POLITECNICO DI TORINO

MASTER's Degree in ICT for Smart Societies

MASTER's Degree Thesis

Physical object identification using images of cholesteric liquid crystals

Supervisors

Prof. TIZIANO BIANCHI

Prof. GABRIELE LENZINI

Candidate

LAURA LAVEZZA

July 2022

Abstract

This thesis deals with a cutting-edge research topic in information science: the possibility of using random patterns formed by tiny spheres of liquid crystals integrated in common materials, as an innovative solution for physical object identification. The aim of the thesis is to generate and analyze images of these liquid crystals, which, due to their optical properties, generate unpredictable colored patterns, and test model-based and data-driven approaches for the recognition of the said crystals. Part of the activities of the thesis has been conducted in the labs of the Center for Security, Reliability, and Trust (SnT) (https://wwwen.uni.lu/snt) of the University of Luxembourg, under the co-supervision of Dr. Gabriele Lenzini.

Acknowledgements

I would like to thank professor Tiziano Bianchi and Gabriele Lenzini for the endless support and guidance provided to me for the whole durantion of the thesis and especially for the chance they gave me by letting me participate in this project. It was a great opportunity to challenge myself with a real world research work and especially to meet many people who enriched me in different ways. A special mention in this regard goes to the women and the people who work in the IRISC group, who welcomed me with kindness and made me feel part of the team even if it was a limited amount of time. I would like to thank Monica Arenas, PostDoc in professor Lenzini's team, who gave me great advices during the internship and who became a person I admire and look up to. A personal acknowledgement must be given to my family, who supported me and made this abroad experience possible, and to my friends and colleagues who emotionally supported me during the whole master. I could have not gone through the several breakdowns related to the exams and the endless online calls for the group projects, if it was not for them and their friendship. Finally, I would like to use this space to thank my significant other, Hannah, for her love and support. Thank you for encouraging me to get out of my comfort zone in any aspect of my life and thank you for being by my side at every step.

Table of Contents

List of Tables

List of Figures

Acronyms and expressions

CSR

Cholesteric Spherical Reflectors

DIFFERENT TAG PAIRS

abbreviation to refer to pairs of images representing different tags

SAME TAG PAIRS

abbreviation to refer to pairs of images representing the same tag

NCC

Normalised correlation coefficient

TPR

True positive rate

FPR

False positive rate

Chapter 1

Introduction

Counterfeiting consists in the production and selling of unauthorized copies of items. According to the OECD summary released on 2007, illegal copies were already a reality in several industries: from the ones responsible for the production of luxury items, such as watches and jewelry, to those that sell consumer goods that can negatively impact the personal health and safety of an individual [1]. Hence, it is not surprising to read that counterfeiting is considered one of the transnational crimes that hurt our society the most [2]. For instance, two of the major drawback of counterfeiting are the undeniable support that this activity provide to the enrichment of criminal organizations and the consequent lost in tax revenue that could have been invested in the improvement of public services [3]. Even though consumers are well aware of the risks and consequences of buying non-original products, OECD studies from different years have reported that the trade of counterfeited products is a growing trend [4]. A very interesting technology that can be used to reduce the counterfeiting spread in the industrial world, is the one currently under development in the University of Luxembourg. It consists of tags of mixed chemical material that can be used on items, such as jewelry, and should create unique patterns to discriminate the original from an illegal copy [5].

1.1 Objective

This thesis aims at image generation and analysis of the latest version of said tags and the investigation of both data-driven and a model-based methods for tag recognition. Thanks to the collaboration of the Physics and Materials Science and the SnT departments, it was possible to produce different samples of the tags and have a large variety of microscopes and choose the one that allowed the collection of the highest quality data possible. The collection of six acquisition sets allowed to build 3 datasets that differ only in the image resolution. The wish to test the two approaches on scenarios close to the future application of this innovative technology supports the decision to create datasets with different resolutions. Indeed, one of the possible future applications could be phone scanning the tag applied on the item as if it was a QR code. This project tests the two algorithms to see how they respond to the resolution change. The model-based algorithm computes the Normalised correlation coefficient over pairs of images. On the other hand, the data-driven method involves a simple Siamese neural network based on convolutional networks.

1.2 Thesis Layout

The thesis is structured as follow:

- **Chapter 1**: *Introduction*. Here the motivations and the objectives of the project are presented together with a brief summary of the activities carried out during the thesis;
- **Chapter 2**: *Background*. This section aims at providing basic information about the notions and concepts used for the development of the project. Particular attention is given to the Cholesteric Spheric Reflectors (CSR) and the use of Deep Learning for the recognition task;
- **Chapter 3**: *Data acquisition*. The considerations and the challanges faced during the collection of the tag acquisitions are discussed.
- **Chapter 4**: *Methodology*. Here the methodologies used to carry out tests over the set of acquired images are examined. The chapter explores the data-driven and model driven approaches chosen for the project, the used data pre-processing procedures and performance metrics.
- **Chapter 5**: *Experiments and results*. This chapter provides a guide through the experiments made and a comment of the results.
- **Chapter 6**: *Conclusions and future work*. It contains some final conclusions and considerations on the possible steps to take in the future of neural networks applied to the CSRs.

Chapter 2

Background

The following chapter provides a brief overview of the literature to allow the reader to understand what the project foundations are. The main topics faced in this chapter are:

- Cholesteric spheric reflectors;
- Deep Neural Network
- Residual Network;
- Siamese Network;

2.1 Cholesteric Spheric Reflectors

Cholesteric Spheric Reflectors, also called CSRs, are an innovative technology in the security field consisting of tiny microspheres of liquid crystals which are able to generate unique colored pattern distributions [5]. The phenomenon that occurs and is responsible of the said colored patterns is known in nature as *Bragg reflection* [6] which induces a spontaneous spatial organization of the crystals in a helical arrangement, in the orientation of the helix axes [7]. The production process is easily explained, also through visual support, by Lenzini et al. in *Security in the Shell An Optical Physical Unclonable Function made of Shells of Cholesteric Liquid Crystals*. It shows how one of the properties that makes the crystals suited for industrial applications is their malleability [6]. This allows a relatively fast and massive production of tags by mean of the low cost process shortly presented in the mentioned paper. Here are discussed the main steps of the production process. It consists of two main phases:

1. Inner phase. Here the crystals are piped out (LC phase) while surrounding a drop of water. The shape of the molecule depends on this procedure;

2. Outer phase. This phase happens concurrently with the Inner phase. Here, the medium where the molecules will be suspended flows outside of the small bottle used for the production process.

Within the bottles, the crystals move until the solidification of the medium occurs. At this point, the molecules are fixed in their position [6]. By disposing the mixture over a flat surface, like glass or a plastic film, it was possible to obtain the tags used for this thesis. The two types of CSR that were considered for the tag production of this work were shells and droplets. According to the team in charge of the tag production, the main difference between them is the refraction angle in the Bragg's phenomena and the difficulty of the production process [8]. However, regardless of the CSR type, it was observed that the two controllable factors in the production process are the size of the crystals and the selective reflection coming from CSRs, while the spatial and color distribution are uncontrollable [5], [9]. The unpredictability of the molecules arrangements make the tags hardly clonable. This makes them potentially good optical Physical Unclonable Function (PUFs) [9],[10]. In information security, PUFs have been the focus of many studies because of the increasing need for high-security keys [11]. Despite the usual way of storing security keys, Physical objects that are PUF provide the advantage on not storing the key in a memory, but the object itself becomes the key [12]. PUF is described by Gao et al. as a device that exploits intrinsic randomness, that comes from the production process, to give a physical object, such as a tag of material, an unclonable fingerprint [13]. Literature counts many works investigating the use of PUFs in anticounterfeiting applications [14], [15]. Devadas et al. proposed PUF-based RFIDs as inexpensive and fast way of providing authentication and secure access. However, the RFIDs proposed by the paper are silicon based, which means that, as pointed out by Arenas et al., the function is embedded into the hosting medium. Optical PUFs based on CSRs should solve the security issue of silicon-based PUFs, making the not clonable feature not based on the unpredictable interaction of the cholestheric crystals with light. Besides, against the common expectations for optical PUFs [16], this new technology is made of low-priced materials that allow massive series production of tags at reasonable prices [9].

2.2 Deep Neural Networks

Neural networks are a set of algorithms based on the human brain and used for pattern recognition. According to the definition provided by Gurney [17], a neural network is :

> *an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the interunit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns.*

To explain what Gurney meant, the reader may think of the following. A neural network can be seen as a box that given an input, such as the image of an apple, returns an output, such as the answer to the question *Is this an apple?*. To answer the question, the Neural Network (NN) needs to *learn* the features that are common to all the images that depict apples. The collection of features that are common to all the apples is, in this simple scenario, the training patterns mentioned in definition 2.2. The way the NN learns patterns is somehow similar to the way humans and animals process information. The input information is passed to the first set (layer) of processing elements (neurons) to then transit to adjacent neurons through connections that works similarly to the synapses in a human/animal brain. When definition 2.2 states 'the processing ability lays in the interunit connection strength', that means that the more information the neurons process, the more the NN learns about that piece of information, and the stronger the *synapses* between neurons are. Figure 2.1 provides visual representation of the Input/Output relationship implemented by each neuron.

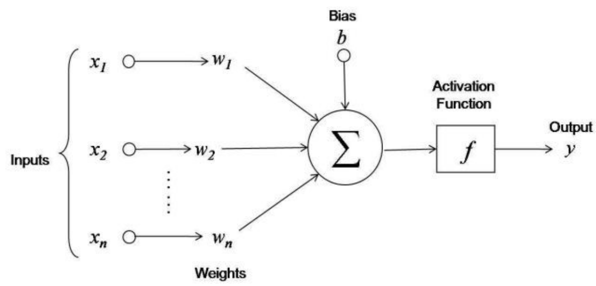


Figure 2.1: Input/Output Relationship of each neuron of a general NN [18]

The mathematical equation of what shown in fig 2.1 is the following [18]:

$$y = b + \sum_{i=1} w_i x_i \tag{2.1}$$

where:

- y is the neuron's output;
- b is the bias. It is considered a systematic error that occurs in the NN model because of incorrect assumptions;
- w_i is the weight associated to the i-th edge. It is the mathematical parameter that the network update during its learning process, hence it is the *interunit connection strength* mentioned in definition 2.2;
- x_i is the input information coming from th i-th synapsis, also called edge.

In general, neurons are aggregated into layers. Neural Network with many layers are called deep neural networks. To update the weights and the biases of all the neurons of the model, one of the most important tools is the loss function. The loss function is the function that allows the user to minimize the error in the neural network, by quantifying the difference between the expected outcome and the produced one [19]. Therefore, its importance lays in the ability to break the model performance down to a single number, which allows the comparison between different solutions [20]. From the loss function, we can derive the gradients which are used to update the weights. The fact that the learning of the NN is based on the use of gradients can be a problem for very deep neural network. In such models, during each iteration of training each of the weights receives an update proportional to the partial derivative of the loss function with respect to the current weight [21]. In some cases the gradient will be so small that the change of the weight will be non existent. As a consequence, this phenomena called *vanishing gradient* might lead to the complete stopping of the network's learning process. This problem is reduced by the use of Residual Neural Network, whose overview is shown in subsection 2.2.1.

2.2.1 Residual Neural Networks

One of the most used and effective solutions to the problem of vanishing gradient was the Residual Neural Network (RESNet) presented in 2015 by He [22]. The innovative element introduced by the paper was the concept of *Residual Blocks.* the architectural building block makes use of a technique called *skip connections.* It allows to connect the activations of a layer to further layers by skipping some layers in between, as shown in fig.2.2.

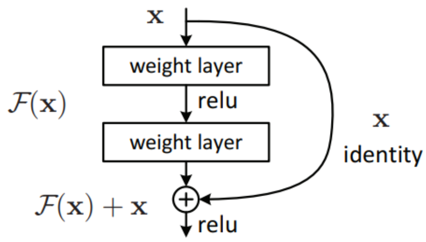
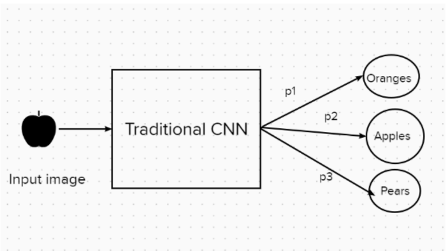


Figure 2.2: Residual Block in RESNet architecture

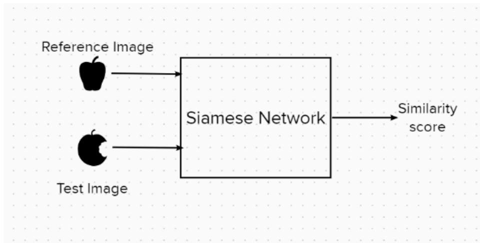
The reasoning behind the introduction of residual blocks is the following. Instead of having layers to learn the underlying mapping *H(x)*, the network learns the mapping until a certain point to then infer the residual mapping *F(x)*. This technique helps at the propagation of the gradient to the initial layers, without incurring in the problem of vanishing gradient [22]. Concerning the use of RESNet, literature is rich with works where the architecture is applied to authentication applications. For istance, Hammad et al. presented a higly accurate RESNet-based model for the authentication of ECG signal [23], while A.Husain and V.P.Vishvakarma used a RESNet152v2 for facial recognition reaching an accuracy of 97% [24].

2.3 Siamese Network

One of the main limitations of classical deep neural networks is the need for a large amount of labeled training data. For instance, if the task is the classification of an object between four possible classes (e.g. oranges, apples, bananas), the network would compute the probability that the image under study belongs to each of the classes. To correctly conduct the classification task it is fundamental to train the model over a large number of images for each of the categories. Moreover, even if there was the possibility to feed the neural networks with millions of images for each class, there is still a major drawback in using standard neural network. If the model is asked to classify an object that does not fall under one of the category it was trained to recognize, it will fail the classification task. The network can only classify images whose category it has already seen during the training phase. So if it is asked to identify a kiwi to a model that was trained only on bananas and apples, for example, the model will fail. To not fail the classification task, the model should be trained over the kiwi class of images as well. Unfortunately there are applications where it is not possible to collect enough data for each class and the total number classes is changing over time. For these cases, the concept of *one-shot learning* was introduced. That is a learning approach that should allow the neural network to learn from a reduced amount of data with respect to usual neural networks which need thousands of labeled samples [25]. According to Fei et al., this is done by extracting the general features of the training samples and use them to classify objects that were never seen by the network during its training phase. The Siamese Network is an example of one-shot classifier. It translates the discrimination task in a similarity problem by computing the distance between two vectors of features extracted from the input images by means of two identical networks [26]. Figure 2.3 shows the comparison between the two approaches.



(a) Standard CNN



(b) Siamene Network

Figure 2.3: Standard CNN vs Siamese Network. Classification task.

The difference between the two models is clear. The standard CNN takes an input image and computes the probability of the said image of belonging to one of the available classes. The Siamese Network takes the input image and produces a similarity score between the said picture and each reference image available in the database. This score denotes the chances that the two input images depict the same object. Typically that is in the range 0, 1 where 0 denotes no similarity and 1 denotes full similarity. Based on the performance metrics, a threshold is decided. Above the threshold the two images will be considered as representing the same object, below the threshold they will represent different objects. The characteristics of the one-shot learning approach overcome the limitations of the standard CNN for the following reasons:

- the Siamese Network requires less labeled images than the CNN for the training;
- if a new class is added, the network need only few images of the class in the database.

Because of the considerations made so far, it appears reasonable to use the Siamese Network for the task of CSR tags recognition.

Chapter 3

Data Acquisition

The first fundamental step of any machine learning model implementation, especially neural networks, is the creation of the dataset. For deep neural networks, there is usually the problem of gathering enough labeled data for the training phase [27]. Using Siamese Neural Network should ease the task of data collection and generation. figure[28]. However, Koch et al. train a siamese Network on a training set that starts with 30.000 pairs in the first experiment and it rises til 90.000 pairs in the third experiment. Because of technical limits, it was not possible to train the neural network implemented for this project with such a large number of pairs, however a considerably amount of images were collected. This was done to allow a more numerous pairs creation for future projects that will use the data presented in this chapter. As literature can confirm, data collection is always a delicate task in any AI project [27]. In this specific case, the main difficulty lays its origins in the fact that the target of the object recognition does not exist in its final form yet. Indeed, one of the main challenges of this thesis was the fact that the technology is still under development and that it is constantly evolving. Thus, the following approach had been adopted:

- Organizing the existing images;
- Increase or generate acquisitions for the dataset.

3.1 Existing data

The first activity done for the data collection for this project was the division in categories of the available images. In order to let the network work correctly it is important that the acquisitions are from the same distribution. A consequence of a non uniform dataset would be a high inaccuracy in the results [29]. Revising the images already available for the project, it was possible to divide the acquisition

according to two main categories. The first division is made on the technology used for the production of the tags .
Each category had images that were whether acquired with the more advanced microscope from the Physic Labs or the microscope from the SNT department. Therefore, the second division was made based on the acquisition tool. fig.3.1 provides a scheme that summarizes all the image categories.

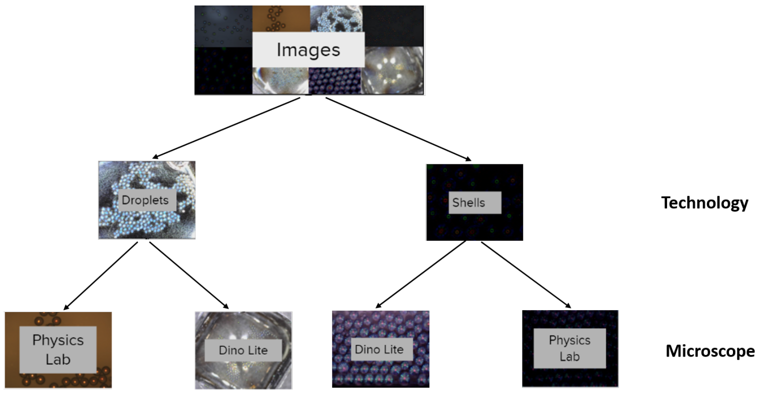


Figure 3.1: Available images divided in categories and subcategories

What we obtained from the categorization of the available images was something that did not meet the ideal requirements for the wanted data. Indeed all available images were not only coming from many (different technology and microscope), but all with JPEG compression. The reason why dealing with compressed images is not the best possible situation is the following. As it will be mentioned in section 3.4, the size of the crystals tags is that the change of few pixel in an image acquisition can be fundamental for the classification problem the project aims to face. Since compression can affect the image quality and the idea is to deal with a very controlled dataset, it would be beneficial to have images that are not compressed, hence with extension TIFF, BMP, or PNG (the latter, with no compression option). To recall what was said at the start of the chapter, neural networks require images sampled from the same distribution for the training set and for the validation and test set respectively. This means that it is acceptable to have little variation of distribution between the test/validation and training set, but it would hurtful or the project to have images coming from mixed distributions within the same training set, for example. If that happened, the network would not be able to understand how to classify the tags. In order for it to learn how to correctly do its task, it is then important to have a large enough dataset, both in terms of classes and images per classes. As shown by 3.1, what we had in terms of total images and images per classes did not meet the wanted requirements.

Table 3.1: Comparison between the old data and the wanted data in terms of features

	Old Data	Ideal Data
Image Compression	JPEG	TIFF, BMP, PNG
Images per Tag	5-20	At least 100
Number of Tags	5	At least 10
Distribution Data	many distributions	one distribution

3.2 Generated Data

For the reasons presented in the previous section, the generation of a new dataset was needed. In order to do so, two choices were made:

- CSR Technology;
- Microscope.

Regarding the technology, there were two possible options: droplets and shells. The in detail explaination of how the two types of CSR work, is explained by Urbanski et al. in the 2017 article *Liquid crystals in micron-scale droplets, shells, fibers* [8] After talking with the Physics department about it, it was decided to opt for the droplets technology. The decision was taken based on the possibility to have a faster and easier to control series production with respect to the shell production. Concerning the acquisition tool, different ones were provided. To choose the right one, a comparison in terms of features was made. fig.3.2 and Table 3.2 show and list the tools available for the acquisition.

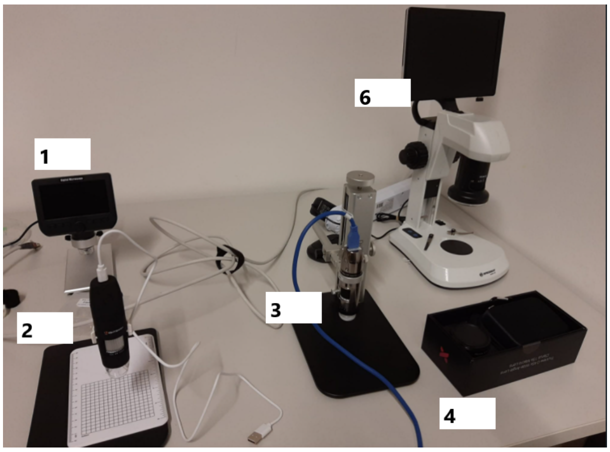
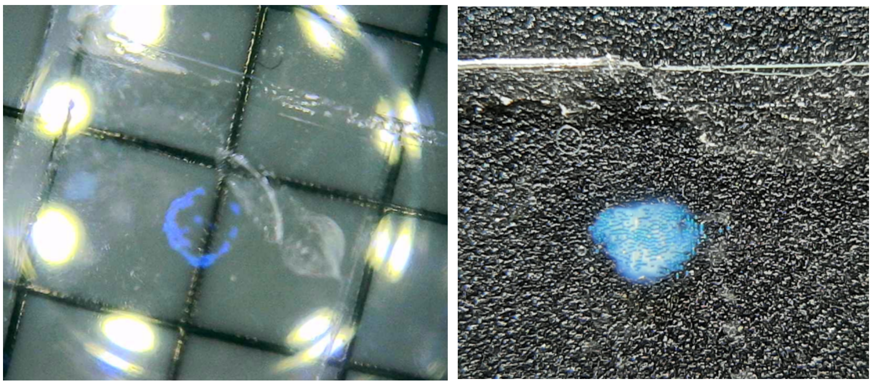


Figure 3.2: Acquisition tools

Table 3.2: List of acquisition Tools

	Acquisiton Tool
1	Koolertron Y01AS-ASMXW49S-A_F USB
2	Lightswim TENG-MICR-BLK
3	Dino Lite
4	Xenvo Pro Lens Kit
5	YINAMA 317 WiFi and USB
6	Bresser LCD microscope 5809100

The feature comparison was made considering one tag sample and taking images with each acquisition tool that fulfilled the requirements. Of all the microscope, only n.6 from Table 3.2 was not considered because of the lack of interoperability with a computer. Regarding the other devices, the worst ones for our purposes were the LightSwim (fig.3.3a) and Xenvo Lens (fig.3.3b).

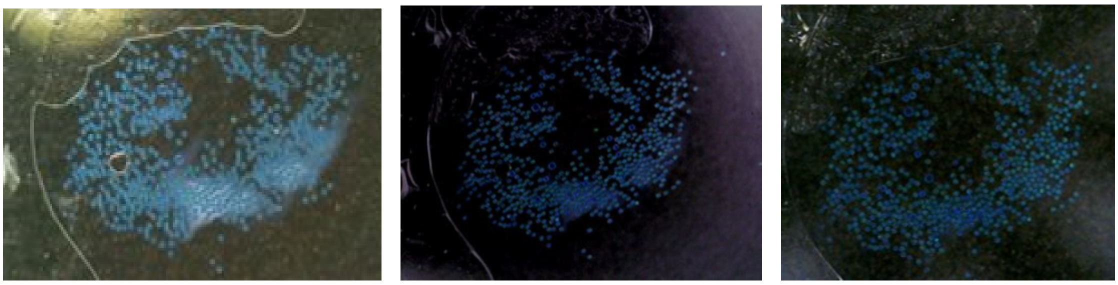


(a) Lightswim TENG-MICR-BLK **(b)** Xenvo Pro Lens Kit

Figure 3.3: Worst acquisitions are from Light Swim and Xenvo

Several problems occurred. fig.3.3 shows that both tools provided pictures with an ununiform background that was also highly sensitive to the light and not at all invisible. Since the wanted situation is a dataset of images extremely controlled, all this additional noise is not ideal. Besides, the main problem with such tools is the insufficient magnification capabilities. In both pictures, the tag is barely visible and nothing more than a stain. It was possible to obtain better res with the remaining three microscopes.

From the images in fig.3.4, regardless of the acquisition tool of consideration, it is possible to observe the tag more clearly than the images taken with the Xenvo Lens and the microscope LightSwim (fig 3.3). Given the poor performance of the Xenvo Lens and the LightSwim microscope, the choice was between one of the



(a) Koolertron **(b)** Dino Lite **(c)** YINAMA 317

Figure 3.4: Best Acquisitions

three microscopes. As discussed earlier (section 3.1), it would be ideal to have decompressed images that can be taken through software easily mountable on a pc. As it appears from table 3.3, the only microscope that provides both requirements is the Dino lite.

Table 3.3: Comparison between the microscopes in terms of resolution and compression of the images and possibility of PC connection

Microscope	**Resolution**		**Compression**		**PC connection**
	Fixed	**Max (pixel)**	**Fixed**	**Format**	
Koolertron	NO	3648x2736	YES	JPEG	YES
Dino Lite	NO	2560x1920	NO	TIFF, BMP, JPEG, PNG	YES
YINAMA	NO	2560x1920	YES	JPEG	YES

The chosen microscope comes from the family of Dino-Lite, model AM73115MZTL. It supports a 5MP camera with 8 LEDs, controllable through software, to get the clearest possible image of the tag. The microscope can be easily connected to a PC via USB and and controlled with the aid of the software Dino Capture 2.0, fig. 3.5.

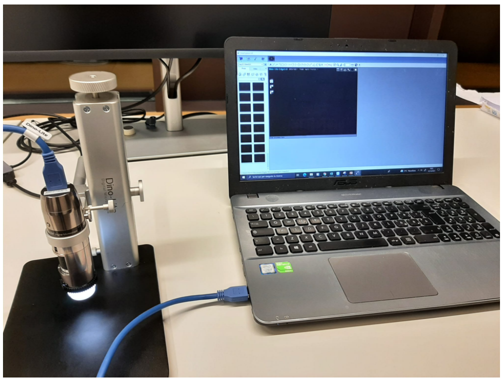


Figure 3.5: Microscope Dino Lite AM73115MZTL connected to PC

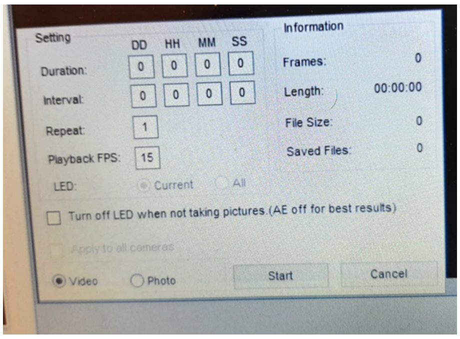


Figure 3.6: Dino Capture 2.0 interface: Window for automated acquisitions

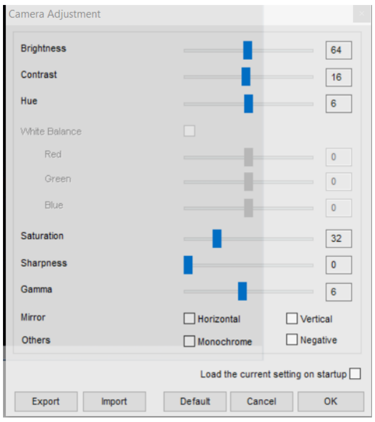


Figure 3.7: Dino Capture 2.0 interface: Window to set image parameters

Fig. 3.6 and 3.7 shows the very intuitive software interface that allows to automatically take pictures, by setting the number of acquisition to take in a defined time interval, for example, and by indicating some image parameters, such as brightness and saturation, to improve the quality of the acquisition. Unfortunately, both the artificial illumination (i.t. the LEDs and the light in the room) and the natural light (such as the light coming from the window) made it impossible to have clear and clean tag acquisitions. However, the quality of the image depends not only on the environment but also on the tag itself.

3.3 CLC68II

The first set of tags to be provided was the CLC68II. The arrangement of the tags on the sample is shown in Fig. 3.8. They appear blue to the bare eye. According to the team that worked on the production process, the blue color comes from the following procedure. The CSRs were supposed to be red, but they went through a polymerization technique that made them turn yellow. After the polymerization, they got squished in a UV matrix that gives them the blue color visible in fig.3.9a.

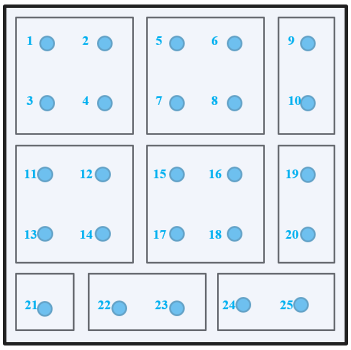


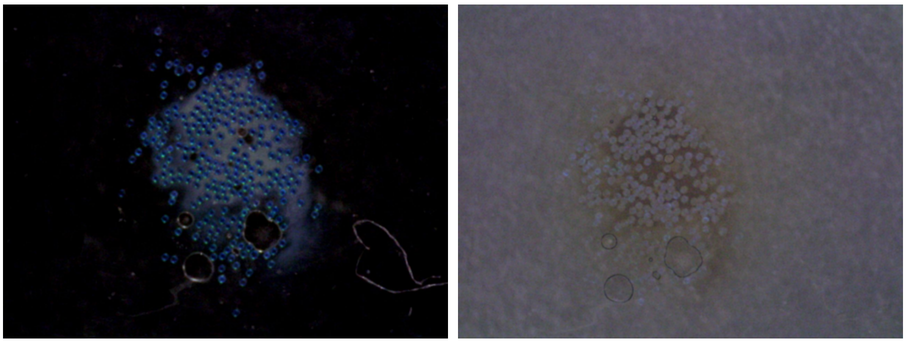
Figure 3.8: CLC68II : position map of the tags on the laboratory slide

To see if it was possible to see the tag regardless of the background color, two different background colors were proposed. From fig.3.9, it is clear that the best image is fig.3.9a since the tag is barely visible when using a white background, as shown in fig.3.9b.

Since the visibility of the tag is foundamental, black background is preferred due to the optical behavior of the material. Even after choosing the most appropriate background for the case study, different problems occurred with the attempt to take clear images of the provided tags.

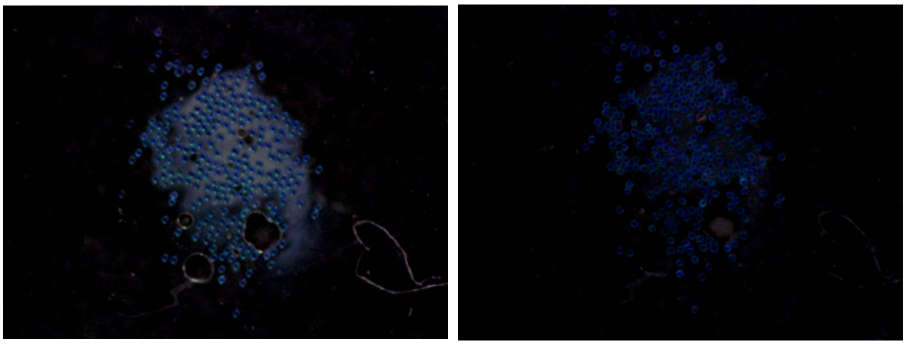
As it is visible from fig.3.9a, the tag presents several defects, for instance air bubbles and a a blue-ish liquid surrounding the CSR. By taking a picture of the same tag under different lighting conditions, it was possible to observe a significant difference shown in fig.3.10.

In fig. 3.10a all the 8 LEDs are on and with the natural light coming from the window. Fig. 3.10b was taken a few minutes after fig.3.10a, with all the 8 LEDs



(a) Black background **(b)** White background

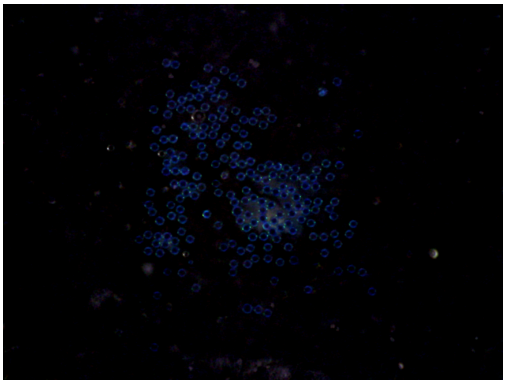
Figure 3.9: Tag n.1 from CLC68II

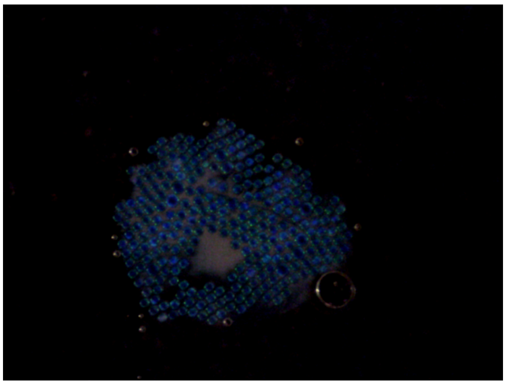


(a) Normal Light **(b)** Low Light

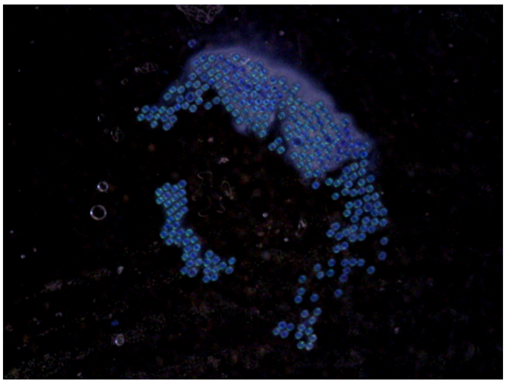
Figure 3.10: Tag n.1 from CLC68II. Different lighting conditions.

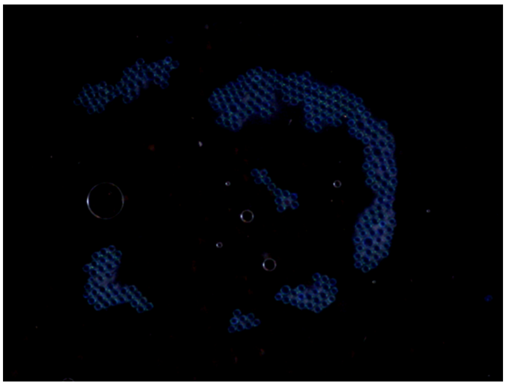
on but with black cardboard placed next to the microscope, therefore blocking the natural light.



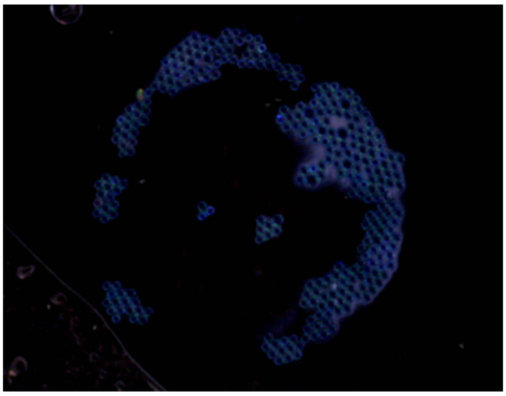


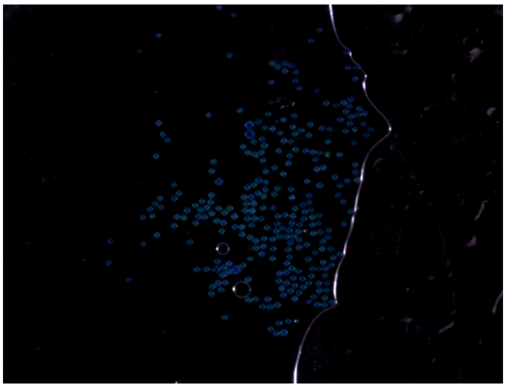
(a) Tag 4 **(b)** Tag 5





(c) Tag 4 **(d)** Tag 19

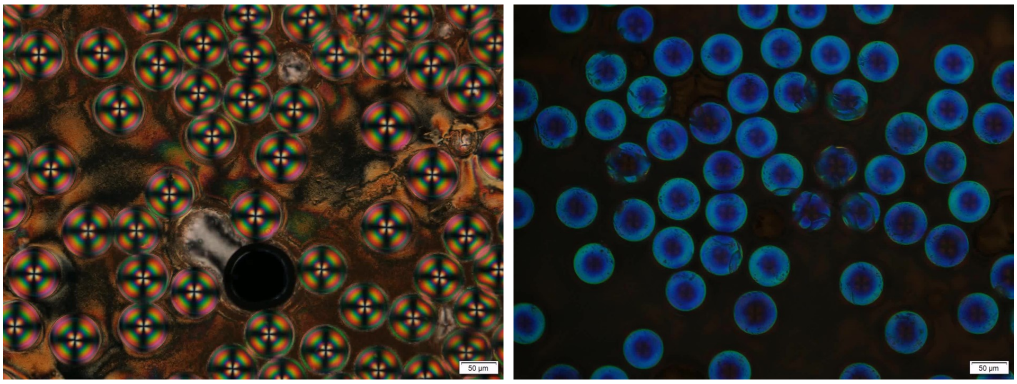




(e) Tag 20 **(f)** Tag 21

Figure 3.11: Examples of tags CLC68II.

Fig.3.10 shows a significant sensitivity of the glue medium to the light. Since it affects the visibility of the tag and makes the acquisition procedure unnecessarily complicated, it is an issue to solve. Indeed, the light adjustment attempts to get the best picture in terms of background, and medium visibility determines a darker and less clear view of the tag itself. The mentioned problems are common to almost all the provided tags. Fig.3.11 proposes some examples of the *bad tags*. Thanks to the assistance of the Physic team, it was possible to identify the problem. According to them, during the production process, something went wrong: the chrystals did not get to the Cholesteric stage. The presence of black crosses in fig.3.12a suggests that they stopped at the nematic phase. In both pictures, it is also visible that the tag under study was damaged at different points.



(a) Transmission Mode **(b)** Reflection Mode

Figure 3.12: Images taken with Polarized light microscope in transmission and reflection mode

In conclusion, the first sample was not suitable for the experiments that were to follow. However, it provided the opportunity to understand the characteristics that a tag should have to be able to be used and what kind of defects coming from the production procedure would be better to avoid in the final product. Hence, it would be advantageous to have a dataset of images where:

- The tags should be visible on the background;
- No bubble air in the tag;
- No visible medium between the droplets;
- The glass should be as less prone to scratches as possible.

3.4 Datasets after the CLC68II

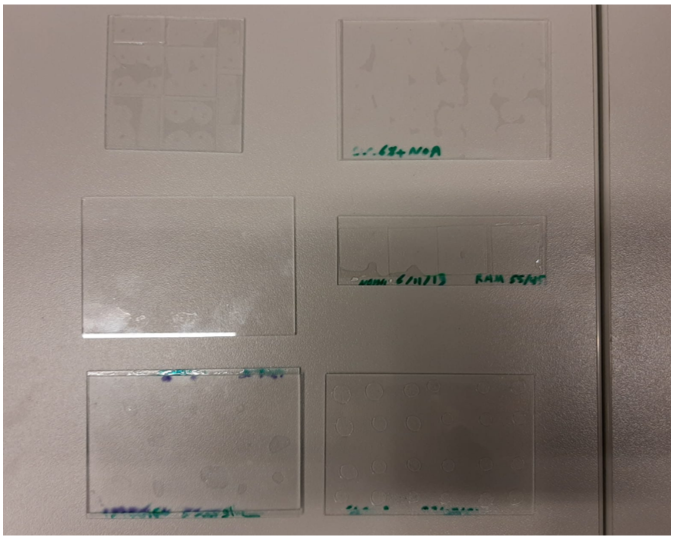


Figure 3.13: Different samples of CSRs

After the first attempt of tag production, different samples were produced. From the top left to the bottom right of fig.3.13 the following samples are shown:

- CLC68II. Sample described in section 3.3;
- CLC68 + NOA: they are similar to CLC68II (fig.3.14);
- NOA 6/11/19: the CSRs are embedded in the glue and left drying to the air (fig.3.15);
- RMM 55/45: chemical mixture owned by an external company. The composition of the liquid crystal is unknown (fig.3.16);
- CLC68+glue: the CSRs are embedded in the glue and fixed between two glasses (fig.3.17);
- CLC 07/07/21: they do not use any binder(fig.3.18).

All the sample contains a number of tags around 25-30 tags and for each sample 100 acquisition per tag were taken with the Dino Lite microscope All the CSR droplet diameters, apart from the RMM 55/45 one that is unknown, are between 70-80 µm.

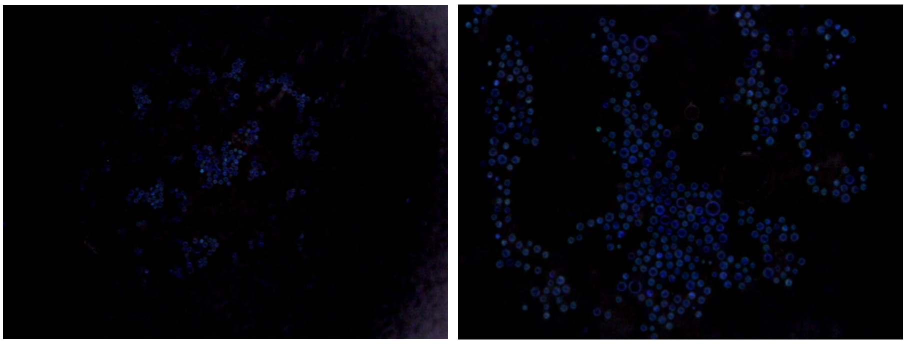


Figure 3.14: CLC68+NOA

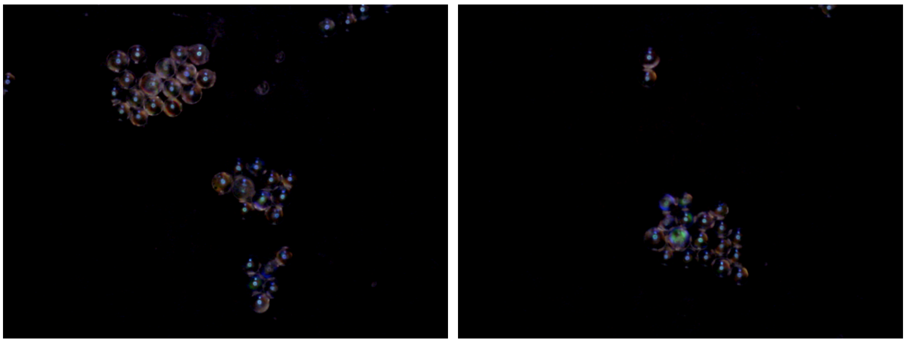


Figure 3.15: NOA 6/11/19

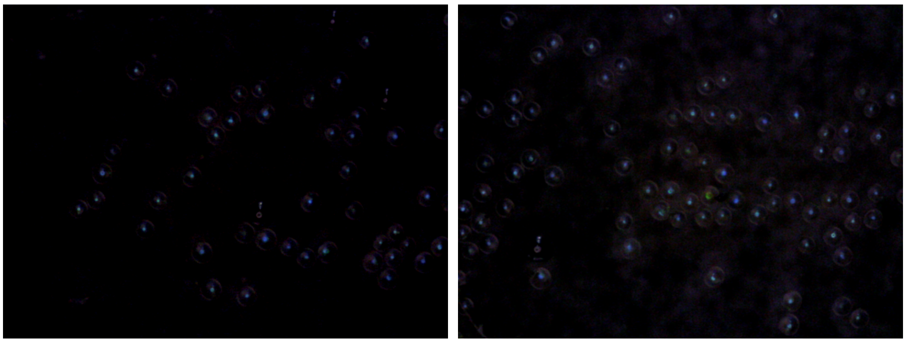


Figure 3.16: RMM 55/45

Considering the requirements listed at the end of the previous section, the only set of unusable tags is CLC 07/07/21, which presents several cases of disrupted and defected droplets, as shown in fig.3.18. Regarding the other samples, they all satisfy the requirements. So, any of them could be chosen for further analysis. In this case, the sample chosen to proceed with the project is the set of CLC68+NOA. Fig.3.19 shows a map of the tags on the sample.

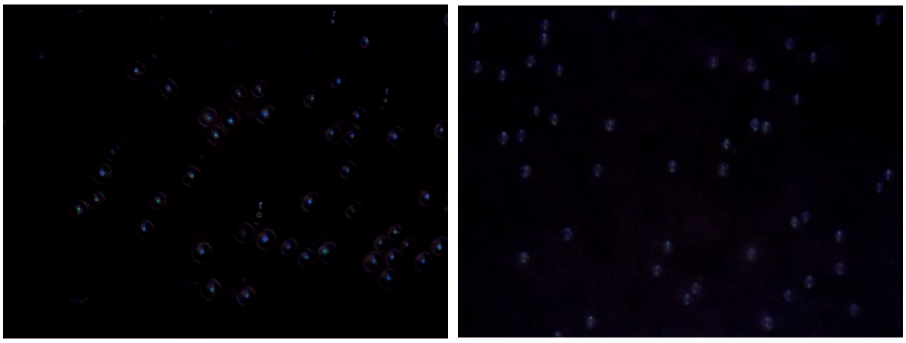


Figure 3.17: CLC68+glue

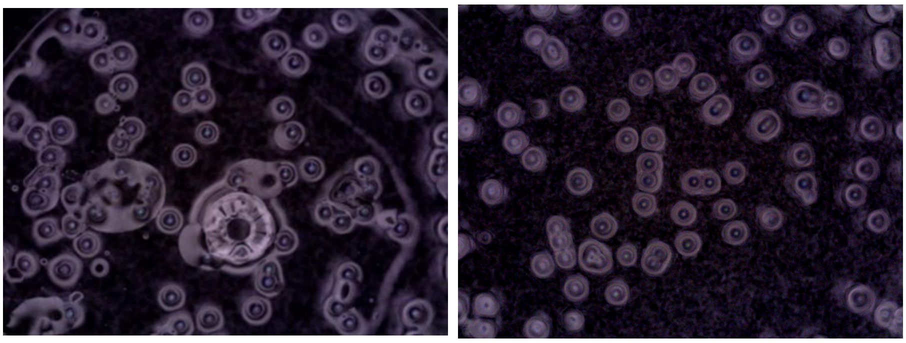


Figure 3.18: CLC 07/07/21

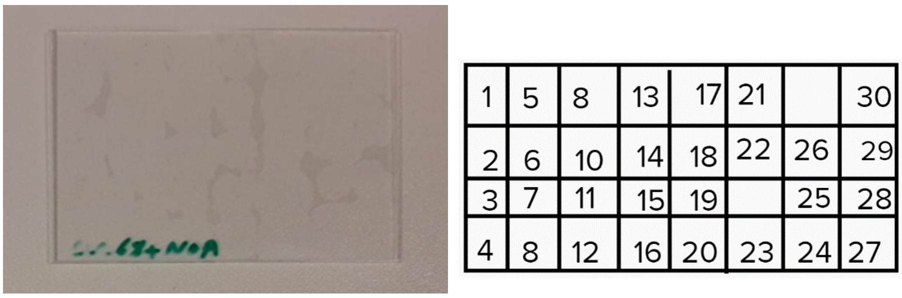
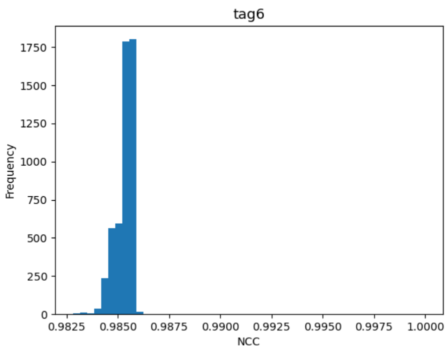


Figure 3.19: CLC68+NOA: map and laboratory slide

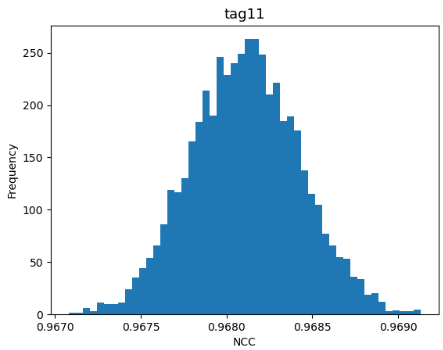
Using the default conditions of the software that allows to control the microscope and keeping all the LEDs on, 100 images of dimensions 2560x1920 pixels for 26 of the given 30 tags were acquired. The magnification level depended on the specific tag and the acquisitions were taken trying to have the full tag in the image.

3.4.1 Preliminar Analysis of the collected images

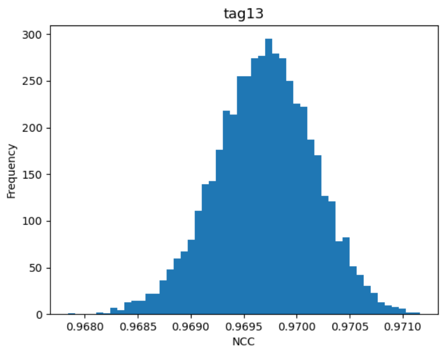
In order to run some preliminary analysis on the images, the algorithm was applied on a subset of the available tags to compute the Normalised Correlation Coefficient (NCC). As explained in more details in section 4.1, the NCC algorithm computes an image similarity score over two types of pairs: pairs of images of the same tag, for simplicity they will be addressed as *same_tags_pairs*, and pairs of images from different tags, also called in this thesis *diff_tags_pairs*. What resulted from the tests was that the distribution of the NCC parameter was, as expected, really close to 1 with some variance for the *same_tag_pairs* (see fig.3.20).



(a) Tag 6



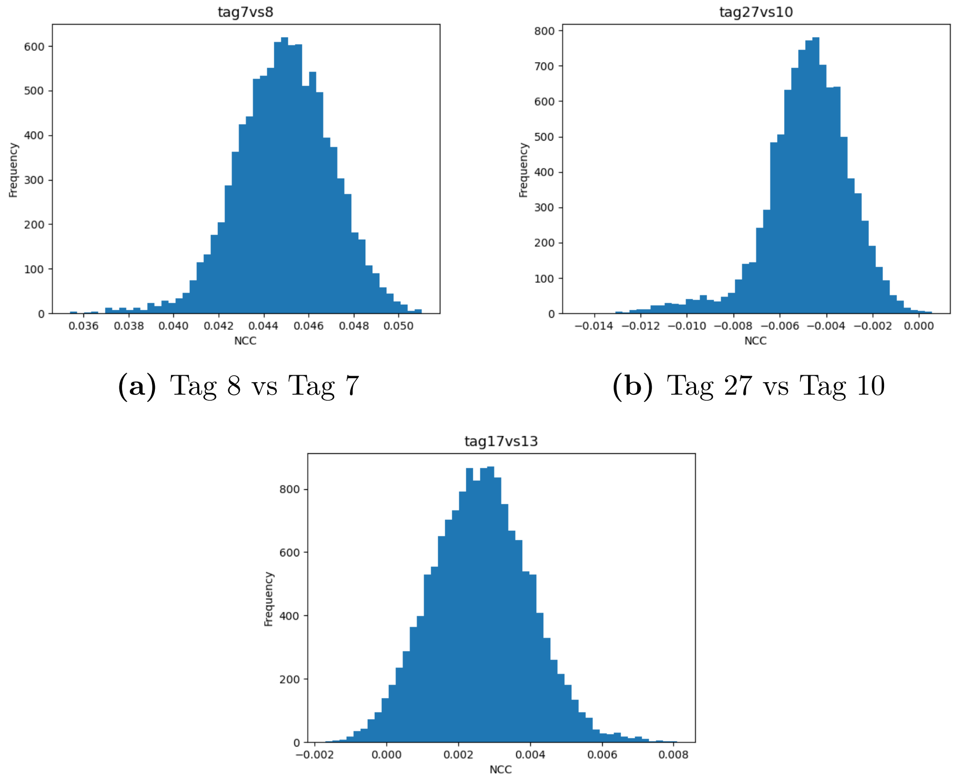
(b) Tag 11



(c) Tag 13

Figure 3.20: NCC results for same tag acquisitions

Regarding the (*diff_tags_pairs*), some anomalies occurred. What was expected to be found was a distribution of the NCC with mean value close to zero. Even if this happened for some of the considered pairs (fig. 3.21) it did not happened for all of them, as shown in fig.3.22. The reason was found by comparing images of the tags that showed anomalies. Fig.3.23 shows the tags of the pairs from fig.3.22, whose NCC distribution were atypical.



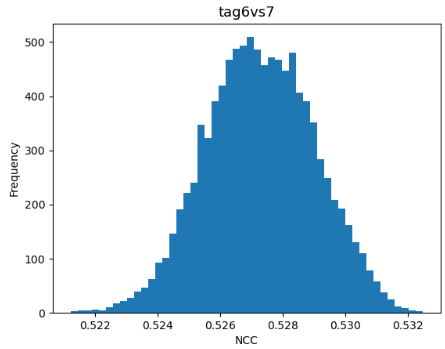
(a) Tag 8 vs Tag 7

(b) Tag 27 vs Tag 10

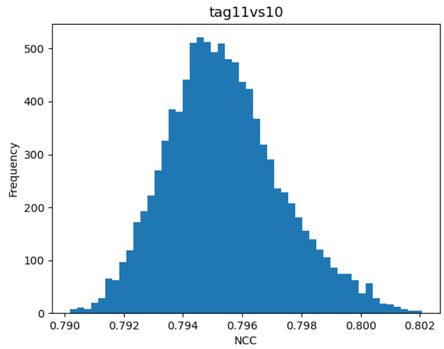
(c) Tag 13 vs Tag 17

Figure 3.21: NCC results for diff_tag_pairs acquisitions. The behaviour was as expected: NCC distribution centered on near-zero values

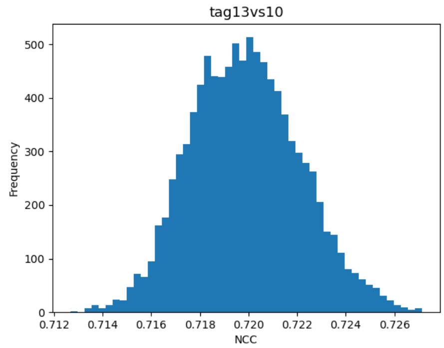
What is immediately visible is that all the pictures have one major similarity: the presence of a white light region in the some corners of the acquisition. To verify that the unusual distribution was due to the poor robustness of the NCC algorithm to the defect and not to a basic implementation error, the following test was run. The original 2560x1920 images were cropped into squares of 1200x1200 pixels. In this way, the white region was deleted from all the acquisitions. The process did not cause a significantly high loss of information since most of the tag was already present in the central region rather than on the corners. By running the NCC algorithm on the set of cropped images, the NCC distribution go back to normal for all the *diff_tag_pairs*, as shown in fig.3.24.



(a) Tag 6 vs Tag 7



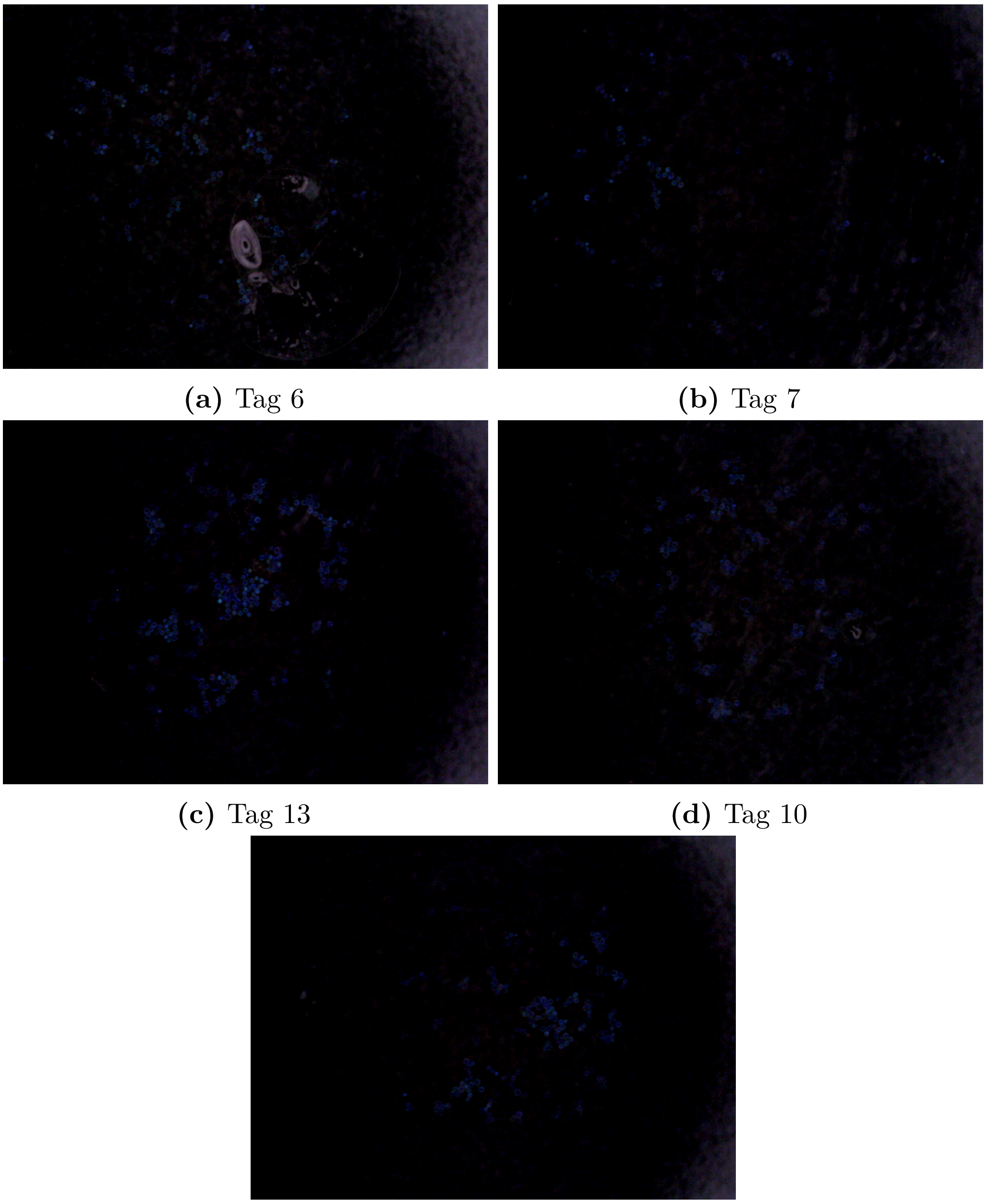
(b) Tag 11 vs Tag 10



(c) Tag 13 vs Tag 10

Figure 3.22: NCC results for diff_tag_pairs acquisitions have the distribution centered not on near-zero values

The tests mentioned above were important to evaluate the quality of the acquisitions taken so far on the CLC68+NOA for the creation of what will be the dataset for the Neural Network. In addition to the considerations made at the end of Section 3.3, the preliminary analysis lead to two more features which the final dataset should have for the project's purposes. As stated already, it would be ideal to work with *controlled data*. This means that it would be preferable to work with images that do not present defects that depends neither on the physical production process nor the microscope setting. Example of the said defects are the broken droplets mentioned in section 3.3, the big glass scratch in the pictures for tag 6 (fig.3.23a) and the white region visible for the acquisitions in fig.3.23. Moreover it would be beneficial to ensure that the images are centered on the tag, so that any cropping process would not cause significant information loss. For the reasons above, a new set of pictures was acquired. The microscope settings used for the new set of images differ from the old one mentioned at the end of the previous



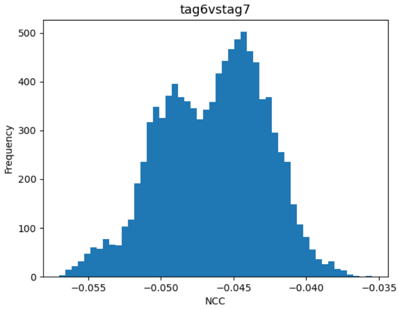
(a) Tag 6 **(b)** Tag 7

(c) Tag 13 **(d)** Tag 10

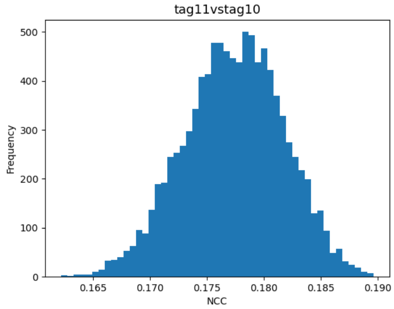
(e) Tag 11

Figure 3.23: Tags from the pairs with atypical NCC distribution

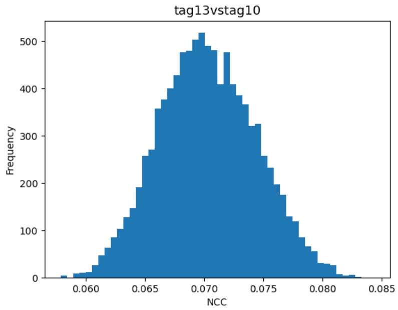
section only for the magnification level. In the old set, the zoom choice was the one that allowed to take the whole tag in a picture. In the new one, the magnification level is 160 for all the CSRs. Moreover, for each tag the images focus on the regions with the higher concentration of material. A comparison between old and new set



(a) Tag 6 vs Tag 7



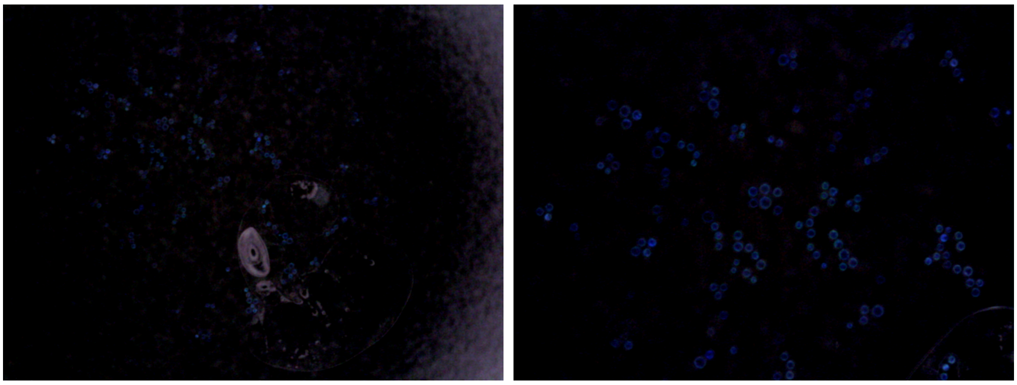
(b) Tag 11 vs Tag 10



(c) Tag 13 vs Tag 10

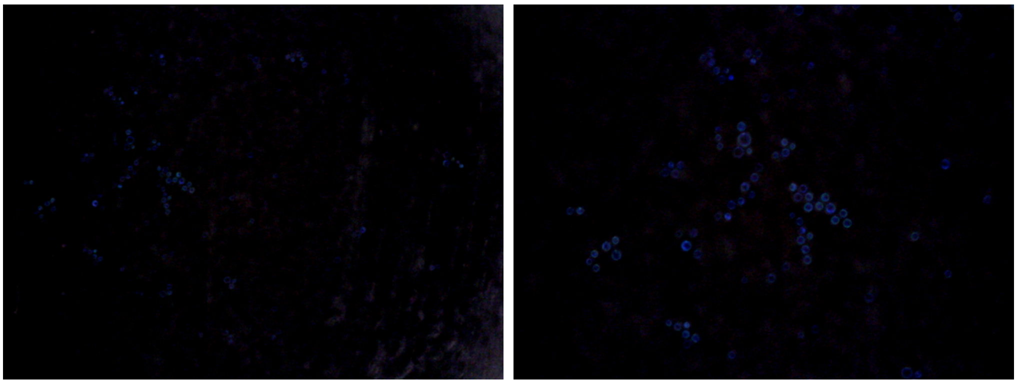
Figure 3.24: NCC results for diff_tag_pairs acquisitions after cropping procedure

of images is presented in Figures 3.25, 3.26, 3.27.



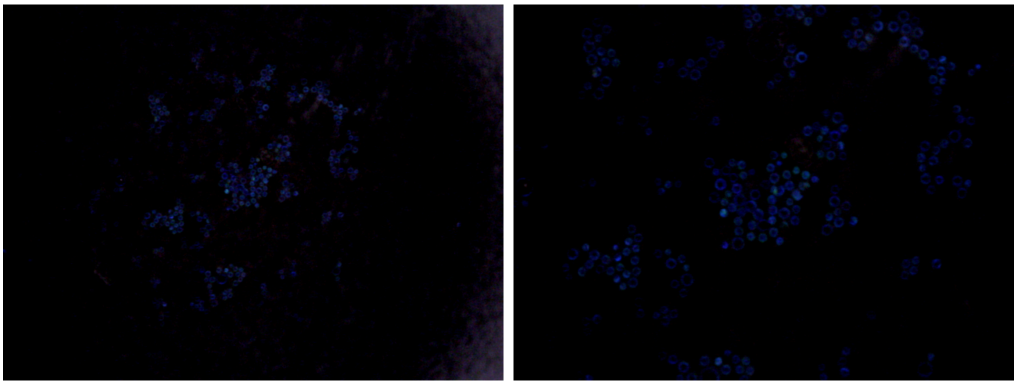
(a) Old **(b)** New

Figure 3.25: Comparison between old set and new set for tag6



(a) tag7 - Old **(b)** tag7 - New

Figure 3.26: Comparison between old set and new set for tag7



(a) Old **(b)** New

Figure 3.27: Comparison between old set and new set for tag13

Chapter 4

Methodology

The following chapter discusses the methodology used for the project. Section 4.3 goes over the way the collected images are organised and augmented in the project. Sections 4.1 and 4.2, the two approaches used for the experiments are presented. The first one is an algorithm implements a model based approach based on the computation of the cosine distance between two images. The latter use a data driven approach based on the use of Siamese Network to evaluate the image similarity.

4.1 Normalised Correlation Coefficient

The algorithm chosen to analyse the similarity of the images collected for the dataset is the Normalised correlation coefficient (NCC), also known as Pearson coefficient. By definition, NCC is the covariance of the two variables divided by the product of their standard deviations. Given pair data, the correlation coefficient r_{xy}, can be derived with formula 4.1:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{4.1}$$

Where:

- n is the number of samples
- x_i, y_i are the individual sample points indexed with i
- $\bar{x}$ and $\bar{y}$ are the sample mean for x and y respectively.

In this case, the NCC has to be computed over pairs of images. Each image has 3 channels of colour, hence the NCC must be calculated over each channel.

Each channel is read as a 2D matrix that can be flattened to a 1D array without loss of information. The NCC is then computed over each channel and the global coefficient value corresponds to the average of the three obtained values.

4.2 Neural Network architecture

The data-driven approach used to run test over the generated dataset involves the use of a siamese neural network based on the RESNet18 architecture. Its purpose is to extract a vector of features out of each image in order to learn which pictures represent the same object and which do not. Fig.4.1 shows the Network structure that implemented in Pytorch for this project.

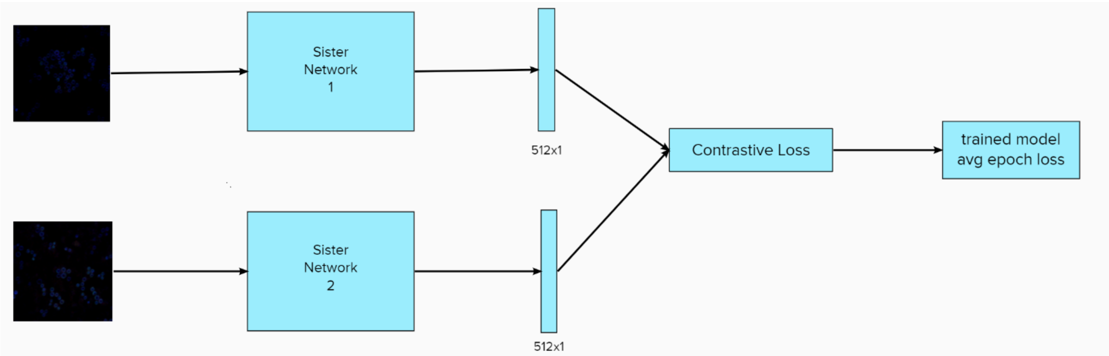


Figure 4.1: Neural Network architecture

Two input images randomly paired as explained in subsection 4.3.1, are given to two networks sharing the same architecture and weights. The sister networks extract a vector of 512 features from each image resized to a matrix of 224x224 pixels. This is possible by dropping the last layer of the RESNet18 used as sister network, that is usually used for classification. During the training and validation phases, the contrastive loss is computed to update the weights of the two sister networks. As a result, the trained model and the average epoch loss are obtained in output. During the testing phase, the flow of information follows the same path apart from the last stage. Instead of computing the contrastive loss over the vectors of features, just their distance is calculated. As testing result, over a batch of pairs, a vector of distance scores is obtained.

4.2.1 Implementation details of the Network

In this paragraphs, some implementation details and the metrics during training and testing phase are presented.

Contrastive Loss and decision rule

In order to allow the model to learn discriminating features, it was defined a low dimensional space such that similar images in the input space result in similar representation (near-zero distance value) and dissimilar images result in varied representation (non-zero distance value). This was done in the implemented Siamese Neural Network, by using the Contrastive Loss based on Euclidean distance. Following the methodology presented by R.Hadsell et al [30], equation 4.2 was implemented.

$$L(W, Y, \overrightarrow{X_1}, \overrightarrow{X_2}) = (1 - Y)\frac{(D_W)^2}{2} + \frac{Y}{2}max(0, m - D_W)^2 \tag{4.2}$$

where:

- $\overrightarrow{X_1}, \overrightarrow{X_2}$ are the features vectors of the two images;
- Y is the true label, that will be 0 for negative (different tags) pairs and 1 for positive (same tag) pairs;
- m is the margin;
- D_W is the Euclidean Distance between $\overrightarrow{X_1}$ and $\overrightarrow{X_2}$.

Given the euclidean distance between two images, the following decision rule was used to train the network. For values of the euclidean distance greater than a certain threshold, the pair is considered to be a negative pair labeled as a zero. This because the higher is the distance between two images in the euclidean space, the higher is the probability of them not showing the same objects.

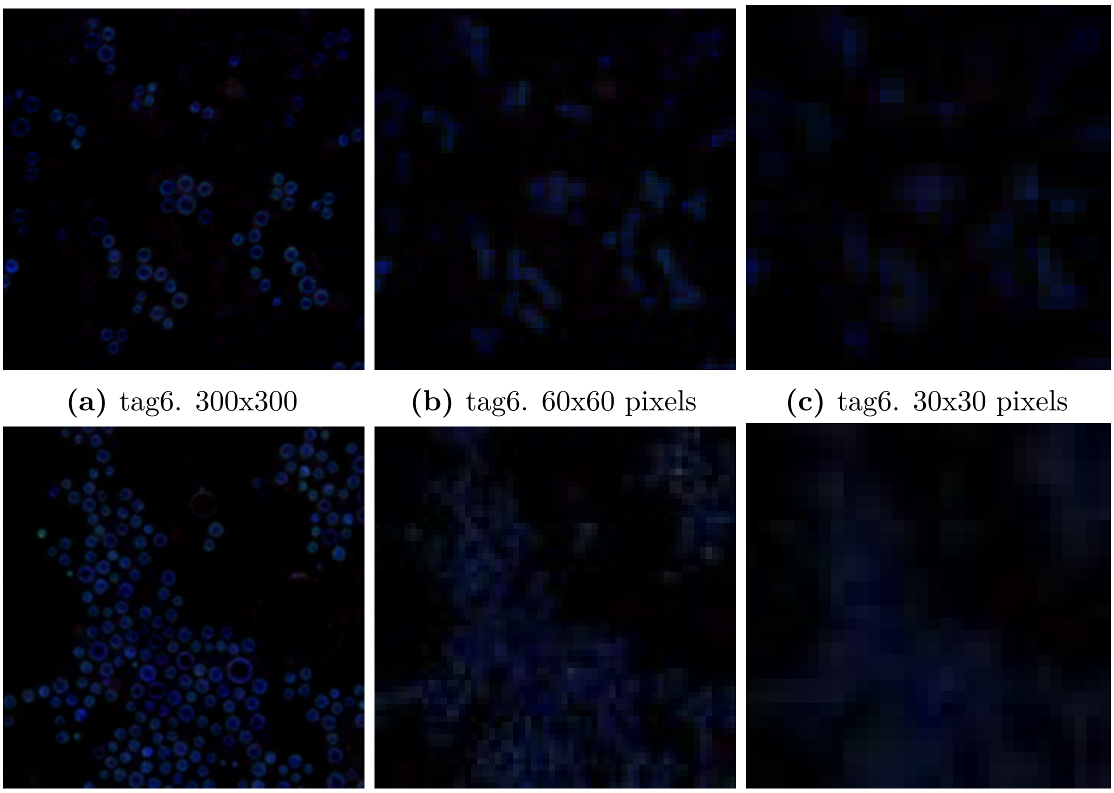
Learning Strategy

The learning rate strategy used for the Neural Network during training and validation phases was as follows. Starting from a small value of the learning rate, s.t. 0.0001, the Network was trained for certain number of epochs, usually around 15, until a local minima occurs. At that point, the learning rate is incremented of a factor of 10. The training goes on for few epochs with the incremented learning rate to then decrease it again of a factor 10 before it starts to overfit on the training samples. The drawback of the whole procedure is that it was not implemented in an automatic way, hence it required the constant check of the learning trend by means of the average epoch loss. The average epoch loss corresponds to the value of loss that the network wants to reduce through optimization.

4.3 Dataset

The final set of images presented at the end of chapter 3 is used for the creation of the datasets. From the 30 available tags, 10 tags were chosen to compose 3 dataset that differ from each other in terms of resolution. The reason behind this choice is the following; one of the hoped outcome for the SNT and Physic departments of the University of Luxembourg is the possibility to scan the tag on any surface with just the aid of a phone. Since a laboratory microscope can usually take higher quality acquisitions than a smartphone, it was thought to consider dataset with different resolution to compare the performance of the Neural Network and the NCC algorithm. Hence, 3 datasets were produced:

1. Dataset of 1000 images of resolution 300x300 pixels;
2. Dataset of 1000 images of resolution 60x60 pixels;
3. Dataset of 1000 images of resolution 30x30 pixels



(a) tag6. 300x300 **(b)** tag6. 60x60 pixels **(c)** tag6. 30x30 pixels

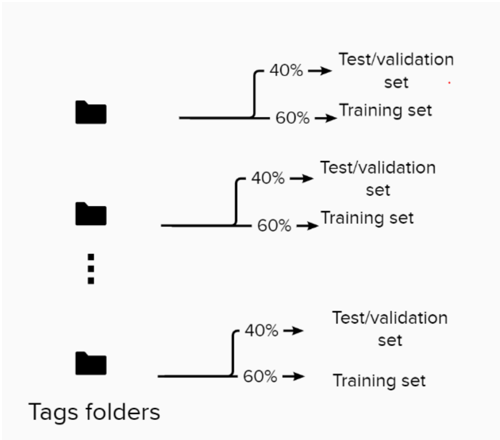
(d) tag3. 300x300 pixels **(e)** tag3. 60x60 pixels **(f)** tag3. 30x30 pixels

Figure 4.2: Retake 1 of Tag n.6, n.3, from the 3 different dataset.

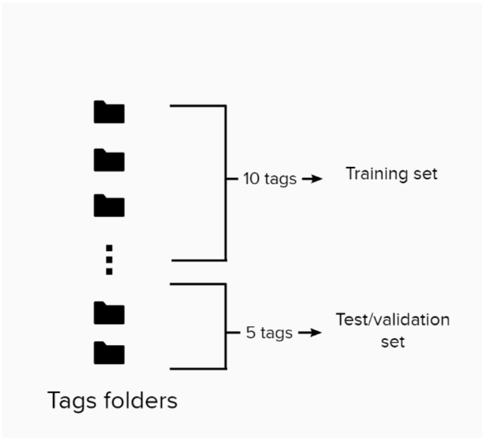
A comparison between images of the same tag from the three different datasets can be observed in figure 4.2.

4.3.1 Data preparation

The approaches explained in section 4.1 and 4.2 were tried over random pairs of images. The pairs creation process depends on the set of experiments. Regarding the ones implying the use of the NCC algorithm, two data structures were made: one containing the path to the images for all the possible same tag pairs and another one for all the different tags pairs. The same road cannot be followed for the Neural Network. Training the network with all possible pairs is infeasible due to the limited physical resources. Hence, just a portion was used to feed the neural network. Data preparation is a delicate step in machine learning. Mislabelling strongly impact the ability of a model to classify images as it can be observed from the AUC results shown by Dao et al. in their paper [31]. Furthermore, the dataset used for this project is a novel manually collected dataset. For simplicity, the pairs were created offline. Offline mining compared to online, is way easier to manage and debug. The division in training validation and test sets was the next decision to take. In this project, the neural network is trained for all the 3 versions of the dataset by using two different dataset splitting strategies. The first one consists of migrating the 60% of the images for each folder to the training set. From this set, pairs are randomly generated to ensure that invalid pairs will be excluded by the process. *Invalid* means that same tag pairs cannot contain the same retake twice. Identical procedures were followed to generate the pairs of the validation and testing set. Both the sets take the 20% of images from each tag folder. This approach is used in experiment 1 (section 5.2.1). In the second strategy, the splitting is not done on an image level but on a tag level. For example, given a certain dataset, the new approach consists of taking all the images in the first n tags folders and using these images to generate random pairs. The remaining tags go half to the validation set and half to the test set. This strategy was adopted in experiment 2 (section 5.2.2) together with a different and larger version of the dataset, as a way to increase the training set to improve the classification ability of the siamese network. The two splitting methods are also shown in Figure 4.3.



(a) Splitting Strategy n.1



(b) Splitting Strategy n.2

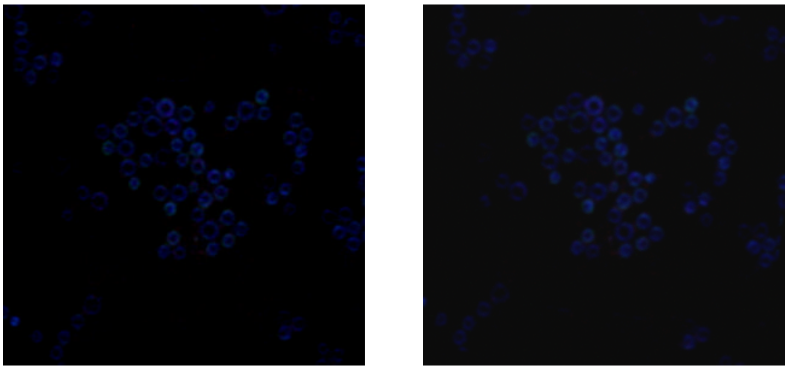
Figure 4.3: Splitting strategies used for the dataset of the neural network

4.3.2 Data Augmentation

Both in the experiments ran with the NCC algorithm and the neural network (sections 4.1 and 4.2), there was the need to produce augmented images. For this purpose, a computer vision tool Albumentation was used. It is a Python library for fast and flexible image augmentations that efficiently implements a rich variety of image transform operations. The transformations that were used in this project are:

- Rotation;
- Shift;
- Blur;
- Gaussian Noise;
- Alteration of the brightness level.

Rotation and shift are used to simulate the possibility of the user not correctly centering the tag while using the acquisition tool. The brightness level alteration is supposed to model different lighting conditions in the room where the image is taken while the blur and Gaussian noise are used to simulate noise variation caused by the acquisition tool. The augmentation is randomly applied by instantiating the class *Compose* of the Albumentation library. The class returns a function that applies image augmentation following the pipeline based on the transformations listed in the class instance. In particular, the shift transformation is applied by taking the float provided as shift factor, to translate the pixels of the image leftwards or rightwards in the horizontal direction and upwards or downwards in the vertical direction. The shift direction depends on the sign of the parameter value. If only a float x is indicated, the augmentation will apply a shift factor randomly picked from the interval (-x,x). In order to apply the operation listed above, the parameters used were the shift factor, the rotation factor, the brightness factor, the blur factor and the variance range for noise. Since blurring is a technique that will neutralise the Gaussian noise, it is applied before than the Gaussian noise in the augmentation pipeline. An example of augmentation can be seen in figure 4.4.



(a) No augmention **(b)** With augmentation

Figure 4.4: Example of augmentation on tag 11

4.4 Performance Metrics

The metrics used to evaluate the performances of a trained model are based on the concept of false/true positive/negative. The first metric is visual and it corresponds to the Receiver operating characteristic (ROC), while the second one is the True positive rate value obtained for fixed values of False positive rate. A ROC curve is a graph showing the performance of a classification model at all classification thresholds. This curve plots the true positive rate against the false positive rate. ROC is usually used to evaluate binary classification model. Even though the dataset is composed of as many classes as the number of distinct tags are, the problem that the thesis aims to solve is binary. The Neural network is trained to understand whether two images are representative of the same crystal or not, regardless the specific class. Hence, there can be four possible outcomes:

1. true positive: a pair of images classified correctly as representing the same tag;

2. false positive: a pair of images classified wrongly as representing the same tag;

3. true negative: a pair of images classified correctly as representing different tags;

4. false negative: a pair of images classified wrongly as representing different tags.

The true positive rate, or sensitivity, in this project, is a measure of the probability that a pair of images is correctly classified as coming from the same tag. It can be expressed as the ratio between the true positives and the sum of the true positives and false negatives, as shown in expression 4.3.

$$TPR = \frac{TP}{TP + FN} \tag{4.3}$$

Since this type of object recognition is fundamental to discriminate original from copies, the true positive rate is a very important metric. On the other hand, the false positive rate is a measure of the probability that images representing different tags are recognised as the same tag. The formula used to express the FPR is formula 4.3 and consists of the ration between false positives and the sum of false positive and true negative.

$$FPR = \frac{FP}{FP + TN} \tag{4.4}$$

The relationship between the two can be shown by a ROC curve. It plots the true positive rate against the false positive rate over some thershold values. The ROC curves helps at visualizing how the threshold choice affect the classification results. In particular, the TPR for fixed values of FPR is a fundamental metric. What it is expected in the future is the possibility to effectively use the tags described in chapter 3 in anti-counterfeiting applications. Thus, the tolerance level for wrong positive classification should be considerably low. This means that for FPR values like the 0.01%, the correspondent TPR should be really high, i.t. TPR above the 90%. To help the understanding of the behaviour of the model for low FPR values, logarithmic plots of the ROC and tables containing the TPR and threshold computed for fixed FPR values were used.

Chapter 5

Experiments and results

In this chapter the experiments conducted using the model-based and data driven approach are presented. The objective is to find a scenario where the NCC algorithm fails in order to justify the use of a neural network.

5.1 NCC

For the set of experiments presented in this section, the similarity coefficient was computed for all possible pairs from the dataset. This means that the NCC was computed for a total of 4.950 *same_tag* pairs and about 9.900 of *diff_tag* pairs. This is due to the wish to investigate the distribution of the NCC over the data space and its relationship with the performance results shown in the next sections.

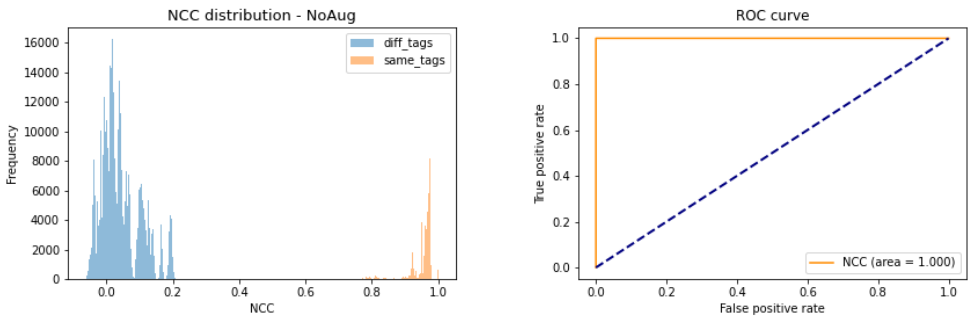
5.1.1 Experiment 1: NCC over not augmented images

The experiment was conducted using images where no augmentation was applied. The distribution of the NCC value over the pairs can be observed in figures 5.1a, 5.2a and 5.3a.

Tables 5.1, 5.2, 5.3 show the values of TPR and threshold for fixed values of FPR. The values of interest are, in particular:

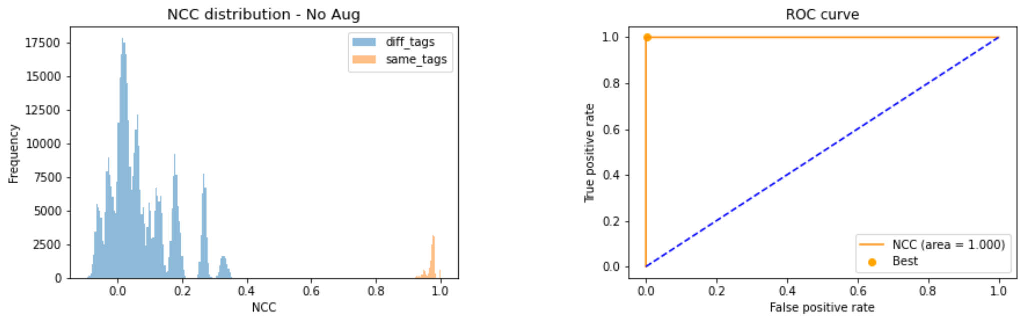
- FPR = 0.1;
- FPR = 0.01;
- FPR = 0.001;

It is recalled that it would be good to have TPR values of about the 90% and above for very low values of FPR. The results show that for the experiment with no augmentation applied on the image, the NCC succeds at guaranteing the wanted



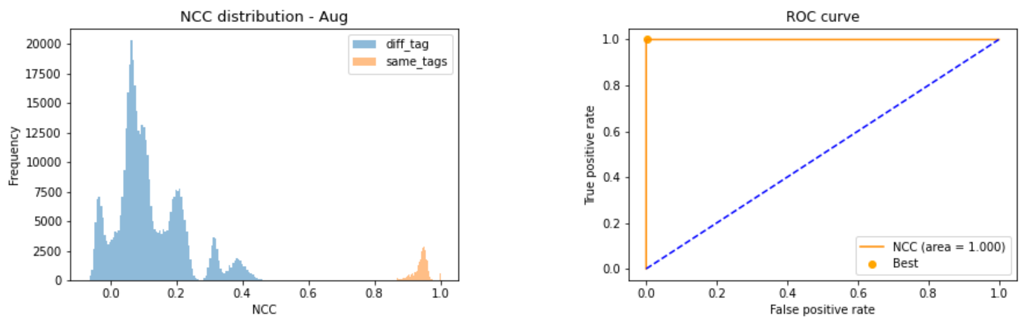
(a) NCC distribution **(b)** ROC curve

Figure 5.1: Images 300x300. Not Augmented



(a) NCC distribution **(b)** ROC curve

Figure 5.2: Images 60x60. Not Augmented



(a) NCC distribution **(b)** ROC curve

Figure 5.3: Images 30x30. Not Augmented

performance. However, it is worth to remember that the dataset on which these tests were carried out, was collected in a highly controlled environment and does

Table 5.1: TPR and threshold values for fixed FPR. Images 300x300

FPR	TPR	Threshold
0.1	1.0	0.1943
0.01	1.0	0.330
0.001	1.0	0.3461

Table 5.2: TPR and threshold values for fixed FPR. Images 60x60

FPR	TPR	Threshold
0.1	1.0	0.198
0.01	1.0	0.286
0.001	1.0	0.307

Table 5.3: TPR and threshold values for fixed FPR. Images 30x30

FPR	TPR	Threshold
0.1	1.0	0.236
0.01	1.0	0.408
0.001	1.0	0.441

not take into account eventual imperfections due to a human operator's errors nor due to the noise caused by the acquisition tool.

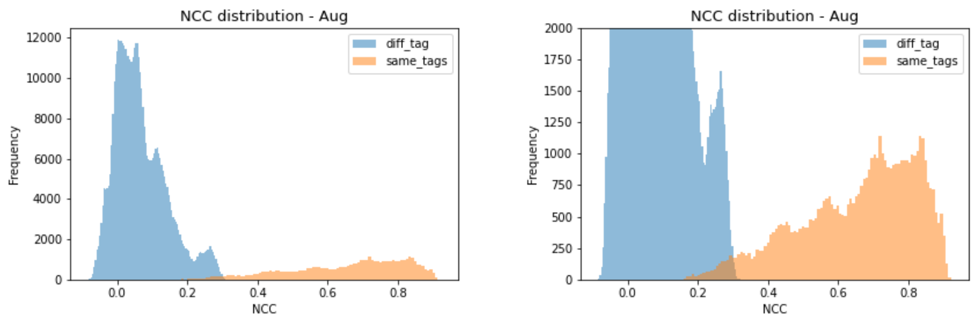
5.1.2 Experiment 2: NCC computed over Augmented Images. First set of parameters

To test the NCC algorithm in a more realistic setting, augmented images were used. The augmented operations are listed in section 4.3.2. Parameters for this experiment are shown in table 5.4.

Table 5.4: Augmentation parameters values for NCC - Experiment 2

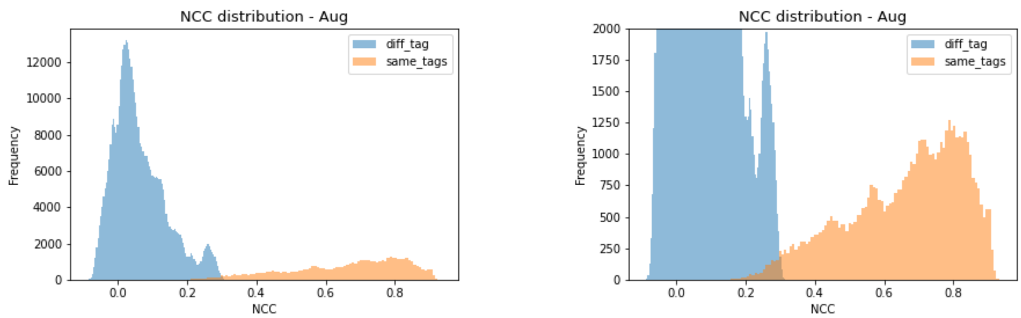
Shift	Rotation	Blur	Gaussian Noise	Brightness Level
0	3	0.3	3.0	0.3

The computed NCC distribution over the pair is showed in figures 5.4, 5.5 and 5.6 for all of the 3 considered datasets. The blue and orange parts of the plots correspond to the *diff_tag* and the *same_tag* distributions, respectively.



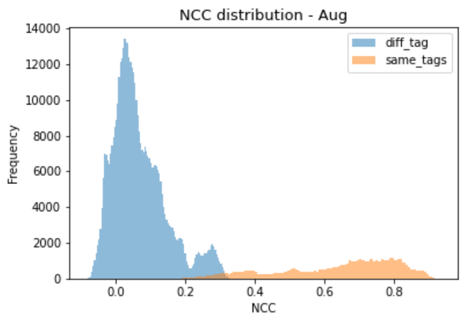
(a) NCC distribution **(b)** Zoom on intersection region

Figure 5.4: Images 300x300. Augmentation 1

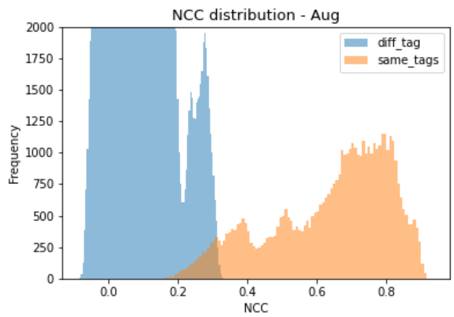


(a) NCC distribution **(b)** Zoom on intersection region

Figure 5.5: Images 60x60. Augmentation 1



(a) NCC distribution



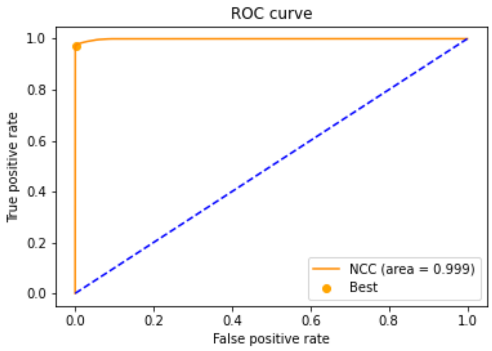
(b) Zoom on intersection region

Figure 5.6: Images 30x30. Augmentation 1

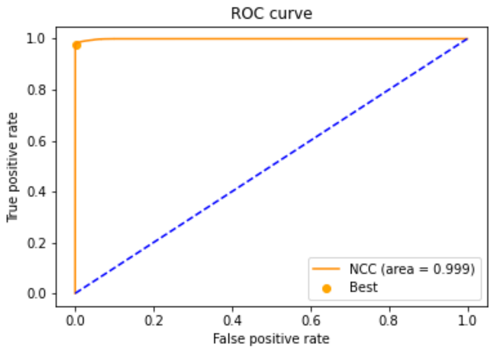
What is immediate by comparing the distributions between experiment 1 and 2 is that the add of augmentation to the dataset flattens the distribution for the pairs built with same tag images. This is justified by the fact that the transform operation will change the value of the image pixels. If the pixel value change, the similarity coefficient will also change in a proportional way to the global pixel difference between the two retakes. For the same reason, a slight spread out of the blue distribution is also observed. Since the images will have their pixels randomly changed, the NCC value for two images of the same type will result a little higher than the no-augmentation case. However, the distribution for different tag pairs is still centered around near-zero values. The zoom on the lower part of the plots 5.4b, 5.5b and 5.6b show an intersection region between the orange and the blue zones. This suggests that there will be some pairs of images that will be mislabelled for a certain range of thresholds. This is mirrored in the ROC curve computed for the whole distribution. Figure 5.7 shows a small degradation of the NCC performance. However, as also proved by the FPR tables 5.5, 5.6 and 5.7, the NCC still performs really well and in the experiment described in section 5.1.1.

Table 5.5: TPR and threshold values for fixed FPR. Augmentation 1. Images 300x300

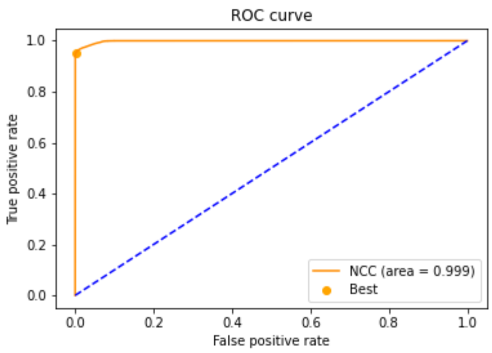
FPR	TPR	Threshold
0.1	0.99	0.168
0.01	0.98	0.276
0.001	0.97	0.297



(a) 300x300



(b) 60x60



(c) 30x30

Figure 5.7: ROC Curves for Experiment 2

5.1.3 Experiment 3: NCC computed over set n.2 of Augmented images

Experiment 3 is the same as the one in 5.1.2 but with another set of augmentation parameters. The said set is summarised in table 5.8.

Table 5.6: TPR and threshold values for fixed FPR. Augmentation 1. Images 60x60

FPR	TPR	Threshold
0.1	0.99	0.173
0.01	0.98	0.278
0.001	0.97	0.297

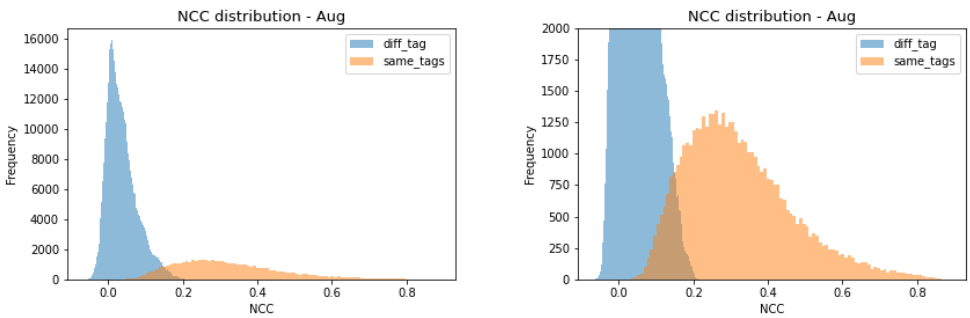
Table 5.7: TPR and threshold values for fixed FPR. Augmentation 1. Images 30x30

FPR	TPR	Threshold
0.1	0.99	0.178
0.01	0.96	0.292
0.001	0.94	0.312

Table 5.8: Augmentation parameters values for NCC - Experiment 2

Shift	Rotation	Blur	Gaussian Noise	Brightness Level
0.002	3	0.3	5.0	0.5

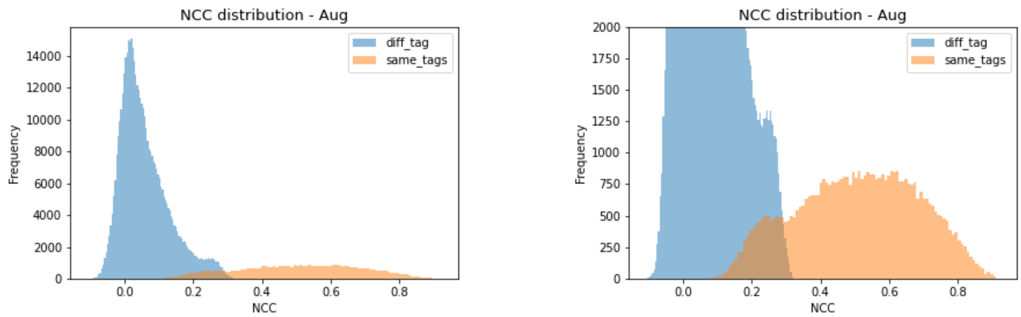
With respect to the parameters value in table 5.4, the differences are that the shift factor, rotation and brightness level were all increased.



(a) NCC distribution **(b)** Zoom on intersection region

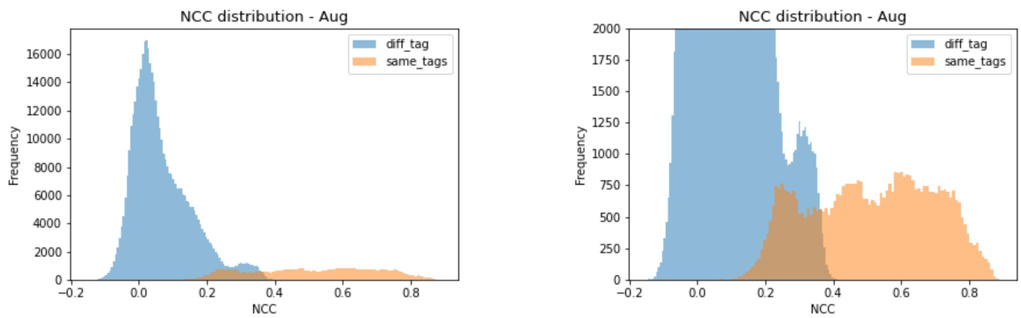
Figure 5.8: Images 300x300. Augmentation 2

Figures 5.8, 5.9 and 5.10 show the distribution for the NCC value over the pairs for the 3 datasets. The effect of the augmentation is more remarked with respect to experiment 2 and that is visible from the larger intersection area. The said area present not only a wider interval of NCC values that are assigned to both types of pairs (diff_tags and same_tags pairs) but the frequency with which the similar



(a) NCC distribution **(b)** Zoom on intersection region

Figure 5.9: Images 60x60. Augmentation 2



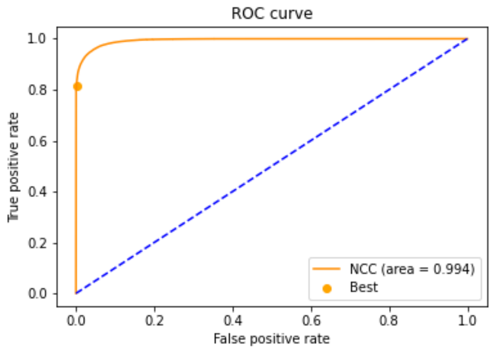
(a) NCC distribution **(b)** Zoom on intersection region

Figure 5.10: Images 30x30. Augmentation 2

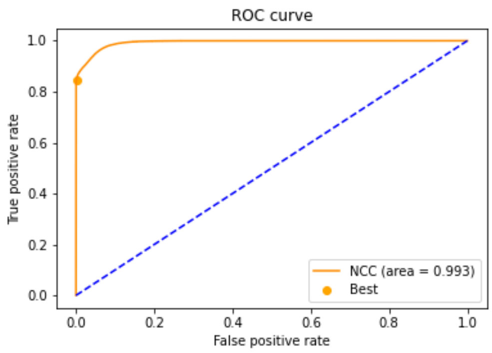
values are assigned to both pair types is higher with respect to experiment 2. These plots suggest a lower value of TPR for the FPR values proposed in section 5. The expectations are met by the ROC curves plotted in figure 5.11 and by tables 5.9, 5.10 and 5.11. The NCC algorithm fails at providing a TPR higher than 90%, especially for the lowest values of proposed FPR and for the images with lower resolution. This shows that the model-based approach taken under consideration can fail depending on the imperfections of the image.

Table 5.9: TPR and threshold values for fixed FPR. Augmentation 1. Images 300x300

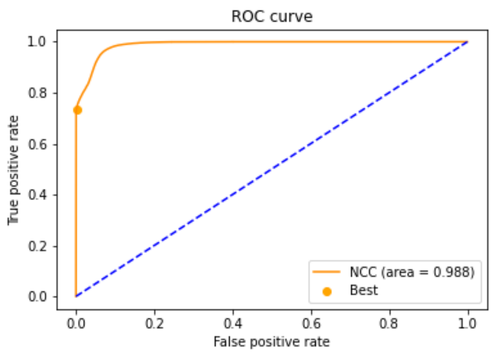
FPR	TPR	Threshold
0.1	0.98	0.094
0.01	0.89	0.157
0.001	0.81	0.190



(a) 300x300



(b) 60x60



(c) 30x30

Figure 5.11: ROC Curves for Experiment 3

Table 5.10: TPR and threshold values for fixed FPR. Augmentation 1. Images 60x60

FPR	TPR	Threshold
0.1	0.98	0.161
0.01	0.87	0.272
0.001	0.84	0.301

Table 5.11: TPR and threshold values for fixed FPR. Augmentation 1. Images 30x30

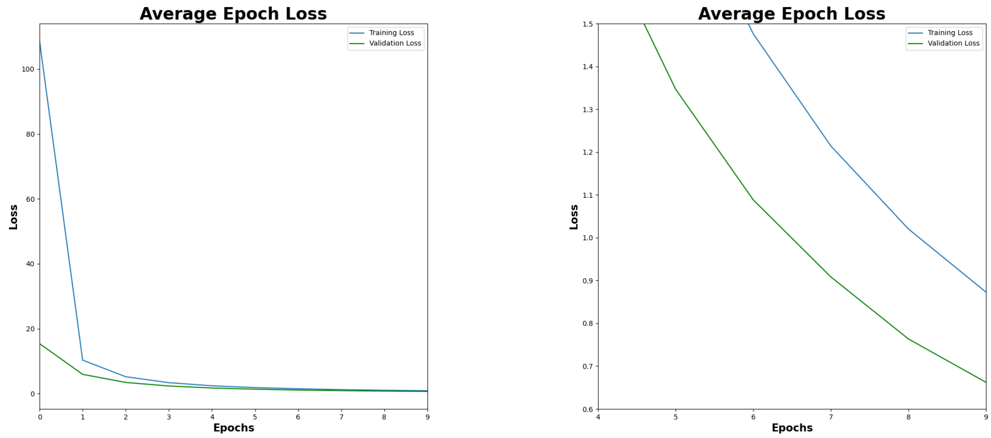
FPR	TPR	Threshold
0.1	0.98	0.190
0.01	0.77	0.340
0.001	0.73	0.373

5.2 Siamese Network

The goal of the following experiments try to solve the physical-object detection problem with the augmentation parameters that lead the NCC to non-ideal performances. The approach used for the experiments in sections 5.2.1 and 5.2.2 make use of the Siamese network from section 4.2. In both experiments, the network is fed with batches of size 64 due to the technical limitations of the used GPU. The main differences between the two are the data preparation and the number of pairs generated for training, validation and testing. In experiment 1 the model is trained over 10 tags and the splitting is the usual strategy in machine learning and presented in figure 4.3a. From the three sets, pairs for both the experiments are generated offline for simplicity of implementation and bug detection, as said in section 4.3.1. The number of pairs generated for the network training in the first experiment is 10000 pairs for the training set, 5000 for the validation and test set. Regarding the second one, the training pairs were 20000, the validation and test were 10000. For what concerns the splitting strategy, the models are trained over 10 tags and validated/tested over an additional set of 5 never seen tags. Thus, experiment 2 follows the splitting strategy illustrated in figure 4.3b.

5.2.1 Experiment 1: Splitting strategy n.1

The first dataset on which the neural network was trained was the one with images of resolution 300x300 pixels. The learning schedule used is the one explained in section 4.2.1, with a starting learning rate was 0.0001. What was interesting to investigate was the behaviour of the network changing during the training and validation phases. After the first 10 epochs the average epoch loss computed for the model showed a great misalignment between training and validation set. this behavior is due to the fact that training loss is averaged over the epoch, while validation is evaluated at the end of the epoch, hence on a better network. Indeed, figure 5.12 and table 5.12 show a validation loss that is less than the training loss at every epoch.



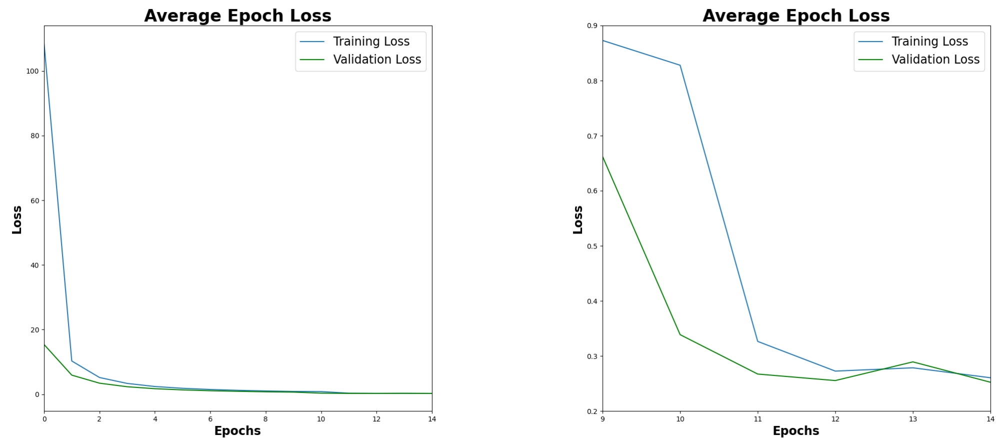
(a) No zoom **(b)** Zoom on last 6 epochs

Figure 5.12: Average epoch loss. 10 epochs. resolution of 300x300

Table 5.12: Average epoch loss for training and validation. resolution 300x300.

Train loss	Validation loss
108.607	15.384
10.293	5.919
5.1838	3.428
3.353	2.329
2.399	1.722
1.845	1.346
1.476	1.088
0.873	0.662

By continuing the training over more epochs there is an increasing improvement both in terms of loss, as shown in figures 5.13 and 5.14, and the ROC curves on the testing dataset, figure 5.15 and 5.16. Figure 5.15b and 5.16b underline the positive shift in the ability of the model to do its task for low values of FPR. The model trained over 15 epochs, even if it could show a better behaviour, it is still better than a random classifier. This does not happen for low values of TPR, which are the ones of our interest, as shown by figure 5.15b. The shape of the ROC for the Network is clearly similar to the one computed for a random classifier. This observation is confirmed by the TPR values for the fixed FPR values in table 5.13. Results significantly improves in terms of TPR when the Network reaches the 30 epochs. Both figure 5.16b and table 5.13 show a perfect classifier.



(a) No zoom **(b)** Zoom on last 5 epochs

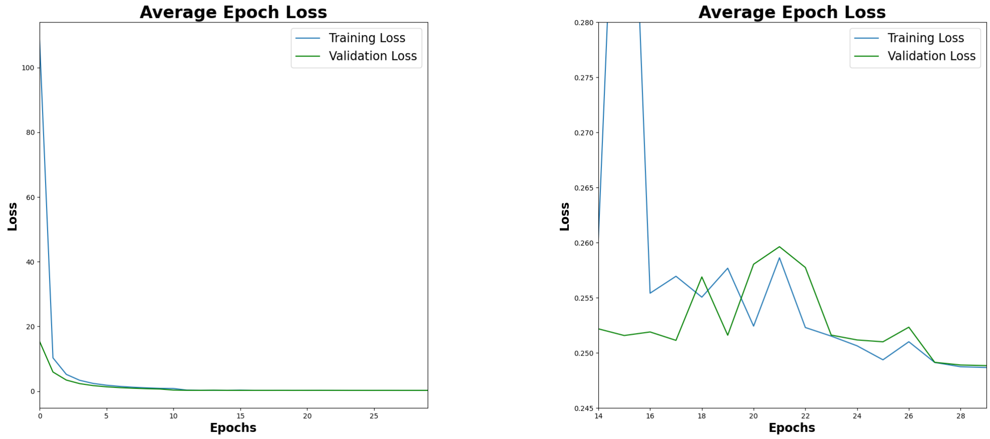
Figure 5.13: Average epoch loss. 15 epochs. resolution of 300x300

Table 5.13: TPR and threshold values for fixed FPR. Images 300x300. Model trained over 15 epochs

FPR	TPR	Threshold
0.1	0.4132	0.073
0.01	0.227	0.084
0.001	0.182	0.091

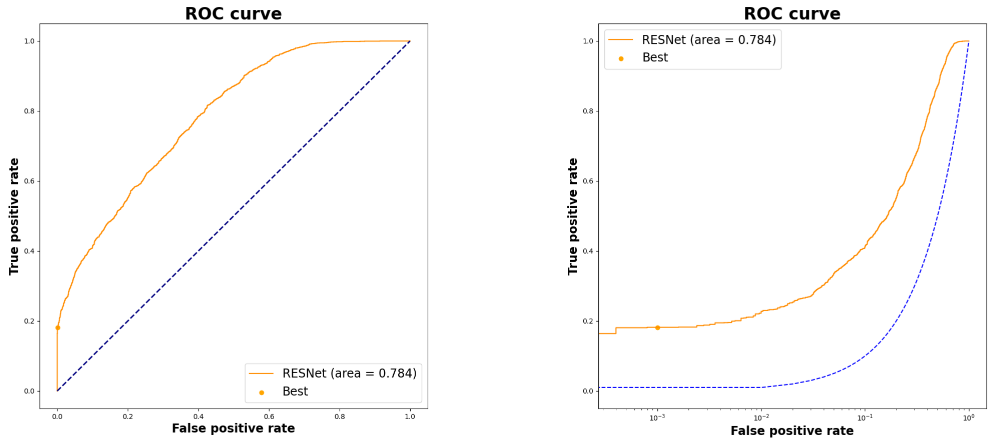
Table 5.14: TPR and threshold values for fixed FPR. Images 300x300. Model trained over 30 epochs

FPR	TPR	Threshold
0.1	1.0	0.041
0.01	0.98	0.048
0.001	0.95	0.053



(a) No zoom **(b)** Zoom on last 15 epochs

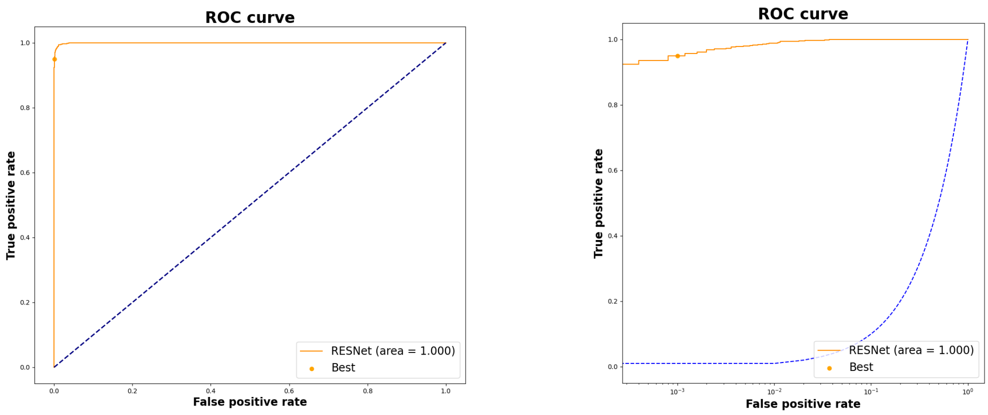
Figure 5.14: Average epoch loss. 30 epochs. resolution of 300x300



(a) Linear scale **(b)** Semi-Log scale

Figure 5.15: ROC Curve computed over testing set. 15 epochs. Resolution of 300x300

It is worth to mention that, in this experiment, the TPR, the ROC curve and the AUC parameter are all performance metric evaluated on tags that the model has already seen during its training phase. Hence, it is not possible to have a measure of how well the network is at generalization. In order to understand the ability of the trained network to discriminate same tag pairs from different tags

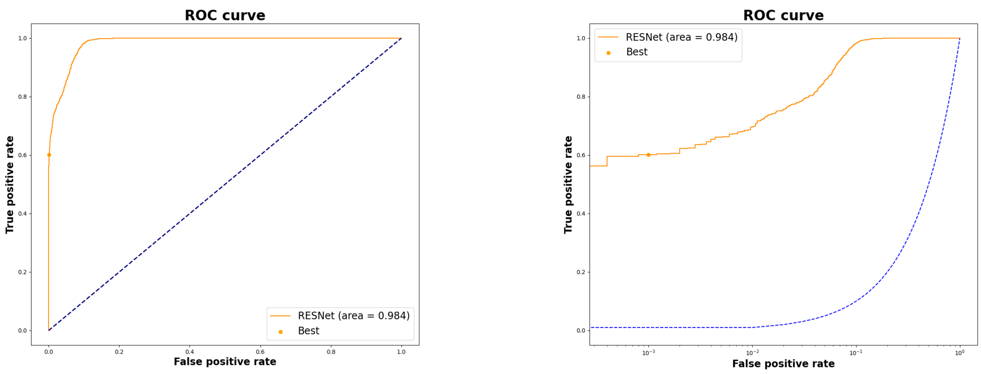


(a) Linear scale

(b) Semi-Log scale

Figure 5.16: ROC Curve computed over testing set. 30 epochs. Resolution of 300x300

pairs on new tags, an additional set of 5 tags was chosen. The results of the tests made on the new test set can be observed in figure 5.17 and table 5.15.



(a) Linear scale

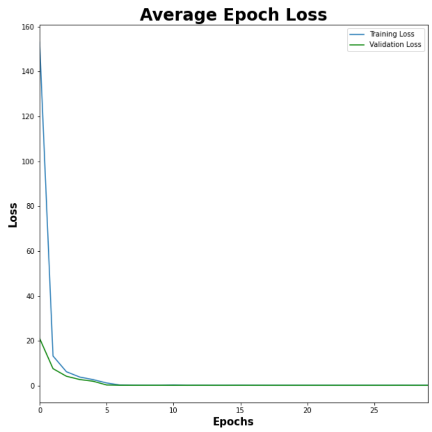
(b) Semi-Log scale

Figure 5.17: ROC Curve computed over new testing set. 30 epochs. Resolution of 300x300

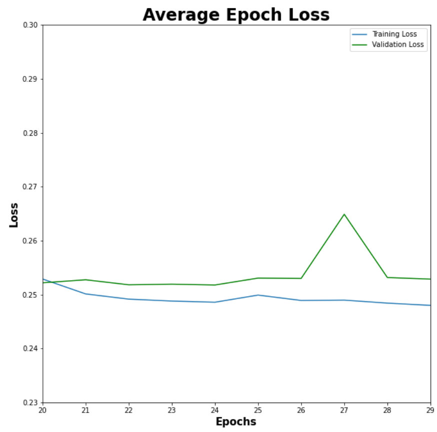
The ROC curve and the high AUC score (visible from the legend of the ROC plot) indicate an excellent classifier for the new 5 tags, this does not happen if very low values of FPR are requested. The TPR lowers to a value between the 60 and 70%. The same behaviour can be observed for the models trained over the dataset

Table 5.15: TPR and threshold values for fixed FPR. Images 300x300. Model trained over 30 epochs. Test on new tags.

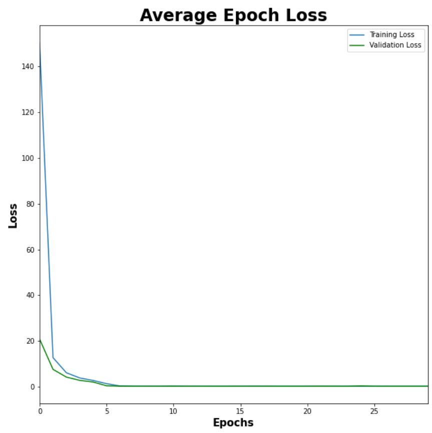
FPR	TPR	Threshold
0.1	0.9832	0.036
0.01	0.6972	0.046
0.001	0.6012	0.049



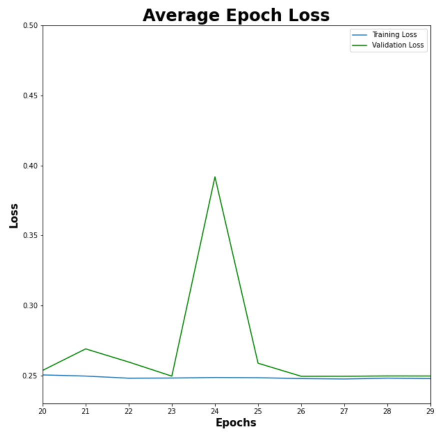
(a) No zoom. 30 epochs. 60x60



(b) Zoom on last 10 epochs. 60x60

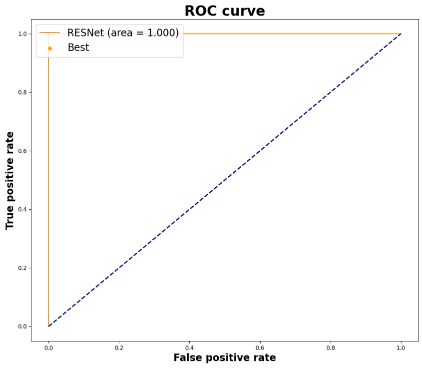


(c) No zoom. 30 epochs. 30x30

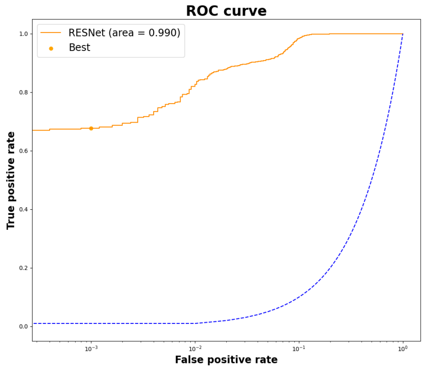


(d) Zoom on last 10 epochs. 30x30

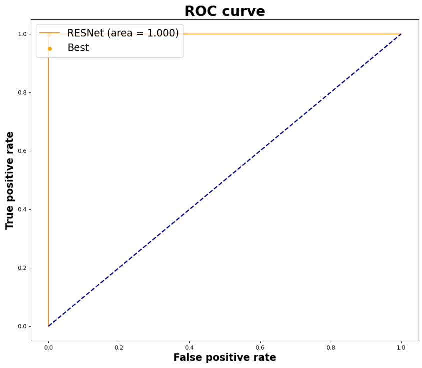
Figure 5.18: Average epoch loss



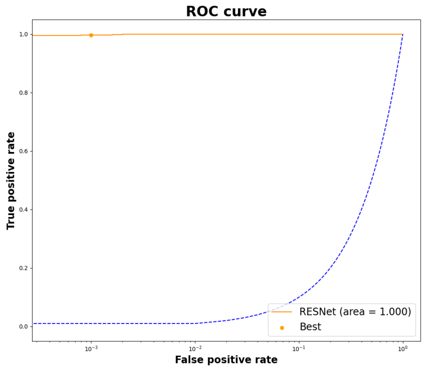
(a) Linear Scale. Dataset 60x60



(b) Semi-Log Scale. Dataset 60x60



(c) Linear Scale. Dataset 30x30



(d) Semi-Log scale. Dataset 30x30

Figure 5.19: ROC Curves for dataset 30x30 and 60x60

with lower resolutions. Figures 5.18 and 5.19 show the loss and the ROC curve for the model trained over the 60x60 and 30x30 datasets respectively. The two models, trained till the saturation point, behave well on a test set made of tags already seen during the training phase. However, when tested on a set of new tags, the TPR for low values of FPR significantly decrease. This is made more evident by tables 5.17 and 5.16 that show a comparison between test tags and new test tags for both the dataset 60x60 and 30x30. The fact that all 3 networks presents outstanding results if tested on already seen tags, but not as well on tags never seen before is hinting at a possible overfitting oto the training set. For this reason, 2 strategies were adopted in section 5.2.2 to improve the models' performances.

Table 5.16: TPR and threshold values for fixed FPR. Test ober known tags vs Test over never seen tags. Images 30x30.

Test set	FPR	TPR	Threshold
Known Tags	0.1	1.0	0.42
	0.01	1.0	0.43
	0.001	0.99	0.44
Never seen Tags	0.1	0.97	0.035
	0.01	0.74	0.047
	0.001	0.70	0.051

Table 5.17: TPR and threshold values for fixed FPR. Test over known tags vs Test over never seen tags. Images 60x60.

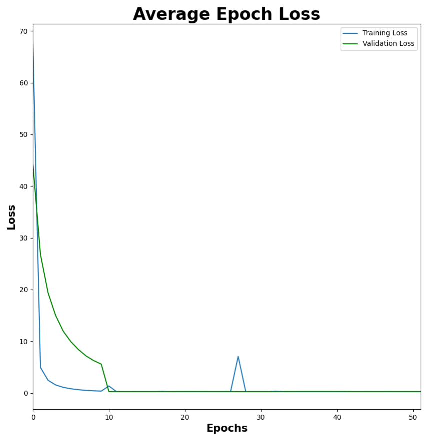
Test set	FPR	TPR	Threshold
Known Tags	0.1	1.0	0.0618
	0.001	1.0	0.05301
	0.0001	1.0	0.05
Never seen Tags	0.1	0.98	0.034
	0.01	0.82	0.047
	0.001	0.67	0.051

5.2.2 Experiment 2: splitting strategy n.2

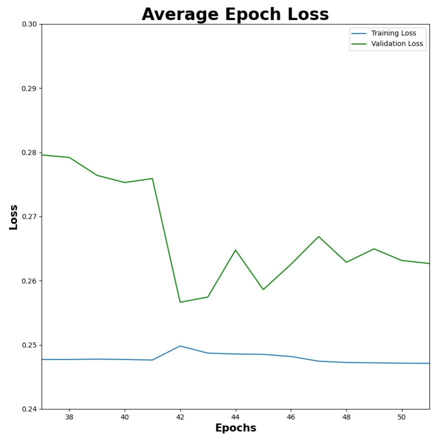
The two actions taken for this experiment both aim at increasing the number of data on which the model is trained. For simplicity, they are listed below:

1. increase the number of images in the training set
2. increase the number of generated pairs

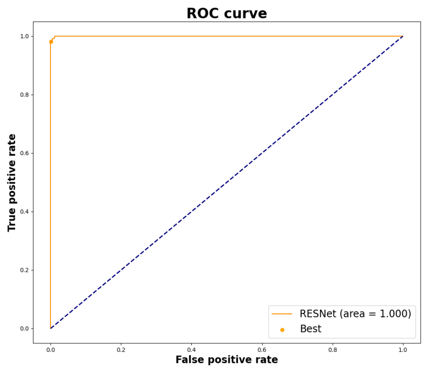
By implementing the splitting strategy illustrated in figure 4.3b, the images were assigned to the training, validation and test set. The reason why the new strategy allows to satisfy the first point listed above, is the following. By taking all the acquisitions of certain tags, the model will be able to have access to more versions of each tag and to learn its features better. In this way the training set will be composed of 100 images multiplied 10 tags, for a total of 1000 images. In experiment 1, the training set was composed of 60 images multiplied 10 tags, which corresponds to a total of 600 images. After the splitting of the dataset, 20000, 10000 and 10000 pairs were randomly generated for the training, validation and test set respectively. Once the list of pairs was generated, batches of augmented pairs with size 64 are fed to the neural network in the same way as for the models of section 5.2.1. The learning schedule used for this experiment is, again, the one explained in section 4.2.1, with a starting learning rate was 0.0001. The results for the models trained on 300x300, 60x60 and 30x30 images can be observed in figures 5.20, 5.21 and 5.22.



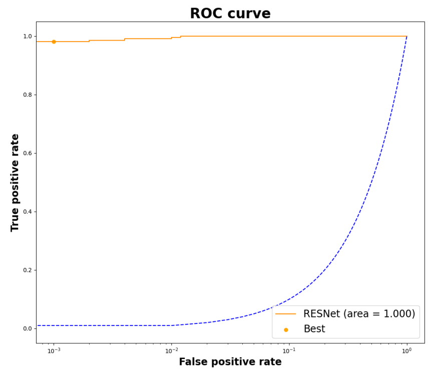
(a) No zoom. 52 epochs. 300x300



(b) Zoom on last 15 epochs. 300x300

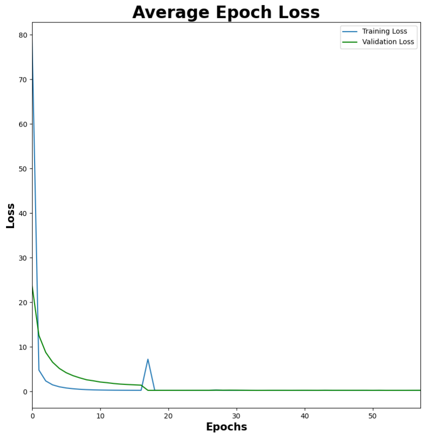


(c) ROC. Linear Scale.

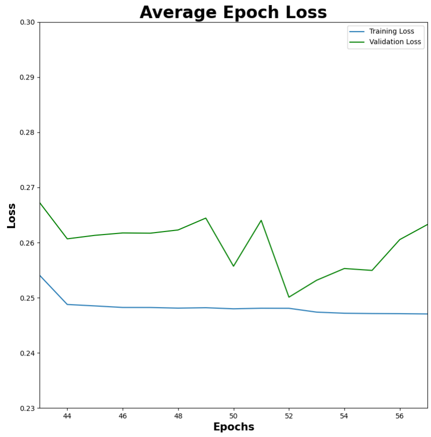


(d) ROC. Semi-Logarithmic scale

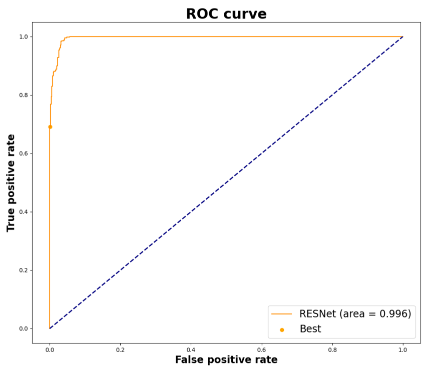
Figure 5.20: Experiment 2. Model trained over images with resolution 300x300 pixels



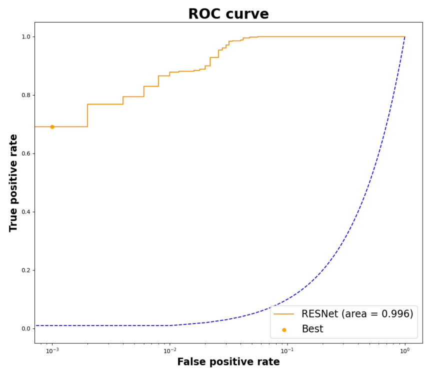
(a) No zoom. 58 epochs. 300x300



(b) Zoom on last 15 epochs. 60x60

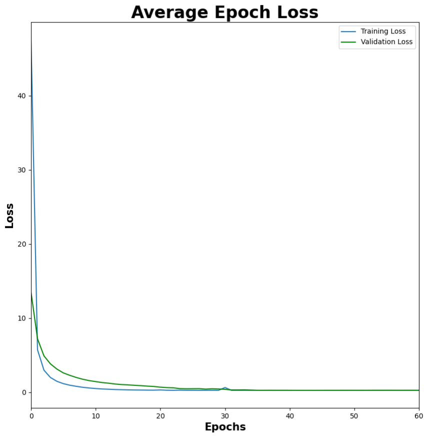


(c) ROC. Linear Scale.

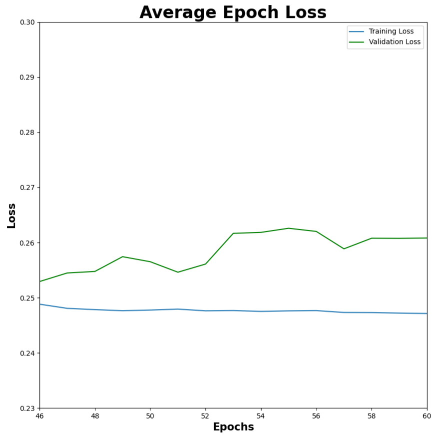


(d) ROC. Semi-Logarithmic scale

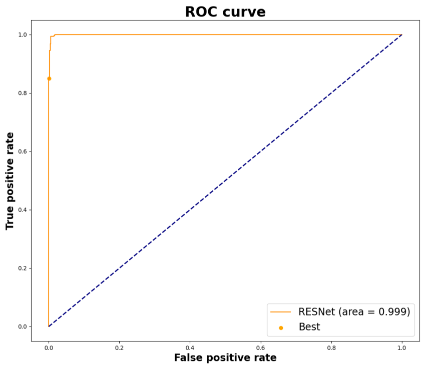
Figure 5.21: Experiment 2. Model trained over images with resolution 60x60 pixels



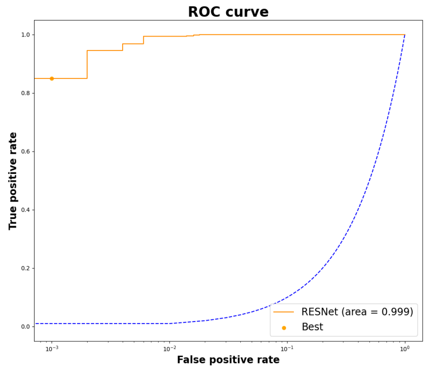
(a) No zoom. 61 epochs. 300x300



(b) Zoom on last 15 epochs. 300x300



(c) ROC. Linear Scale.



(d) ROC. Semi-Logarithmic scale

Figure 5.22: Experiment 2. Model trained over images with resolution 30x30 pixels

As expected, different trends in the training and validation loss were registered. The loss plots from figures 5.20, 5.21 and 5.22 show that the difference between the two curves is more prominent than the case presented in section 5.2.1, regardless the image resolution. However, the ROC plots from the same figures show good classifiers when tested on tags that were never seen by the network during its training phase. The TPR values computed for fixed values of FPR, show better results with respect to the ones obtained in section 5.2.1. For both experiments the tests were carried out on the same set of tags to allow a comparison in terms of performances. Table 5.18 show that for the highest image resolution the performances are more than excellent, since the model registers a TPR that is always above the 90%. Different are the results for the models trained over images of 30x30 and 60x60

Table 5.18: Dataset of 300x300 images. Experiment 1 vs Experiment 2. TPR and Threshold computed for fixed FPR values.

	FPR	**TPR**	**Threshold**
Experiment 1	0.1	0.98	0.036
	0.01	0.69	0.046
	0.001	0.60	0.049
Experiment 2	0.1	1.0	0.036
	0.01	0.99	0.037
	0.001	0.98	0.038

Table 5.19: Dataset of 60x60 images. Experiment 1 vs Experiment 2. TPR and Threshold computed for fixed FPR values.

	FPR	**TPR**	**Threshold**
Experiment 1	0.1	0.9840	0.034
	0.01	0.82	0.047
	0.001	0.67	0.051
Experiment 2	0.1	1.0	0.030
	0.01	0.88	0.039
	0.001	0.69	0.047

Table 5.20: Dataset of 30x30 images. Experiment 1 vs Experiment 2. TPR and Threshold computed for fixed FPR values.

	FPR	**TPR**	**Threshold**
Experiment 1	0.1	0.97	0.035
	0.01	0.74	0.047
	0.001	0.70	0.050
Experiment 2	0.1	1.0	0.040
	0.01	0.99	0.042
	0.001	0.85	0.050

pixels. Table 5.19 show an improvement of the model trained over images 60x60 in experiment 2 compared to experiment 1. Same comparison and conclusions can be made by looking at table 5.20 for the model trained over 30x30 images.

Chapter 6

Conclusions

What can be drawn from the results presented in chapter 5 are three main considerations:

- High quality data was fundamental for the project;
- The model-driven approach used for this project has limits if the images are acquired in a non-ideal setting;
- A data-driven approach based on siamese network can be used for physical object detection but the number of pairs should be increased.

Indeed, results from section 5.2.2, show that just by increasing the dataset used to train a simple Siamese Network (see architecture from 4.2), it was possible to obtain really high values of TPR, which is the metric that is the most interesting for our project. Considering the anti-counterfeiting applications the tags will be used for and their small dimension, it is reasonable to prioritize methodologies that do not allow the frequent misclassification of fake tags. Hence, the importance of the TPR for low FPR values must be taken under consideration. In this thesis it was provided proof of the fact that a model-driven approach alone, like the NCC algorithm presented in section 4.1, cannot be used alone for the CSRs detection task. The limit of this type of algorithms is that it cannot be trained to filter out the noise from those features that make the tag, especially the technology it is made of, unique. Different is the case for data-driven approaches like the ones based on the use of neural network. Given the final version of the technology, a reasonable amount of data can be collected and used to train a siamese network to eventually detect fake tags from real ones. As mentioned in section 1.1, the objective of this work was to lay down a first initial milestone in the field of physical-object detection with cholesteric crystals. The main contribution of this thesis to the project that involves the csr technology are the proposal of a framework for the generation of high quality data (chapter 3) specifically for the project, the proof that even simple siamese networks can provide a great aid for the discrimination of tags acquisitions (chapter 5) with a limited amount of labeled images.

6.1 Future Work

In this section, some ideas to further improve the performance of a data-driven model for the project are presented. The priority should be given to the increase of training pairs. It would be interesting to use the numbers of training, validation and test pairs used by Koch et al. in their paper [28]. If increasing the pairs would not be enough, it is suggested to investigate the impact of a larger dataset on the network's performances. Considering that in this project 100 acquisitions for tag were collected, it would be appropriate to collect between 200 and 500 acquisitions for each tag of a given sample. Since the final version of the tag will be supposedly able to generate optical pattern that will make the CSR unique, it would be beneficial to have more acquisitions per class to let the neural network learn the features that characterise the technology of the tags. Secondly, experiments with a network trained with contrastive loss based on the cosine distance could be proposed. The main reason lays in the results obtained for the NCC algorithm (section 5.1). Even if the TPR values were not as high as desired, the NCC algorithm was still able to get a TPR of around the 80% for low values of FPR. Hence, two leads could be followed. The first one is the comparison between the Siamese network with the architecture illustrated in chapter 4 and a Siamese network whose architecture differs only on the use of a contrastive loss based on the cosine distance. The second experiment would consist on the use of sister networks whose layers are based on cosine similarity. A type of network architecture like the one implemented by Luo Chunjie et al. show that using cosine similarity instead of dot product can significantly improve the performance of a neural network on the test set. [32]

Bibliography

[1] OECD. *The Economic Impact of Counterfeiting and Piracy*. 2007, p. 25. DOI: \url{https://doi.org/https://doi.org/10.1787/9789264037274-en}. URL: %5Curl%7Bhttps://www.oecd-ilibrary.org/content/publication/9789264037274-en%7D (cit. on p. 1).

[2] Jeremy Haken. «Transnational crime in the developing world». In: *Global financial integrity* 32.2 (2011), pp. 11–30 (cit. on p. 1).

[3] OECD and European Union Intellectual Property Office. *Executive summary*. 2019. DOI: \url{https://doi.org/https://doi.org/10.1787/fc47e382-en}. URL: %5Curl%7Bhttps://www.oecd-ilibrary.org/content/component/fc47e382-en%7D (cit. on p. 1).

[4] OECD and European Union Intellectual Property Office. *Counterfeiting – The current landscape*. 2019. DOI: \url{https://doi.org/https://doi.org/10.1787/fc8b3ec5-en}. URL: %5Curl%7Bhttps://www.oecd-ilibrary.org/content/component/fc8b3ec5-en%7D (cit. on p. 1).

[5] Yong Geng, JungHyun Noh, Irena Drevensek-Olenik, Romano Rupp, Gabriele Lenzini, and Jan PF Lagerwall. «High-fidelity spherical cholesteric liquid crystal Bragg reflectors generating unclonable patterns for secure authentication». In: *Scientific reports* 6.1 (2016), pp. 1–9 (cit. on pp. 1, 3, 4).

[6] Gabriele Lenzini, Samir Ouchani, Peter Roenne, Peter Y. A. Ryan, Yong Geng, Jan Lagerwall, and JungHyun Noh. «Security in the shell: An optical physical unclonable function made of shells of cholesteric liquid crystals». In: *2017 IEEE Workshop on Information Forensics and Security (WIFS)*. 2017, pp. 1–6. DOI: 10.1109/WIFS.2017.8267644 (cit. on pp. 3, 4).

[7] WD St John, WJ Fritz, ZJ Lu, and D-K Yang. «Bragg reflection from cholesteric liquid crystals». In: *Physical Review E* 51.2 (1995), p. 1191 (cit. on p. 3).

[8] JungHyun Noh et al. Martin Urbanski Catherine G Reyes. «Liquid crystals in micron-scale droplets, shells , fibers». In: *Scientific reports* 29.13 (2017) (cit. on pp. 4, 11).

[9] Monica Arenas, Huseyin Demirci, and Gabriele Lenzini. «Cholesteric Spherical Reflectors as Physical Unclonable Identifiers in Anti-counterfeiting». In: *The 16th International Conference on Availability, Reliability and Security*. 2021, pp. 1–11 (cit. on p. 4).

[10] Gabriele Lenzini, Samir Ouchani, Peter Roenne, Peter YA Ryan, Yong Geng, Jan Lagerwall, and JungHyun Noh. «Security in the shell: An optical physical unclonable function made of shells of cholesteric liquid crystals». In: *2017 IEEE Workshop on Information Forensics and Security (WIFS)*. IEEE. 2017, pp. 1–6 (cit. on p. 4).

[11] Alireza Shamsoshoara, Ashwija Korenda, Fatemeh Afghah, and Sherali Zeadally. «A survey on physical unclonable function (PUF)-based security solutions for Internet of Things». In: *Computer Networks* 183 (2020), p. 107593 (cit. on p. 4).

[12] Shital Joshi, Saraju P Mohanty, and Elias Kougianos. «Everything you wanted to know about PUFs». In: *IEEE Potentials* 36.6 (2017), pp. 38–46 (cit. on p. 4).

[13] Yansong Gao, Said F Al-Sarawi, and Derek Abbott. «Physical unclonable functions». In: *Nature Electronics* 3.2 (2020), pp. 81–91 (cit. on p. 4).

[14] Srinivas Devadas, Edward Suh, Sid Paral, Richard Sowell, Tom Ziola, and Vivek Khandelwal. «Design and implementation of PUF-based" unclonable" RFID ICs for anti-counterfeiting and security applications». In: *2008 IEEE international conference on RFID*. IEEE. 2008, pp. 58–64 (cit. on p. 4).

[15] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. «Physical one-way functions». In: *Science* 297.5589 (2002), pp. 2026–2030 (cit. on p. 4).

[16] Akshay Wali, Akhil Dodda, Yang Wu, Andrew Pannone, Likhith Kumar Reddy Usthili, Sahin Kaya Ozdemir, Ibrahim Tarik Ozbolat, and Saptarshi Das. «Biological physically unclonable function». In: *communications Physics* 2.1 (2019), pp. 1–10 (cit. on p. 4).

[17] Kevin Gurney. *An introduction to neural networks*. CRC press, 2018 (cit. on p. 5).

[18] Arthur Arnx. *First neural network for beginners explained*. URL: `https://towardsdatascience.com/first-neural-network-for-beginners-explained-with-code-4cfd37e06eaf` (visited on 12/10/2021) (cit. on pp. 5, 6).

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. «Deep learning (adaptive computation and machine learning series)». In: *Cambridge Massachusetts* (2017), p. 82 (cit. on p. 6).

[20] Russell Reed and Robert J MarksII. *Neural smithing: supervised learning in feedforward artificial neural networks.* Mit Press, 1999 (cit. on p. 6).

[21] Sunitha Basodi, Chunyan Ji, Haiping Zhang, and Yi Pan. «Gradient amplification: An efficient way to train deep neural networks». In: *Big Data Mining and Analytics* 3.3 (2020), pp. 196–207 (cit. on p. 6).

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. «Deep residual learning for image recognition». In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 770–778 (cit. on pp. 6, 7).

[23] Mohamed Hammad, Paweł Pławiak, Kuanquan Wang, and Udyavara Rajendra Acharya. «ResNet-Attention model for human authentication using ECG signals». In: *Expert Systems* 38.6 (2021), e12547 (cit. on p. 7).

[24] Arshi Husain and Virendra P Vishvakarma. «Face Recognition Method Based on Residual Convolution Neural Network». In: *IOP Conference Series: Materials Science and Engineering.* Vol. 1228. 1. IOP Publishing. 2022, p. 012005 (cit. on p. 7).

[25] Li Fei-Fei, Robert Fergus, and Pietro Perona. «One-shot learning of object categories». In: *IEEE transactions on pattern analysis and machine intelligence* 28.4 (2006), pp. 594–611 (cit. on p. 8).

[26] Davide Chicco. «Siamese neural networks: An overview». In: *Artificial Neural Networks* (2021), pp. 73–94 (cit. on p. 8).

[27] Yuji Roh, Geon Heo, and Steven Euijong Whang. «A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective». In: *IEEE Transactions on Knowledge and Data Engineering* 33.4 (2021), pp. 1328–1347. DOI: 10.1109/TKDE.2019.2946162 (cit. on p. 9).

[28] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. «Siamese neural networks for one-shot image recognition». In: *ICML deep learning workshop.* Vol. 2. Lille. 2015 (cit. on pp. 9, 60).

[29] Noemi G Gyori, Marco Palombo, Christopher A Clark, Hui Zhang, and Daniel C Alexander. «Training data distribution significantly impacts the estimation of tissue microstructure with machine learning». In: *Magnetic resonance in medicine* 87.2 (2022), pp. 932–947 (cit. on p. 9).

[30] Raia Hadsell, Sumit Chopra, and Yann LeCun. «Dimensionality reduction by learning an invariant mapping». In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06).* Vol. 2. IEEE. 2006, pp. 1735–1742 (cit. on p. 31).

[31] Degan Hao, Lei Zhang, Jules Sumkin, Aly Mohamed, and Shandong Wu. «Inaccurate labels in weakly-supervised deep learning: Automatic identification and correction and their impact on classification performance». In: *IEEE journal of biomedical and health informatics* 24.9 (2020), pp. 2701–2710 (cit. on p. 33).

[32] Chunjie Luo, Jianfeng Zhan, Xiaohe Xue, Lei Wang, Rui Ren, and Qiang Yang. «Cosine normalization: Using cosine similarity instead of dot product in neural networks». In: *International Conference on Artificial Neural Networks*. Springer. 2018, pp. 382–391 (cit. on p. 60).