learning has expressed its potential in image processing thanks to Convolutional Neural Networks (CNNs). CNN is an old technique, which was developed in the 1980s and 1990s [84]. Conceptually CNNs are inspired by the visual system as proposed in the works of Hubel and Wiesel on cat and monkey visual cortex [85]. A CNN accepts an image directly as input and applies a hierarchy of different convolution kernels to it. The first layers allow to extract elementary visual features such as oriented edges, end-points, corners and are gradually combined with subsequent layers in order to detect higher-order features [86].

The success of the CNNs is certainly due, in addition to the high classification performance demonstrated by these classification methods, also by the ease of carrying out a classification process using these tools. In fact, the traditional chain composed of preprocessing, feature extraction, training model, is entirely replaced by CNNs which in their training process include feature extraction. CNNs are networks specialized in data processing which have the form of multiple vectors with a known grid-form topology. An example of this type of data can be a time series, which can be seen as a grid at a size sampled from regular intervals, or an image, which can be seen as a two-dimensional grid of pixels containing the intensity value for the three color channels (RGB).

Convolutional neural networks (CNNs) enable learning trainable, highly representative and hierarchical image feature from sufficient training data which makes rapid progress in computer vision possible [87].

In the field of image classification in general, CNNs can be used, proposing an ad-hoc architecture and proceeding with its training, or using a pre-trained architecture. In the latter case, the training on specific data to the problem can be carried out from scratch, or through the fine-tuning of a part of the parameters / weights of the pre-trained network.

Fine-tuning is known as transfer learning as the knowledge of another problem is exploited to solve the object of the study. Furthermore, a pre-trained CNN architecture can be modified in its architecture before carrying out training or fine-tuning.

Another way to use a pre-trained CNN is to extract features in combination with a classifier, for example a Support Vector Machine (SVM) classifier. The advantage of this type of work is the simplicity of implementation (no retraining of the pre-trained networks must be carried out), the disadvantage is usually relatively lower performances than those obtained from the pre-trained networks with the fine-tuning method.

CNN architecture

The CNN architecture consists of a neural network that extracts features of the input image and another neural network that classifies the feature image. The input image enters into the feature extraction network, the extracted feature signals enter the classification neural network, and the classification neural network then operates based on the features of the image and generates the output. [88].

The second part of the CNN is a neural network or fully connected layers that take in input the features extracted from the first part of CNN. The architecture of the first part, the convolutional layers, is structured as a series of representations made up of two types of layers: the convolution layers and the pooling layers (see Figure A.11).

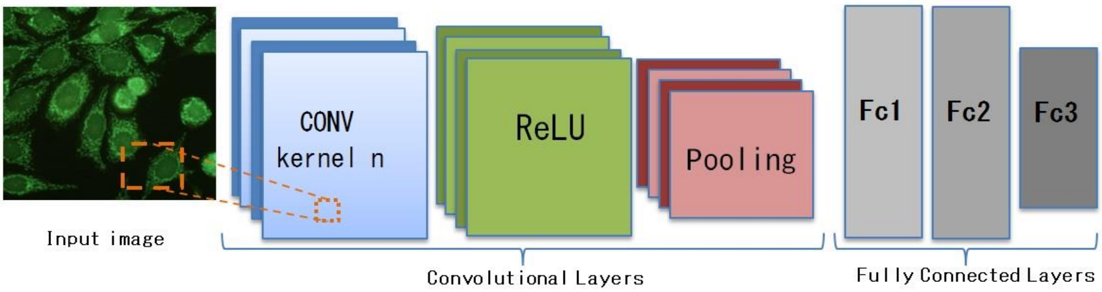


Figure A.11: General scheme of the architecture of a CNN.

The convolutional layers take their name from the convolution operations that are applied to the images in input to CNN while the pooling layers combine the neighboring pixels into a single pixel to progressively reduce the size of the image.

The units in the convolution layers are organized into feature maps, in which each unit is connected to a local portion of the map of the next layer through a set of weights called a filter bank. The result of this local weighted average is then passed through a nonlinear function such as ReLU (RectifiedLinear Unit). All units in a feature map share the same filter bank. The ReLU function produces zero for negative inputs and conveys the input for positive intput. It is known to better transmit the error than the sigmoid function and it is the solution to the vanishing gradient. The vanishing gradient in the training process of a deep neural network with the back-propagation algorithm occurs when the output error is more likely to fail to reach the farther nodes. The back-propagation algorithm trains the neural network as it propagates the output error backward to the hidden layers. However, as the error hardly reaches the first hidden layer, the weight cannot be adjusted. Therefore, the hidden layers that are close to the input layer are not properly trained.

The convolutional layers apply convolutional filters to produce feature maps, the number of feature maps in output from a convolutional layer is equal to the number of filters applied. These filters or kernels, are two-dimensional matrices of chosen size (1x1, 3x3, 5x5 and so on) and the matrices values are determined through the training process.

Consider briefly the concept of convolution, in general *H* is an operator Si consideri brevemente il concetto di convoluzione, in generale given *H* an operator which, applied to an input image *f(x,y)*, provides an output image *g(x,y)*:

$H[f(x,y)] = g(x,y)$

Chosen, for example, a 3x3 kernel, this is progressively applied to all the pixels of the image, and changes its value as follows:

$R = w_1z_1 + w_2z_2 + \ldots + w_9z_9 = \sum_{i=1}^{9} w_iz_i$

where *R* is the pixel value calculated thanks to the application of the convolution mask. If *f* have size *m* x *n* then:

$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)$ with $a = (m-1)/2$ and $b = (n-1)/2$

in this case it is assuming that *a* and *b* are positive integers and the filter is supposed to have odd dimensions, at least 3x3. The center coefficient of the filter, w (0, 0), is aligned with the pixel at position (x, y):

f(x-1,y-1)	f(x-1,y)	f(x-1,y+1)
f(x-1,y-1)	f(x,y)	f(x-1,y+1)
f(x+1,y-1)	f(x+1,y)	f(x+1,y+1)

w(-1,-1)	w(-1,0)	w(-1,1)
w(0,-1)	w(0,0)	w(0,-1)
w(1,-1)	w(1,0)	w(1,1)

The operation is repeated for all pixels of the image (sliding of the mask: at each step the origin of the filter advances to the next pixel). Let's consider a small example:

consider an image of 4x4 and a kernel 3x3:

1	2	3	4
0	2	0	2
10	9	0	1
1	10	1	2

1	0	0
0	1	1
1	1	1

the first step performs the following operations:
(1*1) + (2*0) + (3*0) + (0*0) + (2*1) + (0*1) + (10*1) + (9*1) + (0*1) = 22

1	2	3	4
0	2	0	2
10	9	0	1
1	10	1	2

*

1	0	0
0	1	1
1	1	1

=

22	

the second step performs the following operations:

(2*1) + (3*0) + (4*0) + (2*0) + (0*1) + (2*1) + (9*1) + (0*1) + (1*1) = 14

$$\begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 4 \\ \hline 0 & 2 & 0 & 2 \\ \hline 10 & 9 & 0 & 1 \\ \hline 1 & 10 & 1 & 2 \\ \hline \end{array} \quad * \quad \begin{array}{|c|c|c|}\hline 1 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad = \quad \begin{array}{|c|c|}\hline 22 & 14 \\ \hline & \\ \hline \end{array}$$

the third step performs the following operations:
(0*1) + (2*0) + (0*0) + (10*0) + (9*1) + (0*1) + (1*1) + (10*1) + (1*1) = 21

$$\begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 4 \\ \hline 0 & 2 & 0 & 2 \\ \hline 10 & 9 & 0 & 1 \\ \hline 1 & 10 & 1 & 2 \\ \hline \end{array} \quad * \quad \begin{array}{|c|c|c|}\hline 1 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad = \quad \begin{array}{|c|c|}\hline 22 & 14 \\ \hline 21 & \\ \hline \end{array}$$

the fourth step performs the following operations:
(2*1) + (0*0) + (2*0) + (9*0) + (0*1) + (1*1) + (10*1) + (1*1) + (2*1) = 16

$$\begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 4 \\ \hline 0 & 2 & 0 & 2 \\ \hline 10 & 9 & 0 & 1 \\ \hline 1 & 10 & 1 & 2 \\ \hline \end{array} \quad * \quad \begin{array}{|c|c|c|}\hline 1 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad = \quad \begin{array}{|c|c|}\hline 22 & 14 \\ \hline 21 & 16 \\ \hline \end{array}$$

A convolutional layer can be followed by an activation function like the ReLU function but also sigmoid or other. Subsequently a pooling layer is applied to reduce the size of the image: it binds neighboring pixels and replaces them with a representative value such as the maximum or mean value of the pixels.

$$\begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 4 \\ \hline 0 & 2 & 0 & 2 \\ \hline 10 & 9 & 0 & 1 \\ \hline 1 & 10 & 1 & 2 \\ \hline \end{array} \quad \text{Max-pooling} \quad \begin{array}{|c|c|}\hline 2 & 4 \\ \hline 10 & 2 \\ \hline \end{array}$$

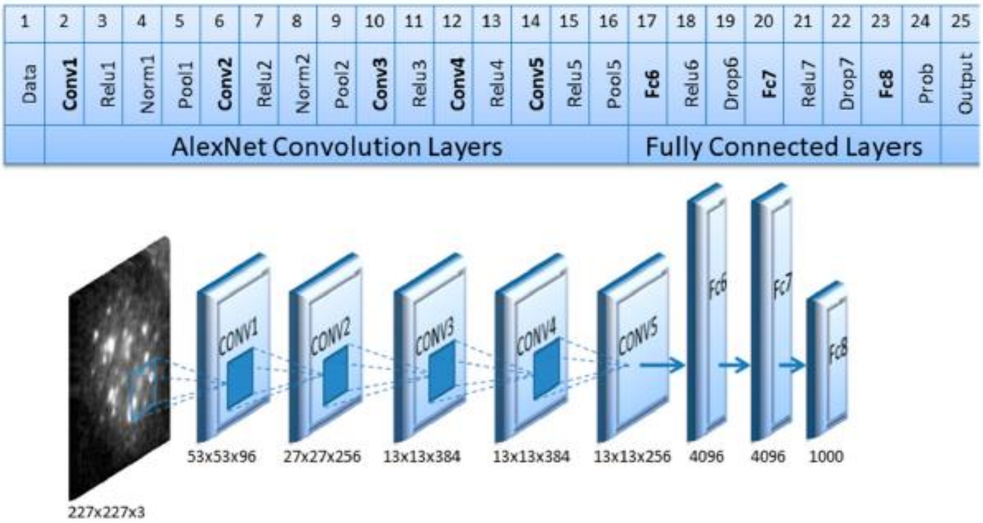
$$\begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 4 \\ \hline 0 & 2 & 0 & 2 \\ \hline 10 & 9 & 0 & 1 \\ \hline 1 & 10 & 1 & 2 \\ \hline \end{array} \quad \text{Average-pooling} \quad \begin{array}{|c|c|}\hline 1 & 2 \\ \hline 8 & 1 \\ \hline \end{array}$$

The max-pooling layer takes a pool size as a parameter, usually 2 by 2. It then processes its input image in the following way: divide the image in 2 by 2 areas (like a grid), and take from each four-pixel pool the pixel with the maximal value. Compose these pixels into a new image, with the same order as the original image. A 2 by 2 max-pooling layer produces an image that is half the size of the original. The idea behind max-pooling is that important information in a picture is seldom contained in adjacent pixels. The capacity of a convolutional neural network can vary based on the number of layers it has. In addition to having different types of layers, a CNN can have multiple layers of the same type. In fact, there is rarely a single convolutional level, unless the network in question is extremely simple. Usually a CNN has a series of convolutional levels, the first of these, starting from the input level and going towards the output level, is used to obtain low-level characteristics, such as horizontal or vertical lines, angles, various contours, etc. The levels closest to the output level produce high-level characteristics, i.e. they represent rather complex figures such as faces, specific objects, a scene, etc.

Pre-trained CNNs

The design and training of a CNN is an extremely complex problem, both for the necessary data but also for the useful computing power. One way to overcome the problem in the literature is to use pre-trained CNN networks. Thanks also to competition like ImageNet, extremely performing CNN networks have been created and published that are able to classify images in 1000 object categories.
The most known architectures are described below:

- **AlexNet** [83]: the AlexNet network is one of the first convolutional neural networks that has achieved great classification successes. Winner of the 2012 ILSVRC (Image-Net Large-Scale Visual Recognition Challenge) competition, this network was the first to obtain more than good results on a very complex dataset such as ImageNet. This network consists of 25 layers, the part relating to convolutional layers sees 5 levels of convolution with the use of ReLU and (only for the first two levels of convolution and for the fifth) the maxpooling technique. The second part of CNN is composed of full connected layers with the use of ReLU and Dropout techniques and finally by softmax for a 1000-d output. The AlexNet architecture is show in the Figure A.12.



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Data	**Conv1**	Relu1	Norm1	Pool1	**Conv2**	Relu2	Norm2	Pool2	**Conv3**	Relu3	**Conv4**	Relu4	**Conv5**	Relu5	Pool5	**Fc6**	Relu6	Drop6	**Fc7**	Relu7	Drop7	**Fc8**	Prob	Output
	AlexNet Convolution Layers															Fully Connected Layers								

Figure A.12: Scheme of the AlexNet architecture.

- **SqueezeNet** [89]: in 2016 the architecture of this CNN was designed to have performances comparable to AlexNet, but smaller with fewer parameters, so as to have advantage in distributed training, in export new model from the cloud, and in deploy on FPGA with limited memory. Specifically, this network consists of 68 layers with the aim of producing large activation maps. The filters used instead of being 3x3 are 1x1 precisely to reduce the computation by 1/9. CNN is made up of blocks called "fire modules", which contain a squeeze convolution layer with 1x1 filters and a expand layer with a mix of 1x1 and 3x3 convolution filters. This CNN has an initial and a final convolution layer, while the central part of the CNN is composed of 8 fire module blocks. No fully connected layers are used but an average pooling before the final softmax.

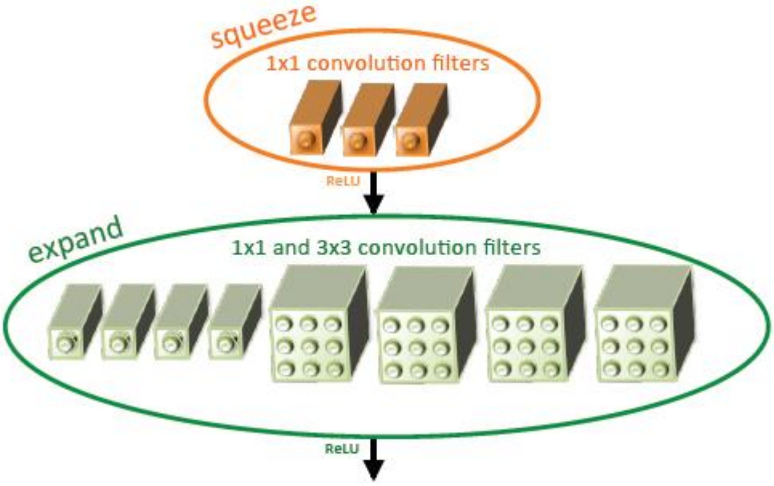


Figure A.12: Scheme of fire module [89].

- **ResNet18** [90]: this CNN, introduced in 2015 inspired by the connection between neurons in the cerebral cortex, uses a residual connection or skip connections which jumps over some layers. With this method it is possible to counteract the problem of degradation of performance as the depth of the net increases, in particular the "vanishing gradient". CNN is made up of 72 layers, the various convolutions are followed by batch normalization and ReLU, while the residual connection exploits an additional layer of two inputs. The last layers consist of an average pooling, a fully connected layer and softmax.

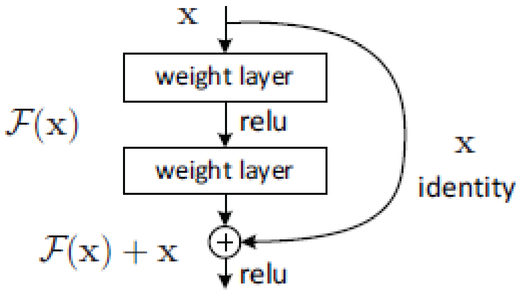


Figure A.13: Scheme of residual learning [90].

- **GoogLeNet** [91]: this architecture is based on the use of "inception modules", each of which includes different convolutional sub-networks subsequently chained at the end of the module. This network is made up of 144 layers, the inception blocks are made up of four branches, the first three with 1x1, 3x3 and 5x5 convolutions, the fourth with 3x3 max pooling. After that, all feature maps at different paths are concatenated together as the input of the next module. The last layers are composed of an average pooling and a fully connected layers and the softmax for the final output.

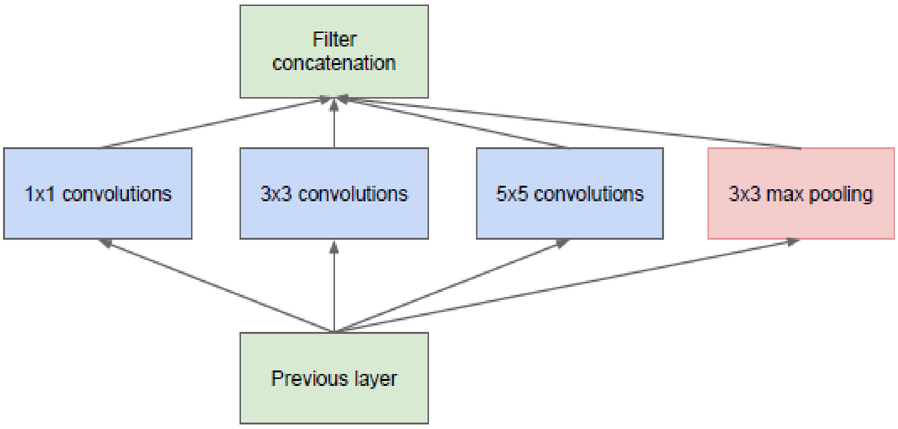


Figure A.14: Scheme of the Inception module [91].

Transfer learning

Since the design and training of a CNN is a complex problem, where exhaustive research cannot be used and training requires large and accurate databases for training and a computing power that is not always accessible to researchers, a well-established way in the literature is to use pre-trained CNN networks. Thanks to ImageNet competition, extremely CNN networks have emerged [92]. These networks have been trained on over a million images for this reason the network has modeled generic rich feature representations.

Pre-trained CNNs can be used considering the following two strategies:

- as feature extractors and coupled to a traditional classifier such as the appropriately trained SVM;
- performing pre-trained CNN transfer learning; in this case, by appropriately replacing the last layer based on the classes to be discriminated, fine-tuning is performed using the database of data to be classified in the training.

Fine-tuning is a transfer learning technique that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [93]. Since CNNs are composed of numerous layers and a huge number of parameters, e.g. AlexNet has 650K neurons and 60M parameters, the network training phase should benefit from the use of databases rich in examples that allow to avoid the problem of overfitting. Unfortunately, it is not always possible to take advantage of such large databases.

Furthermore, training a CNN well on a large database is demanding both in terms of computation time and the computational resources required.

Fine-tuning arises from the need to make up for these deficiencies. This method consists in the possibility of using a Neural Network, pre-trained on a large database, through a further training phase with another database, even a small one. The output level is replaced with a new softmax output level, adjusting the number of classes to the classification problem being faced. The initial values of the weights are used those of the pre-trained network, except for the connections between the penultimate and last level whose weights are randomly initialized. New training iterations are performed to optimize the weights with respect to the peculiarities of the new dataset (it does not need to be large). Fine-tuning can be done in two ways. One way is to freeze the weights of some layers and carry out new training cycles to modify the weights of the remaining layers. The concept of fixing the weights of the layers is defined as freeze of the layers. Generally, they are the first layers to be freeze as the first layers capture low level features. The greater the number of freezed layers, the less the fine-tuning effort in terms of time and resources.

In this case, the weights of the first CNN levels are frozen and the remaining parameters / weights are trained. The other way is to have the architecture re-train entirely on the new database. This method is called training from scratch. It is intuitive that the greater the number of frozen layers the lower the computational cost of training, so training from scratch is the most expensive form of computational training.

Appendix B – Image processing

This paragraph presents, in general, some theories and techniques of image processing, many of which were used in this thesis work.

Spatial domain techniques

Spatial domain techniques directly deal with the image pixels. The pixel values are manipulated to achieve desired enhancement. The processing in the spatial domain can be expressed as:

$$g(x, y) = T[f(x, y)]$$

where *f* is the input image to processing, *g* that of output and *T* an operator on *f* defined in a neighborhood of *(x, y)*. In some cases *T* acts on a set of input images, as in the case of processing image sequences or, more simply, in the case of pixel-by-pixel operations (sums, etc.). The size of the neighborhood of *(x, y)* defines the character of the processing:

- punctual: the neighborhood coincides with the pixel itself;
- local: in the most common cases the neighborhood is a small square region centered on the pixel;
- global: the neighborhood coincides with the whole image *f*.

The result of a homogeneous punctual processing depends only on the value of the pixel to which it is applied, so these processing are also called manipulations of the gray scale. If, on the other hand, the result of the processing also depends on the position of the pixel, it is talking about non-homogeneous point processing.

The homogeneous point processing is carried out by applying a specific operation to each pixel of the starting image, thus a new image is obtained in which each pixel takes on a value that is the result of the operation itself. The homogeneous punctual processing can therefore be represented by a transformation or mapping of the gray levels. A trivial example of a punctual (invertible) operation is the inversion of the gray scale (or negation of the image).

The contrast normalization and the equalization of the histogram are also included among the punctual operations. Given a grayscale image, its histogram consists of the histogram of its gray levels; that is, a graph indicating the number of times each gray level occurs in the image. The (normalized) histogram of the gray levels of a digital image is the discrete function:

$$P(r_k) = n_k / n \quad for \quad k = 0, 1, \ldots, L-1$$

where n_k is the number of pixels of the image with gray level k, n is the total number of pixels and L is the number of gray tones available. Therefore, this is a posteriori estimate of the probability of occurrence of the gray levels of the image, useful because it provides a global description of the image. The information given by the histogram can give a general idea of the possibility of image improvement, especially in terms of contrast manipulation, but are also applied in other processing (compression, segmentation). In a dark image, the gray levels would be clustered at the lower end; in a uniformly bright image, the gray levels would be clustered at the upper end, in a well contrasted image, the gray levels would be well spread out over much of the range.

The contrast normalization enhance the histogram contrast by spreading out its histogram. Normalization is sometimes called histogram stretching and changes the range of pixel intensity values applying a piecewise linear function. If a given histogram has a range between a value a and b both greater than zero and with $b > a$, the linear function to stretch in the new range c and d both greater than zero and $d > c$, operates on each *i-th* pixel producing the new *j-th* value according to the equation:

$$j = (c - d) / (b - a) * (i - a) + c$$

with the condiction that $a < c$ and $d > b$.

Pixel values less than c are all converted to c, and pixel values greater than d are all converted to d. Generally, a $_k$ective information, for this reason a small percentage of the pixels with higher values and those with lower values are discarded. In this thesis, for example, the contrast normalization function has been implemented such that, linearly remapping the intensity values so that 1% of data is saturated at low and high intensities.

Another renowned technique for modifying the histogram is the equalization. This consists in transforming the histogram of gray tones in such a way as to make it as uniform as possible (each level of gray features almost the same number of pixels in the image); a perfectly uniform distribution of gray levels cannot be obtained due to the discrete nature of the quantities involved. The function to be used for the equalization of the histogram in the discrete case is the following:

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} n_j/n \qquad 0 \le r_k \le 1 \quad and \quad k = 0, 1, \ldots, L-1$$

The transformation from each pixel of intensity r_k into a pixel of s_k value occurs through the function $T(r_k)$ which is calculated directly from the histogram of the starting image.

An adaptive variant of histogram equalization is the CLAHE (Contrast Limited Adaptive Histogram Equalization). It is a block-based processing, and it can overcome the over amplification of the noise problem in the homogeneous region of the image with standard histogram equalization. It operates on small regions in the image, called tiles, rather than on the entire image. Each tile's contrast is enhanced, the neighbouring tiles are then combined using bilinear interpolation to eliminate artificially induced boundaries.

Local processing

Local processing has been the most used techniques since the dawn of imaging [94], they are the procedures that somehow emphasize or attenuate certain characteristics of the images, leaving their format unchanged. The objectives can be different, to reduce noise, eliminate cyclic perturbations, increase and intensify the characteristics of the contours present in the image. These techniques are generally achieved through the convolution of a kernel (or spatial filter), that is an $m \times n$ matrix used in convolutions.
If $f(i, j)$ represents an image, with $(i, j) \in (\{1, N_i\} \times \{1, N_j\})$, the *convolution* of f with a kernel K is given by:

$$f_C(i, j) = \sum_{k=1}^{m} \sum_{l=1}^{n} f(i + k - 1, j + l - 1)\, K(k, l)$$

The convolution mask generally has a rectangular shape (usually with sides of odd length). Spatial filtering requires 3 steps: position the mask over the current pixel, form all products of filter elements with the corresponding elements of the neighborhood, and add up all the products. This must be repeated for every pixel in the image.
It speaks of spatial filtering as the operation takes place directly in the pixel domain or spatial domain, while the concept of filtering has its roots in the use of the Fourier transform, in the frequency domain.
Among the most used spatial filters are smoothing ones. Smoothing filters are used for image blurring and for noise reduction. The blurring operation is normally used in the pre-processing phase, in order to eliminate small unnecessary or even harmful details for subsequent processing. A simple example of a smoothing filter is the average filter: it replaces the average of the pixel values in the neighborhood defined by the Kernel for each pixel; in this way the extent of the gray tone differences between neighboring points is reduced. It can be accomplished using the following Kernel:

$$k = 1/9 * \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

Among the most used smoothing filters is the Gaussian filter. Gaussian filters are a class of low-pass filters, all based on the Gaussian probability distribution function. To build the kernel you need to specify the σ which is the standard deviation: a large value of σ produces a flatter curve, and a small value σ leads to a more “pointed” curve. Gaussian blurring is highly effective in removing Gaussian noise from an image.
Another filter widely used to remove noise from images and especially the salt and pepper type, is the median filtering. The function takes the median of all the pixels under the kernel area and the central element is replaced with this median value. Interestingly, in the gaussian filters, the central element is a newly calculated value which may be a pixel value in the image or a new value. But in median blurring, the central element is always replaced by some pixel value in the image. The median filter

is not one of the fastest filters due to the search for the median between a series of values. Therefore it is important to use efficient algorithms that exploit the overlapping convolutions.

Another type of filtering used in this thesis is bilateral filtering [95]. The Bilateral filtering is highly effective in noise removal while keeping edges sharp. But the operation is slower compared to other filters. It already saw that a Gaussian filter takes the neighbourhood around the pixel and finds its Gaussian weighted average. This Gaussian filter is a function of space alone, that is, nearby pixels are considered while filtering. It doesn't consider whether pixels have almost the same intensity. It doesn't consider whether a pixel is an edge pixel or not. So it blurs the edges also, which it doesn't want to do. Bilateral filtering also takes a Gaussian filter in space, but one more Gaussian filter which is a function of pixel difference. The Gaussian function of space makes sure that only nearby pixels are considered for blurring, while the Gaussian function of intensity difference makes sure that only those pixels with similar intensities to the central pixel are considered for blurring. So it preserves the edges since pixels at edges will have large intensity variation.

Similar to the bilateral filter, another filter that allows noise reduction without smoothing the edges is the anisotropic diffusion filter, also called Perona-Malik diffusion [96].

Morphological filters

Another approach to filtering is adapted from the mathematical morphology. Morphological processing is a type of processing that changes the shape or structure space of the objects in the image. Although morphologic operators can also be defined for gray scale images, morphologic filtering is principally performed on binary input images [97]. According to a general convention, the white pixels in the binary image indicate relevant segments and the black pixels indicate the background.

Typical morphological operations are: dilation, erosion, skeletonization.

Before defining the morphological operators it is necessary to define the type of connectivity of the objects on a discrete grid. Given a pixel *p* of coordinates (x, y), the four neighbors (horizontally and vertically) have coordinates:

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$$

and constitute the set $N_4(p)$ of the 4-neighbors of *p*.

The four diagonal neighbors of *p* have coordinates:

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)$$

and form the set $N_D(p)$.

The set of 8-neighbors of p is given by $N_8(p) = N_4(p) \cup N_D(p)$.

On the basis of the spatial adjacency relationship used, the type of connectivity is defined (see Figure B.1):

Morphological operators of erosion and dilation: these two operators are fundamental in the morphological analysis of images; in fact, most of the morphological analysis techniques are obtained starting from these. Erosion and expansion are carried out through a structural element B (a $n \times m$ matrix) and, given an object A, are defined as follows:

- Erosion of A: the set of points included in all possible translations of B that are completely internal to A (see Figure B.2);
- Dilation of A: the set of points included in all possible translations of B for which at least one point of B is included in A (see Figure B.2).

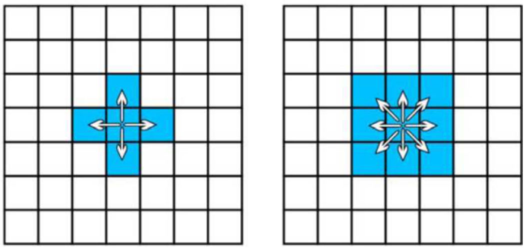


Figure B.1: 4-connectivity (left) 8-connectivity (right). The arrows identify the neighboring pixels starting from the central pixel.

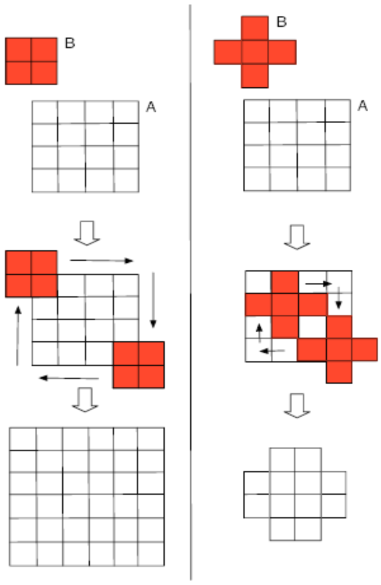


Figure B.2: Erosion and Dilation. On the left, the application of a dilation; on the right the application of an erosion. A is the original object, and B the structural elements used.

- Opening of *A*: the combination of erosion followed by dilation is called opening, in reference to its ability to separate shapes that touch each other (see Figure B.3); this operator is widely used to remove noise pixels from binary images.
- Closure of *A*: the opposite combination, dilation followed by erosion, is called closure in reference to its ability to unite neighboring forms; this operator is widely used to remove isolated background pixels from binary images.
- Thinning and Skeletonization: these techniques consist in reducing a region to a minimal set of points representing an invariant of its geometric form. It is called skeleton [98]. The strict definition of skeleton is due to Blum [99], who introduced the Medial Axis Tranform (MAT) defining the skeleton in the following way: if *A* is an object in an image, the skeleton of *A* is the subset *I* of *A* such that, if $x \in I$ then there exist at least two points on the edge of *A* equidistant from *x*.

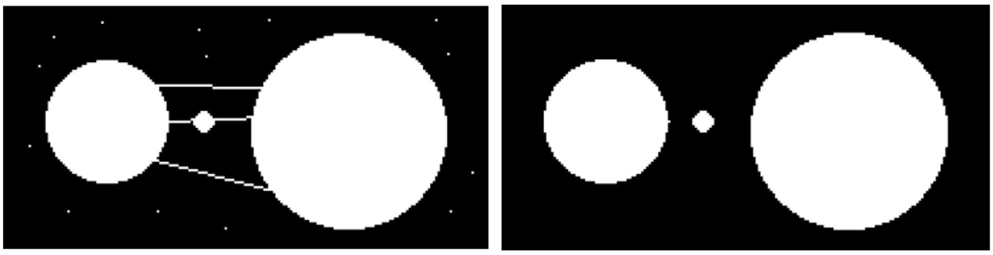


Figure B.3: Opening example. 4-connectivity, one cycle (one erosion + one dilation).

The erosion reduces the size of a segment, and the dilation leads to its enlargement. The opening removes small details on the outline of segments or the background, without affecting the total size of relevant regions. The closing is able to remove holes in the interior of a region and smooth its contour.

Il filtro FAS (Filter Alternating Sequential) is obtained by the iterative subsequent application of morphological opening and closing transformations and process an image by filtering both bright and dark structures. FAS is widely used for achieving a simplification of a scene and for the removal of noisy structures.

Binary morphology is applied frequently in medical image processing, for instance to clean up shapes after pixel-based segmentation.

Gray scale morphology is simply a generalization from 1 bit (binary) images to images with multiple bits per pixel, where MIN and MAX operations replace the AND and OR operations of binary morphology, respectively.

Thresholding

The histogram thresholding assumes that the image is composed of regions that differ from each other in the range of values of the pixels that compose them, so that the histogram of an image has peaks corresponding to these values. The values that correspond to valleys in the histogram are used as thresholds to distinguish the regions from each other. Thresholding can be used to create binary images, indeed, the method replace each pixel in an image with a black pixel if the image intensity is less than some fixed constant *T* (the threshold value), or a white pixel if the image intensity is greater than that constant. To make thresholding completely automated, it is necessary for the computer to automatically select the threshold *T*. Among the most known algorithms to automatically find the T value there are the Otsu threshold [57] and the threshold based on max entropy [58]. The Otsu algorithm returns a single intensity threshold that separate pixels into two classes, foreground and background. This threshold is determined by minimizing intra-class intensity variance, or equivalently, by maximizing inter-class variance.

The Max entropy thresholding is very similar to Otsu's method, rather than maximising the inter-class variance, it maximises the inter-class entropy. Entropy is a measure of the uncertainity of an event taking place. Let h_i be value of normalized histogram

Entropy of black pixels: $H_B(t) = -\sum_{i=0}^{t} [h_i / \sum_{j=0}^{t} h_j] \log [h_i / \sum_{j=0}^{t} h_j]$

Entropy of white pixels: $H_W(t) = -\sum_{i=t+1}^{i\_max} [h_i / \sum_{j=t+1}^{i\_max} h_j] \log [h_i / \sum_{j=t+1}^{i\_max} h_j]$

Optimal threshold can be selected by maximizing the entropy of black and white pixels:

$T = Arg\ Max_{t=0 \ldots i\_max}\ H_B(t) + H_W(t)$

A global threshold on the histogram of the whole image might not be good in all cases, e.g. if an image has different lighting conditions in different areas. In that case, adaptive thresholding can help. In this thesis is considered the adaptive variant of the Otsu algorithm. The algorithm determines the threshold for a pixel based on a small region around it. So it gets different thresholds for different regions of the same image which gives better results for images with varying illumination.

Used processes list

Table B.1 lists the processes used in this thesis work.

Processes	Abbreviation	Reference
Nothing	Nt	-
Contrast normalization	Cn	implemented
Equalization	Eq	http://docs.opencv.org
CLAHE	Ch	http://docs.opencv.org
Gaussian filter	Gs	http://docs.opencv.org
Median filter	Md	http://docs.opencv.org
Morphological filters	Dl / Er / Op / Cl	http://docs.opencv.org
Morphological FAS	Fs	http://www.pkuwwt.tk/ofeli/doc/index.html
Anisotropic diffusion filter	An	http://www.pkuwwt.tk/ofeli/doc/index.html
Bilateral filter	Bl	http://docs.opencv.org
Otsu thresholding	Ot	http://docs.opencv.org
Max entropy thresholding	Me	implemented
Adaptive Otsu thresholding	Ao	http://docs.opencv.org
Remove boundary cells	Rb	implemented
Filling	Fl	implemented

Table B.1: Processing functions used.

Appendix C – Features

Used features list

Table C.1 lists of the features used in this thesis work.

Features	Short description
Mean value	Calculate the average value of the pixels intensity
Standard deviation	Calculate the standard deviation of the pixels intensity
Ratio of the standard deviation to the mean value (intensity)	Calculate the value of the ratio between the sigma and the mean of the intensity
Entropy	Calculate the value of the mean entropy of the pixels intensity
Skewness	Evaluate the asymmetry index (Skewness) of the pixels intensity distribution
Kurtosis	Evaluate the Kurtosis value associated with the pixels intensity distribution
Area	Evaluate the ROI area
Perimeter	Evaluate the ROI perimeter
Convex area	Evaluate the convex hull area of the ROI
Mean radius	Calculate the average value of the radius
Standard deviation of radius	Calculate the standard deviation of the radius
Ratio of the standard deviation to the mean value (radius)	Calculate the value of the ratio between the sigma and the mean of the radius
Maximum radius	Calculate the maximum value of the radius
Anisotropy	Calculate the anisotropy value
Entropy of the contours gradient	Calculate the entropy of the contours gradient
Fractal index	Evaluate the fractal index to be associated with the ROI
Eccentricity	Evaluate the eccentricity of the ROI to get a measure of the elongation of the region
Circularity	Calculate the circularity of the ROI (4*π*area / perimeter^2)
Contrast	Calculate the contrast of the ROI
Convex deficiency	Evaluate the convex deficiency area of the ROI
Roundness	Evaluate the roundness of the ROI (4*π*area / perimeter)
Compactness	Evaluate the compactness of the ROI (4*π*area / perimeter^2)
Solidity	Evaluate the solidity of the ROI (area / convex_area)
Inertia of co-occurrence matrix	Calculates the inertia of the co-occurrence matrix of the gray levels
Entropy of HOG	Calculate the entropy value of the HOG descriptor (Histogram of Oriented Gradients)
Entropy of HAG	Calculate the entropy value of the HAG descriptor (Histogram of Amplitude Gradients)
Euler's number	Evaluate the euler number of the ROI (object number – holes number)

Table C.1: List of the features used.

References

[1] J.H. Klippel, P.A. Dieppe, **Rheumatology, 2nd edition**, Mosby International, **1998.**

[2] A. Kavanaugh, R. Tomar, J. Reveille, et al, **Guidelines for Clinical Use of the Antinuclear Antibody Test and Tests for Specific Autoantibodies to Nuclear Antigens**, American College of Pathologists, Archives of Pathology and Laboratory Medicine, Vol. 124, No.1, pp. 71-81, **2000**.

[3] R. Marcolongo et al, **Presentazione Linee Guida del Forum Interdisciplinare per la Ricerca sulle Malattie Autoimmuni (F.I.R.M.A.)**, Reumatismo, Vol. 55, pp. 9-21, **2003**.

[4] K. Conrad, R-L. Humbel, et al, **Autoantigens and autoantibodies: diagnostic tools and clues to understanding autoimmunity**, Berlin: Pabst Science Publisher, **2000**.

[5] U. Sack, et al, **Standardization of Autoimmune Diagnostics in Germany. Activities of the German Group in the European Autoimmune Standardization Initiative**, Ann. N.Y. Acad. Sci. 1109: 31–36, **2007**.

[6] U. Sack, K. Conrad, et al, **Autoantibody Detection Using Indirect Immunofluorescence on HEp-2 Cells**, Contemporary Challenges in Autoimmunity: Ann. N.Y. Acad. Sci. 1173: 166–173, **2009**.

[7] P. Soda, G. Iannello, **ANN-based classification of immunofluorescence images,** Enformatika Transactions on Engineering, Computing and Technology, 14:252*257, **2006**.

[8] P. Soda, G. Iannello, **A multi-expert system to classify fluorescent intensity in antinuclear autoantibodies testing**, In Computer Based Medical, Systems, pages 219-224, Los Alamitos, CA, USA, IEEE Computer Society, **2006**.

[9] M. J. Fritzler, **Toward a new autoantibody diagnostic orthodoxy: understanding the bad, good and indifferent**, Autoimmunity Highlights, vol. 3, no. 2, pp. 51–58, **2012.**

[10] R. Tozzoli, N. Bizzaro, E. Tonutti, et al, **Guidelines for the laboratory use of autoantibody tests in the diagnosis andmonitoring of autoimmune rheumatic diseases**, American Journal of Clinical Pathology, vol. 117, no. 2, pp. 316–324, **2002.**

[11] N. Bizzaro, A. Wiik, **Appropriateness in anti-nuclear antibody testing: from clinical request to strategic laboratory practice**, Clinical and Experimental Rheumatology, vol. 22, no.3, 349–355, **2004.**

[12] The Autoimmune Diseas Coordinating Committee, **Progress in Autoimmune Diseases Research**, NationalInstitutes of Health: Bethesda, MD, USA, 2005; pp. 1–146, **2018.** (Available online: https://www.niaid.nih.gov/sites/default/files/adccfinal.pdf)

[13] N. Bizzaro, A. Antico, S. Platzgummer, et al, **Automated antinuclear immunofluorescence antibody screening: a comparative study of six computer-aided diagnostic systems**, AutoimmunityReviews, vol. 13, no. 3, pp. 292–298, **2014.**

[14] AS. Wiik, et al, **Antinuclear antibodies: A contemporary nomenclature using HEp-2 cells,** Autoimmun 2010; 35: 276-90, **2010.**

[15] PL. Meroni, et al, **ANA screening: an old test with new recommendations**, Ann Rheum Dis 2010; 69: 1420-2, **2010.**

[16] N. Agmon-Levin, J. Damoiseaux, C. Kallenberg, et al, **International recommendations for the assessment of autoantibodies to cellular antigens referred to as anti-nuclear antibodies**, Annals of the Rheumatic Diseases, vol. 73, no. 1, pp. 17–23, **2014.**

[17] NCCLSI/LA2-A: '**Quality assurance for the indirect immunofluorescence test for autoantibodies to nuclear antigen (IF-ANA): Approved guideline**', 1996 Waine, PA, 16(11), **1996.**

[18] Morag Crichton Timbury, '**Notes on medical virology',** Churchill Livingstone, p. 27. ISBN 978-0-443-04872-2, Retrieved 18 March **2012.**

[19] E.K.L. Chan, et al, **Report of the First International Consensus on Standardized Nomenclature of Antinuclear Antibody HEp-2 Cell Patterns**, ICAP 2014-15, Front. Immunol., Aug 20;6:412, **2015.**

[20] E.K.L. Chan, et al, **The International Consensus on Standardized Nomenclature of Antinuclear Antibody HEp-2 Cell Patterns**, ICAP initiative - Current state and perspectives. From Autoantibody Research to Standardized Diagnostic Assays in the Management of Human Diseases series, Report on the 12th Dresden Symposium on Autoantibodies, Germany, p282-288, **2017.**

[21] M. Herold, et al, **International Consensus on Antinuclear Antibody Patterns: defining negative results and reporting unidentified patterns**, Clin Chem Lab Med. 2018;56:1799-802, **2018.**

[22] K. Conrad, et al, **Autoantibodies in systemic autoimmune diseases: a diagnostic reference**, 3 th edn. Autoantigens autoantibodies autoimmunity, **2015.**

[23] JD. Northway, EM. Tan, **Differentiation of antinuclear antibodies giving Speckled staining patterns in immunofluorescence,** Clin Immunol Immunopathol 1972; 1:140–54, **1972.**

[24] H. Imai, et al, **Nucleolar antigens and autoantibodies in hepatocellular carcinoma and other malignancies,** Am J Pathol 1992; 140:859–70, **1992.**

[25] R. Tozzoli, A. Antico, B. Porcelli, D. Bassetti, **Automation in indirect immunofluorescence testing: a new step in the evolution of the autoimmunology laboratory**, Autoimmunity Highlights, 3:59–65, **2012.**

[26] M. Fenger, et al, **Detection of antinuclear antibodies by solid-phase immunoassays and immuno-fluorescence analysis**, Clin Chem 2004 Nov; 50(11):2141–7, **2004.**

[27] BN. Pham, et al, **Impact of external quality assessment on antinuclear antibody detection performance**, Lupus 2005;14:113–9, **2005.**

[28] L. Song, et al, **Photobleaching kinetics of fluorescein inquantitative fluorescence microscopy**, Biophys J 1995; 68:2588–600, **1995.**

[29] A. Willitzki, et al, **New platform technology for comprehensive serological diagnostics of autoimmune diseases,** Clin. Dev. Immunol. 2012, 284740, **2012.**

[30] http://mivia.unisa.it/datasets/biomedical-image-datasets/hep2-image-dataset/

[31] P. Foggia, et al, **Benchmarking HEp-2 cells classification methods**, IEEE Transactions on Medical Imaging 32,1878–1889, **2013.**

[32] https://hep2.unisa.it/dbtools.html

[33] P. Hobson, et al, **Competition on cells classification by fluorescent image analysis**, in: Proc. 20th IEEE Int. Conf. Image Process. ICIP, pp. 2–9, **2013.**

[34] P. Hobson, et al, **Benchmarking human epithelial type 2 interphase cells classification methods on a very large dataset**, Artificial Intelligence in Medicine 65, 239–250, **2015.**

[35] BC. Lovell, et al, **International contest on pattern recognition techniques for indirect immunofluorescence images analysis,** in: 23rd International Conference on Pattern Recognition ICPR, IEEE. pp. 74–76, **2016.**

[36] E. Bennamar, et al, **Computer-assisted classification patterns in autoimmune diagnostics: the A.I.D.A. Project**, BioMed Res. Int. 2016, 1–9, **2016.**

[37] http://www.aidaproject.net/

[38] J. Yanase, et al, **A Systematic Survey of Computer-Aided Diagnosis in Medicine: Past and Present Developments,** Expert Systems with Applications. 138: 112821, **2019.**

[39] M. Thomas, **Biomedical Image Processing**, Springer **2011.**

[40] D. Klaus Toennies, **Guide to medical image analysy. Methods and algorithms**, Springer **2012.**

[41] L. LU, et al, **Deep learning and convolutional neural networks for medical image computing,** Advances in Computer Vision and Pattern Recognition, **2017.**

[42] H-C. Shin, et al, **Three Aspects on Using Convolutional Neural Networks for Computer-Aided Detection in Medical Imaging,** Deep Learning and Convolutional Neural Networks for Medical Image Computing, Springer, Cham, 113-136, **2017.**

[43] JA. Swets, **Measuring the accuracy of diagnostic systems,** Science 240.4857 1285-1293, **1988.**

[44] JA. Hanley, et al **The meaning and use of the area under a receiver operating characteristic (roc) curve,** Radiology, 143:29–36, **1982.**

[45] P. Pemer et al, **Texture Classification based on the Boolean Model and its Application to Hep-2 Cells,** 1051-465U02 Q 2002 IEEE, **2002.**

[46] P. Perner, et al, **Mining knowledge for HEp-2 cell image classification**, Journal Artificial Intelligence in Medicine, 26:161*173, **2002.**

[47] R. Hiemann, et al, **Automatic analysis of immunofluorescence patterns of HEp-2 cells**, Annals of the New York: Academy of Sciences, 1109(1):358—371, **2007.**

[48] Yu-Len Huang, et al, **Adaptive Automatic Segmentation of HEp-2 Cells in Indirect Immunofluorescence Images**, 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, IEEE Computer Society, **2008.**

[49] Yu-Len Huang, et al, **Outline Detection for the HEp-2 Cell in Indirect immunofluorescence Images Using Watershed Segmentation**, 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 2008, IEEE Computer Society, **2008.**

[50] CC. Cheng, et al, **Segmentation of Anti-nuclear Antibody Images Based on the Watershed Approach**, 5th IEEE Conference on Industrial Electronics and Applicationsis, **2010.**

[51] P. Elbischger, et al, **Algorithmic framework for HEp-2 fluorescence pattern classification to aid auto-immune diseases diagnosis,** IEEE International Symposium on Biomedical Imaging: From Nano to Macro, IEEE, **2009.**

[52] S. Roy, et al, **A Modified Rough-Fuzzy Clustering Algorithm with Spatial Information for HEp-2 Cell Image Segmentation**, IEEE International Conference on Bioinformatics and Biomedicine (BIBM), **2016.**

[53] CC. Cheng, et al, **Design of a Computer-Assisted System to Automatically Detect Cell Types Using ANA IIF Images for the Diagnosis of Autoimmune Diseases**, J Med Syst, 39 (117), **2015.**

[54] G. Percannella, et al, **A classificationbased approach to segment HEp-2 cells**, 25th international symposium on computer-based medical systems (CBMS), pp. 1-5, **2012.**

[55] S. Tonti, et al, **An automated approach to the segmentation of HEp-2 cells for the indirect immunofluorescence ANA test**, Computerized Medical Imaging and Graphics, 40, pp. 62–69, **2015.**

[56] LR. Dice, **Measures of the amount of ecologic association between species,** Ecology, 26: pp. 297–302, **1945.**

[57] N. Otsu, **A threshold selection method from gray-level histograms**, IEEE transactions on systems, man, and cybernetics 9.1 (1979): 62-66, **1979.**

[58] S. Sahoo, et al, **A Survey of Thresholding Techniques**, Computer Vision, Graphics, and Image Processing, Vol. 41, pp.233-260, **1988.**

[59] A. Cosmin, et al, **Randomized Hough Transform for Ellipse Detection with Result Clustering**, In Proceedings of the IEEE EUROCON, International Conference on Computer as a Tool, Belgrade, Serbia and Montenegro, 21–24 November 2005; pp. 1397–1400, **2005.**

[60] TF. Chan, et al, **Active Contours Without Edges,** IEEE Transactions on image processing, vol. 10, no. 2, **2001.**

[61] M. Merone, et al, **A computer-aided diagnosis system for HEp-2 fluorescence intensity classification**, Artificial intelligence in medicine 2019, vol. 97, pp. 71-78, **2019.**

[62] S. Di Cataldo, et al, **ANAlyte: A modular image analysis tool for ANA testing with indirect immunofluorescence**, Comput. Methods Programs Biomed., 128, 86–99, **2016.**

[63] G. Iannello, et al, **A slightly supervised approach for positive/negative classification of fluorescence intensity in hep-2 images**, In International Conference on Image Analysis and Processing (pp. 319-328). Springer 2013, Berlin, Heidelberg, **2013.**

[64] L., Zhou, et al, **Positive and negative HEp-2 image classification fusing global and local features,** 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). IEEE, **2017.**

[65] S. Ensafi, et al, **A bag of words based approach for classification of hep-2 cell images**, Pattern Recognition Techniques for Indirect Immunofluorescence Images (I3A), 1st Workshop, pp. 29-32, **2014.**

[66] S. Manivannan, et al, **An automated pattern recognition system for classifying indirect immunofluorescence images of HEp-2 cells and specimens**, Pattern Recognition, 51, **2016.**

[67] R. Nosaka, et al, **HEp-2 cell classification using rotation invariant co-occurrence among local binary patterns**, Pattern Recognition, 47 (7), pp. 2428-2436, **2014.**

[68] Z. Gao, et al, **HEp-2 cell image classification with deep convolutional neural networks**, IEEE Journal of Biomedical and Health Informatics, vol. 21, pp. 416-428, **2016.**

[69] H. Li, et al, **HEp-2 specimen classification via deep CNNs and pattern histogram**, In ICPR 2016, 23rd International Conference on Pattern Recognition, **2016.**

[70] Z. Oraibi, et al, **Learning Local and Deep Features for Efficient Cell Image Classification Using Random Forest,** In ICIP 2018, 25th International Conference on Image Processing, **2018.**

[71] X. Xu, et al, **Automated classification for HEp-2 cells based on linear local distance coding framework,** EURASIP J. Image Video Process, **2015**.

[72] J. Furnkranz, **Round Robin Classification**, Journal of Machine Learning Research, 721-747, **2002.**

[73] R. Debnath, et al, **A decision based one-against-one method for multi-class support vector machine**, Pattern Anal. Appl. 7 (2) 164–175, **2004.**

[74] C. Cortes, et al, **Support-vector networks**, Machine learning, 20(3), 273-297, **1995.**

[75] A. Astorino, **Classificazione Binaria e Support Vector Machines,** Istituto di calcolo e reti ad alte prestazioni C.N.R., c/o Dipartimento di Elettronica Informatica e Sistemistica, Università della Calabria, Rende (CS). Online Available:
http://dns2.icar.cnr.it/manco/Teaching/2006/datamining/lezioni/classificazione_binaria_SVM.pdf

[76] CC. Chang, et al, **LIBSVM: a Library for Support Vector Machines**, Department of Computer Science National Taiwan University, Taipei, Taiwan, **2001.**

[77] J.C. Burges, **A Tutorial on Support Vector Machines for Pattern Recognition**, Data Mining and Knowledge Discovery, 2, 121-167, **1998.**

[78] G. Dreyfus, **Neural Networks – Methodology and Applications,** Springer-Verlag Berlin Heidelberg, **2005.**

[79] **MATLAB NEURAL NETWORK toolbox**; Natick, Massachusetts: The MathWorks Inc.

[80] NJ. Nilsson, **Artificial Intelligence: a New Synthesis,** Morgan Kaufman Publishers, **1998.**

[81] J. M. Johnson, T. Khoshgoftaar, **Survey on deep learning with class imbalance**, Journal of Big Data 2019, 6(1), 27, **2019.**

[82] M. Abadi, et al, **TensorFlow: large-scale machine learning on heterogeneous systems**, Software available from tensorflow.org. **2015.** URL https://www.tensorflow.org

[83] A. Krizhevsky, et al, **Imagenet classification with deep convolutional neural networks,** In Advances in neural information processing systems 2012 (pp. 1097-1105), **2012.**

[84] Y. LeCun, et al, **Handwritten digit recognition with a back-propagation network**, In Proc. Advances in Neural Information Processing Systems, 396–404, **1990.**

[85] D.H. Hubel, et al, **Receptive fields and functional architecture of monkey striate cortex**, The Journal of physiology, 1968, 195(1), 215-243, **1968.**

[86] Y. LeCun, et al, **Gradient-based learning applied to document recognition**, Proceedings of the IEEE 1998, 86(11), 2278-2324, **1998.**

[87] HC. Shin, et al, **Three Aspects on Using Convolutional Neural Networks for Computer-Aided Detection in Medical Imaging**, Deep Learning and Convolutional Neural Networks for Medical Image Computing. Springer, Cham, 113-136, **2017.**

[88] P. Kim, **Matlab deep learning With Machine Learning,** Neural Networks and Artificial Intelligence 130, 21, **2017.**

[89] FN. Iandola, et al, **Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size,** arXiv 2016, arXiv:1602.07360, **2016.**

[90] K. Simonyan, A. Zisserman, **Very Deep convolutional networks for large-scale image recognition,** arXiv 2014, arXiv:1409.1556, **2014.**

[91] C. Szegedy, et al, **Going deeper with convolutions,** In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, USA, 7–12 June, **2015.**

[92] O. Russakovsky, et al, **ImageNet Large Scale Visual Recognition Challenge**. IJCV 2015, 115, 211–252, **2015.**

[93] W. Jeremy, et al, **Spring Research Presentation: A Theoretical Foundation for Inductive Transfer**, College of Physical and Mathematical Sciences, Posted: **2007.**

[94] WK. Pratt, **Digital image processing**, New York: Wiley-Interscience, **1978.**

[95] C. Tomasi, et al, **Bilateral filtering for gray and color images,** Sixth International Conference on Computer Vision. Bombay. pp. 839–846, **1998.**

[96] P. Perona, et al, **Scale-space and edge detection using anisotropic diffusion**, Proceedings of IEEE Computer Society Workshop on Computer Vision. pp. 16–22, **1987.**

[97] TM. Deserno, **Biomedical image processing–biological and medical physics**, biomedical engineering, **2001.**

[98] JL. Davidson, **Thinning and skeletonization**, In Digital Image Processing Methods, Marcel Derrek Inc. New York, Basel, Hong Kong, **1994.**

[99] H. Blum, **A trasformation for extracting new description of shape**, In Models for the Perception of Speech and Visual Forms, MIT Press, Cambrige, Mass, **1967.**