



**ScuDo**  
Scuola di Dottorato ~ Doctoral School  
WHAT YOU ARE, TAKES YOU FAR



Doctoral Dissertation  
Doctoral Program in Control and Computer Engineering (30<sup>th</sup> cycle)

# Computer-Aided Technologies for Food Risk Assessment

**Francesco Rossi**

\* \* \* \* \*

**Supervisor**

Prof. Alfredo Benso

**Doctoral Examination Committee:**

Cornale Paolo, Referee, UNITO - Italy  
Montrucchio Bartolomeo, POLITO - Italy  
Peletto Simone, IZSTO - Italy  
Tonda Alberto, Referee, INRA - France  
Valentini Sabina, IAR - Italy

Politecnico di Torino  
April 2018

This thesis is licensed under a Creative Commons License, Attribution - Noncommercial-NoDerivative Works 4.0 International: see [www.creativecommons.org](http://www.creativecommons.org). The text may be reproduced for non-commercial purposes, provided that credit is given to the original author.

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Francesco Rossi  
April 2018

# Summary

Nowadays, the relationship among food, health, and economy is an emerging topic that engages the modern global society from an interdisciplinary perspective. The Food Risk assessment has been formalized and incorporated into the specific discipline addressed to everyone involved with food from production to consumption, including growers, processors, regulators, distributors, retailers and consumers. However, both intentional and unintentional actions committed for economic gains could make an attempt on people's health.

In recent years, many tools have been developed to help the authorities involved in controls and consumers too. The integration of multidisciplinary techniques has favorably supported the study and the development of tools related to the field of the Systems Biology as well as the application of state-of-the-art techniques deriving from other application fields such as the Computer Science.

To counteract and operate with reaction and prevention in my Ph.D. I investigate the use of original Computer-Aided technologies in two particular instances. The first one refers to a Food Traceability issue related to dairy product control. I studied and implemented a heuristic procedure that allows food inspectors to highlight possible adulterations in cheese production into the small farm environment. The procedure is mainly based on Short Tandem Repeat investigation to compare the DNA fingerprint among cows, milk, and cheese. The second one regards the Food Fraud discipline. I developed a mobile application to counteract the problem of fish species substitution and mislabelling. The infrastructure implemented is composed of a cloud remote server where both image analysis and machine learning algorithm take part. The main breakthrough on this topic has been reached with a deep learning classification system which allowed to obtain an improvement in the global accuracy to correctly identify the fish species. Eventually, in the last topic, I deal with the problem of fish fillets identification. The main outcome of this preliminary study is the application of a portable Near Infra-Res molecular sensor that was specifically trained to discriminate the fish fillets available in a sample database.



# Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Alfredo Benso for the continuous support of my Ph.D. research, for his patience, motivation, and for allowing me to grow as a research scientist.

Besides my advisor, I would like to thank Prof. Stefano Di Carlo and all the SysBio members. I appreciate all their contributions of time, ideas, and knowledge that made my Ph.D. experience stimulating.

I would especially like to thank Pier Luigi Acutis and Paola Modesto of IZSTO for always being available and confident in my work.

My sincere thanks also go to all the coworkers and friends of Lab-6 at Politecnico di Torino for the time we spent together and for the mutual support.

Last but not the least, I would like to acknowledge my family, all my friends, and all the people that endorsed me in these years.



*Dedicated to Lara*

# Contents

<b>List of Tables</b>	<b>x</b>
<b>List of Figures</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Scientific Context . . . . .	1
1.2 Quality, Safety, Fraud, Defense in Food . . . . .	2
1.3 <i>Computer-Aided</i> Technology . . . . .	3
1.4 Structure of the thesis . . . . .	4
1.4.1 Dairy Farming Product . . . . .	4
1.4.2 Fish Species Identification . . . . .	5
1.4.3 Molecular Sensor . . . . .	6
1.5 Notes and Further Considerations . . . . .	6
<b>2 Heuristic Molecular Dairy Farming Analysis</b>	<b>7</b>
2.1 Introduction . . . . .	7
2.2 Domain and Limits . . . . .	7
2.3 The “From Farm to Fork” project . . . . .	8
2.4 STR and RFU . . . . .	9
2.5 DNA Pool Analysis . . . . .	11
2.5.1 CMA-ES algorithm . . . . .	12
2.6 Computer-assisted traceability pipeline . . . . .	12
2.7 Experimental Setup . . . . .	16
2.8 Results . . . . .	17
2.9 Conclusion . . . . .	20
<b>3 Vision and Learning in Fish Species Identification</b>	<b>21</b>
3.1 Introduction . . . . .	21
3.2 State of the Art . . . . .	21
3.2.1 Filling the gap . . . . .	23
3.3 The “FishApp” project . . . . .	23
3.3.1 Infrastructure Design . . . . .	23

3.3.2	Mobile App and Remote Server Interaction . . . . .	24
3.3.3	Image Processing Unit . . . . .	26
3.3.4	Features Extraction . . . . .	29
3.3.5	ANN Classifier . . . . .	30
3.3.6	Dataset . . . . .	32
3.3.7	Results . . . . .	33
3.3.8	Research outcome on national media . . . . .	34
3.3.9	Limitations and Constraint . . . . .	35
3.4	Deep Learning in Image Recognition . . . . .	35
3.4.1	Transfer Learning . . . . .	37
3.5	The “F.I.S.HUB” project . . . . .	38
3.5.1	Software architecture . . . . .	39
3.5.2	Picture Cloud Database . . . . .	40
3.5.3	Server Classifier . . . . .	44
3.5.4	Method and Classifiers . . . . .	46
3.5.5	Deep Model Zoo . . . . .	48
3.5.6	Results . . . . .	50
3.6	Standard Classification v.s. One-Shot . . . . .	52
3.6.1	F.I.S.HUB Embedding setup . . . . .	54
3.6.2	Results . . . . .	55
3.7	Endless Learning . . . . .	59
3.8	The “F.I.S.HUB” Mobile Application . . . . .	60
<b>4</b>	<b>From Image to Molecular Analysis</b>	<b>67</b>
4.1	Introduction . . . . .	67
4.2	Method . . . . .	68
4.2.1	Method 1 - Fillet Picture Analysis . . . . .	68
4.2.2	Method 2 - Fillet Microscope Picture . . . . .	69
4.2.3	Method 3 - Staining . . . . .	70
4.2.4	Method 4 - NIR sensor . . . . .	71
4.3	Results . . . . .	72
<b>5</b>	<b>Conclusion</b>	<b>75</b>
	<b>Bibliography</b>	<b>77</b>

# List of Tables

1.1	Food Risk Matrix . . . . .	2
2.1	Example of a data farm organization. Here the $a^{(i,j)x}, a^{(i,j)y}$ notation represents the two alleles for each cow in each STR. . . . .	10
2.2	BM and CH data organization. Here $a_P^j$ represents the pool P allele vector for each STR. . . . .	11
2.3	The height of the RFU allele's peak (h instead of a) in each STR . . . . .	11
2.4	Normalized cows and pool (BM and CH) STR-RFU peak tabular data . . . . .	13
2.5	Summary of the STR dataset used in the analysis . . . . .	17
3.1	F.I.S.HUB fish species list . . . . .	41
3.2	F.I.S.HUB list and number of images for each fish species . . . . .	44
3.3	Images for each Order in the DB . . . . .	45
3.4	Training results for AlexNet and GoogleLeNet models . . . . .	50
3.5	FISHUB accuracy trained on GoogleLeNet model pretrained on ImageNet dataset until the second last inception node . . . . .	52
3.6	Table 3.5 grouped per fish Order . . . . .	52
3.7	F.I.S.HUB Siamese Network species accuracy . . . . .	58
3.8	Table 3.7 grouped per fish Order . . . . .	58
4.1	Merlangius merlangus and Melanogrammus aeglefinus accuracy of picture analysis . . . . .	68
4.2	Solea solea, Pleuronectes platessa, and Pangasianodon hypophthalmus accuracy of picture analysis . . . . .	69
4.3	Microscope analysis accuracy . . . . .	70
4.4	Staining analysis accuracy . . . . .	71
4.5	Results summary for every method . . . . .	73

# List of Figures

2.1	Dairy Product Traceability scheme . . . . .	8
2.2	Data extraction scheme . . . . .	10
2.3	Global scheme of the Forgery Detection Model . . . . .	14
2.4	W boundary condition during CMA-ES routine . . . . .	17
2.5	Results of the mean score values for Farm A (left side) and Farm B (right side) for the BULK MILK analysis for each available month. Black lines are related to 100% true cows setup analysis, the blue ones are related to 50% of adulterated milk origin, and the red ones are 100% forged milk origins. . . . .	18
2.6	Results of the mean score values for the Farm A (left side) and the Farm B (right side) for the CHEESE analysis for each available month. Black lines are related to 100% true cows setup analysis, the blue ones are related to 50% of adulterated cows and the red ones are 100% forged cows. . . . .	18
2.7	Box plots of grouped scores for the Farm A and B in the bulk milk and cheese analysis. Black box are related to 100% true cows setup analysis, the blue ones are related to 50% of adulterated cows and the red ones are 100% forged cows . . . . .	19
2.8	Global simulation scores. Both farms and dairy products are grouped. The black box is related to 100% true cows setup analysis, the blue one is related to 50% of adulterated cows and the red one is 100% forged cows. The “*” indicate significant difference between groups ( $p < 0.05$ , K-S test) . . . . .	20
3.1	FishAPP software architecture . . . . .	25
3.2	Example of fish picture taken from smartphone local storage library & FishAPP GUI . . . . .	26
3.3	Anatomical set points detected by FishAPP image processing unit . . . . .	27
3.4	Summarized FishApp image processing unit . . . . .	28
3.5	Fish segmentation result (i.e. binary mask) . . . . .	29
3.6	FishAPP feature extracted. Blue and pink lines represent the morphological dimensions of the fish. . . . .	30
3.7	Classification Pipeline . . . . .	32
3.8	Presentation image of FishApp . . . . .	34
3.9	Television demonstration screenshot . . . . .	34

3.10	Newscast demonstration screenshot . . . . .	34
3.11	Relationship between the brain visual process and modern neural network	36
3.12	Three ways in which Transfer Learning can contribute . . . . .	38
3.13	FoodIntegrity logo . . . . .	39
3.14	F.I.S.HUB architecture . . . . .	40
3.15	F.I.S.HUB picture guideline . . . . .	42
3.16	F.I.S.HUB mass storage system . . . . .	43
3.17	DB hierarchical organization . . . . .	43
3.18	F.I.S.HUB global scheme . . . . .	45
3.19	Feature engineering and classification . . . . .	46
3.20	CNN feature extraction and classification . . . . .	47
3.21	AlexNet architecture . . . . .	49
3.22	GoogleLeNet architecture . . . . .	49
3.23	Barplot of training results for AlexNet and GoogleLeNet models . . . . .	50
3.24	The game mode of Pinko . . . . .	53
3.25	F.I.S.HUB Siamese Network architecture . . . . .	55
3.26	t-SNE representation of SN training result over F.I.S.HUB specie . . . . .	56
3.27	The same as in Figure 3.26 with the highlight of flat/non-flat fish graphic separation . . . . .	56
3.28	t-SNE representation of SN training result over F.I.S.HUB Order . . . . .	57
3.29	Distance-based results for same species . . . . .	59
3.30	Distance-based results for different species . . . . .	60
3.31	App storyline and functionalities . . . . .	61
3.32	App initial screen . . . . .	62
3.33	Home screen after selecting a photo . . . . .	63
3.34	Species selection . . . . .	64
3.35	Results page . . . . .	64
4.1	From fish to fillet morphological information . . . . .	67
4.2	Merlangius merlangus and Melanogrammus aeglefinus fillets . . . . .	68
4.3	Camera Phone Lenses 200X Lens Microscope LED . . . . .	69
4.4	Solea solea, Pleuronectes platessa, and Pangasianodon hypophthalmus examples . . . . .	70
4.5	Fillets images with Toluidine blue stain preparation . . . . .	70
4.6	SCiO sensor . . . . .	71
4.7	SCiO Lab View screenshot . . . . .	72
4.8	SCiO accuracy calculated on the validation set . . . . .	73

# Chapter 1

## Introduction

### 1.1 Scientific Context

One of the main topics of interest in today's society is the food-health-economy relationship. In particular, in the global market, this issue has become more and more important since in the path that connects together production, processing, selling and consumption of food many other roads are intertwined. Nowadays it is possible to address these issues both from an economic and ethical point of view, particularly involving the nutrition and health of people. The safeguarding and improvement of the attention in these fields have positive effects summarized in these few words: to protect the rights, and fulfill the duties, both for consumers and producers.

Rightly, people claim to buy and/or consume healthy and certified food where information about the origin and traceability constitute the main guarantees. Although a lot of progress has been made in recent years, there are still many things to improve. However, the shortcomings can be balanced by a good level of knowledge and awareness of the consumer. In recent years, especially from a communicative point of view, many energies have been invested so that the consumer is not only protected but it is also responsible.

On the other hand, modern food supply chains and manufacturing infrastructure have vastly expanded its scale and potential impact but must guarantee and comply with the health and hygiene regulations in force in their context. The standards are often disregarded, excluding the intentional cases, mainly for economic interests and also for legislative failings or inadequate controls.

Many public and private institutions, both national and international, are playing in synergy an important role to provide products and services to defend the citizen's health through "food protection" and the health of the animals that produce them.

In this scenario, how is it possible to counter the threat to public health due to *food risk*? Moreover, what role could *technological innovation* play in this area?

## 1.2 Quality, Safety, Fraud, Defense in Food

Whether or not food counterfeiting is an intentional or unintentional act, in both cases it may have two different motivations. The first one is an economic gain, the other is a harm to health [63]. This concept is globally represented in the “Food Risk Matrix” as described in Table 1.1.

Table 1.1: Food Risk Matrix

Food Quality	Food Fraud	<i>Economical Gain</i>	<b>Motivation</b>
Food Safety	Food Defense	<i>Health Harm</i>	
<i>Unintentional</i>	<i>Intentional</i>		
<b>Action</b>			

The risk establishment condition can therefore be declined based on the factors that characterize it. From the previous table it is possible to consider these four major disciplines:

- **Food Quality** for an economic threat that is unintentional;
- **Food Safety** for an accidental action with health consequences;
- **Food Fraud** for an intentional economic imposture ;
- **Food Defense** for a voluntary action to harm people’s health (e.g. terrorism).

For each of these disciplines, it is possible to determine the two main actions, which are reaction and prevention. Technological development and increasing legal protection have made prevention play a fundamental role. Despite the common awareness that is “*better safe than sorry*” it is necessary to consider that prevention, made by controllers, is only possible if the problem has been appropriately defined before and only if the nature of the *risk* has been understood (i.e. estimate the effectiveness of response actions).

The Food Risk Matrix does not take into account other so-called disciplines such as *Food Traceability* and *Food Integrity*. What do they mean exactly? Food Traceability investigates the ability to track any food, feed, food-producing animal or substance that will be used for consumption, through all stages of production, processing and distribution [15]. Food Integrity implies an unrestricted viewpoint that includes food production, distribution, and everything in between (procurement, processing, packaging, testing, etc.). Thus, these definitions are de-facto involved in all the actions and motivations described before.

In recent years, tools have also been developed to help both the authorities involved in controls and consumers. Even if the consumer is at the last stage of the chain it is,

however, important to provide him with effective tools for self-protection. The controllers, otherwise, can act at any stage of production and distribution.

Depending on the context, the four aforementioned disciplines (i.e. Food Quality, Safety, Fraud and Defense) can be applied thanks to chemical-biological techniques and together with a technical-engineering awareness. In fact, the integration of multidisciplinary techniques has favorably supported the study and the development of tools related to the field of the Systems Biology as well as the application of state-of-the-art techniques deriving from other application fields such as the Computer Science.

In this thesis, I here document some solutions for both controllers and consumers thanks to innovative *Computer-Aided* techniques.

### 1.3 *Computer-Aided Technology*

“Computer-Aided” generally means that the use of a computer is as an essential tool. During last decades the qualitative and quantitative improvement of the processors’ performances concurrently come up beside the constant growing algorithm innovation in the field of Data Science and Machine Learning (usually reduced with the general “Artificial Intelligence” expression). Furthermore, in this age, it is possible to generate and manipulate considerable quantities of information, such as numerical data and images, that can, in turn, be used to obtain a better understanding of several previously intractable problems.

In this thesis, I address several critical issues in the Computer-Aided world, from data collection, storage, organization and integration to computational performances and translational research. Many causes of these problems may be identified, but probably the most critical one is the methodological and “linguistic” distance between the Computer Science and the Life Sciences (i.e. *Bio-*) worlds. From the perspective of this work, this problem results in several key issues that make the development, deployment, use and practical translation of applications very slow, difficult, and often prone to errors.

From an engineering point of view the most obvious issue is the lack of a structured and formalized software development process, as well as integrated, validated, and certified programming-level supports (in terms of libraries or data management methodologies). This results in a wide availability of (bio)informatics tools, very often designed without taking into account the interoperability and/or re-usability of their building blocks in different pipelines. Moreover, such a limitation is more evident when a new methodology has to be implemented for an ad-hoc problem that has never been studied before. A lot of development time is therefore spent in integrating known algorithms into novel procedures, parsing and translating data from one format to another, and making different programming languages, libraries and file formats work together. An additional but not less critical problem comes from the need to aggregate non-standardized and often completely custom datasets.

From a Life Sciences end-user point of view, the generalized cultural low “trust” in the bioinformatics solutions is aggravated by the tools’ user-unfriendliness, the limited measurability of the quality level of the applications and their output results (that cannot usually be benchmarked against each other), and the often not straightforward interpretation of the results by researchers who do not have a computer science background. This has also the secondary effect of trusting only software developed in-house, thus limiting the dissemination and sharing of potentially very valuable algorithms and results.

The “Computer-Aided” technology I implemented during my Ph.D. research is described in the next section.

## 1.4 Structure of the thesis

During the three years of my Ph.D at Politecnico di Torino in the Department of Control and Computer Engineering as a member in the Systems Biology Research Group (SysBio) <sup>1</sup> I have been involved in different projects related to “food”. These works have been possible thanks to a cooperative collaboration mainly with the “Istituto Zooprofilattico Sperimentale del Piemonte, Liguria e Valle d’Aosta” (IZSTO) <sup>2</sup>.

My background and foregoing work experience sparked in me a steady and growing interest in the world of research. The opportunity to deal with interdisciplinary projects with professionals from other fields has been an awe-inspiring motivation. I put into practice all the knowledge acquired during these years and I also appended more and more expertise in the computer science field and in the biological area too.

The projects are briefly illustrated in the following subsections and are exhaustively described in the next chapters of this document.

### 1.4.1 Dairy Farming Product

The first part of the thesis is focused on a research funded by the Italian Ministry of Health entitled “*From Farm to Fork*” acting as a partner of the IZSTO Genetics and Immunobiochemistry Team.

Among dairy products, traceability of traditional cheeses produced on small farms is a challenging issue. It seems feasible that, in order to increase the product supply and to transform surplus milk, an amount of sold cheeses does not originate from the accredited farms. Since nothing is known about the hygienic conditions of the not certified farms, consequences on consumer health could occur.

---

<sup>1</sup>Systems Biology Research Group (SysBio) at the Department of Control and Computer Engineering of Politecnico di Torino [link](#)

<sup>2</sup>Istituto Zooprofilattico Sperimentale del Piemonte, Liguria e Valle d’Aosta (IZSTO) [link](#)

The aim of this study was to set up a genetic tool for origin traceability of traditional dairy products. We investigated the use of Short Tandem Repeats (STRs) to create a DNA fingerprint of small dairy farms and to assign dairy products (milk and cheese) to the corresponding producer.

The implemented algorithm is able to produce, as output, a final score useful for assigning (as producer) a group of animals to a specific dairy product or to highlight a gradual significance of adulteration. The core of the procedure is based on a heuristic analysis over the STRs between cows and dairy products.

In conclusion, the project produced new knowledge on the efficacy and applicability of the STRs to analyze complex matrices like milk and cheese. Moreover, from a biological point of view, a panel of markers used to trace dairy products has been tested successfully on small dairy farms.

This work is fully described in Chapter 2 on page 7.

## 1.4.2 Fish Species Identification

The subsequent part of the thesis is relative to the work focused on the *fish species identification* task. This study lasted throughout all the three Ph.D. years in collaboration with IZSTO Genetics and Immunobiochemistry Team.

In the beginning, the activity was planned as an exploratory work aimed to demonstrate the feasibility of a prototype-system able to counteract the problem of fish species substitution in a fish market real setup. We chose to develop a smartphone mobile application targeted for consumers able to correctly identify the species of a fish only between two very similar common species: anchovy and sardine (i.e. *Engraulis encrasicolus* and *Sardina pilchardus*).

The mobile application was essentially a compound of a Computer Vision procedures (needed to extract the features of the fish) and a Machine Learning classifier (Artificial Neural Network) able to discriminate one species from the other.

Since this prototype demonstrated to be effective, and also thanks to an encouraging mass media feedback, this work then had the strength to find a continuity within the network of the *Food Integrity Project* <sup>3</sup> funded by the European Union in the Work Package #15 entitled “*Fish Identification Software Hub*” (F.I.S.HUB) <sup>4</sup>.

In partnership with IZSTO and the University of Salford (Manchester, UK) <sup>5</sup> we then extended the previous research with the main goal to create an objective fraud detection software usable in the field by trained personnel as well as un-experienced end users.

The F.I.S.HUB software is based on cutting-edge Image Analysis and Machine Learning technologies (Deep Learning) and it is able to identify the species of a fish from its

---

<sup>3</sup>Food Integrity EU Project [link](#)

<sup>4</sup>Fish Identification Software Hub (F.I.S.HUB) [link](#)

<sup>5</sup>University of Salford (Manchester, UK) [link](#)

photo with relevant accuracy. More than twenty fish species have been considered in this study. The F.I.S.HUB Database consists of thousands of pictures collected under definite guidelines in fish markets. In addition, the newly implemented method is also capable to estimate the affinity of an unknown fish species with respect to the knowledge domain of the fish pictures available in the database. To enable the widest possible use of the software, its access will be open and available through a mobile application designed for the most common mobile platforms.

This work is fully described in Chapter 3 on page 21.

### 1.4.3 Molecular Sensor

This last contribute is a pilot study (Work Package #21 of the Food Integrity Project funded by the European Union) to explore the possibility of applying Near-infrared spectroscopy (NIR) in order to identify the species in fish fillets, in partnership with IZSTO and the University of Salford.

To measure the NIR spectrum and to perform a classification model of the fish fillets we adopted the SCiO™ molecular sensor, developed and distributed by Consumer Physics <sup>6</sup>.

We considered in this preliminary research three different species: *Solea solea*, *Pleuronectes platessa*, and *Pangasianodon hypophthalmus*. The acquired scans were realized on different points of the fillet in order to evaluate possible variation with respect to the portion of the fillet side. The obtained very high global accuracy suggests the SCiO™ as a promising tool to use in the context of on-site controls.

This work is fully described in Chapter 4 on page 67.

## 1.5 Notes and Further Considerations

Since my Ph.D. has been accomplished on an interdisciplinary field, for this reason, I deliberately decided to describe all the different thesis parts with a detail level that could allow a full comprehension by all the scientists that may have a different background not strictly related to computer engineering. On the other hand, for all those who came from the widespread “food” field, I warrant that I kept the minimum detail level that allowed me to correctly discuss the topics of this thesis.

---

<sup>6</sup>SCiO™ - by Consumer Physics [link](#)

## Chapter 2

# Heuristic Molecular Dairy Farming Analysis

### 2.1 Introduction

Food integrity and food safety have received much attention in latest period due to the dramatically increasing number of food frauds. Traceability is a useful practice to assure foodstuff safety and quality, to guarantee hygiene standards, and to preserve consumers health and choices. Over the past years, DNA analysis has been extensively recognized as a capable tool to deal with genetic traceability issues, obtaining an important role in tracing and testing the food origin and the food safety. The main reason is that DNA analysis are objective, irrefutable proofs and can be applied to many food matrices, even though processed.

In this chapter, I investigate dairy products for which one of the high-priority issues is traditional cheese traceability. In the case of frauds, it may occur that a selected dairy product that should be produced by milk coming from a certified farm, is instead produced using a variable amount of milk coming from uncertified farms. The nature of this problem is illustrated in Figure 2.1. Green lines represent the correct traceability between the product originate from the same farm (i.e. cows). On the contrary, the red line indicates an example of nonconformity in the supply chain.

### 2.2 Domain and Limits

DNA analysis for dairy product traceability implicates many technical challenges. The cheese (CH) is obviously produced from the bulk milk (BM) of a certain farm. Therefore the BM contains DNA from different cows of the farm and during the ripening process is subject to different biochemical changes.

To combine the correspondence of a dairy product (BM and CH) with the small farm producer, we investigate the use of Short Tandem Repeats (STRs) analysis to create a

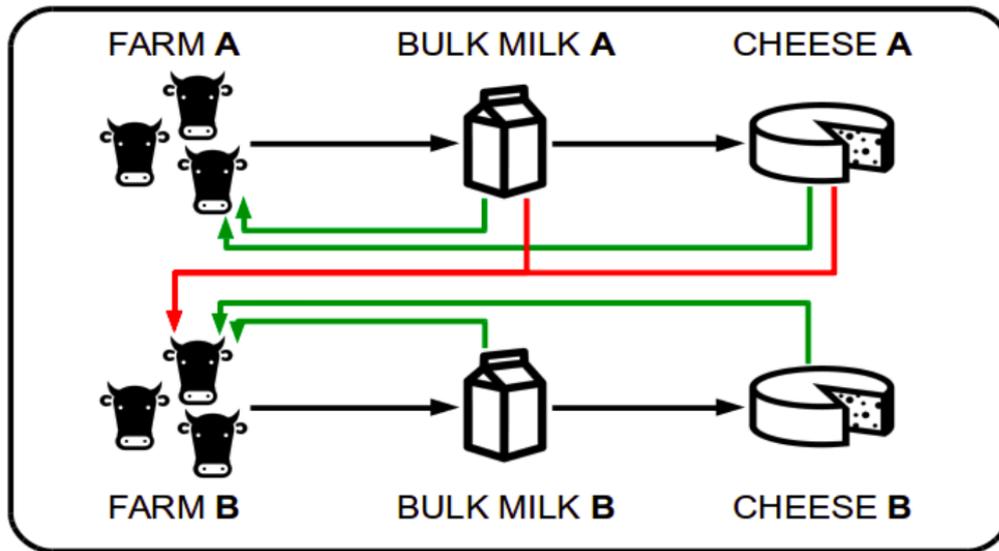


Figure 2.1: Dairy Product Traceability scheme

unique DNA fingerprint. Up to now, the STRs analysis has been applied to blood samples for genetic population analysis [43] [60] [62] [38] [71] and also to milk samples only to identify the Quantitative Trait Locus (QTL) associated with the pattern in animal science [39] [4]. On the other hand, the application of STR analysis applied for dairy product origin traceability is a different task with respect to cited research, and display much more complex issue too.

Indeed, dairy products contain the DNA belonging to several dissimilar individuals, and so it is not possible to perform single-animal traceability. In literature, dairy product traceability has been mainly addressed by studying Fatty Acids and Triacylglycerols Content using Gas Chromatography [49]. Moreover the STR marker analysis proved to be valid only in mono-breed setup [58].

As far as we know this research is the first attempt to explore the use of pooled STR analysis for food product traceability.

### 2.3 The “From Farm to Fork” project

In the next sections, I describe the computer-assisted molecular traceability system in order to examine the pooled STRs data of dairy products to certify the origin of the products within the correct farms both from a qualitatively and quantitatively point of view.

In this project were included two farms having different cow breeds. First, the DNA of each animal was extracted to compute a DNA signature based on the analysis of known STRs loci. Then the same STR analysis was performed on their dairy products.

The obtained STRs genetic datasets were globally analyzed by using a Covariance Matrix Adaptation Evolution Strategy (CMA-ES) algorithm in order to evaluate the traceability between the dairy products and the corresponding cows that took part in the farm production. Eventually, the implemented algorithm is able to highlight possible adulterations and/or inconsistencies.

Results obtained from this study showed that the dairy product presents an STRs profile that is composed of a subgroup of the STRs identified in the initial animals involved in the production. In conclusion, the profile could be efficiently used to trace the origin of the dairy product.

This study has been possible thanks to the collaboration with the Genetics and Immunobiochemistry Team of the IZSTO <sup>1</sup> and was supported by the Italian Ministry of Health grant IZS PLV 01/14 RC.

These results have been published in [52] and [53].

## 2.4 STR and RFU

The dataset derived from two Italian farms with different geographic locations from Biella and Imperia, Italy. The breed cows taken into account for the tuning of the method are respectively *Pezzata Rossa d’Oropa* and *Bruna Alpina*. At the beginning of the study, veterinarians collected blood and milk samples from each cow. Afterward, they monthly sampled BM and CH for 12 months in the first farm and 11 months in the second one. Samples were then cold-stored for the tuning of the analysis protocol and the choice of the best genotyping process. In Figure 2.2 are illustrated the principal steps of the STRs selection and the row data generation as resumed in the following list:

- Sample Collection: in the course of collecting the DNA is extracted from blood, milk somatic cells and cheese;
- STRs selection: from a panel of 280 available STRs (from literature), 20 STRs were selected taking into account some of their characteristics, as well as other technical parameters conditioned by the tuning phase of the protocol analysis (the STR selection process is proprietary by IZSTO and, for the moment, it cannot be fully disclosed);
- Genotyping Process: capillary electrophoresis using a 3130 Genetic Analyzer (Applied Biosystems) and fragments sizing using the STRAnd software [67];
- Data extraction: the allele’s peak height in Relative Fluorescence Unit (RFU) of the electropherogram track was considered as an estimation of its quantity. This will be used in the following analyses.

---

<sup>1</sup>IZSTO Genetics and Immunobiochemistry Team [link](#)

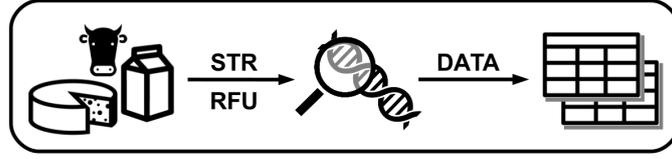


Figure 2.2: Data extraction scheme

After the genotyping process, the obtained raw data were organized in a tabular format as shown in Table 2.1 where the allele frequencies for each STR and for each cow have been organized. To better understand its content consider the following notation:

- $n$  is the number of processed STRs;
- $m$  is the number of cows available within the examined farm;
- $a^{(i,j)}$  ( $i \in [1, m], j \in [1, n]$ ) is the specific allele's dimension (bp) of the  $i^{\text{th}}$  cow for the  $j^{\text{th}}$  STR. This notation includes the indication of the polymorphism occurrence of being heterozygote ( $a^{(i,j)x} \neq a^{(i,j)y}$ ) or homozygote ( $a^{(i,j)x} = a^{(i,j)y}$ ).

Table 2.1: Example of a data farm organization. Here the  $a^{(i,j)x}, a^{(i,j)y}$  notation represents the two alleles for each cow in each STR.

Cows	STR1	STR2	STR3	...	STR $n$
COW1	$a^{(1,1)x}, a^{(1,1)y}$	$a^{(1,2)x}, a^{(1,2)y}$	$a^{(1,3)x}, a^{(1,3)y}$	...	$a^{(1,n)x}, a^{(1,n)y}$
COW2	$a^{(2,1)x}, a^{(2,1)y}$	$a^{(2,2)x}, a^{(2,2)y}$	$a^{(2,3)x}, a^{(2,3)y}$	...	$a^{(2,n)x}, a^{(2,n)y}$
COW3	$a^{(3,1)x}, a^{(3,1)y}$	$a^{(3,2)x}, a^{(3,2)y}$	$a^{(3,3)x}, a^{(3,3)y}$	...	$a^{(3,n)x}, a^{(3,n)y}$
...	...	...	...	...	...
COW $m$	$a^{(m,1)x}, a^{(m,1)y}$	$a^{(m,2)x}, a^{(m,2)y}$	$a^{(m,3)x}, a^{(m,3)y}$	...	$a^{(m,n)x}, a^{(m,n)y}$

Furthermore, in Table 2.2, the BM and the CH genotyping pool analysis row data were arranged. Nevertheless, in a different manner from Table 1, the information associated to each cell  $aP_j$  ( $P \in \{BM, CH\}, j \in [1, n]$ ) of the table is a vector including all the allele values obtained from the genotyping process of the pool  $P$  for the  $j^{\text{th}}$  STR.

Eventually, following the format of the previous tables it is possible to configure the absolute RFU alleles peak (h) of each allele for each cow of the farm. The same has been done also for BM and CH. An example of this representation is shown in Table 2.3. At the end, all data were stored in comma-separated values format text files (\*.CSV).

Table 2.2: BM and CH data organization. Here  $\underline{a}_P^j$  represents the pool P allele vector for each STR.

Pool	STR1	STR2	STR3	...	STR $n$
BM	$\underline{a}_{BM}^1$	$\underline{a}_{BM}^2$	$\underline{a}_{BM}^3$	...	$\underline{a}_{BM}^n$
CH	$\underline{a}_{CH}^1$	$\underline{a}_{CH}^2$	$\underline{a}_{CH}^3$	...	$\underline{a}_{CH}^n$

Table 2.3: The height of the RFU allele's peak (h instead of a) in each STR

RFU	STR1	STR2	STR3	...	STR $n$
COW1_h	$h^{(1,1)x}, h^{(1,1)y}$	$h^{(1,2)x}, h^{(1,2)y}$	$h^{(1,3)x}, h^{(1,3)y}$	...	$h^{(1,n)x}, h^{(1,n)y}$
COW2_h	$h^{(2,1)x}, h^{(2,1)y}$	$h^{(2,2)x}, h^{(2,2)y}$	$h^{(2,3)x}, h^{(2,3)y}$	...	$h^{(2,n)x}, h^{(2,n)y}$
COW3_h	$h^{(3,1)x}, h^{(3,1)y}$	$h^{(3,2)x}, h^{(3,2)y}$	$h^{(3,3)x}, h^{(3,3)y}$	...	$h^{(3,n)x}, h^{(3,n)y}$
...	...	...	...	...	...
COW $m$ _h	$h^{(m,1)x}, h^{(m,1)y}$	$h^{(m,2)x}, h^{(m,2)y}$	$h^{(m,3)x}, h^{(m,3)y}$	...	$h^{(m,n)x}, h^{(m,n)y}$
BM_h	$\underline{h}_{BM}^1$	$\underline{h}_{BM}^2$	$\underline{h}_{BM}^3$	...	$\underline{h}_{BM}^n$
CH_h	$\underline{h}_{CH}^1$	$\underline{h}_{CH}^2$	$\underline{h}_{CH}^3$	...	$\underline{h}_{CH}^n$

## 2.5 DNA Pool Analysis

The first experiments we performed attempted to evaluate the capacity to trace dairy products using standard software algorithms regularly used in genetic distance analysis, such as FSTAT [22], PHYLIP [17] and SMOGD [16] and then resorting to STRUCTURE [30]. However, results showed that these algorithms were not well suited to accomplish the intended purpose of this work. The main reason is that they usually apply a Bayesian algorithm approach to assign a sample genotype to a specific dataset representing the candidate group of origin. While this method works well in diploid data (i.e. only two alleles), they did not perform properly in the experimental setup considered in this work for the reason that we have a mutable number of alleles for each STR in every sample (i.e. pooled DNA samples of BM and CH).

Hence, I designed a new approach able to overstep this limitation. The innovative method here described is at first glance an automatic heuristic procedure based on the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) algorithm. The heuristic is employed to estimate the likelihood of a dairy product STRs profile to be originated by a combination of the STR profiles of the cows from which the dairy product was originated from.

In the next subsection are briefly described the general principles of the CMA-ES, which is necessary to better understand the proposed computer-assisted molecular

traceability method described next.

### 2.5.1 CMA-ES algorithm

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is an optimization method first proposed by Hansen, Oster Meier, and Gawelczyk [27] and further developed in subsequent years [26] [25]. The CMA-ES performs an exploration in a solution space exploiting a covariance matrix, closely related to the inverse Hessian on convex-quadratic functions. The approach is particularly suited to solve difficult problems such as non-linear, non-convex and non-separable, of at least moderate dimensionality  $\in [10, 100]$ .

Due to its biological foundations in CMA-ES the iteration steps are called *generations* like in other evolutionary algorithms. The value of a generic algorithm parameter  $y$  during generation  $g$  is denoted with  $y^{(g)}$ . The mean vector  $m^{(g)} \in \mathbb{R}^n$  represents the ideal and most promising solution so far. The step size  $\sigma^{(g)} \in \mathbb{R}^+$  controls the step length, and the covariance matrix  $C^{(g)} \in \mathbb{R}^{n \times n}$  determines the shape of the ellipsoid distribution in the search space. Conversely, its goal is to fit the search distribution to the contour lines of the objective function  $f$  to be minimized:  $C^{(0)} = I$ .

One of the principal feature of the CMA-ES is that it requires almost no parameter tuning for its usage unlike most habitual heuristic optimization methods [31]. In fact, the selection of its internal parameters is not left to the user. Notably, the default population size  $\lambda$  is comparatively small to allow for rapid convergence. Restarts with increasing population size have been demonstrated [3] to be useful to improve the global search performance and is nowadays included as an option in the conventional procedure.

In this research, I used the CMA-ES package [68] developed in R environment [66].

## 2.6 Computer-assisted traceability pipeline

We assume that, if in a farm some defined cows produced the BM or CH, then the BM or CH genetic STRs profile should be a linear combination of the STR profiles of those cows. Under this primary hypothesis, I have implemented the automated forgery detection software. This method is composed of two main steps: data normalization, and heuristic simulation.

The purpose of the first step is to pre-process the RFU raw data (see Table 2.3) of a specific dairy product (CH or BM pool analysis) together with the profiles of the cows belonging to the declared farm. This, in turn, makes them comparable and allows us to perform forgery detection. All RFU peak profiles are therefore normalized between [0,1] producing the normalized dataset reported in Table 2.4 where:

$$H^{(i,j)} = \left[ \frac{h^{(i,j)x}}{\max(h^{(i)x})}, \frac{h^{(i,j)y}}{\max(h^{(i)y)}} \right] \quad (2.1)$$

is the normalized pair values of alleles' RFU peaks for cow  $i$  and STR  $j$ ;

$$\underline{H}_p^{(j)} = \frac{h_p^{(j)}}{\max(h_p)} \quad (2.2)$$

is the normalized vector of alleles' RFU peaks for the pool (P) and STR  $j$ .

Table 2.4: Normalized cows and pool (BM and CH) STR-RFU peak tabular data

Normalized	STR1	STR2	STR3	...	STR $n$
COW1_H	$H^{(1,1)x}, H^{(1,1)y}$	$H^{(1,2)x}, H^{(1,2)y}$	$H^{(1,3)x}, H^{(1,3)y}$	...	$H^{(1,n)x}, H^{(1,n)y}$
COW2_H	$H^{(2,1)x}, H^{(2,1)y}$	$H^{(2,2)x}, H^{(2,2)y}$	$H^{(2,3)x}, H^{(2,3)y}$	...	$H^{(2,n)x}, H^{(2,n)y}$
COW3_H	$H^{(3,1)x}, H^{(3,1)y}$	$H^{(3,2)x}, H^{(3,2)y}$	$H^{(3,3)x}, H^{(3,3)y}$	...	$H^{(3,n)x}, H^{(3,n)y}$
...	...	...	...	...	...
COW $m$ _H	$H^{(m,1)x}, H^{(m,1)y}$	$H^{(m,2)x}, H^{(m,2)y}$	$H^{(m,3)x}, H^{(m,3)y}$	...	$H^{(m,n)x}, H^{(m,n)y}$
BM_H	$\underline{H}_{BM}^1$	$\underline{H}_{BM}^2$	$\underline{H}_{BM}^3$	...	$\underline{H}_{BM}^n$
CH_H	$\underline{H}_{CH}^1$	$\underline{H}_{CH}^2$	$\underline{H}_{CH}^3$	...	$\underline{H}_{CH}^n$

The heuristic starts to work by analyzing the normalized data reported in Table 2.4. Our model assumes that the amount of milk produced by each cow is unknown. Hence the goal of the model is to “find the best cows' weighted combination (W) in such a way that the sum of the weighted cows' STR profiles produces a pattern that is as similar as possible to the profile of the derived dairy product”.

In the end, the traceability model returns the sum of the squared errors (SSE) of the differences between the alleles of the expected BM or CH STR profile and the predicted one by the heuristic. Then the SSE is multiplied by two penalty coefficient that is computed in sync during the simulation and finally, the output score is figured out.

The first penalty (P1) represents the rate of alleles that are included in the STR profile of P but that are not present in any STR cow profile. The second penalty (P2) is the percentage of alleles within the cow profiles, but not detected in P, as a consequence of the genotyping extraction process. In other words, P1 represents the plausible attendance of a forgery, while P2 estimates the loss of the alleles information in the cow's pattern. This last behavior may depend on the ripening process of the sample collection procedure. The summary of the proposed method is shown in Figure 2.3.

The algorithm receives two main inputs:

- COW\_H is the  $m \times n$  matrix containing all normalized data for the cows composing the farm (Table 2.4). This table comprises all data needed to identify the target (i.e. production farm) for the dairy product under investigation;

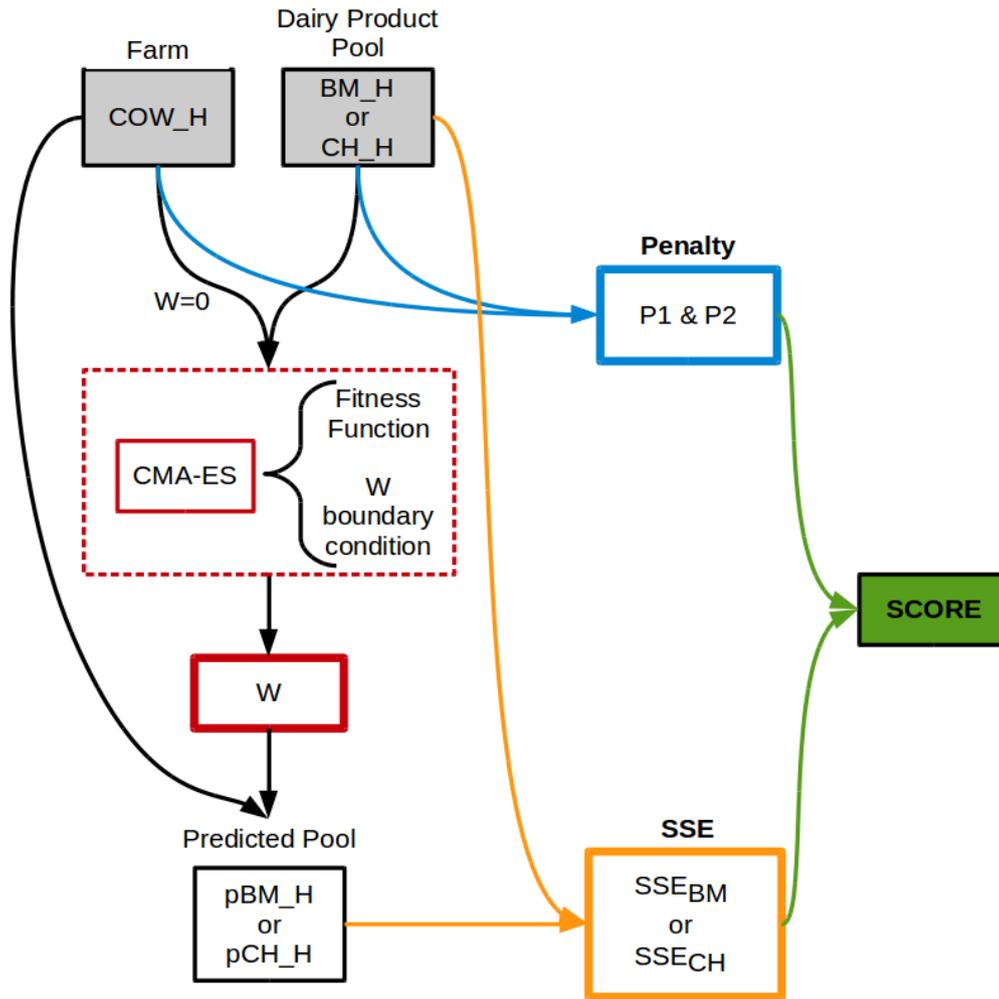


Figure 2.3: Global scheme of the Forgery Detection Model

- BM\_H/CH\_H is a vector reporting the normalized STR-RFU peaks for the dairy product under investigation (BM or CH) following the format reported in Table 2.4.

At the beginning, the algorithm exploits the optimization capability of the CMA-ES to search for the best linear combination of the STR-RFU peaks of the cows composing the farm (COW\_H) able to generate the STR-RFU profile of the dairy product under investigation (BM\_H or CH\_H). This translates into the computation of a vector  $W$  of size  $m$  representing the contribution of each cow to the target dairy product.

Essentially the CMA-ES starts with a standard weight vector initialized to 0 ( $W=0$ ). The CMA-ES then works over several generations until a stop condition is reached:

maximum number of iterations or convergence. The best solution  $W$  determined by the CMA-ES is finally used to calculate the predicted profile for the target dairy product as:

$$pP = W \times COW\_H \quad (2.3)$$

where  $pP \in \{pBM\_H, pCH\_H\}$

The so computed profile ( $pP$ ) and the original  $P$  profile ( $BM\_H$  or  $CH\_H$ ) can then be compared to evaluate the sum of squared error ( $SSE_{BM}$  or  $SSE_{CH}$ ) between them. This error, in common with the two penalty scores  $P1$  and  $P2$ , can then be used to compute the final forgery score of the dairy product as:

$$\boxed{SCORE = SSE_p \cdot P1 \cdot P2} \quad (2.4)$$

$$SSE_p \in \{SSE_{BM}, SSE_{CH}\}$$

$$SSE_{BM} = SSE(BM\_H, pBM\_H) \quad \text{and} \quad SSE_{CH} = SSE(CH\_H, pCH\_H)$$

In fraud condition it may happen that an allele appeared in a specific STR of  $BM$  or  $CH$  does not take part in any STR allele of the cows. In this case, the RFU peak of that allele is taken into account in the SSE computation against a default value equal to 0.

On the other hand, if the occurrence of a certain allele in an STR of a cow does not appear in the  $P$  STR profile, then the routine automatically inserts a default value equal to 0 for that allele in the pool's STR vector. This coincidence is caused by when in the genotyping process, or due to the ripening of the cheese, some alleles are lost or not adequately amplified.

The final score is expected to return a value as close as possible to 0 in the case of appropriate matching between the dairy products and the cows of a farm. On the other hand, in case of frauds, it is expected that the automatic forgery detection returns a higher score value motivated by the inconsistency.

In order to perform its optimization, the CMA-ES requires the definition of a fitness function. The goal is to minimize the SSE between the  $BM$  or  $CH$  genetic profile and the corresponding predicted one, that is computed as a linear combination of the cows' profiles. For this reason, SSE can be exploited as an efficient fitness function for this purpose. The temporary weight vector that is generated iteratively during the generation ( $g$ ) is multiplied by the cows' profile to predict the temporary pool's profile. The SSE fitness function value is computed as follows:

$$\boxed{Fitness = SSE_P^{(g)}} \quad (2.5)$$

$$SSE_P^{(g)} \in \{SSE_{BM}^{(g)}, SSE_{CH}^{(g)}\}$$

$$SSE_{BM}^{(g)} = SSE(BM\_H, pBM\_H^{(g)})$$

$$SSE_{CH}^{(g)} = SSE(CH\_H, pCH\_H^{(g)})$$

$$pBM\_H^{(g)} \text{ or } pCH\_H^{(g)} = W^{(g)} \times COW\_H$$

$W^{(g)}$  is the temporary weight vector at generation  $g$

One more significant feature implemented in the routine analysis regards the value of the weight vector  $W$ . Since during the lactation period each cow has contributed with an unknown amount of milk  $w$  (that is what the model tries to evaluate), we assume that every  $w$  contribution cannot overpass a predefined range that is:

$$lower\_boundary < w < upper\_boundary \quad (2.6)$$

$$lower\_boundary = \frac{0.5}{m} \quad \text{and} \quad upper\_boundary = \max\left(\frac{3}{m}, 1\right)$$

This weight boundary condition is shown in Figure 2.4. A cow cannot over or under produce a certain milk rate in coupling with the number of the other milking cows ( $m$ ) in the farm. These constraints were chosen after analyzing several bulk milk batches and also after consultation with the farm and veterinary staff. The range has been defined supposing that each cow cannot produce more than a half and less of the triple of the mean quantity of dairy product (i.e.  $1/m$ ). Obviously, the upper boundary cannot exceed the value 1 since a cow has no choice to produce all the BM or CH only by itself.

Anyway, these constraint values can be freely modified since they could be used to further refine the simulations in case of explicit information from producers.

## 2.7 Experimental Setup

To demonstrate the usability of this study I designed three experiments. The first one consists in analyzing the dairy product produced with 100% by milk originated from the same farm (i.e., COW\_H, BM\_H or CH\_H taken from the same farm). In the second one, I analyzed a partial forgery in which a dairy product is produced from 50% randomly selected cows from a farm and 50% randomly selected cows from the other farm. Finally, in the last experiment, I analyzed a full falsification setup in which I compared the dairy product from a farm against the STR profile of the cows of the second farm.

To test the robustness of the procedure, for each farm and for each of the three forgery levels, every dairy product has been analyzed 24 times also to highlight possible diversification in results.

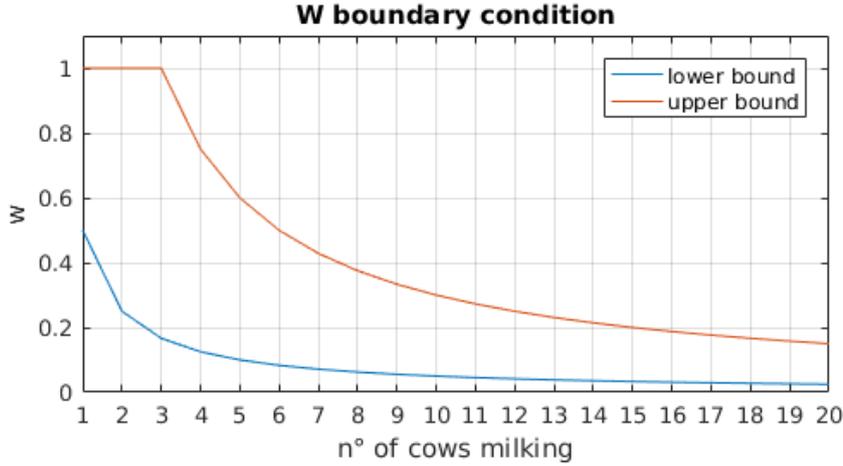


Figure 2.4: W boundary condition during CMA-ES routine

The whole experiment was executed in parallel on an eight-core machine Intel Xenon CPU E5-2680 @ 2.70GHz, 64 GB RAM, Ubuntu 14.04 LTS.

The dataset previously described in Section 2.4 is summarized in Table 2.5.

Table 2.5: Summary of the STR dataset used in the analysis

Farm	No. Cows	No. Pool Samples
A	12	Bulk milk: 12
		Derived Cheese: 12
B	14	Bulk milk: 11
		Derived Cheese: 11

## 2.8 Results

The main goal of this work was to develop an innovative automatic methodology to highlight possible adulterations in dairy products thanks to a computational heuristic analysis. Adopting the method described in the previous sections, we obtained the results reported in Figure 2.5 and Figure 2.6.

Figure 2.5 reports the mean score values computed over the 24 repetitions for the Bulk Milk analysis in Farm A and B. For each sampled P, and for each month, the figure shows the estimation of the three experimental setups delineated in section 2.7 with the changing forgery rate. Figure 2.6 reflects the results of the cheese forgery simulation following the same criteria of Figure 2.5.

Results are overall very good since we obtained higher scores in case of adulteration and, oppositely, scores close to 0. Moreover, it can be seen that partial forgery

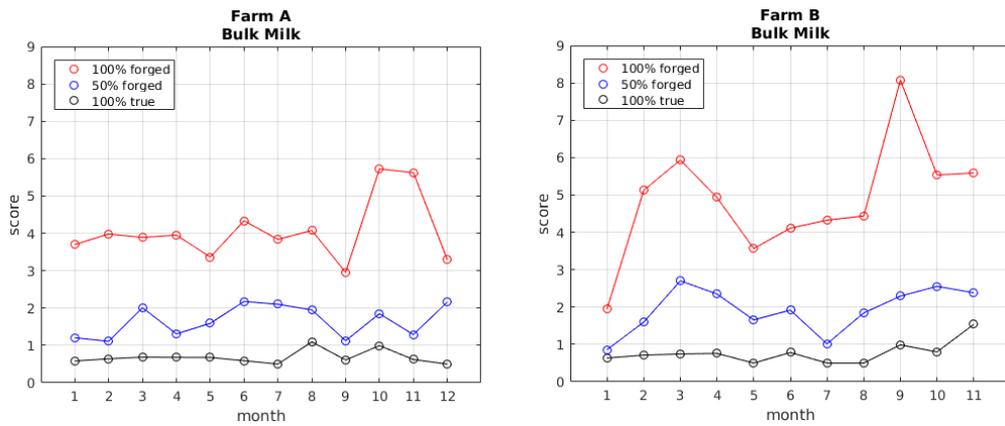


Figure 2.5: Results of the mean score values for Farm A (left side) and Farm B (right side) for the BULK MILK analysis for each available month. Black lines are related to 100% true cows setup analysis, the blue ones are related to 50% of adulterated milk origin, and the red ones are 100% forged milk origins.

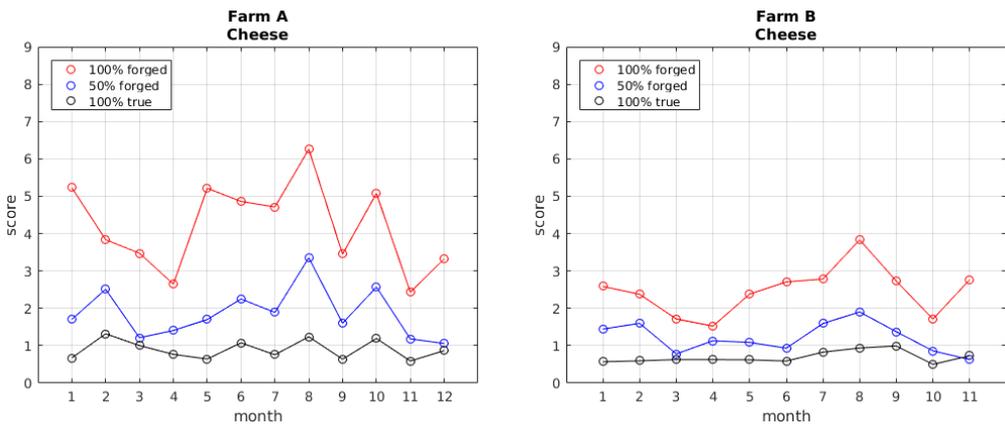


Figure 2.6: Results of the mean score values for the Farm A (left side) and the Farm B (right side) for the CHEESE analysis for each available month. Black lines are related to 100% true cows setup analysis, the blue ones are related to 50% of adulterated cows and the red ones are 100% forged cows.

simulations are globally between 100% forged and 100% true examples. This trend is confirmed both in BM and CH simulation. In the majority of the cases, the proposed automatic forgery detection reveals a considerably good accuracy with the exception of a few examples.

A summary of the aggregated results is given in Figure 2.7. Here the box plots represent the grouped results of Figure 2.5 and Figure 2.6, respectively. In general, the scores obtained for BM and CH simulation indicate that it is possible to characterize our model with progressive cut-offs able to define if a forgery has occurred. In the figure, BM boxes

are noticeably well separated, while the CH boxes show a less sharp separation in particular in the Farm B between the 50% forged and the 100% true group. Maybe the reason is that probably the STR profiles of the Farm A, that occur in the random selection of false cows in Farm B, are too similar to the correct ones and only with an increasing rate of forgery the output scores are extensively revealed.

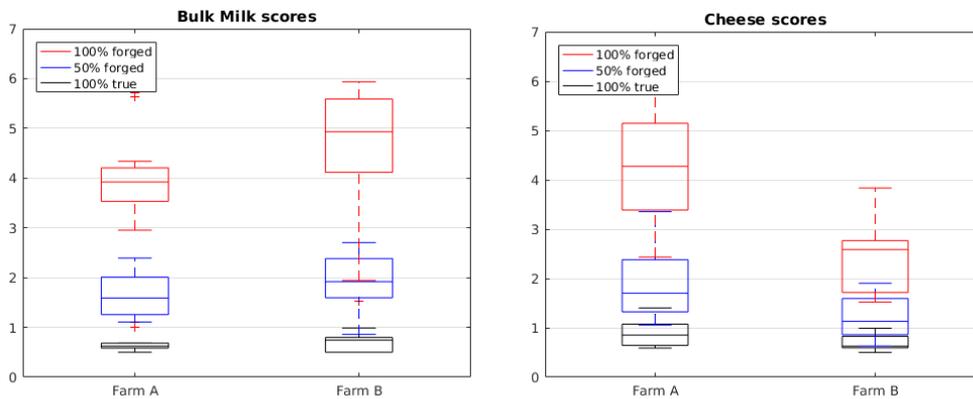


Figure 2.7: Box plots of grouped scores for the Farm A and B in the bulk milk and cheese analysis. Black box are related to 100% true cows setup analysis, the blue ones are related to 50% of adulterated cows and the red ones are 100% forged cows

The overall results of the dairy product analysis are shown in Figure 2.8. The global scores are grouped together only to highlight the differences between the true simulation and the other two percentages of adulteration. Notice that here Farm A and Farm B are merged, just like BM and CH.

The distinctness among the three simulation groups (100% true, 50% forged and 100% forged) is statistically significant ( $p < 0.05$ , with the Kolmogorov–Smirnov test). This result suggests that it is possible to define a cut-off between different levels of dairy product counterfeiting score (e.g. score=1 define adequately the limit for the not forged product against half or complete falsified ones, moreover a score=2.5 could be effective for absolute falsifications).

It is evident that the proposed automatic forgery detection model is capable to identify the occurrence of irregular dairy product manufacturing. Moreover, results prove that is also possible to quantify the fraud magnitude. Moreover, this project advice that this methodology may provide a feasible strategy suitable to other “food traceability” context with similar setup and characteristics.

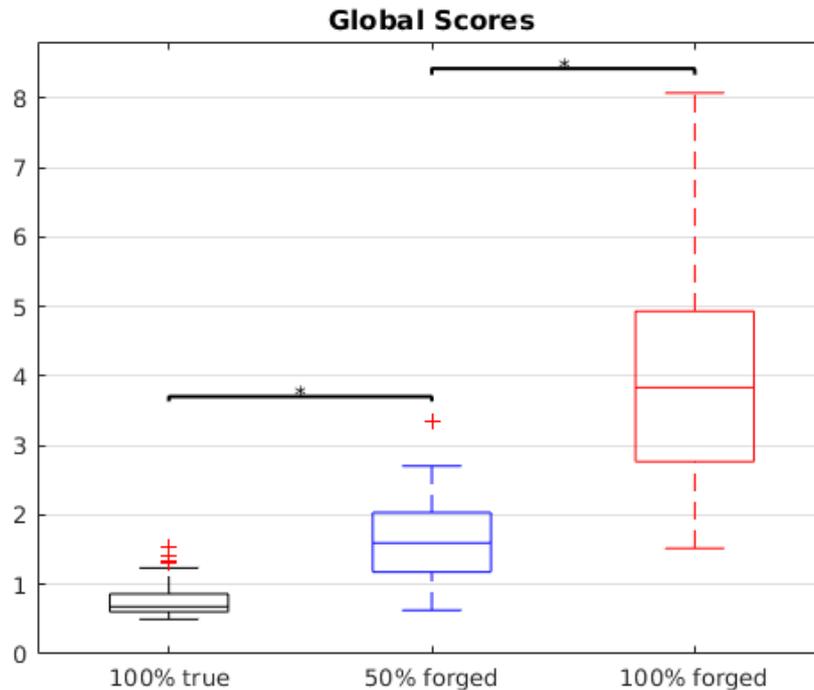


Figure 2.8: Global simulation scores. Both farms and dairy products are grouped. The black box is related to 100% true cows setup analysis, the blue one is related to 50% of adulterated cows and the red one is 100% forged cows. The “\*” indicate significant difference between groups ( $p < 0.05$ , K-S test)

## 2.9 Conclusion

In this chapter, I described an innovative automatic forgery detection method based on a heuristic procedure. This tool proves to be able to assess the likelihood that a traditional dairy product originates from a known farm or not. Furthermore, the measure of the potential counterfeiting is quantified. I investigated the use of STR associated to their RFU to estimate the cows' contribution in the final pool. I made use of a CMA-ES algorithm able to predict the traceability of dairy products and their corresponding producer.

Results obtained in several experiments provided excellent outcomes and can encourage the research community to further investigate trying to employ this method to other foodstuff traceability issues. Moreover, from a biological point of view, the STRs have been tested successfully on small dairy farms. This outcome could promote additional investigation into other correlated studies.

## Chapter 3

# Vision and Learning in Fish Species Identification

### 3.1 Introduction

In this chapter, I will present my contributions to the discipline of food fraud [1.2](#). Actually, this is the circumstance where an intentional generic manipulation is performed firstly to obtain an economic gain. Anyway, this forgery could also impact on secondary effect like unintentional consequences on peoples' health, becoming an articulated socio-economic concern, which contributed to increase peoples' awareness of what they eat.

One of the most important food fraud regards the identification of fish species which represents a commercial fraud implemented by substitution of valuable species with others of lower value. Fish species identification is mainly performed by morphological recognition of anatomical features of the entire fish. However, there are two important factors that make nowadays morphological species identification so challenging. On one hand, there is a constant growing lack of awareness by consumer, on the other hand on the market it has been observed an increasing presence of new little-known species [\[33\]](#).

In this scenario, it appears of greatest importance to predispose analytical methods used to perform a correct identification of fish species to assure the end consumer about sureness and origin of fisheries product but also to facilitate health inspectors analysis work.

### 3.2 State of the Art

In recent years a lot of emerging techniques have been developed to face the problem of fish identification. The FAO review on fish identification tools for biodiversity

and fisheries assessments has focused the attention on a large number of effective current practices [19]. In this report, they include a universal long-trusted method such as trained taxonomist, reference collections of guides based on dichotomous keys, along with more up to date automated techniques. Some examples are image recognition systems (IRSs), computer-based morphometric identification (IPez [23]), interactive electronic keys and genetic methods. However, such methods are not suitable and user-friendly for non-specialist, they are not designed for consumers and, most important, they have not yet been remodeled into convenient application tools.

The principal attributes to evaluate a fish recognition system are [19]:

- Response Time: denotes how fast is possible to obtain an outcome and consequently if it can be applied in the field or not (e.g. labs);
- Accuracy: defines the precision level of the tool and sets the error rate;
- Resolution: explicit the standard of the information, for example, if it is acceptable the order of a specimen or if it is necessary to determine more details such as the family or the species;
- Type: defines whether the characterization should be achieved by examining fresh instead of a frozen specimen or if an examination need the whole or just a portion of their body/tissue;
- Resources: determine which are the costs and expertise required for the activities. They depend on the skill level of the operator and equipment that are devices, facilities and can range from very low to high.

Among all the available approaches, we can focus and compare two of them: genetic and image recognition system. The first procedure gives the guarantee to have the highest accuracy and resolution [50]. Anyway, this technique requires time to elaborate a result, it needs complex and expansive instrumentation under expert supervision. This is obviously limited to health inspector and qualified controllers. The second one, on the contrary, can benefit from fast response time and fewer resources since imaging techniques can be used also by unskilled users. Besides, thanks to the expanding artificial intelligence methodology applied in the image learning field, they are also acquiring attention thanks to their significant accuracy. In literature, there are many approaches concerning image processing and machine learning techniques applied to fish species recognition. Anyway, their employment still remains at an academic level and should be transferred to end user technology solution [76] [75] [64] [47] [70] [77] [1] [29] [20] [37].

### 3.2.1 Filling the gap

In this scenario, the widespread diffusion of mobile devices (e.g. smartphone and tablet) combined with cloud computing resources can be used as a valid tool for filling the gaps and limitations summarized above. Mobile devices, also thanks to their camera performance, are an affordable sensor system enabling a multitude of users to collect high-quality pictures of fishes. Furthermore, through internet connection, it is easily possible to communicate and transfer images to Cloud/Web services convenient to perform an intensive computational process. Cloud computing can serve as infrastructure to implement an algorithm, to process the acquired images and to put into action machine learning techniques to carry out classification and computer vision tasks. Afterward, whatever information such as generic data or images can be quickly sent back to the user or saved for additional purposes. For these reasons, cloud computing in combination with mobile applications can represent a particular instrument to provide a valuable service to counteract fish falsification.

## 3.3 The “FishApp” project

In the first part of this chapter, I will present FishAPP, a cloud-based infrastructure for fish species recognition. At bottom, FishAPP is composed of a mobile application and a remote cloud server. The mobile application developed both for Android and iOS operating system, allows users to take a picture of a whole fish and submit them for remote analysis to the cloud server. On the other hand, the cloud-based processing system performs a complex image processing pipeline and a neural network machine learning classifier to analyze the obtained images and to solve the classification into predefined fish classes.

In the next subsection are described in more details the infrastructure and the working principles of FishApp. Eventually, preliminary results obtained from this study will be presented.

This work has been possible thank to the collaboration with the Genetics and Immunobiochemistry Team of the IZSTO <sup>1</sup>.

The content of this activity has been published in [54].

### 3.3.1 Infrastructure Design

The FishApp front-end module is a responsive mobile application designed for the two most widespread mobile platforms, iOS and Android. The application employs the camera of the mobile phone to use the sensor for capturing an image of the fish. This picture will be then used for further analysis on the cloud server. The interface has been

---

<sup>1</sup>IZSTO Genetics and Immunobiochemistry Team [link](#)

implemented to work in a multi-touch mode. This allows the user interacting during the recognition analysis by providing useful information that both increases the quality and accuracy of the analysis as described later. The mobile application is connected to a remote cloud server that executes the computational analysis stage. The remote server performs two main tasks:

- Image Processing pipeline: to extract *information* (i.e. features) from *data* (i.e. fish picture);
- Machine Learning classifier: to employ the *information* to generate *knowledge* (i.e. classification based on training).

The Cloud is also equipped with a storage system to record all analyzed images and data. That information may be then used to improve the recognition capabilities of the system or to revise any step of the process.

Figure 3.1 shows the FishAPP software architecture described above.

Details regarding FishAPP mobile application and the remote server will be provided in the following subsections.

### 3.3.2 Mobile App and Remote Server Interaction

The FishAPP mobile application software enables smartphones and tablets to capture the photo of a fish, or to select one from the local device photo library, and to connect with the FishAPP remote server (see Figure 3.2). The mobile application software has been developed with PhoneGap [69], a free framework and open source project that allows developers to create mobile apps using a set of standardized web APIs for the desired platforms.

The picture of the fish must include its full body and must respect the following guidelines:

- The fish must be photographed sideways in landscape orientation;
- The caudal fin must be left in relaxed anatomical position;
- The other fins should be set in a close-fitting shape, in order to not modify the fish's body contour.

In fact, fishes for foodstuff aim are inanimate and they cannot maintain the other fins completely visible. Then I decide to consider only the caudal fin to further derive anatomical features (as visible in Figure 3.2).

Once the fish photo is ready, the user has to select the supposed fish species name (label) from a menu icon. At this point, the image and its selected label can be transmitted together to the cloud server. The choice to let the user selecting the species' label is motivated from a classification point of view. Instead of querying the server for all

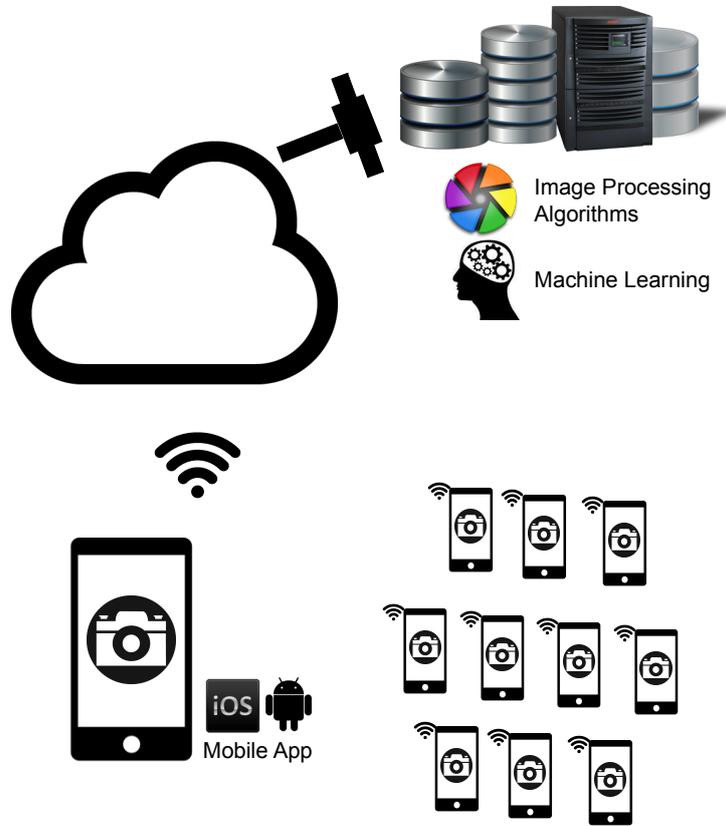


Figure 3.1: FishAPP software architecture

the available fish species in the database, the classification method looks only for the specific user-selected species. This means, from a methodological point of view, that the classifier can be designed to execute a one-class classification instead of a multi-class one. Moreover, this distinctive feature is comparable to the way the user (i.e. both health inspectors and consumers) perform during controls or into the selling point when they look at the label exhibited upon the fish. In fact, food traceability applied fish task must answer this question: *Is really this the species declared on the label or not?*

When the cloud server receives the picture of the fish, then the image processing unit starts. The pictures are analyzed to perform the identification with a set of marker points corresponding to some specific anatomical position. These markers can be confirmed immediately by the user via through the multi-touch user interface. In fact, the user can refine this points selection by dragging and then the remote server computes the feature extraction and performs the final classification from the modified

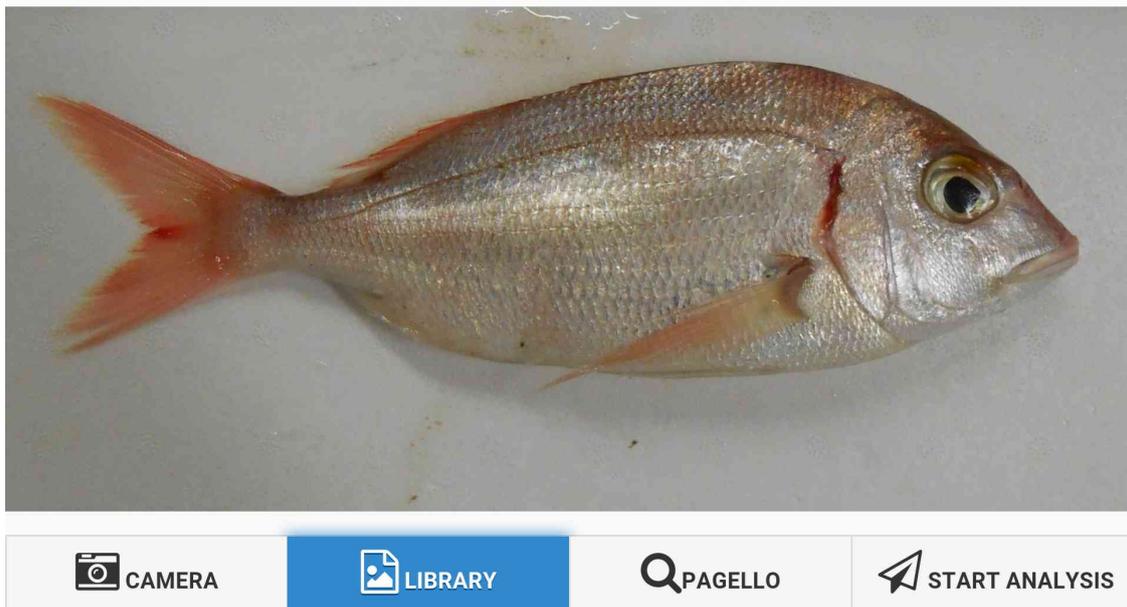


Figure 3.2: Example of fish picture taken from smartphone local storage library & FishAPP GUI

ones. When the server receives the confirmed marker point set, then it can execute the last classification step. As soon as the server produces the result of the classifier the outcome is sent to the FishAPP mobile application and the fish species identification is now concluded.

The whole image processing pipeline, the key-points identification steps to extract the features and the classification method are all described in the next subsections.

### 3.3.3 Image Processing Unit

The picture of the fish needs to be processed to extract discriminative features for classification. In this work, I have designed 30 geometrical features that represent some fishes anatomical characteristic. All these features are calculated when the 12 marker-points are detected as shown in Figure 3.3. The image process unit has been developed in C++ language using OpenCV free cross-platform computer vision library [10].

To find the key-point on the fish the first step to perform is the fish segmentation. Once the specimen is disconnected from its background, it is then possible to apply the find-point routine. In Figure 3.4 is represented the image processing pipeline to perform the fish segmentation.

The first operation is the de-noise step, performed by the Bilateral filter. The next step has been designed to find a very approximative fish border. It is computed by applying the Adaptive Threshold filter (AT) to the original grayscale image in combination

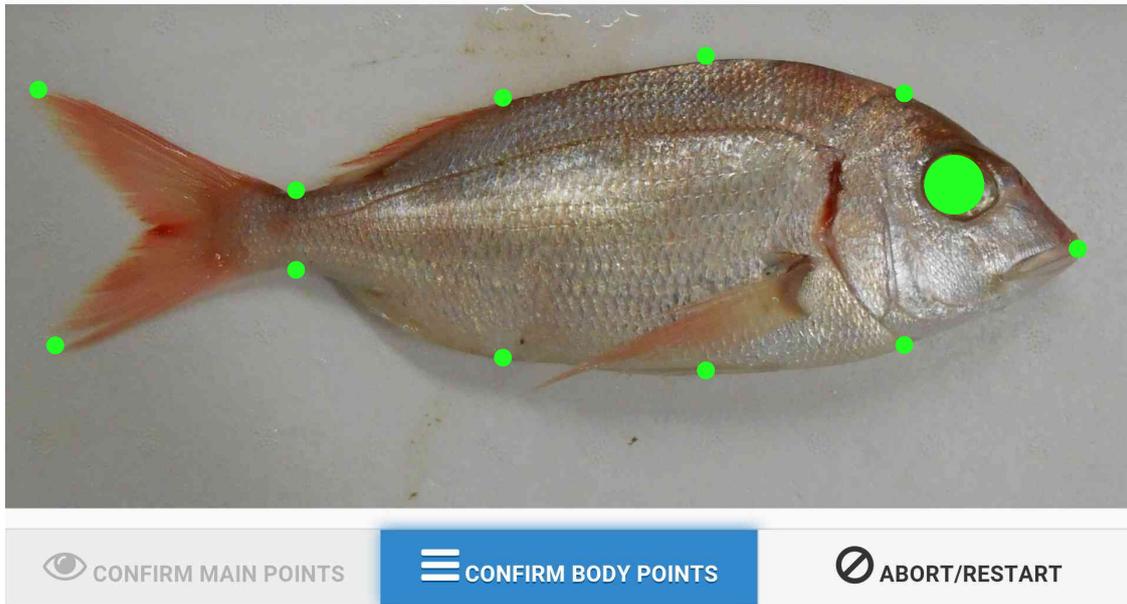


Figure 3.3: Anatomical set points detected by FishAPP image processing unit

with the Canny-Edges (CE) filter followed by the Laplacian filter (L) computed the original colored image. I found that this sequence of filters (i.e.  $ATx(CE+L)$ ) allows identifying complementary border since the light reflection of the fish (which is often wet) and the shadow projection make not reliable a direct image segmentation in only one step. More in details, the AT compensate the color gradient irregularity caused by the shadow intensity, while the CE+L makes more evident the color difference between the background and the fish.

The following macro-step is the identification of the rough fish mask body detection. After a simply Blend with previous border images, then the Contour operation finds a unique shape-line that is therefore filled with FloodFill filter. At this point, some simple dilation-erosion are performed to complete the first binary mask detection indicated as ROI (i.e. region of interest).

Eventually, on the rough mask, is performed the GrabCut filter (GC), that is an image segmentation method based on graph cuts iterative steps [55]. The ROI mask over the original picture is the starting point for GC, called ROI Fish (ROI-F). Then the GC is executed by setting as a foreground the ROI-F and as a background the rest of the image. It is important to notice that, at this stage, to make the entire process faster, but yet precise, a downsampling of the image is applied (scale factor = 2) before the GC process. When the GC is done, then the obtained result is upscaled to figure out as it was before.

At this point the very high-quality fish segmentation took part. The foreground fish's body is segmented whereas the background, together with shadow effects and/or non-homogeneity, is filtered out. Obviously, the more the fish picture is correctly taken

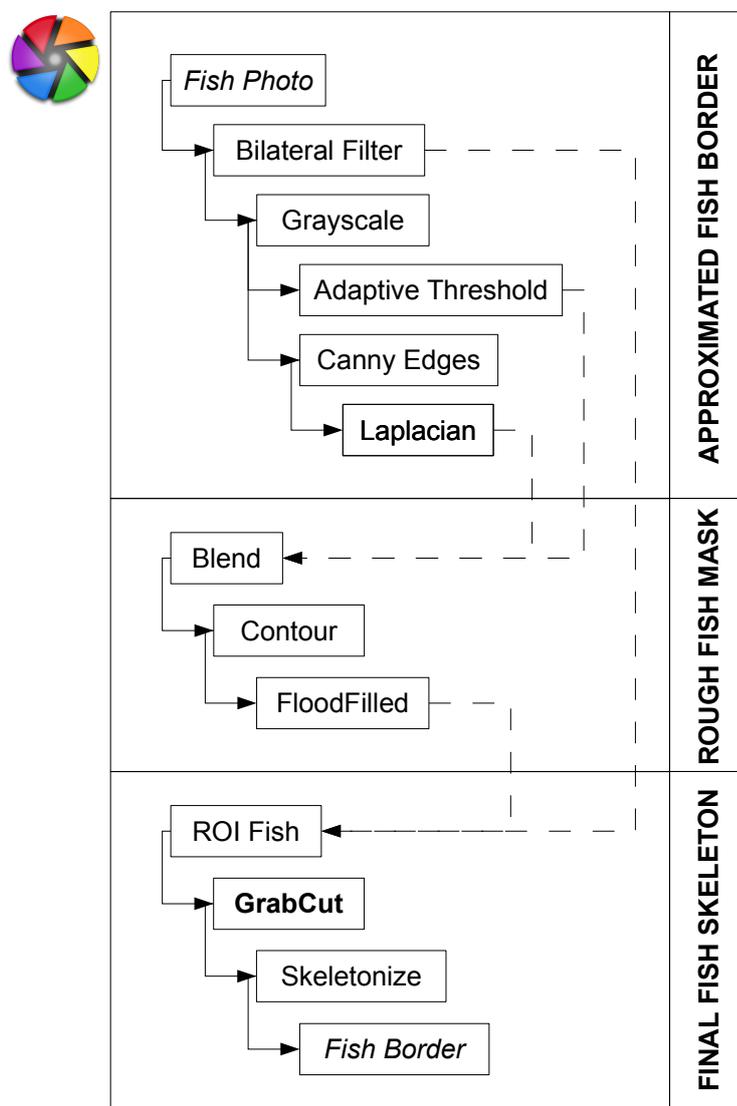


Figure 3.4: Summarized FishApp image processing unit

by the camera the faster and the more accurate is the final segmentation. Eventually, the final mask is then skeletonized. The resulting binary image (see Figure 3.5) then passes to the next key-points identification step.

This image process unit has demonstrated to be effective in a real setup where the fishes have been photographed over a homogeneous background without other objects.

To whom it may interest the OpenCV official documentation, the content is available at the following link: <https://docs.opencv.org/2.4/>.

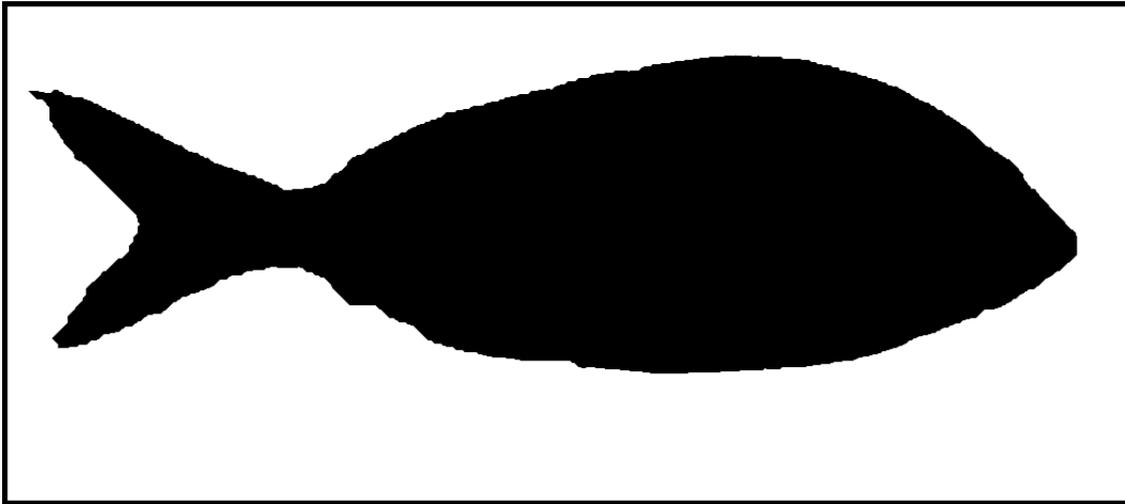


Figure 3.5: Fish segmentation result (i.e. binary mask)

### 3.3.4 Features Extraction

The output of the first part of the image processing unit is the segmentation of the fish shape, the following is the key-point detection of the fish. The C++ procedure developed for this purpose has the goal to find the main twelve points as illustrated in Figure 3.3. Eleven points are located along the border (PtsBody) while the last point is the fish eye detector (PtEye). The implemented procedure to find the PtsBody start with the mouth of the fish, since it is the extremity that converges into a singular convex point. Then, on the opposite side, the same criteria is used to locate the two convex points that determines the final part of the caudal fin. Once both the extremities are done, the next step of the key-point search regards the caudal peduncle. This couple of points are found at the intersection between the fish mask and a particular transverse axis. This last one is the line that crosses the fish binary mask projection along the longitudinal axis in the minimum height. In other word, the caudal peduncle points are in the minimum width of the fish mask. Taking into account this last two points and the mouth one, the 2 main key-point of the body are found in a similar but opposite way looking for the maximum width in the trunk. The remaining four points are finally identified as the point along the border in the middle projected distance between the trunk points toward the mouth and the caudal points respectively.

On the other hand, the PtEye is determined inside the fish body but only in the head region, defined as the fish part from the two main points in the trunk toward the mouth. Into this fish portion is then executed the following pseudo over the original image:

1. Mean Shift filter (MS),
2. Contrast Limited Adaptive Histogram Equalization filter (CLAHE);
3. Hough Circles filter [18] (HF).

The MS serves to perform a basic color segmentation inside the selected area in order to reduce color noise and exclude little cluster of the colored pixel with a different characteristic of the surround. The CLAHE is then applied to intensify the contrast so that the HC can perform the circles' identification. The desired result is an eye-centered circle with the same eye diameter.

Because of possible light reflection, blood stains presence in the cornea and due to intra-family color difference the PtEye is sometimes less accurate than the other ones. Anyway, if there is any millimetric gap with respect to the fish eye position, the user can easily modify the position and the diameter of PtEye and then confirm all the detected key-points before performing the feature extraction as described before in 3.3.2.

At this point, the remote server analysis has detected all the 12 key-points and it is now able to find the final 30 features. These features are deduced as the main straight geometrical line that connects the key-point to each other as disclosed in Figure 3.6. Four features are related to the fish eye position (i.e. the three pink lines) and the fish eye diameter. The remaining feature is represented by the cyan lines and is a sort of inter-key-point measure that on the whole describe the dimension, the geometry and the relationship between specific anatomic point. All these measures are finally normalized with respect to the maximum length of the fish. This trick is also needed to train correctly the ANNs classifier. Eventually, all the features are saved into Comma Separated Value format file (\*.csv).

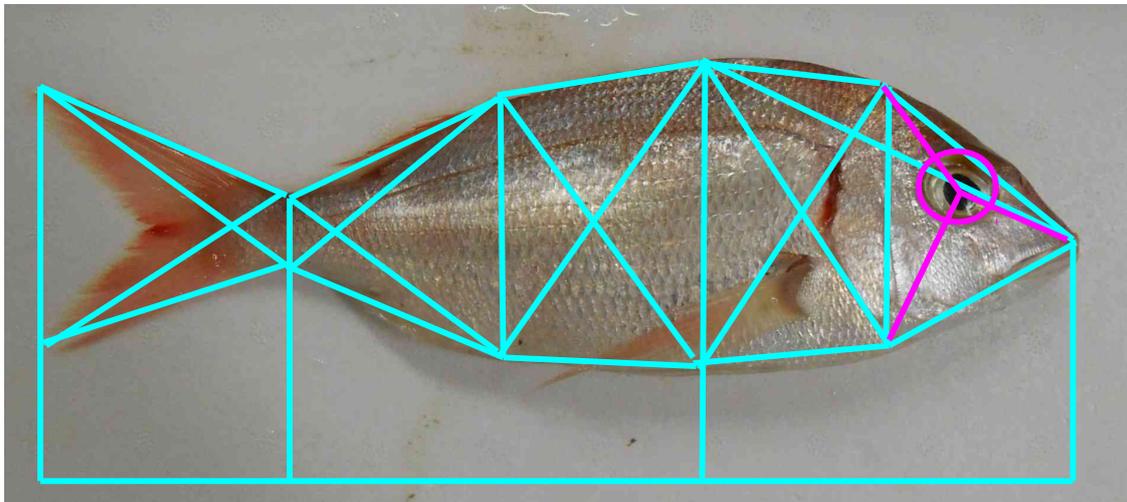


Figure 3.6: FishAPP feature extracted. Blue and pink lines represent the morphological dimensions of the fish.

### 3.3.5 ANN Classifier

Artificial neural networks (ANNs) are a family of machine learning models inspired by biological neural networks and are widely used to solve pattern recognition problems

[7]. Such models are able to learn by considering examples. They automatically evolve their own set of internal parameter from the learning data that they are designed to process.

In this project I trained two different types of ANNs in order to divide the classification into two distinct parts: the first one performs a cluster identification that is the fish Order while the second one is the fish species recognition. From a computational point of view, the fish Order clusterization is performed with ANNs trained as a One-Class classifier (ocC) whereas the fish species identification is achieved thanks to ANNs trained as a Multi-Class classifier (mcC) [9] as reported in the Figure 3.7. The identification of the Order from the species information is based on a structured taxonomy list that operates at the beginning of classification.

As it was described in the previous section 3.3.2 when the user takes a picture of a fish and makes a request to identify the species he must select the label of the species from a list in the app interface considering its own Order membership. The selected name species is hereafter associated with the fish picture and the image is at first employed in the ocC. If the ocC output confirms that the fish Order is correct, then the mcC for species identification takes part in the second classification step. Otherwise, if the ocC rejects the fish, then the user can change the fish name and try with another query to the FishAPP server. Eventually, the final result of the mcC is a list of intra-Order species membership confidence scores.

Both the ocC and mcC have been designed with near the same characteristic as reported in the following list:

- n° neuron input layer = 30 (i.e. the features number)
- n° hidden layer = 2
- n° neurons hidden layer = [20, 10]
- n° output layer =  $\Omega$
- optimization: stochastic gradient descent (SGD)

The two ANN typologies, ocC and mcC have the following differences. For ocC  $\Omega$  is equal to 1, since it produces only a binary output (e.g. “in/out”, “yes/no”, “Clupeidae/not Clupeidae” and so on ...). Otherwise, in the mcC  $\Omega$  is equal to the number of species available in the database training dataset belonging to the same Order. The second difference is that mcC are designed with softmax cost function in the last layer [8]. Softmax is generally used in the case where we want to handle multiple output classes (i.e. fish species). The mcC output is a  $\Omega$ -dimensional vector (whose elements sum to 1) giving us our  $\Omega$  estimated probabilities.

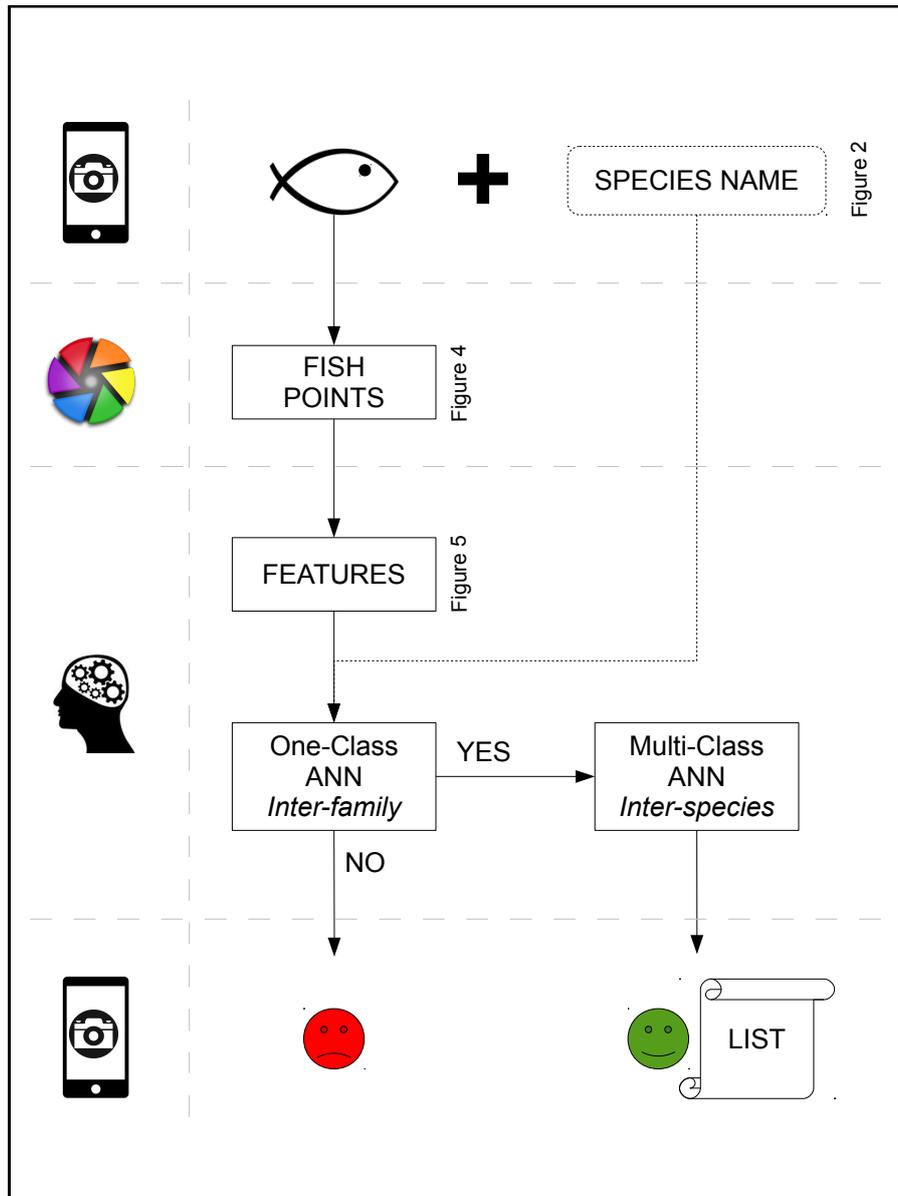


Figure 3.7: Classification Pipeline

### 3.3.6 Dataset

The dataset has been collected by me in diverse sessions at the Turin wholesale fish market <sup>2</sup>. I was able to photograph 339 fish samples. In the following table are reported the species with the corresponding binomial nomenclature and the number of pictures:

<sup>2</sup>COMIT - Consorzio Mercato Ittico Torino, Italy [link](#)

- European Anchovy, *Engraulis encrasicolus* (125);
- European Pilchard, *Sardina pilchardus* (107);
- Common Pandora, *Pagellus erythrinus* (20);
- Atlantic Mackerel, *Scomber scombrus* (18);
- Gilt-Head Bream, *Sparus aurata* (22);
- European Hake, *Merluccius merluccius* (19);
- Striped Red Mullet or Surmullet, *Mullus surmuletus* (28).

Due to the number of pictures and the species available in the market I was able to train one ocC for Clupeidae Order with a target class composed of European Anchovies and European Pilchards against the rest of the dataset (232 (125+107) vs 339). The Clupeidae inter-species mcC was then trained considering these two classes: European Anchovies and European Pilchards (125 vs 107).

The entire dataset has been processed with the FishAPP image processing unit and the feature extraction pipeline. Hence, for each image in the remote database corresponds its respectively feature csv file. During the pre-training data processing at the key-points confirmation step some adjustment to the points position have been done, by dragging the incorrect points as reported in the previous section 3.3.2.

### 3.3.7 Results

To evaluate the accuracy of the FishAPP species identification we implemented a leave-one-out cross-validation (LOO-cv) routine. Therefore, we also performed several ‘in-field’ and ‘real time’ validation in the fish market.

The LOO-cv result final accuracy was excellent: *Engraulis Encrasicolus* v.s. 100% (339/339). In fact, all the European Anchovies (125/125) and European Pilchards (107/107) were correctly identified for their Order and species membership (ocC and mcC classifiers), whereas all the other fishes were filtered in the first ocC classifier (232/232). In particular, in the mcC the European Anchovies and the European Pilchards were perfectly recognized. Results obtained during two different ‘in-field’ validations have confirmed the same accuracy results.

### 3.3.8 Research outcome on national media

This project has been named “Ok il pesce è giusto” (the English translation is “Ok, the fish is right”) and it was presented for the first time on June 2015 in Turin in the city context related to “Expo Milano 2015”. In Figure 3.8 it is shown the FishApp image presentation.



Figure 3.8: Presentation image of FishApp

Thanks to the media resonance from “Expo” presentation we were later called by RAI Italian Public Television. On that occasion, I demonstrate on live stage the FishApp working principles in the broadcast “Uno Mattina Estate” on RAI1 channel (see Figure 3.9).



Figure 3.9: Television demonstration screenshot

Finally, a further demonstration in a Turin fish market was also asked by the regional “TGR Piemonte” newscast (see Figure 3.10).



Figure 3.10: Newscast demonstration screenshot

### 3.3.9 Limitations and Constraint

The FishApp project demonstrated to be effective on the available dataset at that time. The obtained results were encouraging and the media exposure underlines the growing interest in this Food Traceability issue. Both from health inspectors and consumers, the user-test usability demonstrates that the combination of remote cloud infrastructure and machine learning techniques accessible via mobile application is a very interesting and promising auxiliary method to counteract the problem of fish species substitution.

Anyway, we can underline some of its principal limitations. The most important one concerns the image processing unit (3.3.3) and the feature extraction (3.3.4). It is clear that the key-point identification and the consequent feature extraction mainly depend on the previous step regarding the fish image segmentation. The remote cloud server procedure, that analyzes the fish pictures, demand for images taken under specific conditions. If the fish do not have a uniform contrasted background, then the fish segmentation could fail. Furthermore, the same effect may be also caused if the environmental lighting produces intense shadow effects which can deface the fish shape. Moreover, we must observe that in the middle of the mobile-server pipeline procedure the user have the possibility to slightly modify all the key-points in case those are not identified correctly. This aspect has two main important consequences: the first one is the time consuming, the second one is that the users are asked to actively take part in the analysis. If users are not so experienced then they could affect the classification accuracy due to incorrect key-points correction.

Another important limitation is that all the classifier must be trained with the features file previously elaborated, preferably by a skilled expert, and this is really time-consuming from a managing point of view.

To increase the number of fish species in the database means also that maybe the original image processing unit must be modified or adapted to new need. In fact, the key-point and then the features could be not useful or representative for other fish species such as flatfish or species with the eye barely visible.

In order to overcome all these limitations, finally, we decide to restart, trying to look at the same task (see 3.1) but with a different technique and point of view: the Deep Learning. In the next section are described the reasons for this choice.

## 3.4 Deep Learning in Image Recognition

Deep Learning (DL) is a class of a wide family of Machine Learning (ML) methods based on learning data representations. It uses a cascade of multiple layers of nonlinear processing units (e.g. perceptron) to perform feature extraction and transformation. It can be used to learn in supervised or unsupervised mode (e.g. regression, classification, pattern recognition). The multiple layers sequence corresponds to the abstraction of hierarchy levels of representation [6] [36] [59] [57].

In DL the “deep” refers to a large number of layers, or connections, through which the data are transformed. Deep Neural Network (DNNs) is the most common model that performs such computations [21].

In the last decade, DL improvement has been effective also thank the progress of Graphics Processing Units (GPUs). In fact, GPUs are convenient for vector/matrix math operation. The speedup in term of training time has been upgraded by several orders of magnitude with respect to standard CPUs [51].

DL is actually part of state-of-the-art systems in various disciplines. One the most interesting field in which DL is applied are the Computer Vision (CV) tasks. In CV it is mainly used a class of DNNs called Convolutional Deep Neural Networks (CNNs). CNNs use a variation of multilayer perceptrons, that perform convolution, to solve many tasks such as identification, detection, recognition, and in general, to analyze visual imagery [61] [73]. CNNs, like ANNs, were inspired by biological processes. In particular, in the convolutional layers, they derive from the connectivity pattern between neurons like in the animal visual cortex organization (see Figure 3.11) and hierarchy operation [41].

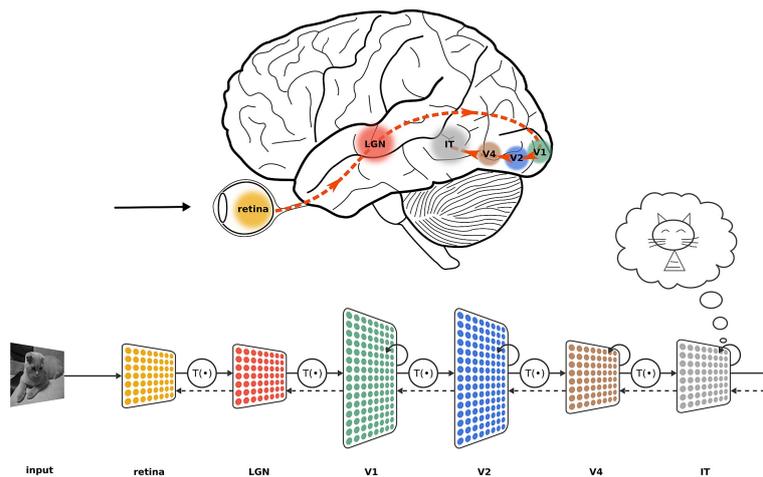


Figure 3.11: Relationship between the brain visual process and modern neural network

A CNN consists of an input layer, multiple hidden layers, and an output layer as well ANN. The same as visual stimuli, the CNNs layers work as a receptive field, passing the result to next layers, thus to avoid feature engineering. This is the most important peculiarity that produces more accurate results than human contestants in some specific applications [14] [34].

Since DL models have a huge number of parameter to learn, the require a lot of data to be accurately trained [36]. In this era of big data transformation, is possible to obtain many data concerning a specific task. For almost every kind of application is possible

to rely on large datasets to perform effective learning<sup>3</sup>. Anyway, it may happen that, for a specific task, researchers and developers do not have such availability of large and challenging datasets on which deep learning models are trained. Moreover, it can also be possible that for hardware limitation (both GPUs and CPUs) is not feasible to deploy deep models in reasonable time. To face these limitations is possible to adopt the transfer learning principles.

### 3.4.1 Transfer Learning

Transfer Learning (TL) is a ML technique where a model trained on one task is re-used on a secondly related task [45]. From a different point of view, TL is an optimization technique that allows rapid progress or improved performance when modelling the second task from a related task that has already been learned before [48].

In CV this means that is possible to apply the knowledge modelled on a specific dataset to a new CV task where many data are not available. Since CNNs are able to extract visual features during training, we can transfer this know-how to represent a similar problem.

The feature representation in CNNs is structured hierarchically. Usually, the very first layer is able to identify *low-level features* (e.g. pixel-clusters of colors and gradient line orientation). Going into the deep, in the next layers, these features are combined in *mid-level* and eventually in *high-level features*. For example, the model is then capable to recognize more complex features, such as geometrical figures or specific shapes related to training data, and so on. All these features, in the end, are used to generate discrimination or detection (depending on the specific task).

These patterns are obviously trained depending on input data. Anyway, the feature extraction capability of the network can be used to facilitate a slightly different problem. This allows reducing the training time and potential to increase the accuracy on the second task [46] [72]. This form of TL used in DL is called inductive transfer. This is where the model bias is narrowed in an advantageous way by using a model fit on a different but related or similar task.

In CV it is common to use a DL model by transferring the knowledge from a large image classification task, such as the ImageNet photograph classification competition<sup>4</sup>. This induction is also called “pre-training” and in some cases is a must to start correctly a new classification problem. In fact, these very complex models can take days or weeks to train on modern hardware.

According to [45] there are three possible benefits to look for when using transfer learning (see Figure 3.12):

---

<sup>3</sup>Some Deep Learning datasets [link](#)

<sup>4</sup>ImageNet is freely available here: [link](#)

- Higher start: before refining the model the initial skill on the source model is higher than it otherwise would be.
- Higher slope: the rate of improvement of skill during training of the source model is steeper than it otherwise would be.
- Higher asymptote: the converged skill of the trained model is better than it otherwise would be.

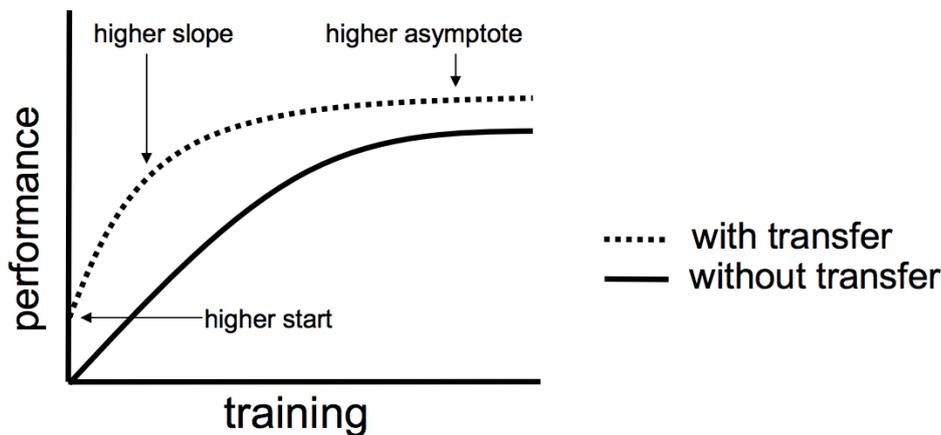


Figure 3.12: Three ways in which Transfer Learning can contribute

Nevertheless, the choice of source model or dataset to perform TL is an open problem and may require domain expertise or intuition matured via experience.

### 3.5 The “F.I.S.HUB” project

In the previous section, 3.3 has been presented the FishAPP project and the mission of that work. The limitations explained in subsection 3.3.9 stimulate us to investigate a more reliable method to face the problem of fish species substitution and mislabelling.

Thanks to the funding received in the context of the FoodIntegrity framework<sup>5</sup> (under the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement No. 613688) we were able to follow the previous project.

The objective of the Fish Identification Software Hub (F.I.S.HUB) project is to overcome the limitations of both methods by developing a software framework to be used in the field, by both professionals and lay people, to detect species substitution. The

<sup>5</sup>Food Integrity Project [link](#)



Figure 3.13: FoodIntegrity logo

F.I.S.HUB software strives for identifying the species of a fish from its picture, without user interaction (like in 3.3.3). It will be based on a photo database and a machine-learning server for image analysis and classification and will be accessible through a user-friendly application for mobile phones and other portable devices. We adopted a DL classifier to solve the problem of fish species identification in order to perform a one-shot detection in a fast and accurate manner.

The partner involved in the F.I.S.HUB project, beyond the Politecnico di Torino, are the “Istituto Zooprofilattico Sperimentale del Piemonte, Liguria e Valle d’Aosta” (IZSTO) <sup>6</sup> and the University of Salford (Manchester, UK) <sup>7</sup>.

In the next subsection are described more particularly the infrastructure and further details related to this project.

### 3.5.1 Software architecture

The software architecture designed for the F.I.S.HUB project consists of three main blocks (see Figure 3.14):

1. Picture Cloud Database: is the collection of all the photographs and the list of photos for each species;
2. Server Classifier: contain the classification engine able to classify each new photo into one of the available species, or to detect “suspicious” features that may indicate a fish substitution;
3. Mobile Application: tool enabling users to take a picture of a fish and, in a short time, connect to the classifier and obtain the result of the classification task.

To improve and increase the training set for the classifier a data augmentation function has been added to the architecture. Data augmentation is a method used to generate

---

<sup>6</sup>Istituto Zooprofilattico Sperimentale del Piemonte, Liguria e Valle d’Aosta (IZSTO) [link](#)

<sup>7</sup>University of Salford (Manchester, UK) [link](#)

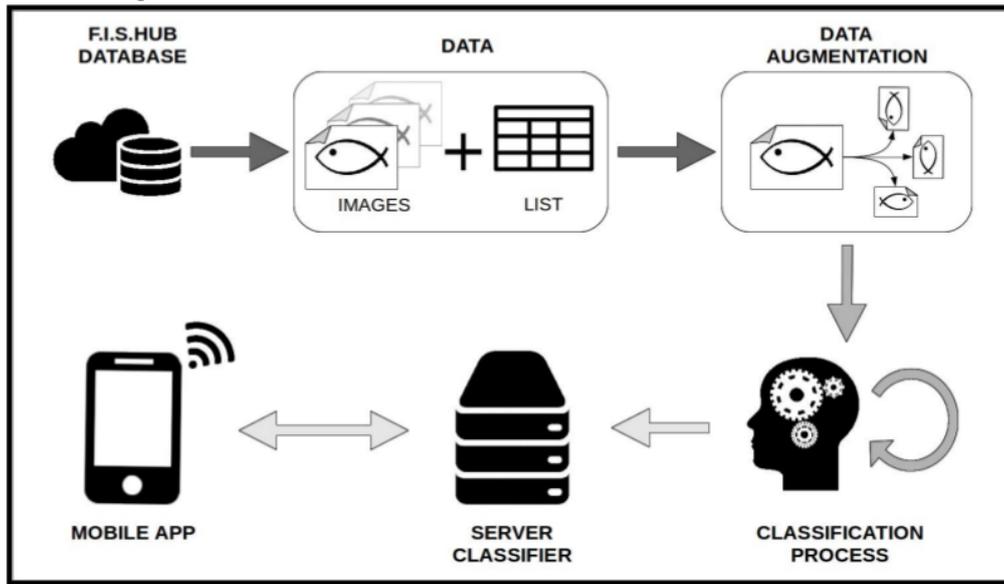


Figure 3.14: F.I.S.HUB architecture

new images, starting from the available ones, by applying some standard modifiers: rotation, shift, shear, flips, whitening and so on. In this way, it is possible to increase the size of the training set (the images used to train the classifier) and enhance the generalization of the classifier without generating overfitting. The amount and the type of data augmentation are chosen in a fitting-friendly way, according to the requirement of the classification process.

### 3.5.2 Picture Cloud Database

The F.I.S.HUB database stores all reference photos of the species that are selected as primary targets for the project. These photos will be used as “reference model” to train the machine learning algorithm that will classify the fish species, therefore a large number of photos will be required to properly represent each single species.

Firstly, we defined the **fish species list** to be initially classified by F.I.S.HUB software to detect possible substitution. The selection was made according to commercial importance and likelihood of substitution. This choice does not limit the scope of the software, because more species and families may be added later to the database. The agreed complete fish list is reported in the following Table 3.1, with the taxonomic classification:

All species belong to four different orders: *Clupeiformes*, *Gadiformes*, *Perciformes*, and *Pleuronectiformes*. The list is globally divided into eight families: *Clupeidae*, *Engraulidae*, *Gadidae*, *Merluccidae*, *Sparidae*, *Pleuronectidae*, *Scophthalmidae*, and *Soleidae*.

ORDER	FAMILY	GENUS	SPECIES	Italian common name	English common name
Clupeiformes	Clupeidae	Clupea	harengus	aringa, aringa atlantica	herring, atlantic herring
Clupeiformes	Clupeidae	Sardina	pilchardus	sardina	sardine, pilchard
Clupeiformes	Clupeidae	Sprattus	sprattus	spratto, papalina	sprat, bristling, skipper
Clupeiformes	Engraulidae	Engraulis	encrasicolus	acciuga, alice	european anchovy, anchovy
Gadiformes	Gadidae	Gadus	morhua	merluzzo nordico/bianco	cod, codling
Gadiformes	Gadidae	Melanogrammus	aeglefinus	asinello, eglefino, haddock	haddock
Gadiformes	Gadidae	Merlangius	merlangus	merlano, molo	merling, whiting
Gadiformes	Gadidae	Pollachius	virens	merluzzo nero/carbonaro	saithe
Gadiformes	Merlucciidae	Merluccius	merluccius	nasello, merluzzo	european hake
Perciformes	Sparidae	Dentex	dentex	dentice	common dentex, dentex
Perciformes	Sparidae	Dentex	gibbosus	dentice corazziere/dalla corona	pink dentex
Perciformes	Sparidae	Diplodus	annularis	sparaglione, sarago	annulus sea bream
Perciformes	Sparidae	Pagellus	acarne	pagello/fragolino bastardo	axillary sea bream
Perciformes	Sparidae	Pagellus	bogaraveo	occhione, besugo, pezzogna, rovello	blackspot/red seabream
Perciformes	Sparidae	Pagellus	erythrinus	pagello fragolino, fragolino	pandora, common pandora
Perciformes	Sparidae	Pagrus	caeruleostictus	Pagro azzurro/reale	blue-spotted sea bream
Perciformes	Sparidae	Pagrus	pagrus	pagro, pauro	red porgy, common seabream
Pleuronectiformes	Pleuronectidae	Hippoglossus	hippoglossus	halibut, ippoglosso	atlantic halibut, halibut
Pleuronectiformes	Pleuronectidae	Limanda	limanda	limanda	dab, common dab
Pleuronectiformes	Pleuronectidae	Microstomus	kitt	limanda-sogliola, sogliola limanda	lemon sole
Pleuronectiformes	Pleuronectidae	Pleuronectes	platessa	platessa	european plaice
Pleuronectiformes	Pleuronectidae	Reinhardtius	hippoglossoides	halibut della Groenlandia	greenland halibut/turbot
Pleuronectiformes	Scophthalmidae	Psetta	maxima	rombo chiodato/maggiore	turbot
Pleuronectiformes	Scophthalmidae	Scophthalmus	rhombus	rombo, rombo liscio/di rena	brill
Pleuronectiformes	Soleidae	Solea	vulgaris	sogliola, sogliola comune	sole, dover/black sole

Table 3.1: F.I.S.HUB fish species list

Many similarities are shared among species on an intra-family level. Such similarities are identified and highlighted during the classification process. Additionally, due to morphological differences among families, the future family addition will be easily managed since a large set of features are already required for this inter-family classification.

Then we define a **photo protocol** to ensure high quality of the photographs stored in the database. It has been considered as standard procedure and followed when acquiring photographs for the F.I.S.HUB database. The photo protocol contains the main guidelines for the pictures that are the following:

- Picture size >5 Mpx
- Auto White Balance (AWB) setting
- Landscape Orientation
- No flash
- No more than one picture per fish
- No zoom in fish parts/details
- No missing or flattened caudal fin
- No texture in the background

- No bent fish
- No fishes in box
- No multiple pictures of the same fish

The guideline has been shared with project partners in a mobile-friendly size format that is easy to use. The guideline is reported below in Figure 3.15.

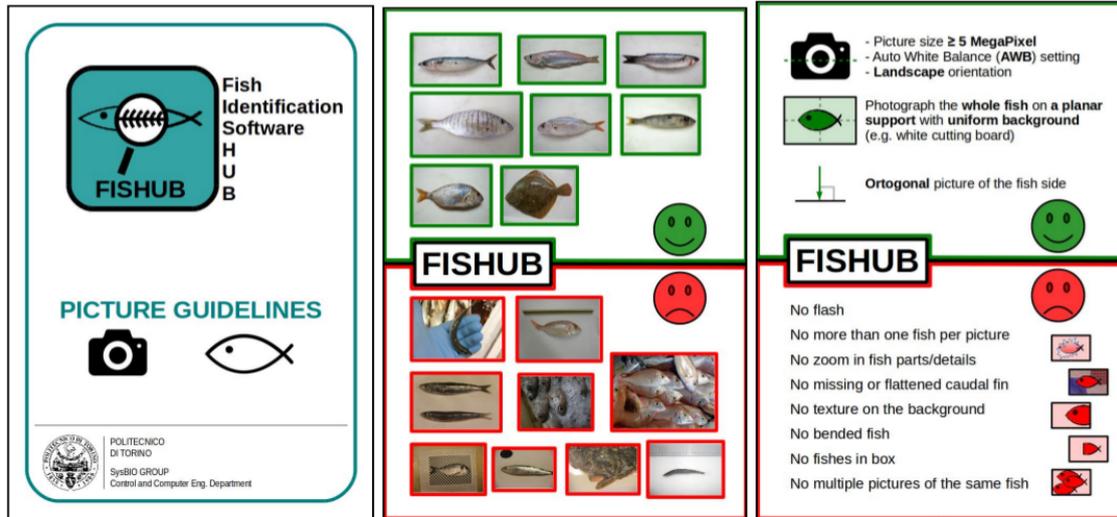


Figure 3.15: F.I.S.HUB picture guideline

Pictures have been taken in Italy and the United Kingdom, involving an Italian food company (Esselunga) and project partner personnel, adopting the previously standardized image-capture protocol. The list of the species and number of pictures/species are shared among all collaborators involved in the collection. These two countries have been selected to cover species with intra and inter-specific variability that could be attributed to the area/stock of origin of the fish.

A **cloud storage service** has been set up and is actually constantly maintained. It is used to store and share all the documents (list of species, guidelines ...) and photos. Images and files can be uploaded, renamed and modified easily. The cloud provides access to data through a web interface, and it allows versioning for handling modified or deleted files. The F.I.S.HUB database, connected to the cloud service, uses a data virtualization technology called RAID (Redundant Array of Inexpensive Disks), that combines multiple physical disk components into a single logical unit for data redundancy. The cloud service allows simultaneous access to multiple users as described in Figure 3.16.

The F.I.S.HUB database is scalable in order to be continuously expanded with new fish families and species, and is not limited to the ones chosen as the primary targets for the project. The structure of the DB follows the list of species defined above. There is one directory for each order, that contains a subdirectory for each family. The latter

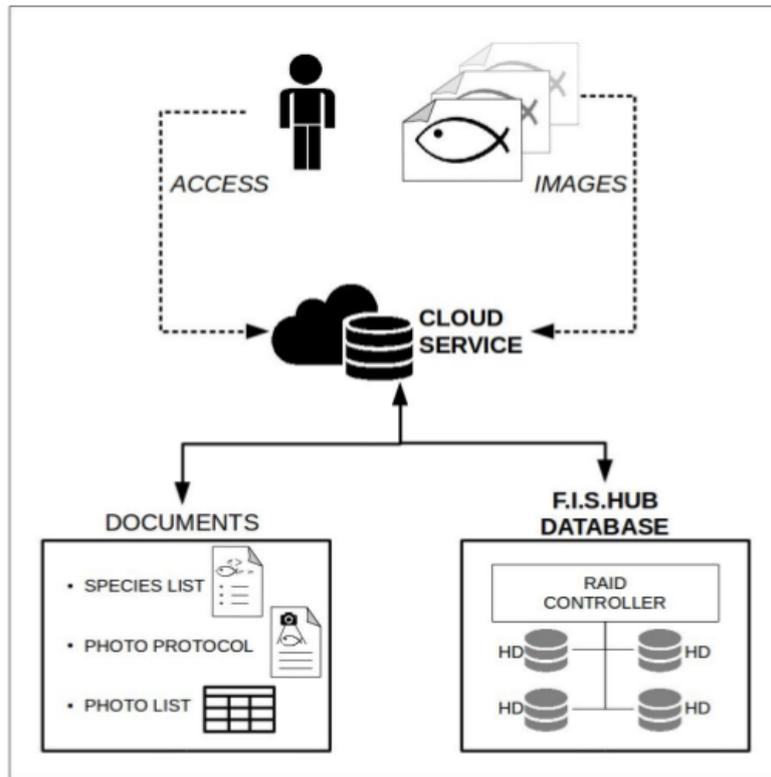


Figure 3.16: F.I.S.HUB mass storage system

contains a folder for every related genus followed by the species. The aforementioned structure is depicted on the following Figure 3.17 for *Clupeidae*:

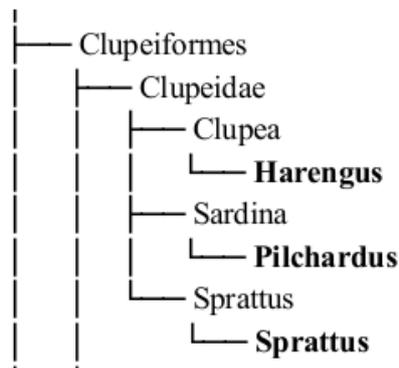


Figure 3.17: DB hierarchical organization

Each collection has its own Photo List file that is updated every time new pictures are uploaded and stores the total number of pictures available in the database. The F.I.S.HUB quality control group periodically checks if the newly uploaded pictures are

saved in the right database and belong to the right species directory. In case of uncertainty or mistake, photographs are manually inspected and subsequently re-saved in the correct location.

The following Table 3.2 summarizes the number of pictures uploaded in the Cloud Database for each species (as of Jan. 2018).

ORDER	FAMILY	GENUS	SPECIES	Number of Images
Clupeiformes	Clupeidae	Clupea	harengus	119
Clupeiformes	Clupeidae	Sardina	pilchardus	586
Clupeiformes	Clupeidae	Sprattus	sprattus	585
Clupeiformes	Engraulidae	Engraulis	encrasicolus	547
Gadiformes	Gadidae	Gadus	morhua	63
Gadiformes	Gadidae	Melanogrammus	aeglefinus	463
Gadiformes	Gadidae	Merlangius	merlangus	277
Gadiformes	Gadidae	Pollachius	virens	103
Gadiformes	Merluccidae	Merluccius	merluccius	321
Perciformes	Sparidae	Dentex	dentex	14
Perciformes	Sparidae	Dentex	gibbosus	128
Perciformes	Sparidae	Diplodus	annularis	507
Perciformes	Sparidae	Pagellus	acarne	238
Perciformes	Sparidae	Pagellus	bogaraveo	71
Perciformes	Sparidae	Pagellus	erythrinus	495
Perciformes	Sparidae	Pagrus	caeruleostictus	144
Perciformes	Sparidae	Pagrus	pagrus	111
Pleuronectiformes	Pleuronectidae	Hippoglossus	hippoglossus	39
Pleuronectiformes	Pleuronectidae	Limanda	limanda	281
Pleuronectiformes	Pleuronectidae	Microstomus	kitt	483
Pleuronectiformes	Pleuronectidae	Pleuronectes	platessa	504
Pleuronectiformes	Pleuronectidae	Reinhardtius	hippoglossoides	12
Pleuronectiformes	Scophthalmidae	Psetta	maxima	425
Pleuronectiformes	Scophthalmidae	Scophthalmus	rhombus	29
Pleuronectiformes	Soleidae	Solea	vulgaris	703
<b>TOTAL</b>				<b>7248</b>

Table 3.2: F.I.S.HUB list and number of images for each fish species

The total number of pictures in the DB is 7248. For each Order, the number of images is reported in Table 3.3.

### 3.5.3 Server Classifier

The F.I.S.HUB software has been designed with an architecture able to guarantee a seamless communication among the classifier, the F.I.S.HUB database and the mobile App (which is the main user access point to the framework classifier). Figure 3.18 shows a high-level view of the F.I.S.HUB global software architecture.

ORDER	PICTURES
Clupeiformes	1848
Gadiformes	1245
Perciformes	1723
Pleuronectiformes	2513

Table 3.3: Images for each Order in the DB

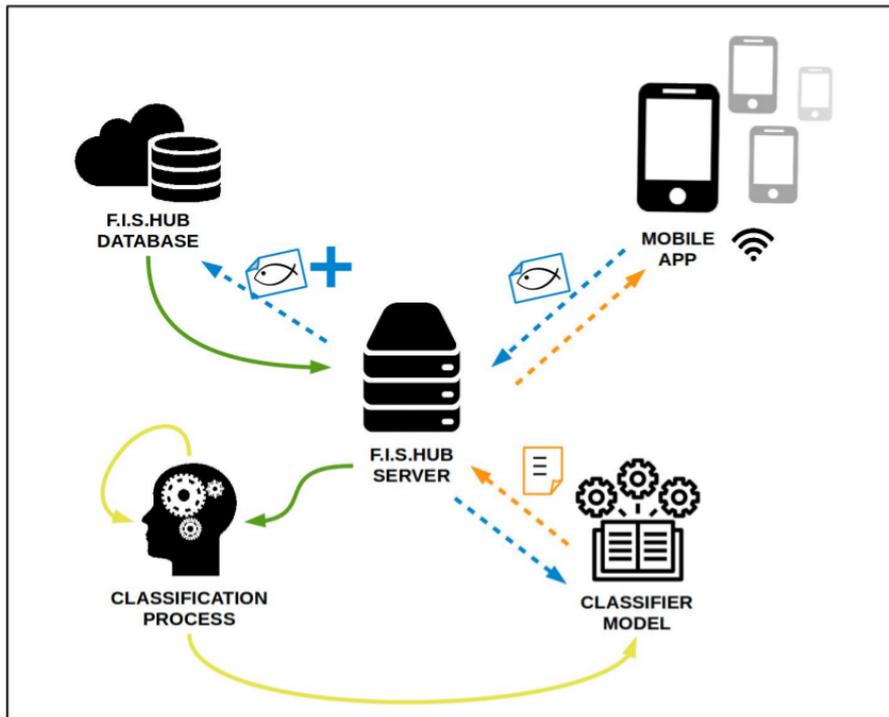


Figure 3.18: F.I.S.HUB global scheme

The mobile App is connected to a remote server that performs the classification through the model classifier generated by the classification process. The App exploits the smartphone camera as a sensor to capture an image of the fish to be analyzed. The picture is sent by the App to the server where the classifier model processes the picture that gives back the classification output. The model can be constantly updated every time the classification process takes place. The F.I.S.HUB cloud database, that contains the fish pictures, is connected to the server and it is used by the classification process during the training procedure. The F.I.S.HUB server is in charge of all the computational efforts related to classification process and to the training model procedure. The capability of the server is able to guarantee multiple and simultaneous classifier predictions from different mobile devices.

Whenever a new fish image is received by the remote server to carry out a classification instance the new picture is stored in a dedicated location in the database. All the pictures received can be constantly validated by expert and assigned to a specific species for further elaboration and analysis or discarded because of the uncertainty. In future the more the picture is sent to the server the more F.I.S.HUB database could increase.

### 3.5.4 Method and Classifiers

In section 3.3 is described a procedure able to extract few geometrical features from fish images. Eventually, the features extracted were used by an Artificial Neural Network (ANN) to perform the classification.

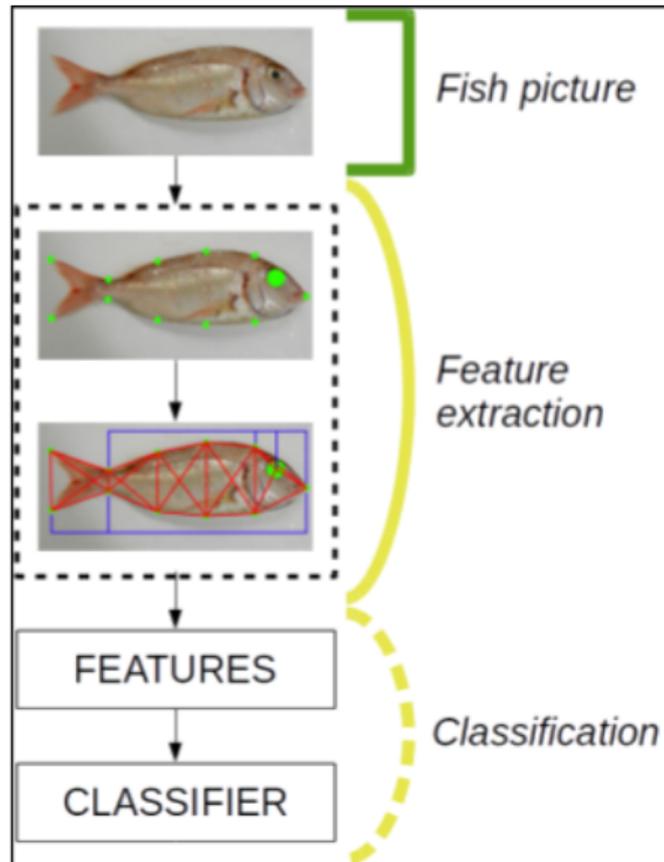


Figure 3.19: Feature engineering and classification

The implemented pipeline, showed in Figure 3.19, performed very well in the limited picture domain used in that study. Anyway, the execution does not succeed with the new and different fish species available in the F.I.S.HUB cloud database. The weakness is that the ad-hoc engineered feature extraction should be modified or adapted to each

fish order or family because the morphological differences cause failure. In fact, it is not possible to generalize this approach to every fish species in a reliable way, even if this method does not need many pictures per species to reach good results. One of the main limitations is that these kinds of high-level engineered procedures need severe conditions to work correctly (illumination, orientation, scale factor, orientation, ..) that cannot be guaranteed in a real setup application. Lastly, it is important to underline that a so rigid procedure may affect negatively the geometrical features if only some key-points are inaccurate or the foreground fish segmentation fails since it is not always reliable in some circumstances.

Actually, the cutting-edge technology for such a problem in computer vision are the CNNs. These models exploit the strong spatially local correlation present in natural images and achieve better generalization on vision problems. The principal aspect of a CNN is that it can learn itself the feature from images adopted during the training phase. This significant peculiarity is shown in Figure 3.20.

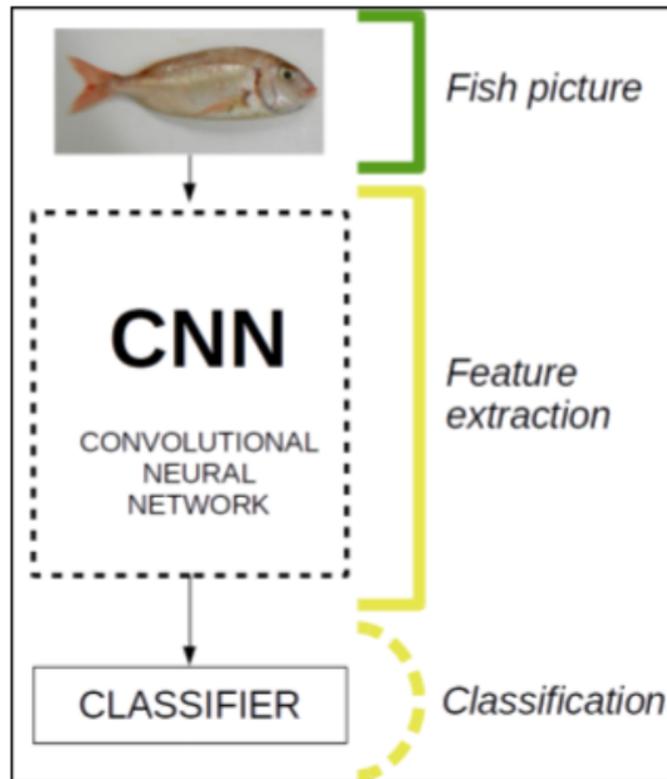


Figure 3.20: CNN feature extraction and classification

The scheme is the same as the Figure 3.19:

1. Fish pictures
2. Feature extraction

### 3. Classification

The main difference concerns the step that figures out the features of the fish. With CNNs we do not need to design or fine-tune ad-hoc feature extraction for every fish species. The paradigm for this kind of image recognition tasks has changed and it is justified by a data driven approach. Nowadays, having a big quantity of picture is a must for training a DL classifier from scratch. This is the reason why the F.I.S.HUB project is focusing its efforts to increase the number of pictures in the database. To improve and increase the training set for the classifier the data augmentation is a common routine within the pre-training step adopted in almost all DL framework.

For the development of the F.I.S.HUB classifier it has been adopted the Caffe Deep Learning Framework [32] and it has been used the NVIDIA® Tesla K20X GPU accelerator<sup>8</sup>.

#### 3.5.5 Deep Model Zoo

Firstly, we decide to implement two different CNNs models. We adopted the two most commonly used models for image recognition taken from the Caffe Model Zoo<sup>9</sup>. In this repository are available some of the most known models for DL applied to every kind of topic.

The first CNN model is the AlexNet [35] which completed the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. AlexNet contains eight layers; the first five are convolutional layers, and the last three are fully connected layers (see Figure 3.21).

The second CNN model is the GoogleLeNet [65] that won the ILSVRC 2014 challenge. The network uses *Inception* modules to reduce the parameters and improve the utilization of the computing resources inside the network. GoogleLeNet is mainly composed of 22 layers (see Figure 3.22).

To test the performance of the two CNN models, we execute for each CNN four different training. The first training (*Pretrained*) performed is a complete transfer learning from the pre-trained model (based on ImageNet database) except the last fully connected layers at the end of the network. In fact, since the number of classes for the ImageNet dataset is more than 1000, while in our database are only 25 fish species, this setting was obligated. The second experiment (*Pretrained-1*) was performed by using the pre-trained model until the last layer node of convolution. In AlexNet it means that only the last layer weight has been reset, while in GoogleLeNet all the last inception modules has been reset. In the third experiment (*Pretrained-2*), similar to the second one, we applied the pre-trained model until the second last convolution node. Finally, in the last experiment (*From scratch*), we trained the two models from scratch. This means that in

---

<sup>8</sup>NVIDIA® Tesla K20X GPU accelerator - Board Specification: [link](#)

<sup>9</sup>Caffe Model Zoo [link](#)

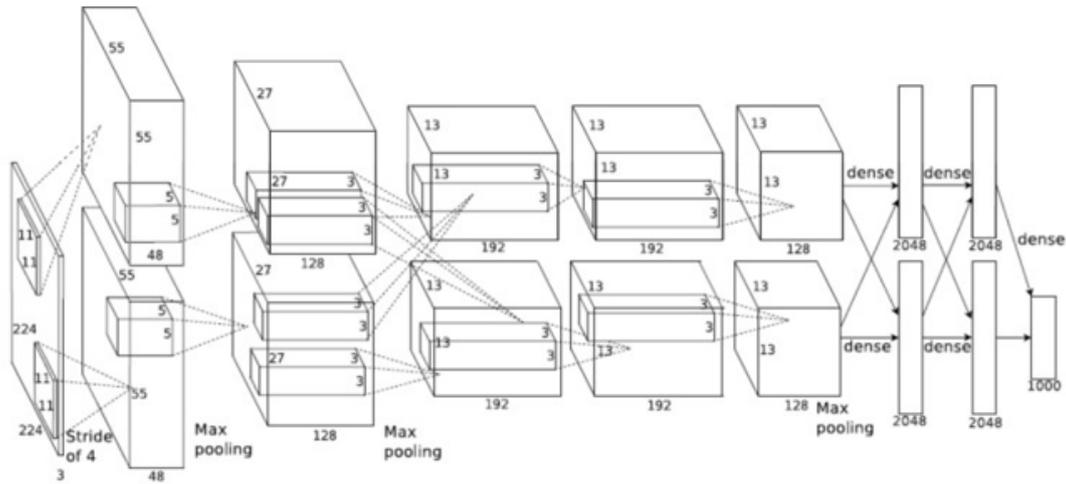


Figure 3.21: AlexNet architecture

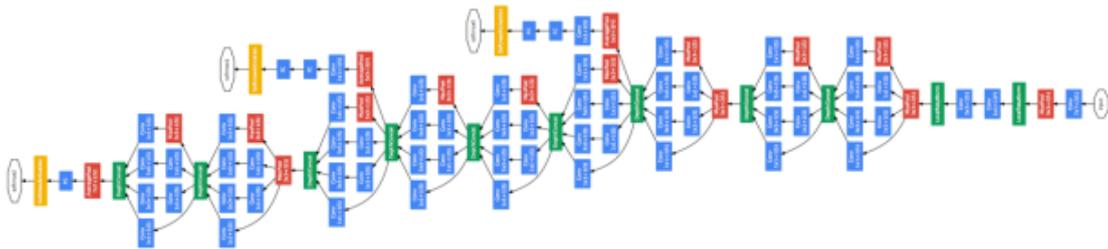


Figure 3.22: GoogleLeNet architecture

no way the transfer learning has been used. In this last case, the architecture models are completely reset, for this reason, the training time increased.

In these experiments, we set to zero the learning rate to the CNN pre-trained layers. This means that the training time is highly reduced and that the network can apply totally or partially the feature extraction capability acquired from the original models used for transferring knowledge (and original dataset too).

The dataset used during training is the F.I.S.HUB database where each image has been multiplied 6 times, performing flipping and random brightness, contrast, and rotation. Every experiment has been performed to scaled images at 256x256 pixels, both in color mode and grayscale.

All the trainings were designed with Stochastic Gradient Descend (SGD) solver type, base learning rate equal to 0.01 with step down policy (step down with size 33% and Gamma equal to 0.1).

Other solver algorithms have been tested, such as Nesterov’s Accelerated Gradient

(NAG), Adaptive Gradient (AdaGrad), RMSprop, AdaDelta, and Adaptive Moment Estimation (Adam) [56]. Eventually, SGD was selected as the best time-accuracy solution.

### 3.5.6 Results

Results obtained from training are presented in the following Tables 3.4. Moreover, in Figure 3.23 are reported the bar plot to visualize the experimental results together.

Table 3.4: Training results for AlexNet and GoogleLeNet models

AlexNet			GoogleLeNet		
Accuracy [%]	gray	color	Accuracy [%]	gray	color
<b>Pretrained</b>	74	84	<b>Pretrained</b>	80	87
<b>Pretrained-1</b>	77	88	<b>Pretrained-1</b>	81	90
<b>Pretrained-2</b>	78	89	<b>Pretrained-2</b>	81	92
<b>From Scratch</b>	69	80	<b>From Scratch</b>	75	78

The global accuracy has been calculated over a validation set of 10% taken from the augmented data set. Looking at the accuracy value we can observe that, in general, GoogleLeNet better perform than AlexNet model. This main outcome is true both for grayscale and color images. As expected in every experiment the colored images have reached a higher accuracy value. The reason is that RGB images contain more information that is discriminative among all the fish species to perform classification.

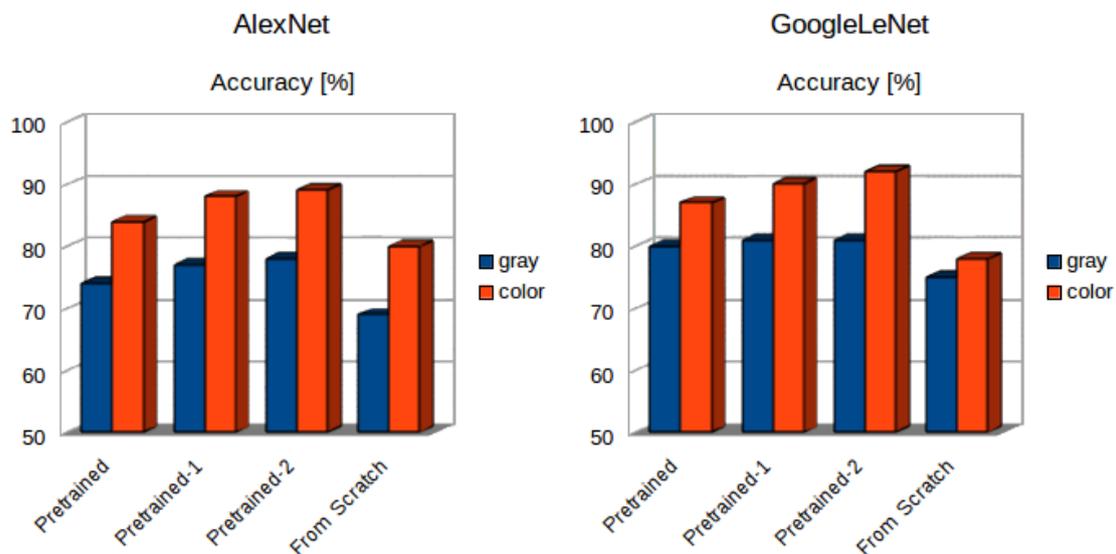


Figure 3.23: Barplot of training results for AlexNet and GoogleLeNet models

We can observe that from *Pretrained* to *Pretrained-2* the accuracy never decreases. This indicates that the more we re-train the last layers the more is accurate the classification. In fact, since the models have been already trained over the ImageNet dataset, probably the deepest features abstraction are not discriminative for our task. Finally, we can prove that transfer learning is effective.

Anyway, the most important result concerns the accuracy value between pretrained and scratch experiments. The *From scratch* training are the worst ones. The reason is that, with respect to the other results, the network does not have enough data to be correctly trained. Even if the automatic features extraction is performed completely on the F.I.S.HUB dataset this not guaranteed to reach high accuracy. This is a further demonstration that for the small dataset (where “small” refers to DL standard dataset) the transfer learning should be applied.

The higher accuracy has been reached with RGB images by the GoogleLeNet *Pretrained-2* with the values of 92%. In Table 3.5 are presented the accuracy values for each species in this experiment.

According to Table 3.2 the higher accuracy is related to those fish species with more images. This is coherent with the fact the DL supervised models learn from data and consequently are more capable to discriminate the species that have more image examples.

In general, these results are not so satisfactory since a lot of species are under an acceptable accuracy value (i.e. harengus, morhua, dentex, gibbosus, bogaraveo, and hippoglossoides are under 85% of accuracy).

On the other hand, for some species, the accuracy is very high (e.g. pilchardus, encrasicolus, kitt, platessa, maxima, and vulgaris that reached 95% of accuracy or more).

In Table 3.5 the global accuracy is grouped for fish Order. A very interesting result has been obtained for Clupeiformes and Pleuronectiformes whereas for Gadiformes and Perciformes the performance is under 90%.

Observing these results we are fully aware that this DL classifier did not achieve an excellent global accuracy. Anyway, as also mentioned in 3.3.2 the mobile application, and so the server classifier model, should have to answer to this question: *Is really this the species declared on the label or not?* But, if we consider the CNN architectures used above, we cannot query the classifier to analyze an image of a fish species never seen before. In fact, the models described before always give back an answer identifying the supposed fish species as the output result with the higher value. From a different point of view, this architecture is more or less “blind”. Consider the *Plinko* game in Figure 3.24), where the red ball represents the fish image, the internal obstacle is the CNN layers, and the final container is the fish species label. The ball (i.e. input image) will flow for sure into a container (i.e. class label). There is no way to “open” the game and let the ball going outside. In other words, the net generates as an output the result that the CNN produces, but it is not able to exclude or reject the fish image even if that species is not included in the database.

This solution is described in the next section.

ORDER	FAMILY	GENUS	SPECIES	ACCURACY [%]
Clupeiformes	Clupeidae	Clupea	harengus	82
Clupeiformes	Clupeidae	Sardina	pilchardus	95
Clupeiformes	Clupeidae	Sprattus	sprattus	92
Clupeiformes	Engraulidae	Engraulis	encrasicolus	96
Gadiformes	Gadidae	Gadus	morhua	80
Gadiformes	Gadidae	Melanogrammus	aeglefinus	89
Gadiformes	Gadidae	Merlangius	merlangus	85
Gadiformes	Gadidae	Pollachius	virens	88
Gadiformes	Merluccidae	Merluccius	merluccius	89
Perciformes	Sparidae	Dentex	dentex	66
Perciformes	Sparidae	Dentex	gibbosus	72
Perciformes	Sparidae	Diplodus	annularis	94
Perciformes	Sparidae	Pagellus	acarne	92
Perciformes	Sparidae	Pagellus	bogaraveo	83
Perciformes	Sparidae	Pagellus	erythrinus	92
Perciformes	Sparidae	Pagrus	caeruleostictus	85
Perciformes	Sparidae	Pagrus	pagrus	78
Pleuronectiformes	Pleuronectidae	Hippoglossus	hippoglossus	92
Pleuronectiformes	Pleuronectidae	Limanda	limanda	94
Pleuronectiformes	Pleuronectidae	Microstomus	kitt	95
Pleuronectiformes	Pleuronectidae	Pleuronectes	platessa	97
Pleuronectiformes	Pleuronectidae	Reinhardtius	hippoglossoides	76
Pleuronectiformes	Scophthalmidae	Psetta	maxima	96
Pleuronectiformes	Scophthalmidae	Scophthalmus	rhombus	91
Pleuronectiformes	Soleidae	Solea	vulgaris	96

Table 3.5: FISHUB accuracy trained on GoogleLeNet model pretrained on ImageNet dataset until the second last inception node

Table 3.6: Table 3.5 grouped per fish Order

ORDER	ACCURACY [%]
Clupeiformes	93
Gadiformes	87
Perciformes	89
Pleuronectiformes	96

### 3.6 Standard Classification v.s. One-Shot

The F.I.S.HUB classifier has not been designed to perform a standard fish discriminative classification (e.g. “What is this fish?”). The main purpose is to provide users to

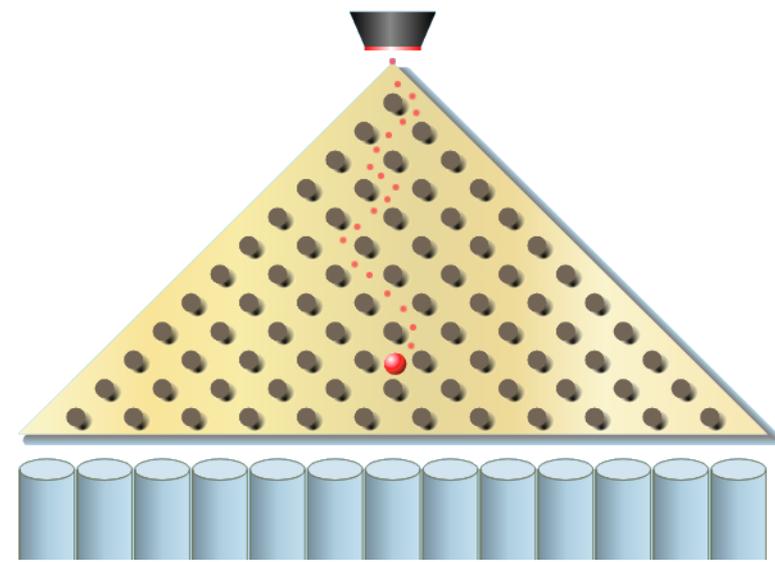


Figure 3.24: The game mode of Pinko

verify if a fish is the one declared in the label (e.g. “Does this fish is a turbot?”). Discriminative approaches require that all the categories be known in advance. In addition, these approaches also require that the training samples are available for all categories and they cannot guarantee a priori to work correctly for all of them when adding a new one.

A common methodology to such kind of problems are distance-based methods, also called embedding [21] [42]. This involves computing a similarity metric between the pattern to be classified/verified and a library of stored prototypes (i.e. database). It is the same setup when the user wants to verify the fish species with respect to his knowledge. Anyway, in the F.I.S.HUB specific domain is be possible to perform a deeper and widespread similarity metric according to the species prototype available in the database. One of the machine learning architecture able to perform this method is called Siamese Network (SN) [11] [28] [13].

SNs are a class of architecture that comprises two identical CNN with weight sharing ( $W$ ) and one cost module layer at the end. The input to the network is a pair of fish images and two labels ( $X_1$  and  $X_2$ ), that is the species definition, one for each picture. The images are passed through the sub-networks that generate two outputs,  $D(X_1)$  and  $D(X_2)$ , which are then passed to the contrastive loss function at the top of the network. This will cause the network to try and push together images of the same class and pull apart images from different classes. The L2 distance between extracted features is used to measure the distance between images. If the images belong to the same class the metric should to be 0, while the distance between images of different classes is greater than a determined value. In fact, the contrastive loss function employed to learn the parameters of a parameterized function (e.g.  $D(X_1)$  and  $D(X_2)$ ). It operates in order that

neighbours are pulled together and non-neighbours are pushed apart. Prior knowledge can be used to identify the neighbours for each training data point [24]. In Figure 3.25 is represented a scheme of SN.

The pre-training protocol generates a list of coupled images taken from the F.I.S.HUB database, some of them belong to the same class (e.g. null distance) while others belong to different species (e.g. distanced). When labels of the two images are the same, then the SN tries to pull together the images. Otherwise, if the coupled images contain different species, then the SN operates vice-versa.

The SN model is widely used to be applied in the so-called “one-shot learning”. This method wants to represent the same as we do when we learn to discriminate or recognize whatever. We are able to describe an object and to expand how knowledge every time we made the experience. The associative/dissociative operation is the way we are able to identify a specific class even if we have a poor experience with it. For this reason, SNs are used to map the features of a certain problem when the available dataset is not too much wide.

In a multi-class setup, like in F.I.S.HUB, the finale training pairwise image dataset is larger than the original one due to the couple image generation that allows using the same image multiple times. The same fish species image can be trained against many other images with the same class or belonging to different species. Despite this, overfitting should be always avoided.

The output result of a SN training is the net itself. In fact, once the training set has been analyzed, then the SN can be executed only through one of its branches to generate the features. The weight sharing, in fact, produces two identical CNN parameterization. The features computed on the training set are then analyzed to perform a-posteriori clustering among the class. This is the features representation that SN is able to generate during deployments. Instead of learning how to define the class of an image (e.g. standard classification) the SN generates the features used to represent that image with respect to the other ones. The number of the features that the SN is able to produce is an integer number freely settable. The same also for the contrastive loss layer margin, which is settled and defined in the network architecture design.

The models trained with the F.I.S.HUB database are described in the next subsection.

### 3.6.1 F.I.S.HUB Embedding setup

First of all, we test a Siamese Network on the F.I.S.HUB database. The two branches of the net are two GoogleLeNet CNN taken from the previous experimental setup, in order to reduce the training time. The database is composed of RGB images scaled at 256x256 pixels. Each image in the database has been coupled five times: one with an image of the same class e the other four with random images derived from other species. Also, in this case, data augmentation is performed the same as in the standard CNN setup. The resulting dataset is then composed of 217440 coupled images (that is 7248 multiplied by 6 times due to augmentation and then by 5 times due to coupling siamese

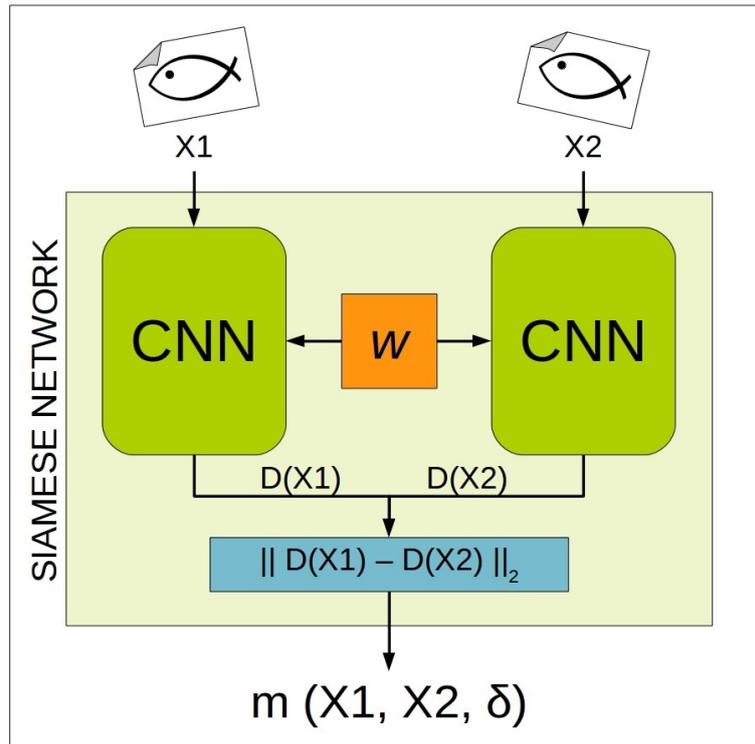


Figure 3.25: F.I.S.HUB Siamese Network architecture

dataset creation).

The output features vector of each CNN has dimension 10. The margin of the loss layer has been set to 1 (i.e. images from different class are represented by a distance the more greater to 1 the more they are featured differently). All the other training parameters are the same as in 3.5.5.

Because of computational cost, this experiment has been done on an Amazon EC2 instance with two NVIDIA® Tesla K20X GPU accelerator.

### 3.6.2 Results

In the following Figure 3.26 we can see how the model has learnt to separate fish classes into clusters. The figure is obtained with the t-Distributed Stochastic Neighbor Embedding (t-SNE) technique [40].

In this figure, it is evident the capability to represent the different fish species. In particular in Figure 3.27 is highlighted the separation between flatfish species and non-flat ones.

Another representation of the training concerning the Order clusters is shown in Figure 3.28.

As we can see the from t-SNE scatter plots the SN seems to be able to embed together

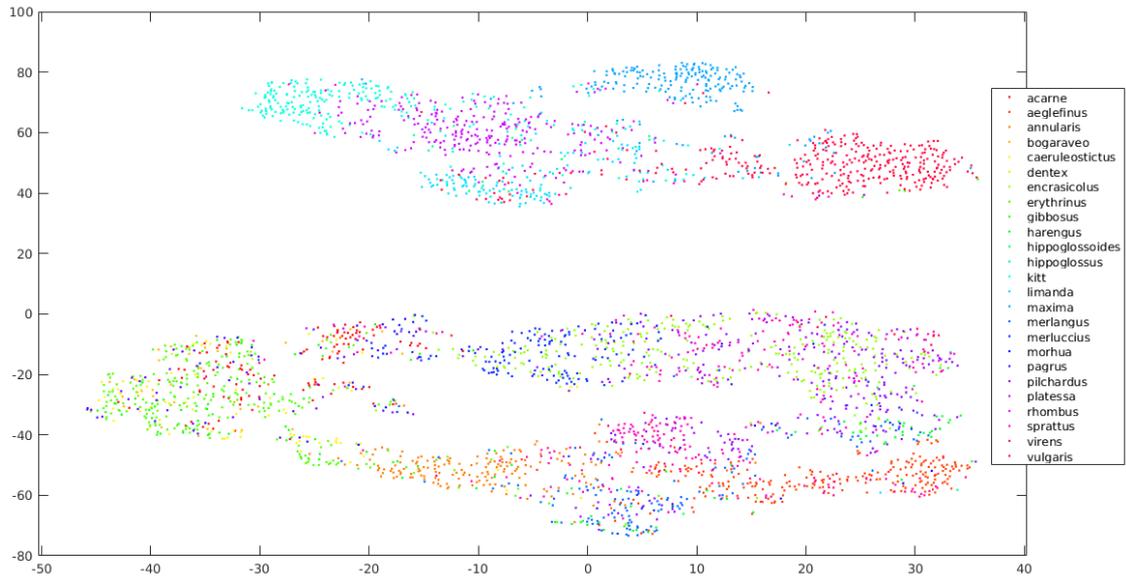


Figure 3.26: t-SNE representation of SN training result over F.I.S.HUB specie

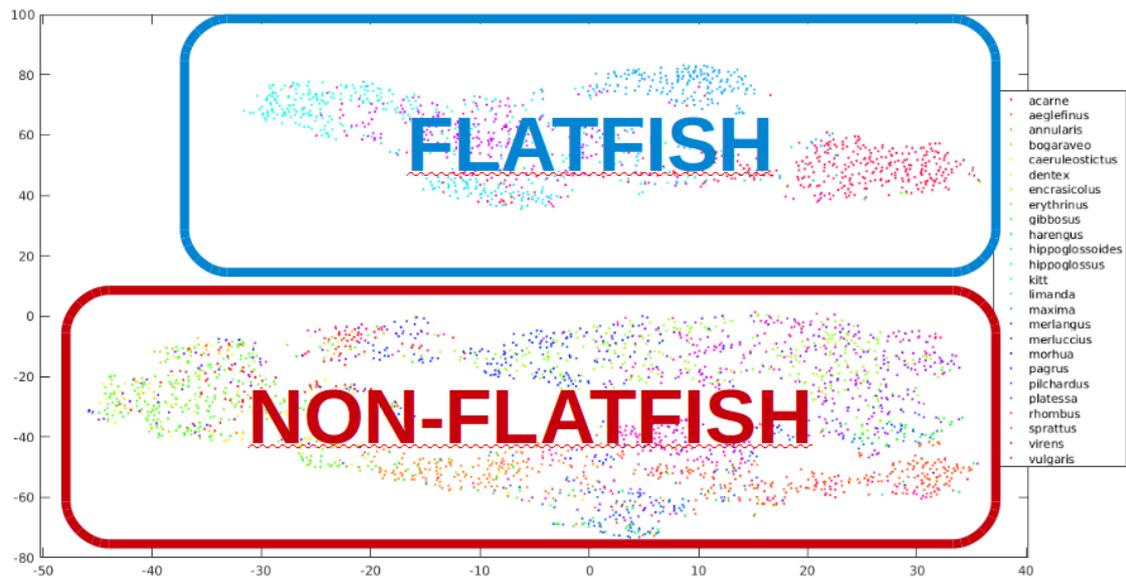


Figure 3.27: The same as in Figure 3.26 with the highlight of flat/non-flat fish graphic separation

the fish belonging to the same species and to create a distance able to discriminate different species.

To measure the accuracy of the SN training, the featured images have been analyzed with the k-nearest neighbours algorithm (kNN). This is a non-parametric method used for classification and regression [2]. In our case, we used kNN for regression since we

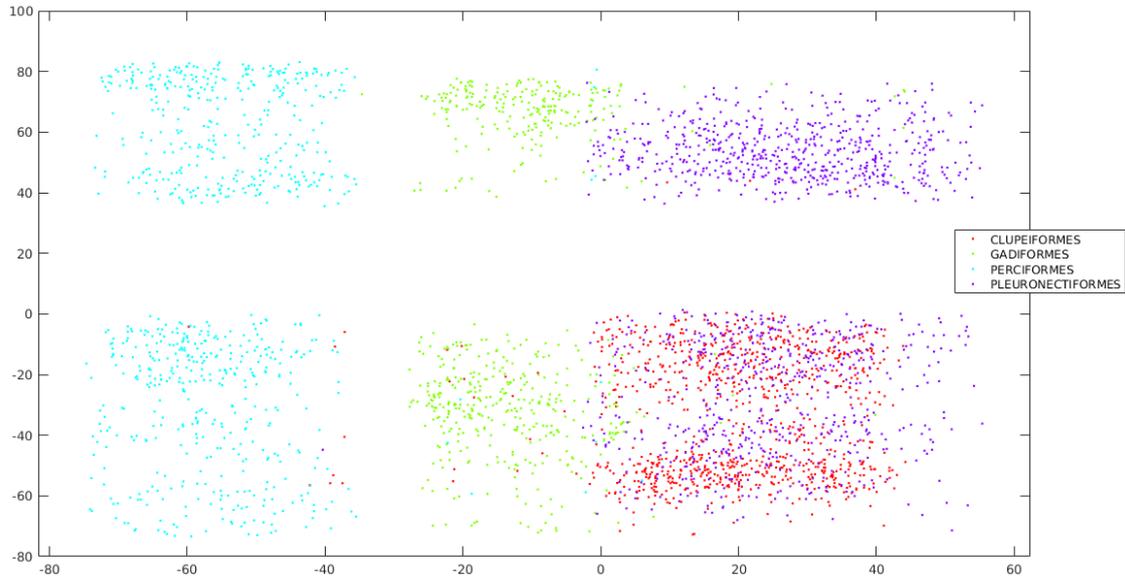


Figure 3.28: t-SNE representation of SN training result over F.I.S.HUB Order

want to produce an output able to measure the membership of an image to a precise fish species class. The output of the kNN is a value that predicts with continuous values the average of the values of its k nearest neighbours.

Running the kNN over the SN feature map representation, we obtain the following result in Table 3.7.

ORDER	FAMILY	GENUS	SPECIES	ACCURACY [%]
Clupeiformes	Clupeidae	Clupea	harengus	89
Clupeiformes	Clupeidae	Sardina	pilchardus	96
Clupeiformes	Clupeidae	Sprattus	sprattus	94
Clupeiformes	Engraulidae	Engraulis	encrasicolus	96
Gadiformes	Gadidae	Gadus	morhua	89
Gadiformes	Gadidae	Melanogrammus	aeglefinus	91
Gadiformes	Gadidae	Merlangius	merlangus	89
Gadiformes	Gadidae	Pollachius	virens	90
Gadiformes	Merluccidae	Merluccius	merluccius	95
Perciformes	Sparidae	Dentex	dentex	86
Perciformes	Sparidae	Dentex	gibbosus	87
Perciformes	Sparidae	Diplodus	annularis	96
Perciformes	Sparidae	Pagellus	acarne	93
Perciformes	Sparidae	Pagellus	bogaraveo	90
Perciformes	Sparidae	Pagellus	erythrinus	94
Perciformes	Sparidae	Pagrus	caeruleostictus	89
Perciformes	Sparidae	Pagrus	pagrus	88
Pleuronectiformes	Pleuronectidae	Hippoglossus	hippoglossus	93
Pleuronectiformes	Pleuronectidae	Limanda	limanda	95
Pleuronectiformes	Pleuronectidae	Microstomus	kitt	96
Pleuronectiformes	Pleuronectidae	Pleuronectes	platessa	96
Pleuronectiformes	Pleuronectidae	Reinhardtius	hippoglossoides	89
Pleuronectiformes	Scophthalmidae	Psetta	maxima	97
Pleuronectiformes	Scophthalmidae	Scophthalmus	rhombus	94
Pleuronectiformes	Soleidae	Solea	vulgaris	95

Table 3.7: F.I.S.HUB Siamese Network species accuracy

The accuracy grouped for fish Order is shown in Table 3.8.

Table 3.8: Table 3.7 grouped per fish Order

ORDER	ACCURACY [%]
Clupeiformes	95
Gadiformes	91
Perciformes	93
Pleuronectiformes	96

The global accuracy calculated over all the 25 fish species is 94

## 3.7 Endless Learning

In the next Figures 3.29 and 3.30 there are some examples of the distance-base value that creates the cluster of picture with the same species. This is the demonstration that we are now able with the kNN trained to perform regression to define a probabilistic value that a picture belong to a specific species.



Figure 3.29: Distance-based results for same species

On the other hand, we can also outline that is possible to reject a species with a low returned probability and return a score that indicate when a fish image and a specific label are not coherent, thus to avoid the substitution.

This approach has been also very effective for those species with a less number of pictures. This demonstrates that SN can correctly perform the “one-shot learning” and that it can analyze every fish species, even if it is not available in the DB. In the future, if more pictures or species will be added in the F.I.S.HUB database, this architecture can be easily re-trained to enlarge its features representation capability.

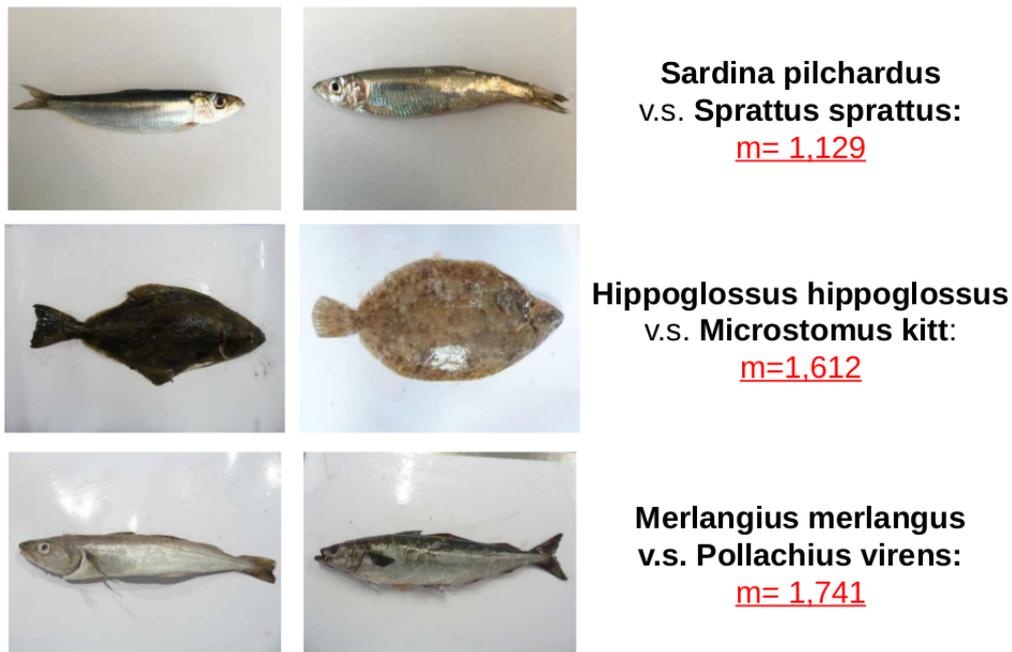


Figure 3.30: Distance-based results for different species

### 3.8 The “F.I.S.HUB” Mobile Application

In this section we describe the structure of the F.I.S.HUB App that will allow users to use their mobile devices (IOS and Android) to take pictures of whole fish samples and submit them to the FisHUB classification server. The F.I.S.HUB App is just a mobile interface to the FisHUB database and classification server.

The F.I.S.HUB App actually allows to take or select a picture (from the phone photo album), to Select the corresponding species, send them to the F.I.S.HUB classification server, and finally to display the classification results. The App also allows to browse the species covered by the F.I.S.HUB project and, for each of them, display a representative photo and its main morphological characteristics.

On the server side, the App stores all submitted images in order to create a pool of “crowd sourced” images that, after a careful review can be included in the FisHUB database in order to increase its size and, eventually, the classifier performances. The F.I.S.HUB App will be further available in the Apple and Google Play Stores. The app implements two main functionalities:

1. The validation of a fish photo against a declared label (species)
2. A simplified browsing of the species included in the FisHUB database

Figure 3.31 shows the organization of the two functionalities implemented by the F.I.S.HUB App. The App does not require any kind of authentication. The access to the F.I.S.HUB

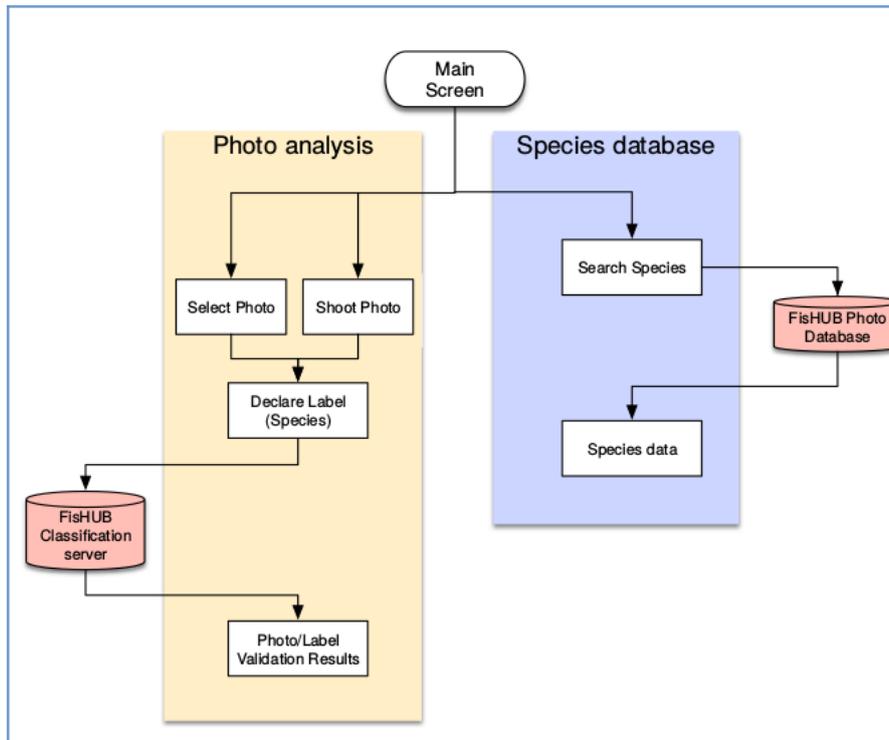


Figure 3.31: App storyline and functionalities

database and classification server is anonymous (only the IP addresses of the device running a photo analysis flow are logged for security purposes).

As soon as the App is ready it immediately allows the user to start with a fish identification/analysis flow (Figure 3.32). Before submitting something to the F.I.S.HUB server, the user has to complete two tasks:

1. Select the image source by clicking one of the two round buttons
2. Select the species that the user is submitting (typically the species that is reported on the mandatory label in the marketplace) by clicking the “Select Species” button.

The two operations can be completed in any order, but the submission to the server is not possible until both tasks are completed. If the user selects the mobile phone camera, the default operating system camera tool allows the user to take a square picture (the square frame is consistent with the shape of the pictures in the F.I.S.HUB database and cannot be changed). Alternatively, the user can select a previously taken picture from the mobile phone Camera Roll. After this phase the two buttons are replaced by a thumbnail of the selected picture (Figure 3.33).

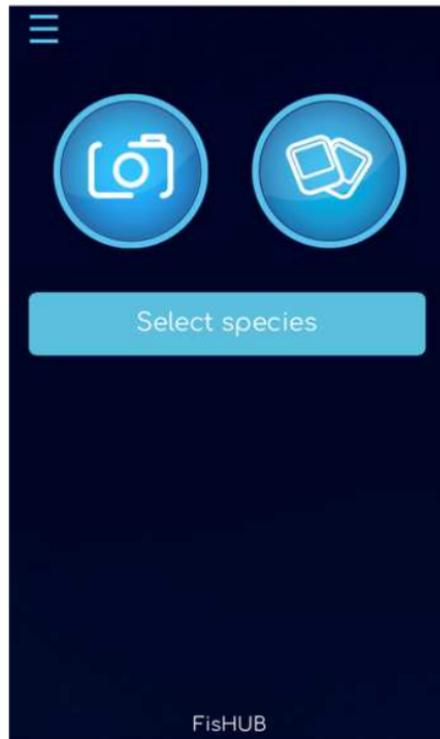


Figure 3.32: App initial screen

The second required step is the selection of the fish species. After clicking the “Select species” a popup appears where the user can search and then select the desired label (Figure 3.34).

After selecting a label, the “Analyze” button appears. By clicking it the image and the label are sent to the F.I.S.HUB classification server for validation. As soon as the result is received from the server (a few seconds in the worst case), the results of the validation process are displayed (Figure 3.35).

Four important information are displayed on this page:

- The original label (in the blue rounded box)
- The probability that the analysed picture actually matches the species declared by the label. This information is reported inside a green circle if it is > 80%, in a yellow circle otherwise
- A reference picture of the declared label, taken from the F.I.S.HUB database
- A set of icons that, when clicked, explain the particular morphological characteristics of the selected species

The App also offers the user the possibility of browsing the F.I.S.HUB database in order to learn morphological information about the species present in the F.I.S.HUB

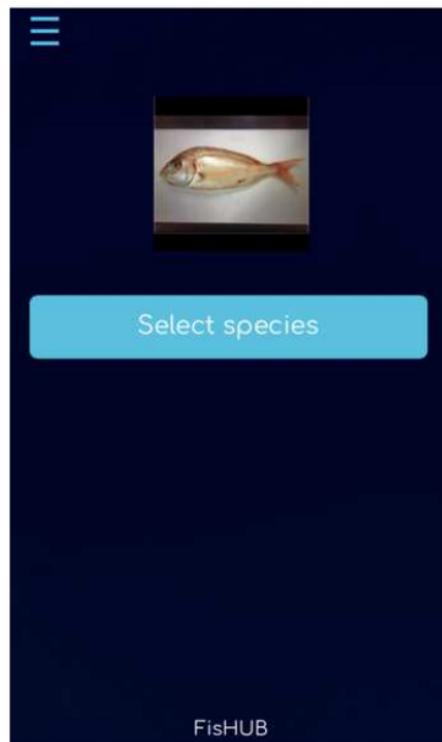


Figure 3.33: Home screen after selecting a photo

photo database. To access this functionality it is necessary to click on the “Guide” link accessible in the App menu accessible by clicking the hamburger button present in the top-left corner of all App pages. The “Guide” link takes the user to a page where it is possible to scroll through the list of all the species that the F.I.S.HUB server has been trained to recognize. By clicking on a species name, the App will query the server and return to the user a picture of a representative fish (taken from the database) and, if available, some particular morphological characteristics of the species itself, just like in the Results page.

At the moment the Hamburger button allows five possible choices:

- Home: it takes the user to the initial page where to start a new analysis
- Legend: it takes to a static page explaining the meaning of the icons that describe, in the Guide and Results pages, the morphological characteristics of each species
- Results: it takes the user to the results of the last analyzed image (if available)
- Guide: it allows to browse the F.I.S.HUB photo database
- Settings: it allows to select the app language (the available languages are English and Italian)

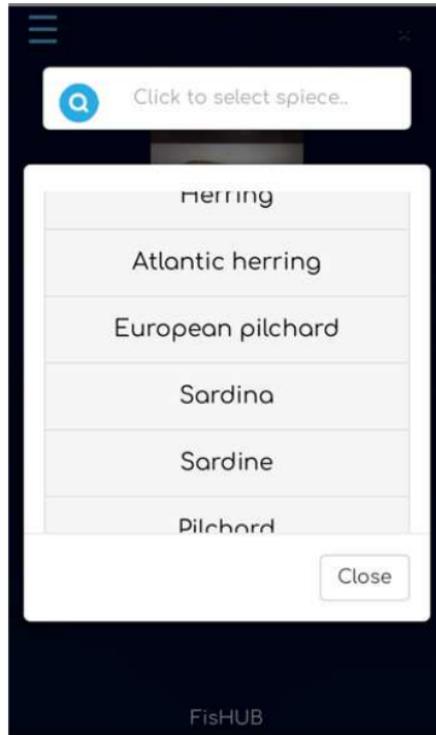


Figure 3.34: Species selection



Figure 3.35: Results page

The F.I.S.HUB app was designed using standard html/javascript/ajax technology on the client side, and Php scripts on the server side to interface with the F.I.S.HUB photo database and classification server. On the client side (App) Bootstrap 4.0 has been chosen as the main front-end component library. To create the App the code was wrapped into

Adobe PhoneGap, a powerful framework that allows to create cross-platform apps using HTML, CSS and Javascript. In particular, the App was generated for IOS and Android devices. All data (photos, species descriptions, ...) are stored on the server side of the App. In this way, future updates and improvements of the F.I.S.HUB server, species, or species descriptions will not require the deployment of an updated version of the App. The UI interface design as well as logos and graphics have been developed by two students of the Degree in Design and Visual Communication of Politecnico di Torino as their bachelor Thesis.



# Chapter 4

## From Image to Molecular Analysis

### 4.1 Introduction

In this exploratory study, we explored the possibility of extending the F.I.S.HUB image classification approach to forgery of fish fillets. This operation is more complex than for fishes, where geometrical features make the classification task much simpler; here the most important task is therefore to identify visual features that could be extracted to reliably distinguish a fish fillet from its substitute (see Figure 4.1).

To do this we have analyzed different possible types of the photo (normal, micro), as well as fillet staining techniques (to show the fibrous structure), in order to maximize the number of visible features that could be used by the image analysis software. Eventually, we have also explored the use of the SCiO Near Infra-Red (NIR) Spectroscopy molecular sensor instrument to investigate an alternative method to achieve the desired results.

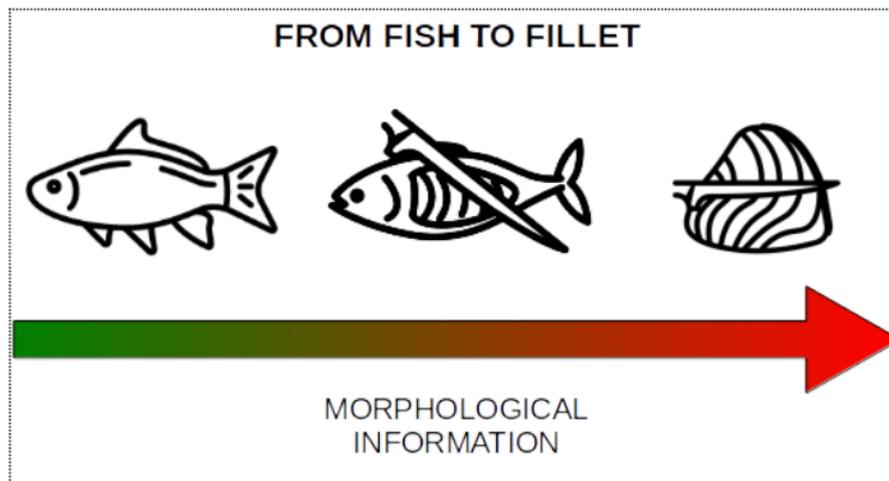


Figure 4.1: From fish to fillet morphological information

## 4.2 Method

Firstly, we defined the methods to be investigated for fish fillet recognition. Then, the proof-of-concept (POC) tool has been defined and tested and is here proposed as a possible prospective method to be studied further. All the methods and results are hereafter described.

### 4.2.1 Method 1 - Fillet Picture Analysis

The first method analyzed was the fillets picture analysis to find a set of geometrical features able to discriminate fillets from different species of fish. Since fillets are boneless cut or slice of the whole fish they do not have all the typical visible distinction such as shape or color. In fact, the only significant features concern the fish muscles which however could have undeniable differences depending on the cutting plane angle and may vary over the specific portion of the body muscles.

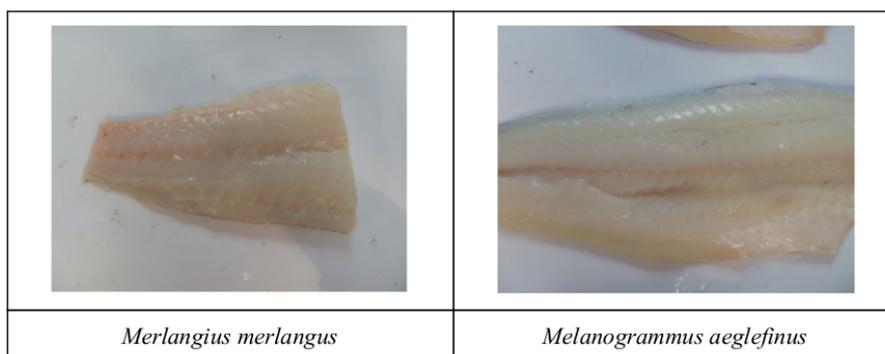


Figure 4.2: *Merlangius merlangus* and *Melanogrammus aeglefinus* fillets

The first test was realized with *Merlangius merlangus* and *Melanogrammus aeglefinus* fillets (see Figure 4.2). For each species 72 pictures were collected and processed with image feature extraction techniques as: Scale-invariant feature transform (SIFT) [12], Speeded Up Robust Feature (SURF) [5] and Gray Level Co-Occurrence Matrix (GLCM) [44]. The overall results are listed in the Table 4.1 below. The extracted features were then used to train a Support Vector Machines (SVM) classifier [74].

Table 4.1: *Merlangius merlangus* and *Melanogrammus aeglefinus* accuracy of picture analysis

ACCURACY [%]	SIFT	SURF	GLCM
<i>Merlangius merlangus</i>	65.3	61.1	59.7
<i>Melanogrammus aeglefinus</i>	63.9	56.9	58.3
TOTAL	64.6	60.3	59

The same procedure was also carried out for three different species of flat fish: *Solea solea* (41 pictures), *Pleuronectes platessa* (35 pictures), and *Pangasianodon hypophthalmus* (36 pictures). In the following Table 4.2 are presented the obtained results.

Table 4.2: *Solea solea*, *Pleuronectes platessa*, and *Pangasianodon hypophthalmus* accuracy of picture analysis

ACCURACY [%]	SIFT	SURF	GLCM
<i>Solea solea</i>	56.1	56.1	53.7
<i>Pleuronectes platessa</i>	60	62.9	54.3
<i>Pangasianodon hypophthalmus</i>	55.6	58.3	52.8
TOTAL	57.1	59.8	53.6

#### 4.2.2 Method 2 - Fillet Microscope Picture

The second method proposed is the microscopic image analysis realized with the Camera Phone Lenses 200X Lens Microscope LED (see Figure 4.3). This tool has been selected since it is compatible with any ordinary smartphone and also taking into account its usability-in-the-field as one key requirement for the fillets recognition procedure.



Figure 4.3: Camera Phone Lenses 200X Lens Microscope LED

Three pictures per fillet were acquired using the same fillet dataset described before and it is composed by: *Solea solea*, *Pleuronectes platessa*, and *Pangasianodon hypophthalmus* (see Figure 4.4).

The obtained results are listed below in Table 4.3:

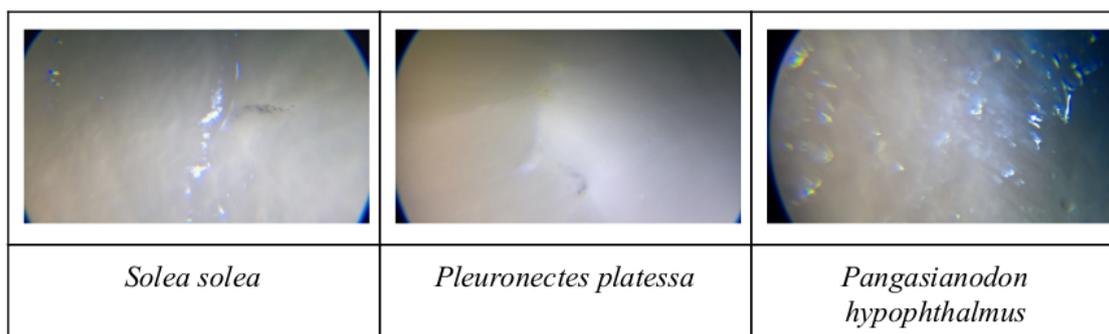


Figure 4.4: *Solea solea*, *Pleuronectes platessa*, and *Pangasianodon hypophthalmus* examples

Table 4.3: Microscope analysis accuracy

ACCURACY [%]	SIFT	SURF	GLCM
TOTAL	48.1	49.5	44.9

### 4.2.3 Method 3 - Staining

One more method applied to image analysis is the fillet staining preparation with “Toluidine blue stain” applied to both normal and microscopic images. This colorant has been proposed by IZSTO. With the aid of a brush a small amount of dye was laid on the fillet surface and after few seconds a normal picture and a microscopic one were taken. The fillets dataset is the same as the one described in the previous subsections.

In the next Figure 4.5 there are some examples of fillets images with Toluidine blue stain preparation.

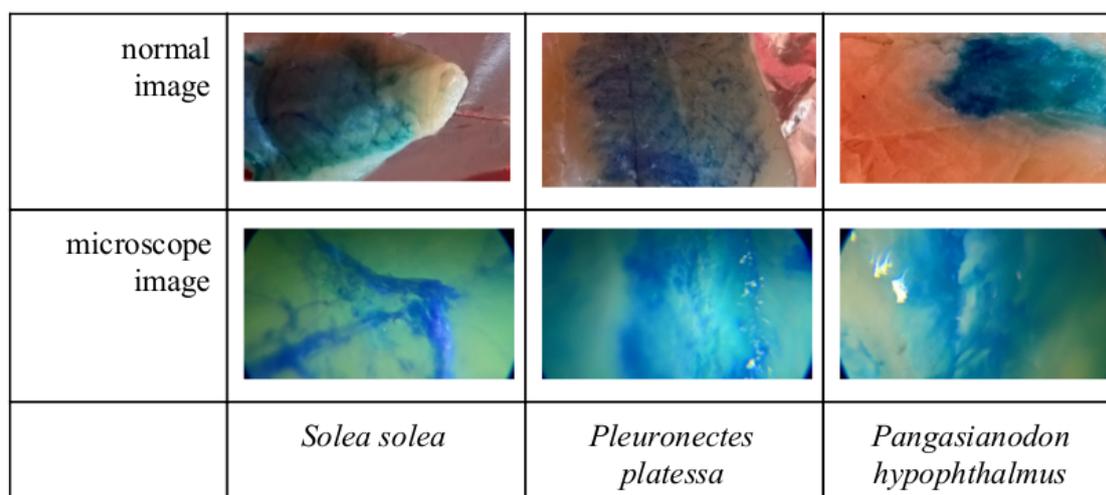


Figure 4.5: Fillets images with Toluidine blue stain preparation

The SVM classification results for all the feature extraction method used for normal and microscopic images are reported in the next Table 4.4.

Table 4.4: Staining analysis accuracy

ACCURACY [%]	SIFT	SURF	GLCM
TOTAL normal	56.6	53.3	51.2
TOTAL microscope	49.7	51.8	53.2

#### 4.2.4 Method 4 - NIR sensor

The Near-Infrared spectroscopic is a method that makes use of the electromagnetic spectrum from about 700 nm to 2500 nm. It can penetrate tissues and it is useful to probe bulk material with essentially no preparation. The NIR sensor produces an absorption spectrum that can be used to create discriminative model depending on the feature evaluated by the instrument.

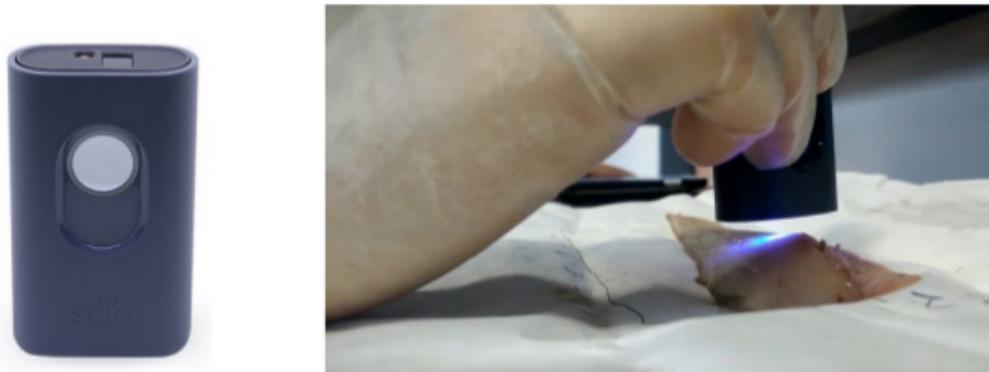


Figure 4.6: SCiO sensor

To measure the NIR spectrum and to perform a classification model of the fish fillets we adopted the SCiO molecular sensor, developed and distributed by Consumer Physics <sup>1</sup> (see Figure 4.6). In connection with the instrument the SCiO Developer Toolkit enables to customize and collect the spectra of the desired materials via mobile application. Eventually the SCiO Lab Web analyzes the spectral data and generates the classification model that can be later used to analyze material with other SCiO devices (see Figure 4.7).

For the POC we selected 10 fish fillets for each considered species: *Solea solea*, *Pleuronectes platessa*, and *Pangasianodon hypophthalmus*. Every species were sampled 60

<sup>1</sup>SCiO molecular sensor by Consumer Physics [link](#)

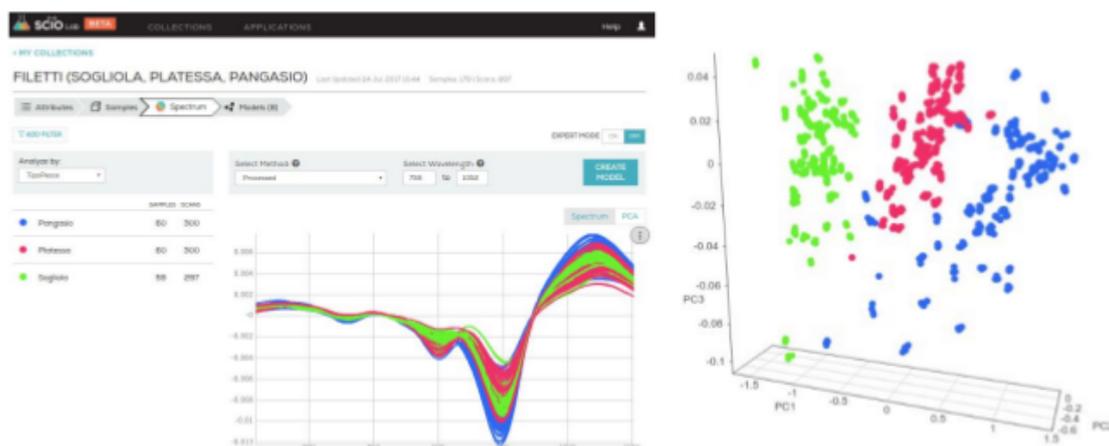


Figure 4.7: SCiO Lab View screenshot

times, with 5 different scans for each sample. Notice that multiple scans per sample are recommended in the guideline to perform a better evaluation. The validation set, on the other hand, was carried out over 3 fish fillets per species, comprising 30 samples per species with 3 different scans per sample. The acquired scans were realized at different points of the fillet in order to evaluate possible variation with respect to the portion of the fillet side. All the fresh fish fillets were gathered from two different fish market located in Turin (Italy) and verified by IZSTO controllers.

### 4.3 Results

A classification model was trained in the SCiO Lab Web with the following setup:

- Select WL [759 nm - 1052 nm]
- Selected Method: Processed and Normalized
- Outlier Detection: OFF

Hereafter in Figure 4.8 are shown the results evaluated over the validation set for SCiO analysis.

In Table 4.5 there is the summary that reports all the obtained results for the four different methods proposed in this pilot study. Concerning Method 1, 2 and 3 we outline only the best result among SIFT, SURF and GLCM. We here consider the result for the three fish fillet that was analyzed in every proposed method: *Solea solea*, *Pleuronectes platessa*, and *Pangasianodon hypophthalmus*.

The fish fillets identification seems to be not feasible with image processing techniques because the visual features are not informative and there is no improvement even with staining techniques.

Expected label →			
↓ Classification result	<i>Solea solea</i>	<i>Pleuronectes platessa</i>	<i>Pangasianodon hypophthalmus</i>
<i>Solea solea</i>	<b>98</b>	0	2
<i>Pleuronectes platessa</i>	2	<b>97</b>	1
<i>Pangasianodon hypophthalmus</i>	0	0	<b>100</b>

Figure 4.8: SCiO accuracy calculated on the validation set

On the other hand, the NIR spectrometer puts in evidence very promising results to fight the fish fillet forgery and it is the proposed POC ( Milestone 70 ). Finally, the SCiO development tool seems very accurate and quick to use in particular in the context of on-site controls, and it can be surely employed also by non-expert users.

Table 4.5: Results summary for every method

METHOD	DETAILS	ACCURACY [%]
1	normal img	59.8
2	microscope img	49.5
3	normal img + staining	56.6
3	microscope img + staining	51.9
4	SCiO NIR	98.3



# Chapter 5

## Conclusion

This Ph.D. thesis contributed to developing innovative methodologies in the wide field of Food Risk assessment. In particular, it introduces the Computer-Aided capability to counteract the economic and health consequences of intentional and unintentional actions. Moreover, it offers practical solutions that can be potentially used both for controllers and consumers. In this thesis, the proposed methodologies were applied to cases related with Food Traceability, Food Safety, and Food Quality. Every contribution has been evaluated from an interdisciplinary point of view, taking care to the biological and engineering constraints, and trying to design the most cost-effective and efficient solution.

The experimental results presented in this work demonstrate that every single method can be applied to other related analogous tasks. For example, the molecular traceability heuristic-based software for dairy product may be applied to other DNA-pooled analysis. Even the designed Computer Vision and Deep Learning methodologies are employable into other circumstances where the visual understanding plays an important role. Moreover, Near Infra-Red sensors are just in their beginning and many efforts should be spent to validate their own would be. All the contribution discussed in this thesis started from a real practical problem and were able to introduce a real feasible solution. The endpoints of these technological aided-tools underline a potential path where it is possible to start further investigations. The ever-growing computing resource and the biological data availability can concur positively for revolutionary advancement in their respective field of application.



# Bibliography

- [1] Alsmadi. “Fish Classification Based on Robust Features Extraction From Color Signature Using Back-Propagation Classifier”. In: *Journal of Computer Science* 7.1 (2011), pp. 52–58. DOI: [10.3844/jcssp.2011.52.58](https://doi.org/10.3844/jcssp.2011.52.58). URL: <http://dx.doi.org/10.3844/jcssp.2011.52.58>.
- [2] Naomi S Altman. “An introduction to kernel and nearest-neighbor nonparametric regression”. In: *The American Statistician* 46.3 (1992), pp. 175–185.
- [3] Anne Auger and Nikolaus Hansen. “A restart CMA evolution strategy with increasing population size”. In: *Evolutionary Computation, 2005. The 2005 IEEE Congress on*. Vol. 2. IEEE. 2005, pp. 1769–1776.
- [4] A Bagnato et al. “Quantitative trait loci affecting milk yield and protein percentage in a three-country Brown Swiss population”. In: *Journal of dairy science* 91.2 (2008), pp. 767–783.
- [5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. “Surf: Speeded up robust features”. In: *European conference on computer vision*. Springer. 2006, pp. 404–417.
- [6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.
- [7] Christopher Bishop. “Neural Networks”. In: *Handbook of Neural Computation*. Informa UK Limited, 1996. DOI: [10.1201/9781420050646.ptb6](https://doi.org/10.1201/9781420050646.ptb6). URL: <http://dx.doi.org/10.1201/9781420050646.ptb6>.
- [8] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [9] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN: 0387310738.
- [10] G. Bradski. “The OpenCV Library”. In: *Dr. Dobb’s Journal of Software Tools* (2000).
- [11] Jane Bromley et al. “Signature verification using a” siamese” time delay neural network”. In: *Advances in Neural Information Processing Systems*. 1994, pp. 737–744.

- 
- [12] Frédéric Cao et al. “The SIFT Method”. In: *A Theory of Shape Identification* (2008), pp. 185–208.
- [13] Sumit Chopra, Raia Hadsell, and Yann LeCun. “Learning a similarity metric discriminatively, with application to face verification”. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 539–546.
- [14] Dan CireşAn et al. “Multi-column deep neural network for traffic sign classification”. In: *Neural networks* 32 (2012), pp. 333–338.
- [15] European Commission. *Traceability factsheet*. General food law, 2007. ISBN: 978-92-79-05725-0.
- [16] Nicholas G Crawford. “SMOGD: software for the measurement of genetic diversity”. In: *Molecular Ecology Resources* 10.3 (2010), pp. 556–557.
- [17] Michael P Cummings. “PHYLIP (phylogeny inference package)”. In: *Dictionary of Bioinformatics and Computational Biology* (2004).
- [18] Richard O. Duda and Peter E. Hart. “Use of the Hough transformation to detect lines and curves in pictures”. In: *Communications of the ACM* 15.1 (1972), pp. 11–15. DOI: [10.1145/361237.361242](https://doi.org/10.1145/361237.361242). URL: <http://dx.doi.org/10.1145/361237.361242>.
- [19] Johanne Fischer. *Fish identification tools for biodiversity and fisheries assessments: review and guidance for decision-makers*. Food and Agriculture Organization of the United Nations, 2013.
- [20] Mohamed Mostafa M. Fouad et al. “Automatic Nile Tilapia fish classification approach using machine learning techniques”. In: *13th International Conference on Hybrid Intelligent Systems (HIS 2013)*. Institute of Electrical & Electronics Engineers (IEEE), 2013. DOI: [10.1109/his.2013.6920477](https://doi.org/10.1109/his.2013.6920477). URL: <http://dx.doi.org/10.1109/his.2013.6920477>.
- [21] Ian Goodfellow et al. *Deep learning*. Vol. 1. MIT press Cambridge, 2016.
- [22] J Goudet. “FSTAT, a program to estimate and test gene diversity and fixation indices (version 2.9.3)”. In: <http://www2.unil.ch/popgen/softwares/fstat.htm> (2001).
- [23] C. Guisande et al. “IPEz: An expert system for the taxonomic identification of fishes based on machine learning techniques”. In: *Fisheries Research* 102.3 (2010), pp. 240–247. ISSN: 0165-7836. DOI: <https://doi.org/10.1016/j.fishres.2009.12.003>. URL: <http://www.sciencedirect.com/science/article/pii/S0165783609003191>.
- [24] Raia Hadsell, Sumit Chopra, and Yann LeCun. “Dimensionality reduction by learning an invariant mapping”. In: *Computer vision and pattern recognition, 2006 IEEE computer society conference on*. Vol. 2. IEEE. 2006, pp. 1735–1742.

- [25] Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)”. In: *Evolutionary computation* 11.1 (2003), pp. 1–18.
- [26] Nikolaus Hansen and Andreas Ostermeier. “Completely derandomized self-adaptation in evolution strategies”. In: *Evolutionary computation* 9.2 (2001), pp. 159–195.
- [27] Nikolaus Hansen, Andreas Ostermeier, and Andreas Gawelczyk. “On the Adaptation of Arbitrary Normal Mutation Distributions in Evolution Strategies: The Generating Set Adaptation.” In: *ICGA*. 1995, pp. 57–64.
- [28] Yen-Chang Hsu and Zsolt Kira. “Neural network-based clustering using pairwise constraints”. In: *arXiv preprint arXiv:1511.06321* (2015).
- [29] Jing Hu et al. “Fish species classification by color, texture and multi-class support vector machine using computer vision”. In: *Computers and Electronics in Agriculture* 88 (2012), pp. 133–140. DOI: [10.1016/j.compag.2012.07.008](https://doi.org/10.1016/j.compag.2012.07.008). URL: <http://dx.doi.org/10.1016/j.compag.2012.07.008>.
- [30] Melissa J Hubisz et al. “Inferring weak population structure with the assistance of sample group information”. In: *Molecular ecology resources* 9.5 (2009), pp. 1322–1332.
- [31] I Ismail and A Hanif Halim. “Comparative study of meta-heuristics optimization algorithm using benchmark function”. In: *International Journal of Electrical and Computer Engineering* 7.3 (2017), pp. 1643–1650.
- [32] Yangqing Jia et al. “Caffe: Convolutional Architecture for Fast Feature Embedding”. In: *arXiv preprint arXiv:1408.5093* (2014).
- [33] Marc Kochzius et al. “Identifying Fishes through DNA Barcodes and Microarrays”. In: *PLoS ONE* 5.9 (Sept. 2010), e12620. DOI: [10.1371/journal.pone.0012620](https://doi.org/10.1371/journal.pone.0012620). URL: <http://dx.doi.org/10.1371%2Fjournal.pone.0012620>.
- [34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.
- [35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.
- [36] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), p. 436.

- [37] Lian Li and Jinqi Hong. “Identification of fish species based on image processing and statistical analysis research”. In: *2014 IEEE International Conference on Mechatronics and Automation*. Institute of Electrical & Electronics Engineers (IEEE), 2014. DOI: [10.1109/icma.2014.6885861](https://doi.org/10.1109/icma.2014.6885861). URL: <http://dx.doi.org/10.1109/icma.2014.6885861>.
- [38] CP Likhitha, P Ninitha, and V Kanchana. “DNA Bar-Coding: A Novel Approach for Identifying an Individual Using Extended Levenshtein Distance Algorithm and STR Analysis”. In: *International Journal of Electrical and Computer Engineering* (2016).
- [39] Ehud Lipkin et al. “Quantitative trait locus mapping in chickens by selective DNA pooling with dinucleotide microsatellite markers by using purified DNA and fresh or frozen red blood cells as applied to marker-assisted selection”. In: *Poultry science* 81.3 (2002), pp. 283–292.
- [40] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.
- [41] Masakazu Matsugu et al. “Subject independent facial expression recognition with robust face detection using a convolutional neural network”. In: *Neural Networks* 16.5-6 (2003), pp. 555–559.
- [42] Patrick McClure and Nikolaus Kriegeskorte. “Representational Distance Learning for Deep Neural Networks”. In: *Frontiers in computational neuroscience* 10 (2016), p. 131.
- [43] Hendrik-Jan Megens et al. “Biodiversity of pig breeds from China and Europe estimated from pooled DNA samples: differences in microsatellite variation between two areas of domestication”. In: *Genetics Selection Evolution* 40.1 (2008), p. 103.
- [44] P Mohanaiah, P Sathyanarayana, and L GuruKumar. “Image texture feature extraction using GLCM approach”. In: *International Journal of Scientific and Research Publications* 3.5 (2013), p. 1.
- [45] Emilio Soria Olivas. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques: Algorithms, Methods, and Techniques*. IGI Global, 2009.
- [46] Maxime Oquab et al. “Learning and transferring mid-level image representations using convolutional neural networks”. In: *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1717–1724.
- [47] J. Palma and J.P. Andrade. “Morphological study of *Diplodus sargus*, *Diplodus puntazzo*, and *Lithognathus mormyrus* (Sparidae) in the Eastern Atlantic and Mediterranean Sea”. In: *Fisheries Research* 57.1 (2002), pp. 1–8. DOI: [10.1016/S0165-7836\(01\)00335-6](https://doi.org/10.1016/S0165-7836(01)00335-6). URL: [http://dx.doi.org/10.1016/S0165-7836\(01\)00335-6](http://dx.doi.org/10.1016/S0165-7836(01)00335-6).

- [48] Sinno Jialin Pan and Qiang Yang. “A survey on transfer learning”. In: *IEEE Transactions on knowledge and data engineering* 22.10 (2010), pp. 1345–1359.
- [49] Jung-Min Park et al. “Determination of the authenticity of dairy products on the basis of fatty acids and triacylglycerols content using GC analysis”. In: *Korean journal for food science of animal resources* 34.3 (2014), p. 316.
- [50] Angela Di Pinto et al. “Species identification in fish fillet products using DNA bar-coding”. In: *Fisheries Research* 170 (2015), pp. 9–13. DOI: [10.1016/j.fishres.2015.05.006](https://doi.org/10.1016/j.fishres.2015.05.006). URL: <http://dx.doi.org/10.1016/j.fishres.2015.05.006>.
- [51] Rajat Raina, Anand Madhavan, and Andrew Y Ng. “Large-scale deep unsupervised learning using graphics processors”. In: *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 873–880.
- [52] Francesco Rossi et al. “Computer-Assisted Molecular Traceability for Dairy Farming Products”. In: *Proceedings Extended Abstracts 4th International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO 2016)* (2016), pp. 371–378.
- [53] Francesco Rossi et al. “DNA Pool Analysis-based Forgery-Detection of Dairy Products”. In: *International Journal of Electrical and Computer Engineering (IJECE)* 8.6 (2018). ISSN: 2088-8708.
- [54] Francesco Rossi et al. “FishAPP: A mobile App to detect fish falsification through image processing and machine learning techniques”. In: (2016), pp. 1–6. DOI: [10.1109/AQTR.2016.7501348](https://doi.org/10.1109/AQTR.2016.7501348).
- [55] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. “GrabCut”. In: *TOG* 23.3 (2004), p. 309. DOI: [10.1145/1015706.1015720](https://doi.org/10.1145/1015706.1015720). URL: <http://dx.doi.org/10.1145/1015706.1015720>.
- [56] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016).
- [57] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.
- [58] Maria Teresa Sardina et al. “Application of microsatellite markers as potential tools for traceability of Girgentana goat breed dairy products”. In: *Food Research International* 74 (2015), pp. 115–122.
- [59] Jürgen Schmidhuber. “Deep learning in neural networks: An overview”. In: *Neural networks* 61 (2015), pp. 85–117.
- [60] Hugo G Schnack et al. “Accurate determination of microsatellite allele frequencies in pooled DNA samples”. In: *European journal of human genetics* 12.11 (2004), p. 925.
- [61] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

- [62] Garrick T Skalski et al. "Evaluation of DNA pooling for the estimation of microsatellite allele frequencies: a case study using striped bass (*Morone saxatilis*)". In: *Genetics* 173.2 (2006), pp. 863–875.
- [63] John Spink and Douglas C. Moyer. "Defining the Public Health Threat of Food Fraud". In: *Journal of Food Science* 76.9 (2011), R157–R163. ISSN: 1750-3841. DOI: [10.1111/j.1750-3841.2011.02417.x](https://doi.org/10.1111/j.1750-3841.2011.02417.x). URL: <http://https://doi.org/10.1111/j.1750-3841.2011.02417.x>.
- [64] Frank Storbeck and Berent Daan. "Fish species recognition using computer vision and a neural network". In: *Fisheries Research* 51.1 (2001), pp. 11–15. DOI: [10.1016/S0165-7836\(00\)00254-X](https://doi.org/10.1016/S0165-7836(00)00254-X). URL: [http://dx.doi.org/10.1016/S0165-7836\(00\)00254-X](http://dx.doi.org/10.1016/S0165-7836(00)00254-X).
- [65] Christian Szegedy et al. "Going deeper with convolutions". In: *Cvpr*. 2015.
- [66] R Core Team. *R: A language and environment for statistical computing*. *R Foundation for Statistical Computing, Vienna, Austria*. 2016. 2017.
- [67] Robert J Toonen and Shayne Hughes. "Increased throughput for fragment analysis on an ABI Prism® 377 automated sequencer using a membrane comb and STRand software". In: *Biotechniques* 31.6 (2001), pp. 1320–1325.
- [68] Heike Trautmann, Olaf Mersmann, and David Arnu. "cmaes: Covariance matrix adapting evolutionary strategy". In: *R package*, URL <http://cran.r-project.org/package=cmaes> (2011).
- [69] John M. Wargo. *PhoneGap Essentials: Building Cross-Platform Mobile Apps*. 1st. Addison-Wesley Professional, 2012.
- [70] D.J. White, C. Svellingen, and N.J.C. Strachan. "Automated measurement of species and length of fish by computer vision". In: *Fisheries Research* 80.2-3 (2006), pp. 203–210. DOI: [10.1016/j.fishres.2006.04.009](https://doi.org/10.1016/j.fishres.2006.04.009). URL: <http://dx.doi.org/10.1016/j.fishres.2006.04.009>.
- [71] M Rahmat Widyanto, Reggio N Hartono, and Nurtami Soedarsono. "A Novel Human STR Similarity Method using Cascade Statistical Fuzzy Rules with Tribal Information Inference". In: *International Journal of Electrical and Computer Engineering* 6.6 (2016), p. 3103.
- [72] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems*. 2014, pp. 3320–3328.
- [73] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer. 2014, pp. 818–833.
- [74] Yongli Zhang. "Support vector machine classification algorithm and its application". In: *International Conference on Information Computing and Applications*. Springer. 2012, pp. 179–186.

- [75] B. Zion, A. Shklyar, and I. Karplus. “In-vivo fish sorting by computer vision”. In: *Aquacultural Engineering* 22.3 (2000), pp. 165–179. DOI: [10.1016/S0144-8609\(99\)00037-0](https://doi.org/10.1016/S0144-8609(99)00037-0). URL: [http://dx.doi.org/10.1016/S0144-8609\(99\)00037-0](http://dx.doi.org/10.1016/S0144-8609(99)00037-0).
- [76] B Zion, A Shklyar, and I Karplus. “Sorting fish by computer vision”. In: *Computers and Electronics in Agriculture* 23.3 (1999), pp. 175–187. DOI: [10.1016/S0168-1699\(99\)00030-7](https://doi.org/10.1016/S0168-1699(99)00030-7). URL: [http://dx.doi.org/10.1016/S0168-1699\(99\)00030-7](http://dx.doi.org/10.1016/S0168-1699(99)00030-7).
- [77] Boaz Zion et al. “Real-time underwater sorting of edible fish species”. In: *Computers and Electronics in Agriculture* 56.1 (2007), pp. 34–45. DOI: [10.1016/j.compag.2006.12.007](https://doi.org/10.1016/j.compag.2006.12.007). URL: <http://dx.doi.org/10.1016/j.compag.2006.12.007>.

This Ph.D. thesis has been typeset by means of the  $\TeX$ -system facilities. The typesetting engine was  $\text{Lua}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ . The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete  $\TeX$ -system installation.