

University of L'Aquila

Department of Information Engineering, Computer Science and Mathematics

PhD in Information and Communication Technologies

Curriculum – Systems Engineering, Telecommunications and HW/SW platforms

SSD ING-INF/04

XXXIII Cycle

Real-Time Algorithms for Spectral Classification Tasks

Candidate

Stefano Di Frischia

ID number 255757

Thesis Advisor

Prof.ssa Elena De Santis

Co-Advisor

Dr. Federico Angelini

Chair of PhD School in ICT

Prof. Vittorio Cortellessa

14th June 2021

Real-Time Algorithms for Spectral Classification Tasks
Ph.D. thesis. University of L'Aquila

© 2021 Stefano Di Frischia. All rights reserved

This thesis has been typeset by LaTeX and the uaqthesis class.

Author's email: stefano.difrischia@graduate.univaq.it

Light and matter are both single entities, and the apparent duality arises in the limitations of our language.

- Werner Heisenberg -

Abstract

The present PhD thesis covers a research about real-time spectral classification exploiting machine learning tools and in particular deep learning with neural networks. Spectroscopy is the study of the interaction between matter and electromagnetic radiation. In particular, we focused on Raman spectroscopy because it allows a chemical compound identification that has many fields of application. Our approach to solve spectral classification is a combination of signal processing methods and machine learning tools. First, it has been analysed the most important spectrum feature that could affect classification accuracy: signal-to-noise ratio. Since the level of noise must be reduced both at acquisition time and at processing time, an investigation of the noise sources in a CCD device has been carried on in order to identify the main parameters that concur to generate noise. Once that the best acquisition strategy has been demonstrated, we focused on a Cultural Heritage case of study about pigment spectra classification, exploiting machine learning tools to overcome the problem of a limited reference database. A hierarchical pipeline of signal processing methods has been designed to implement the necessary Data Augmentation. In addition, these methods have been enhanced with the use of Generative Adversarial Networks (GANs). Once that an augmented dataset has been generated, we tested it on 3 different neural network architectures, finding that the Convolutional Neural Networks (CNN) with GAN enhancement is the best approach for classification. The implemented technique of the present thesis is potentially applicable in every spectroscopic analysis that lacks a sufficient number of training reference examples.

Keywords: Deep Learning, Generative Adversarial Network, Data Augmentation, Raman Spectroscopy, Signal Processing, Pattern Recognition.

Contents

List of Figures v

List of Tables viii

List of Code ix

1 Introduction 1

2 Spectroscopy 5

2.1 Raman Spectroscopy 8

2.2 Raman Spectrum 10

2.3 Components of a Raman LIDAR Apparatus 11

2.4 Spectral Matching 14

2.5 Spectral Classification Background 18

2.6 State of the Art 21

3 Real-Time Classification Approach 24

3.1 Pre-Processing vs Machine Learning 25

3.2 Pre-Processing: Baseline Removal 27

3.3 Pre-Processing: Noise Filtering 32

3.4 Artificial Neural Networks 35

3.4.1 Evaluation of ANNs on Spectral Classification Tasks 38

3.5 Most promising ML technique: CNN 48

4 The Method 51

4.1 Charge Coupled Device (CCD) 51

4.2 Sources of Noise in a CCD 53
4.3 Acquisition strategy to optimize SNR 56
4.3.1 Laser Safety 56
4.3.2 Acquisition Strategies Comparison 57
4.3.3 Simulation Results 60
4.4 Data Augmentation 62
4.5 Hierarchical Pipeline 64
4.6 Generative Adversarial Networks 70

5 Use Case Study **73**
5.1 Cultural Heritage 73
5.2 Pigment Dataset 75
5.3 Pigment Data Augmentation Pipeline 79
5.3.1 Additive White Noise 79
5.3.2 Baseline Removal 80
5.3.3 Noise Reduction 82
5.3.4 K-Means Clustering, Shifting and Merging 84
5.4 GANs Augmentation Enhancement 85
5.5 Pigment Classification 92
5.5.1 Neural Network Architectures 92
5.5.2 Results 100

6 Conclusions **103**

Appendices **107**
.1 Appendix - List of Pigments 107

Bibliography **109**

List of Figures

2.1 Electromagnetic Spectrum (from https://www.miniphysics.com/) . 6
2.2 The Jablonski Diagram of molecular absorbance and fluorescence (from https://www.horiba.com) 7
2.3 Schematic representation of Rayleigh and Raman scattering [1]. In (a) μ oscillates at the incident photon frequency, in (b) the molecular vibration can change α, while in (c) the final oscillation results in an amplitude modulated dipole moment, where the components (d) emit electromagnetic radiation. 9
2.4 Raman spectrum of Teflon $[C_nF_{2n+2}]$[2] 10
2.5 Schematic representation of a Raman LIDAR apparatus 12
2.6 Representation of a diffraction grating (from https://physicsopenlab.org/) 14
2.7 Schematic representation (above) and 3D-model (below) of RADEX system [3] 15
2.8 Stokes bands of two Raman measurements of Teflon. On the left a reference spectrum found in literature [2]. On the right two experimental acquired spectra: the red one is acquired with an optical slit of $100\mu m$, the blue one with a slit of $200\mu m$ 16
2.9 An experimental acquired Raman spectrum that presents a significant component of fluorescence 17
2.10 Skills required for a successful spectroscopist [4] 20

3.1 Pre-Processing vs Machine Learning approach 26
3.2 Examples of Baseline Removal algorithms (1of2) 29

3.3 Examples of Baseline Removal algorithms (2of2) 30
3.4 Examples of Noise filtering algorithms (1of2) 33
3.5 Examples of Noise filtering algorithms (2of2) 34
3.6 Basic representation of an artificial Neuron [5] 36
3.7 Some examples of ANN activation functions [5] 37
3.8 A multi-layered Artificial Neural Network [5] 37
3.9 A Gaussian function (left) and an example of input set represented as a heat-map (right) with vertically stacked spectra 39
3.10 Classification Confusion Matrix 41
3.11 ANN Accuracy vs Input Set Dimension (left), ANN processing time vs Input Set Dimension (right) 41
3.12 High SNR spectra (above) vs Low SNR spectra (below) 42
3.13 ANN accuracy vs Signal-to-Noise Ratio 43
3.14 ANN accuracy vs Hidden Layer Size 43
3.15 ANN False Negatives and False Positives Histogram related to the shifting distance 44
3.16 ANN False Positives heat-map 45
3.17 Optical acquisition system setup 46
3.18 Examples of Hg spectra acquired from the highest level of SNR (upper-left) to the lowest level of SNR (bottom-right) 47
3.19 Example of Convolutional Neural Network [6] 49

4.1 A Charge Coupled Device designed for UV imaging (from https://en.wikipedia.org/wiki/Charge-coupled_device) 52
4.2 A basic scheme of CCD operating mechanism (from https://www.gxccd.com/) 53
4.3 MPE vs wavelength for some exposure times for $\alpha < 1.5mrad$. Solid lines refer to eye MPE, while dotted lines refer to skin MPE. Time in legend is expressed in s [7] 57
4.4 Four different acquisition strategies in a spectroscopic experiment[8] 58
4.5 A-B values [a.u.] vs Exposure Time [s] 60
4.6 Noise components [a.u.] vs Exposure Time [s] 62

4.7 Data Augmentation Pipeline Flow-chart 69
4.8 GAN architecture [9] 71

5.1 A spectroscopic diagnostic campaign led by ENEA in the Catacombs of Priscilla - Rome, July 2016 74
5.2 Fresco cross-section with relative layers (adapted from https://benedettalopezbani.wordpress.com) 76
5.3 The commercial Raman System used for sample measurements (from https://bwtek.com/) 77
5.4 Raman spectra of four pigment samples 78
5.5 Example results of the first step of the pipeline 80
5.6 Example results of the second step of the pipeline 82
5.7 Example results of the third step of the pipeline 84
5.8 K-means Clustering of some pigment classes 85
5.9 GAN architecture scheme [9] 86
5.10 Generator Loss Values Trend for different z values 87
5.11 GAN generating spectra for D1 subset 93
5.12 GAN generating spectra for D2 subset 94
5.13 GAN generating spectra for D3 subset 95
5.14 Binary Neural Network architecture 97
5.15 Categorical Neural Network architecture 98
5.16 Convolutional Neural Network architecture 99
5.17 Binary NN classification accuracy 101

List of Tables

3.1 Classification confusion matrix of Hg spectra test 47

4.1 List of four different acquisition strategies and relative parameters . 58
4.2 SNR comparison 59

5.1 Raman spectra classification results 102

.1 List of pigment samples used for the Raman measurements 108

List of Code

5.1 White Noise augmentation 79
5.2 Baseline removal augmentation 81
5.3 Noise reduction augmentation 82
5.4 GAN training 88
5.5 Generator and Discriminator network architecture 90
5.6 Binary Neural Network model 96
5.7 Categorical Neural Network model 97
5.8 Convolutional Neural Network model 99

Chapter 1

Introduction

Spectral classification is the task of identifying a sample through its electromagnetic spectrum measured by a certain spectroscopic technique. It is a multidisciplinary field of study that embraces several sciences like physics, chemistry, engineering and computer science. The chemical compound identification allowed by some types of spectroscopy (e.g. Raman spectroscopy, the specific technique chosen for the present research) results useful in many fields of applications, both scientific and industrial. Among the many fields in which Raman spectroscopy is being used, the most notable in literature are medicine, biology, applied chemistry, security and cultural heritage.

The main objective of the present PhD thesis is to find an innovative method to achieve real-time spectral classification exploiting both signal processing and machine learning algorithms. In particular, this research focuses on experimental scenarios where the spectra acquisition and classification can be difficult due to several issues: concerning the acquisition, measurements acquired through laser devices must adhere to safety requirements, and therefore the intensity of the signal respect to the noise could not be optimal as in laboratory scenarios. Consequently, the sample classification could not be performed with the usual accuracy standards, or could become a time-consuming task for the operator. The present research tries to overcome these issues, introducing an innovative data augmentation pipeline that allows to exploit the potentials of deep learning and signal processing tools in the classification tasks.

Given the multitude of fields that experiment with Raman spectroscopy, there are plenty of scientific articles and studies in literature which try to find a correct classification approach. The methods can be divided in pre and post deep learning era, whereas in the last years most of the classification techniques focus on neural networks and machine learning in general. What is lacking in literature is a unique and validated approach that does not change along with the target data. In fact, each research often uses a methodology that is calibrated upon the data that are being investigated, with the result of creating a technique not suitable for other domains. Further, the deep learning and signal processing methodologies are so far being exploited separately or, at best, in a fixed approach that uses signal filters in the spectra pre-processing step and deep learning tools only for classification purposes. This research, instead, tries to find a convergence between the two approaches, exploiting the signal processing methods as data augmentation tools that concur to generate an input set suitable for neural networks. Moreover, the objectives of this research are not focused on classifying data of a particular domain, but rather on finding a general method that can be exploited in several scenarios and with several spectroscopic techniques.

The present research cover three years of work, whereas the first two years have been spent in the ENEA Research Centre of Frascati (Rome) in the Diagnostic and Metrology Laboratory, while the third year has been spent at the Department of Engineering, Information Science and Mathematics at the University of L'Aquila. At the beginning, after the study of spectroscopy concepts and the operational behaviours of several Raman devices, it has been analysed the most important spectrum feature that could affect classification accuracy: signal-to-noise ratio. Since the level of noise must be reduced both at acquisition time and at processing time, an investigation of the noise sources in a CCD device has been carried on in order to identify the main parameters that concur to generate noise. Once that the best acquisition strategy has been demonstrated, the research focused on a Cultural Heritage case of study about pigment spectra classification, where the exploitation of machine learning tools has been found as the solution to overcome the problem of a limited reference database. A hierarchical pipeline of signal processing methods

has been designed to implement the necessary data augmentation. In addition, these methods have been enhanced with the use of Generative Adversarial Networks (GANs). Once that an augmented dataset has been generated, it has been tested on 3 different neural network architectures, finding that the Convolutional Neural Networks (CNN) with GAN enhancement is the best approach for real-time spectral classification.

The rest of the dissertation is organized as follows.

Chapter 2 introduces the concepts and the background of spectroscopy, also listing the most notable techniques. Then it focuses on Raman spectroscopy, the main spectroscopic method used in this research, analysing its characteristics and the features of a Raman spectrum. The components of a typical Raman apparatus are also examined, and then the issues that affect a spectral classification and the state of the art are taken into account.

Chapter 3 introduces the classification approach that must be a merging of pre-processing and machine learning techniques. Concerning the signal processing methods, several baseline and noise removal techniques are listed, while for machine learning, a deeper study on neural networks and their evaluation on spectral classification tasks has been conducted.

Chapter 4 introduces the innovative method both on the acquisition phase and on the dataset generation phase. Concerning the acquisition phase, the sources of noise in a CCD are investigated, and an acquisition strategy that optimize the signal-to-noise ratio has been demonstrated. Then, the chapter describes in details the hierarchical augmentation pipeline with the additional enhancement of Generative Adversarial Networks.

Chapter 5 focuses on a real use case study that concerns art pigment classification, in which all the aforementioned methods are tested with relative plots and programming code. At the end of the chapter, the effectiveness of our method is validated through three different neural network architectures.

Chapter 6 describes the conclusions, along with a summary of the research project, the innovative results achieved, and the future developments of the present research.

Chapter 2

Spectroscopy

Spectroscopy is a discipline that concerns the study and the measurement of the interaction between matter and electromagnetic radiation. The electromagnetic radiation, including what we common refer as "light", is described from a classical point of view by a continuous sinusoidal wave-like motion of the electric and magnetic fields[10][4]. Its fundamental parameters are the wavelength (λ, length of 1 wave), frequency (υ, number cycles per unit time), and wavenumbers ($\overline{\nu}$, number of waves per unit length). These parameters are related with the following formula, where c indicates the speed of light and n the refractive index of the medium:

$$\overline{\nu} = \frac{\upsilon}{c/n} = \frac{1}{\lambda}$$

Following the quantum theory, electromagnetic radiation is emitted from a source in discrete units called photons, where photons energy and photons frequency are related through the Planck's Law (given the Planck's constant h):

$$E_p = h\nu$$

While we refer to the term electromagnetic spectrum as the whole range of frequencies of the electromagnetic radiation (Fig.2.1), with the term spectrum we indicate, in an experimental scenario, the result of a spectroscopic experiment and the subject of a spectroscopic study. A spectrum is basically represented as the intensity of radiation as a function of the wavelength or frequency.

During the doctoral course some different spectroscopic techniques have been investigated, in order to understand the relative features of each class of spectra

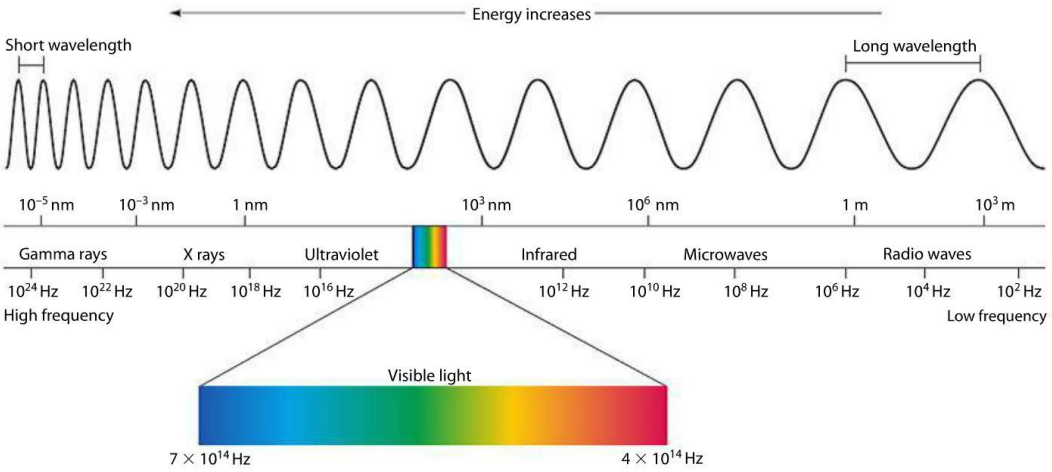


Figure 2.1. Electromagnetic Spectrum (from https://www.miniphysics.com/)

and the information that would result useful for a chemical identification. Generally speaking, we can ideally divide the spectroscopic techniques in two separate sets, following the main characteristics of their produced spectra.

In the first set, the spectra are composed of broad-amplitude bands that can easily overlap among each other, as occurs in Fluorescence Spectroscopy. This type of spectroscopy exploits the fluorescence phenomenon, analysing its intensity, to characterize the vibrational state of the target molecule. The sample is first excited from its ground electronic state by absorbing an electromagnetic radiation generally emitted from a source in the visible or UV range[11]. Once excited, the molecule reaches its vibrational levels in the excited state. Eventually, it begins to drop down to one of the vibrational states of its ground state, emitting photons during this process (Fig.2.2). Since the emitted photons will have different energies, thus different frequencies, analysing their relative intensities it can be determined the structure of the energy levels, and therefore the structural chemical fingerprint of the molecule. If the excitation source is a laser, the technique is named LIF (Laser-Induced Fluorescence).

In the second set, the spectra present themselves with separated and narrow peaks, which indicate the spectral intensity of the emission, and gives us information about the investigated target. Among these types of spectroscopy, we can list XRF, LIBS, Mass Spectroscopy, FTIR and Raman Spectroscopy.

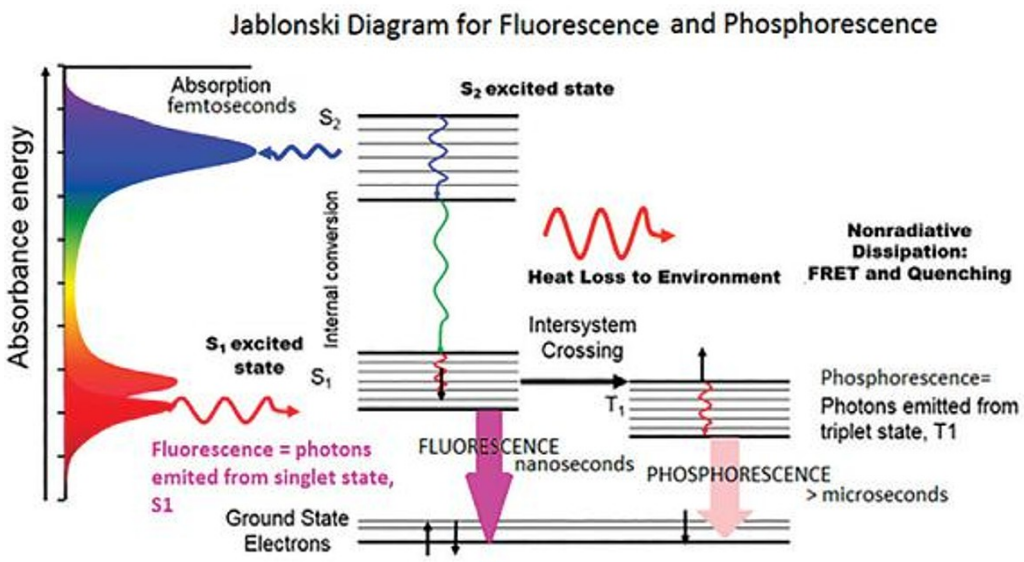


Figure 2.2. The Jablonski Diagram of molecular absorbance and fluorescence (from https://www.horiba.com)

XRF (X-Ray Fluorescence Spectroscopy) is a non-destructive technique based upon the emission of secondary X-rays from a sample that has been exposed to short-wavelength X-Rays or even Gamma rays. The resulting spectrum is characteristic of the investigated element[12].

LIBS (Laser Induced Breakdown Spectroscopy) belongs to the class of atomic emission spectroscopy. It uses an high-energy laser pulse as emission source to generate a plasma within the interaction of the laser source itself and the sample to investigate. The wavelengths of the atomic spectral line in the emission spectrum gives the identity of the element while the intensity of the emitted light is proportional to the number of atoms of the element[13].

Mass Spectroscopy is a technique which exploits the measurement of the mass-to-charge ratio of the ions formed when a sample is ionized. The analysis of the mass spectrum can reveal the chemical identity of the sample since it is correlated to its chemical structure and the ionization conditions[14].

FTIR (Fourier Transform Infrared Spectroscopy) allows to obtain the infrared spectrum of a sample using the Fourier Transform, converting the raw data produced by an interferometer (light absorption for each mirror position) into the frequency

domain (light absorption for each wavelength)[4].

An exhaustive study or comparison of the aforementioned techniques is beyond the scope of the present thesis. On the other hand, Raman spectroscopy is dealt with in detail in section 2.1, due to its relevance in this research and in the relative use cases.

2.1 Raman Spectroscopy

Raman spectroscopy is the main spectroscopic technique treated in the present doctoral research. It allows to detect rotational or vibrational modes of target molecules in order to determine their structural chemical fingerprint[4]. It relies upon the inelastic scattering of photons (a.k.a. Raman effect), a process that causes a difference in energy shifting between the incident and the emerging photons resulting in an increase or loss of energy that gives us information about the structure of the sample. In fact, this difference in energy shifting is strictly related to the vibrational and rotational levels of the molecule, which uniquely characterize the investigated chemical elements.

In detail, when a source of monochromatic light (typically a laser) interacts with a target sample, this amount of light can be reflected, absorbed or scattered (i.e. deviated from a straight trajectory) in some manner. If we analyse the scattering inducted from a light source, we observe that most of the radiation is elastically scattered by a dipole (Rayleigh Scattering). In scattering phenomenon, the incident photon is absorbed by a transition from the ground state into a virtual state and a new photon is created and scattered by a transition from this virtual state. In Rayleigh scattering the diffused radiation has the same energy (and hence the same wavelength) of the incident light.

In Raman scattering, instead, this scattered photon results from a rapid decay ($10^{-12}s$) from the virtual state to the excited states of the molecular vibration. This is described as an inelastic collision between photon and molecule, since the molecule acquires different vibrational energy and the scattered photon now has different energy and frequency. The characteristic energy gap between the two states turns into the radiation shift, according to the Planck's Law.

If the energy gap is negative (i.e. the molecule starts from the ground state and ends in an excited state) the energy of the emitted photon will be red-shifted with respect to the incident photon (Stokes lines). On the other hand, if the molecule comes from an excited state and returns to the ground state, the emitted photon will have higher energy than the incident one, and will be blue-shifted compared to it. A schematic representation of these processes is depicted in Fig. 2.3, considering that the electromagnetic field perturbs the charged particles of the molecule according to the following equation:

$$\mu = \alpha E$$

where α is the polarizability, E the incident electric field, and μ the induced dipole moment.

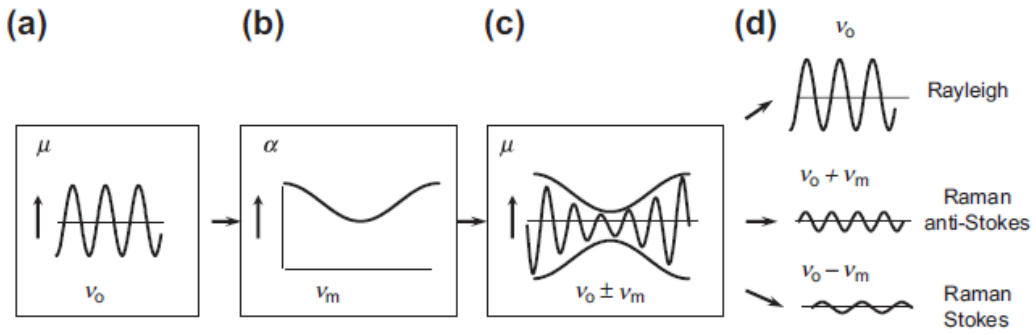


Figure 2.3. Schematic representation of Rayleigh and Raman scattering [1]. In (a) μ oscillates at the incident photon frequency, in (b) the molecular vibration can change α, while in (c) the final oscillation results in an amplitude modulated dipole moment, where the components (d) emit electromagnetic radiation.

Since every chemical compound absorbs and releases energy following its particular vibrational state, if we measure the shift between the incident and the back-scattered frequency, we can determine the structural fingerprint of the compound, and hence we can identify it. Compared to the Rayleigh scattering, the Raman scattering is a weak phenomenon, whereas only one in every $10^6 - 10^8$ photons is scattered. The intensity of the Raman scattered radiation I_R can be calculated as:

$$I_R \propto \nu^4 I_0 N \left(\frac{\partial \alpha}{\partial Q}\right)^2$$

where ν is the frequency of the exciting laser, I_0 the intensity of the laser, N the number of scattering molecules in a given state, α the polarizability of the molecules, and Q the vibrational amplitude.

The result of a Raman scattering measurement is the Raman spectrum which is discussed in detail in section 2.2.

2.2 Raman Spectrum

As mentioned at the beginning of the present chapter, spectroscopic data are often represented by a spectrum, i.e. a plot of the intensity of radiation as a function of the frequency or wavelength. Following this definition, a Raman spectrum is a plot of the intensity of Raman scattered light (y-axis) as a function of its frequency difference from the incident radiation (x-axis). This difference is called Raman shift, and it is expressed in units of wavenumbers [cm^{-}1] (less often in Δcm^{-}1). The reason for using this type of unit of measure is because it is proportional to the energy (given the Planck's Law $E = h\nu$) while it results invariant to the incident wavelength. An example of Raman spectrum is shown in Fig.2.4.

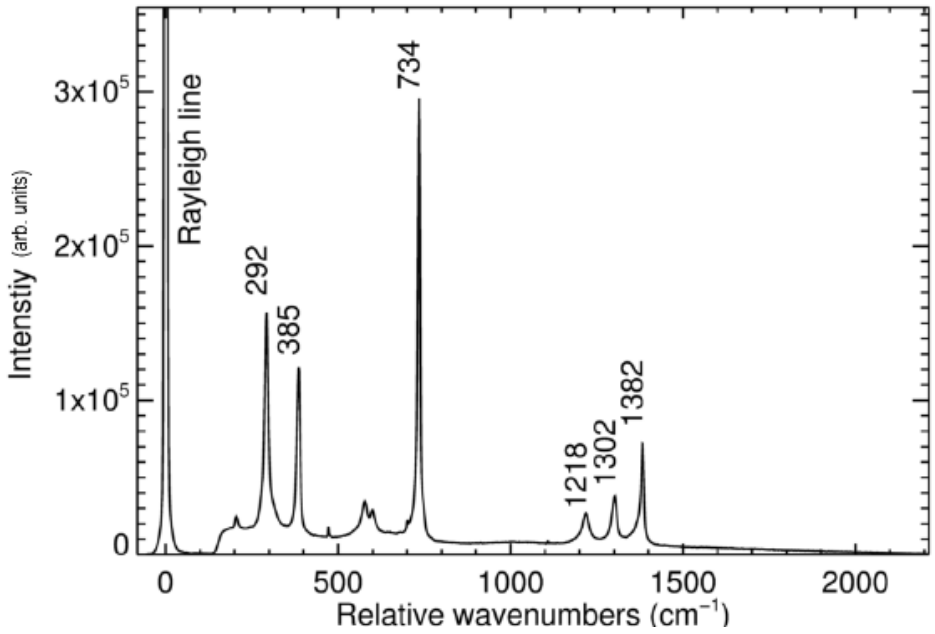


Figure 2.4. Raman spectrum of Teflon $[C_nF_{2n+2}]$[2]

Since the intensity of Raman radiation is weaker than Rayleigh scattered light, every modern Raman acquisition apparatus is equipped with a notch or edge filters to cut off the elastic straylight that could mask the Raman spectrum. Hence, normally, a Raman plot does not show the Rayleigh radiation, but only the bands at lower frequencies than the incident frequencies (Stokes band), or those at greater frequencies than the incident light (Anti-Stokes band).

Given that each chemical element has its characteristic rotational-vibrational state, each chemical compound can show one or more points of high intensity at specific wavelengths in a Raman spectrum[1]. Hence, each material can be represented by a spectrum with one or more intensity peaks. The identification of the compound characteristic peaks are the basis of Raman classification, whereas a comparison of the acquired spectrum with a dataset of known reference spectra remains the main method to classify the investigated sample.

From a Computer Science point of view, an acquired Raman spectrum is basically represented as a numerical array, where the index of each element corresponds to the Raman shift, while the value pointed by that index corresponds to the relative intensity.

2.3 Components of a Raman LIDAR Apparatus

In this section a brief summary of the components and the configuration of a Raman Lidar System is presented, in order to explain the technical features that allow Raman spectra acquisition.

A LIDAR (Light Detection and Ranging) system is an apparatus that uses a laser source aimed to a target and measures the reflected signal with a sensor. The difference in laser return times can be used to detect the distance from the target. On the other hand, the acquired spectrum, related to the specific spectroscopic method, can reveal chemical composition information of the target as in this use case study. A Raman LIDAR system makes use of the inelastic scattering of light to characterize the chemical fingerprint of the investigated sample. Raman LIDAR

[1] Not all molecules are Raman-active: Raman and mid-IR spectroscopy, as complementary techniques, are usually required to completely measure the vibrational modes of a molecule[4].

systems have become widely used since the past decades for a multitude of tasks [15] [16] [17].

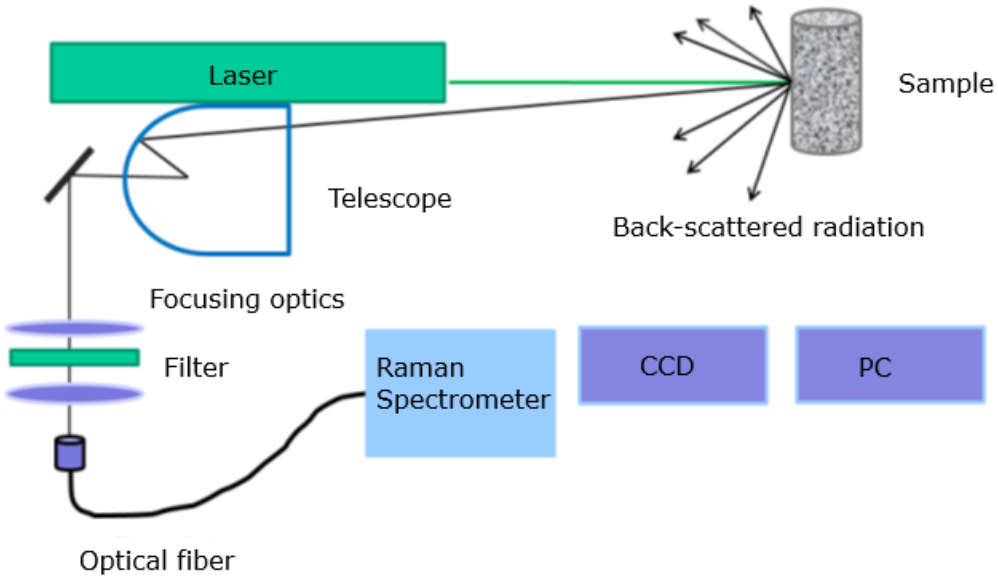


Figure 2.5. Schematic representation of a Raman LIDAR apparatus

In figure 2.5 a schematic representation of the components of a Raman experimental apparatus is depicted. The first component of a Raman apparatus is the laser source. Raman measurements do not require specific laser wavelengths. However, the choice of the laser wavelength must be carefully handled for several reasons: the generally small Raman scattering cross-section, which is proportional to λ^{-4} (with λ denoting the wavelength of the laser source radiation), and the super-imposed fluorescence signal (both from the sample and the substrate) that possibly covers the weaker Raman signal. If short wavelengths ($250 - 400\,nm$) are used to increase the Raman cross-section, fluorescence may become a problem. In theory, the spectral range below 270 nm can be used as a relatively fluorescence-free signal, since most fluorescence spectra start from about 270 nm. However, in this region the absorption spectra may show sharp increases due to resonance, and no absorption below the resonance [18]. This causes non-linearity in responses among different compounds, and limits the number of detectable species, since every compound may show resonance at a different wavelength. To overcome the problem of fluorescence, a variety

of techniques have been set over in the past few decades [19]. Since the Stokes spectrum is the one that it is normally observed in standard conditions (i.e. when molecules are in the fundamental state) and being itself red-shifted with respect to the Rayleigh scattering, it may be often overlapped to the fluorescence spectrum. If the impact of the fluorescence spectrum is not too high, we can implement baseline removal algorithms (as explained in details in section 3.2), otherwise the intensity of fluorescence can totally mask the Raman spectra. Another reason that could occur in some scenarios regards the further radiation that is possibly requested for the excitation of the energetic levels entering the Raman emission, causing photo-degradation of the sample[20].

The chosen laser radiation is then collimated towards a target that, in case of a Raman spectroscopic measurement, requires minimal or no sample preparation at all. However, in a Raman measurement set-up, one of the caveat is the choice of the best trade-off between laser power and wavelength because of the safety issues and the aforementioned risk of sample degradation. This activity is dealt in detail in sec. 4.3.1.

The back-scattered light is then collected through an optical system that varies along with the experimental conditions. The aim of this component is to collect the Raman scattered light and pass it to the spectrometer. As it was explained in section 2.1, since the Rayleigh scattering is much more intense than the Raman one, a Notch or long-pass optical filter is always equipped by the apparatus to filter out the unnecessary Rayleigh bands and obtain a high quality Raman measurement.

Then, the spectrometer receives the filtered radiation and disperses it in order to pass it to the detector. A polychromator equipped with a diffraction grating is typically used to sort spatially the polychromatic radiation into monochromatic components within the spectrometer (Fig. 2.6).

In modern Raman acquisition systems, the dispersed light is then measured with a CCD (Charge Coupled Device), an integrated circuit containing a grid of semiconductor elements. The aim of this component is to collect, transfer and convert the electrical charge generated by the inbound light. The number of electrons transferred by these devices is proportional to the intensity of inbound light, with a probability

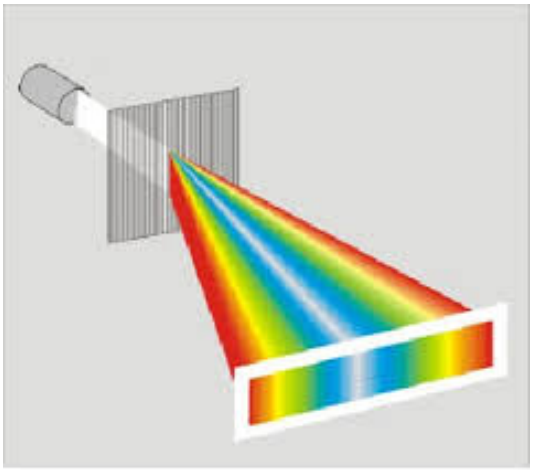


Figure 2.6. Representation of a diffraction grating
(from https://physicsopenlab.org/)

that represents the Quantum Efficiency of the CCD. The use of a CCD is not the only detection approach, but it allows to acquire the whole spectrum in a single acquisition. CCDs are discussed in details in section 4.1.
The final component is a Computer, equipped with a specific software, that is connected with the CCD in order to visualize, manipulate and store the data.
In fig.2.7 it is depicted a schematic representation and a 3D model of RADEX system, an innovative Raman-based apparatus for proximal detection of energetic materials on people [3].

2.4 Spectral Matching

As explained in section 2.2, a Raman spectrum of a given chemical compound is characterized by a set of high intensity peaks at specific wavelengths, that uniquely identify the investigated material. So, each chemical compound can be represented by a specific Raman spectrum that reveals at least one high intensity peak.

Therefore, a Raman spectral classification problem presents itself as an apparently simple task: given a reference dataset of Raman spectra, a matching between the acquired spectrum and a known reference one can be performed to obtain the spectral classification of the sample. In other words, if a clean spectrum (i.e. with a low noise level) is acquired, we can identify its peaks and compare it with known

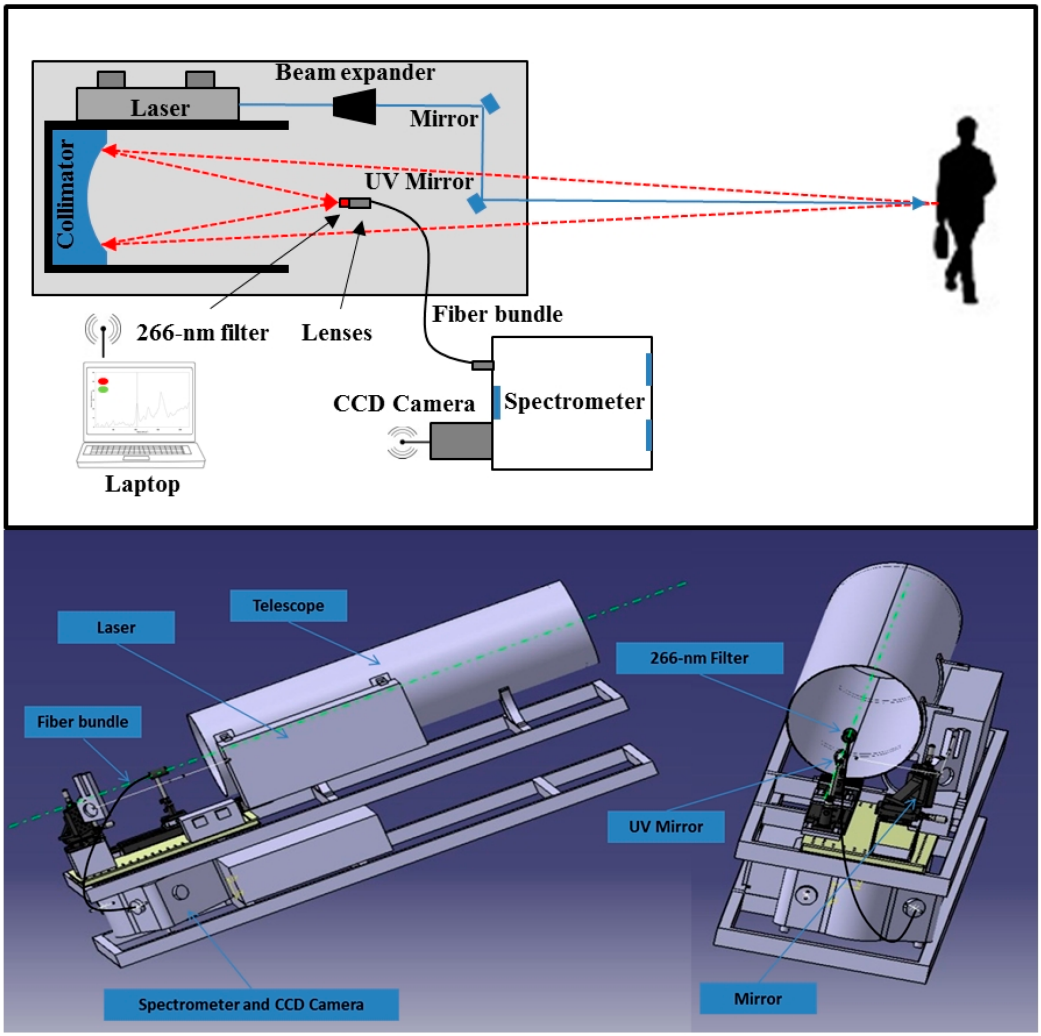


Figure 2.7. Schematic representation (above) and 3D-model (below) of RADEX system [3]

spectra in a database that show the most similar peaks distribution. However, this apparently simple task is affected by a multitude of issues that inevitably increase the degree of difficult of this problem's resolution, concerning both its correctness and its accuracy. In Figure 2.8 a comparison of two Raman spectra of the same material (Teflon) is presented, in order to highlight the differences that could occur in two measurements from different instrumentation. It is then self-evident that using a dataset of standard spectra as a reference in a Raman matching problem is often practically unfeasible, especially for measurement in non-laboratory scenarios, where acquiring a reference dataset from scratch has become a common practice.

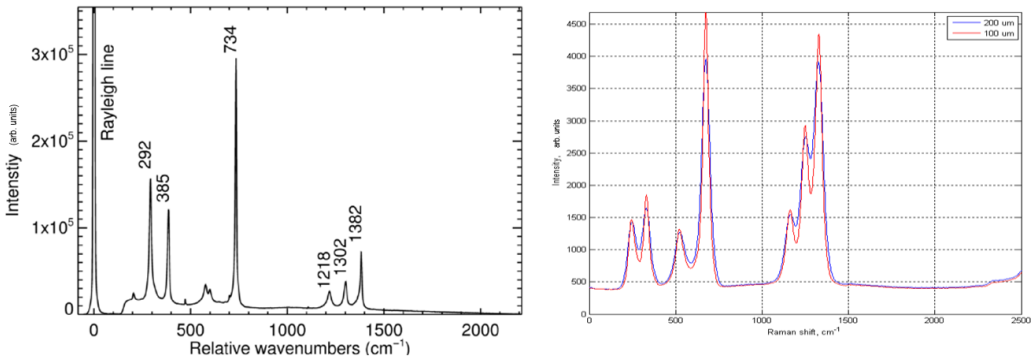


Figure 2.8. Stokes bands of two Raman measurements of Teflon. On the left a reference spectrum found in literature [2]. On the right two experimental acquired spectra: the red one is acquired with an optical slit of $100\mu m$, the blue one with a slit of $200\mu m$

The features that may affect the correctness of a Raman spectral matching are therefore many and could depend on different acquisition conditions: laser source wavelength, number of laser pulses, acquisition integration time, light environment conditions, pressure, temperature, specific sample set-up, presence of contaminant substances, and many others. These features could alter both the intensity scattering effect and the position of the characteristic Raman peaks (i.e. peak shifting). Further, the amplitudes of the peaks are subjected to variations due to the different source wavelengths, since different energies excite different rotational-vibrational modes.

In addition, in cases where a mixture of more than one substance is investigated, it can occur the possibility of Raman peaks overlapping, with one or more peaks present at the same value of wavenumber. It is also been observed that the mixture spectrum is not qualitatively proportional to the sum of pondered individual spectra [21]. Further, a strong background signal of a measurement could hide the characteristic peaks of a molecule, especially where the presence of fluorescence significantly alters the spectrum baseline with its characteristic slope. The fluorescence phenomenon often occurs in presence of organic materials, but it can also be observed in inorganic materials. A Raman spectrum with a significant component of fluorescence is shown at Figure 2.9. These kind of spectra show an intrinsic Poisson distribution (i.e. the probability of a given number of events occurring in a fixed interval of time/space if

these events occur with a known constant mean rate and independently of the time since the last event) that could mask the main signal, even after the subtraction of the mean value. In other words, the subtraction of fluorescence component does not guarantee the reveal of the main signal.

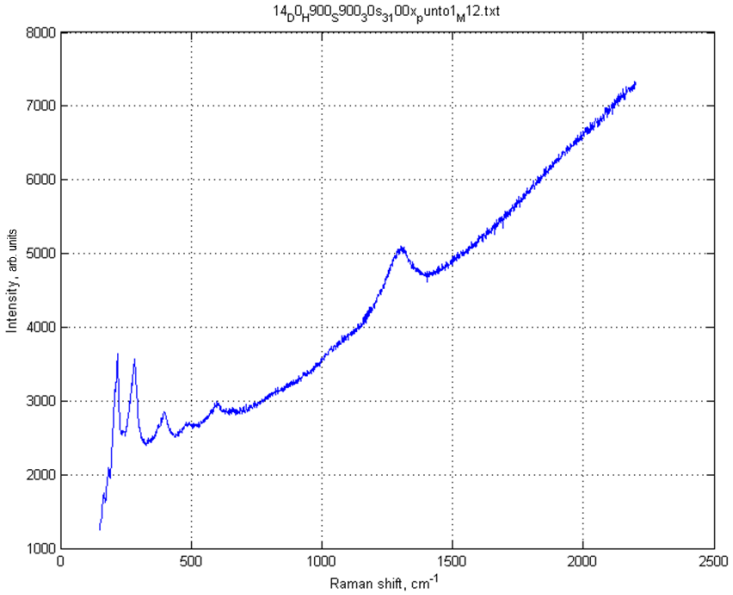


Figure 2.9. An experimental acquired Raman spectrum that presents a significant component of fluorescence

Finally, as it will be explained in the next chapters, the amount of noise that can affect a Raman measurement must be kept at a low level in order to avoid further peaks overlapping, or mismatching between characteristic and noise peaks. Since spectra can be mathematically represented as uni-dimensional vectors, the task of Raman classification can be viewed as finding the smallest distance between two vectors (in this case, a vector that represents the acquired spectrum, and a vector that represents the reference spectrum). Two of the most common methods to perform this measure are the Euclidean Distance and the Dot Product (i.e. the correlation coefficient that corresponds to the cosine of the angle between the two vectors). Euclidean Distance of two vectors $P = (p_1, p_2, ..., p_n)$ and $Q = (q_1, q_2, ..., q_n)$ is defined as:

$$d(P, Q) = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

The correlation coefficient expressed as a Dot Product of two vectors P and Q is defined as:

$$r = cos(\theta) = \frac{P \cdot Q}{||P|| \cdot ||Q||} = \frac{\sum p_i \cdot q_i}{\sum p_i \cdot \sum q_i}$$

We will give some examples of how these methods could not work properly in our case of study, regarding the aforementioned issues given by noise, fluorescence and peak shifting. For instance, acquiring a noisy spectrum and performing a vector distance with a reference spectrum appears immediately as an incorrect technique, since the fluctuations of noise at each value of wavenumber give as result wrong distance values. Furthermore, regarding the peak shifting issue, even a slight contraction or expansion of the signal with respect to the x-axis makes the Euclidean Distance method inefficient, whereas a characteristic peak shifts around a certain range increasing the distance and decreasing the vector correlation. Finally, the presence of a fluorescence slope, or a sample contamination with one or more substances, can lead to the sum of unwanted peaks and intensity values that make inconsistent the final correlation. For these reasons, traditional vector matching methods are ineffective especially for experimental case of studies, where the possibilities of acquiring noisy Raman spectra are concrete.

2.5 Spectral Classification Background

The origins of modern spectroscopic studies begin with the German physicist Joseph von Fraunhofer (1787-1826). He developed the first modern spectroscope in 1814, combining a prism, a diffraction slit and a telescope in order to analyse the different components of the visible light in a spectral resolution never achieved before[22]. Eventually, he also developed a diffraction grating, an optical element that splits and diffracts the incident light, sorting spatially the poly-chromatic radiation into monochromatic components. During his experiments about spectral line detection, he discovered that the spectra of the sun and the other stars differed from each other, concluding that the spectra carried on information about the sources of light generated from those celestial objects, regardless of the distance of the objects. This is generally considered the birth of the astronomical spectroscopy,

a field of study still very widespread nowadays.

The years between 1920 and 1930 saw many innovations in the field of spectroscopy. The English physicist Cecilia Payne-Gaposchkin demonstrated in her 1925 doctoral thesis that all stars have very similar relative chemical abundances with hydrogen and helium comprising 99% by number, resulting as the most abundant elements in stars and, therefore, in the universe. This result was achieved observing the correlations between temperature and chemical compositions in the stellar spectra.

The Polish physicist Aleksander Jabłonski described in his 1930 doctoral thesis the relaxation mechanisms for excited state molecules. These studies led to a deeper understanding of the phenomenon of fluorescence, i.e. the emission of light (mostly at a different wavelength) by a substance that has absorbed an electromagnetic radiation.

In 1928, after years of published studies, the Indian physicist C.V. Raman discovered the inelastic scattering of photons that, in the following years, would be named after him [23]. With the discovery of the Raman effect, a new spectroscopic technique, i.e. Raman spectroscopy, started to be exploited by physicists and chemists in order to collect information about material identification and analysis for a variety of purposes. Raman spectroscopy is the main technique that has been used in the present research for spectra acquisition (section 2.1).

Spectral classification has been, therefore, not only an important research topic but also a challenging problem since many years. The chemical identification based on the analysis of a spectrum must be related to the specific spectroscopic technique used for sample investigation. In fact, each type of spectroscopy produces spectra of different characteristics, from the point of view of signal analysis, and provides different physical information.

However, even taking into account only one type of spectroscopy (e.g. Raman Spectroscopy), the classification problem is still far from a unified solution. In fact, methodologies change along with the data, resulting in a multitude of techniques that span from statistic science, signal processing, artificial intelligence, and many others. In addition, for non-laboratory scenarios, many sources of variability tend to

affect a correct sample identification: two measurements of the same substance can lead to two acquired spectra of different characteristics, due to various conditions or different equipment. The difficulty of the classification tasks further raises in the presence of mixtures of substances or thin layers of substances deposited on a substrate. In Fig.2.10 an illustrative scheme of the ideal skills that a modern spectroscopic analyst should own is depicted.

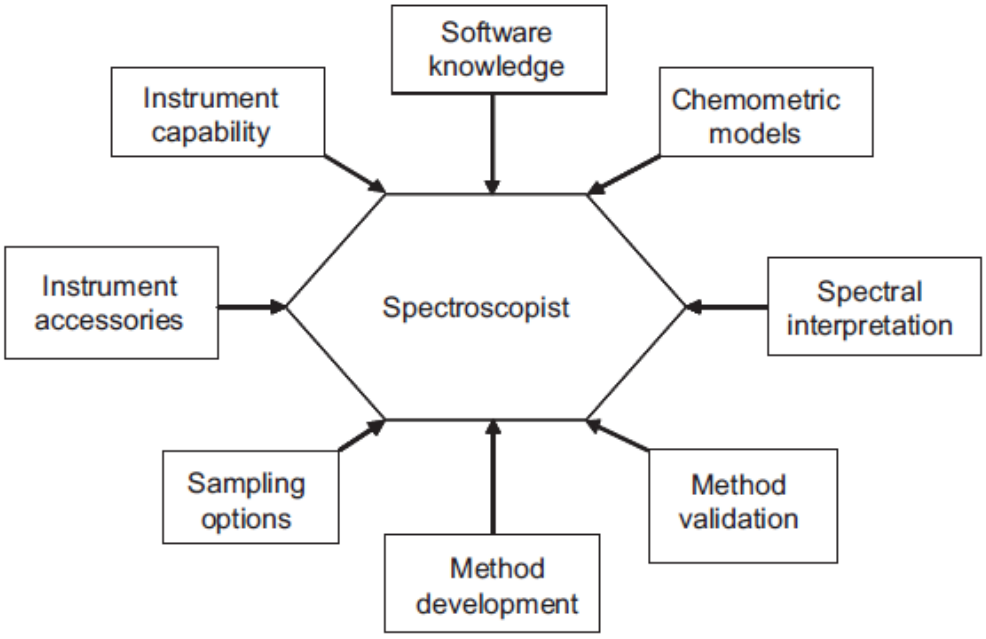


Figure 2.10. Skills required for a successful spectroscopist [4]

Even at the present day, spectral classification cannot be configured as an automatic process, but it is often a task led by human supervision, where the expertise of the specific domain owned by the researcher is necessary to accomplish the sample identification. The specialists must still rely on large reference datasets, in order to perform comparisons and correlations between acquired and reference spectra.

Hence, the developing of an automatic spectral classification tool can bring innovation in many fields of application, and a real-time identification feature could make a device ideally suitable for a vast range of non-laboratory scenarios. Here is a list of some possible application fields of spectral identification:

- Medicine: it can be used in a multitude of medical examinations for screening and diagnosis purposes [24][25].
- Biology & Chemistry: it can be used in order to characterize the chemical/ biological sample that needs to be investigated[26][27].
- Industry: it can be used for material quality controls and other analysis tasks[28].
- Security and Forensic Sciences: it can be used to discover explosives, drugs or other hazardous substances[29][3].
- Cultural Heritage: it can be used, for instance, to detect a specific pigment leading to identify authors, art studios, epochs, re-touchings, and other features[30][31].

2.6 State of the Art

If spectral classification methods originate along with the discover of the related spectroscopic techniques, the automatic identification of chemical species from their spectral data is a research field that was born in 1960s. Since the 1980s, with the widespread use of computers and the increasing of their computational power, more sophisticated techniques and algorithms have been tested in order to achieve the best classification accuracy with a minimal human assistance. In the 2010s the growth of Artificial Intelligence in the field of Computer Science applications, and in particular the exploiting of Deep Learning methodologies, has boosted the attempts to an automatic spectra classification through Machine Learning techniques. In the following paragraphs some of the most notable researches regarding spectral classification tasks are presented, highlighting their milestones and their critical issues.

In [32] (1984), two Pattern Recognition techniques have been performed, both of them exploiting unsupervised learning (i.e. learning from unlabeled data). First a clustering, using the matrix of dot products between the vectors. Then, a PCA (Principal Component Analysis, a dimensionality reduction technique) has been

applied to the dot products matrix. As a result, the metric of dot products appears encouraging but the investigated spectra belong to a standard reference library where the spectra have been acquired in optimal laboratory conditions.

In [33] (1986), a Pattern Recognition technique with a SIMCA training set (Soft Independent Modeling of Class Analogy) has been used to solve a mass spectroscopy classification task. A binary encoding for each mass spectrum has been performed (0 or 1 if a peak is present in the channel), then the 17 results with the highest binary information have been selected as training set (17 out of 78). The results show that the data structure is divided into two chemical super-classes and 4 sub-classes, while the classification accuracy is 96% super-classes and 82% sub-classes.

In [34] (1992) a neural network with a back-propagation mechanism has been used, whereas the spectral data to be entered as neural network input are chosen by selecting the most significant infrared spectrum range. The result with 2 hidden layers are 62% accuracy. One important concept is the intuition that the use of neural networks for infrared spectra is restricted to the recognition of functional groups, while for more complex problems more sophisticated algorithms and/or neural networks are needed.

In [35] (1996) a measurement of acoustic back-scatter coefficients to identify fish species has been performed. The first step is an average of the frequency spectra of the measurements, while a subsequent analysis uses alternatively neural networks (1 hidden layer) and discriminating analysis (i.e. method used to find linear combinations of features that characterize or separate two or more classes of objects). The encouraging results reach a 95% accuracy in both methods.

In [36] (1993) and [37] (2001), the studies deal with the discipline of cytometry, i.e. a laser technique used for counting, separation and detection of cells. In particular, there is a state-of-the-art study of clustering problems and identification of flow cytometry mainly using two different techniques: neural networks and analysis of statistical clusters. The results show that, despite the large amount of data, with neural networks it could be also possible a real or near-real time identification. The problem is the construction of the training set which requires a prior statistical reduction given the «almost infinite» variety of cell types present in the samples.

In [38] (2009) there is a study about automatic identification of traces of explosives deposited on adhesive tape through an analysis of Raman spectra in the region $0-300$ cm^{-1}. The spectra are acquired in laboratory conditions with high signal/noise ratio and $10s$ integration time, with the additional use of a Kaiser filter in frequency domain to eliminate fluorescence and noise. The classification is obtained through an internal product between the filtered spectrum and the reference one. The results show that there is a promising performance for some types of explosives, but many false positives remain (a further study of ROC curves allows to minimize the latter). The critical issue is that the method is not promising for noisy signals.

Studies that concern the exploitation of the Raman effect for molecular identification have continued to this day in various fields of application as stated in section 2.5. The difficulty is the lack of a unified and reliable method for spectral classification ([39], [40], [41], [42], [43]). In fact, none of these proposed methods have resulted to be an ultimate solution for automatic Raman spectra recognition, in particular for specific problems where large reference datasets are not available. Further, most of the researches focus on acquired spectra in optimal laboratory conditions, while the processing and classification of signals obtained in «real» conditions is still a significant problem. Finally, these processes are often computationally hard, and an automatic spectra classifier that implements a cutting-edge feature like real-time classification would be of a great interest for the scientific community. In the next chapter an innovative approach that exploits both signal processing and machine learning tools is introduced.

Chapter 3

Real-Time Classification Approach

The task of Raman spectral classification can be reduced to a vector similarity problem, whereas the closest correlation between a reference spectrum and an acquired spectrum indicates that they belong to the same chemical compound. However, as explained in the previous chapter, this approach carries on many critical issues and it could be considered reliable and accurate only in laboratory scenarios where clean spectra are acquired at the most profitable conditions. Regarding other experimental scenarios, two different approaches have achieved notable results overall the last years. The first approach uses a variety of signal processing tools in order to filter any type of interference from the acquired spectra so that it can be matched with the reference library. The second approach exploits Machine Learning tools that, in the last years, have proved to be capable in achieving high accuracy rates in similar classification tasks. As it is explained in the next paragraphs, the present research tries to find an approach that can be considered a merging of these two fields of study, in order to develop a real-time spectral classifier that ensures the highest degree of accuracy.

3.1 Pre-Processing vs Machine Learning

In the design of an hypothetical real-time Raman spectra classifier system, we must determine the main requirements that will lead the development of this system. Further, these requirements must be focused both on the expected objectives that the system is supposed to reach and on the possible experimental scenarios where the classifier is supposed to operate. In the following, a list of high-level requirements is enumerated. Eventually, in the next sections, we will go into detail among low-level requirements, in particular technical requirements for the acquisition device and software requirements for the classifier development. The main high-level requirements that a real-time Raman classifier is supposed to own are:

- modularity of the system. The system must be composed by several independent modules that may be separated or combined in order to gain flexibility and adapt itself at the different operational environments. The modules could implement signal processing tools as well as machine learning algorithms.

- automatic data tuning. The system must recognize and deal with Raman spectra acquired at any possible conditions, so that it is expected to operate well with clean spectra as well as with noisy spectra, or in the presence of substrate and mixtures of compounds.

- a minimal end-user interaction. The user is supposed to supervise the system operations and not to be a technical expert that must re-design and customize the classifier at each operational scenario.

- fast evaluation. The system is supposed to operate in real-time or near real-time, sending classification feedback to the users along with the predicted accuracy.

- lack of reference library. The system is supposed to work properly even in the case that a reference spectra library cannot be available, or in the case that reference spectra are acquired *in-situ*.

- reasonable computational resources. In the case of a future developing of a portable Raman classifier device, the system must adhere to reasonable

technical and computational requirements, both for the acquisition-detection device and for the spectra processing software.

If we analyse separately the features of a signal processing software application and a machine learning tool, we can conclude that none of these two applications can individually satisfy the aforementioned requirements. Hence, our innovative approach must take into consideration the combination of the two processing strategies in order to benefit from the different features that each technique allows. In fig. 3.1 a schematic representation of the designing approach is depicted.

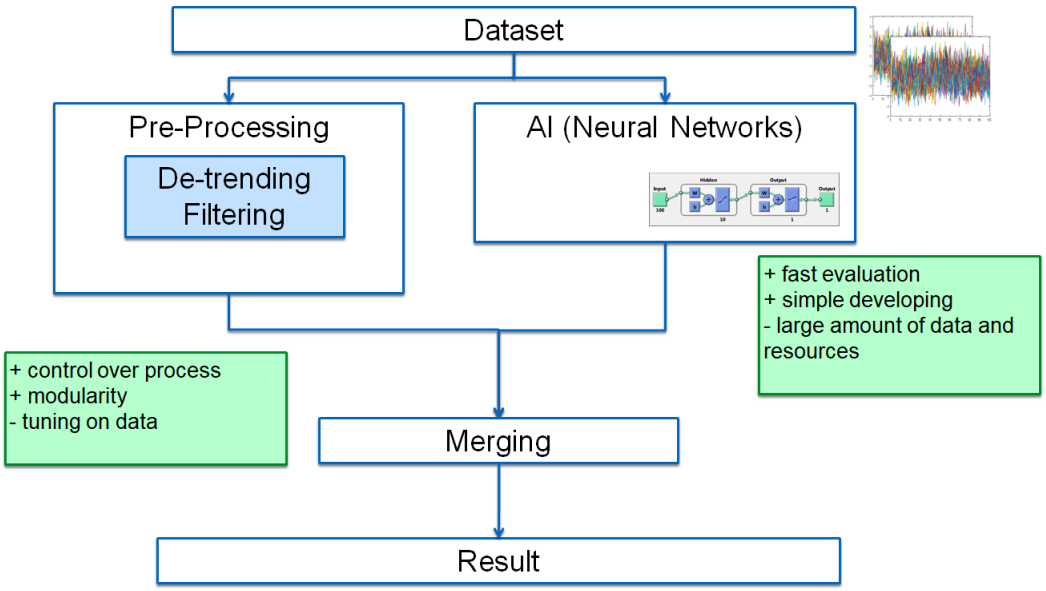


Figure 3.1. Pre-Processing vs Machine Learning approach

Starting from a dataset that needs to be classified, on the left side of fig. 3.1 there is a pre-processing pipeline approach that exploits the filtering properties of signal processing tools. The advantages of this strategy is that the operator has a strict control over the entire process, which can be divided in subsequent steps in order to finally obtain clean spectra at their lowest level of noise and interference. In these type of systems, the modularity is related to the different software processing tools implemented by the system, so that the operator can choose which set of tools must be used. One of the disadvantages is that this approach does not allow to perform a unique strategy over different sets of input spectra, but it rather needs

a data tuning on each different set of input spectra, resulting in a time-consuming task from the point of view of the operator.

Instead, on the right side of fig. 3.1, an approach that uses one of the many machine learning tools such neural networks is represented. The advantage of this strategy is that these tools require a minimal developing and deploying compared to signal processing tasks, so that the operator can immediately feed these applications with the acquired data. Another advantage is the fast evaluation that these systems ensure, which make them perfectly suitable for a real-time application. The main disadvantage of neural networks (that are explained into details in sec. 3.4) is the preliminary training phase in which the network learns how to classify the input data. In order to reach a valuable accuracy during the evaluation phase, the network must be trained with a large amount of input examples which cannot always be available. Lastly, the computational requirements of most of the machine learning tools are higher than the standard signal processing tools.

To summarize, only a classifier system that merges the best features of these two strategies can achieve valuable results in the field of real-time spectra classification. The innovative approach of the present research will be implemented as a hierarchical pipeline of algorithms which, instead of being run at a pre-processing phase, are used to generate an input set that results suitable for a neural network classifier. In this manner, the disadvantages of both approaches are being reduced, while the potentialities are being exploited at their best.

3.2 Pre-Processing: Baseline Removal

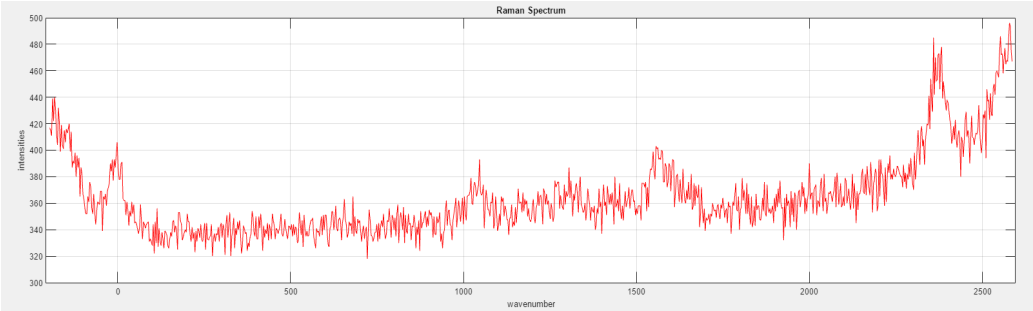
In this section we start to analyse several pre-processing algorithms that will be used in our final merged approach. As explained in section 2.4, the presence of a strong background signal is one of the issues that affect Raman experimental measurements. Raman spectra can be generally represented as a number of peaks and noise superimposed on a background (baseline)[44]. This baseline can be either flat or linear, with a positive or negative slope, and it may also be orders of magnitude stronger than the Raman signal, so its removal is a common task for any analysis of Raman data[45]. Many causes may alter significantly the spectrum baseline:

the presence of background illumination, a strong substrate signal, or even the residual Rayleigh scattering at low Raman wavenumbers. Yet one of the most significant is the fluorescence phenomenon, which presents a characteristic slope that could mask relevant peaks for the classification. Though many instrumental approaches and device hardware modifications have been applied to solve this task, nowadays computational methods have become the standard way to correct the contribution of fluorescence from the raw Raman signal[46], provided that Raman peaks show sufficient intensity to emerge above the noise. Baseline removal is then a task that must be handled carefully: since signal intensity S obeys to Poisson statistic, it carries a noise component that corresponds to the standard deviation $\sqrt{S}$. If the intensities (i.e. in this case, the Raman peaks) are comparable with the standard deviation, a baseline removal could also remove characteristic peaks that are necessary for the compound classification.

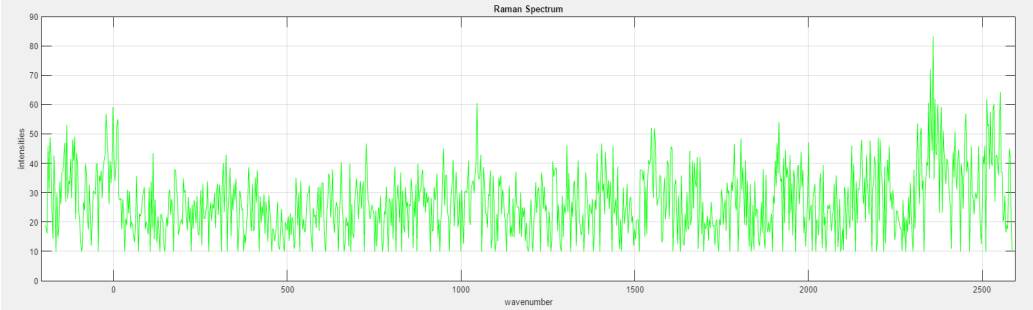
The following list presents a short description of the methods used in the present research, highlighting for each one of them the parameters that can be tuned. In fig.3.2 and fig.3.3 an example of processed spectra for each baseline removal algorithm is depicted. They have been chosen mostly for their popularity in the literature[47]. Baseline removal can be also performed using other type of techniques (e.g. linear regression), and each one of them could be possibly added as an independent component if the chosen method would demonstrate to improve the effectiveness of the final classifier. The baseline removal algorithms selected for this research have been proved to properly fulfill the objectives of the present use case.

Rolling Ball [48] - A frequency differentiated non-linear digital filter. It can be described as the mathematical equivalent to rolling a ball below the spectra and marking out the background as the locus of the ball at each moment. The variable parameters are *radius* (i.e. the radius of the ball running under the graph) and *deltax* (i.e. the spacing of sample points).

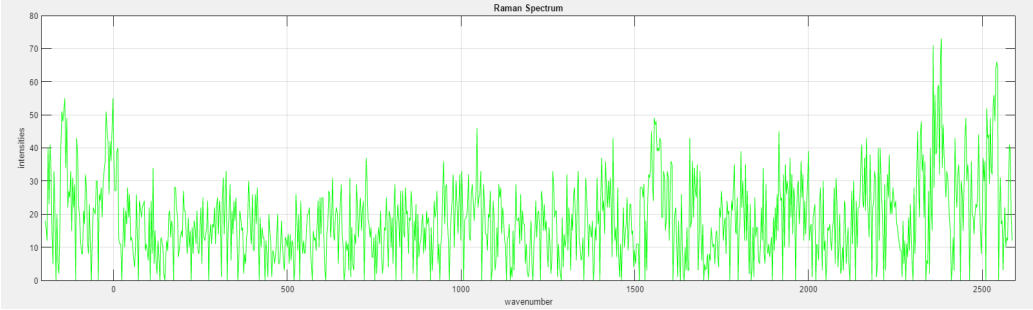
Rubber Band [49] - An algorithm that divides the spectrum into n ranges of equal size. Then, the minimum y-value of each range is determined. Connecting the minima with straight lines creates the baseline. Starting from below, a rubber band



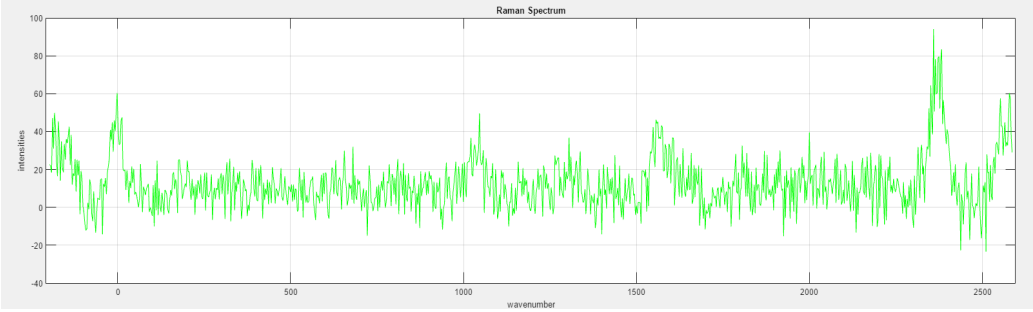
(a) Raw Spectrum



(b) Processed spectrum with Rolling Ball algorithm

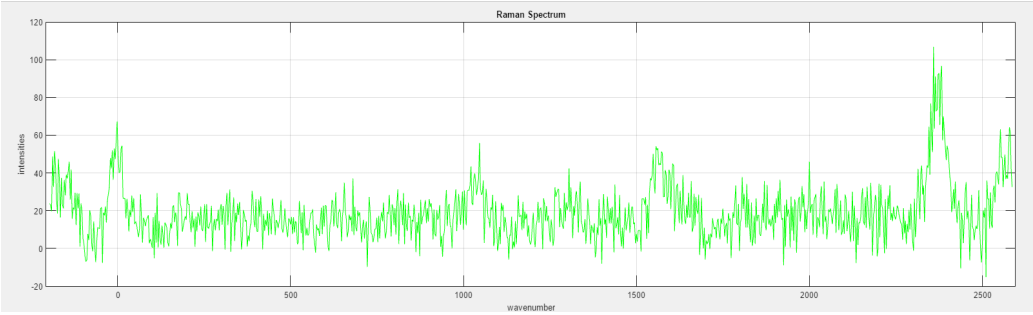


(c) Processed spectrum with Rubber Band algorithm

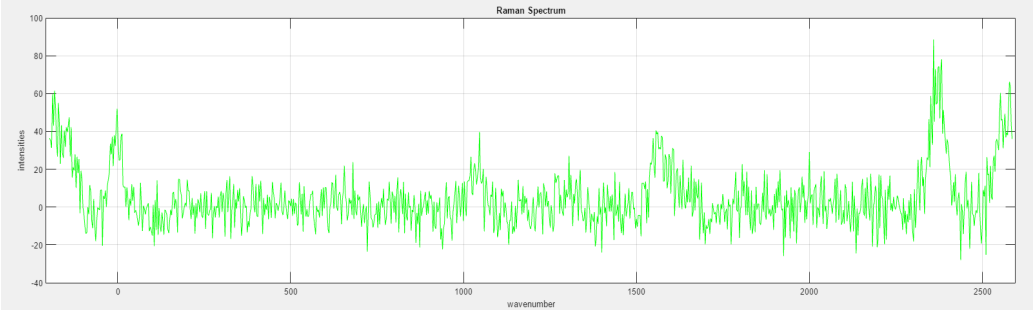


(d) Processed spectrum with AsLS algorithm

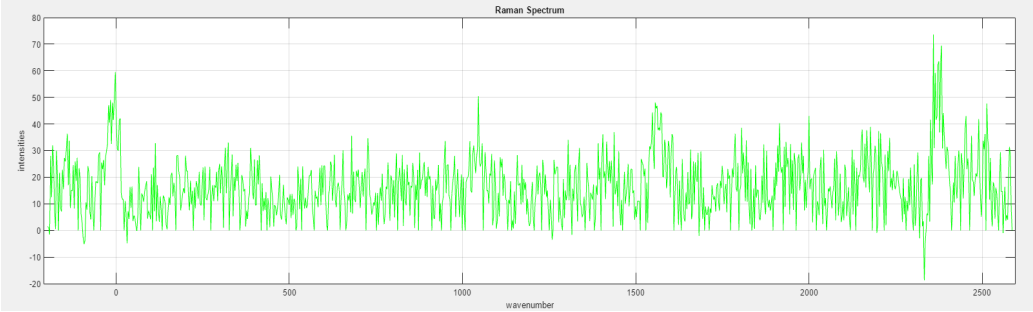
Figure 3.2. Examples of Baseline Removal algorithms (1of2)



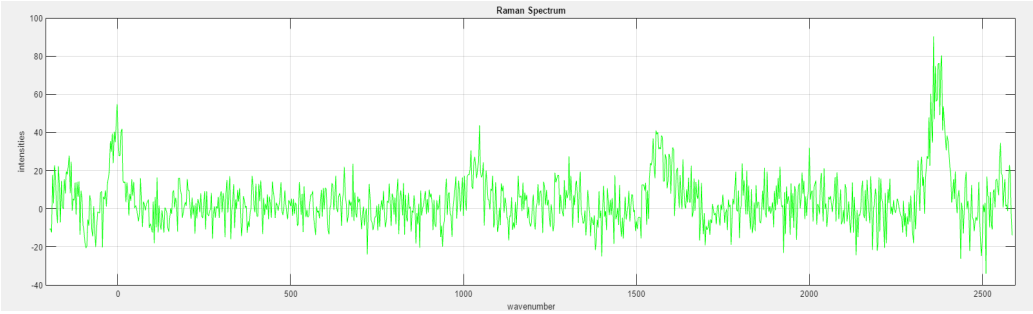
(a) Processed spectrum with airPLS algorithm



(b) Processed spectrum with arPLS algorithm



(c) Processed spectrum with Adaptive Least Squares Thresholding algorithm



(d) Processed spectrum with Polynomial Fitting algorithm

Figure 3.3. Examples of Baseline Removal algorithms (2of2)

is stretched over this curve and the baseline points that do not lie on the rubber band are discarded. Finally, by subtracting the corrected spectrum from the original spectrum, the baseline is obtained. The variable parameter is n.

AsLS (Asymmetric Least Squares) [50] - An algorithm that combines a smoother with asymmetric weighting of deviations from the (smooth) trend to get an effective baseline estimator. The two variable parameters are p for the asymmetry and *lambda* for the smoothness.

airPLS (Adaptive Iteratively Reweighted Penalized Least Squares) [51] - A method that works by iteratively changing weights of sum squares errors (SSE) between the fitted baseline and original signals, and the weights of the SSE are obtained adaptively using the difference between the previously fitted baseline and the original signal. The variable parameter is *lambda* for the smoothness.

arPLS (Asymmetrically Reweighted Penalized Least Squares) [52] - A weighting scheme based on the generalized logistic function. The method estimates the noise level iteratively and adjusts the weights correspondingly. The variable parameter is *lambda* for the smoothness.

Minima Identification and Discrimination via Adaptive Least Squares Thresholding [53] - A technique that combines spectral segmenting, minima searching, differencing and statistic-based band discrimination. The variable parameter is n as the number of ranges in which the spectrum is divided.

Polynomial Curve Fitting [54][55] - A function that subtracts the background of a spectrum in an iterative way. It fits a polynomial through the data-points and those points, either of the fitted polynomial or those of the spectrum below the fitted spectrum, are used for the following iteration. No variable parameters are used for this algorithm, thus it generates one augmented spectrum from every input spectrum.

3.3 Pre-Processing: Noise Filtering

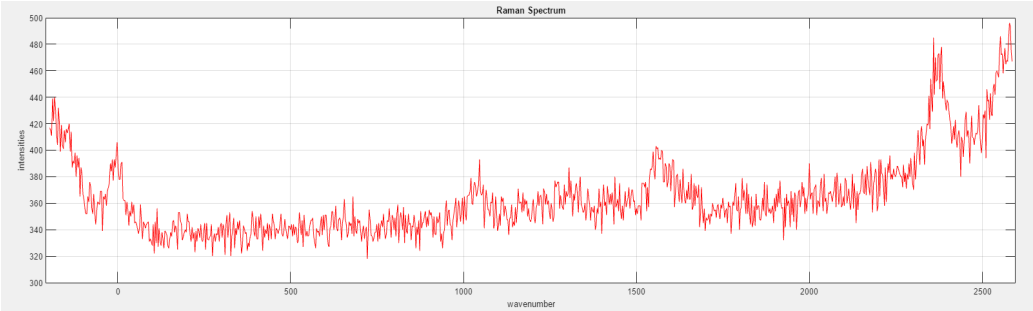
Another common pre-processing step in a Raman spectra classification task is the reduction or filtering of the noise. The whole acquired signal could be decomposed as the sum of its stationary and variable components:

$$S = E(S) + N$$

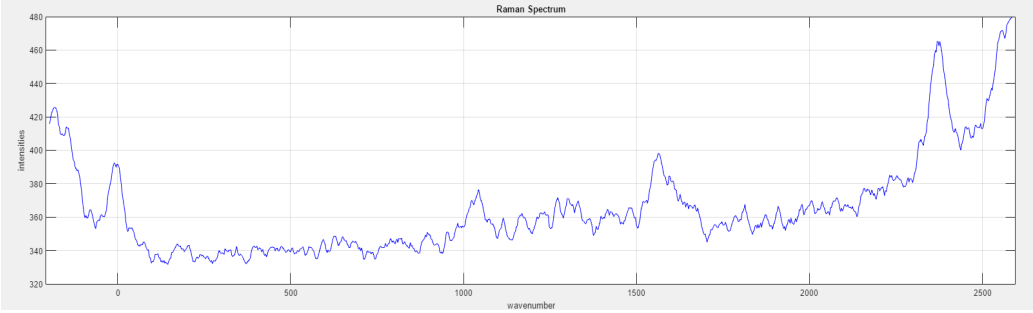
where S is a generic signal, $E(S)$ its expected value and $N = (S - E(S))$ a zero-mean aleatory term which we common refer as "noise". In Raman experimental measures, noise represents a serious limitation for spectral identification of the target, since the noise fluctuations can reach the level of the signal and thus hide or overlap the characteristic peaks. Noise sources are mainly related to the detection phase, and their causes can be many, so that different components of noise can be identified and possibly reduced. This aspect of spectra acquisition and processing is dealt in detail in section 4.2.

In the following paragraphs a list of de-noising filters is presented. All the methods are used in the present research, and as the baseline removal techniques, the parameters that can be tuned for each one of them are highlighted. In fig.3.4 and fig.3.5 an example of processed spectra for each noise filtering algorithm is depicted. They have been chosen among the most common methods used in signal analysis nowadays. Since there is no available ground truth for the use case that will be subsequently introduced, it is not possible to assess the effectiveness of a filtering procedure until the final evaluation of the classifier. Notwithstanding this issue, even a filtering algorithm that does not work properly with a certain class of spectra, can still be useful for the input set generation. This is the main reason that led to the widest possible differentiation in the choice of the following de-noising filters.

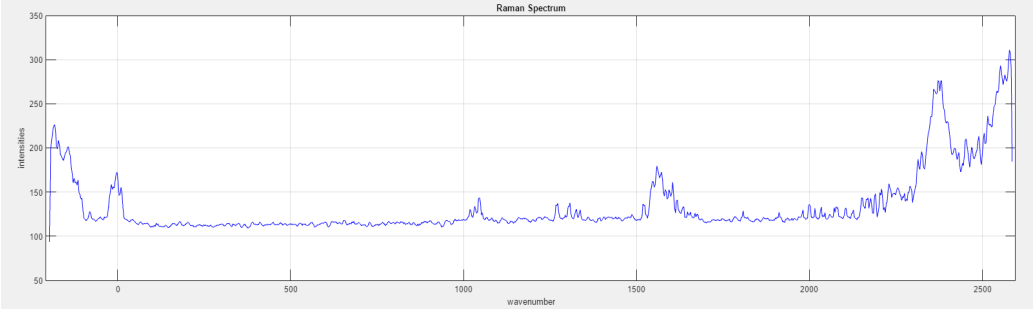
Savitzky-Golay filter [56] - A digital filter whose purpose of smoothing the data is achieved with a convolution, i.e. by fitting successive sub-sets of adjacent data points with a low-degree polynomial by the method of linear least squares. The variable parameters are the *order* of the polynomial and the *length* of the weighting vector.



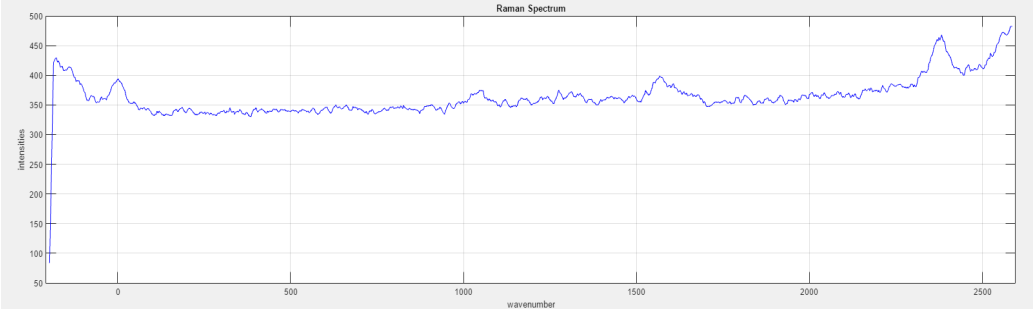
(a) Raw Spectrum



(b) Processed spectrum with Savitzky-Golay filter

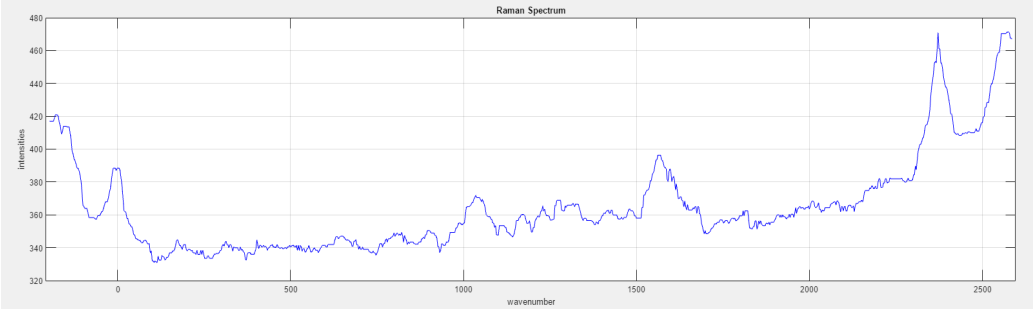


(c) Processed spectrum with Wiener filter

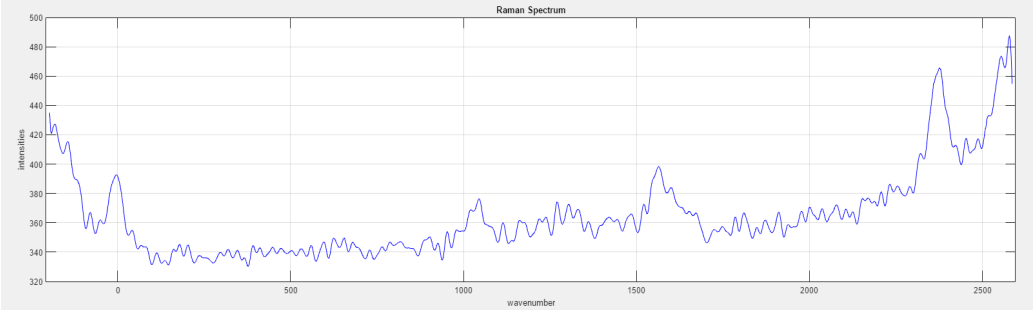


(d) Processed spectrum with a Moving Average filter

Figure 3.4. Examples of Noise filtering algorithms (1of2)



(a) Processed spectrum with a Percentile filter



(b) Processed spectrum with a Discrete Wavelet Transform

Figure 3.5. Examples of Noise filtering algorithms (2of2)

Wiener filter [57] - A filter that uses a statistical approach given by the minimum mean square error (MMSE) estimator method to estimate an unknown output signal from a known related input signal. The variable parameter is the *size* of the window.

Moving average filter [58] - A calculation method to analyze data points by creating a series of averages of different subsets of the full data set. The variable parameter is the *size* of the window.

Percentile filter [59] - A rank-order filter that calculates a p percentile on a moving fixed-size N window that spans each data point. The variable parameters are p and N.

Undecimated Wavelet transform [60] [61] - A stationary wavelet transform that performs a wavelet decomposition and re-scaling. Mathematically, the wavelet correlates with the signal if the input signal contains information of similar frequency. The variable parameter is the decomposition *level*, i.e. the maximum number of

levels in which the signal can be decomposed.

3.4 Artificial Neural Networks

In this section we start to analyse the alternative approach to pre-processing algorithms: Machine Learning tools, and in particular Artificial Neural Networks (ANNs). Machine Learning (ML) is a field that belongs to Artificial Intelligence science, and can be defined as the study of computer algorithms that improve automatically through experience [62]. ML tools accomplish their specific tasks building a model on their sample data (i.e. training data) in order to be able to make predictions/decisions without being explicitly programmed.

An artificial neural network (ANN) is a computing system that was originally designed to work the way the human brain works. It is made up of a number of highly interconnected elements which process information by their dynamic response to external inputs [63]. An ANN learn by examples, so every ANN is configured for a specific application, such as pattern recognition, data classification, and so on. The advantages of using ANNs include: adaptive learning, self-organization, real-time operation, fault tolerance, and many others. The ANNs are composed by multiple nodes called neurons. The nodes are connected by links and they interact with each other. The nodes can take input data and perform simple operations on the data. The result of these operations is passed to other neurons. In fig.3.6 an artificial neuron is represented, i.e. the basic processing unit of an ANN, whose aim is to combine the input signals and to produce an output value.

An artificial neuron (a.k.a. perceptron) implements a mathematical function that operates on the input signals and generates outputs. The learning objective of a neuron is to determine the ideal weights for each input signal. The learning algorithm arbitrarily assigns weights to each input signal, then the signal value is multiplied by its corresponding weight. The product (weight times signal value) of each signal is added to compute an output, plus a bias factor, as in the following equation:

$$z = f(x) = w_1x_1 * w_2x_2 * ...w_n * x_n + b_0$$

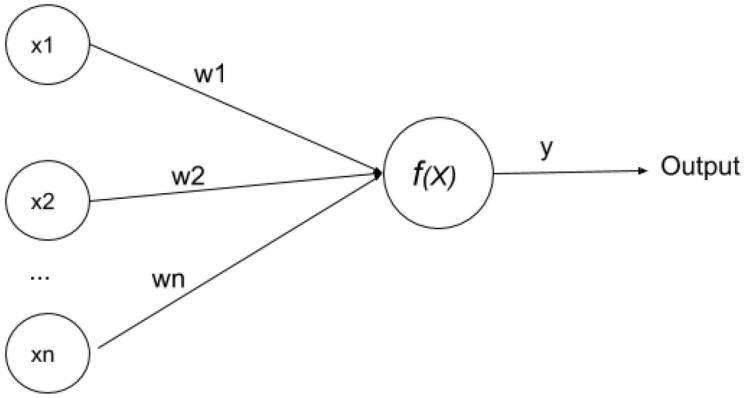


Figure 3.6. Basic representation of an artificial Neuron [5]

The mathematical function that determines the output of a neuron is called the activation function. Still, the output of this function is not the value z, but rather the value determined by the activation function that operates on the value z. In fact, since the value of z can be anything ranging from $-\infty$ to ∞, the activation function determines whether the neuron should be activated or not (turned on or off), normalizing the output of each neuron in a fixed range. The choice of a certain activation function is often driven by the particular prediction problem that the network is expected to resolve (e.g. nowadays ReLU functions are the most common ones used in multi-layer and convolutional networks because they do not constraint the weighted sum in a finite interval, while Sigmoid and TanH are still often used in certain neural networks such Recurrent Neural Networks). In fig.3.7 some of the most popular ANN activation functions are represented.

An ANN contains several neurons. Inputs are processed by a group of neurons (input layer). Each neuron in the group processes the inputs independently. Outputs from this group of neurons are fed to another group of neurons (hidden layers) for further processing. It can be imagined as successive neuron layers where the output from one layer is fed as inputs to the next layer, until reaching the last one (output layer). An ANN can have as many layers as needed. A representation of a typical ANN structure is depicted in fig.3.8.

In order to train a neural network to perform some tasks, we must adjust the weights of each unit in such a way that the error (or loss) between the desired output and the actual output is reduced. This is performed through an optimization

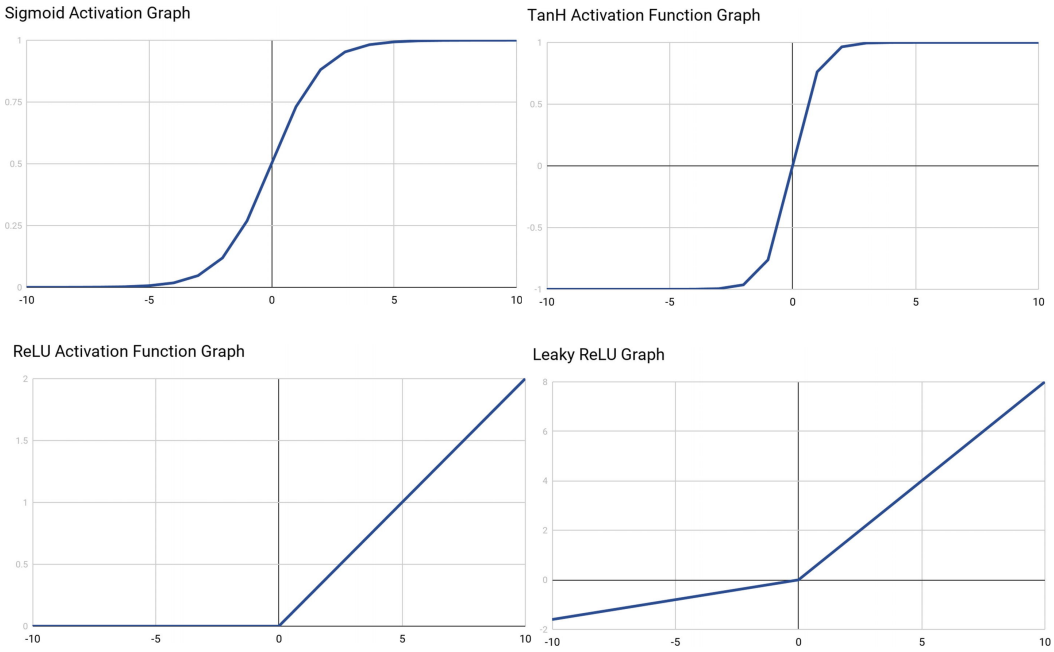


Figure 3.7. Some examples of ANN activation functions [5]

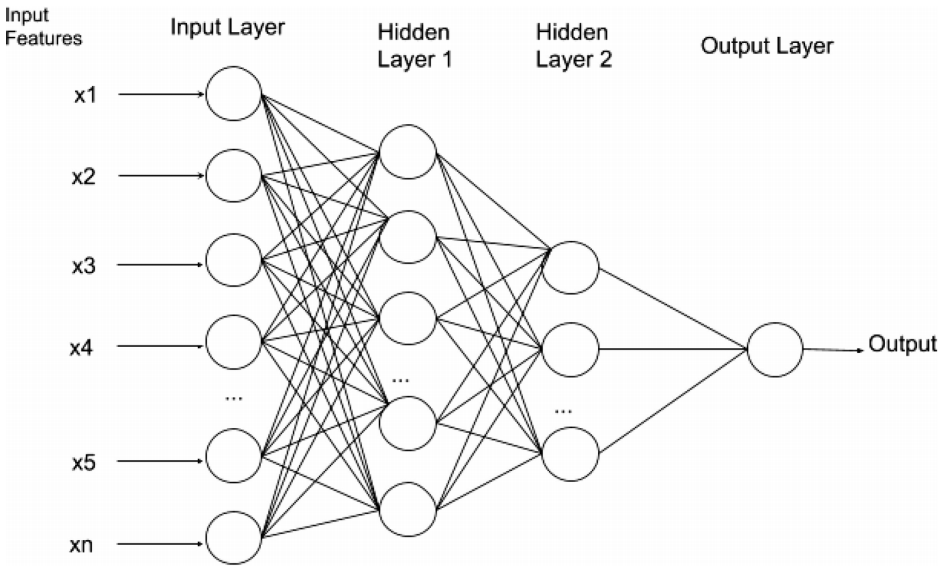


Figure 3.8. A multi-layered Artificial Neural Network [5]

algorithm, which is a mathematical function that optimizes the loss function. In neural network tasks the most suitable one is the Gradient Descent, whose aim is to find weights values such that the loss function results in a local minimum. From

calculus, we know that the first derivative of a function at a point gives the slope or gradient of the function at that point. Hence, in Gradient Descent, the derivative is calculated to get the gradient in order to determine in which direction along the curve to move. Since the goal is to minimize the cost (loss function), the algorithm moves to the direction of the negative gradient. In neural networks this process is also referred as error derivative of the weights (EW).

The back-propagation algorithm is the most widely used method for determining the EW. Taking a feed-forward network (an ANN where the connections between cycles do not form a cycle), the output of the neuron is initially calculated in the forward direction, starting from the first hidden layer to the final output layer. Then, in order to update the weights to minimize the error, the gradients of weights are first calculated at the last layer, while the subsequent ones are determined in the backward direction. The computation of the gradient (in this case a partial derivative with respect to the weights) from one layer are reused in the computation of the gradient for the previous layer. This backward flow of the error information allows for an efficient computation of the gradient at each layer, since the calculations are not done independently at each layer. Hence, the final objective of back-propagation is to train a multi-layered neural network such that it can learn the appropriate internal representation (i.e. the most optimized weights for each input feature for each connection to neurons of each layer) to allow it to learn any arbitrary mapping of input to output[64].

As in other ML methods, the supervised learning of an ANN consists in two distinct phases: a training phase, where a labeled set of input data is fed to the network, so that it can be adjust its weights in order to model the input, and a test phase where an unlabeled independent input set is fed to the network to calculate its classification confidence over new data.

3.4.1 Evaluation of ANNs on Spectral Classification Tasks

Once that we decided to experiment with ANNs, the next step is to identify the most important spectrum feature that could affect classification accuracy. We built an experimental spectroscopic apparatus in laboratory and experimented with

neural networks, varying many parameters of the input data, that are both artificial and real. Initially, our aim is to use an ANN in order to classify a set of random noisy spectra respect to a target vector. Half of these spectra set are Gaussian signals with noise, the other half are pure noisy signals. The ANN that we want to create should classify these spectra as a Gaussian or not. The Gaussian function has been chosen for its comparability to Raman spectra peaks. It has the following form, with a and b real constants and non-zero c:

$$f(x) = a \cdot \exp\left(-\frac{(x-b)^2}{2c^2}\right)$$

In fig.3.9 it is depicted an example of Gaussian function on the left, and a heat-map of our input set that is composed of a half part of different Gaussian functions and a other half part of a pure noisy signals. All the ANN processing have been performed with MATLAB©.

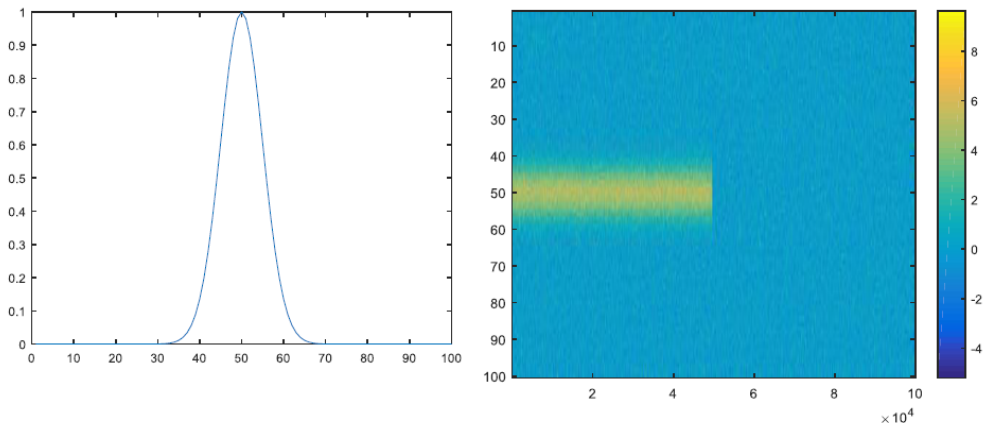


Figure 3.9. A Gaussian function (left) and an example of input set represented as a heat-map (right) with vertically stacked spectra

We used a very simple ANN for our tests, with an input vector size of 100, a hidden layer of 10 neurons, and an output layer of 1 boolean node that predicts a 0 value for an empty (i.e. pure noise) signal and a 1 value for a Gaussian signal. The number of input vectors are generally set on 100000 (10^5), unless otherwise noted. We use Cross-Entropy loss as a loss function to measure the performance of the classification model, while the input set is divided in 70% training set, 15% validation set (used to measure the ANN generalization) and 15% test set. Training set is used

to train the neural network in order to fit the classification model, and as in the most machine learning methodologies includes the largest amount of input samples. Validation set is used during the training phase to provide an unbiased evaluation of the training model, in order to prevent unwanted features like overfitting (discussed in detail in section 4.4). Test set provides the final unbiased evaluation over the model.

Regarding the evaluation of the results, we must first introduce some new concepts. In statistic test theory, false positives and false negatives are the two types of error distinguished in the result of a test. They are more correctly defined as Type I Error (FP) and Type II Error (FN). A type I error occurs when the null hypothesis (H0) is true, but is rejected. It is asserting something that is absent, a false hit. A type I error may be likened to a so-called false positive (a result that indicates that a given condition is present when it is actually not present). A type II error occurs when the null hypothesis is false, but erroneously fails to be rejected. It is failing to assert what is present, a miss. A type II error may be compared with a so-called false negative (where an actual "hit" was disregarded by the test and seen as a "miss").

In order to measure the confidence of the Neural Network predictions, a classification confusion matrix has been used. On this plot, the rows correspond to the predicted class (Output Class) while the columns show the true class (Target Class). The diagonal cells show for how many (and what percentage) of the examples the trained network correctly estimates the classes of observations. The off diagonal cells show where the classifier has made mistakes. In fig.3.10 it is represented a classification matrix that highlights the related types of error. Generally, in the next paragraphs, when we assert that the ANN has an accuracy of a certain percentage (e.g. 98%), we are indicating the overall accuracy, i.e. the sum of the True Positives and True Negatives percentages. In the next paragraphs we present a brief description of the performed ANN tests. Unless otherwise noted, the classification results in the plot and diagrams represent the accuracy on the overall dataset, and not only on the test set.

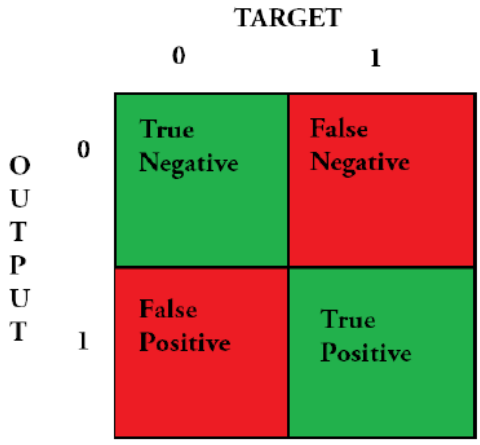


Figure 3.10. Classification Confusion Matrix

Accuracy vs Input Set Dimension In the first test, the performance of the ANN are measured with different dimensions of the Input Set (N) (from 10^2 to 10^6). These tests have been particularly useful to get an idea of the processing time and the relative resources needed. The results are depicted in fig. 3.11.

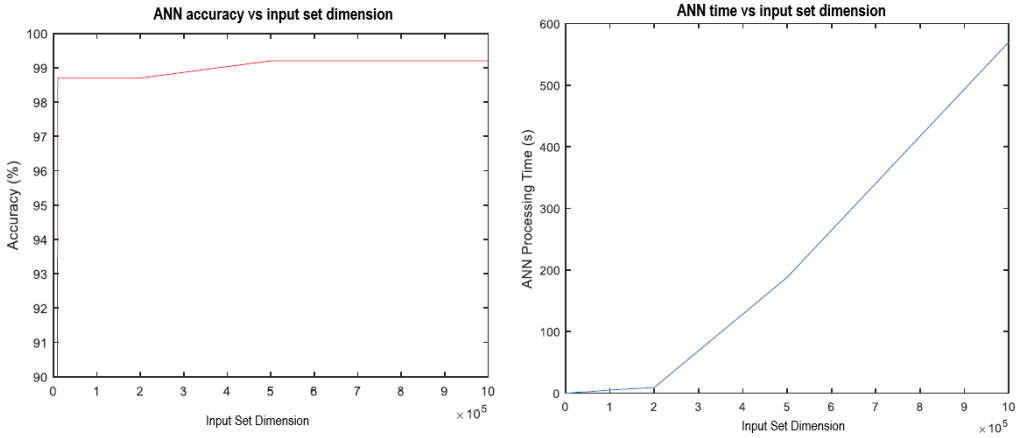


Figure 3.11. ANN Accuracy vs Input Set Dimension (left), ANN processing time vs Input Set Dimension (right)

Accuracy vs Signal-to-Noise Ratio The Signal-to-Noise Ratio (SNR or S/N) is a parameter that compares the level of a desired signal to the level of background noise in any system that acquires, processes or transmits information. SNR is defined as the ratio of signal power to the noise power, and expresses how powerful is the

signal compared to the noise; these two measures are often expressed in decibel (db) or Watt (W). In fig.3.12 there is an example plot of high SNR (powerful signals) spectra, and of low SNR (noisy signals) spectra.

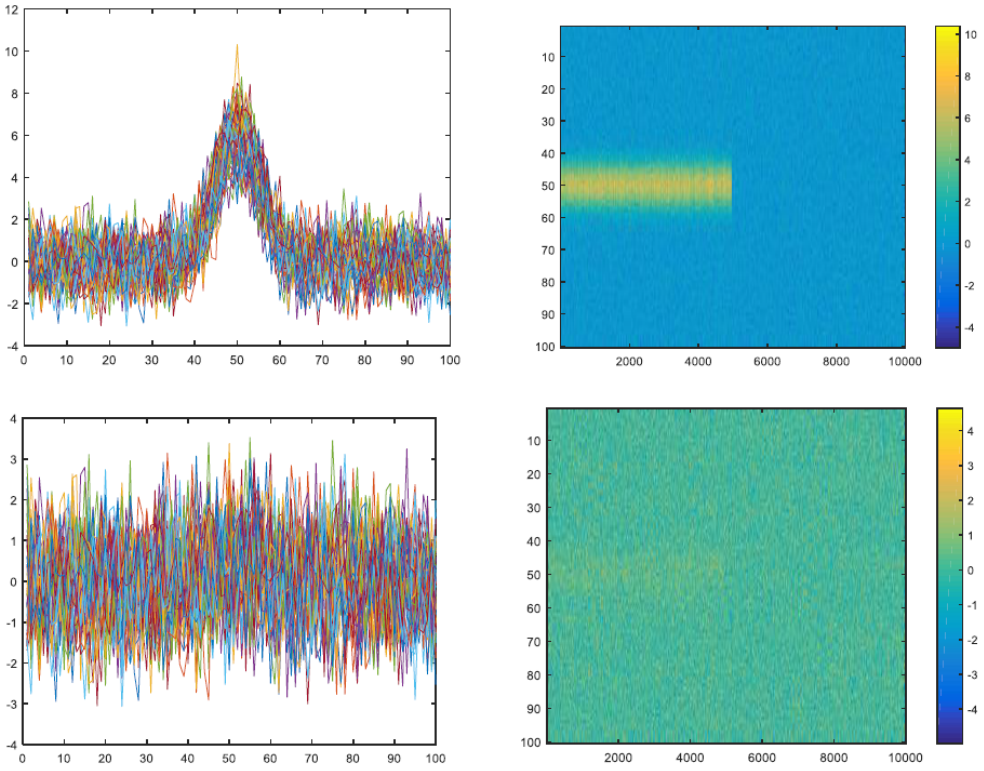


Figure 3.12. High SNR spectra (above) vs Low SNR spectra (below)

We make several ANN computational tests, decreasing the input set SNR, and the results are depicted in fig.3.13.

Accuracy vs ANN Hidden Layer size In the third test the accuracy of the ANN is measured with different network architectures. The performance of the ANN is measured both decreasing the initial size of the hidden layer (the number of neurons) and increasing it up to the value of 20. The results are depicted in fig.3.14, where it results clear that the accuracy value of 92.2% remains fixed in all the layer configurations. In addition, the tests have been performed also adding another layer, with no substantial changes in the final accuracy. Though the number of 2 neurons in the hidden layer turns out to be sufficient to obtain the best classification result

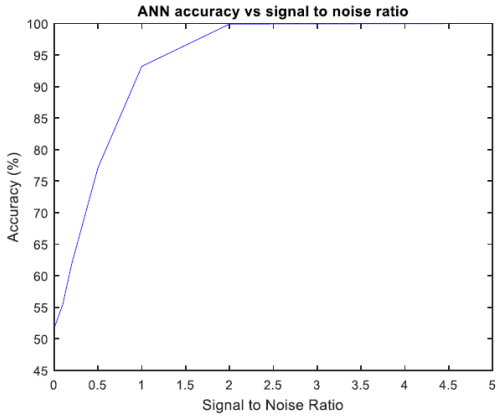


Figure 3.13. ANN accuracy vs Signal-to-Noise Ratio

in this initial simple problem, we chose to maintain the number of 10 neurons in the hidden layer of this standard experimental ANN for the next refined experiments.

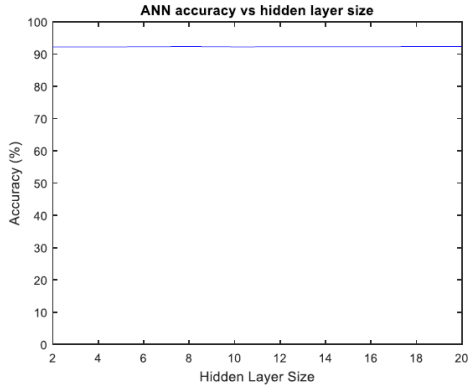


Figure 3.14. ANN accuracy vs Hidden Layer Size

Accuracy vs Peak Position In this test the ANN accuracy is initially run with the standard parameters described above. The difference is the position of the Gaussian peak of the vector input set: every vector is set with a random peak position, so that we can investigate the behavior of the ANN with a more realistic set. In other words, every sample has a single Gaussian peak as before, but shifted in a different position. The task is slightly different: the ANN must not only recognize a Gaussian signal from a pure noisy signal, but it must discriminate only a subset of Gaussian signals whose peaks fall into a certain strict interval (which has been

labeled with a True value). With a high value of SNR ($\simeq 5$) the ANN shows an accuracy of 96.4%, while with a low value of SNR ($\simeq 1$) it shows an accuracy of 81.8%.

In order to understand the inner logic of the network, the False Positives (i.e. the noisy signals incorrectly classified as Gaussian signals) and False Negatives (i.e. the Gaussian signals incorrectly classified as noisy signals) sets will be analyzed. This is necessary to understand if the network's false predictions fall around the shifting limit value of 5 (and hence the error is acceptable) or if the false predictions are calculated with a wrong logic that won't help the classification in our future difficult tasks. The results are depicted in fig.3.15. The histogram plots demonstrate that

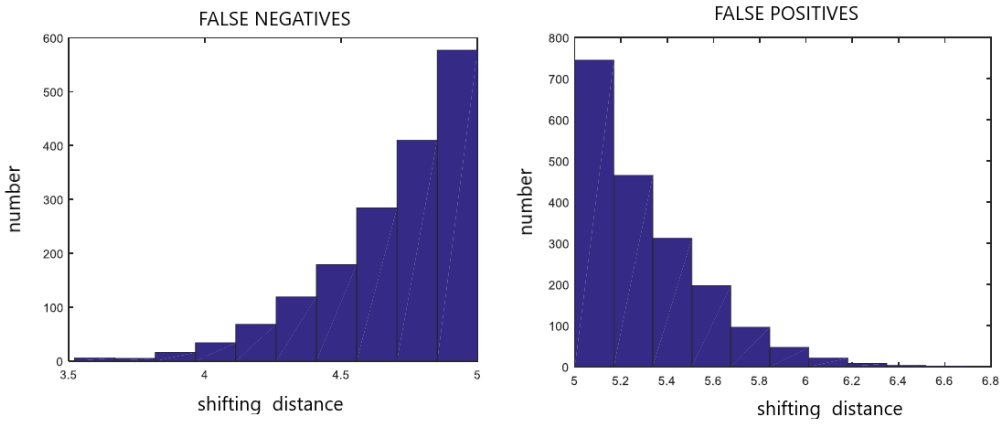


Figure 3.15. ANN False Negatives and False Positives Histogram related to the shifting distance

the ANN is working properly: most of the wrong predictions are located in a small interval around the initial value of 5. The more the Gaussian signal moves far from the acceptable shifting value, the smaller is the number of wrong predictions by the network.

Accuracy vs Different Peaks In this task the vectors that belong to the input size are different from the previous tests. A second Gaussian signal (randomly centered) is added to the standard Gaussian signal at the center of the vector that has been used since the first tests. A certain rate of probability is set at the beginning of the script to determine if the vector contains both the two Gaussian functions,

only one of the two signals, or none of them. The aim of the ANN is to choose as good output only the vectors that contain the central Gaussian, discarding the cases where the second Gaussian is present. It is especially interesting to analyze the cases where the center of the second random Gaussian signal falls around the center of the vector, so that it can be confused with the standard "good" Gaussian.
With a SNR$\simeq$ 5 the ANN shows 98.6% accuracy, while decreasing until SNR$\simeq$ 1 we obtain 78.3%, though quite good. A preliminary conclusion is that the ANN, as it is configured, is very suitable for this kind of task to resolve. Analysing in detail the False Positives of this test (fig. 3.16) , it seems that a standard Gaussian is present at the center of the vector in many false positives vectors (centered yellow stripes). We know instead that the vectors in this set are the ones without the center Gaussian that we want to identify. This can be explained with the fact that the random generated noise has created a peak in our central interval of interest; so, the ANN has wrongly identified the vector as a good target. Despite the mistake, the analysis demonstrates the correct logic that rules the ANN processing.

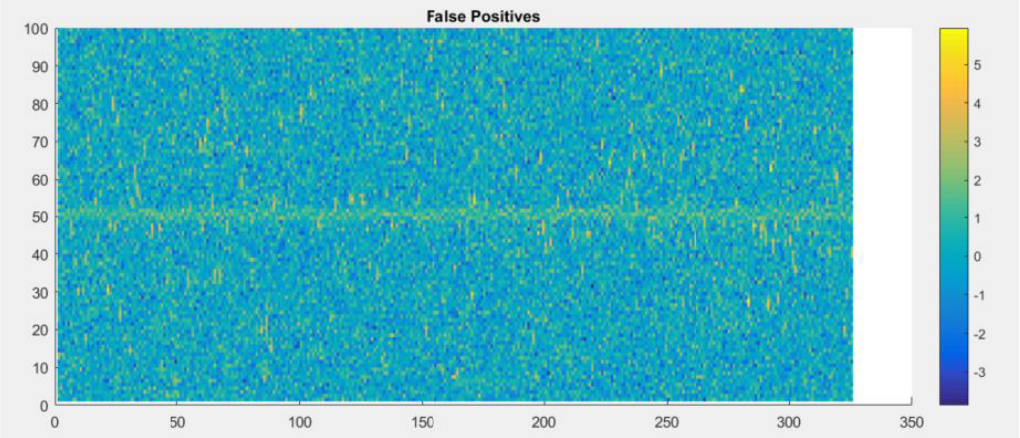


Figure 3.16. ANN False Positives heat-map

Laboratory test with real spectra After we've artificially created the spectra for the input set in the previous examples, we tested the behavior of the ANN with real spectra acquired in laboratory conditions. The aim of the test is to acquire and identify a set of mercury (Hg) spectra emitted by a mercury-vapor lamp, a gas discharge lamp that uses an electric arc through vaporized mercury to produce

light. The main peak that characterizes these spectra is located at the wavelength of 350nm. The other devices used in this test are a spectrometer HR250 and a camera Basler F312A. The spectrometer main parameters are invariable throughout the tests (Gain=700, Integration time=0.8ms). The laboratory setup is depicted in fig.3.17. In order to simulate a variable SNR in the input sets, the spectrometer's

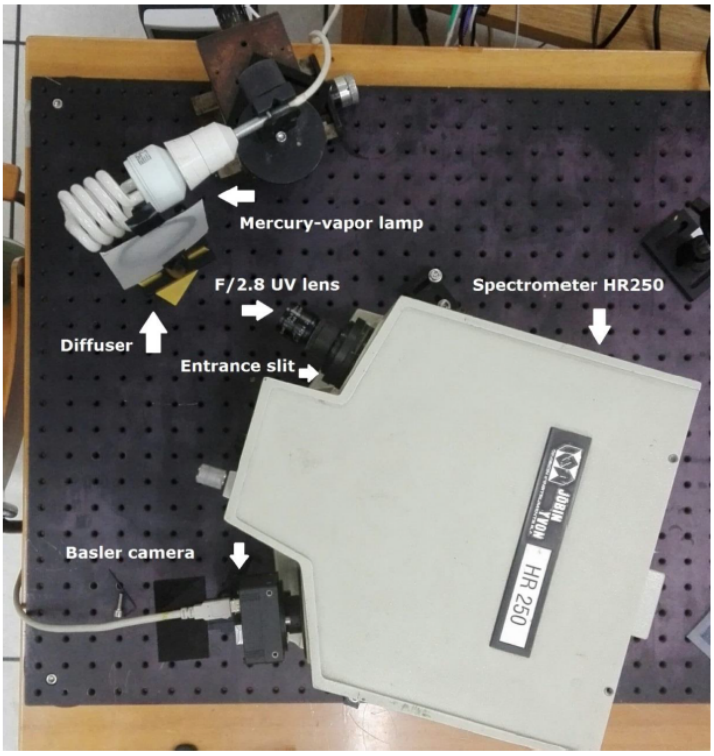


Figure 3.17. Optical acquisition system setup

diaphragm is adjusted at different f-number, increasing or decreasing the amount of light that passes through the lens. The f-number of an optical system is the ratio of the system's focal length to the diameter of the aperture. We will have a greater amount of light with small values of f-number (i.e. f/2.8) and therefore a cleaner signal will be acquired. Otherwise we will have a smaller amount of light with high values of f-number (i.e. f/16) that will result in a noisy signal with poor SNR. An example of acquired spectrum for each level of SNR is depicted in fig.3.18. After the training phase, the ANN is launched with each one of the test set with different f-number. The results are shown in the classification confusion matrix at Tab.3.1.

As we can see in the table 3.1, the ANN accuracy is perfect (100%) for spectral

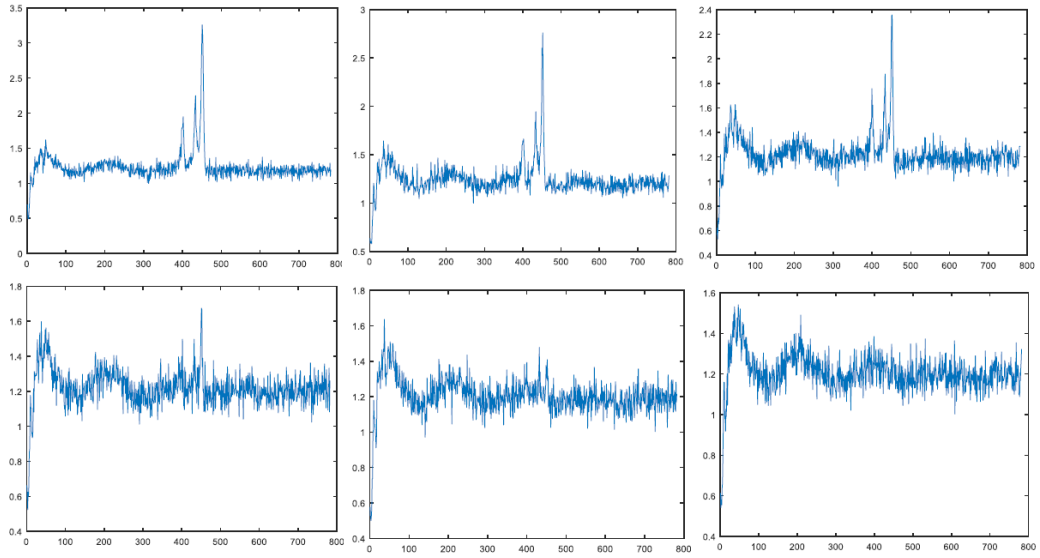


Figure 3.18. Examples of Hg spectra acquired from the highest level of SNR (upper-left) to the lowest level of SNR (bottom-right)

	Training		SNR=27.2		SNR=24.4		SNR=16.4		SNR=8.5		SNR=4.6		SNR=1.9	
	0	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
0	19%	1.1%	0	0	0	0	0	0	0	1%	0	8%	94%	0
1	1%	78.9%	0	100%	0	100%	0	100%	0	99%	0	92%	6%	0

Table 3.1. Classification confusion matrix of Hg spectra test

classification with $SNR > 8.5$. With $SNR = 8.5$ the network registers only a false-negative (missed hit) over 100 spectra, while at $SNR = 4.6$ the false negatives amount raises at 8%, with an overall accuracy of 92%. In these 2 subsets there are no false positives because the target values are all set to 1 (Hg spectra). With $SNR = 1.9$, hence with all targets set to 0 (noisy spectra), the network registers 6 false positives over 100 spectra, with an overall accuracy of 94%.

Conclusions After these series of tests, some conclusions about the ANN performances evaluation in spectral classification tasks can be listed. First, for the simplest classification tasks, the input size does not appear to affect the ANN overall accuracy. A similar conclusion can be asserted with the results of the third test where we have

analyzed the impact of the hidden layer size (the number of neurons) in the network accuracy. Apparently this parameter does not seem to affect the accuracy at all, and this can be explained with the simple nature of the tasks. The main parameter that hardly affects the network performances for these spectral classification tasks is surely the Signal-to-Noise Ratio, as it can be observed in the second test and in the following ones. Regarding to the two Gaussian tests, we already demonstrated that most part of the false positives are noise-induced peaks that fall into the interval of interest. The ANN performance, conversely, does not seem to be affected from the presence of the second interferent Gaussian signal. This is a good promising feature for the possibility of a Raman spectral classification based on ANN: the contemporary presence of several peaks should not affect the accuracy of the ANN in the identification of a certain peak of interest.
So, the only critical issues emerged in these tasks are the known problems of the noise that can alter the accuracy and the performance of the ANN, as well as the necessity to have a large number of vectors (of the order of thousands for these tests) to properly train the network. These issues can be challenging in a real operational scenario, and they will be studied in details in next sections.

3.5 Most promising ML technique: CNN

Deep Learning (DL) is a term coined in the last decade to denote a class of Machine Learning techniques, such Artificial Neural Networks, that exploit multiple layers to progressively extract higher-level features from the raw input[65]. Deep Learning is mainly used in Computer Vision (CV) tasks, an area of research that it's been advancing rapidly in the last years, which include a lot of application in many different fields (e.g. Face Recognition, Self-Driving Cars, and so on). Recently, the traditional (shallow) ANNs showed some limitations in handling Computer Vision problems where the inputs (like images or videos) can get really big. As a demonstration example, even a small 64x64 RGB image (64x64x3=12288 pixel values) requires a big input layer to be processed by an ANN. A solution to the complexity of the CV tasks and its memory requirements has been proposed with the introduction of a new type of neural networks that exploit the mathematical

convolutional operation: Convolutional Neural Networks (CNN)[66].

A conventional ANN consists of an input layer, one or more hidden layers, and an output layer. CNN has a set of additional layers, called convolution layers. The input is fed to the first one of this convolution layer. The output from the first convolution layer is usually fed to other convolution layers whose aim is mainly to perform a dimensionality reduction. Then, the output of the last convolution layer is fed to an "input" layer of a fully connected ANN. Each convolution layer implements an algorithm that performs feature selection and sub-sampling of the received input. An example of CNN designed for classification tasks is depicted in fig.3.19

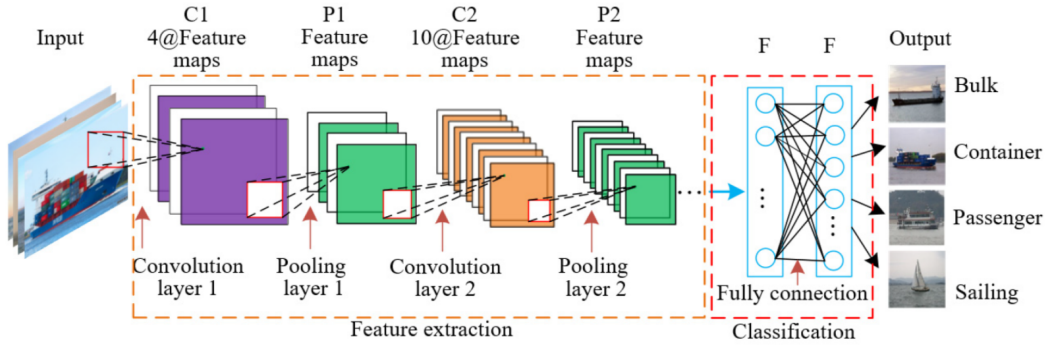


Figure 3.19. Example of Convolutional Neural Network [6]

The encouraging classification results achieved by CNNs have opened the way to many experimental studies about spectral classification. In fact, we can divide chronologically the automatic spectral classification techniques in pre and post deep learning era. Regarding Raman spectra classification, among the pre-DL era several statistical approaches like Multivariate Analysis (MVA) and Principal Component Analysis (PCA) can be included [67] [68]. For the post-DL era, CNN remains the most widely used method in Raman spectra classification [47] [69] [70], though other attempts have been tested, like combining Machine Learning and statistical based techniques [71] or exploring other Machine Learning approaches [72]. A recent one-shot learning approach tries to overcome the problem of a number of limited training spectra with the use of a Siamese Neural Network to reformulate the problem of spectra classification from a multi-class classification to a binary spectral matching [73]. Nevertheless, none of these proposed methods have resulted

to be as an ultimate solution for automatic real-time Raman spectra recognition, in particular for specific problems (e.g. historical pigments identification, the main case of study of the present research) where large reference datasets are not available or spectra show a consistent amount of noise.

Chapter 4

The Method

In this chapter we propose an innovative method to overcome the issues that can affect a Raman measurement in a non-laboratory scenario in order to maximize the spectral classification accuracy. This method regards both the spectra acquisition phase, where the main objective is to obtain the highest Signal-To-Noise ratio, and the evaluation phase, where a complex algorithm that merges the pre-processing and the Deep Learning approach is presented.

4.1 Charge Coupled Device (CCD)

As described in section 2.3, CCDs are the most common detector devices used in a Raman apparatus to measure the light intensity, allowing an image data acquisition and a subsequent analysis (Fig.4.1). The CCD sensors refer to a semiconductor architecture in which the charge is transferred via a storage area [74]. They are basically designed as an array of linked or coupled capacitors. Each capacitor (a.k.a. gate) can convert the amount of incident photons into a proportional amount of electric charge. Once the charge packets are collected, they are sequentially transferred from gate to gate until they are measured at the detection node. The operating cycle of a CCD can be summarized in four steps:

1. charge creation;

2. charge collection;

3. charge transfer;

4. conversion of the charge into measurable voltage.

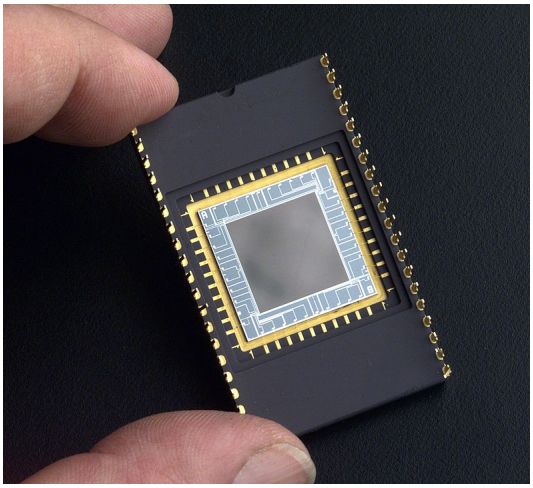


Figure 4.1. A Charge Coupled Device designed for UV imaging (from https://en.wikipedia.org/wiki/Charge-coupled_device)

Due to their design, CCD sensors are represented as a matrix of pixels. A pixel can be constituted by several gates. After exposure and charge creation by photoelectric effect for each pixel, the charge is stored generally in columns by channels. At the end of each column, there is an horizontal pixel register (shift register) that collects one line at a time and then transfers the charge packets to an output amplifier (fig.4.2). Then, the signal can be digitally encoded by an electronic processor independent from the CCD.

Among the main CCD parameters like number of pixels, spatial resolution, temporal resolution, transfer rate, and so on, one parameter of great interest in this research is the Quantum Efficiency (Q_e), which represents the device probability of detecting a photon impinging its surface, and hence the ability to correctly measure the proportionality between incident photons and electric charge. Another parameter of interest is the Binning Factor (M), as with the binning process we indicate a pixel gathering technique where the charges of adjacent pixels are grouped and summed. In our case of study, Raman spectra, which are represented by 1-Dimensional arrays, are obtained from the 2-Dimensional CCD grid after a vertical binning process.

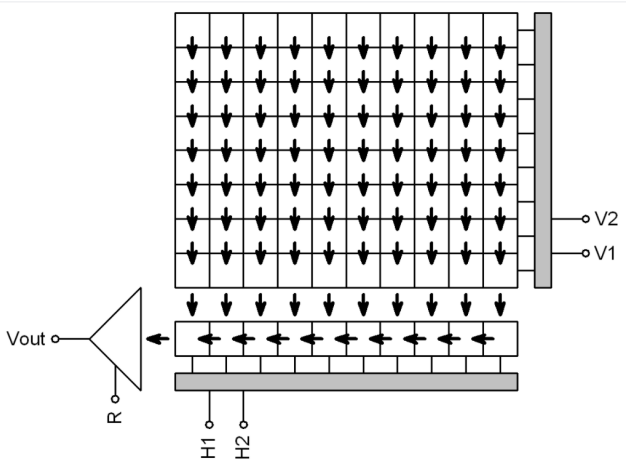


Figure 4.2. A basic scheme of CCD operating mechanism (from https://www.gxccd.com/)

In the present research, besides the aforementioned canonical CCD devices, two further types of CCD have been tested: EMCCD and ICCD. An Electron-Multiplying CCD (EMCCD) is a charge-coupled device in which a gain register is placed between the shift register and the output amplifier. In the gain register the electrons are multiplied by impact ionization in a similar way to an avalanche diode. An Intensified-CCD (ICCD) uses instead an image intensifier that allows an electron multiplying at the gate level. Image intensifier are basically composed of three elements (a photocathode, a microchannel plate (MCP), and a phosphor screen) which are super-imposed on the CCD capacitors grid.

4.2 Sources of Noise in a CCD

An important issue in CCD sensors concerns the processes leading to noise because, even without incident photons, these devices can produce an unwanted dark signal which may hide the signal to be detected. As we stated in the previous chapters, noise represents a serious limitation for spectral identification, since it causes a fluctuation that can reach the level of the signal. Many stratagems have been developed to limit the impact of noise, such as signal binning, slow reading speed or cooling down the CCD. However, sometimes laser applications may require specific safety standards, limiting the target exposure and, as a direct consequence, the signal quality. In these cases it is of primary importance to reduce the noise.

Due to the equal probability to generate a count for each photon hitting a CCD element (given by its quantum efficiency Q_e), the spectroscopic signal can be considered as perfectly additive. The Planck hypothesis allows the determination of the number of photons composing a pulse of a given energy E_0: $N = E_0 \ / \ (h\ \nu)$. This means that a given amount of light energy can be divided in a single shot E_0 or n shots E_0/n. Hence, the total acquired signal S can be expressed as the sum of its components.

Similarly, if we define the noise as the standard deviation of the intensity in a region where no spectral lines are present, we can express the total amount of noise as the sum of its components. In a spectroscopic measurement the total signal is not only the result of the incident light measured by the CCD, but it contains other components caused by internal processes of the sensor. For instance, the energy can be also provided by thermal agitation, leading to a small current even in absence of photons (*Dark current* (DC)). DC is due to this thermal energy and it follows that by cooling down the semiconductor the DC should be strongly reduces [75][1]. A further source of uncertainty is related to the process of charge readout from the CCD. Hence, based on the previous description, the signal S can be expressed as follows:

$$S = S_0 + S_D + S_R \tag{4.1}$$

S_0 is the signal measured by the detector, S_D is the dark current and S_R is the readout.

Each one of the components of the total signal carries out its amount of noise, i.e. stochastic fluctuation. The *Shot Noise* (SN) is related to the discrete nature of the light: any light measurement obeys the Poisson statistics, leading to an error proportional to its mean value. Since SN depends on the measured signal level but it is independent by temperature, it represents the best achievable performance for a CCD[76], i.e. the minimum possible uncertainty.

On the other hand, Dark Current may vary among pixels because of constructive inhomogeneities, thus each pixel may lead to different DC at the same temperature

[1] As a first approximation $DC = DC_0\ e^{-\Delta E_g/2kT}$, DC_0 is the DC @ 300K, T is the temperature, k the Boltzmann constant and E_g the gap energy in eV.

and for the same exposure time. This phenomenon is called Dark Signal Non Uniformity (DSNU). Indeed, the most important issue related to the DC is represented by its fluctuations, because they lead to the so-called *Dark Noise* (DN).
Finally, the CCD readout process causes a source of the so-called *Readout Noise* (RN). At least five different effects contribute to this noise: trapping-state noise, reset noise, background noise, charge-transfer noise and output amplifier noise[76].

To summarize, in order to estimate noise components according to eq.(4.1), it is assumed that N_0^2, N_D^2, e N_R^2 contribute to the variance of the measured signal, as follows:

- N_0^2 **SN**, caused by photon counting in optical devices; it follows Poisson distribution ($\sigma_x^2 =< x >$), hence: $N_0^2 = M\ S_0$ with M as Binning Factor.
- N_D^2 **DN**, generated by the charges in the detector; it follows Poisson distribution and the variance is: $N_D^2 = M\ N_d\ \delta t$ with δt as the exposure time in s, N_D the dark current $[e\ s^{-1}\ px^{-1}]$[2].
- N_R^2 **RN**, it depends on the manufacturing technology and on the readout mode (it does not depend on the signal nor on the exposure time. It rather depends on the frame rate).

The total variance is given by the sum of these three terms:

$$N^2 = N_0^2 + N_D^2 + N_R^2 \tag{4.2}$$

The SN is dominant for high signal levels, while DN and RN become relevant for low and very low signal levels [76].

Since the whole signal can be decomposed as the sum of stationary and variable components, given S a generic signal, $E(S)$ its expected value and $N = (S - E(S))$ a zero-mean aleatory term, it follows that eq.(4.1) becomes:

$$S = S_0 + N_0 + E(S_D) + N_D + E(S_R) + N_R \tag{4.3}$$

[2] $N_D = 2.55\ 10^{15}\ N_{d0}\ T^{3/2}\ d^2\ \tau\ e^{-\frac{E_g}{2kT}}$ with N_{d0} as the dark current at 300K $[nA/cm^2]$, d as the pixel linear dimension $[cm]$, T the temperature $[K]$, E_g as the gap energy $[eV]$ and k the Boltzmann constant. $E_g(T) = 1.1557 - \frac{7.02110^{-4}T^2}{1108+T}$ [77].

In a spectroscopic measurement we have the possibility to implement the dark field technique (i.e. background subtraction) to eq.(4.3), in order to remove from eq.(4.1) the non-zero mean constant components ($E(S_D)$,$E_(S_R)$). This is particularly useful when dealing with non-homogeneous CCD behavior.

$$S - S_{dark} = S_0 + N_0 + N_D + N_R, \qquad S_{dark} = E(S_D) + E(S_R) \tag{4.4}$$

As it can be noted in the equation, by performing the dark field subtraction it is possible to eliminate the average component (bias or polarization), but the stochastic fluctuation cannot be eliminated. It is therefore evident that a naive removal of the bias with a simple dark field subtraction leaves a fluctuation whose effect increases the signal noise. In the next section it is proposed the best acquisition strategy whose aim is to minimize the amount of noise.

4.3 Acquisition strategy to optimize SNR

4.3.1 Laser Safety

In laser spectroscopy the signal is strongly related to the laser pulse characteristics, both for timing and for energy (i.e. signal level). Thus, a synergic choice of laser characteristics and acquisition parameters may increase the SNR. With Diode Pumped Solid State lasers (DPSS) the Pulse Repetition Rate may reach hundreds of MHz, allowing to send laser pulses as close as tens of ns. The strategy of distributing the required energy among different pulses is then a valid alternative than delivering the whole energy with a single pulse. This choice allows to fulfill the eye-safety/skin-safety requirements such as defined in IEC 60825-1 International Standard [78], and in 2006/25/EC (ED) directive [79].

The aforementioned two directives define the maximum permitted exposure (MPE) to artificial radiations according to exposure duration, wavelength, coherence of the radiation and beam divergence. As described in [7], in case of a single exposure (i.e. continuous illumination for a time duration T) the thresholds and the algorithms for calculating the eye and the skin exposure are described in the ED directive.

Fig.4.3 shows the single pulse MPE for eye and skin for exposure times ranging from 1 ns to 10 s, and wavelengths from 180 to 2700 nm.

Otherwise, if the exposure is obtained through multiple pulses, some additional conditions must be fulfilled. In this case, the MPE also depends on the number of laser pulses employed within the total time T, so that the possibility to deploy the whole MPE depends on the characteristics of the specific laser, especially wavelength, power and pulse duration [7].

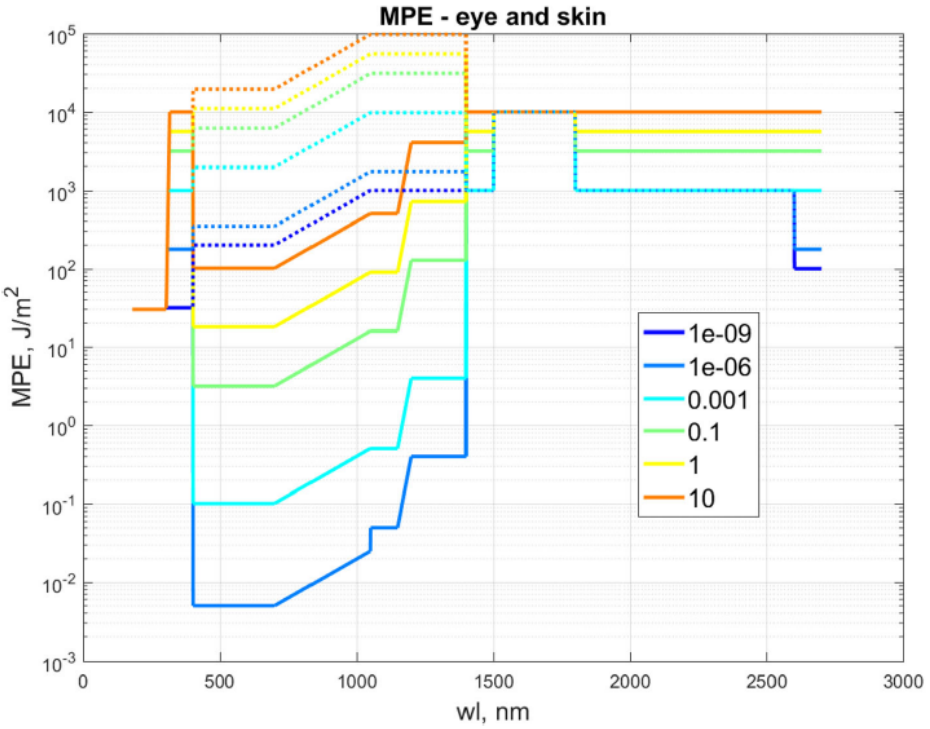


Figure 4.3. MPE vs wavelength for some exposure times for $\alpha < 1.5mrad$. Solid lines refer to eye MPE, while dotted lines refer to skin MPE. Time in legend is expressed in s [7]

.

4.3.2 Acquisition Strategies Comparison

Under the perspective of fulfilling laser safety directives, it is essential to identify the best conditions to perform a spectroscopic experiment, i.e. the conditions that maximize the SNR. Three different acquisition strategies can be defined, involving the same incident energy, as it is depicted in fig.4.4[8]. The relative parameters for each acquisition strategy are summarized in tab.4.1.

- **Case 0**: the reference. One pulse of Energy E_0 is acquired in a single frame of exposure interval δt.
- **Case 1**: k signals, each of Energy E_0 and multiple acquisitions, i.e. one reading for each pulse. It takes a short acquisition time, but involves more reading procedures.
- **Case 2**: one signal, of Energy kE_0 and single acquisition.
- **Case 3**: k signals, each of Energy E_0 and single acquisitions, i.e. one reading for all the pulses. The CCD exposure has duration $\Delta T \geq k\delta t$. It takes a longer acquisition time but just one readout.

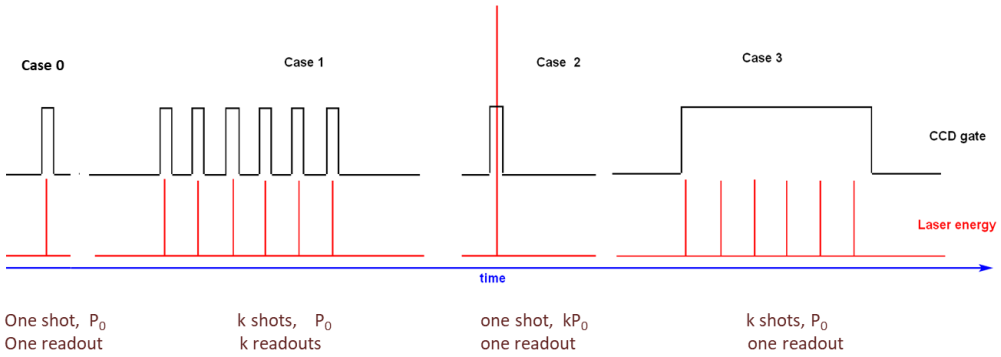


Figure 4.4. Four different acquisition strategies in a spectroscopic experiment[8]

.

Case	Single signal amplitude	Accumulations on CCD	Repetitions	Total signal	Binning factor	Exposure time
0	P_0	1	1	P_0	M	δt
1	P_0	1	k	kP_0	M	$k\delta t$
2	kP_0	1	1	kP_0	M	δt
3	P_0	k	1	kP_0	M	$\Delta T \geq k\delta t$

Table 4.1. List of four different acquisition strategies and relative parameters

The SN associated to each signal is different, and it is not trivial to define the best strategy: in fact, it depends on features both of the signal and the CCD. Given

the signal S_0 measured by the CCD (with dark-subtraction performed) as[76]:

$$S_0 = k\,\bar{G}\,M\,P_0\,Q_e \tag{4.5}$$

with $\bar{G}$ as the gain of the device, M the binning factor, P_0 the signal amplitude and Q_e the quantum efficiency. Given the total noise as from 4.2:

$$N^2 = N_0^2 + N_D^2 + N_R^2 = (\,F\bar{G}\,M\,P_0\,Q_e) + (\bar{G}MN_d\,\delta_t) + N_R^2 \tag{4.6}$$

with F as the excess noise factor, N_d as the dark current and δ_t as the exposure time. The evaluation of SNR for all the acquisition strategies is summarized in tab.4.2.

Case	Signal	Noise variance	SNR
0	S_0	$FS_0 + \bar{G}MN_d\delta t + N_R^2$	$SNR_0 = \frac{S_0}{\sqrt{FS_0+\bar{G}MN_d\delta t+N_R^2}}$
1	kS_0	$k(FS_0 + \bar{G}MN_d\delta t + N_R^2)$	$\sqrt{k}SNR_0$
2(∗)	kS_0	$kFS_0 + \bar{G}MN_d\delta t + N_R^2$	$\sqrt{k}\,SNR_0\sqrt{\frac{A+1}{A+\frac{1}{k}}}$
3(∗∗)	kS_0	$kFS_0 + k\bar{G}MN_d\delta t + N_R^2$	$\sqrt{k}\,SNR_0\sqrt{\frac{B+1}{B+\frac{1}{k}}}$

$$(*)A = \frac{S_0}{\bar{G}\,M\,N_d\,\delta t+N_R^2} \qquad (**)B = \frac{S_0+\bar{G}\,M\,N_d\,\Delta T}{N_R^2}$$

Table 4.2. SNR comparison

Analysing the comparison results, the best acquisition strategy is to deliver all the available energy in a single pulse, to avoid both the accumulation of readout errors and the increase of DN. In fact, the parameter A in **Case 2** is less than the parameter B in **Case 3**, so $SNR_{case2} > SNR_{case3}$. However, the readout noise generally dominates the DN up to very long exposures, as explained in section 4.3.3, so **Case 2** and **Case 3** should be almost equivalent for practical purposes. Moreover, although the best result derives from a single intense pulse within a single exposure, this strategy may lead to several drawbacks related to the practical application:

- it requires high power laser, which could not be an affordable solution (dimensions, cost, power request, etc.);

- the interaction between laser and target can create several substance degradation issues, specially with respect to biological samples;
- if the spectroscopic application requires eye-safety or skin-safety conditions, this strategy could not satisfy the international laser safety directives, as explained in section 4.3.1.

Therefore, for many applications, it can be profitable to deposit the same amount of energy within several pulses of low energy than one single pulse of high energy (i.e. **Case 3** is better than **Case 2**).

4.3.3 Simulation Results

The noise contributions are estimated on three different CCD devices: CCD, EMCCD and ICCD. The simulation parameters are based on the technical specifications of three commercial CCD models: Andor Newton N DU920P (CCD), Andor Ixon 897 (EMCCD) and Andor Istar 320T (ICCD).

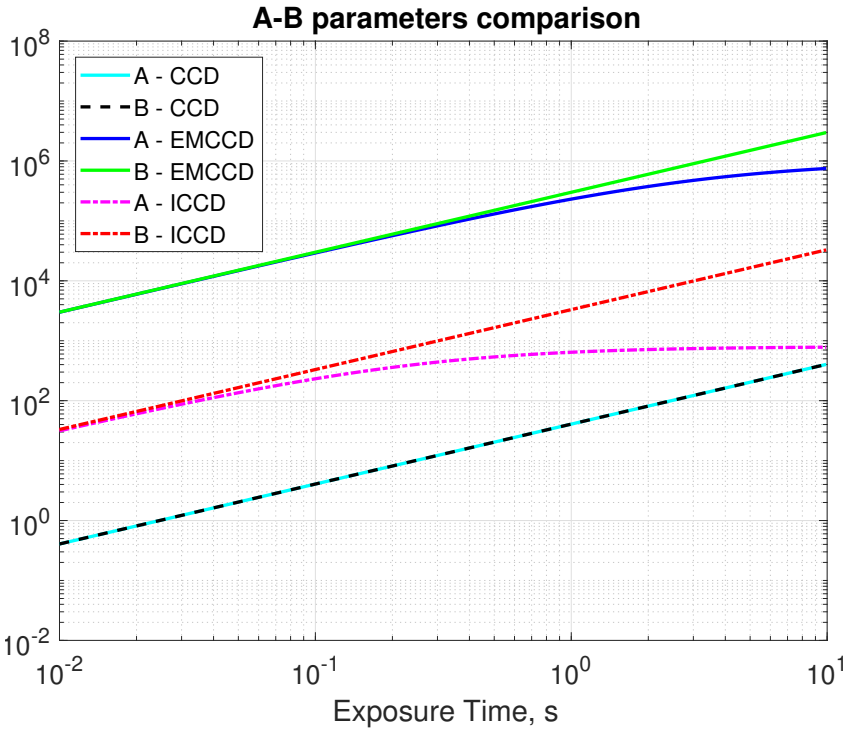


Figure 4.5. A-B values [a.u.] vs Exposure Time [s]

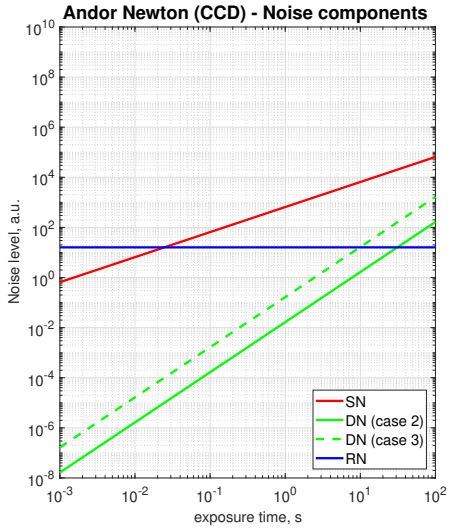
Analysing the two most profitable strategies (**Case 2** and **Case 3**), the first goal

is to determine at which exposure time the dark noise does not overcome the readout noise, given constant signal amplitude and number of pulses. By the numerical simulation, it was possible to demonstrate that SNR increases with the exposure time. The EMCCD device shows the highest SNR values for the whole considered range of exposure times in both cases. Moreover, in Fig. 4.5, the trend of parameters A and B shows clearly at which exposure time the signal accumulation approach (**Case 2**) becomes profitable compared to pulsed signal within a single exposure (**Case 3**), i.e. $A < B$.

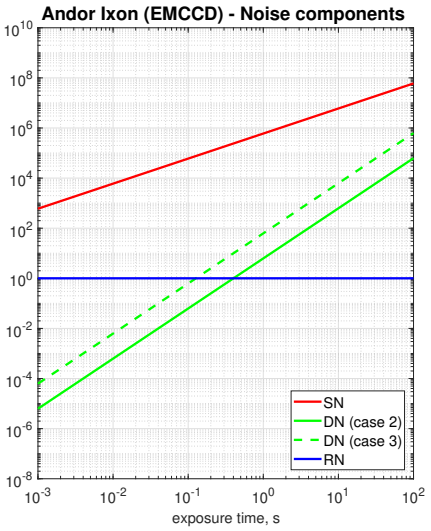
About the noise components in the different time intervals, the simulations produce the results in Fig. 4.6, that can be summarized as:

- In the CCD device the RN dominates the DN up to very long exposure times
- In the EMCCD device the RN dominates the DN until $\delta t \leq 0.1s$; at $\delta t = 0.1s$ DN<RN for accumulated signal (**Case 2**) while for pulsed signal (**Case 3**) DN<RN only with perfect cooling temperature performance; with exposure times $\delta t \geq 1s$ the DN dominates the RN.
- In the ICCD device at $\delta t = 0.01s$ DN<RN in **Case 2**, while in **Case 3** the DN does not overcome RN only if device temperature is less than air cooled temperature (i.e. coolant chiller); with $\delta t \geq 0.1s$ DN dominates RN.

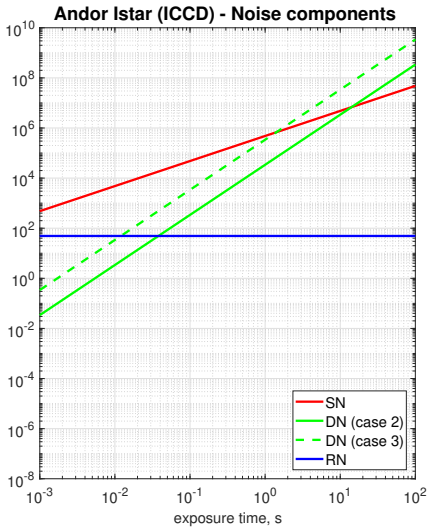
As a conclusion, since Shot Noise is related to the signal amplitude, the trade-off between Dark Noise and Readout Noise results to be a key factor in order to achieve the best performance in a spectroscopic application. For reasonable operative exposure times, the approach to deliver more pulses at lower energy within a single exposure appears to be equivalent to the approach that accumulates all the signal energy in a single pulse. Following the simulations based on CCD technical specifications, the two strategies are practically equivalent with $\delta_t < 1s$ for EMCCD, and $\delta_t < 0.1s$ for ICCD.



(a) Simulation of CCD noise



(b) Simulation of EMCCD noise



(c) Simulation of ICCD noise

Figure 4.6. Noise components [a.u.] vs Exposure Time [s]

4.4 Data Augmentation

Once that we established the best practices to maximize the Signal-to-Noise ratio (the most important feature that influences the classification accuracy in a Raman spectroscopic measurement), we must face a further issue that could raise in an experimental scenario: the lack of a large reference spectra dataset.

As stated in section 3.1, since our aim is to exploit the classification features offered by Neural Networks, we need a proper reference dataset to train the network. If we refer to some domains like, for instance, the minerals, a large quantity of spectra can be found on online databases (e.g. RRUFF Project [80]). For real cases the situation is rather different, because reference resources could not be always available. In addition, even for available dataset, the reference Raman spectra could have been acquired in "perfect" laboratory conditions, resulting quite different from the experimental acquired spectra, so that the subsequent classification could become a difficult task. For these reasons it is crucial to build models robust even in case of a small amount of measures for training. The starting point is thus the retrieval and/or the creation of a sufficiently large dataset for training operations.

A well-known problem in the field of Machine Learning is overfitting[81], i.e. the production of an analysis strictly related to a specific set of data, and therefore hardly accurate for additional or different sets of data. As stated in section 3.5, Convolutional Neural Networks (CNN) have proved to be very accurate in a lot of object detection, object recognition, feature extraction and generally classification tasks[82]. However, their requirement of a large number of training samples (even simple image classification tasks easily reach the order of 1000 examples per class) make them prone to overfitting in case of small-scale datasets. Along with regularization methods to be applied to the network, a common approach to prevent this issue is to generate artificial data that simulate different training samples useful for the network. Within this approach, called Data Augmentation, the options are: to create new data from scratch, or to perturb existing data in order to produce different samples.

Data Augmentation methods are commonly used in Computer Vision tasks that must rely on a large image dataset. The techniques include geometric and color transformation, blurring and noise addition, image blending and many others. An important statement of this approach is that the deformations applied to the labeled training images do not change the semantic meaning of the class they belong to[83]. Several studies have been conducted to demonstrate the effectiveness of these elaborations on 2D data[84][85][86].

Data Augmentation approaches have been proved also useful in speech recognition tasks, where the involved techniques concern time/frequency warping and/or masking, speed perturbation and superimposing noise to clean audio data[87].

Although brand new Data Augmentation techniques (e.g. GANs, Neural Style Transfer, Neural Architecture Search) achieved significant results during the last years, some questions remain open[88]: there is no consensus about the best strategy for combining oversampling and data warping techniques, no techniques taxonomy have been established yet, and there is the necessity to extend the augmentation principles to other data types.

Studies about Data Augmentation applied to spectral data have been conducted in several scenarios: in [89] the difficulty to generate coherent augmented vibrational spectroscopic data in absence of a valid noise model is underlined. In [90] augmentation techniques like random variations in offset, multiplication and slope on NIR spectra are used. In [91] an additive white Gaussian noise added to the original Raman signal is used. In [92] two augmentation steps are proposed for its Transfer-learning-based Raman identification, i.e. an additive white noise and a right/left shifting of Raman wavenumbers. In [93] an augmentation pipeline mixing theoretical data and experimental data of X-ray diffraction spectra is developed.

In the next sections an innovative Data Augmentation method to improve Raman spectra classification is proposed. This method exploits both the aforementioned pre-processing algorithms (section 3.2, section 3.3) and a further Machine Learning enhancement through the Generative Adversarial Networks (GANs).

4.5 Hierarchical Pipeline

In order to overcome the issue of the lack of a large training set in a spectral classification scenario, we propose a hierarchical pipeline of data augmentation steps that involves several signal processing algorithms whose aim is simulating the widest range of possible cases in an experimental Raman measurement. Starting from a single sample for each labelled class, the pipeline generates thousands of artificial spectra per class that preserve, according to data augmentation principles, the semantic meaning of the class. In the present case, the characterization of the

labelled class is related to the peaks position and intensity, that is not altered by the chosen algorithms. The pipeline will be eventually tested in a real use case study in chapter 5.

The basic idea behind the pipeline method is to replace the approach of pre-processing every single raw Raman spectrum acquired in an experimental scenario (a common practice to obtain good matching performance yet time/resources-consuming, requiring human supervision and often dependent of the measurement peculiarities) with a single automatic pipeline to be performed before the Raman measurements. In fact, instead of executing the spectra elaborations multiple times, an augmentation pipeline can be performed only once for all the labelled classes in our dataset, converting the baseline removal and noise filters in data augmentation tools, in order to immediately feed the raw acquired data into the classifier (in our case a CNN) at acquisition time, so that a real-time classification become possible despite of the data quality and their signal-to-noise ratio.

The data augmentation pipeline is described step by step in the next paragraphs, starting from a single reference Raman spectrum until a consistent set of thousands of spectra belonging to the same labeled class is obtained through subsequent elaborations. All the steps are executed for each class of spectra, and furthermore the partial results at the end of each step are saved to memory, both for debugging and validation purposes. The whole pipeline script and the relative plots of the present work have been developed using MATLAB®.

Additive White Noise The first data augmentation step consists in adding a white Gaussian noise (AWGN) to the original signal, varying the level of signal-to-noise ratio with 5 different values, in order to obtain 5 new spectra at the end of this process. This kind of noise model is widely used in signal analysis and information theory to mimic the effects of many random processes that could occur in an experimental measurement. It is also very popular as a data augmentation method in most image recognition tasks, and in general in deep learning tasks that suffer a lack of sufficient number of samples. The number of 5 levels is purely empirical and prove to be effective for this kind of classification, but nothing prevents us from

using different values. On the other side, the choice of placing this noise-adding step at the beginning of the pipeline, and not at the second step is motivated from the fact that applying baseline removal algorithms to noisy spectra produces more unpredictable results than adding white noise to already baseline removed spectra. The choice of an additive white Gaussian noise to generate new spectra is related to the idea of distributing uniform random values across the whole frequency band, thus avoiding the risk to overlap a possible frequency interval in which characteristic Raman peaks are present. Moreover, the average distribution of noise tends to be Gaussian (because of the central limit theorem).

Baseline removal algorithms The second data augmentation step consists in applying a series of baseline removal algorithms to the original and noisy-augmented spectra generated in the previous step. Raman spectra can be generally represented as a number of peaks and noise superimposed on a background (baseline) as explained in details in section 3.2.

Since experimental acquired spectra can often show a prominent baseline slope, the choice to generate augmented spectra with the application of several baseline removal algorithms is related to the idea of simulating the widest range of fluorescence contributions in order to eventually train the neural network with the largest number of possible cases. In particular, seven baseline correction methods have been applied to the augmented spectra of the previous step, varying for each one of them the most significant parameters, with the aim of generating a various set of simulated spectra from the ones with a still prominent fluorescence curve to the flattest ones. The algorithms used in this pipeline step are the ones described in section 3.2:

- Rolling Ball, 2 variable parameters
- Rubber Band, 1 variable parameter
- Asymmetric Least Squares (AsLS), 2 variable parameters
- Adaptive Iteratively Reweighted Penalized Least Squares (airPLS), 1 variable parameter

- Asymmetrically Reweighted Penalized Least Squares (arPLS), 1 variable parameter
- Minima Identification and Discrimination via Adaptive Least Squares Thresholding, 1 variable parameter
- Polynomial Curve Fitting, no variable parameters

Noise Reduction Filters The third data augmentation step consists in applying a series of noise removal filters to the two different sets of spectra generated in the previous steps. In details, the filters are applied both to the noisy spectra from the first step and to the noisy baseline-removed spectra from the second step. This method allows to generate a large artificial set of Raman spectra that can be ideally divided in four different subsets: spectra with a significant component of noise and fluorescence (i.e. prominent baseline curve), spectra with a flat baseline and a significant component of noise, spectra with a significant component of fluorescence but noise filtered, and spectra that show minimal amount of noise and fluorescence. The final aim of this augmentation step is to obtain a careful balance of the four subsets that will serve as the most possibly precise training input to the artificial neural network. The simulation of the whole range of cases that could occur during a Raman spectroscopic measurement is crucial to achieve an optimal classification rate with this machine learning approach.

In particular, five different smoothing filters have been chosen, varying for each one of them the most significant parameters. The algorithms used in this pipeline step are the ones described in section 3.3:

- Savitzky-Golay filter, 2 variable parameters
- Wiener filter, 1 variable parameter
- Moving average filter, 1 variable parameter
- Percentile filter, 2 variable parameters
- Undecimated Wavelet transform, 1 variable parameter

Clustering, shifting and merging data Once the whole set of augmented spectra has been generated, it is necessary to cluster the spectra in several subsets in order to achieve a balanced training set for the neural network. As explained in the previous paragraph, 4 different subsets are identified to represent the Raman spectra cases following their amount of fluorescence and noise. (e.g. Cluster 1 with noisy and baseline-contribution spectra, Cluster 2 with noisy and baseline removed spectra, Cluster 3 with noise-filtered and baseline-contribution spectra, Cluster 4 with noise-filtered and baseline removed spectra). The clustering algorithm used for this purpose is the k-means clustering, due both to its success in cluster analysis for data mining scenarios, and mostly to its suitable approach of vector quantization for K data groups generated from normal (Gaussian) distributions.

A preliminary elaboration of k-means algorithm is performed over all the spectra datasets in order to find an average value of the clustering sets size, and to examine the standard deviation trend for every class. A suitable average value is calculated as a result of the average of the minimum size subsets of every labelled class. Once a subset size value has been established, this quantity of spectra is collected from each of the 4 k-groups, and finally merged in a single set that reasonably includes a coherent representation of several cases of Raman spectra measurements.

A last augmentation step consists in slightly horizontally shifting the one dimensional array that stores the spectra values. The index positions of the array represent the Raman-shifts [cm^{-1}], so this further processing aims to simulate a possible effect of any Raman spectroscopic acquisition, i.e. the shifting of the characteristic peaks due to intrinsic chemical bound energies and/or extrinsic measurement devices configuration. The choice of performing this step at the end of the pipeline is due to save computation time and memory resources.

Once a quantity of 2500 spectra per-class is achieved, this set will serve as the neural network positive training set for the specific class. A parallel elaboration that leverages as well the k-means clustering will serve as the pigments negative set: each 2500 spectra set of a specific class will be divided into the quantity of the other labeled classes, and each one of this chunks will compose a portion of the negative training set of the other classes. Hence, for each class, the Positive Set represents

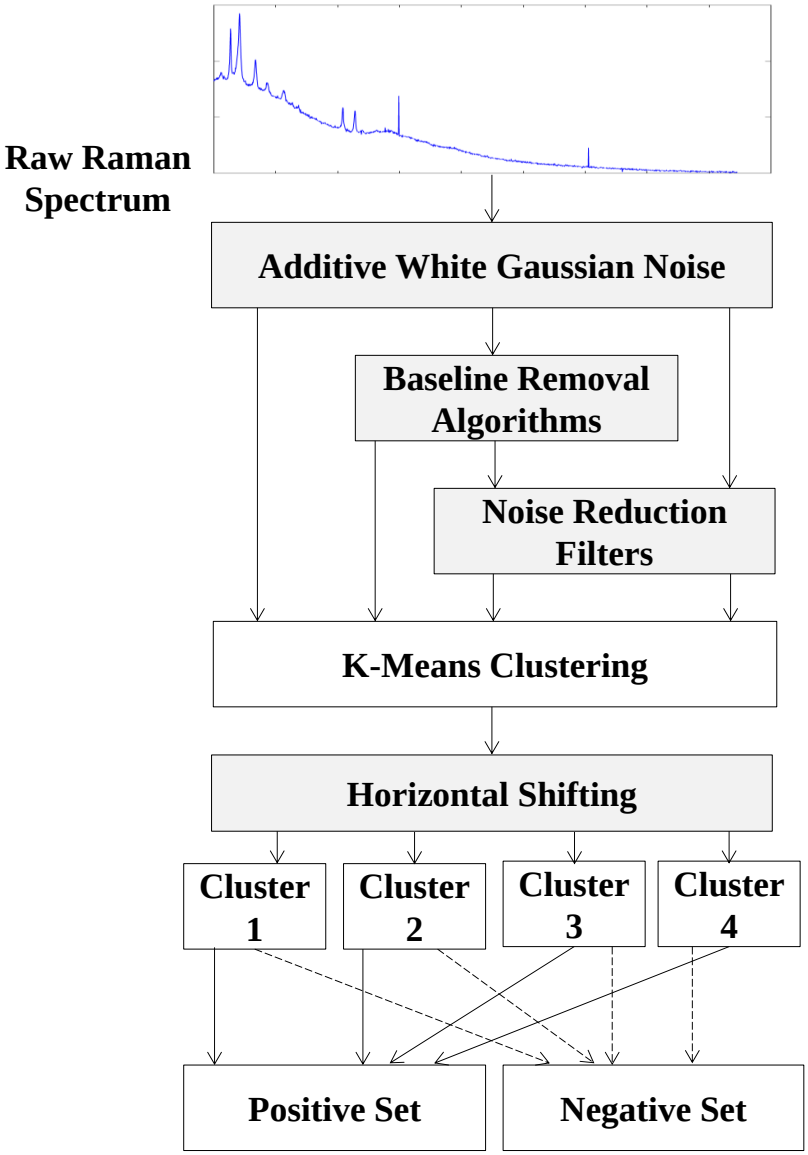


Figure 4.7. Data Augmentation Pipeline Flow-chart

the spectra that would be labeled as *True* in a binary classifier, while the Negative Set includes samples from all the other classes that would be labeled as *False* in a binary classifier. In the case of a categorical classifier there would be no need to set up a parallel procedure to generate a negative set. A summary flow chart of the entire data augmentation pipeline is represented in Fig. 4.7. The quantity of 5000 spectra per labelled class is aligned with the most usual practices regarding

the training-set size requirements in deep learning scenarios.

4.6 Generative Adversarial Networks

Once that we generated our spectra input set with the aforementioned data augmentation pipeline, the question is: can we further enhance data augmentation with Machine Learning tools? Since the augmentation algorithms used so far belong to the signal processing field, it can be interesting evaluating also Machine Learning techniques that could improve the quality of our input set and eventually boost the accuracy of the final classifier. The ML tool chosen for this task is Generative Adversarial Networks.

Generative Adversarial Networks (GANs) [94] are a class of machine learning frameworks that consist of two models that are trained in parallel: the Generator is trained in order to generate fake data, while the Discriminator is trained to discern the fake data from real examples of the original training dataset. The word *generative* indicates the overall aim of the model that is creating new data with the same characteristics of the training set. Therefore, this technique can suit our purpose to artificially enrich our spectral dataset with different cases that will improve the accuracy of the final classifier. GANs have become a common tool for data augmentation, especially with 2D data [95][96].

In this work, the aim of the Generator is to produce Raman spectra whose characteristics are comparable to the ones of the training dataset, so much that the samples it generates look indistinguishable from the training data. Otherwise, the goal of the Discriminator is to discern whether an input spectrum is real (coming from the training dataset) or fake (created by the Generator). So, as the Generator improves in producing realistic Raman spectra, the Discriminator improves in its capability at distinguishing fake data from real data, and both networks continue to improve simultaneously. A scheme of the GAN architecture is represented in Figure 4.8[3].

At a certain epoch of the GAN training process, the spectra created by the Generator are supposed to be as useful for the input set of the Raman spectra

[3]Figure taken from [9] and partially edited for our purpose

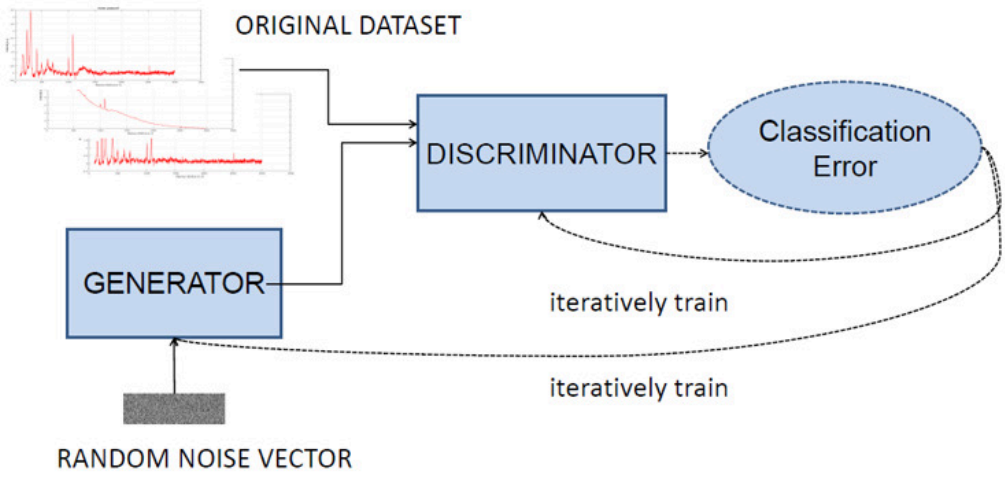


Figure 4.8. GAN architecture [9]

classifier as any spectra of the real dataset, resulting in a perfect tool for data augmentation. This achievement can be obtained when the loss function of the Generator reaches the minimum value and, simultaneously, the accuracy of the Discriminator gets near to its value of 50%, meaning that the Discriminator is randomly guessing whether its input sample is real or fake. Analysing these network parameters, we can detect when the fake spectra produced by the Generator are practically indistinguishable from the real ones. As the GAN architecture can be related to a zero-sum game, this condition can be defined as a Nash Equilibrium, i.e. a point at which neither player can improve their situation or payoff by changing their actions. However, given the extremely complexity of the GAN's non-convex optimization, it is not possible to find the existence of the Nash equilibrium [97]. In this situation, the choice of the number of training epochs is purely empirical.

One of the key parameters in the GAN architecture is the raw input vector of the Generator network (z) indicated as Random Noise Vector in Figure 4.8. This input is a vector of initially random numbers that are used as a starting point for producing artificial samples. This vector can be defined as a latent space, i.e. a set of variables that act as a representation in a smaller dimension of other observable variables, in our case the resulting spectra. In our use case study, a data-driven

analysis will be used to observe the potential best values to determine the dimension of the z vector.

Before using the GAN approach, in order to enrich our limited pigment dataset with artificial spectra, a crucial challenge must be faced: how to train a Generative Adversarial Network in order to serve as a data augmentation tool since it needs many input samples as well to work properly? It looks pretty clearly that a GAN cannot be trained with a single or few spectra as it is composed our initial dataset. So, in this case, the GAN augmentation approach cannot be alternative to the augmentation pipeline approach described in the previous section, but rather complementary.
One of the aims of the present research is to experiment combinations of both approaches in order to obtain different subsets of artificially generated spectra. These subsets will be subsequently tested on a range of neural network architectures to find the most accurate Raman spectra classifier, and therefore the most effective data augmentation algorithm.

The combination of signal processing algorithms and GAN enhancement to produce an effective Raman spectra input set in order to train the Raman classifier with the highest accuracy is tested in details in the next chapter. A real case of study about pigment spectra classification in a non-laboratory scenario is carried on to evaluate the proposed innovative method that can lead to the implementation of a real-time Raman spectra classifier.

Chapter 5

Use Case Study

In this chapter an experimental use case study about Raman spectra classification is developed and analysed, demonstrating the effectiveness of the innovative data augmentation method presented in the previous chapter. This study concerns historical art pigment identification, a common task in the Cultural Heritage field of study. Starting from a pigment spectra dataset where each class contains just a single Raman spectrum, the aforementioned hierarchical pipeline of data augmentation steps (sec. 4.5) is developed and reinforced by the use of the GANs approach (sec. 4.6). At the end of this data augmentation process, three neural network models are examined and compared, training them through the pigment dataset. The aim is to obtain a neural network model able to classify in real-time the Raman spectra of the pigments, that are supplied as input, with the best possible accuracy. The experimental results demonstrate the correctness of the data augmentation approach with the proposed pipeline, showing moreover an evident improvement in the classification accuracy with the GANs reinforcement.

5.1 Cultural Heritage

Raman spectroscopy is a well-established technique for rapid analysis for Cultural Heritage characterization. In particular, for historical and artistic pigments, Raman spectroscopy represents one of the most used and reliable technique being non invasive and usable, even on-site, without any sample preparation[31]. Exactly for the role

played by Raman spectroscopy in this frame, the automatic classification of spectra of unknown materials would be a result of great interest. Actually, overall in the last decade, several methods have been tested with this purpose[98][99][100][101][102]. As stated in section 2.6, even in this field we can divide chronologically the classification approaches in pre and post deep learning era. However, as in other classification tasks, the problem remains still far from an ultimate solution, especially in non-laboratory scenarios (Fig. 5.1).



Figure 5.1. A spectroscopic diagnostic campaign led by ENEA in the Catacombs of Priscilla - Rome, July 2016

The spectra dataset that we refer to contains historical artistic pigments, which are earth and mineral pigments. In particular, we deal with the Renaissance ones. If we refer to the minerals, a large quantity of spectra can be found on online databases

(e.g. RRUFF Project [80]). For real cases the situation is rather different. In order to identify the belonging to a specific studios, ateliers or ancient and modern retouching (even fakes) you may have to refer to very specific pigments present on already done artworks, for which the number of measurements and therefore of spectra available can be very limited[103]. In particular for these cases, it is important to build models that result robust also in case of a small amount of measures for training. The recent advances in the applications of Raman spectroscopy for cultural heritage diagnostic and investigation purposes, especially in the development of better transportable and mobile Raman devices for on-site measurements, are boosting the interest around real-time and efficient spectra classification systems[104] exploiting machine learning techniques. Thus, the starting point of this approach is for sure the retrieval of a sufficiently large dataset for training operations, with the final proposal of a neural network model able to classify a given pigment, having as input a specific Raman spectrum.

The most widespread applications and goals of pigment classification can be summarized as follows:

- identify the authors, the studios and the epoch in which an artwork has been created;
- retrieve information from pieces of work that are damaged and no more visible in order to reconstruct them;
- detect hazard substances on the artwork (mold, mushrooms, ...);
- fight art forgery in order to uncover fake artworks.

Before diving into the algorithm development, we first analyse the historical artistic pigment dataset that serves us as basis for the classification task.

5.2 Pigment Dataset

In the Diagnostic and Metrology Laboratory of the ENEA Research Centre of Frascati (Rome), a set of about 70 fresco specimens were prepared with pigments and binders, typical of the Renaissance period in Rome, in order to build a database, as

complete as possible, of spectroscopic data useful for their characterization. Aiming at providing the restorers and art historians with a useful tool for the study of ancient frescoes, the samples have been prepared in agreement with the Cennino Cennini recipes[105], both for materials and procedures[1]. Moreover, much attention was paid to the geographic origin and chemical composition of plaster (intonachino) and pigment components. Techniques and materials used for the sample preparation are described in more detail elsewhere[106]. For this work we took into account 38 samples, avoiding mixtures and more complex preparations.

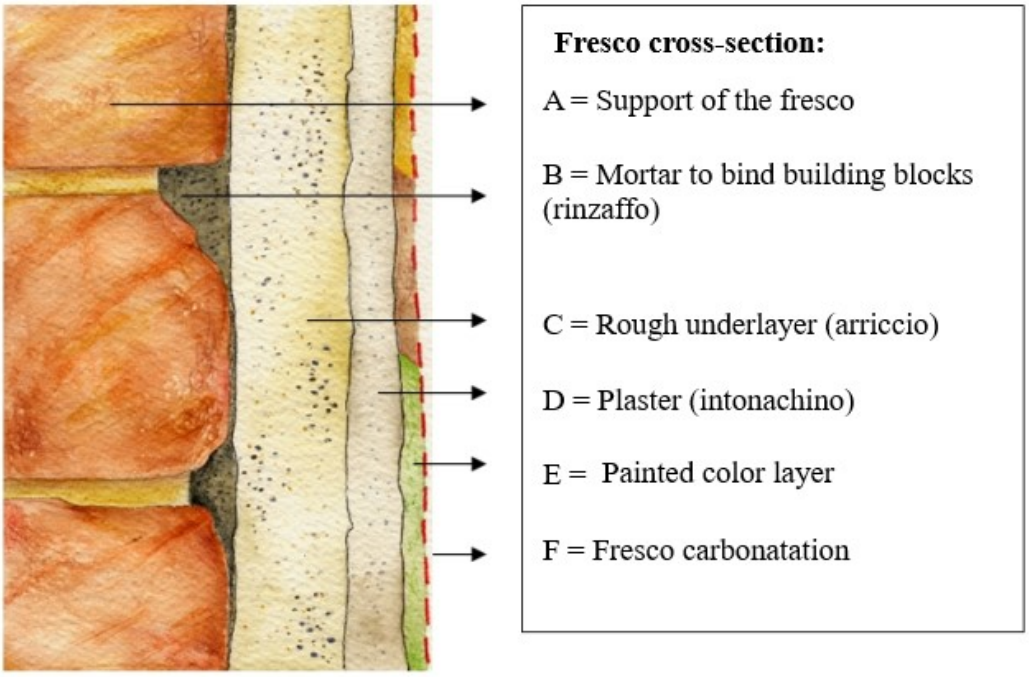


Figure 5.2. Fresco cross-section with relative layers (adapted from https://benedettalopezbani.wordpress.com)

For the sake of clarity, we list the typical layers of which a fresco is composed, as it can be seen in Fig. 5.2:

(A) Support - the supporting wall upon which a fresco is painted. The wall can be made of bricks, stones or mixed materials.

[1]Pigments have been purchased from Zecchi - Serie Cennino Cennini dei Pigmenti Antichi - http://www.zecchi.it/

(B) Rinzaffo - a workable paste made of mortar which harden to bind building blocks, in order to fill and seal the irregular gaps between them.

(C) Arriccio - a rough underlayer made of coarse plaster, that is allowed to dry for some days.

(D) Intonachino - the final, very thin layer of plaster on which a fresco is painted.

(E) Pigmento - the painted color layer, which it is traditionally made of earth and mineral compounds.

(F) Carbonatation - the process that fixes the colour in the plaster and ensures durability of the fresco for future generations.

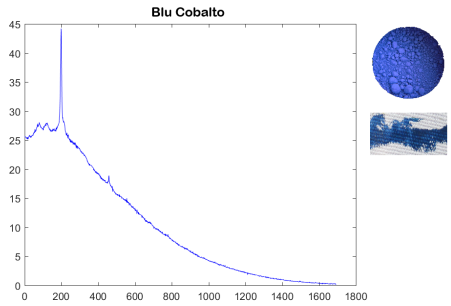
Most of the pigments chosen for this research have been already studied in many works, both in laboratory and on-site experiments, confirming the interest about their characterization and the implications related to their use[31][107].
In appendix .1 a complete list of the pigments used in the present research can be examined.



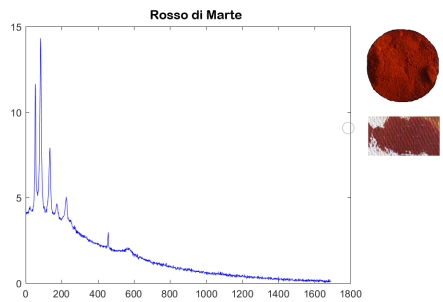
Figure 5.3. The commercial Raman System used for sample measurements (from https://bwtek.com/)

Raman spectroscopy acquisitions were carried out with a micro-Raman system: a BWTek i-Raman system equipped with a laser source emitting at 785 nm, a cooled

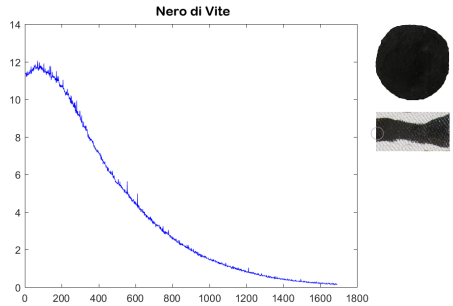
CCD detector array (2048 pixels) with spectral resolution of 3 cm^{-1} in the spectral range 0–4000 cm^{-1} and a contact probe that can be replaced with a microscope with objectives from 4x to 80x (Fig. 5.3). In particular, for this dataset, measurements have been carried out with the contact probe. For every sample at least three different measurements in adjacent points of the surface have been recorded and averaged to obtain a mean spectrum. The choice of selecting the mean spectrum as the final reference spectrum, instead of taking into account the different measurements as distinct reference samples that could have enriched the initial dataset, is justified by the minimal spectra fluctuations that result from the different measurements acquired within the same laboratory conditions. Data relative to Raman shift lower than 100 cm^{-1} and higher than 3000 cm^{-1} were not considered because they do not have analytical value for our purpose. Therefore, vector data taken into account in this work have size of 1692.



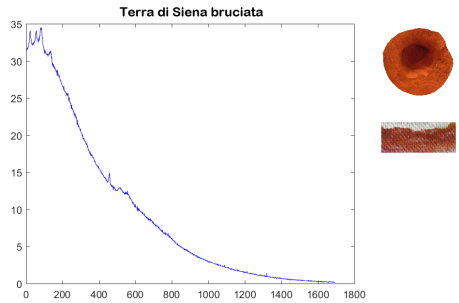
(a) Cobalt Blue Raman spectrum



(b) Red Hermatite Raman spectrum



(c) Vine Black Raman spectrum



(d) Brown Iron Oxide Raman spectrum

Figure 5.4. Raman spectra of four pigment samples

In fig. 5.4, Raman spectra of four pigment samples are depicted. Some important

features to be noted are the prominent slope caused by the fluorescence, and the different signal-to-noise ratio among the samples, which causes different ranges of intensities of the characteristic peaks.

5.3 Pigment Data Augmentation Pipeline

After the initial acquisition of raw Raman spectra, our dataset is composed of 38 different pigments. Each one of them is a particular chemical compound, and therefore has its unique Raman spectrum. These Raman spectra measurements are represented as one-dimensional arrays which will be used as input of our neural networks. Thus, our initial dataset is basically a matrix of dimension 38×1692. The acquired spectra show some of the issues that could affect a correct classification, like noise, fluorescence or even peak shifting. However, since we decided to use neural networks, our main problem is the limited available dataset for the training set, because the dataset contains just one reference spectrum for each pigment class. In order to generate a suitable input dataset, we apply the data augmentation pipeline described in section 4.5, so that we are able to obtain a balanced training set that simulate the widest range of cases in Raman spectroscopy measurements. The whole pipeline has been developed with MATLAB® mainly for the available functions which implement the pre-processing algorithms included in the pipeline. Anyway, a software porting into the Python language is currently in progress.

5.3.1 Additive White Noise

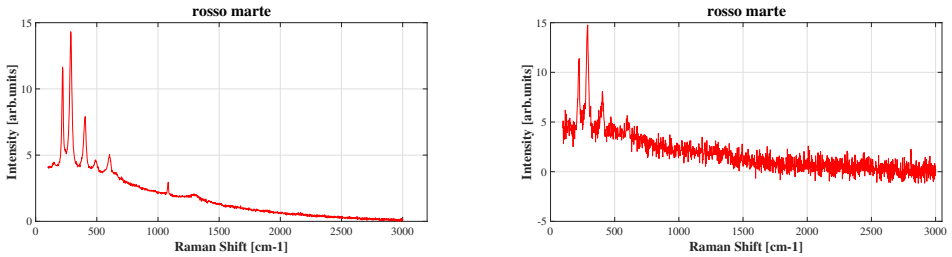
The first step involves the use of additive white noise to the raw signal. Five different values of noise level are used, in order to obtain five new spectra from the single reference one. Code for white noise augmentation is provided in listing 5.1.

```
compound = 'rosso_marte';
%Load the single original spectrum
raman_shift = dlmread(path_raman);
intensities = dlmread([pathin compound '_orig.txt']);
%Select only the wavelengths of interest
raman_shift = raman_shift(139:1830);
intensities = intensities(:,139:1830);
```

```
total_intensities = intensities;
% Add noise
total_noise_intensities = [];
for tt=1:size(total_intensities,1)
    spectrum_to_add = total_intensities(tt,:);
    %for each level of noise
    for nn=5:10:45
        % ADDITIVE WHITE NOISE FUNCTION
        wn_intensities = awgn(spectrum_to_add,nn);
        total_noise_intensities = [total_noise_intensities;
            wn_intensities];
    end
end
%Save original and augmented spectra to file
aug_intensities = [total_intensities;total_noise_intensities];
dlmwrite([pathin compound '_aug.txt'],aug_intensities,'delimiter','
    \t');
```

Listing 5.1. White Noise augmentation

At the end of this step, 6 total spectra are obtained from 1 raw reference spectrum. An example of generated spectrum from the pigment class *Rosso Marte* is depicted in fig. 5.5.



(a) Original raw spectrum **(b)** Spectrum with additive white noise

Figure 5.5. Example results of the first step of the pipeline

5.3.2 Baseline Removal

The second step involves the use of 7 baseline removal algorithms, each one of them with its variable parameters as described in section 4.5. Code for one of the

baseline removal algorithm augmentation (airPLS) is provided in listing 5.2 as an example. The other algorithms follow the same code structure.

```
%Load augmented spectra from previous step
compound = 'rosso_marte';
intensities = dlmread([pathin compound '_aug.txt']);
%AirPLS algorithm
airpls_intensities = [];
%for each value of parameter lambda
lambdas = [1e2,1e3,1e4,1e5,1e6];
for l=1:size(lambdas,2)
    lambda = lambdas(l);
    %for each input spectrum
    for i=1:size(intensities,1)
        raw_spectrum = intensities(i,:);
        %Apply AIRPLS
        [airPLS_fun,xb]=airPLS(raw_spectrum, lambda,2,0.05);
        br_spectrum=airPLS_fun;
        airpls_intensities = [airpls_intensities;br_spectrum];
    end
end

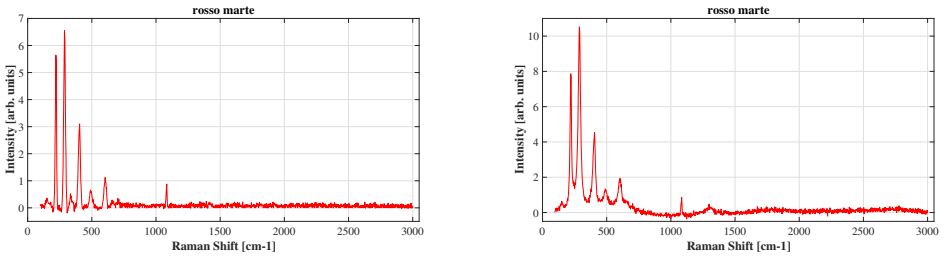
%Apply all the other baseline removal algorithms
...

%Save generated spectra to file
baseline_removed_spectra = [rolling_ball_intensities;
    rubber_band_intensities;
    asls_intensities;airpls_intensities;arpls_intensities;
    weakley_intensities;poly_intensities];
pathout = [pathin_alt compound '_baseline_rem.txt'];
output_set = [intensities;baseline_removed_spectra];
dlmwrite(pathout,output_set,'delimiter','\t');
```

Listing 5.2. Baseline removal augmentation

At the end of this step, about 1100 total spectra are obtained from 6 input spectra from first step. The number of generated spectra in this step can slightly change from class to class since baseline removal algorithms can produce a different number

of outputs due to the specific input spectra structure. An example of generated spectra from the pigment class *Rosso Marte* is depicted in fig. 5.6.



(a) Baseline removed spectrum with airPLS **(b)** Baseline removed spectrum with MinId

Figure 5.6. Example results of the second step of the pipeline

5.3.3 Noise Reduction

The third step involves the use of 5 noise reduction algorithms, each one of them with its variable parameters as described in section 4.5. One important thing to keep in mind is that, in this step, the noisy spectra from first step and baseline removed spectra from second step are processed in parallel by the noise reduction algorithms. The reason for this choice is to obtain both noise reduced spectra with baseline and noise reduced spectra without baseline. Code for one of the noise reduction algorithm augmentation (Savitzky-Golay filter) is provided in listing 5.3 as an example. The other algorithms follow the same code structure.

```
%Load augmented spectra from previous steps
compound = 'rosso_marte';
intensities = dlmread([pathin compound '_aug.txt']);
baseline_removed_spectra = dlmread([pathin compound '_baseline_rem.txt'
    ]);

%Calculate a factor that reduces the generated spectra at the end of
    this step
jump_factor = round(size(baseline_removed_spectra,1)/size(intensities
    ,1));

%Savitzky-Golay
    sgolay_intensities = [];
```

```
    sgolay_br_intensities = [];
    %for each value of parameter order
    for order=1:10
        %for each value of parameter window
        for window=1:2:21
            if order<window
                %Cycle on augmented noisy spectra
                for i=1:size(intensities,1)
                    raw_spectrum = intensities(i,:);
                    dn_spectrum=sgolayfilt(raw_spectrum,order,window);
                    sgolay_intensities = [sgolay_intensities;
                        dn_spectrum];
                end
                %Cycle on baseline removed spectra
                for j=1:jump_factor:size(baseline_removed_spectra,1)
                    br_raw_spectrum = baseline_removed_spectra(j,:);
                    br_dn_spectrum = sgolayfilt(br_raw_spectrum,order,
                        window);
                    sgolay_br_intensities = [sgolay_br_intensities;
                        br_dn_spectrum];
                end
            end
        end
    end

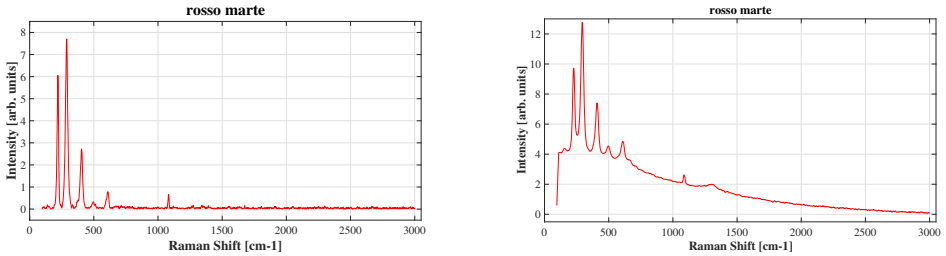
%Apply all the other noise reduction algorithms
...

%Save generated spectra to file
noise_removed_spectra = [sgolay_intensities;wiener_intensities;
        avg_intensities;percent_intensities;dwt_intensities];
baseline_noise_removed_spectra = [sgolay_br_intensities;
    wiener_br_intensities;
        avg_br_intensities;percent_br_intensities;dwt_br_intensities];
pathout = [pathin_alt compound '_noise_rem.txt'];
output_set = [intensities;baseline_removed_spectra;
    noise_removed_spectra;
            baseline_noise_removed_spectra];
```

```
dlmwrite(pathout,output_set,'delimiter','\t');
```

Listing 5.3. Noise reduction augmentation

At the end of this step, about 1900 spectra are obtained from the input spectra of first step and about 2800 spectra from the input spectra of second step, for a total amount of 2700 generated spectra. Even in this step the number of generated spectra can slightly change from class to class. An example of generated spectra from the pigment class *Rosso Marte* is depicted in fig. 5.7.



(a) Baseline and noise reduced spectrum **(b)** Noise reduced spectrum

Figure 5.7. Example results of the third step of the pipeline

5.3.4 K-Means Clustering, Shifting and Merging

The fourth step involves the division of the whole set of augmented spectra in four subsets that must hopefully represent the most common cases in Raman experimental measurements: spectra with a significant component of noise and fluorescence (i.e. prominent baseline curve), spectra with a flat baseline and a significant component of noise, spectra with a significant component of fluorescence but noise filtered, and spectra that show minimal amount of noise and fluorescence. Some description plots of the results given at the end of K-Means clustering are presented in fig. 5.8. The graphs demonstrate how the K-Means algorithm applied to the augmented pigment data results in a coherent and effective clustering of our dataset. The four different subsets will be eventually used to train the deep neural network, adhering to the requirements of a balanced training set. The effectiveness of our balancing technique will result crucial to obtain a high level of accuracy of the neural network classifier.

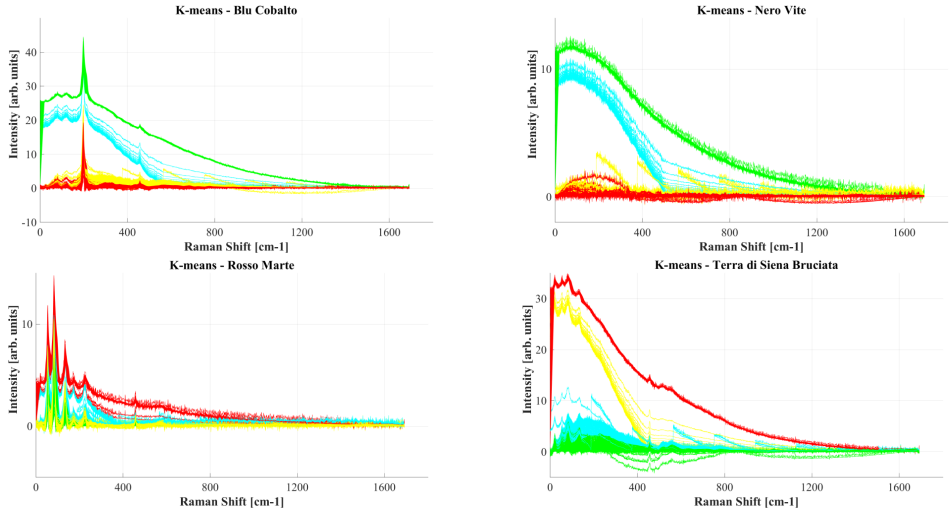


Figure 5.8. K-means Clustering of some pigment classes

Once that the 4 subsets are created, we apply a final augmentation step that involves a small horizontal array shifting, and finally we obtain an amount of 5000 spectra per class which are ready to be fed into the classifier. The hierarchical pipeline has hugely augmented the original 38 reference spectra in $38 \cdot 5000 = 190000$ artificial spectra. Each class can be represented as a matrix of dimension 5000×1692 Since we planned to test the augmented dataset both on a binary neural network and on a categorical neural network, we further divide each one of the pigment set in a Positive Set (i.e. a set that contains spectra of that specific class) and a Negative Set (i.e. a set that contains spectra of all the pigment classes except the ones of that specific class). This final step is mandatory if we want to test a binary neural network.

5.4 GANs Augmentation Enhancement

Once we generated an augmented artificial dataset of pigment Raman spectra through the application of several signal processing algorithms, the subsequent question is: can we further enhance the quality of the input dataset with Machine Learning tools? The answer resides in the use of Generative Adversarial Networks as a further data augmentation tool. Eventually, in order to demonstrate the

effectiveness of the GANs enhancement, the final classifier will be trained and tested with both the artificial datasets, so that a comparison between the two approaches could be analysed.

As stated in section 4.6, the challenge that must be faced in this kind of tasks is the input set for the GAN. Since GANs themselves are neural networks, they need a proper large input set to work properly. Hence, how to train them, given our very small original reference database? In this use case we are bound to use the GANs not as an alternative augmentation tool, but rather as a complementary tool along with the hierarchical pipeline. In fact, since we have just 1 reference spectrum for each of the 38 pigment classes, training GANs with only these spectra would result in generating useless data that would show no differences with the reference spectra.

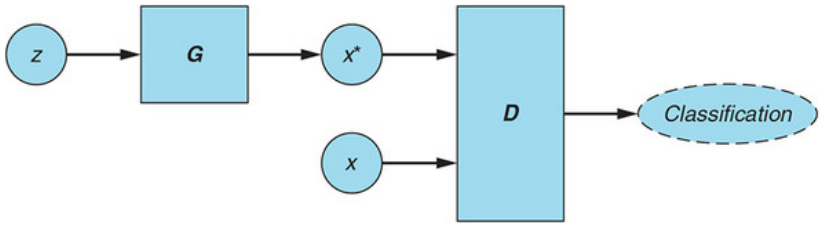


Figure 5.9. GAN architecture scheme [9]

As we are forced to use both the augmentation approaches, we experiment with combinations of the two approaches in order to obtain different subsets of artificially generated spectra. Each one of the 3 subsets corresponds to an output of an intermediate step of the hierarchical pipeline. These spectra set are fetched from the memory and fed into the GAN to test at which step the pipeline produces a good quality dataset that allows the best generalization by the GAN. The input sets fed to the GAN are:

- D_1 = Original spectrum + White Gaussian Noise spectra
- D_2 = D_1 + Baseline Corrected spectra
- D_3 = D_2 + Noise Corrected spectra

As explained in section 4.6, GANs are composed of two different neural networks: the Generator, whose aim is to produce fake input as similar as possible to the

original dataset, and the Discriminator, whose aim is to discern with the best accuracy a fake example from a real one.

If we analyse closer the architecture (a scheme is depicted in fig. 5.9), we shall find that the Generator takes in a random noise vector z to produce the fake spectrum $x^\star$, thus $G(z) = x^\star$. This z input is a vector of initially random numbers that are used as a starting point for producing artificial samples. This vector can be defined as a latent space, i.e. a set of variables that act as a representation in a smaller dimension of other observable variables, in our case the resulting spectra.

In our use case study, a data-driven analysis has been used to observe the potential

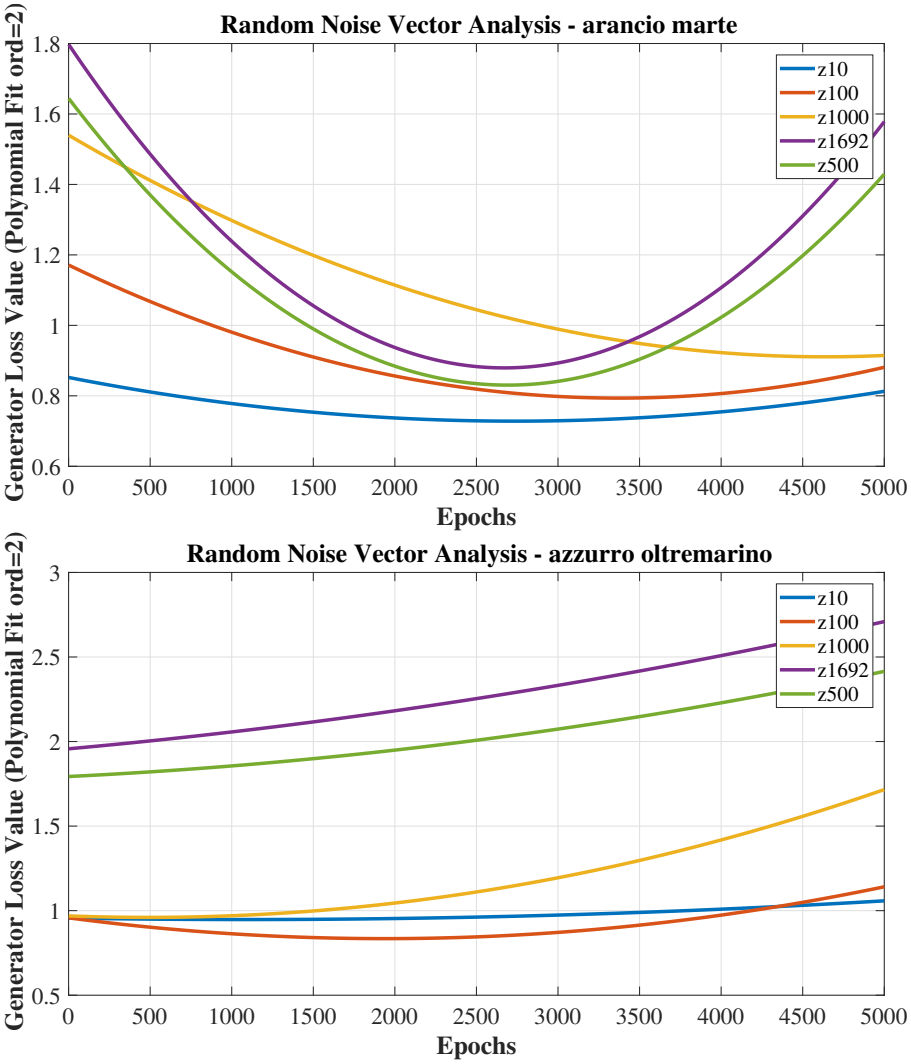


Figure 5.10. Generator Loss Values Trend for different z values

best values to determine the dimension of the z vector. It has been observed that, for all the pigment classes, the best choice for z is a value between 10 and 100, as it is presented in Figure 5.10, where it is plotted the trend of Generator Loss Value in different z configurations for a couple of chosen classes. For all the next experiments it has been chosen a z value of 100. Recall that, as any other neural network, the Generator Loss Value is a parameter that must be kept as low as possible, since it measures the discrepancy error between the output of a labeled input and the output of a predicted one. In GANs, the Generator loss is calculated from the Discriminator's classification: it gets rewarded if it successfully fools the discriminator, and gets penalized otherwise.

We start to experiment GAN data augmentation with the D1 spectra subset. For all the tests we keep the GAN hyperparamters fixed: the GAN is trained for 6000 epochs, a learning rate of 0.2 and a batch size of 32 are chosen. The number of epochs are set due to the fact that after the selected value, the Generator appears to not improve its fake data creation. As the previous augmentation steps, the pigment class selected as example is *Rosso Marte*. The main method that develops GAN training algorithm is provided in listing 5.4.

```
#Main GAN training method
def train(self, pigment, epochs, batch_size=32, sample_interval=50,
    statistics=True):

    #Open statistics file
    ...

    # Load the dataset
    X_train = self.load_real_samples(pigment,dataset_path)
    m_train = X_train.shape[0]
    print("Number of training examples: m_train = " + str(m_train))

    # Adversarial ground truths
    valid = np.ones((batch_size, 1))
    fake = np.zeros((batch_size, 1))

    for epoch in range(epochs):
```

```
        # ---------------------
        #  Train Discriminator
        # ---------------------

        # Select a random batch of vectors
        idx = np.random.randint(0, X_train.shape[0], batch_size)
        spectra = X_train[idx]

        #Generate the z-latent dimension vector
        noise = np.random.normal(0, 1, (batch_size, self.latent_dim))

        # Generate a batch of new vectors
        gen_spectra = self.generator.predict(noise)

        # Train the discriminator
        d_loss_real = self.discriminator.train_on_batch(spectra, valid)
        d_loss_fake = self.discriminator.train_on_batch(gen_spectra,
            fake)
        d_loss = 0.5 * np.add(d_loss_real, d_loss_fake)
        accuracy = d_loss[1]

        # ---------------------
        #  Train Generator
        # ---------------------

        noise = np.random.normal(0, 1, (batch_size, self.latent_dim))

        # Train the generator (to have the discriminator label samples
            as valid)
        g_loss = self.combined.train_on_batch(noise, valid)

        #Write statistics file
        ...

        # If at save interval => save generated image samples
        if epoch % sample_interval == 0:
            # Saves losses and accuracies so they can be plotted after
```

```
                training
            self.losses.append((d_loss[0], g_loss))
            self.accuracies.append(100.0 * accuracy)
            self.epoch_checkpoints.append(epoch + 1)
            # Plot the progress
            print("%d [D loss: %f, acc.: %.2f%%] [G loss: %f]" % (epoch
                , d_loss[0], 100 * d_loss[1], g_loss))
            # Print images of the generated dataset
            self.sample_images(epoch, pigment, X_train)

    # Close output file
    ...
```

Listing 5.4. GAN training

The network architecture of both Generator and Discriminator is provided in listing 5.5. They are structured as two classical neural networks since the one dimensional input data does not require the complexity of other neural networks architectures like CNNs. The layers of the Discriminator are horizontally sequenced in decreasing sizes towards the output, as usual in classical neural networks (layer sizes: 1692 - 512 - 256 - 1). On the contrary, the layers of the Generator are positioned in increasing order (from z length to input length) since this network shall generate vectors of the same dimension of the input data (layer sizes: 100 - 256 - 512 - 1024 - 1692).

```
# GENERATOR
def build_generator(self):

    model = Sequential()

    model.add(Dense(256, input_dim=self.latent_dim))
    model.add(LeakyReLU(alpha=self.alpha))
    model.add(Dense(512))
    model.add(LeakyReLU(alpha=self.alpha))
    model.add(Dense(1024))
    model.add(LeakyReLU(alpha=self.alpha))
    model.add(Dense(self.vec_shape, activation='tanh'))
    model.summary()
```

```
    noise = Input(shape=(self.latent_dim,))
    vec = model(noise)

    return Model(noise, vec)

# DISCRIMINATOR
def build_discriminator(self):

    model = Sequential()

    model.add(InputLayer(input_shape=self.vec_shape))
    model.add(Dense(512))
    model.add(LeakyReLU(alpha=self.alpha))
    model.add(Dense(256))
    model.add(LeakyReLU(alpha=self.alpha))
    model.add(Dense(1, activation='sigmoid'))
    model.summary()

    vec = Input(shape=self.vec_shape)
    validity = model(vec)

        return Model(vec, validity)
```

Listing 5.5. Generator and Discriminator network architecture

After the first test, the subset D_1 results still too much small to be effective as a GAN input in augmenting our pigment dataset. Indeed, the problem of overfitting affects the Generative Network as any other Neural Network, and the artificial spectra dataset appears as not enough generalized. The figure 5.11 shows the generation process: the red spectra come from the pipeline (original and noise added spectra) while the blue spectra are the GAN generated ones. Some example spectra are retrieved from epoch 0,100,1000,6000. It is evident that GAN works well in creating new spectra, which look indistinguishable from the original ones after few iterations.

The second subset D_2 includes the augmented spectra of D_1 and the spectra corrected with the baseline removal algorithms. Therefore, the GAN will generalize better in the creation of new spectra, as it can be seen in Figure 5.12.

The third subset D_3 includes both the previous subsets and the spectra corrected with the noise removal algorithms. In this case, the GAN is supposed to generalize at best in the creation of artificial spectra, since it collects in its input training the widest range of cases in which an experimental Raman spectrum can be acquired. Figure 5.13 show the result of the process.

Once we find out that D3 subset shows the best performance in GANs data augmentation, the obtained spectra will be fed into three types of neural networks in order to verify both the best augmentation method and the best network architecture that are suited to solve Raman spectral classification tasks.

5.5 Pigment Classification

The last step of the present use case study consists in demonstrating the effectiveness of the aforementioned data augmentation techniques, and measuring the classification performances of three types of neural networks. At the end of the evaluation, the network architecture and the data augmentation approach which achieve the best performances will be chosen as a suitable classifier architecture that is able to solve Raman spectral classification tasks in real-time.

5.5.1 Neural Network Architectures

Since Raman data have one dimension, both classic (i.e. fully connected) neural networks and convolutional neural networks are analysed. The choice of the following architectures is justified by two crucial open questions in this use case: the first one concerns about the evaluation of using a binary classifier, i.e. a network that only discriminates if a certain spectrum belongs to a specified class or not, or a categorical classifier, i.e. a network that processes the input spectrum and outputs, with a measure of probability, the class to which a spectrum belongs. In the first case, the requirement is that every class (in our use case, every pigment) has its own binary classifier, and then, at testing time, the spectrum will be fed, inside a loop, into each of the single network. In the case of a sample multiple classification (e.g. a sample evaluated positively by two different classifiers), the class with the highest

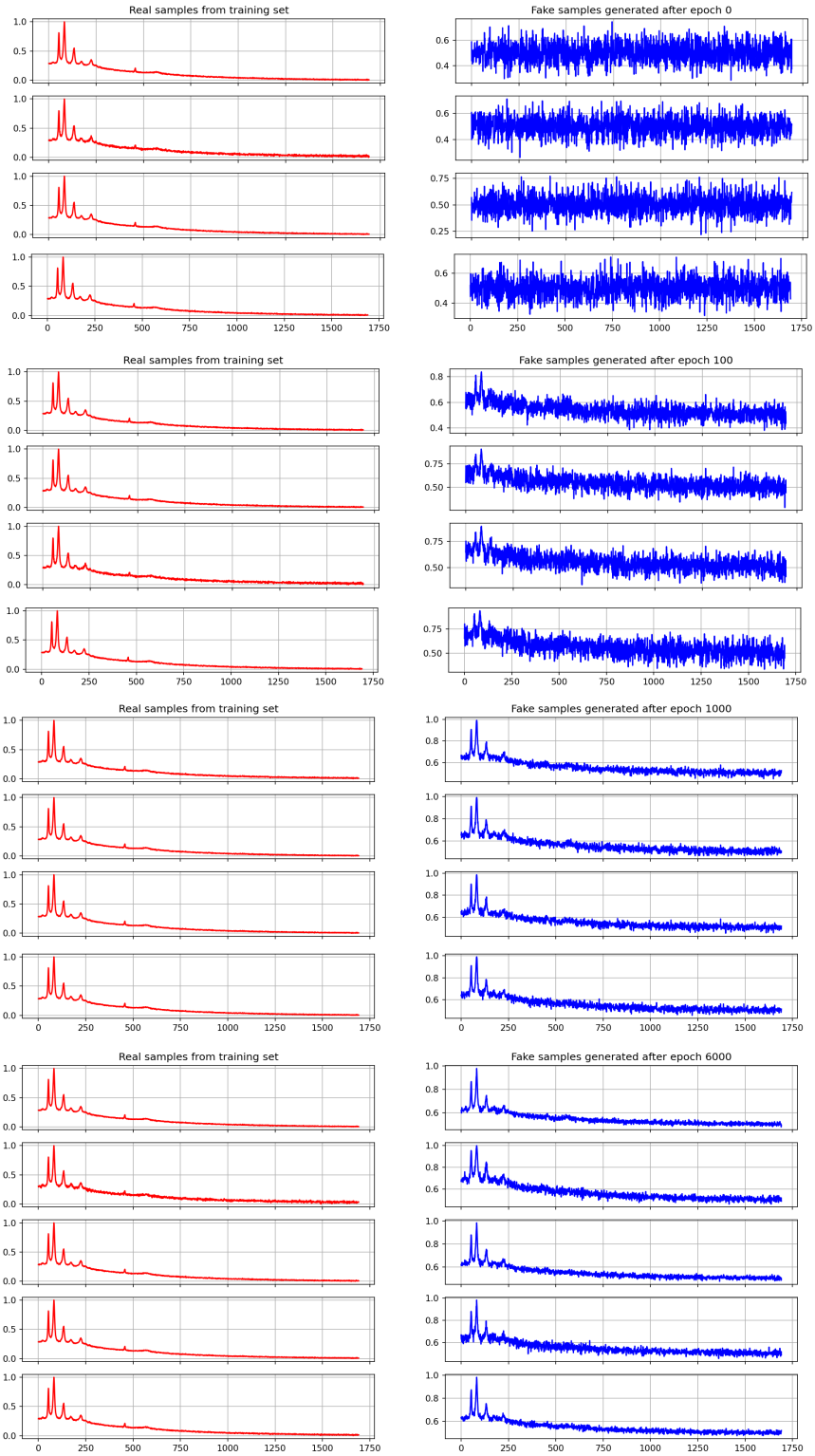


Figure 5.11. GAN generating spectra for D1 subset

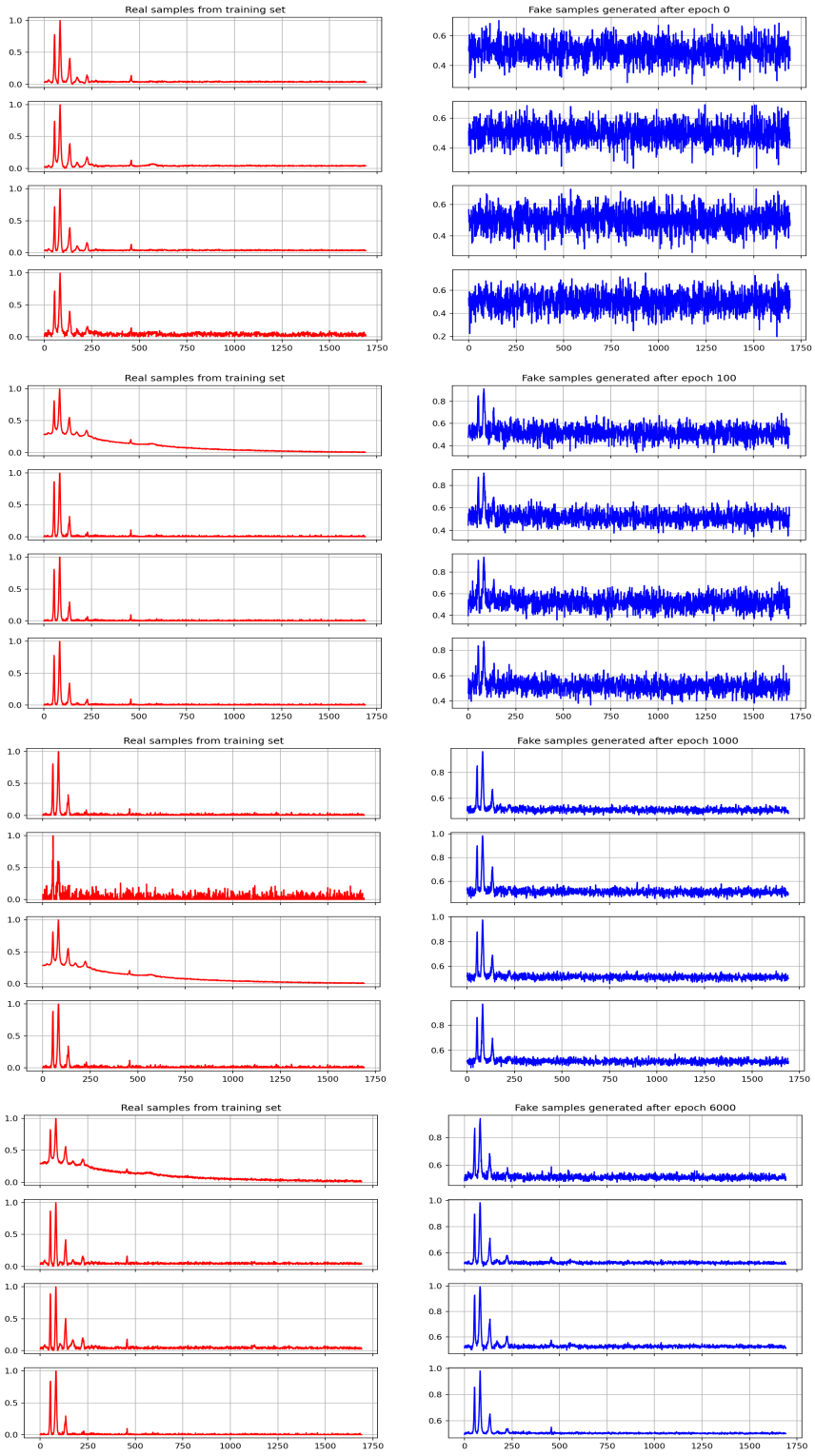


Figure 5.12. GAN generating spectra for D2 subset

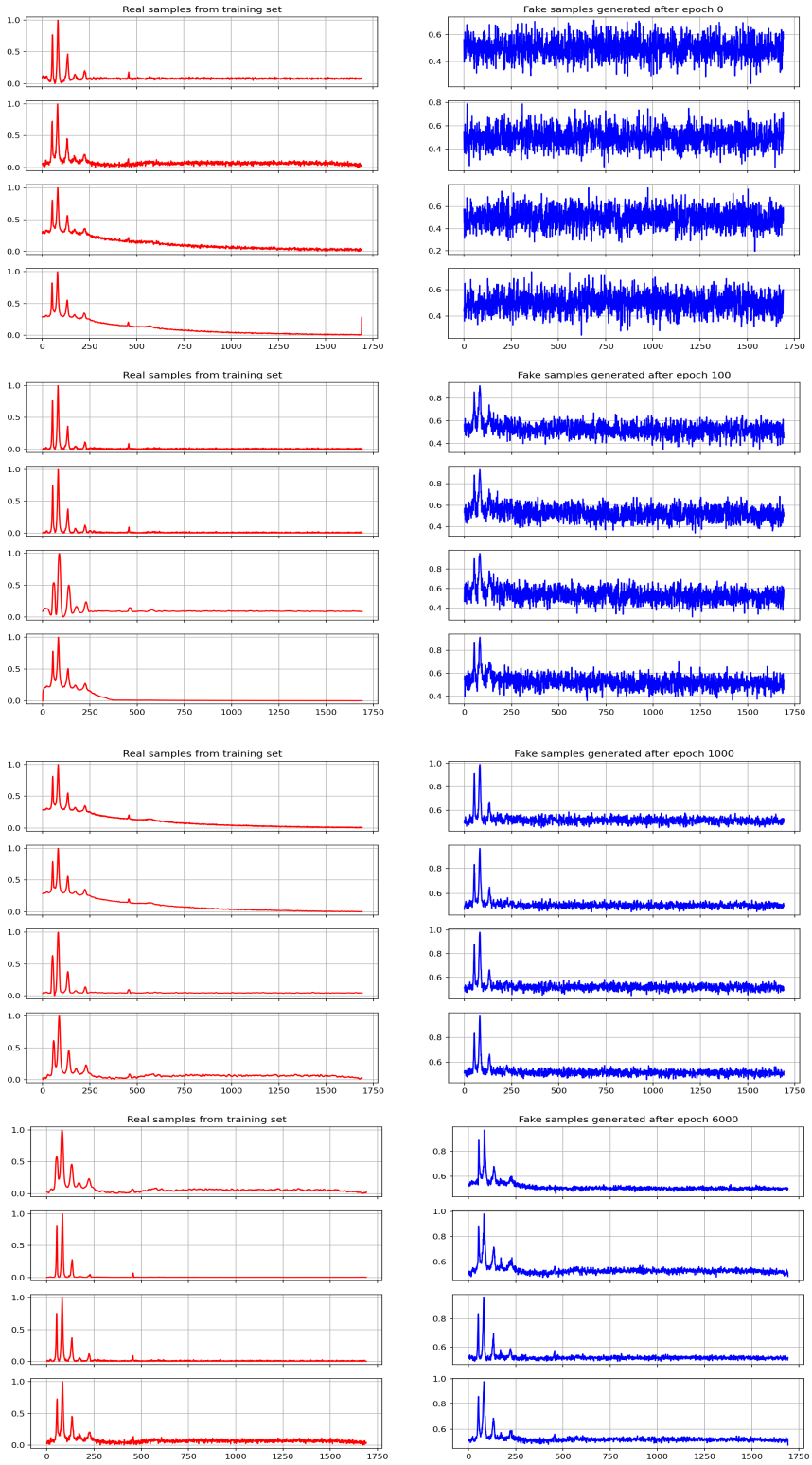


Figure 5.13. GAN generating spectra for D3 subset

value on the output neuron (z value fed into the activation function as explained in sec. 3.4) will be considered. In the second case there will be only one network that has been trained with all the pigments categories.
The second open question is whether the classification accuracy of one-dimensional data benefits from the complexity of the convolutional neural network (that has been widely proved regards 2D data), or whether the trend of the accuracy values remains stable in the comparison of the two approaches.

Binary Classifier The binary neural network is composed of an input layer that has the size of the input vector (1692), two hidden layers of size respectively 512 and 128, and the final output layer that produces a boolean output. A graphical representation is depicted in fig. 5.14. The loss function used in the training phase is the *Binary Cross-entropy loss*[2]. The programming code of the model of this network is provided in listing 5.6.

```
'''NN Model'''
def model(self, input_shape):
    # Define the input placeholder as a tensor with shape input_shape
    X_input = Input(input_shape)
    X = X_input
    initializer = TruncatedNormal()
    # First hidden layer of 512 nodes
    X = Dense(512, activation='relu', name='fc1')(X)
    # Second hidden layer of 128 nodes
    X = Dense(128, activation='relu', name='fc2')(X)
    # Softmax layer
    X = Dense(self.Y_train.shape[1], activation='sigmoid', name='fc3')(
        X)
    self.model = Model(inputs=X_input, outputs=X, name='PigmentiModel')
    return self.model
```

Listing 5.6. Binary Neural Network model

[2] $H_p(q) = -1/N \sum_{i=1}^{N} y_i * log(p(y_i)) + (1 - y_i) * log(1 - p(y_i))$

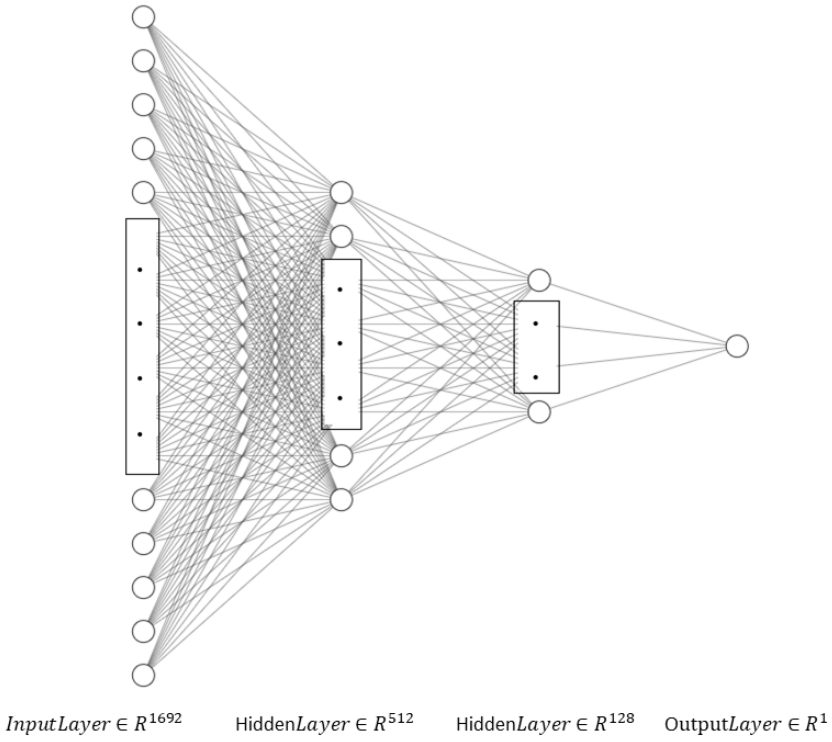


Figure 5.14. Binary Neural Network architecture

Categorical Classifier The categorical neural network reflects the model of the previous one except for the output layer: it is composed of an input layer of size 1692, two hidden layers of size respectively 512 and 128, and the output layer of size 39. This last value is related to the number of pigment classes plus the "zero" category, since the network will output a probability classification value for each one of the classes through the *Softmax* function[3]. A graphical representation is depicted in fig. 5.15, while the programming code is provided in listing 5.7.

```
'''NN Model'''
def model(self, input_shape):
    # Define the input placeholder as a tensor with shape input_shape
    X_input = Input(input_shape)
    X = X_input
    initializer = TruncatedNormal()
    # First hidden layer of 512 nodes
```

[3] $softmax(x)_i = \frac{exp(x_i)}{\sum_j exp(x_j))}$

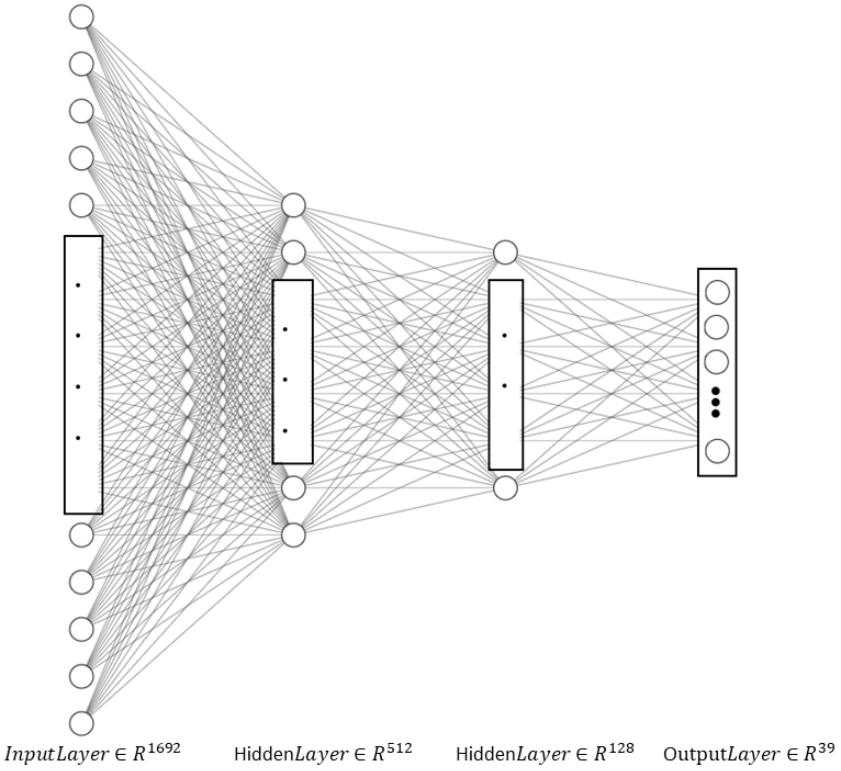


Figure 5.15. Categorical Neural Network architecture

```
X = Dense(512, activation='relu', name='fc1')(X)
# Second hidden layer of 128 nodes
X = Dense(128, activation='relu', name='fc2')(X)
# Softmax layer
X = Dense(self.Y_train.shape[1], activation='softmax', name='fc3')(
    X)
self.model = Model(inputs=X_input, outputs=X, name='PigmentiModel')
return self.model
```

Listing 5.7. Categorical Neural Network model

Convolutional Neural Network The third model consists in a CNN, whose architecture is based on the one used in [47], which is a variant of LeNet (CNN are explained in details in section 3.5). Several CNN architectures have been tested with pigment datasets before choosing one of them as representative of this class of classifiers. In fact, the different CNN networks have proved to produce very similar accuracy values for the same training set. The activation functions in the

convolutional layer is *LeakyReLU*[4], while the output value is obtained with the *Softmax* function. A graphical representation is depicted in fig. 5.16, while the programming code is provided in listing 5.8.

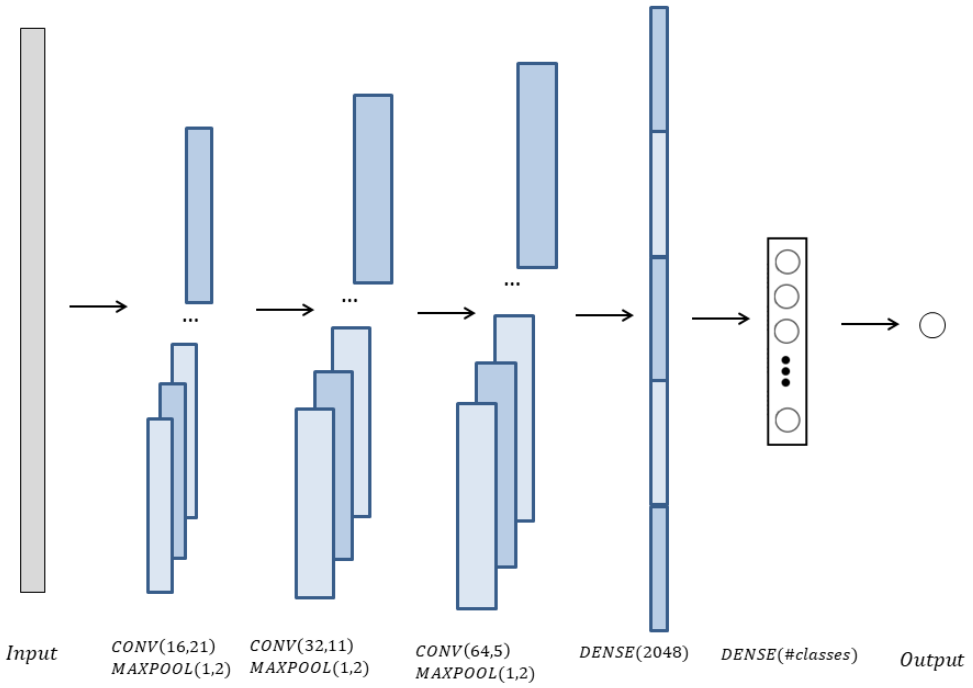


Figure 5.16. Convolutional Neural Network architecture

```
'''CNN Model'''
def set_model(self, input_shape):
    # Define the input placeholder as a tensor with shape input_shape
    X_input = Input(input_shape)
    X = X_input
    initializer = TruncatedNormal()
    # CONV -> BN -> LeakyRELU Block applied to X
    X = Conv1D(filters=21, kernel_size=16, name='conv1',
        kernel_initializer=initializer)(X)
    X = BatchNormalization(name='bn1')(X)
    X = LeakyReLU()(X)
    # MAXPOOL
    X = MaxPooling1D(pool_size=2, name='max1')(X)
```

[4] *f(x)= x if x>0 | 0.01x otherwise*

```
# CONV -> BN -> LeakyRELU Block applied to X
X = Conv1D(filters=11, kernel_size=32, name='conv2',
    kernel_initializer=initializer)(X)
X = BatchNormalization(name='bn2')(X)
X = LeakyReLU()(X)
# MAXPOOL
X = MaxPooling1D(pool_size=2, name='max2')(X)

# CONV -> BN -> LeakyRELU Block applied to X
X = Conv1D(filters=5, kernel_size=64, name='conv3',
    kernel_initializer=initializer)(X)
X = BatchNormalization(name='bn3')(X)
X = LeakyReLU()(X)
# MAXPOOL
X = MaxPooling1D(pool_size=2, name='max3')(X)

# FLATTEN X (means convert it to a vector) + FULLYCONNECTED
X = Dropout(0.5)(X)
X = Flatten()(X)
X = Dense(2048, activation='tanh', name='fc1')(X)
X = Dense(self.Y_train.shape[1], activation='softmax', name='fc2')(
    X)

self.model = Model(inputs=X_input, outputs=X, name='CNN_Pigmenti')
return self.model
```

Listing 5.8. Convolutional Neural Network model

5.5.2 Results

The aforementioned neural network architectures have been tested both with the standard pipeline generated data and with the further augmented GAN data. Each test has been repeated three times and the accuracy percentage intervals have been taken into account. The results for categorical-NN and CNN are shown in table 5.1. For the sake of clarity the binary-NN results are instead shown in figure 5.17, where the average of 3 executions for every pigment has been plotted.

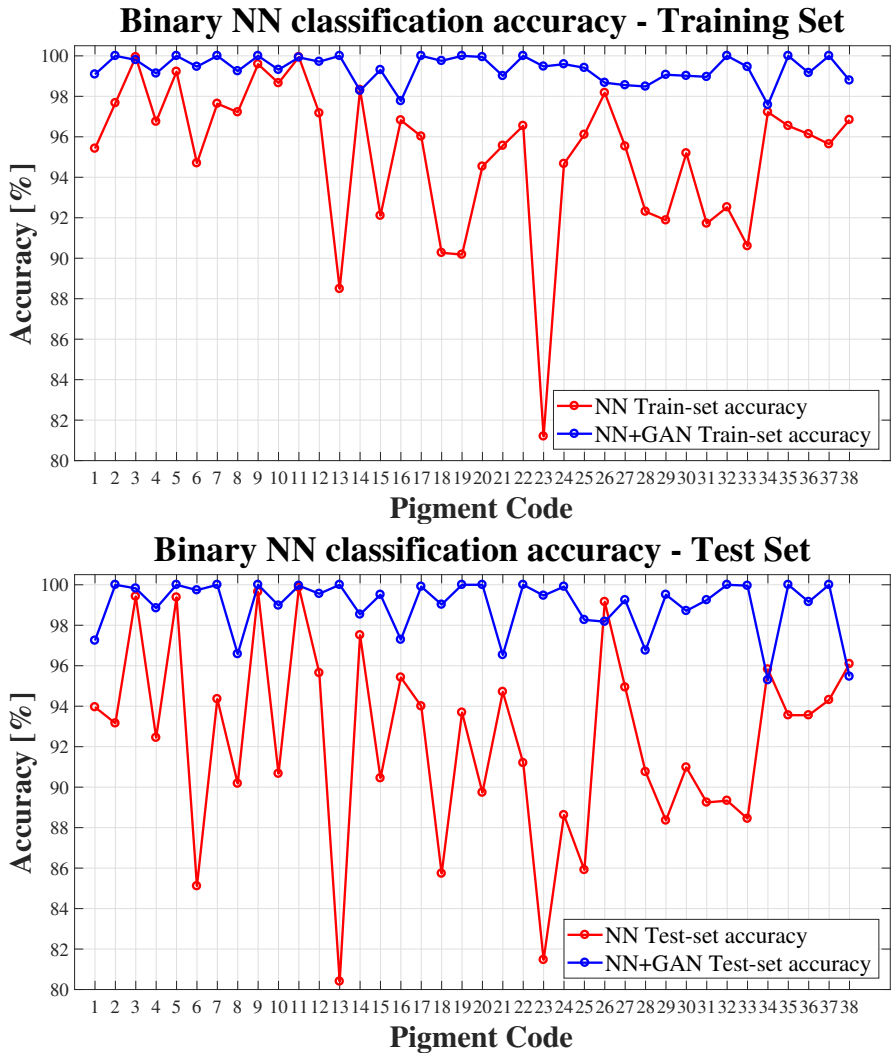


Figure 5.17. Binary NN classification accuracy

From the experimental results, the following statements can be clearly deduced:

- the total accuracy of the categorical-nn classifier is strictly related to the accuracy of the single pigments in the binary-nn classifier, therefore a possible choice between the two architectures will be influenced from other technical factors (e.g. resource consumption).
- The addition of the GAN generated spectra in the training dataset boosts the classification accuracy in all the three neural network architectures, therefore it will be crucial to exploit the advantage of combining the GAN approach with the augmentation pipeline.

NN Architecture	Training set Accuracy [%]	Test set Accuracy [%]
cat-NN	91.40 - 91.88	79.15 - 90.6
cat-NN with GAN	**97.53 - 99.14**	**96.12 - 97.78**
CNN	98.23-98.61	99.25-99.49
CNN with GAN	**99.33-99.64**	**99.82-99.98**

Table 5.1. Raman spectra classification results

- The Convolutional Neural Network architecture demonstrates its effectiveness even with the one dimensional data, increasing the accuracy both for the GAN and non-GAN approach. In particular, the best suitable classifier for Raman spectra classification tasks appears to be the Convolutional Neural Network with GAN-reinforced dataset, with a significant increase of the accuracy, exceeding the 99.98% threshold.

Chapter 6

Conclusions

This research focuses on spectral classification tasks, a field of study that embraces several sciences and disciplines and has many scientific and industrial applications. The different topics covered in the present research reflects the multi-disciplinary nature of spectral classification: the physics fundamentals of spectroscopy and optics that are necessary to understand the behaviour of the electromagnetic radiation and the information that a spectrum can bring within; the chemical models and phenomenons (such as fluorescence) that are needed to explain or predict the behaviours of the investigated compound and to consequently operate; the mathematical models and operations that belong to the field of signal processing to correctly estimate and process a signal; some electronic engineering concepts to properly work with electronic devices that compose a spectroscopic apparatus such CCDs; a cultural heritage knowledge (in the specific case, Renaissance frescoes) to operate in a non laboratory scenario and apply the proposed classification method. Above all, being Information Engineering the main subject of the present PhD thesis, the contribution of software analysis and development, with the addition of a brand new technology in the field of Artificial Intelligence (i.e. Generative Adversarial Networks) remains the core of the present research, which introduces an innovative approach in spectral classification using both signal processing and machine learning, demonstrating the effectiveness of the proposed technique.

Summary of the research project This project starts with the study of spectroscopy and its importance in many fields of applications. In particular, Raman spectroscopy has been chosen due to the interest of the scientific community around an automatic Raman spectra classification and its practical use in chemical compound identification. The characteristic features of a Raman spectrum have been studied in order to understand the unfeasibility of a classical spectral matching approach (that is mainly based upon vector correlation) to solve the classification task. The state of the art about Raman classification has been consulted with the conclusion that there is still no unique and validated method to solve the problem, i.e. methodologies change along with the data.

In the design of an innovative classification approach, the overcoming of the dualism between signal processing and machine learning techniques has been proposed, with the idea of merging the two concepts: since the neural networks have been proved as the actual most accurate classifiers in similar tasks, the signal processing algorithms can be used as data augmentation tools for each class of spectra to generate the large input dataset that a neural network needs. Thus, a real-time automatic spectra classification becomes possible without time-consuming data tuning steps.

In order to exploit the potentials of neural networks, an accurate analysis has been carried about the spectral features that can affect the classification accuracy in this kind of tools. Signal-to-noise ratio (SNR) is the main parameter that has been identified within this analysis. For this purpose, an innovative spectroscopic acquisition technique has been proposed and validated with the aim of optimizing the SNR and fulfilling, at the same time, the laser safety requests that most of the experimental Raman measurements must observe. Eventually, a solid data augmentation technique has been proposed to artificially generate the amount of spectra that a neural network needs to perform its classification task. A hierarchical pipeline with baseline and noise removal algorithms, and a subsequent GAN enhancement step have been developed with the notable result of generating thousands of spectra of the same labelled class from one single reference spectrum.

Finally, the whole method has been tested in a real use case study such pigment identification in Renaissance frescoes, where the aforementioned augmentation

pipeline has been used to generate the input set starting from a single reference spectrum per pigment. The dataset has been tested with three different neural network architectures: the Convolutional Neural Network with the GAN enhancement has shown the best accuracy results.

Innovative results The aforementioned pigment classification results demonstrate the correctness and the effectiveness of the present research and the proposed classification method. Apart from the specific use case, several innovative results can be reported from the research. First of all, in the state of the art literature, there is no mention about the use of baseline and noise filtering algorithms as data augmentation tools for spectral classification tasks. This intuition can be replicated in each spectroscopic measurement that exploits machine learning techniques but lacks of a sufficient input reference dataset. Concerning the optimization of SNR in a spectroscopic measurement, the results obtained through the experimental data of three commercial CCDs can be of great interest in scenarios where signal acquisition must adhere to laser safety directives. Further, at the time of the beginning of this research, there was no scientific paper that proposed the use of a brand new machine learning tool like GANs for Raman classification purpose[1].
One of the most important concept that must be highlighted is that the data augmentation approach that merges signal processing and machine learning is not only related to Raman spectroscopy, but it is potentially applicable with other spectroscopic techniques, especially in experimental applications where no sufficient reference sets are provided.

Future developments The next step of the present research involves the Edge Computing paradigm, i.e. a de-centralized approach that tends to bring computation and data storage closer to the location where it is needed. It is related to the *Internet of Things* concept and, for this specific case, concerns the design and development of a smart and portable Raman apparatus which provides a real-time chemical identification with a high degree of accuracy. Hence, the software developed in this project, from the hierarchical augmentation pipeline to the CNN classifier,

[1] At the time of writing the present thesis only a paper dated 2020 [108] has been found.

would run directly in a mini-computer inside the device, adhering to a standalone approach, instead of running in an external PC that must be connected to the Raman optical devices. This innovative device could be used by operators in many practical scenarios, whereas the introduction of an intuitive Graphical User Interface would make the device usable even without knowing the software algorithm specifics. Further, concerning the algorithm steps, additional improvements can be realized: for instance, the firs step of the hierarchical pipeline that consists in adding white noise to the reference spectrum could be replaced by a more sophisticated noise model that would be based on the specific CCD used in the apparatus, and the relative noise sources parameters which have been already analysed.

The countless applications in which a real-time spectral classifier can be exploited, makes a future standalone portable device very interesting both for scientific and industrial purposes.

.1 Appendix - List of Pigments

Dataset Name	Common Name	Color Ind.	Chemical Formula
Arancio di Marte	Yellow Iron Oxide	PY42	$Fe_2O_3 * nH_2O + Al_2O_3$
Azzurro Oltremare	Ultramarine Blue	PB29	$2Na_2Al_2Si_2O_6 + NaS_2$
Bianco S.Giovanni	Precipitated Chalk	PW18	$CaCO_3$
Blu Cobalto	Cobalt Blue	PB28	$CoAl_2O_4$
Blu Ercolano	Egyptian Blue	PB31	$CaCuSi_4O_{10}$
Blu Oltremare	Ultramarine Blue	PB29	$2Na_2Al_2Si_2O_6 + NaS_2$
Bruno di Marte	Hydroxide Brown	PBr6	Fe_20_3
Caput Mortuum	Violet Hematite	PBr7	Fe_20_3
Cinabro	Cinnabar	PR106	HgS
Giallo di Marte	Yellow Iron Oxide	PY42	$Fe_2O_3 * nH_2O + Al_2O_3$
Giallorino Scuro	Naples Yellow	PY41	$Pb_3(SbO_4)_2$
Lacca Robbia	Madder Lake	NR9	$C_{14}H_8O_4$
Morellone	Brown Iron Oxide	PBr7	Fe_20_3
Nero Avorio	Bone Black	PBk9	$[Ca_3(PO_4)_2] + CaCO_3 + C$
Nero Fumo	Lamp Black	PBk7	C
Nero di Marte	Mars Black	PBk11	Fe_30_4
Nero di Vite	Vine Black	PBk8	C
Ocra di Cipro	Brown Iron Oxide	PBr7	$Fe_2O_3 * nH_2O + + Al_2O_3MnO_2 + Si$
Ocra Gialla	Yellow Iron Oxide	PY42	$FeO(OH) * nH_2O$
Ocra Gialla Pallida	Natural Yellow Iron Oxide	PY43	$Fe_2O_3H_2O + CaCO_3 + + CaSO_4$
Ocra Oro	Hydroxide Brown	PBr6	$Fe(OH)_3$
Ocra Rossa	Natural Red Iron Oxide	PR102	$Fe_2O_3H_2O + CaCO_3 + + CaSO_4$
Ocra Scura	Natural Yellow	PY43	$Fe_2O_3H_2O + CaCO_3 +$

	Iron Oxide		$+CaSO_4$
Rosso Ercolano	Natural Red	PR102	$Fe_2O_3H_2O + CaCO_3+$
	Iron Oxide		$+CaSO_4$
Rosso Inglese	Synth. Red Ir.Ox.	PR101	Fe_2O_3
Rosso di Marte	Red Hematite	PR101:1	$Fe_2O_3 * nH_2O + Al_2O_3$
Rosso Pozzuoli	Natural Red	PR102	$Fe_2O_3H_2O + CaCO_3+$
	Iron Oxide		$+CaSO_4$
Sanguigna Naturale	Red Chalk	PR101:1	$Fe_2O_3 * nH_2O + Al_2O_3$
Terra Rossa	Natural Red	PR102	$Fe_2O_3H_2O + CaCO_3+$
	Iron Oxide		$+CaSO_4$
Terra di Siena	Brown Iron Oxide	PBr6	$Fe_2O_3 * nH_2O + MnO_2+$
Bruciata			$+Al_2O_3 * SiO_2$
Terra di Siena	Natural Yellow	PBr6	$Fe_2O_3 * nH_2O + MnO_2+$
Naturale	Iron Oxide		$+Al_2O_3 * SiO_2$
Terra d'Ombra	Brown Iron Oxide	PBr7	$Fe_2O_3 * MnO_2 + Si+$
Bruciata			$+Al_2O_3$
Terra d'Ombra	Brown Iron Oxide	PBr7	$Fe_2O_3 * MnO_2 + Si+$
Naturale			$+Al_2O_3$
Terra Verde	Green Earth	PG23	$(K[(Al, Fe^{III}), (Fe^{II}, Mg)]$
Antica			$(AlSi_3, Si_4)O_{10}(OH)_2)$
Terra Verde	Green Earth	PG23	$(K[(Al, Fe^{III}), (Fe^{II}, Mg)]$
Brentonico			$(AlSi_3, Si_4)O_{10}(OH)_2)$
Terra Verde	Green Earth	PG23	$(K[(Al, Fe^{III}), (Fe^{II}, Mg)]$
Calda			$(AlSi_3, Si_4)O_{10}(OH)_2)$
Verdaccio Scuro	Verdaccio		*mixture*

Table .1. List of pigment samples used for the Raman measurements

Bibliography

[1] Ian R Lewis and Howell Edwards. *Handbook of Raman spectroscopy: from the research laboratory to the process line*. CRC press, 2001.

[2] Elmar Schmälzlin, Benito Moralejo, Monika Rutowska, Ana Monreal-Ibero, Christer Sandin, Nicolae Tarcea, Jürgen Popp, and Martin M Roth. Raman imaging with a fiber-coupled multichannel spectrograph. *Sensors*, 14(11):21968–21980, 2014.

[3] Roberto Chirico, Salvatore Almaviva, Francesco Colao, Luca Fiorani, Marcello Nuvoli, Wenka Schweikert, Frank Schnürer, Luigi Cassioli, Silvana Grossi, Daniele Murra, et al. Proximal detection of traces of energetic materials with an eye-safe uv raman prototype developed for civil applications. *Sensors*, 16(1):8, 2016.

[4] Peter Larkin. *Infrared and Raman spectroscopy: principles and spectral interpretation*. Elsevier, 2017.

[5] Shamshad Ansari. *Building Computer Vision Applications Using Artificial Neural Networks*. Springer, 2020.

[6] Yongmei Ren, Jie Yang, Qingnian Zhang, and Zhiqiang Guo. Multi-feature fusion with convolutional neural network for ship classification in optical images. *Applied Sciences*, 9(20):4209, 2019.

[7] Federico Angelini and Francesco Colao. Optimization of laser wavelength, power and pulse duration for eye-safe raman spectroscopy. *Journal of the European Optical Society-Rapid Publications*, 15(1):1–8, 2019.

[8] Stefano di Frischia, Andrea Chiuri, Federico Angelini, and Francesco Colao. Optimization of signal-to-noise ratio in a ccd for spectroscopic applications. In *Proc. 15th European Workshop on Advanced Control and Diagnosis, ACD*, 2019.

[9] Jakub Langr and Vladimir Bok. *GANs in Action: Deep Learning with Generative Adversarial Networks*. Manning Publications, 2019.

[10] J Michael Hollas. *Modern spectroscopy*. John Wiley & Sons, 2004.

[11] Joseph R Lakowicz. *Principles of fluorescence spectroscopy*. Springer science & business media, 2013.

[12] Ron Jenkins et al. *X-ray fluorescence spectrometry*, volume 2. Wiley New York, 1999.

[13] David A Cremers and Leon J Radziemski. *Handbook of laser-induced breakdown spectroscopy*. John Wiley & Sons, 2013.

[14] Edmond De Hoffmann. Mass spectrometry. *Kirk-Othmer Encyclopedia of Chemical Technology*, 2000.

[15] Albert Ansmann, Maren Riebesell, and Claus Weitkamp. Measurement of atmospheric aerosol extinction profiles with a raman lidar. *Optics letters*, 15(13):746–748, 1990.

[16] Igor Nabiev, Igor Chourpa, and Michel Manfait. Applications of raman and surface-enhanced raman scattering spectroscopy in medicine. *Journal of Raman Spectroscopy*, 25(1):13–23, 1994.

[17] Fernando Rull, Sylvestre Maurice, Ian Hutchinson, Andoni Moral, Carlos Perez, Carlos Diaz, Maria Colombo, Tomas Belenguer, Guillermo Lopez-Reyes, Antonio Sansano, et al. The raman laser spectrometer for the exomars rover mission to mars. *Astrobiology*, 17(6-7):627–654, 2017.

[18] CT Chadwick, AH Willitsford, CR Philbrick, and HD Hallen. Deep ultraviolet raman spectroscopy: A resonance-absorption trade-off illustrated by diluted liquid benzene. *Journal of Applied Physics*, 118(24):243101, 2015.

[19] Dong Wei, Shuo Chen, and Quan Liu. Review of fluorescence suppression techniques in raman spectroscopy. *Applied Spectroscopy Reviews*, 50(5):387–406, 2015.

[20] Andrea Chiuri and Federico Angelini. Fast gating for raman spectroscopy. *Sensors*, 21(8):2579, 2021.

[21] Mónica Breitman, Sergio Ruiz-Moreno, and Alejandro López Gil. Experimental problems in raman spectroscopy applied to pigment identification in mixtures. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 68(4):1114–1119, 2007.

[22] Myles W Jackson et al. *Spectrum of belief: Joseph von Fraunhofer and the craft of precision optics*. Mit Press, 2000.

[23] Rajinder Singh. Cv raman and the discovery of the raman effect. *Physics in Perspective*, 4(4):399–420, 2002.

[24] Feride Severcan and Parvez I Haris. *Vibrational spectroscopy in diagnosis and screening*, volume 6. IOS Press, 2012.

[25] Luca Guerrini and Ramon A Alvarez-Puebla. Surface-enhanced raman spectroscopy in cancer diagnosis, prognosis and monitoring. *Cancers*, 11(6):748, 2019.

[26] Anthony T Tu and AT Tu. Raman spectroscopy in biology: principles and applications. *N/A*, 1982.

[27] Richard L McCreery. *Raman spectroscopy for chemical analysis*, volume 225. John Wiley & Sons, 2005.

[28] Fran Adar, Ruth Geiger, and Jon Noonan. Raman spectroscopy for process/quality control. *Applied spectroscopy reviews*, 32(1-2):45–101, 1997.

[29] Zbigniew Bielecki, Jacek Janucki, Adam Kawalec, Janusz Mikołajczyk, Norbert Pałka, Mateusz Pasternak, Tadeusz Pustelny, and Tadeusz Stacewicz. Sensors and systems for the detection of explosive devices-an overview. *Metrology and Measurement Systems*, 19(1):3–28, 2012.

[30] Lucia Burgio and Robin JH Clark. Library of ft-raman spectra of pigments, minerals, pigment media and varnishes, and supplement to existing library of raman spectra of pigments with visible excitation. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 57(7):1491–1521, 2001.

[31] G Burrafato, M Calabrese, A Cosentino, AM Gueli, SO Troja, and A Zuccarello. Coloraman project: Raman and fluorescence spectroscopy of oil, tempera and fresco paint pigments. *Journal of Raman Spectroscopy*, 35(10):879–886, 2004.

[32] Donald S Frankel. Pattern recognition of fourier transform infrared spectra of organic compounds. *Analytical Chemistry*, 56(6):1011–1014, 1984.

[33] Donald R Scott. Determination of chemical classes from mass spectra of toxic organic compounds by simca pattern recognition and information theory. *Analytical chemistry*, 58(4):881–890, 1986.

[34] UM Weigel and Rainer Herges. Automatic interpretation of infrared spectra: recognition of aromatic substitution patterns using neural networks. *Journal of chemical information and computer sciences*, 32(6):723–731, 1992.

[35] John E Simmonds, F Armstrong, and Philip J Copland. Species identification using wideband backscatter with neural network and discriminant analysis. *ICES Journal of Marine Science*, 53(2):189–195, 1996.

[36] LiMin Fu, Mark Yang, Raul Braylan, and Neal Benson. Real-time adaptive clustering of flow cytometric data. *Pattern recognition*, 26(2):365–373, 1993.

[37] Lynne Boddy, Malcolm F Wilkins, and Colin W Morris. Pattern recognition in flow cytometry. *Cytometry: The Journal of the International Society for Analytical Cytology*, 44(3):195–209, 2001.

[38] Peter Jander and Reinhard Noll. Automated detection of fingerprint traces of high explosives using ultraviolet raman spectroscopy. *Applied spectroscopy*, 63(5):559–563, 2009.

[39] Jin Hyuk Chung and Soo Gyeong Cho. Standoff raman spectroscopic detection of explosive molecules. *Bulletin of the Korean Chemical Society*, 34(6):1668–1672, 2013.

[40] Aron Hakonen, Per Ola Andersson, Michael Stenbæk Schmidt, Tomas Rindzevicius, and Mikael Käll. Explosive and chemical threat detection by surface-enhanced raman scattering: A review. *Analytica chimica acta*, 893:1–13, 2015.

[41] Jianhua Zhao, Harvey Lui, Sunil Kalia, and Haishan Zeng. Real-time raman spectroscopy for automatic in vivo skin cancer detection: an independent validation. *Analytical and bioanalytical chemistry*, 407(27):8373–8379, 2015.

[42] Susanne Pahlow, Susann Meisel, Dana Cialla-May, Karina Weber, Petra Rösch, and Jürgen Popp. Isolation and identification of bacteria by means of raman spectroscopy. *Advanced drug delivery reviews*, 89:105–120, 2015.

[43] Sanjay Gulia, Kamal K Gulati, Vijayeta Gambhir, Rinku Sharma, and MN Reddy. Trace detection of explosive and their derivatives in stand-off mode using time gated raman spectroscopy. *Vibrational Spectroscopy*, 87:207–214, 2016.

[44] Vincent Mazet, Cédric Carteret, David Brie, Jérôme Idier, and Bernard Humbert. Background removal from spectra by designing and minimising a non-quadratic cost function. *Chemometrics and intelligent laboratory systems*, 76(2):121–133, 2005.

[45] Brooke D Beier and Andrew J Berger. Method for automated background subtraction from raman spectra containing known contaminants. *Analyst*, 134(6):1198–1202, 2009.

[46] PJ Cadusch, MM Hlaing, SA Wade, SL McArthur, and PR Stoddart. Improved methods for fluorescence background subtraction from raman spectra. *Journal of Raman Spectroscopy*, 44(11):1587–1595, 2013.

[47] Jinchao Liu, Margarita Osadchy, Lorna Ashton, Michael Foster, Christopher J Solomon, and Stuart J Gibson. Deep convolutional neural networks for raman spectrum recognition: a unified solution. *Analyst*, 142(21):4067–4074, 2017.

[48] MA Kneen and HJ Annegarn. Algorithm for fitting xrf, sem and pixe x-ray spectra backgrounds. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 109:209–213, 1996.

[49] Siegfried Wartewig. *IR and Raman spectroscopy: fundamental processing*. John Wiley & Sons, 2006.

[50] Paul HC Eilers and Hans FM Boelens. Baseline correction with asymmetric least squares smoothing. *Leiden University Medical Centre Report*, 1(1):5, 2005.

[51] Zhi-Min Zhang, Shan Chen, and Yi-Zeng Liang. Baseline correction using adaptive iteratively reweighted penalized least squares. *Analyst*, 135(5):1138–1146, 2010.

[52] Sung-June Baek, Aaron Park, Young-Jin Ahn, and Jaebum Choo. Baseline correction using asymmetrically reweighted penalized least squares smoothing. *Analyst*, 140(1):250–257, 2015.

[53] Andrew T Weakley, Peter R Griffiths, and D Eric Aston. Automatic baseline subtraction of vibrational spectra using minima identification and discrimination via adaptive, least-squares thresholding. *Applied spectroscopy*, 66(5):519–529, 2012.

[54] Jianhua Zhao, Harvey Lui, David I McLean, and Haishan Zeng. Automated autofluorescence background subtraction algorithm for biomedical raman spectroscopy. *Applied spectroscopy*, 61(11):1225–1232, 2007.

[55] Chad A Lieber and Anita Mahadevan-Jansen. Automated method for subtraction of fluorescence from biological raman spectra. *Applied spectroscopy*, 57(11):1363–1367, 2003.

[56] Ronald W Schafer. What is a savitzky-golay filter?[lecture notes]. *IEEE Signal processing magazine*, 28(4):111–117, 2011.

[57] Jingdong Chen, Jacob Benesty, Yiteng Huang, and Simon Doclo. New insights into the noise reduction wiener filter. *IEEE Transactions on audio, speech, and language processing*, 14(4):1218–1234, 2006.

[58] H Georg Schulze, Rod B Foist, Kadek Okuda, André Ivanov, and Robin FB Turner. A small-window moving average-based fully automated baseline estimation method for raman spectra. *Applied spectroscopy*, 66(7):757–764, 2012.

[59] Miguel Colom and Antoni Buades. Analysis and extension of the percentile method, estimating a noise curve from a single image. *Image Processing On Line*, 3:332–359, 2013.

[60] Markus Lang, Haitao Guo, Jan E Odegard, C Sidney Burrus, and Raymond O Wells. Noise reduction using an undecimated discrete wavelet transform. *IEEE Signal Processing Letters*, 3(1):10–12, 1996.

[61] Jean-Luc Starck, Jalal Fadili, and Fionn Murtagh. The undecimated wavelet decomposition and its reconstruction. *IEEE transactions on image processing*, 16(2):297–309, 2007.

[62] Tom M Mitchell. Machine learning and data mining. *Communications of the ACM*, 42(11):30–36, 1999.

[63] Mohamad H Hassoun et al. *Fundamentals of artificial neural networks*. MIT press, 1995.

[64] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[65] Li Deng and Dong Yu. Deep learning: methods and applications. *Foundations and trends in signal processing*, 7(3–4):197–387, 2014.

[66] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[67] Austin Nevin, Iacopo Osticioli, Demetrios Anglos, Aviva Burnstock, Sharon Cather, and Emilio Castellucci. Raman spectra of proteinaceous materials used in paintings: a multivariate analytical approach for classification and identification. *Analytical chemistry*, 79(16):6143–6151, 2007.

[68] Juanjo González-Vidal, Rosanna Pérez-Pueyo, and María José Soneira. Automatic classification system of raman spectra applied to pigments analysis. *Journal of Raman Spectroscopy*, 47(12):1408–1414, 2016.

[69] Xiaqiong Fan, Wen Ming, Huitao Zeng, Zhimin Zhang, and Hongmei Lu. Deep learning-based component identification for the raman spectra of mixtures. *Analyst*, 144(5):1789–1798, 2019.

[70] Masashi Fukuhara, Kazuhiko Fujiwara, Yoshihiro Maruyama, and Hiroyasu Itoh. Feature visualization of raman spectrum analysis with deep convolutional neural network. *Analytica chimica acta*, 1087:11–19, 2019.

[71] Sascha T Ishikawa and Virginia C Gulick. An automated mineral classifier using raman spectra. *Computers & geosciences*, 54:259–268, 2013.

[72] Vasileios Sevetlidis and George Pavlidis. Effective raman spectra identification with tree-based methods. *Journal of Cultural Heritage*, 37:121–128, 2019.

[73] Jinchao Liu, Stuart J Gibson, James Mills, and Margarita Osadchy. Dynamic spectrum matching with one-shot learning. *Chemometrics and Intelligent Laboratory Systems*, 184:175–181, 2019.

[74] Takao Kuroda. *Essential principles of image sensors*. CRC press, 2017.

[75] Ralf Widenhorn, Morley M Blouke, Alexander Weber, Armin Rest, and Erik Bodegom. Temperature dependence of dark current in a ccd. In *Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications*

III, volume 4669, pages 193–201. International Society for Optics and Photonics, 2002.

[76] James R Janesick. *Scientific charge-coupled devices*, volume 83. SPIE press, 2001.

[77] David Dussault and Paul Hoess. Noise performance comparison of iccd with ccd and emccd cameras. In *Infrared Systems and Photoelectronic Technology*, volume 5563, pages 195–204. International Society for Optics and Photonics, 2004.

[78] IEC. Iec 60825-1 ed.2.0; safety of laser products - part 1:equipment classification and requirements, 2007.

[79] 2006/25/ec directive https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:114:0038:0059:en:PDF, 2006.

[80] RRUFF Project. Database of raman spectroscopy, x-ray diffraction and chemistry of minerals https://rruff.info/, 2021.

[81] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[82] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[83] Daniel Mas Montserrat, Qian Lin, Jan Allebach, and Edward J Delp. Training object detection and recognition cnn models using data augmentation. *Electronic Imaging*, 2017(10):27–36, 2017.

[84] Geoff Nitschke and Luke Taylor. Improving deep learning with generic data augmentation. *arXiv preprint arXiv:1708.06020*, 2018.

[85] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[86] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, pages 117–122. IEEE, 2018.

[87] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

[88] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[89] Jacopo Acquarelli, Twan van Laarhoven, Jan Gerretzen, Thanh N Tran, Lutgarde MC Buydens, and Elena Marchiori. Convolutional neural networks for vibrational spectroscopic data analysis. *Analytica chimica acta*, 954:22–31, 2017.

[90] Esben Jannik Bjerrum, Mads Glahder, and Thomas Skov. Data augmentation of spectral data for convolutional neural network (cnn) based deep chemometrics. *arXiv preprint arXiv:1710.01927*, 2017.

[91] Wooje Lee, Aufried TM Lenferink, Cees Otto, and Herman L Offerhaus. Classifying raman spectra of extracellular vesicles based on convolutional neural networks for prostate cancer detection. *Journal of raman spectroscopy*, 51(2):293–300, 2020.

[92] Rui Zhang, Huimin Xie, Shuning Cai, Yong Hu, Guo-kun Liu, Wenjing Hong, and Zhong-qun Tian. Transfer-learning-based raman spectra identification. *Journal of Raman Spectroscopy*, 51(1):176–186, 2020.

[93] Hong Wang, Yunchao Xie, Dawei Li, Heng Deng, Yunxin Zhao, Ming Xin, and Jian Lin. Rapid identification of x-ray diffraction spectra based on very limited data by interpretable convolutional neural networks. *arXiv preprint arXiv:1912.07750*, 2019.

[94] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[95] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[96] Hoo-Chang Shin, Neil A Tenenholtz, Jameson K Rogers, Christopher G Schwarz, Matthew L Senjem, Jeffrey L Gunter, Katherine P Andriole, and Mark Michalski. Medical image synthesis for data augmentation and anonymization using generative adversarial networks. In *International workshop on simulation and synthesis in medical imaging*, pages 1–11. Springer, 2018.

[97] Farzan Farnia and Asuman Ozdaglar. Gans may have no nash equilibria. *arXiv preprint arXiv:2002.09124*, 2020.

[98] Philippe Colomban. The on-site/remote raman analysis with mobile instruments: a review of drawbacks and success in cultural heritage studies and other associated fields. *Journal of Raman spectroscopy*, 43(11):1529–1535, 2012.

[99] Jeff Newman, Kui Chen, Marco Leona, and Tuan Vo-Dinh. Surface-enhanced raman scattering for identification of organic pigments and dyes in works of art and cultural heritage material. *Sensor Review*, 2007.

[100] Jean Dubessy, M-C Caumon, and Fernando Rull. *Raman spectroscopy applied to earth sciences and cultural heritage*, volume 12. The Mineralogical Society of Great Britain and Ireland, 2012.

[101] Marco Veneranda, Mireia Irazola, Africa Pitarch, Maitane Olivares, Ane Iturregui, Kepa Castro, and Juan Manuel Madariaga. In-situ and laboratory raman analysis in the field of cultural heritage: the case of a mural painting. *Journal of Raman Spectroscopy*, 45(3):228–237, 2014.

[102] Howell GM Edwards and Tasnim Munshi. Diagnostic raman spectroscopy for the forensic detection of biomaterials and the preservation of cultural heritage. *Analytical and bioanalytical chemistry*, 382(6):1398–1406, 2005.

[103] Alberto De Santis, Elisabetta Mattei, and Claudia Pelosi. Micro-raman and stratigraphic studies of the paintings on the 'cembalo'model musical instrument (ad 1650) and laser-induced degradation of the detected pigments. *Journal of Raman Spectroscopy: An International Journal for Original Work in all Aspects of Raman Spectroscopy, Including Higher Order Processes, and also Brillouin and Rayleigh Scattering*, 38(10):1368–1378, 2007.

[104] Silvia A Centeno. Identification of artistic materials in paintings and drawings by raman spectroscopy: some challenges and future outlook. *Journal of Raman Spectroscopy*, 47(1):9–15, 2016.

[105] Cennino Cennini, Carlo MILANESI, and Gaetano Milanesi. *Il Libro dell'arte, o Trattato della pittura... Di nuovo pubblicato, con molte correzioni e coll'aggiunta di più capitoli tratti dai codici fiorentini, per cura di Gaetano e Carlo Milanesi.* Le Monnier, 1859.

[106] Luisa Caneve, Flavia De Nicola, Roberta Fantoni, and Valeria Spizzichino. Spectral database of renaissance fresco pigments by libs, lif and colorimetry. In *Proceedings of the Third Balkan Symposium on Archaeometry. The Unknown Face of the Artwork*, pages 9–16. INOE, 2012.

[107] D Bersani and PP Lottici. Raman spectroscopy of minerals and mineral pigments in archaeometry. *Journal of Raman Spectroscopy*, 47(5):499–530, 2016.

[108] Shixiang Yu, Hanfei Li, Xin Li, Yu Vincent Fu, and Fanghua Liu. Classification of pathogens by raman spectroscopy combined with generative adversarial networks. *Science of The Total Environment*, 726:138477, 2020.

Acknowledgments

I wish to thank everyone that shared with me these intense three years of PhD course and helped me to achieve this important result. First of all, I wish to thank my thesis advisors: prof.ssa Elena De Santis who has been a valid and charismatic tutor and put me at ease in the university environment, and dr. Federico Angelini who motivated me to start this doctoral course and who has been a friend and a colleague with whom we shared many projects and experiences. Then, I wish to thank all the colleagues at ENEA Research Centre of Frascati with whom has been a pleasure and a fun to work with in five years: Andrea Chiuri, Massimiliano Guarneri, Valeria Spizzichino, Francesco Colao, Luisa Caneve, Marcello Nuvoli, Ivano Menicucci, Roberto Chirico, Sofia Ceccarelli, Marco Pistilli, Mario Ferri De Collibus, Massimo Francucci, Adriana Puiu, Federica Caso, all the other colleagues that I am forgetting, and Antonio Palucci and Roberta Fantoni who gave me the opportunity to work in such a nice workplace.

I wish also to thank all the colleagues, professors and students at the University of L'Aquila who have been of great help especially in writing papers and in giving me advices for the scientific conferences: dr. Paolo Giammatteo, prof. Luigi Pomante, prof.ssa Maria Domenica Di Benedetto, all the professors at DISIM and all my PhD course mates with whom I would have liked to spend more time if it hadn't been for the pandemic situation.

Last but not least, I dedicate this research to my love Valentina, who has been an incredible support for all these difficult years and an adventure companion for all the things that we have been through. I wish also to thank my mother and my father who have always been there when I needed to, and the friends who shared with me the moments of life of the last three years.