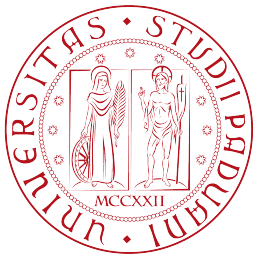


Università degli Studi di Padova
Department of Information Engineering
Master Degree in Telecommunication Engineering

Using Machine Learning to Turn Optical Fiber Specklegram Sensor into a Spatially Resolved Sensing System

Supervisor
Prof. Andrea Galtarossa

Tutor at UNICAN
Dr. Luis Rodríguez-Cobo

Author
Marco Fontana

Academic Year 2017/2018

To Danilo.

Abstract

Optical fibers present a remarkable set of characteristics that make them stand out for sensing applications. They allow geometric versatility, increased sensitivity over existing techniques, and inherent compatibility with fiber optic telecommunications technology. Furthermore, the sensors based on optical fibers represent a low cost and efficient solution for several applications, and they are particularly appreciated for their small size, robustness, and flexibility. These characteristics had permitted a strong growth of the importance of optical fiber sensors in the last decade.

Among sensors based on different phenomena and technologies, many authors have studied the relation existing between external perturbations and the speckle pattern (also called *specklegram*) produced in multimode optical fibers. A specklegram can be created by the propagated modes interfering within an optical fiber. Each point of the field distribution at the output end of the fiber is the sum of several contributions with aleatory phases and depends on the fiber stability. Thus, speckle patterns are sensitive to changes affecting the optical fiber, and they can be analyzed in order to detect different types of perturbations, such as vibrations, displacements, angular alignments, strain and even vital signs.

Fibre Specklegram Sensors (FSSs) has been proved as very sensitive systems that can be implemented using low-cost technologies. However, current state-of-the-art shows that there are almost no publications about the use of the specklegram in optical fiber as a spatially resolved sensing system. The main reason could be found in the very high complexity of this phenomenon, which is affected by a large number of parameters.

One of the primary methods for the detection of a perturbation using an FSSs is based on the comparison between the specklegrams in time. A modification of the configuration of the speckles means that the status of the fiber is changed, that is a perturbation occurred. The use of this approach does not permits to identify the parameters that have changed in the fiber configuration status. Therefore it is not possible to determine what kind of perturbation affected the fiber.

Furthermore, the FSSs implement distributed sensing, which means that all the points along the fiber contribute to determining the specklegram at the output end. Thus, it is not possible to detect the position along the fiber where the perturbation has occurred, at least for pieces of fiber longer than a few centimeters.

This work aims to find out whether there is a clear correlation between the distance at which the perturbation is generated in a fiber optic and the changes that occur in the outer speckle pattern. In this regard, we set up a system able to detect a perturbation occurring along the fiber and to recognize where it has taken place.

In particular, we realized two different experiments. In the first one, we generated and detected perturbations in three distinct positions along the fiber, i.e., at the beginning, at the end and in the middle of it. In the second one, we revised the same approach trying to detect perturbations occurred in ten positions. In both cases, we analyzed the specklegrams at the output end of the fiber using two different machine learning techniques.

The results show classifications over 99% of accuracy when testing new data under

certain conditions and the perturbations are occasioned in three different places, while the accuracy down to 71% when the perturbations are occasioned in ten different points.

While this work can be regarded as a proof of concept, it may afford to give rise to an attractive and promising research line, leading to developing a new set of sensors for various application scenarios.

Keywords: Fiber optics, Fiber optics sensors, Polymer optical fiber sensors, Spatial resolution sensing, Specklegram, Artificial Neural Networks, Deep Learing Algorithms.

Sommario

Le fibre ottiche presentano un insieme di caratteristiche tale da renderle una scelta eccellente per la realizzazione di diversi tipi di sensori. Esse, infatti, sono facilmente maneggiabili, dotate di una notevole sensibilità e sono intrinsecamente compatibili con la tecnologia esistente nel campo delle telecomunicazioni. Inoltre, i sensori realizzati in fibra ottica presentano costi di realizzazione contenuti e forniscono soluzioni efficienti per molte applicazioni. In aggiunta, sono apprezzati anche per le loro dimensioni contenute e per la loro robustezza e flessibilità. L'insieme di tutte queste caratteristiche ha permesso ai sensori in fibra ottica di accrescere notevolmente la loro importanza nell'ultimo decennio.

Tra i vari sensori, basati in generale su fenomeni e tecnologie differenti, molti autori hanno studiato la relazione che intercorre tra le perturbazioni esterne e lo *specklegram* all'uscita di una fibra multi-modo. Uno specklegram può essere creato dall'interferenza tra i modi propagati all'interno della fibra ottica. Ogni punto di cui è costituito il campo elettromagnetico all'uscita della fibra è il risultato della sovrapposizione di contribuiti caratterizzati da fase aleatoria e dipende dalla stabilità della fibra. Questa proprietà rende lo specklegram sensibile alle alterazioni subite dalla fibra e, tramite la sua analisi, è possibile identificare diversi tipi di perturbazioni, quali vibrazioni, spostamenti, allineamenti angolari, tensione ed anche parametri vitali.

I sensori specklegram (FSS, *Fibre Specklegram Sensor*) hanno dato prova di poter costituire sistemi molto sensibili ad un costo ridotto. Tuttavia, allo stato dell'arte non esistono pubblicazioni di rilievo riguardanti l'utilizzo dello specklegram come fondamento di sensori in grado di rilevare la posizione alla quale è avvenuta una perturbazione. La ragione principale può essere ricercata nella estrema complessità del fenomeno, il quale dipende da un numero di parametri piuttosto elevato.

Uno dei principali metodi per la rilevazione di una perturbazione mediante un FSS si basa sulla comparazione tra gli specklegram in istanti successivi di tempo. Una differente configurazione dello specklegram all'uscita della fibra segnala un cambiamento dello stato in cui si trova la fibra ottica, ovvero indica che è avvenuta una perturbazione. Questo approccio non permette di identificare quali parametri abbiano variato la configurazione della fibra, rendendo quindi difficile identificare il tipo di perturbazione subito da essa.

Inoltre, gli FSS realizzano una rilevazione della perturbazione in modo distribuito, ovvero tutti i punti lungo la fibra contribuiscono a determinare la configurazione dello specklegram che si otterrà all'uscita. Questo fatto implica l'impossibilità di rilevare la posizione lungo la fibra presso la quale è avvenuta la perturbazione, se non nel caso di frammenti di fibra lunghi solo pochi centimetri.

Lo scopo di questa tesi consiste nel ricercare una possibile correlazione tra la distanza alla quale la perturbazione è stata generata lungo la fibra ottica e le modificazioni avvenute nello *specklegram*. A questo proposito abbiamo disposto un sistema in grado di rilevare una perturbazione lungo la fibra e di riconoscere dove essa è avvenuta.

In particolare, abbiamo realizzato due esperimenti. Nel primo abbiamo generato e rilevato le perturbazioni in tre distinte posizioni, ovvero all'inizio, alla fine e nel mezzo della fibra. Nel secondo esperimento, abbiamo ripreso il medesimo approccio per rilevare

perturbazioni avvenute in dieci posizioni. In entrambi i casi, l'analisi degli *specklegram* all'uscita della fibra è stata effettuata tramite l'utilizzo di due diverse tecniche di *machine learning* (apprendimento automatico).

I risultati mostrano una classificazione la cui accuratezza supera il 99% nel primo esperimento (nel caso i dati analizzati soddisfino determinate condizioni), mentre l'accuratezza scende al 71% nel secondo esperimento.

Sebbene il risultato di questo lavoro possa essere considerato un prototipo volto ad implementare l'approccio innovativo appena esposto, esso può dare origine ad una linea di ricerca promettente ed attrattiva, portando allo sviluppo di un nuovo insieme di sensori applicabili in differenti scenari applicativi.

Contents

List of Figures

List of Tables

1

Introduction

This first chapter introduces the context of this thesis and its structure. Starting with a presentation about the optical fiber sensors and the speckle interferometry, we formulate the goal of this work. Then, in Section 1.3 we present the thesis structure.

1.1 Context

Over the last three decades, optical fiber sensors have attracted substantial attention and shown to be capable of quantifying a wide range of physical measurands such strain, vibration, temperature, rotation, acceleration, pressure, electric and magnetic fields, current, and so on. During this period, the growth of the optoelectronics and fiber optic communications industries led to a reduction of the components prices and a substantial quality improvement of these. The expansion of these industrial sectors permitted to fiber optic sensors not only to cover market segments where the traditional electronic sensor could not provide a solution, but also an to become competitive with the existing standard sensors. The advantages of optical fiber sensing include the ability to be lightweight, of very minimal size, passive, high bandwidth, low power and resistant to electromagnetic interference.

The core of optical sensing technology is the optical fiber, which structure is represented in Figure 1.1. It is composed of three main components: the core, the cladding, and the jacket. The discontinuity of the refractive index between the core and the cladding confines the electromagnetic field within the fiber, ensuring the transmission of light through the core with minimal loss. This effect is primarily achieved with a higher refractive index in the core relative to the cladding, causing a total internal reflection of light. The outer buffer jacket serves to protect the fiber from external conditions and physical damage.

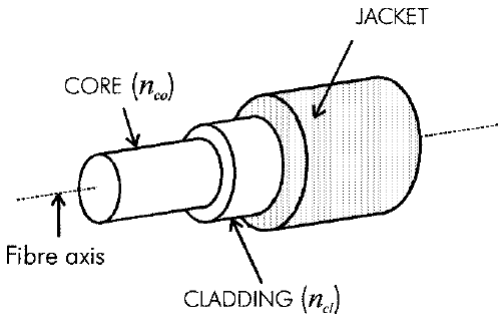


Figure 1.1: Schematic of an optical fiber.

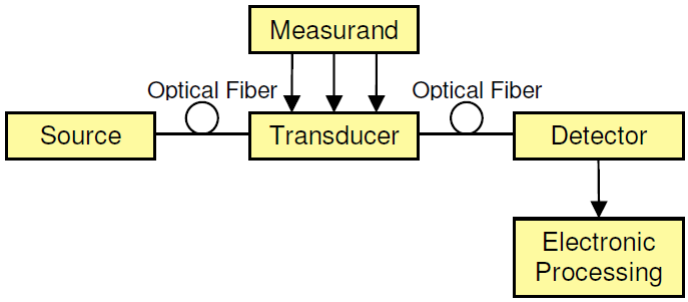


Figure 1.2: Basic components of an optical fiber sensor system.

1.1.1 Fiber-Optic Sensors

Fundamentally, a fiber-optic sensor works by modulating one or more properties of a propagating light wave, including intensity, phase, polarization, and frequency, in response to the environmental parameter being measured. The general structure of an optical fiber sensor system is shown in Figure 1.2. It consists of an optical source (Laser, LED, etc.), the optical fiber, a sensing or modulator element (which transduces the measurand to an optical signal), an optical detector and processing electronics (oscilloscope, optical spectrum analyzer, CCD sensor, etc.).

To date, a number of key optical sensors have been reported, and they are employed in many different application fields. Based on their topology and configuration, Fiber Optic Sensors (FOSs) can be further classified as:

Single-point The measurements are carried out at a single point, typically located at the end of the fiber.

Multipoint Consists of two or more sensing regions along the length of the fiber, where each region can detect the same or a different parameter.

Distributed Use the entire fiber length to sense one or more external parameters. This is a capability unique to fiber-optic sensors and one that cannot easily be achieved using conventional electrical sensing techniques.

The ability of multipoint and distributed sensors to simultaneously measure one or more parameters at different location provides important economic advantages. It allows a significant reduction of the cost of fiber usage and installation compared to using multiple single point sensors.

1.1.2 Fiber-Optic Specklegram Sensors

Another interesting categorization distinguishes between sensor using the temporal or the spatial content of the light waves. The vast majority of the researches and sensing solutions belongs to the first category and includes Bragg grating sensors, interferometric sensors, and intensity-based sensors. A far below number of studies have been done in exploiting the spatial content at the output of the fiber to investigate its sensing capabilities.

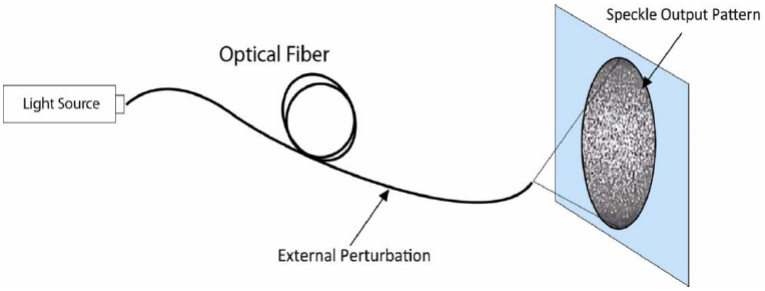


Figure 1.3: Schematic of the fiber optic specklegram sensor.

Fiber-Optic Specklegram Sensors (FSSs) belong to the second category and are called *spatially modulated sensors* or *modal modulated sensors*. They are based on a different intensity distribution of the output that results from inter-modal interference between all guided modes in the fiber. More specifically, each point of the field distribution (speckle) combines several contributions with different phases and depends on the fiber stability. When the optical fiber carrying the coherent light is perturbed, the distribution of the speckle intensities changes, with some speckle becoming brighter, some dimmer, and some not changing at all. The total intensity of the pattern remains unchanged, however. An analysis of the changes in the speckle pattern output from the optical fiber can be performed using a camera and image analysis methods.

In Figure 1.3 is represented a schematic of a fiber optic specklegram sensor. The sensing part of an FSSs consist of a multimode optical fiber and permits the propagation of a certain number of modes, depending on its geometrical characteristics and the material which is made of. In this context, polymer optical fibers (POF) can be an attractive solution, as they have many of the same advantages as conventional silica optical fibers for sensing application. In general, POFs provide a much lower cost alternative, although with higher transmission losses. Furthermore, POFs have some additional advantages, including robustness, ease of handling, high flexibility in bending and high sensitivity to strain. Polymers also have excellent compatibility with organic materials, giving them great potential for biomedical applications.

1.2 Goals

The goal of this thesis work is to demonstrate that the specklegrams at the output end of the fiber include the information relative to the position where the perturbation occurred along the fiber. Furthermore, this information can be extracted using machine learning techniques and used to implement a spatially resolved sensing system.

In order to prove this concept, we set up a system able to perturb the fiber in a series of predetermined points along the fiber. As the specklegram is highly sensitive to vibration and displacements of the fiber, the perturbation was produced by a mechanical arm able to repetitively 'touch' the fiber, maintaining it under the same conditions.

This system was used to produce a very high number of perturbations for each position along the fiber, in order to obtain a significant amount of specklegrams relative at each one of them. This data was employed to train the machine learning algorithms, for the purpose of building a network able to distinguish and categorize them relative to the

position where the fiber has been perturbed.

In particular, we implemented two different supervised machine learning techniques. The first one was based on an artificial neural network (ANN), consisted of three layers of neurons of various sizes. The second one was based on a convolutional neural network (CNN), that provided a structure composed of several intermediate layers between the input and the output ones.

Furthermore, we also collected a relevant set of data relative to perturbations induced manually on the fiber. The aim was to verify that the correct classification was due not just to the similarities introduced by the mechanical generation of the perturbation. Adding a certain amount of randomness, intrinsically provided by the human way to touch the fiber, we induced a series of perturbations different between them.

1.3 Thesis Outline

The thesis will initially provide a complete theoretical background regarding the specklegram phenomena and the state-of-the-art of the sensor based on it. In particular, we will report some examples of FSSs for vibration and strain detection, and the multiplexing techniques available nowadays.

The following chapter will focus on the methods used to analyze and categorize the specklegrams. Firstly, we will present the concept of neuron in an artificial neural network and the biological links that inspired this approach many years ago. Then, we will examine the two main approaches tested in this work.

In the Chapter 4 will be discussed the experimental setup, starting from the physical implementation of the system used to produce the perturbations along the fiber. Then, we will examine some more practical aspects, as the pre-process of the data and the configuration of the networks.

Finally, we will present the obtained result in Chapter 5, and we will draw our conclusions in Chapter 6.

2

Specklegram Sensing

In this chapter, we introduce the most important concepts related to the speckle sensing. Firstly, in Section 2.1.1, we will analyze the interferometry, that is the fundamental phenomena underlying the specklegram effect. Then, in Section 2.2, we will report the primary models and techniques that permit to relate the perturbations affecting the fiber with the output's behavior. Finally, in Section 2.3, we will outline some of the leading sensing applications of these models and other practical considerations.

2.1 Introduction

Among the most known optical metrology methodologies are those based on *speckle* patterns. If the light interacts with a surface whose roughness is comparable with its wavelength, a speckle distribution appears due to the interference of several waves with statistical phase distribution. In the early years of the laser as an instrumental tool, the speckle was considered as an undesired optical noise, especially in holographic applications. However, since the 70's, many experimentalists have used this phenomenon to determine a wide variety of physical parameters by interferometric measures.

On the other hand, speckle also appears when laser radiation is launched in a multimode optical fiber. The result of the interaction of several modes produce what is known as *modal noise* at the output end of the fiber, that is nothing but a complex speckle pattern. Obviously, this is an undesired effect in an optical communication system, but it contains information of the spatial state of the optical fiber, which can be used in metrological applications. A small perturbation of the multimode optical fiber will produce changes in the spatial distribution of the speckle pattern which can be detected in several ways.

The use of a CCD camera for speckle measurement was first proposed by Butters *et al.* and Macovski in the 70's. Their technique is known as Electronic Speckle-Pattern Interferometry (EPSI), and it has been used to obtain a real-time display of the vibration amplitude of an object. The actual use of EPSI for vibration analysis was initiated by Ek and Molin in 1971, and it has become common. Nowadays, the reference techniques have been developed by Spillman [1] and Yu [2].

2.1.1 Interferometry

Interferometry takes advantage of the phenomena of interference of light for measurement purposes. Interference is related to the wave nature of light and is the ability of two or more coherent waves to extinguish themselves. Light can be expressed as an

electromagnetic wave through the Maxwell equations. The combination leads to the well-known Helmholtz equation, here expressed for the electric field vector $\vec{\boldsymbol{E}}$:

$$\frac{\partial^2 \vec{\boldsymbol{E}}}{\partial \boldsymbol{r}^2} - \frac{\varepsilon}{c^2} \frac{\partial \vec{\boldsymbol{E}}}{\partial \boldsymbol{t}} = 0 \tag{2.1}$$

where ε is the dielectric constant, c is the speed of light and where we assume an electrical neutral matter with no free charges ($\rho = 0$, $\sigma = 0$). Since this equation is linear and Fourier synthesis permits us to generate any waveform, we are allowed to use the following exponential wave function as the main solution:

$$\vec{\boldsymbol{E}} = \vec{\boldsymbol{E}}_0 \exp[i(\varphi(\vec{\boldsymbol{r}}) - \omega t)] \tag{2.2}$$

where $\vec{\boldsymbol{E}}_0$ is the electrical field vector amplitude, $\varphi(\vec{\boldsymbol{r}})$ is a space-dependent phase factor and ωt is a time-dependent phase factor. The time-dependent term contains within ω the frequency of the wave ν ($\omega = 2\pi\nu$). All detectors react rather on the intensity of light, which is proportional to the squared amplitude of the electric vector:

$$\boldsymbol{I} \propto |\vec{\boldsymbol{E}}|^2 = \vec{\boldsymbol{E}}\vec{\boldsymbol{E}}^* \tag{2.3}$$

where the star * means complex conjugate.

The necessary condition for interference is the coherence, that can be expressed in the spatial or in the temporal domain (or both). Spatial coherence is concerned with the phase correlation of two different points across a waveform at a given instant of time. The magnitude of the spatial coherence is related to the apparent dimensions of the light source: the nearer the light source approaches a 'point source', the greater the spatial coherence of the emitted light. On the other end, temporal coherence describes the phase correlation of waves at a given point in space at two different instants of time. The temporal coherence of light is dependent upon the frequency bandwidth of the source: the more nearly the output of a source approximates monochromaticity, the more temporally coherent is the emitted light. In practice, a high degree of temporal and spatial coherence is achieved by using laser light. The high degree of temporal coherence arises from the laser's monochromaticity. The high degree of spatial coherence results since the wavefronts in a laser beam are in effect similar to those emanating from a very distant, single point source.

In the case of superposition of two (or more) coherent waves $\vec{\boldsymbol{E}}_1$ and $\vec{\boldsymbol{E}}_2$, the resulting electric filed vector $\vec{\boldsymbol{E}}_r$ is the sum of the contributing single waves:

$$\vec{\boldsymbol{E}}_r = \vec{\boldsymbol{E}}_1 + \vec{\boldsymbol{E}}_2 \tag{2.4}$$

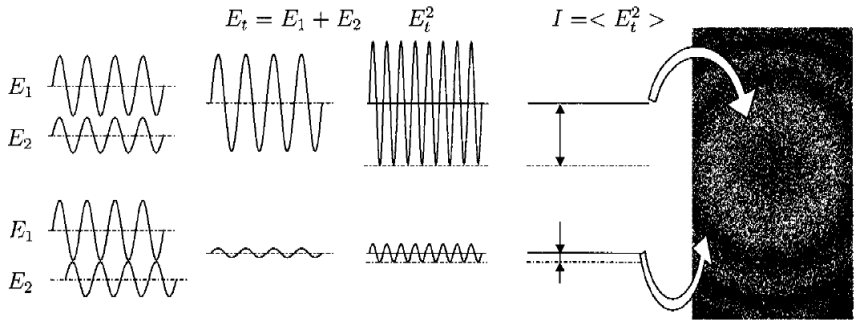


Figure 2.1: Constructive and destructive interferences.

and the time dependent terms in the intensity disappear:

$$\begin{aligned} \boldsymbol{I} \propto |\vec{\boldsymbol{E}}|^2 &= |\vec{\boldsymbol{E}}_1 + \vec{\boldsymbol{E}}_2|^2 && (2.5)\\ &= (\vec{\boldsymbol{E}}_1 + \vec{\boldsymbol{E}}_2)(\vec{\boldsymbol{E}}_1 + \vec{\boldsymbol{E}}_2)^* && (2.6)\\ &= \vec{\boldsymbol{E}}_1\vec{\boldsymbol{E}}_1^* + \vec{\boldsymbol{E}}_2\vec{\boldsymbol{E}}_2^* + \vec{\boldsymbol{E}}_1\vec{\boldsymbol{E}}_2^* + \vec{\boldsymbol{E}}_1^*\vec{\boldsymbol{E}}_2 && (2.7)\\ &= I_1 + I_2 + 2\sqrt{I_1 I_2}\cos[\varphi_1(\vec{\boldsymbol{r}}) - \varphi_2(\vec{\boldsymbol{r}})] && (2.8)\\ &= I_{av}(1 + m\cos(\Delta\varphi)) && (2.9) \end{aligned}$$

with

$$\begin{aligned} I_{av} &= I_1 + I_2 && (2.10)\\ \Delta\varphi &= \varphi_1(\vec{\boldsymbol{r}}) - \varphi_2(\vec{\boldsymbol{r}}) && (2.11)\\ m &= \frac{2\sqrt{I_1 I_2}}{I_1 + I_2} = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} && (2.12) \end{aligned}$$

I_{av} is called background intensity, m is the contrast or modulation, and $\Delta\varphi$ is the phase difference.

When two waves are *in phase*, that is $\Delta\varphi \equiv 0 \pmod{2\pi}$, the interference is said to be *constructive* and the resulting intensity is maximal. When two waves are "out of phase", that is $\Delta\varphi \equiv \pi$, the interference is said to be *destructive*, and the resulting intensity is minimal. An example of interferometric fringe pattern is reported in Figure 2.1.

The two waves propagate along two different paths until they are combined again causing interference. The phase difference can be written as:

$$\Delta\varphi = 2\pi\nu\Delta t = \frac{2\pi\Delta L}{\lambda} \tag{2.13}$$

where Δt is a time delay and ΔL is an optical path difference. So, the phase difference between the two waves is directly related to a time delay or an optical path difference between the two waves.

2.1.2 Propagation in a Multimode Fiber

Mathematically, according to wave theory, a mode is an allowable electromagnetic field that fulfills Maxwell's equations and boundary conditions. The optical fiber is a dielectric

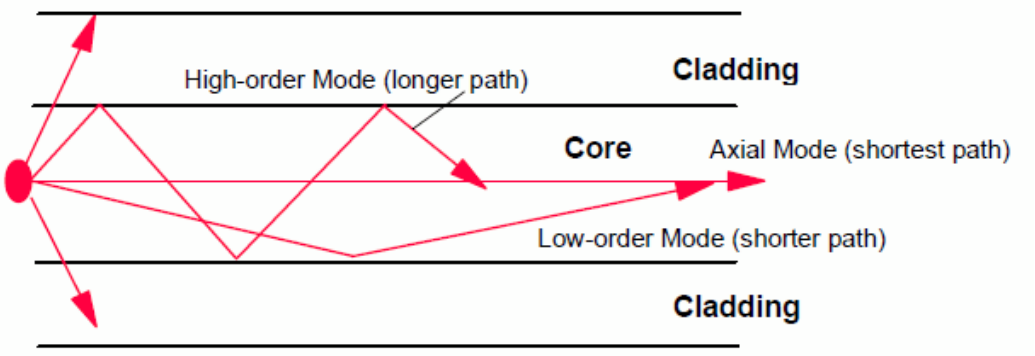


Figure 2.2: Ray propagation in a multimode fiber.

waveguide whereby the light can be transmitted by the process of total internal reflection. An optical fiber can support the propagation of one or more modes, according to its dimensions, the material which is made of, and the frequency of the light through it. According to Maxwell's equations, modes traveling in an optical fiber are transverse.

Assuming the dimensions and the imperfections of the optical fiber much greater then the light's wavelength, we can model the propagation of the modes using the ray theory. From this point of view, modes are simply the several paths that light might travel in a fiber. The fundamental mode travels along the axis of the optical fiber. The higher-order modes have a low incident angle, so light travels a greater distance than the low-angle rays to reach the end of the fiber. A schematic representation of these concepts is depicted in Figure 2.2. Based on the ray model, each mode has a different phase velocity caused by different optical path and each mode has a different spatial position at the output of the fiber.

The number of modes capable of propagating in a fiber can be computed defining the Normalized frequency (V), expressed as

$$V = \frac{2\pi a}{\lambda}(n_1 - n_2)^{\frac{1}{2}} \tag{2.14}$$

where n_1 and n_2 are, respectively, the refractive indices of the core and the cladding, a is the core diameter, and λ is the wavelength of light.

The Numerical Aperture (NA) of an optical fiber is defined as $NA = (n_1 - n_2)^{\frac{1}{2}}$, hence equation (2.14) can be rewritten as

$$V = \frac{2\pi a}{\lambda}NA \tag{2.15}$$

Using the normalized frequency, for a step-index fiber, the number of modes M can be approximated by

$$M = \frac{V^2}{2} \tag{2.16}$$

and, for a graded-index fiber

$$M = \frac{V^2}{4} \tag{2.17}$$

The speckle pattern is formed by interference between different propagation modes. The amount of speckles is approximately equal to the number of modes M, which is

given in equations reported above.

As seen from the equations (2.15), (2.16), and (2.17), the fiber with a large core diameter and a high NA shows a greater number of speckles. Furthermore, the increase of the wavelength of light reduces the number of speckles. Finally, a step-index fiber exhibit more speckles than a graded-index fiber, because from the equations (2.16) and (2.17), half of the modes supported by the step-index fiber cannot be propagated in the graded-index fiber.

2.2 Speckle Phenomena and Analysis Techniques

In this section, we will report the main aspects that affect the speckle phenomena and the most important analysis techniques. It is important to note that some of these methods primarily use the spatial intensity content of the specklegram, while others use the complex amplitude sensing. Despite the complex amplitude approach gives better result regarding sensitivity, our analysis will be set up only using the spatial intensity content, as it permits a simple and low-cost setup able to demonstrate the feasibility of our technique.

2.2.1 Specklegram Sensors

When the light is launched into an optical fiber, a certain number of modes will be propagated and will reach the output end of the fiber. These modes will overlap and interfere in a quite complex way, producing a speckle pattern (also called *specklegram*) at the output of the fiber. This is a well-known phenomenon in optical communication systems and is known as *modal noise*. It is an undesired effect in data transmission but, from the sensing point of view, it provides the spatial information of the fiber status. When a perturbation occurs along the fiber, the speckle pattern changes in a tricky way, expressing the new state assumed by the optical fiber. A hologram forms at the output of the sensor with the speckle field, and the fiber status information can be recorded in a hologram. The sensors based on the analysis of the speckle patterns are called Fiber-Optic Specklegram Sensors (FSSs).

FSSs are among the sensors which are based on exploiting the spatial content. Their operating principle is based on intensity distributions at the sensor output, which result from inter-modal interference between all guided modes in the fiber. Spatial information can be retrieved by the analysis of change of speckle patterns distribution using image processing techniques with statistical features. Due to the use of statistical properties of speckle patterns at the sensor analysis, these sensors were also named as fiber-optic Statistical Mode Sensors in some researches of the literature [1].

FSS can be divided into two groups as holographic and non-holographic specklegram sensors. In holographic techniques for the use of a complex amplitude distribution, a hologram is exploited to capture a reference wavefront for the unperturbed fiber status. The hologram is utilized as a filter to detect the wavefront variation due to fiber status change.

In non-holographic techniques, the intensity content of the speckle field can be measured by detecting the change of statistical properties between perturbed speckle in-

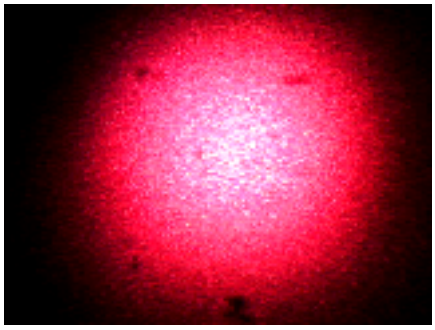


Figure 2.3: Photograph of whole speckle pattern captured by a CCD camera.

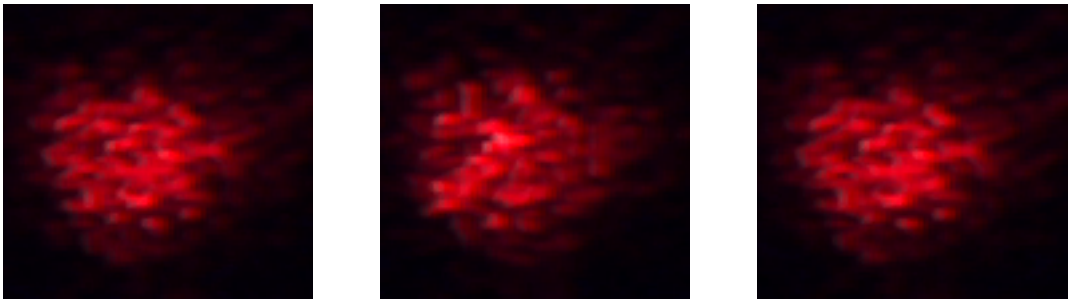
tensity pattern and unperturbed speckle intensity pattern. Holographic FSSs are more sensitive than non-holographic FSSs due to the inclusion of phase information.

When the output from a multimode optical fiber is projected on a screen, a uniform circular pattern is observed at the output end of the fiber. The characteristics of the specklegram, like dimension and quantity of the speckles, are related to the number of modes propagated, as reported in the previous section. Multimode fibers are capable of guiding many modes which have different phase velocities. Propagated modes interfere depending on the source coherence.

The sensitivity of specklegram sensors is affected by properties affected by each element of the sensing system: the optical source, the optical fiber, and the detector. The first one determines the characteristics of the transmitted light, as the degree of coherence and monochromacy, in addition to the power and the frequency of the transmitted light. The optical fiber is the sensitive part of the system, which means that its physical characteristics will determine the ways of interaction with the environment. Different materials can provide distinct degrees of flexibility and different responses to stresses. Finally, the detector has to deal with the majority of the parameters, from the contrast ratio of the received signal to the maximum number of samples per unit of time, if the detection is realized in the digital domain.

The photograph of a whole speckle pattern captured by a CCD camera at the output of a specklegram sensor, whose light source is a laser, is shown in Figure 2.3. If the optical source is coherent, the pattern becomes granular containing host of speckles of changing intensities. Alternatively, if incoherent optical source is used, a smooth distribution of intensity takes place within the pattern. More sensitive sensors were accomplished with more coherent light sources.

Nowadays, CCD cameras are usually used as the detector for the fiber-optic specklegram sensors, because they provide an excellent trade-off between sensitivity and cost. It is possible to record the change of speckles over time and analyze them utilizing image processing techniques. The picture of this speckle pattern is presented in Figure 2.4, where (a), (b), and (c) are taken before, during, and after the perturbation, respectively. The image was obtained using a laser source at 638 nm wavelength as the optical source and a graded-index multi-mode fiber (about 6 m length) with core/cladding dimension of 50/490 μm. As we can see from the Figure 2.4, after the perturbation the fiber returns to the original state, leading to the speckle pattern obtained before the perturbation. The distribution of speckles varies slowly over time; however, the intensity of the total pattern remains constant. Then, it is possible to establish a reference specklegram related



(a) (b) (c)

Figure 2.4: The photographs of local speckle pattern (a) before, (b) during and (c) after a perturbation captured by a CCD camera.

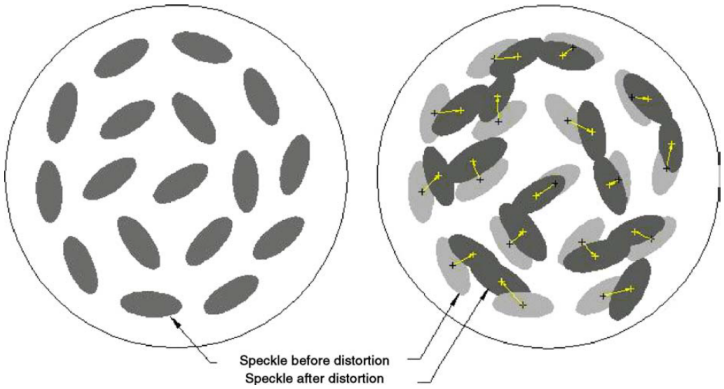


Figure 2.5: The graphical representation of the speckle pattern before and during the perturbation due to mode redistribution.

to the unperturbed fiber and use it to detect pattern changes due to mode redistribution associated to various types of perturbation (stress, temperature, etc.).

Graphical representation of the speckle pattern before and after the perturbation is illustrated in Figure 2.5. Perturbation causes a change in the refractive index of the fiber, as well as a tiny variation in length. The first effect produces small phase changes that affect the speckle pattern. Moreover, the slight variation in length alters the phase changes in the multi-mode fiber. The combination of the two effects induces changes in the distribution of the speckles, resulting in intensity variations in each speckle. Using some of the approaches that will be presented in the next sections, these variations can be analyzed to relate the change in the speckle pattern with the characteristics of the perturbation occurred.

2.2.2 Image Difference Approach

For the analysis of FSS, the mathematical models for speckle intensity variations have been developed in [1] and [2]. Even though it is difficult to predict the speckle distribution, an approximate relationship between the perturbing factor and the speckle intensity distribution can be developed for which the perturbation of the fiber can be determined. Assuming that coherent light is used in a multimode optical fiber, the theory of the operating principle of optical fiber specklegram sensors can be obtained from

the spatial intensity change.

If I_k is the intensity of the k^{th} speckle, the total intensity is given by

$$I = \sum_{k=1}^{N} I_k = \text{constant} \tag{2.18}$$

where N is the total number of pixels present.

Now, let the initial fiber status be S_0 assuming that all the modes are equally exited. The output light field at a given position of the fiber speckle filed can be considered as the contribution from all the modes, for which the complex phasor of the mth mode at point Q in status S_0 can be written as

$$u_{0m}(x, y) = a_{0m}(x, y) \exp[j\phi_{0m}(x, y)] \tag{2.19}$$

where (x, y) is the spatial coordinate of the observation plane that is perpendicular to the optical axis of the fiber. Because the wave filed at point Q is the summatiion caused by M modes of the fiber, it can be written as

$$U_0(x, y) = \sum_{m=1}^{M} u_{0m}(x, y) = \sum_{m=1}^{M} a_{0m}(x, y) \exp[j\phi_{0m}(x, y)] \tag{2.20}$$

where M is the total number of modes. When the status of the fiber changes as a result of perturbation, the speckle field emerging from the fiber will also be changed. Suppose that a new fiber status S is established (as a result of perturbation); then the mth mode complex phasor becomes

$$u_m(x, y) = [a_{0m}(x, y) + \Delta a_m] \exp\{j[\phi_{0m}(x, y) + \Delta\phi_m]\} \tag{2.21}$$

The wave filed at point Q would be

$$U(x, y) = \sum_{m=1}^{M} u_m(x, y) = \sum_{m=1}^{M} [a_{0m}(x, y) + \Delta a_m] \exp\{j[\phi_{0m}(x, y) + \Delta\phi_m]\} \tag{2.22}$$

where Δa_m and $\Delta\phi_m$ are the corresponding amplitude and phase variations, respectively, caused by the fiber-status changes. Defining the intensity as $I(x, y) = |U(x, y)|^2$, the intensity deviation between status S_0 and status S at Q can be written as

$$\Delta I(x, y) = I_0(x, y) - I(x, y) = |U_0(x, y)|^2 - |U(x, y)|^2 \tag{2.23}$$

where I_0 and I are the intensities at point Q caused by the initial and the status changes of the fiber, respectively. For simplicity, we shall drop the (x, y) coordinate from our expressions. By substituting Equation (2.20) and (2.22) into Equation (2.23), we have

$$\begin{aligned} \Delta I = &\sum_{m=1}^{M-1} \sum_{n=m+1}^{M} a_{0m} a_{0n} \exp[j(\phi_{0m} - \phi_n)] \\ &- \sum_{m=1}^{M-1} \sum_{n=m+1}^{M} (a_{0m} + \Delta a_m)(a_{0n} + \Delta a_n) \exp[j(\phi_{0m} - \phi_{0n} + \Delta\phi_m - \Delta\phi_n)] \end{aligned} \tag{2.24}$$

Because the phase variation dominates the speckle filed variation compared with the amplitude variation (assuming a lossless waveguide), Equation (2.24) can be written as

$$\Delta I = \sum_{m=1}^{M-1} \sum_{n=m+1}^{M} 4a_{0m}0n \sin[\phi_{0m} - \phi_{0n} + (\Delta\phi_m - \Delta\phi_n)/2] \times \sin[(\Delta\phi_n - \Delta\phi_m)/2] \quad (2.25)$$

in which $\Delta a_m \approx 0$. By integrating over the speckle field, we can write the MSV as

$$\begin{aligned} MSV &= \iint \Big|\Delta I\Big| dxdx \\ &= \iint \Big| \sum_{m=1}^{M-1} \sum_{n=m+1}^{M} 4a_{0m}a_{0n} \times \sin[\phi_{0m} - \phi_{0n} + (\Delta\phi_m - \Delta\phi_n)/2] \\ &\quad \times \sin[(\Delta\phi_m - \Delta\phi_n)/2] \Big| dxdy \end{aligned} \quad (2.26)$$

For simplicity of notation we let

$$\psi_{mn} = \phi_m - \phi_n \qquad \Delta\psi_{nm} = \Delta\phi_n - \Delta\phi_m \quad (2.27)$$

in which we see that

$$kL \le \psi_{mn}(x, y) \le \psi_{0M}(x, y) \qquad k(\Delta L) \le \Delta\psi_{mn} \le \Delta\psi_{0M} \quad (2.28)$$

where L is the length of the fiber and k is the wave number of the light source.

Now we assume that the fiber is perturbed by a strain $\Delta L/L$; then we have

$$\Delta\psi_{nm} = \Delta\phi_n - \Delta\phi_m = (n - m)\Delta\psi_{0M}/M \quad (2.29)$$

in which we assume that the variation $\Delta\psi_{nm}$ is uniformly distributed in the interval $(0, \psi_{0M})$, where

$$\Delta\psi_{0M} = k\xi\eta\Delta L(1/\cos\theta_M - 1) \quad (2.30)$$

where θ_M is the incident angle of the Mth modal wave field with respect to the fiber axis, η is the refractive index of the fiber, and ξ represent the strain optics correction factor, which has a typical value of ~ 0.78. From Equation (2.30) we see that, as the number of excited modes increases, the more rapidly the modal phase changes. Thus, the sensitivity of the sensing system would be higher.

Because $\sin\Delta\psi_m = \Delta\psi_M$ (for a small phase variation), by substituting Equation (2.29) and (2.30) into Equation (2.26) we have

$$MSV \approx \frac{\Delta L}{L} \iint \Big| \sum_{m=1}^{M-1} \sum_{n=m+1}^{M} 2a_{0m}a_{0n}\Big(\frac{2\pi}{\lambda}\Big)\xi\eta L \times \Big(\frac{1}{\cos\theta_M} - 1\Big)(n - m)/M \sin(\psi_{nm}) \Big| dxdy \quad (2.31)$$

where λ is the wavelength of the light source and L is he length of the probing section. From relation (2.31) we found that MSV would be linearly proportional to the fiber

perturbation. As the number of modes increases (e.g., several hundreds of modes), θ_M can be approximated by

$$\theta_M = \arcsin\left(\frac{\sin\theta_c}{\eta}\right) \tag{2.32}$$

where θ_c is the critical angle of reflection and $\sin\theta_c$ is the N.A. of the multimode fiber. We note that as the N.A. increases, $\Delta\psi_{0M}$ would not be significantly affected by the increase of the modes. Instead, θ_M is proportional to the external strain variation and the N.A. of the fiber.

2.2.3 Image Correlation Approach

In this section, we report another approach originally described in [4], which uses the *intensity inner product* of the speckle fields. In other words, by taking the intensity speckle patterns before and during the perturbation of the fiber, the intensity inner product between the two speckle patterns can be calculated. Since the fiber speckle field is caused by the modal phasing of the fiber, the intensity inner product would be highly sensitive to the fiber status changes.

As already done in the previous section, let's assume that a laser beam is launched into the sensing fiber and all the modes in he fiber are equally excited. Using the definitions of intensity given in the previous section, that is $I(x,y) = |U(x,y)|^2$, we can write the intensity inner product of the speckle fields as

$$\begin{aligned} IPC &= \iint I_0(x,y)I(x,y)dxdy \\ &= \iint \sum_{i=0}^{M}\sum_{j=0}^{M}\sum_{m=0}^{M}\sum_{n=0}^{M} a_{om}a_{oi}(a_{oi}+\Delta a_{oi})(a_{oj}+\Delta a_{oj})\exp[i(\phi_{omn}+\phi_{oij}+\Delta\phi_{ij})]dxdy \end{aligned} \tag{2.33}$$

which can be shown as

$$IPC = \sum_{i=0}^{M}\sum_{j=0}^{M} B_{ij}\exp[i(\Delta\phi_{ij})] \tag{2.34}$$

where

$$B_{ij} = \iint \sum_{m=0}^{M}\sum_{n=0}^{M} a_{om}a_{on}(a_{oi}+\Delta a_{oi})(a_{oj}+\Delta a_{oj})\exp[i(\phi_{omn}+\phi_{oij})]dxdy \tag{2.35}$$

The normalized intensity inner product can be calculated as

$$NIPC = \frac{\iint I_0(x,y)I(x,y)dxdy}{\left[\iint I_0^2(x,y)dxdy \iint I^2(x,y)dxdy\right]^{1/2}} \tag{2.36}$$

We see that if the sensing fiber is not perturbed, the two speckle patterns are virtually identical, for which we have normalized inner product (NIP)=1. However, if the fiber is perturbed, the modal phase deviates. It produces a different speckle field that causes the NIP to reduce. One may note that the smallest NIP would be approaching zero. However, NIP will never go to zero, since the intensity speckle distribution is a positive,

real quantity.

It is important to note that *complex speckle field* detection would be far more sensitive than this *intensity speckle field*. Comparing the results reported in [4] with other works of the same authors, we can observe that the sensitivity obtained with a coupled-mode analysis is about ten times greater than what has been achieved with the inner-product technique.

2.2.4 First and Second Order Moment Statistical Features

Recently, new statistical features were proposed in order to improve FSS's performance [5], namely first and second order moments.

If $I(x, y)$ is the output intensity pattern, x-mean and y-mean values are defined as

$$\mu_x = \frac{\sum_{x,y} xI(x,y)}{\sum_{x,y} I(x,y)} \qquad \text{and} \qquad \mu_y = \frac{\sum_{x,y} yI(x,y)}{\sum_{x,y} I(x,y)} \tag{2.37}$$

The pth order radial moment is

$$\mu_p = E\{r^p\} = \frac{\sum_{x,y}[(x-\mu_x)^2 + (y-\mu_y)^2]^{p/2} I(x,y)}{\sum_{x,y} I(x,y)} \tag{2.38}$$

Using (2.38), first and second moments can be expressed as

$$\mu_1 = \frac{\sum_{x,y}[(x-\mu_x)^2 + (y-\mu_y)^2]^{1/2} I(x,y)}{\sum_{x,y} I(x,y)} \tag{2.39}$$

and

$$\mu_2 = \frac{\sum_{x,y}[(x-\mu_x)^2 + (y-\mu_y)^2] I(x,y)}{\sum_{x,y} I(x,y)} \tag{2.40}$$

First and second order moment statistical features provide a better result in some sensor characteristics for image processing of speckle patterns [6]. So, they can be used as alternatives for specklegram sensor design for the applications where sensor characteristics linearity and precision are more important. Moreover, the first and the second-order moments are normalized features which do not depend on the initial output pattern. Hence, they are expected to be more robust in the presence of light source fluctuations.

Moreover, in [7] Efendioglu *et al.* exploited the nonlinear mapping ability of the Artifical Neural Networks (ANNs) to extract the statistical features coded in the speckle patterns and to design a stress sensor. This approach will be taken up in the next chapters and will be the main focus of this thesis.

2.2.5 Comparison between Different Statistical Features

Statistical features introduced in the previous sections can be used to design statistical mode sensors. In [8] the authors proposed to compare these feature regarding precision, non-linearity, and hysteresis of the statistical mode sensors. These characteristics can be considered as a sort of performance indices, taking into account different desirable properties of a generic sensor.

Precision Relates to the degree of reproducibility of a measurement. That is, if the same effect is applied, an ideal sensor will output the same value every time. The precision of a sensor is defined as

$$Precision = \sqrt{\frac{\sum e_p^2}{\sum y_{av}^2}} \tag{2.41}$$

where e_p is the root-mean-square (RMS) value of the fluctuations of the sensor's output (given a constant magnitude of perturbation), and y_{av} is the average of the estimated force values for the fixed externally applied force.

Non-linearity The linearity of a sensor is expressed as the extent to which the actual measured value departs from the ideal curve, which is the least squares line equation obtained from experimental data. Non-linearity can be calculated in terms of the parameters of the least squares line equation as follows:

$$Non - linearity = \frac{|e_n|}{|F|} \tag{2.42}$$

where F is the magnitude of the applied effect, and e_n is the deviation between the least squares line and the actual force F.

Hysteresis A sensor should be capable of following the changes of the input regardless of which direction the change is made. The difference in the output of the sensor for a given input value is called hysteresis. Referring to Figure 2.6, the calculation of the hysteresis occurs at X_m, which is the mid-point of the curve. This point can be located using the formula:

$$X_m = \left(\frac{X_{max} - X_{min}}{2}\right) + X_{min} \tag{2.43}$$

After the midpoint is located, the hysteresis can be calculated as follows:

$$Hysteresis = 100\left|\frac{Y_{mn} - Y_{mp}}{Y_{max} - Y_{min}}\right| \tag{2.44}$$

In [8], the authors set up a series of experiments applying a varying force to the top of a metal plate, with the optical fiber passing under it. Measurements were performed by adding 0.5 kg of load at a time until 4 kg and then decreased in the same steps until 0 kg.

Figure 2.7 shows the statistical features versus applied force, for each case. The results are also reported in the table below.

Feature	Precision error	Non-linearity	Hysteresis
First moment	0.1547	3.3984	31.0746
Second moment	0.1215	2.4970	30.7947
Correlation	0.3661	8.5623	20.5548
Differencing	5.6160	197.3918	16.8319

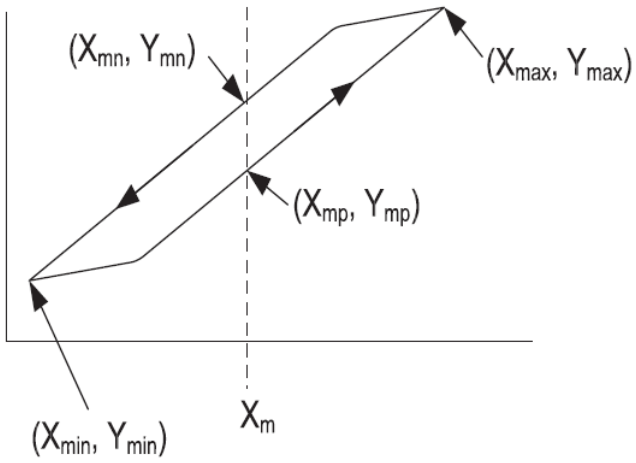


Figure 2.6: Description of hysteresis points.

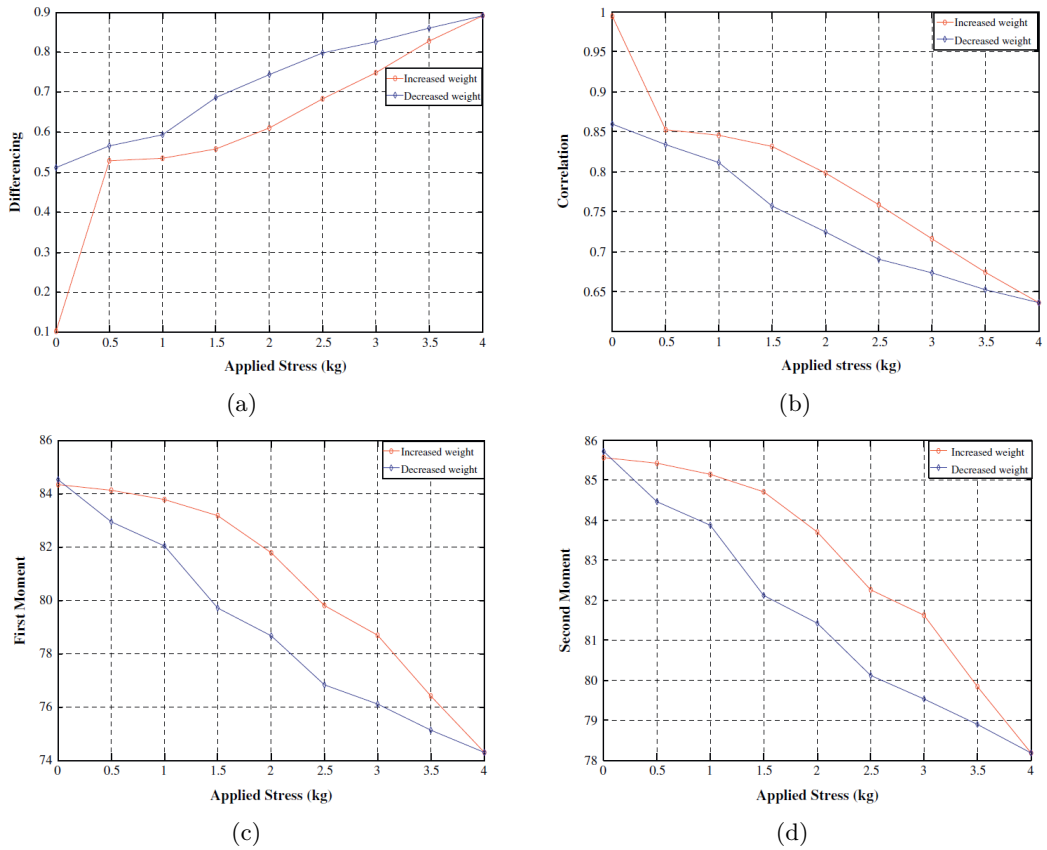


Figure 2.7: Experimental results for different statistical features versus applied stress: (a) image difference, (b) correlation change, (c) first moment, and (d) second moment.

As seen from the tabulated values, first and second order moment turns out to be the best features concerning precision error and non-linearity values. More precisely, the second order moment results to be the best feature regarding these two characteristics. However, first and second order moments are the worst techniques concerning the hysteresis, as it is shown in Figure 2.7(c) and 2.7(d). On the other hand, the differencing feature provides the best value in terms of hysteresis, but also the greater non-linearity value. Finally, correlation exhibit more linear trend respect to the differencing technique, and worse hysteresis value.

In conclusion, all these four features present advantages and disadvantages, depending on the considered characteristic. The implementation of a specific sensor should take into account what are the required characteristics of the system, and choose the appropriate feature regarding these data.

2.3 State of the Art of FSSs

Specklegram sensing has been researched for many years, and several parameters have been successfully measured using FSSs. In this section, we will investigate just two important sensing types of FSSs, regarding detection of vibration and strain. These are the most researched fields, which produced the most interesting applications. Our experimental approach will refer to these sensing techniques and capabilities. On the other hand, other parameters can be sensed with an FSS, as microwave power, temperature, refractive index, angular alignment, wavelength, assessment of hand movements, and so on.

Finally, the last section will report two interesting techniques for multiplexed-sensing implementation. As will be explained, nowadays the possibilities in this sense are strongly limited. Our work aims to offer a possible alternative to these methods, enabling the capacity of sensing a higher number of points at the same time on a single fiber. Specklegram sensing has been researched for many years and several parameters have been successfully measured using FSSs. In this section we will investigate just two important sensing types of FSSs, regarding detection of vibration and strain. The most notable applications will be reported also.

2.3.1 Vibration Sensing

The high speckle sensitivity to external perturbations has been studied for vibration sensing purposes. Although having a very high sensitivity increases the final noise, it also allows the FSS application to measure extremely low perturbations.

FSS for optical fiber vibration sensing has been demonstrated in [1]. Spillman *et al.* implemented a newly designed sensor projecting the pattern at the output end of the fiber on a CCD array, and digitally processing observed changes in the intensity distribution. They developed a mathematical model similar to what already discussed in subsection 2.2.2, but based on the distinct contribution given by the self-mode interaction and the so-called mode-mode interaction. Assuming that the perturbation $F(t)$ is small, the outcome of the Spillman's analysis is the relation between the intensity of each

speckle I_i and the two types of interaction, described by the following equation:

$$I_i = A_i\{1 + B_i[\cos(\delta_i) - F(t)\phi_i \sin(\delta_i)]\} \tag{2.45}$$

where A_i represent the self-mode interaction, and $F(t)$, B_i and δ_i define the interaction between the modes. Particularly, $F(t)$ is the external perturbation of the fiber, which modifies the index of refraction or the length of the fiber or both. The argument of the harmonic functions δ_i describes the difference in the propagation constant and the random phase of all the pairs of modes considered within the same speckle intensity I_i. A_i, B_i, ϕ_i and δ_i are constant values for any given i.

The result is identical to the result that would be expected as the output from an individual interferometer in conditions of small perturbation. Therefore, the speckle pattern can be considered as an array of interferometers subjected to the same perturbation simultaneously. To extract the perturbation information $F(t)$, a differential processing method was proposed and applied to Equation 2.45. The sum of the absolute value of the changes in all the signals can be described by:

$$\Delta I_T = \sum_{i=1}^{N} \left| - A_i B_i \phi_i \frac{dF(t)}{dt} \sin(\delta_i) \right| = \left[\sum_{i=1}^{N} |C_i \sin(\delta_i)| \right] \left| \frac{dF(t)}{dt} \right| \tag{2.46}$$

In Equation 2.46 the term within the brackets sums all the components of the speckle pattern, so the sum will remain constant despite local variations (the total energy is maintained). Equation 2.46 can be written as:

$$\Delta I_T = \left| \frac{dI_T}{dt} \right| = C \left| \frac{dF(t)}{dt} \right| = C\Delta F \tag{2.47}$$

As demonstrated by experimental tests, the dependence between the perturbation and the variation of the intensity patter in linear. This property was used to detect personnel walking over sensing fiber arranged in a serpentine pattern in an indoor environment. In particular, the differential analysis technique discussed above permits to determine the exact number of steps, as demonstrated in [9]. Another very interesting application was proposed in [10], where an FSS intrusion system is presented.

In [11] the differential process described above is developed taking into consideration the variations occurred in the speckle between two consecutive specklegrams (see Figure 2.8). Although the dynamic range of this approach very limited [12], that is the interval where the output is truly proportional to the perturbation is quite modest, it is perfectly valid when the capturing sampling frequency is high enough (e.g., video standard). Given that this method is only related to the current and previous specklegrams, all along term drifts (such as temperature) are removed from the sensor system.

In [13] this approach was proposed to monitor the health of structures as a non-destructive evaluation (NDE) technique. Indeed, the damage to the structures can be detected by analyzing the vibration signals. Since the optical fibers can be embedded within composites, they provide a unique opportunity to monitor vibration. Furthermore, optical fiber sensor can also provide the unique capability of long sensing lengths. Finally, they are the ideal choice for this type of application since they can be implemented with very low-cost components.

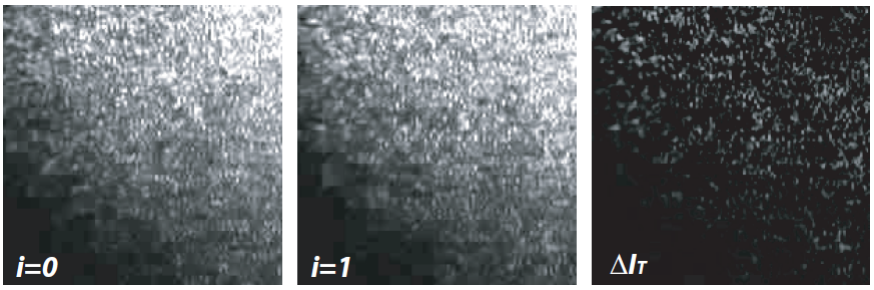


Figure 2.8: Two consecutive specklegrams and their differential processed image.

Another interesting application was presented in [14] [15], where a specklegram sensor was used to detect vital signs. Indeed, FSSs have enough sensitivity to detect the pressure variation on wrists and the changes in the chest diameter during the breathing process. In [16], the authors demonstrated that these physiological parameters can be obtain simply placing the fiber between mattress and the bed sheets, in a non-contact and continuous monitoring modality. This feature implies the ability to distinguish between the vibration due by the vital signs and the signal produced by the movements of the patient lying on the bed. Then, it is also possible to monitor the sleep quality or avoid the formation of pressure sores in mobility-restricted patients [14].

Finally, another interesting approach has been tested in [11], which relies on the distribution of the modes in the speckle pattern. Using the approximation of the propagation within a multimode fiber described by the ray model, we can state that low order modes (whose optical path are closed to the longitudinal fiber axes) will interfere centered in the middle of the speckle pattern. On the contrary, high order modes will produce speckle dots (or darks) all over the output speckle pattern. Thus, when the number of modes propagated by the fiber is high, statistically, the result of the interference between the higher nodes will be projected to the outer part specklegram. Then, we can state that this part has the highest sensibility since the higher modes are the most influenced by the perturbation to be measured. Otherwise, the speckle center is mainly driven by low order modes that are less sensitive to external perturbations.

In [11] the author adapted the differential processing scheme using rings of different radius instead of the whole speckle pattern. As expected, the results show that as the radius increased, the obtained sensitivity also increases, because these areas are mainly influenced by higher order modes, which are also more exposed to external perturbations. On the other hand, the sensitivity enhancement can also add extra noise to the measurements, because the environment variations have more influence in the final sequence.

2.3.2 Strain Sensing

Strain measurement was reported in [12], where the authors proposed a method based on the normalized absolute speckle-intensity variation (NSIV), defined as:

$$\psi_n = NSIV = \frac{\iint_s |\Delta I| dxdy}{\iint_s I dxdy} = \frac{1}{C} SIV \tag{2.48}$$

where SIV is the absolute speckle-intensity variation, and $C = \iint_s Idxdy$ is the total intensity of the speckle field, that is constant under the assumption of losses waveguide and light energy conservation. The difference between propagation constants $\Delta\beta_{ml} = \beta_m - \beta_l$ is a function of mode numbers m and l, and refractive index n. Then, a proportionality constant C_ml can be introduced and the $\Delta\beta_{ml}$ can be redefined as:

$$\Delta\beta_{ml} = nC_{ml} \tag{2.49}$$

where C_{ml} describes the difference of the propagation constants between the mth and lth modes and simplifies the relation between $\Delta\beta_{ml}$ and the refraction index n of the fiber. For any given m and l, C_{ml} is a constant. Variation of propagation constant difference can be written as:

$$\delta(\Delta\beta_{ml}L) = C_{ml}\delta(nL) \tag{2.50}$$

Expressing $\delta(nL)$ as

$$\delta(nL) = \delta nL + \delta Ln \tag{2.51}$$

we obtain the following relation:

$$\delta(\Delta\beta_{ml}L) = C_{ml}n_{eff}L\varepsilon \tag{2.52}$$

Then, we can express ψ_n as

$$\psi_n = \frac{Yn_{eff}L\varepsilon}{C}D_n \tag{2.53}$$

where D_n is constant for any given system state with M modes. Therefore, the $NSIV$ is linearly proportional to the strain ε, the length of the sensing multi-mode fiber L and the effective refractive index n_{eff}. The experiments have demonstrated that the sensor based on this model have good linearity and larger dynamic range respect to other studies [2]. However, the proposed model is still based on the assumption that the perturbation $\delta(\Delta\beta_{ml}L) = C_{ml}n_{eff}L\varepsilon$ is small, i.e. $L\varepsilon$ must be a small value.

In other works, some photorefractive materials are employed to stabilize the processed specklegrams [17] [18]. This effect can be applied to reduce the extra noise due to the typical oversensitivity of specklegrams and to extend the dynamic range for slow varying perturbations. Despite the adaptation capabilities of the proposed schemes, the setup is not suitable for low-cost applications.

In [19] Rodriguez-Cobo *et al.* extended the dynamic range of the strain specklegram sensor by a new processing scheme. The designed algorithm employs the morphological differences between two specklegrams under very different perturbations to determine key-specklegrams (KSs) as reference points. These KSs are determined during the sensor calibration ramp and they are associated with a strain value, establishing different local reference points corresponding to the different modal states.

The morphological processing begins with the conversion of the specklegram to a binary image and the application of a circular averaging filter to reduce the sharpness of the obtained dots, lowering the total amount of noise. Then, a list of sorted spots is computed, ordering them by area. The larger spots will constitute the geometrical representation of a specklegram, and they define its high-level morphological structure.

Once obtained the spots' list, the comparison between two speckles can start. The spots with larger areas of both specklegrams are compared to the closest one taking into

Figure 2.9: Morphological comparison between two specklegrams of different modal states.

account their position. Firstly, it is established the correspondence between the spots of the two speckle patterns. Then, the difference between the areas of the corresponding spots is calculated and normalized against the total area. This value can be thought as an index of similarity of the modal states represented in the two specklegrams, and it is based on the amount of coincident light speckles. A high error value indicates a completely different bright speckle distribution, what implies a different modal state.

In Figure 2.9 two specklegrams of different modal states are depicted. In the two images, the five most relevant spots are highlighted with circles, while the matched spots between the two specklegrams are marked with crosses. In this case, most of the bright speckle are is not coincident, so the obtained value is high, indicating a different modal state.

The intensity variation between two specklegrams can be defined as:

$$\Delta I_T\{i,j\} = \frac{1}{KNM}\sum_{n=1}^{N}\sum_{m=1}^{M}|p^i_{n,m} - p^j_{n,m}| \tag{2.54}$$

where K is the full scale value of the specklegram colormap and $p_{n,m}$ corresponds to the pixel of the n,m position of the specklegram i or j. Applying this notation to Equation 2.47, the perturbation value of the i-th specklegram under a weak perturbation having the initial specklegram as reference can be obtained with:

$$\Delta F\{i\} = \frac{1}{C}\Delta I_T\{0,i\} = L_0\Delta I_T\{0,i\} \tag{2.55}$$

being L_0 the proportionality constant (slope) between the speckle intensity variation, i.e. between the i-th and the first specklegram. During the sensor calibration step, different KSs are detected, and their associated slopes ($L_{KS(k)}$) and strain values ($S_{KS(k)}$) are also saved. Consequently, when a new specklegram is available, $I(i)$, it can be evaluated regarding its closest KS under the weak perturbation assumption. The absolute strain value can be obtained as

$$Strain(i) = \Delta F\{i\} = S_{KS(i)} + d(i)L_{KS(i)}I_T\{KS(i),i\} \tag{2.56}$$

where $KS(i)$ denotes the KS associated with the $I(i)$ specklegram, $S_{KS(i)}$ and $L_{KS(i)}$ denote the strain value and slope correspondent to the KS associated with the i-th

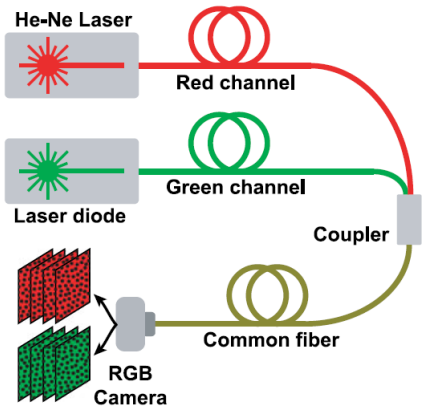


Figure 2.10: Two lasers are launched into two optical fibers and combined using a coupler.

specklegram and $d(i) \in [-1, 1]$ describes the intensity variation direction (positive or negative).

The experimental results showed good accuracy, exploitable in many structural applications. The dynamic range results strongly enhanced, and it is limited in practice only by chance to have two equals KSs (that is very low).

The greatest error was detected during a transition between two references KSs. Nevertheless, the results indicate that this approach can deal with the noisy nature of the FSSs since it provides a good response also out of the calibration points. In addition, the maximum strain error is obtained during the transition between KSs, when distances to both references are larger, but this error is not proportional to the dynamic sensor range.

2.3.3 Multiplexing Techniques

FSSs, in combination with CCDs, have been proved as a very sensitive solution at reduced costs. However, a simple FSS system employs a CCD for each sensing point, what limits their application to multi-point sensing scenarios. In [20], the authors proposed different multiplexing schemes to overcome this limitation, particularly wavelength-division and space-division techniques.

Wavelength Division Multiplexing (WDM) This approach is based on the RGB filters adopted in the modern CCD color cameras. Using three different lasers, each one matched with one of the RGB colors, it is possible to have three simultaneous channels in the same optical fiber and using just a single CCD sensor. The channels are isolated at the receiver by the RGB filters, permitting the independent analysis of each one of the output specklegrams. One of the possible drawbacks of this technique regards the structure of the modern RGB filters. Commercial CCDs usually employ correlation patterns that can produce cross-talk depending on the chosen laser sources, which can introduce some noise. However, Rodriguez-Cobo *et al.* demonstrated that by choosing suitable laser sources, this noise can be insignificant, achieving three independent channels. A schematic representation of a sensor composed of two sensing areas is depicted in Figure 2.10.

Space Division Multiplexing (SDM) This technique employs the entire CCD area by

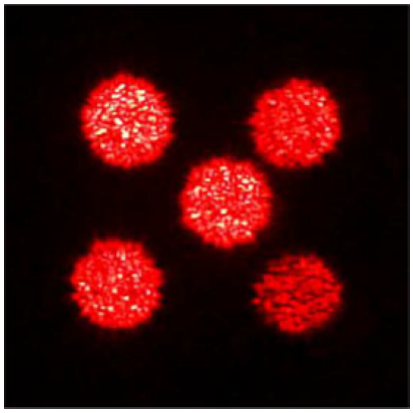


Figure 2.11: Five speckle patterns projected to the same CCD area.

projecting different speckle patterns over the whole CCD. In order to obtain reliable measurements, it is important that the total intensity of the speckle pattern must remain constant. This condition limits the number of possible multiplexed channels since the whole size of specklegrams captured must contain enough speckles. Therefore, the area of each CCD section will be limited by the specklegram diameter and also by the resolution of the CCD. Figure 2.11 shows an example where five sensing areas have been created into the same CCD by projecting specklegrams of different multimode optical fibers.

3

Machine Learning Techniques

This chapter is an introduction to machine learning, with a particular focus on neural networks. In the first section, we will define the concept of learning associated with the computer science. Then, in Section 3.2, we will present the characteristics of Artificial Neural Networks (ANNs) and their biological motivations. Finally, in Section 3.3, we will introduce the backpropagation algorithm, and some important aspects related to the learning process.

3.1 Introduction

In this section, we define what machine learning is, and we present the main approaches developed so far in this field. Furthermore, we report a brief history of this topic, in order to provide a complete view of its development and to deal with the evolution of its terminology, which has not always been linear and coherent.

3.1.1 Machine Learning Basics

Pattern recognition has been a fundamental field of research in the past decades, and it is going through an exceptional development in the last years. The main purpose of this type of analysis is concerned with the automatic detection of regularities coded into data through the use of computer algorithms. The information extracted from the raw data permits to make decisions and take actions, such as classifying the data into different categories or predict the behavior of a certain output.

In [21] Murphy defines machine learning as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data or to perform other kinds of decision making under uncertainty.

There are several ways to design a learning algorithm. Nevertheless, they are usually divided into two main approaches, called *supervised* and *unsupervised* learning. In the first one, the algorithm receives as input a labeled set of input-output pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$. Here $\mathcal{D}$ is called the *training set*, and N is the number of training samples. The final goal is to detect in an autonomous way a mapping from inputs $\mathbf{x}$ to outputs y, acquiring the ability to predict or classify the data. The algorithm consists of a comparison between the prediction y provided by the system for the input $\mathbf{x}$, and an update of the system's structure in order to obtain a more appropriate output. In general, x_i could be whatever kind of complex structured object, such an image, a speech, a video, a time series, a text message. Similarly, the output can assume whatever kind of format, but most methods assume that y_i is a *categorical* variable from the finite set,

$y_i \in \{1, \ldots, C\}$, or that y_i is a real-valued scalar. When y_i is categorical, the problem is known as *classification* or *pattern recognition*, and when y_i is real-valued, the problem is known as *regression*.

In the second approach (unsupervised), the algorithm is provided only with inputs, $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N}$ and the goal is to find 'interesting patterns' in the data. Unlike the supervised case, this learning approach is not a so well-defined problem, since it is not clear what kinds of pattern to look for, and which error metric can be assumed as the reference one.

One of the main aspects that affect the performance of machine learning algorithms is related to the *representation* of the data under analysis. The earliest techniques were based on the execution of a learning algorithm on a set of *features* extracted from the raw data. The choice of the right feature set was usually crucial for the performance of the system, and they had to be hand-designed. However, in many cases, it was difficult to know what features should be extracted. Nowadays, this approach is still in use, but in the last decade a new solution has taken hold. It consists of using machine learning to discover not only the mapping from representation to output but also the representation itself. This approach is known as *representation learning* and often provides better performance than hand-designed based ones. Furthermore, it allows a faster adjustment of the system for new tasks, with minimal human intervention.

As much it is necessary to extract high-level features from the raw data, as much representation learning is unable to provide an appropriate solution to this task. To overcome this limit, in recent years has become popular a series of technologies called *deep learning*. They introduce representations that are expressed in terms of other, simpler representations. In other terms, deep learning enables the construction of complex concepts out of simpler ones, representing data as a nested hierarchy of concepts.

One of the most important examples of a deep learning model is the *artificial neural network* (ANN), also called *feedforward neural network* or *multilayer perceptron* (MLP). This model can be seen as a mathematical function mapping some set of input values to output ones. The overall function is formed by composing many simpler functions.

Figure 3.1 shows the hierarchy between the technologies discussed in this introduction. As can be seen, all these technologies are considered a branch of artificial intelligence field.

In this chapter, we will focus on the MLP approach, starting from the biological concepts that inspired its design. Then, we will follow the evolution that leads to the deep learning technology.

3.1.2 Brief Historical Review

Multilayer perceptron and deep learning approaches are the results of researches started in the 40's. During this extended period of time, the popularity of artificial intelligence was not constant, with peaks of enthusiasm and valleys of skepticism. There have been three waves of development, each of which named these technologies differently [22].

The first one was started with *cybernetics* in the 1940s-1960s, with the development of theories of biological learning. It was mainly due to McCulloch and Pitt's pioneering work on a neuron's model and the implementations of these structures, especially the

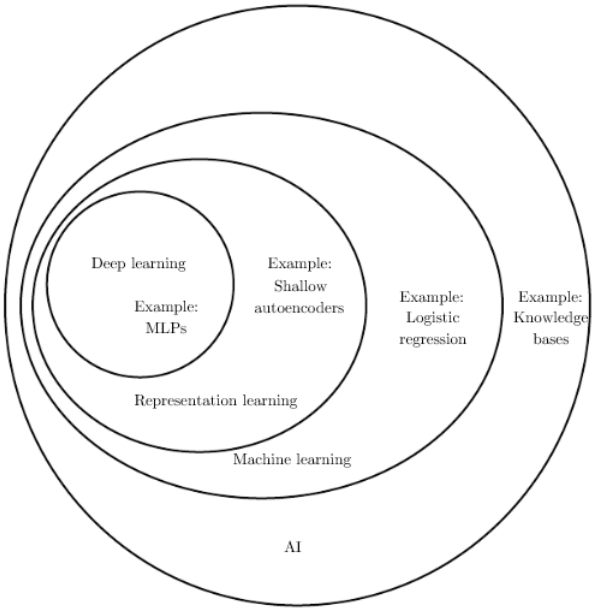


Figure 3.1: A Venn diagram showing how deep learning is a kind of representation learning, which is in turn a kind of machine learning, which is used for many but not all approaches to AI.

perceptron (by Rosenblatt in 1958). The second wave started with the *connectionist* approach of the 1980-1995 period, with backpropagation learning algorithm popularized by Rumelhart in 1986. This algorithm allows training a neural network with one or two hidden layers. Finally, the current and third wave, in which these technologies are called as *deep learning*, started around 2006 and it is still evolving.

As stated above, the earliest learning algorithms were intended to be computational models of biological learning, that is, models of how learning is achieved by neurons in the brain. For this reason, the technologies based on them were named *artificial neural networks* (ANNs). Finally, in recent years these networks were developed to design what today is defined as deep learning.

Despite the origin of these algorithms, nowadays these technologies are generally not designed to be realistic models of biological functions. Furthermore, the term 'deep learning' refers to a more general principle of learning multiple levels of composition, which can be applied in machine learning frameworks that are not necessarily neurally inspired.

3.2 Artificial Neural Networks

In this section, we develop the founding principles regarding the artificial neural networks. Firstly, we report the main motivation that led the researches in this field. Then, starting from the biological structure of the neural networks, which inspired the first studies, we will define the most popular structures that are currently used. Particularly, we will develop the concept of convolutional neural network, and its advantages respect to the previous technologies.

3.2.1 Definition and Biological Analogy

Artificial Neural Networks (ANNs) can be defined as structures comprised of densely interconnected adaptive simple processing elements (called artificial neurons or nodes) that are capable of performing massively parallel computations for data processing and knowledge representation [23]. As already mentioned in the previous section, ANNs' structure does not reflect the characteristics of a real biological system; it just abstracts some of the key features of such information processing system. Particularly, this type of technology permits to deal with nonlinearity, achieve high parallelism, robustness, fault and failure tolerance, learning, ability to handle imprecise and fuzzy information and to generalize concepts. Artificial models possessing such characteristics are desirable because [24]:

- nonlinearity allows the fit of a greater and more complex variety of data;
- noise-insensitivity provides accurate prediction in the presence of uncertain data due to high noise levels and measurements errors;
- high parallelism permits to design faster systems. Furthermore, it reduces the probabilities of hardware failures;
- learning and adaptivity allow dynamic updates of the system, which can change its internal structure in response to changing data conditions;
- generalization enables application of the model to unlearned data.

In Figure 3.2 is depicted a schematic of an oversimplified biological neuron with three major function units: dendrites, cell body (also called soma), and axon. The dendrites are responsible for receiving the input signals from other neurons and pass them over to the cell body. On the other hand, the axon transmits the signals received from the cell body to the dendrites of the neighboring neurons through the *synapse* (a microscopic gap). This element releases a neurotransmitter in quantities proportional to the incoming electrical signal to transmit it to another group of neurons. The receiving neuron generates a new electrical signal depending on the intensity of the incoming signals, or if the quantity of neurotransmitter exceeds its internal threshold. The amount of signal that passes through a receiving neuron depends on the intensity of the signal emanating from each of the feeding neurons, their synaptic strengths, and the threshold of the receiving neuron. As illustrated in Figure 3.2, it is important to note that a neuron has a large number of dendrites/synapses. It means that its output is usually affected by a large number of signals, which are combined and transmitted to other neurons simultaneously. The contribution of each signal to the final output depends by its strength and can either excite or inhibit the firing of the neuron.

The analogy between an artificial neuron and a biological one is that the connections between nodes represent the axons and dendrites, the connection weights represent the synapses, and the threshold approximates the activity of the soma. Besides, both the biological network and ANN learn by incrementally adjusting the magnitudes of the weights or synapses' strengths.

The model of neuron in its simplest form can be considered a threshold unit, that is a processing element that collects inputs and produces an output only if the sum of the

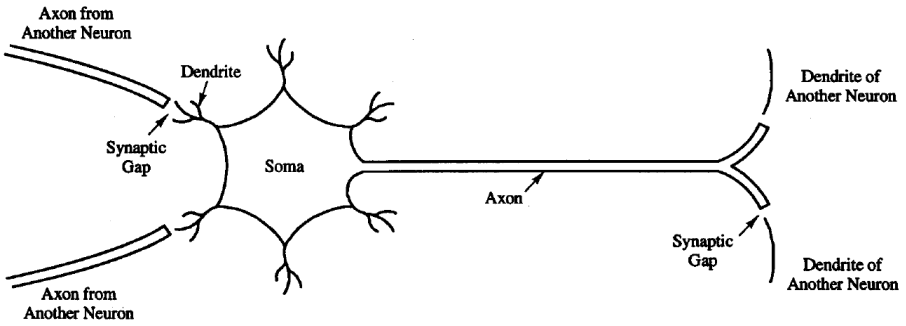


Figure 3.2: Schematic representation of a biological neuron.

inputs exceeds an internal threshold value [25]. The tasks performed by a neuron can be summarized in two steps: 1) collect and sum of all the signal at the synapses, and 2) if the collected signal is great enough to exceed the threshold, send a signal down the axon to other neurons. Artificial neural systems are created by interconnecting many of these simple neurons into a network.

3.2.2 The Artificial Neuron

McCulloch and Pitts is the most widely used neuron model, and it is illustrated in Figure 3.3. Each kth neuron consists of two parts [26]: the net function v_k and the activation function $\varphi(\cdot)$. This mathematical neuron computes a weighed sum of its m inputs signals, x_j, $j = 1, 2, \ldots, m$, and generates an output of 1 if the sum is above a certain threshold b_k. Otherwise, an output of 0 results. Mathematically,

$$y_k = \begin{cases} 1, & \text{if } \sum_{j=1}^{m} w_{kj} x_j \geq b_k \\ 0, & \text{if } \sum_{j=1}^{m} w_{kj} x_j < b_k \end{cases} \tag{3.1}$$

where w_{kj} is the synapse weight associated with the jth input of the kth neuron. Positive connection weights ($w_{kj} > 0$) enhance the net signal (v_k) and excite the neuron, and the link is called *excitatory*, whereas negative weights reduce v_k and inhibit the neuron activity, and the link is called *inhibitory*. In Equation (3.1), the neuron threshold may be considered as another weight $w_{k0} = b_k$ attached to the neuron with a constant input $x_0 = 1$. In such case, the summation in Equation (3.1) is run from 0 to m, and the net signal v_k is compared to 0.

The McCulloch-Pitts neuron has been generalized in many ways. For example, other types of network input combination have been proposed in the literature. Furthermore, alternative activation functions have been adopted, such as piecewise linear, sigmoid, or Gaussian, as shown in Figure 3.4. The sigmoid function is by far the most frequently used in ANNs. It is a strictly increasing function that exhibits smoothness and has the desired asymptotic properties. The standard sigmoid function is the *logistic* function, defined by:

$$g(x) = \frac{1}{1 + \exp(-ax)} \tag{3.2}$$

where a is the slope parameter.

The perceptron presented by Rosenblatt in 1958 is an evolution of the original McCul-

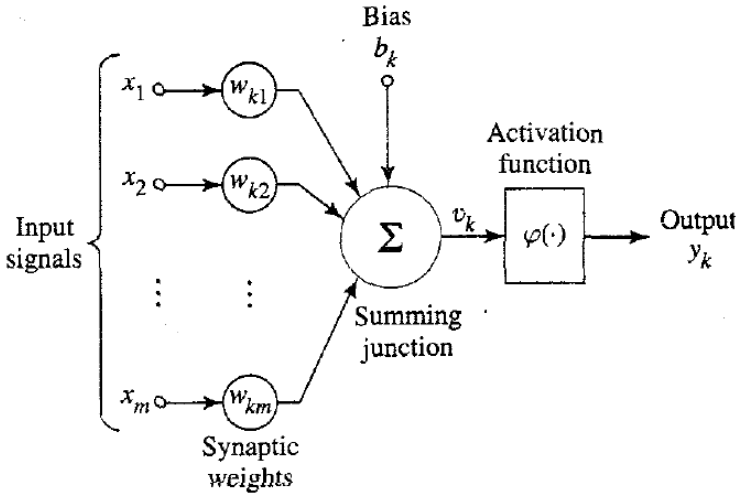


Figure 3.3: McCulloch and Pitt's neuron model.

loch and Pitts' model, which introduced a learning rule and other minor improvements. An insight into learning algorithms is presented in Section 3.3.

3.2.3 Neural Network Topology

As previously mentioned, artificial neural systems are created by interconnecting multiple neurons to form a network. This network can be designed according to different topologies, that yields different behaviors. The topology can be described with a weighted direct graph, where nodes are the neurons, and the directed arcs are the synaptic links, i.e. the connections between neuron outputs and neuron inputs.

The topology configurations can be divided into two main categories, represented in Figure 3.5. The acyclic topology, depicted in Figure 3.5(a), does not provide for feedback loops. The ANNs based on this topology are called *feed-forward*. They are memory-less systems, that is, their response to an input is independent of the previous network state. Multilayer perceptron represents the most common family of feed-forward networks, in which neurons are organized into layers that have unidirectional connections between them (see Figure 3.6) [27]. Such type of neural network is often used to approximate a nonlinear mapping between inputs and outputs.

The second category consisting of neural networks with cyclic topology. As shown in Figure 3.5(b), in this configuration the graph of the network contains at least one cycle formed by directed arcs. This type of network is also known as *recurrent* (or *feedback*) network. Unlike feed-forward networks, feedback ones have internal memory, providing a nonlinear dynamic system model. In other words, the output of the system depends not only on the input but also on the internal state of the network. Furthermore, every time a new input is presented to the network, the actual state of the system is updated. Recurrent neural networks often exhibit complex behavior and remain an active research topic in the field of artificial neural networks.

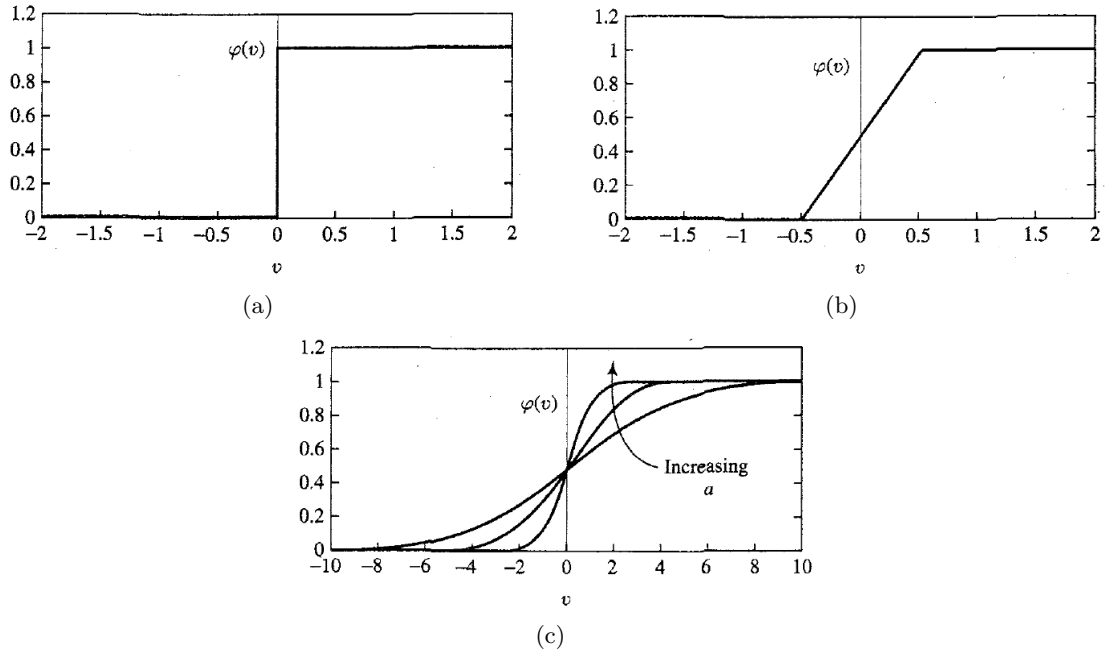
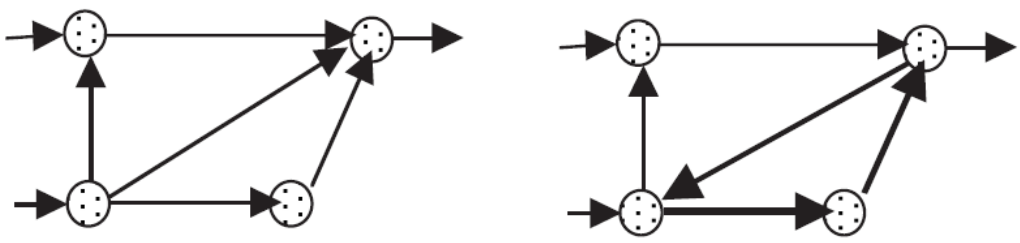


Figure 3.4: Different activation functions: (a) Threshold function, (b) Piecewise-linear function, (c) Sigmoid function for varying slope parameter a.



(a) Acyclic topology

(b) Cyclic topology

Figure 3.5: Illustration of (a) an acyclic graph and (b) a cyclic graph. The cycle in (b) is emphasized.

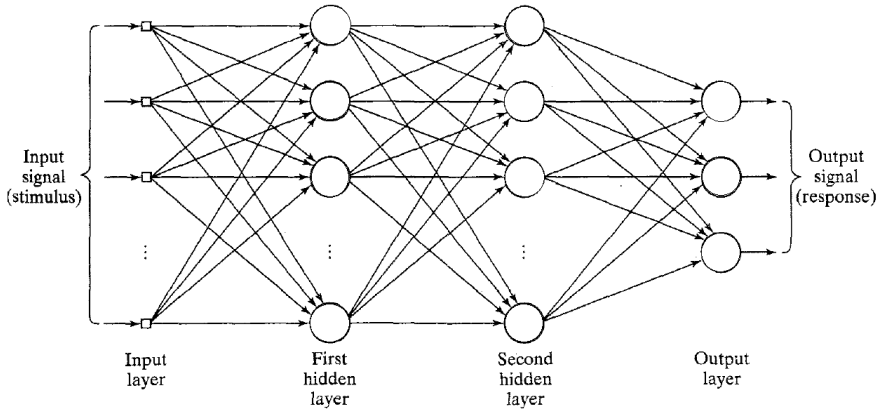


Figure 3.6: Architectural graph of a multilayer perceptron with two hidden layers.

3.2.4 Multilayer Perceptron Neural Networks

A multilayer perceptron (MLP) neural network consists of a feed-forward, layered network of McCulloch and Pitts' neurons. Each neuron in an MLP has a nonlinear activation function that is often continuously differentiable. Some of the most frequently used activation functions for MLP include sigmoid function and the hyperbolic tangent function.

Figure 3.6 shows the architectural graph of a multilayer perceptron, where neurons are represented by circles and are divided into layers. While the inputs at the left are also labeled as the input layer, there is usually no neuron model implemented in that layer. Thus, for example, the network represented in Figure 3.6 is defined as a three-layer feed-forward network since the input nodes are usually not counted as a layer. The inputs pass through a series of *hidden layers*, where this name refers to the fact that the outputs of these neurons remain on the inside part of the network, and it is not available to the user.

The network illustrated in Figure 3.6 is *fully connected*, which means that a neuron in any layer of the network is connected to all the neurons in the previous layer. Furthermore, there are no connections between units in the same layer and no feedback connections between layers. In other words, any acyclic interconnection between neurons is allowed, and the signal flow through the network in a forward direction. For these reasons, this configuration yields to a feed-forward network, and it is one of the most popular for MLP implementation.

The key point that has made MLPs one of the most popular technologies among researches and users of neural networks is its learning algorithm, namely the *back-propagation* (BP) algorithm. It has been proved that with a sufficient number of hidden neurons, an MLP with a few as two hidden layer neurons is capable of approximating an arbitrary complex mapping within a finite support [28]. More details about this algorithm will be reported in Section 3.3.

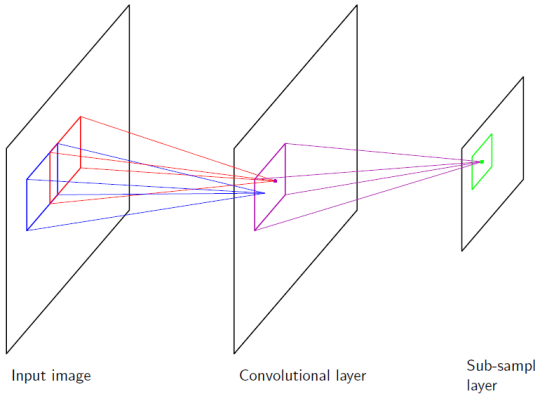


Figure 3.7: Diagram illustrating part of a convolutional neural network, showing a layer of convolutional units followed by a larger subsampling units. Several successive pairs of such layers may be used.

3.2.5 Convolutional Neural Networks

Artificial neural networks can be used to recognize data expressed in a very large number of ways. However, despite the full-connected structure of an ANN can yields good solutions in almost all cases if a large training set is available, this could be a suboptimal approach for some types of data. For example, given an image to analyze, an ANNs considers each pixel as an independent variable, regardless of its absolute and relative positions. On the contrary, the pixels in an image are related each other, and in particular, the nearby ones are more strongly correlated than more distant pixels.

A *convolutional network* is a multilayer perceptron explicitly designed to recognize two-dimensional shapes with a high degree of invariance to translation, scaling, skewing, and other forms of distortion [29]. These characteristics are achieved by extracting *local* features that depend only on small subregions of the image. The extracted information can then be merged in later stages of processing in order to detect higher-order features and ultimately to yield information about the image as a whole. Also, local features that are useful in one region of the image are likely to be useful in other regions of the image, for instance, if the object is translated.

These notions are incorporated into convolutional neural networks through three mechanisms: local receptive fields, weight sharing, and subsampling [30]. An overview of the whole structure of a CNN is depicted in Figure 3.7.

Local receptive fields As we have seen in the previous section, in a traditional neural network the output of each neuron (also called *unit* in this contest) depends on the output of the previous layer through a weight set. Thus, every neuron in a layer interacts with every neuron in the previous one. Convolutional networks, however, typically implement *sparse interactions* (also referred to as *sparse connectivity* or *sparse weights*): each neuron takes its synaptic inputs from a local *receptive field* in the previous layer, thereby forcing it to extract local features. This characteristic leads to a reduction of the number of parameters that need to be stored and to a decrease of the number of operations required to analyze the input data.

In Figure 3.8 this concept is illustrated through a directional graph. The gray circles highlight the neurons that affect the output of neuron g_3. It is clear that

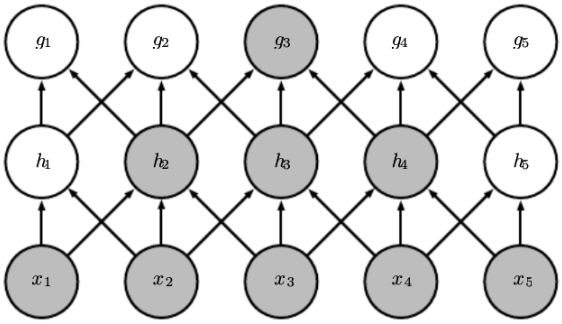


Figure 3.8: The receptive field of the units in the deeper layers of a convolutional network is larger than the receptive fields of the units in the shallow layers.

even if the direct connections in a CNN are sparse, units in the deeper layers can be indirectly connected to all or most of the input image.

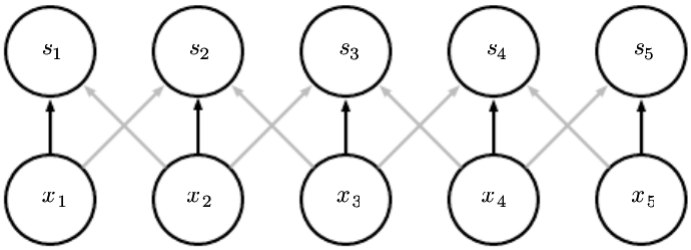
Weight sharing In a traditional neural network, each element of the weight matrix is used exactly once when computing the output of a layer. It is multiplied by one element of the input and then never revised. The structure of the computational layers in a convolutional network is different, since these are composed of multiple *feature maps*. Each map forms a plane within which the individual neurons are shared or, from another point of view, they are tied, because the value of the weight applied to one input is tied to the value of a weight applied elsewhere. Indeed, each member of the set of parameters is used at every position of the input. As units in a feature map take inputs only from a small region of the image, we can note that this process, rather than learning a separate set of parameters for every location, detect the same pattern at different locations in the input image. Since an image contains several features, in order to detect them and build an effective model, this type of systems are equipped with multiple feature maps in the convolutional layer, each having its own set of weight and bias parameters.

This form of structural constraint leads to the following beneficial effects:

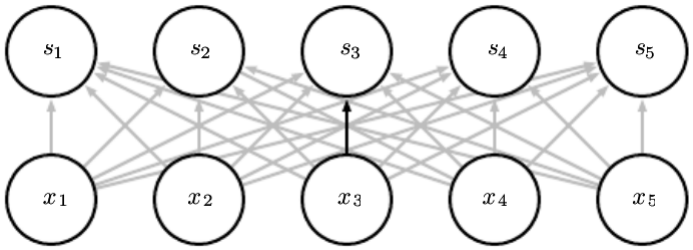
- *Shift invariance*: if a specific feature (or the whole image) is shifted, there will be just a shift of the activations into the corresponding feature map by the same amount.
- *Reduction in the number of free parameters*: in addition to the reduction provided by the sparse connectivity, the weight sharing allows a further reduction of them.

In Figure 3.9 is illustrated a simple example about the concept of parameter sharing. The black arrows indicate the connections that use a particular parameter in two different models. In Figure 3.9(a) each unit of the layer 1 interacts in the same way with the corresponding unit in the layer 2. That is, the parameter represented by the black arrow is shared to be used at all input locations, as in the case in CNNs. Contrary, in Figure 3.9(b) there is a single black arrow, which means that there is no parameter sharing. The gray arrows in the background indicate that the image is relative to a fully connected model.

The term 'convolutional' associated with this type of network is related to how the



(a) Linearly-separable classes



(b) Nonlinearly-separable classes

Figure 3.9

evaluation of the activations of units is implemented. In fact, it is equivalent to a convolution of the image pixel intensities with a 'kernel' comprising the weight parameters.

Subsampling In order to sum up information extracted from the network, the data coming from the convolutional units is subsampled in an appropriate layer. More specifically, the subsampling layer is composed of several planes, one corresponding to each of the feature maps in the convolutional layer. Each unit executes the subsampling on a small receptive field in the corresponding feature map, as is shown in Figure 3.7. As can be noted, differently from the computation of the convolutional layer, in this case the receptive fields are chosen to be contiguous and nonoverlapping.

The operations performed on the inputs are *local averaging* and *subsampling*. They allow a reduction of the sensitivity of the feature map's output, in order to better manage shifts and other forms of distortion.

In a practical architecture, several convolutional and subsampling layers are alternated in the network. The output of each pair of these layers provides a more general representation of the image, summarizing its features and adding invariance to input transformations. The increasing depth related to this type of structure represents the reason why this technology has been called *deep learning*.

In Figure 3.10 is represented a well-known convolutional neural network, introduced by Yann Le Cun *et al.* in 1998. As we can see, each convolutional layer has its own set of feature maps, that accepts inputs from the previous subsampling layer. It is important to note that the number and the dimensions of the feature maps vary through the network: the 'deeper' layers are characterized by a lower spatial resolution, that is compensated by an increasing number of feature planes. This approach was inspired by Hubel and Wiesel's work on the cat's primary visual cortex, which identified orientation-selective

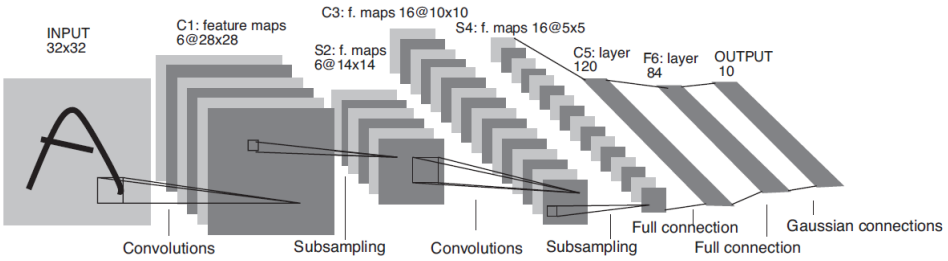


Figure 3.10: LeNet5, a convolutional neural network for classifying handwritten digits.

simple cells with local receptive fields similar to convolutional layers, and *complex cells* performing subsampling-like operations. The alternation between these layers enables the detection of higher-order features at a lower resolution.

Unlike other standard algorithms in image classification, the preprocessing is not frequently performed in CNNs. Instead of setting parameters, we just need to train the filters in CNNs. Moreover, in feature extraction, CNNs are independent of prior knowledge and human interference. The training procedure for a CNN is similar to that for a standard ANN using backpropagation algorithm, which will be described in the next section.

Nowadays, this topology continues to be popular in neurally-inspired models of visual object recognition. In the next chapters, we will use the pattern recognition capabilities of this type of neural network to categorize the specklegram at the output of the fiber. Further details about the specific implementation of the CNN will be reported in Chapter 4.

3.3 Learning from Data

The ability to learn is a fundamental trait of intelligence. In the ANN context, it can be interpreted as the ability to change the internal structure of a system by itself, according to the external stimuli. In practice, this implies the update of the connection weights so that a network can efficiently perform a specific task. This operation is executed by trying to fit the resulting internal model of the system with the underlying rules expressed from a given collection of representative examples.

In this section, we will first define the concept of learning process, in order to understand in which environment a neural network operates. Then, we will analyze how the network weights are updated, with reference to the backpropagation algorithm, based on the error-correction rule. Finally, we will highlight some important problems related to the correct choice of the complexity of the system, that has to concern with the specific task assigned.

3.3.1 Context and Definitions

We can define the learning process quoting Mitchell [31]: 'A computer program is said to learn from experience E with respect to some class of task T and performance measure P, if its performance at task T, as measured by P, improves with experience X'. This is

a very intuitive definition that can be useful to define better what are the elements that will constitute our analysis system.

The Task, T It defines how the machine learning algorithm should process an example, that is an input data. In particular, the task concerns the features that constitute and describe the example, and that have to be detected. For example, in our case, the input will consist of an image, which can be described through the values of its pixels.

There are many kinds of tasks that can be solved with machine learning. In our case, we will have to deal with a *classification problem*. In this type of task, the machine learning algorithm is asked to specify which of k categories some input belongs to. In other words, the learning algorithm has to produce a function $f\colon \mathbb{R}^n \to \{1, \ldots, k\}$ that relates the given inputs with a specific category. The output of the algorithm is indicated by $y = f(\mathbf{x})$, where $\mathbf{x}$ is the array of detected features. It can be in the form of a numeric code (identifying a specific category) or a probability distribution over categories.

The Performance Measure, P For the classification task, the performance measure is simply the *accuracy* of the model. More specifically, the accuracy is the proportion of examples for with the model produces the correct output. Alternatively, the equivalent information can be obtained by measuring the *error rate*, i.e. the proportion of examples for which the model produces incorrect output.

The experience, X It refers to which learning rules govern the updating process, that is how the weights inside the system are adjusted. As already discussed in the introduction, machine learning algorithms can be categorized in *supervised* or unsupervised. In the first case, the system is provided with a correct output for every input pattern. The learning algorithm aims to determine the set of weights that minimize the difference between the output of the system and the correct one. In the second case, unsupervised learning explores the underlying structure in the data, without a correct output that leads the analysis.

3.3.2 Error-Correction Learning Rule

As mentioned in the previous sections, multilayer perceptrons and convolutional neural network are some of the most popular techniques adopted in pattern recognition field. They are both based on the back-propagation algorithm, which implements the supervised learning paradigm already described above. The BP algorithm, in turn, is based on the *error-correction learning rule*, which we are about to introduce in this section.

To illustrate this algorithm, let's consider the simple case in which the output layer is constituted by a single neuron k, as illustrated in Figure 3.11. The input signal $\mathbf{x}(n)$ of the neuron k is originated by one or more layers of hidden neurons, which are driven by an input vector applied to the source nodes at the input layer of the network. The argument n represents the discrete time-dependency of the variable. During the learning process, the actual output, $y_k(n)$, generated by the output of the neuron k (and in this case by the whole network) may not equal the desired output, $d_k(n)$. Consequently, we

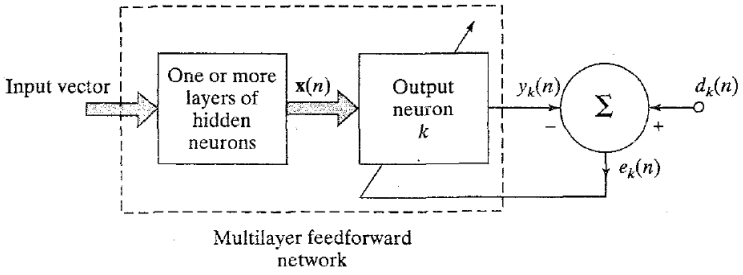


Figure 3.11: Block diagram of a neural network which implements error-correction learning.

can define an *error signal* $e_k(n)$ as

$$e_k(n) = d_k(n) - y_k(n) \tag{3.3}$$

that can be used in a control mechanism (see Figure 3.11) to modify the connection weights, in order to gradually reduce the error itself in a step-by-step manner. This objective is achieved by minimizing a *cost function* or *index of performance*, E, defined as the instantaneous value of the error energy:

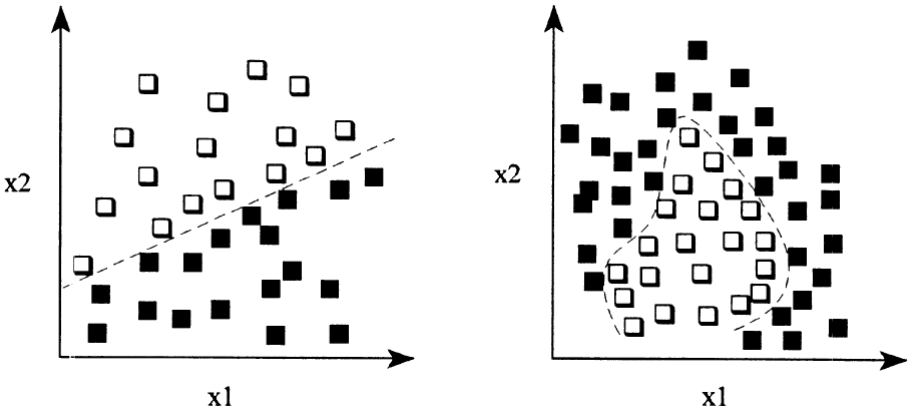
$$E = \frac{1}{2}e_k^2(n) \tag{3.4}$$

The algorithms continue to update the weight of the network until the system reach a steady state. At that point, the learning process is terminated.

Rosenblatt proposed in 1962 a learning procedure that provides the optimal weight vector of a perceptron in a finite number of iterations, given a set of training inputs and regardless of the initial values of the weights. However, it is important to note that this learning procedure takes place only when the perceptron output is different from the desired signal, i.e. when $e_k(n) \neq 0$. Furthermore, Rosenblatt proved that a single-layer perceptron network performs accurately only with linear separable classes, that is when the objects at the input of the network can be divided into classes separated by a linear hyperplane. This statement is called *perceptron convergence theorem*. Figure 3.12 shows an example of linearly and nonlinearly separable two-object classification problems. In practice, it is not possible to know in advance whether the patterns are linearly separable. In order to overcome this limitation, additional layers (the hidden ones) were added, leading to the multilayer perceptron structure.

3.3.3 Backpropagation Algorithm

The learning phase of a back propagation network can easily be viewed as a classical unconstrained nonlinear optimization problem [32]. The solution of such a problem concerns the modification of a set of independent variables, in order to minimize or maximize a certain cost function. The MLP fits this model quite nicely. In this context, the weights can be considered as the independent variables and the function E, defined in Equation (3.4), is the quantity that we aim to minimize. Hence, ultimately, the goal of training the network is to update the weights so that the network outputs for each pattern match the desired outputs.



(a) Linearly-separable classes (b) Nonlinearly-separable classes

Figure 3.12

The cost function defined in Equation (3.4), can be seen as a function depending on the set of weights $\mathbf{w}$ of the network, and denote as $E(\mathbf{w})$. It can be represented in the k-dimensional space of the vector $\mathbf{w}$, where k is the number of weights of the network, i.e. the dimension of the vector $\mathbf{w}$. An example of this type of representation is provided in Figure 3.13 for $k = 2$, where we can note that the cost function forms an irregular multidimensional complex hyperplane, with peaks, saddle points, and minima. The aim of backpropagation algorithm is to find a weight vector $\mathbf{w}$ which minimizes the error function $E(\mathbf{w})$.

Firstly, we can note that changing the weight vector from $\mathbf{w}$ to $\mathbf{w} + \delta\mathbf{w}$, the error function varies of a quantity that can be approximated to $\delta E \simeq \delta\mathbf{w}^T \nabla E(\mathbf{w})$, where $\nabla\mathbf{w}$ points in the direction of greatest rate of increase of the error function. If we assume that $E(\mathbf{w})$ is a smooth continuous function of $\mathbf{w}$, that its minimum will occur at a point in weight space such that

$$\nabla E(\mathbf{w}) = 0 \tag{3.5}$$

If this condition does not hold for the current weight vector, it can be changed in the direction of $-\nabla E(\mathbf{w})$, in order to reduce the error (see Figure 3.13). This is a necessary but not sufficient condition, since there are points at which the gradient vanishes that do not represent a minimum (or at least not a global one). This observation highlight the complexity associated with the BP's aim, i.e. find an optimal weight vector.

The backpropagation algorithm is based on proceeds by iterations, which in turn are constituted of two flows, represented in Figure 3.14.

- The *function signals* are the result of the inputs that propagate through the network, from the input layer to the output one. They produce a solution, that can be considered as a combination of the functions provided by the hidden neurons of the network. If we consider an ANN with assumed initial weights (usually small random values), the functional signals are calculated on the training examples provided to the input layer of the network.

- The *error signals* are originated at the output end of the network and propagate backward through the network. They are error-dependent functions.

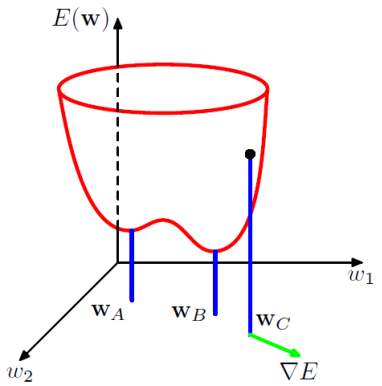


Figure 3.13: Geometrical view of the error function $E(\mathbf{w})$ as a surface sittinf over weight space. Point $\mathbf{w}_A$ is a local minimum and $\mathbf{w}_B$ is the global minimum. At any point $\mathbf{w}_C$, the local gradient of the error surface is given by the vector ∇E.

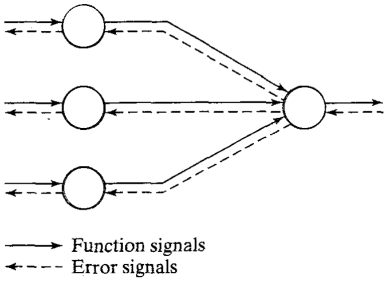


Figure 3.14: Illustration of the directions of two basic signal flows in a multilayer perceptron: forward propagation of function signals and back-propagation of error signals.

In order to develop the BP algorithm in a simpler way, we will consider the learning process of a single neuron. In Figure 3.15 are represented the signals involved in this procedure and the neuron, which is divided into two parts, as already seen in Section 3.2.2. The first one, which output is denoted as u, consists of a summation unit, while the second part represents a nonlinear activation function, $z = f(u)$. Furthermore, we can note that the neuron is driven by two inputs $[x_1 x_2]$, associated to as many weights. In addition, the bias term is represented by a constant input signal (equal to one) associated to the weight w_0. We can define the net function as:

$$u = \sum_{i=0}^{2} w_i x_i = \mathbf{wx} \tag{3.6}$$

where $x_0 = 1$, $\mathbf{w} = [w_0 w_1 w_2]$ is the weight matrix, and $\mathbf{x} = [1 x_1 x_2]^T$ is the input vector.

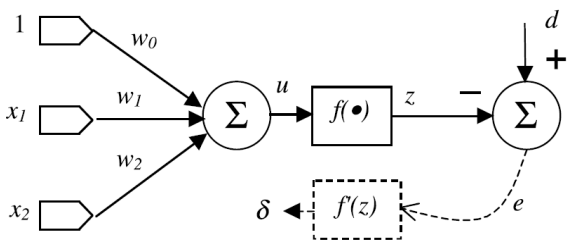


Figure 3.15: MLP example for back-propagation training: single neuron case.

On the rightest part of the Figure 3.15, we can see that the error signal e is calculated as the difference between the desired signal and the output one (as in the previous section). Given a set of training samples $\{(\mathbf{x}(n), d(n)); 1 \leq n \leq N\}$ and $\mathbf{w}$ initialized with small random values, the error back-propagation training begins at the input layer. All N inputs are transmitted to the single neuron, which produces the corresponding output $\{z(n); 1 \leq n \leq N\}$. Then the cost function is computed as:

$$E = \frac{1}{2}\sum_{n=1}^{N}[e(n)]^2 = \frac{1}{2}\sum_{n=1}^{N}[d(n) - z(n)]^2 = \frac{1}{2}\sum_{n=1}^{N}[d(n) - f(\mathbf{w}\mathbf{x}(n))]^2. \tag{3.7}$$

The objective is to adjust the weight matrix $\mathbf{w}$ to minimize the error $E(\mathbf{w})$. This leads to a nonlinear least square optimization problem. Basically, these algorithms adopt a similar iterative formulation:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \delta\mathbf{w}(t) \tag{3.8}$$

where $\Delta\mathbf{w}$ is the correction made to the current weights $\mathbf{w}(t)$. Different algorithms differ in the form of $\Delta\mathbf{w}$. The standard backpropagation algorithm adopts the steepest descend gradient algorithm, which defines the correction of the weights as $\delta\mathbf{w} = -\eta\mathbf{g}(t) = \eta dE/d\mathbf{w}$. In the previous equation, $\mathbf{g}$ is known as the gradient vector, and η is the step size (or learning rate). The derivative of the scalar quantity E with respect to individuals weights can be calculated as:

$$\frac{\partial E}{\partial w_i} = \frac{1}{2}\sum_{n=1}^{N}\frac{\partial[e(n)]^2}{\partial e_i} = \sum_{n=1}^{N}[d(n) - z(n)]\left(-\frac{\partial z(n)}{\partial w_i}\right) \qquad \text{for } i = 0, 1, 2 \tag{3.9}$$

where

$$\frac{\partial z(n)}{\partial w_i} = \frac{\partial f(u)}{\partial u}\frac{\partial u}{\partial w_i} = f'(u)\frac{\partial}{\partial w_i}\left(\sum_{j=0}^{2} w_j x_j\right) = f'(u)x_i. \tag{3.10}$$

Then, we can rewrite Equation (3.9) as:

$$\frac{\partial E}{\partial w_i} = -\sum_{n=1}^{N}[d(n) - z(n)]f'(u(n)x_i(n). \tag{3.11}$$

Let's define the signal $\delta(n)$ as the signal error $e(n)$ multiplied by the derivative of the activation function $f'(u(n))$, that is $\delta(n) = [d(n) - z(n)]f'(u(n))$. In this way, we can express Equation (3.11) as:

$$\frac{\partial E}{\partial w_i} = -\sum_{n=1}^{N}\delta(n)x_i(n). \tag{3.12}$$

The signal $\delta(n)$ represent the amount of correction needed to be applied to the weight w_i for the given input $x_i(n)$. Thus, the overall change of the ith weight is the sum of such contribution over all N training examples. Hence, we can express the weight update as:

$$w_i(t+1) = w_i(t) + \eta\sum_{n=1}^{N}\delta(n)x_i(n). \tag{3.13}$$

Each time the weights are updated is called an epoch. In this case, the epoch size is equal to N, that is the number of training examples.

The gradient descent method very often misbehaves on large-scale problems and because its success depends on the user dependent parameters. Other methods, as the conjugate gradient algorithm provides a faster and more stable method to train a multilayer feed-forward neural network respect to the traditional steepest descent [33]. In particular, this method will be implemented in our analysis of the specklegrams, described in the next chapters.

3.3.4 Training and Testing

The central challenge in machine learning is to build a network able to minimize the output error on new, previously unseen inputs. The ability to provide this type of capacity is called *generalization*.

Typically, when training a machine learning model, we have access to a training set. The first step is to compute some error measure on the training set, called *training error*, in order to reduce it. This first phase is nothing but an optimization problem, as we have seen in the previous sections. A much more relevant measure is the error rate of the system in the field, where it would be used in practice. In order to estimate this parameter, the accuracy of the trained network is measured on a set of samples disjoint from the training set, called the test set. The real goal of the machine learning algorithm is to provide a low error related to this data, called *test error* or *generalization error*.

The gap between test error, E_{test}, and training error, E_{train}, has been investigated in many theoretical and experimental works. They indicate that the gap decreases with the number of training samples approximately as [34]:

$$E_{test} - E_{train} = k(h/P)^{\alpha} \tag{3.14}$$

where P is the number of training samples, α is a number between 0.5 and 1.0, and k is a constant. The quantity h is a measure of 'effective capacity' or complexity of the system, that summarize the ability to fit a wide variety of functions.

It has been proved that the gap always decreases when the number of training samples increases. Furthermore, as shown in Equation (3.14), the width of the gap is related to the capacity h of the system. In general, as the capacity h increases, E_{train} decreases. Thus, when increasing the capacity, there is a trade-off between the decrease of the training error and the increase of the gap, with an optimal value of the capacity h that achieves the lowest test error E_{test}.

Furthermore, as the capacity h increases, E_{train} decreases. Therefore, when increasing the capacity h, there is a trade-off between the decrease of E_{train} and the increase of the gap, with an optical value of the capacity h that achieves the lowest generalization error E_{test}.

Most learning algorithms are designed to minimize the training error as well as some estimate of the gap. These two factors correspond to as many central challenges in machine learning: *underfitting* and *overfitting*. The first one occurs when the algorithm is not able to provide a sufficiently low error value of the training set. In other words, underfitting is usually a symptom of an under-sized system in terms of capacity. On the

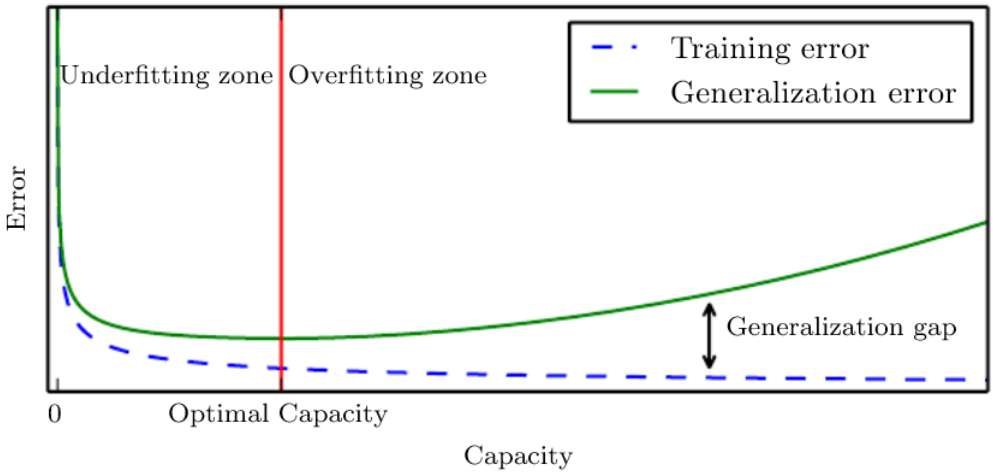


Figure 3.16: Typical relationship between capacity and error.

other hand, overfitting occurs when the system is over-sized and leads to a significant gap between the training error and test error. This problem is due to the wrong identification of the key features from the algorithm, which are not suitable for the desired classification. Thus, a correct quantification of the necessary capacity is important to obtain an efficient system. A graphical representation of these concepts is reported in Figure 3.16, where the training and the test errors are related to the capacity.

Finally, another crucial aspect that affects the performance of a machine learning algorithm is related to the amount of training data they are provided with. It has to be sufficiently high, in relation to the complexity of the task.

4

Experimental Setup

In this chapter, we will define the experimental setup used to test the machine learning capabilities on specklegram categorization. Let us remember that the aim of this thesis is to build a system able to recognize in which position along the fiber optic sensor the perturbation has occurred. Firstly, we will introduce other interesting solutions developed in the last years to reach similar purposes. Then, in Section 4.2.1 we will report all the main characteristics of the types of optical fibers used for our experiments, with particular regard to the number of modes propagated and the related sensitivity. In Section 4.2.2, we will describe the experimental setup used to produce a great number of perturbations, in order to train the ANNs. Before to use the acquired specklegrams, it is necessary to preprocess those, in order make them suitable for the neural networks' input. This process is described in Section 4.3, followed by the presentation of the structures of the ANNs used in our analysis.

4.1 Hypothesis and Motivations

The multimode optical fibers (MMFs) are viewed as unreliable in some sensing context. This consideration is usually due to the supposed 'randomization' of light propagated through it, that is typically attributed to undetectable deviations from the ideal fiber structure. It is a commonly held belief that the chaos provided by MMF is unpredictable and that the light is influenced proportionally to the length of the fiber.

In the last few years, this belief is changing, and some authors have started to study this topic from a deterministic point of view. For example, in [35] Plöschner *et al.* proposed a way to determine the transformation matrix (a subpart of the scattering matrix) through numerical modeling, instead of computing it empirically. This approach allows the ability to predict the system's behavior without performing its characterization whenever a deformation occurs. While this technique was used to compensate the speckle phenomenon, instead of exploiting it, it is clear that its feasibility implies a predictable and deterministic (at least in part) behavior of the MMFs.

These considerations motivated us to design a new sensing system based on the categorization of the specklegrams at the output of an optical fiber. The aim of this new approach is not only to provide an efficient way to detect the perturbations occurred along the fiber, but also the position where they happened.

In this respect, an interesting application was proposed by Fujiwara *et al.* in [36]. They designed a tactile sensor based on optical fiber specklegram analysis, developed on a 30×30 mm^2 plate. The sensing capabilities are provided placing three MMFs in a parallel fashion and using the inner-product coefficient (see Section 2.2.3) to deter-

mine the similarities between the output speckle and the reference one. Although this configuration allows precise detection of the position where the perturbation (in this case, an applied pressure) occurs, this method is suitable only for such a controlled and size-limited environment.

A similar drawback also affects the method introduced by Plöschner, which was developed for a 300-mm-long fiber segment. Although the author states that light propagation should not be obstructed by significant randomizing processes in longer segments of an optical fiber, another problem interferes in the sensing context. It is important to remember that the FSSs are integral sensors: they provide at the output end a specklegram that is the result of the configuration of the whole fiber. Thus, the output will be affected not only by the primary perturbation which we are interested in, but also by the vast multitude of imperceptible perturbations detected along the fiber, depending on the sensitivity of the sensor (i.e., the number of supported modes). Hence, this contribution is considered as a noisy effect, that is necessary to reduce or, at least, to handle.

An interesting solution for a longer sensor was proposed in [37]. In a traditional multimode interferometer, the system is longitudinally-independent, which means that its properties can be considered independent respect to the position along the fiber. This characteristic hinders the design of spatially-resolved sensing systems. Kotov *et al.* solved this problem varying the number the modes propagated along the fiber. This property can be achieved through the use of either mode controllers or technology inhomogeneities of fiber. Since some output signal parameters, as amplitude and spectrum, significantly depend on the number of propagating modes at the point of perturbation, it is possible to determine the position where the fiber has been perturbed.

Although this configuration provides an effective solution for the proposed purpose, it also has some relevant drawbacks. Firstly, the setup appears quite complex and costly, since it is composed of several mode controllers or segments of different types of fiber. Furthermore, the cost of the system grows as much the desired spatial resolution increase. Finally, the proposed system suffers from losses due to the connections between the different elements through which the light is propagated. The increase of the spatial resolution leads to higher losses here too, which reduce the power transmitted to the detector and limit the sensibility of the system.

To overcome these disadvantages, we propose a solution based on a completely different approach. Our system does not rely on the longitudinal differentiation of the propagation medium, rather on the implementation of more advanced image analysis techniques. As already proved by Efendioglu in [7], ANNs can be usefully exploited to extract the meaningful features coded into the specklegrams at the output of the optical fiber. Their ability to build nonlinear models through automatic learning algorithms based on the analysis of large training sets results particularly useful in this context.

Assuming to use an optical fiber able to support a great number of modes, that is very high-order modes, the image detected at the output end of the fiber will consist of a large number of tiny speckles. Thus, the set of the possible speckle patterns is sufficiently large in order to permit a different configuration of the modes for each position of perturbation. Our idea is to build a model able to categorize the input speckle patterns in relation to the position where the perturbation occurred. In order to do this, we exploit the advantages of the machine learning techniques, training different types of neural networks with very

large datasets of specklegrams. The resulting model has been tested using the same system configuration and providing to the network new inputs.

The results of our analysis will be presented in Chapter 5, while in the next sections we will report the structure of the system used to train and test the model. Furthermore, in Section 4.4 it is presented the structures of the ANNs used for our analysis.

4.2 Setup

In this section, we will describe the setup used to produce the perturbations along the fiber and to record the specklegrams at the output end of the POF. As we will see, all the elements of the system are cost-effective, enabling the possibility to design a competitive real-world sensor, in principle.

Firstly, we will present the types of optical fiber used as sensing elements. Particular importance will be given to the properties and the physical characteristics of these POFs, which lead to different advantages and drawbacks. Then, we will describe the components of our system, highlighting the problem related to the automatic production of perturbations and providing an alternative solution.

4.2.1 Sensing Medium

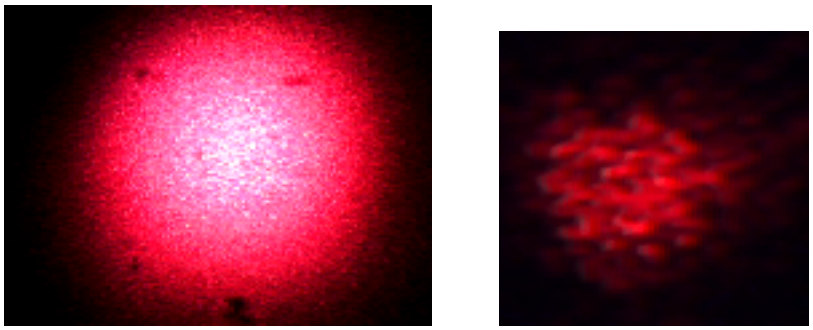
In our experiment, we used polymer optical fibers (POF) as sensing medium. This type of fiber has many of the same advantages of the standard silica optical fiber for sensing applications. However, it constitutes a more cost-effective solution, although with higher transmission losses. This drawback does not affect our experiment, as the length of the fiber will never exceed 12 meters. In any case, this type of fiber can be used up to 100 m, enabling its implementation in improving future applications. In addition, POFs are easier to handle, more robust, and they have high flexibility. All these characteristics are appreciated in our experimental context, and they are usefully used also for real-world applications [38].

In order to test our system, we conducted several experiments with slight modifications of the experimental setup. Probably the most important one is related to the specific type of POF used. In particular, we used a step-index POF, here called 'type 1', and a graded index POF, here called 'type 2'. The main difference between these two kinds of fibers concerns the diameter of the core, as reported in Table 4.1. As already mentioned in Section 2.1.2, this parameter directly affects the normalized frequency, and therefore the number of modes supported by the fiber at certain wavelength λ. In this specific case, we can note that the type-1 fiber has three times the number of modes propagated by type-2 fiber. This data has been calculated at $\lambda = 638$ nm, that is the wavelength of the light used as input. We will report more details about this in the next section.

This distinctive feature between the two types of fiber leads to a different configuration of the specklegram at the output end, illustrated in Figure 4.1. As expected, the dimension of the speckles varies greatly between the two kinds of fiber. The output of type-1 fiber is composed of a great number of tiny speckles, that forms an almost perfect circle, as depicted in Figure 4.1(a). The internal part, brighter than the external one,

Properties of the fibers	Type 1	Type 2
Refractive index profile	Step Index	Graded Index
Numerical aperture	0.5	0.19
Core diameter (μm)	240	50
Normalized frequency at $\lambda = 638\ nm$	1181.8	93.56
Number of modes at $\lambda = 638\ nm$	6983	2188

Table 4.1: Characteristics of the tested optical fibers.



(a) Type 1 (b) Type 2

Figure 4.1: Example of specklegram at the output of the two different types of fibers: (a) type 1, (b) type 2.

mainly contains the lower-order modes, while the external one contains the interactions between the higher-order modes. Despite this last part can be considered as the richest of information, it is important to note that it is affected by a great amount of noise. For this reason, we considered the whole speckle pattern in our analysis.

On the other hand, the output of the type-2 fiber is composed of a lower amount of speckles, which appear to have greater diameter respect of those related to type-1 fiber. As we can see in Figure 4.1(b), in this case the circle produced by the speckles is not so well defined, leading to a grainy figure at the output. An interesting advantage of this type of fiber concerns the lower amount of noise present at the output; on the other hand, the sensitivity is significantly reduced.

Finally, the different brightness between Figure 4.1(a) and 4.1(b) is not necessarily due to the type of fiber. The reason is mainly related to the use of a different set of detector's parameters to record the specklegrams at the output of the fibers.

4.2.2 Perturbation and Recording System

As already mentioned, the system able to categorize the specklegrams at the output end of the optical fiber has been built using different types of neural networks, which have been trained on very large datasets through supervised learning algorithms. In order to produce these datasets, we designed a system able to perturb the fiber in different positions.

Having regard to the complexity of the classification problem that the system has to deal with, we needed sets composed of thousands of different specklegrams, in order to

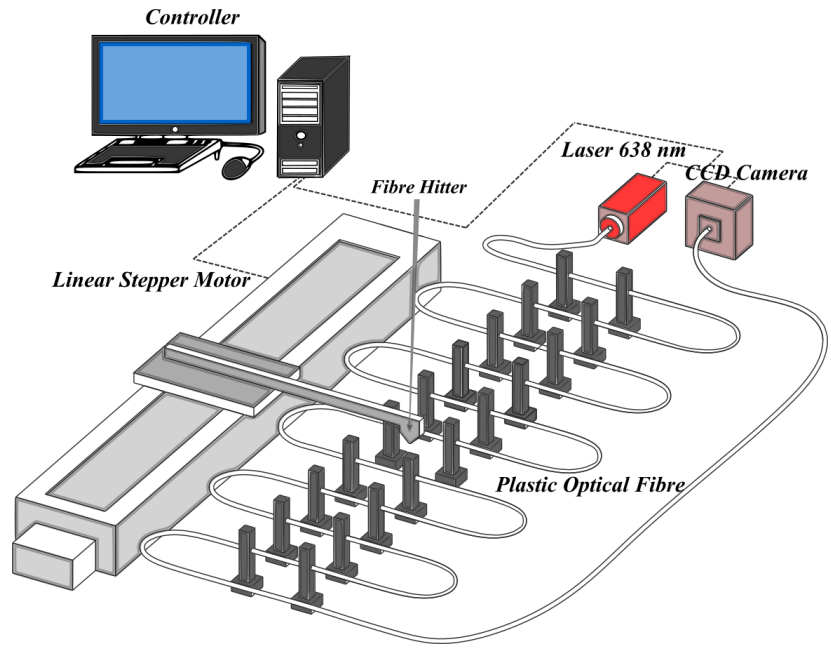


Figure 4.2: Schematic of the experimental setup used to automatically produce perturbations along ten different points of the fiber. The specklegrams are recorded by the CCD sensor the output end of the POF.

obtain a better fit between the data and the model. For this reason, we implemented an automatic system that hits the fiber in predetermine positions, record the speckle pattern and assign a label relative to the position where the perturbation occurred.

We realized two different version of the system: one consisting of three points of perturbation, called System 1, and a second one consisting of ten points of perturbation, called System 2. In the first case, we used a 6-meters long POF, while in the second one we used a 12-meters long fiber. The other characteristics of the system, as the light source, the mechanism of perturbation, and the speckle detector are common to both systems. A schematic of System 2 is illustrated in Figure 4.2, while in Figure 4.3 and 4.4 is depicted the real system from different point of views.

As can be seen from the figures, the POF passes through a series of supports, which have a dual task. The first one is to enable the correct interaction between the POF itself and the fiber hitter, i.e. the fiber has to be 'touched' in predetermined positions and with a limited amount of force. The second task is related to the stability of the system. In this context, it is important that the fiber assumes almost the same position after every perturbation. More specifically, each piece of POF related to a perturbation position has to be in a configuration straight and parallel to the ground. In this way, the fiber hitter perturbs the POF in the same way in all the positions. If it is not, the position is no longer the only distinctive feature between the recorded specklegrams, and this can lead the machine learning algorithms to an improper detection of the speckle pattern's features. In order to avoid this problem, the fiber has been fixed to the supports with some adhesive tape. It permits a flexible movement of the POF during the perturbation, and it brings the fiber back to the initial position during the unperturbed phase.

A very important element of our system is the fiber hitter, colored red in Figure 4.3 and 4.4. It is the element that directly perturbs the POF, and it has been carefully

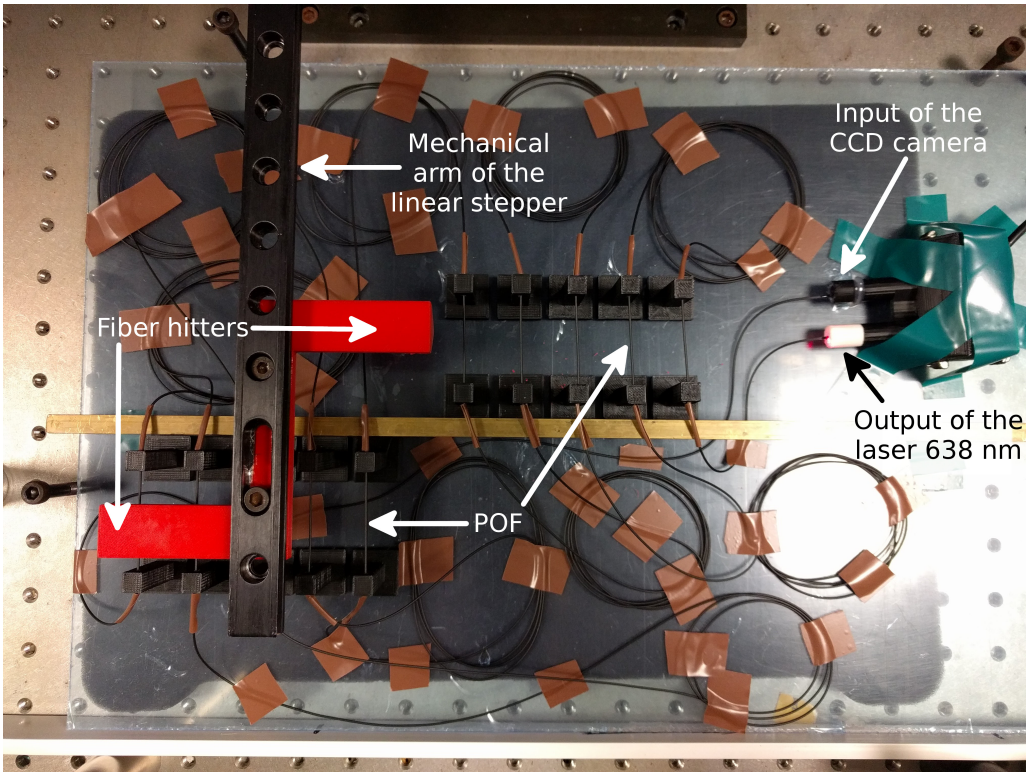


Figure 4.3: Setup of the experimental system seen from above. The shape of the fiber hitter is slightly different respect to Figure 4.2, in order to provide a faster data acquisition.

designed in order to induce a proper vibration of the fiber. In fact, since this FSS provides a great sensitivity, it is important to produce a limited perturbation, in order avoid the saturation of the signal detected at the output. Further details about this aspect will be reported in the next section. In addition, if the perturbation is too intense, the stability of the system can be compromised, leading to the recording of a faulty dataset.

The fiber hitter is connected to the linear stepper motor, which in turn is controlled by an advanced motion driver. For our experiments, we used a Newport MM4005 driver, programmed through Matlab®Instrument Control Toolbox. This device allowed us to drive the stepper motor at a constant and prefixed speed, producing a very long sequence of identical movements along the fiber.

The last two elements of our systems are the light source, and the detector at the output end of the fiber. The first one consists of a low-cost laser emitting at $\lambda = 638$ nm (red light), while the detection of the specklegrams is provided by an economic CCD sensor, derived from a digital camera. An important aspect concerning the CCD sensor is the calibration, which needs to be performed whenever the POF is replaced. In fact, the circle formed by the speckle at the output of the fiber needs to be centered respect to the CCD sensor, in order to detect the whole speckle pattern, i.e. the interactions between all the propagated modes. Furthermore, the movements produced by the stepper motor lead to a misalignment of the specklegram. Then, it is not possible to record too large datasets, since a periodic calibration is needed to avoid faulty data.

In addition, it is important to determine an optimal CCD's parameters set, as well as choose a proper distance between the optical fiber and the sensor itself. For example,

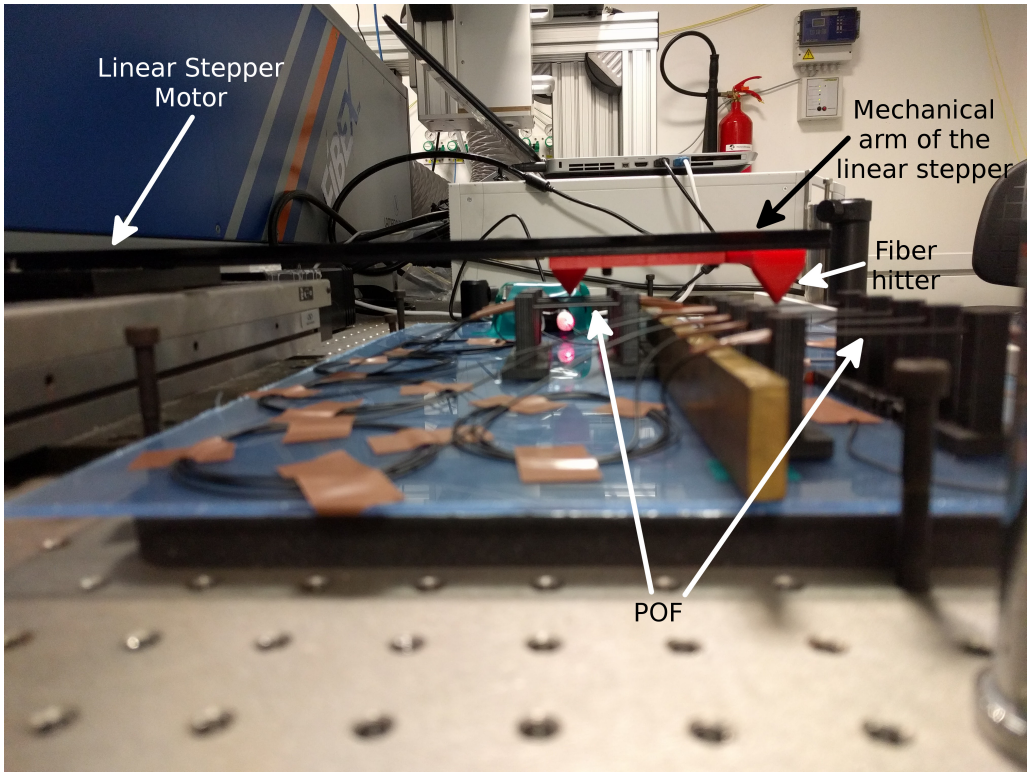


Figure 4.4: Setup of the experimental system seen from the ground. The shape of the fiber hitter is slightly different respect to Figure 4.2, in order to provide a faster data acquisition.

the brightness has to be adjusted in order to avoid the saturation of the inner part of the specklegram, which would prevent the detection of the changes of the low-order modes' configuration. At the same time, the contrast has to be set so that the sparse speckles into the external part of the specklegram can be easily detected. Thus, this optimization is essential to obtain a well-defined and high-contrast record of the speckle patterns, suitable to be correctly analyzed by the machine learning algorithms.

Despite the notable results obtained with this system, which will be reported in the next chapter, it is clear that these are relative to a very controlled environment. That is, the model provided by the machine learning algorithms trained on the specklegrams supplied by the system lacks of generalization. This means that its performances are brilliant if we try to classify speckle patterns related to perturbations produced by our system. However, if we attempt to categorize the perturbations produced in another way, the results appear to be poor. Thus, the supposed equality between the perturbations over the time prevents the correct identification of the feature set suitable to detect the position where the perturbation occurred.

Then, taking into account these considerations, we implement a procedure able to overcome this problem. In order to produce more various perturbations, we used the same basic setup reported in Figure 4.2, but we avoided using the linear stepper to hit the fiber. Instead, we manually induced the perturbations 'touching' the fiber in different ways with the fingertip, or using short sticks. This procedure allowed us to create some manual datasets, where each perturbation can be considered unique and different from the others.

4.3 Acquisition and Processing of the Specklegrams

In this section, we will describe how the data acquired by the CCD sensor at the output end of the optical fiber has been processed, in order to obtain a proper representation of the induced perturbations. The data expressed in this form will constitute the input of the ANNs both for the training and the testing phases.

As we have seen in the previous section, the movement provided by the linear stepper motor produces the collisions between the fiber hitter and the optical fiber. At the output end of the POF, the CCD sensor is able to record the specklegrams with a rate of 30 RGB frames per second. We designed the system in order to obtain a sufficient definition of the evolution of the perturbation by the CCD sensor. The same conditions were applied during the recording of the manual datasets.

As mentioned in the previous section, the system needs to be calibrated after a certain amount of perturbations. We will define a 'dataset' as the collection of the data derived by the specklegrams acquired under a certain calibration session. The distinction between different datasets is due to the dependence of the results by the data used to train or to test the system. This aspect will be discussed in more detail in the next chapter.

The processing phase starts when the whole dataset has been completely recorded. It consists of three processes: compression of information, normalization, and classification.

In the first process, the single perturbation is represented by a sequence of 9 or 18 frames, depending on the type of POF used. In particular, the type-1 fiber showed a longer duration of the perturbations, due to its higher sensitivity to the vibrations that persist on the POF after the hit. We intend to compress of information provided by these frames in a unique one, which will proceed to the normalization process. An overview of the entire compressing process is provided in Figure 4.7. In order to better explain this process, it will be divided into four steps:

Differential processing In this step, the videos belonging to a dataset are converted to grayscale and considered frame by frame. Then, in order to detect the variations in the modes' configuration, a differential process is applied, similarly to the procedure reported in Section 2.3.1.

Let's define F^k as the kth frame, $k \in \{1, \ldots, K\}$, in the video sequence, where each frame can be seen as a $N \times M$ matrix of pixels. The value of a single pixel belonging to a specific frame is denoted as $F^k_{n,m}$, where $n \in \{1, \ldots, N\}$ and $m \in \{1, \ldots, M\}$. Then, the differential process is defined as the computation of $K - 1$ frames, denoted as D^k, each one composed by $N \times M$ pixels $D^k_{n,m}$, which are defined as:

$$D^k_{n,m} = \left| F^{k+1}_{n,m} - F^k_{n,m} \right| \tag{4.1}$$

The result of the differential process is a video sequence in which each frame represents the pixel-wise variation respect the next one in the sequence. If $D^k_{n,m}$ is related to an output position where the intensity of the light remained constant, it will be a dark pixel. Otherwise, it will as much white as the intensity varied in that position. An example of the result provided by this step is reported in Figure 4.7 (b).

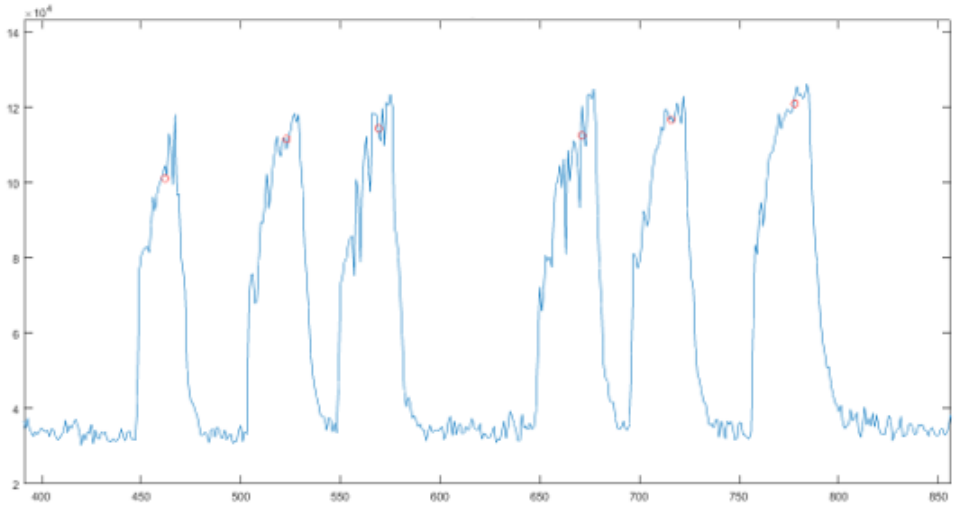


Figure 4.5: Example of differential signal obtained perturbing a type-1 fiber through the linear stepper motor. The red circles represent the detected center of the perturbation.

Peak detection Since each video is composed of a large number of frames that are not related to the perturbations occurred along the fiber, then we can consider deleting those. In this way, we can better exploit our storage space, allowing the record of a greater amount of videos. Furthermore, this step is fundamental to provide correct labeling of the speckle patterns.

The precondition to properly execute this operation is to know the exact position of the perturbation into the video. This process aims to pursue this goal.

In order to perform the peak detection, it is necessary to extract from the differential video D the amount of changes that occurred in each instant of time k. This operation is equivalent to sum the values of all the pixels belonging to a differential frame D^k, that is

$$P(k) = \sum_{n=1,m=1}^{N,M} D^k_{n,m} \tag{4.2}$$

where $P(k)$, $k \in \{1, \dots, K-1\}$, is the one-dimensional differential signal. This type of signal is especially useful to visualize the general characteristics of the perturbations, as the duration and the intensity. Furthermore, it permits to verify if the fiber has been perturbed by agents external to the system and to determine the amount of noise that affects the measurements.

An example of a differential signal is reported in Figure 4.5, where we can identify six perturbations. As we can see, the noise level does not affect the detection of the perturbations significantly. However, it is not completely negligible, since it differs by less than one order of magnitude respect to the point of maximum perturbation. The high noise level is one of the aspects that will have to be managed by the artificial neural networks.

Once calculated the differential signal, it can be used for automatic peak detection. This operation is not always straightforward because the perturbations can in some

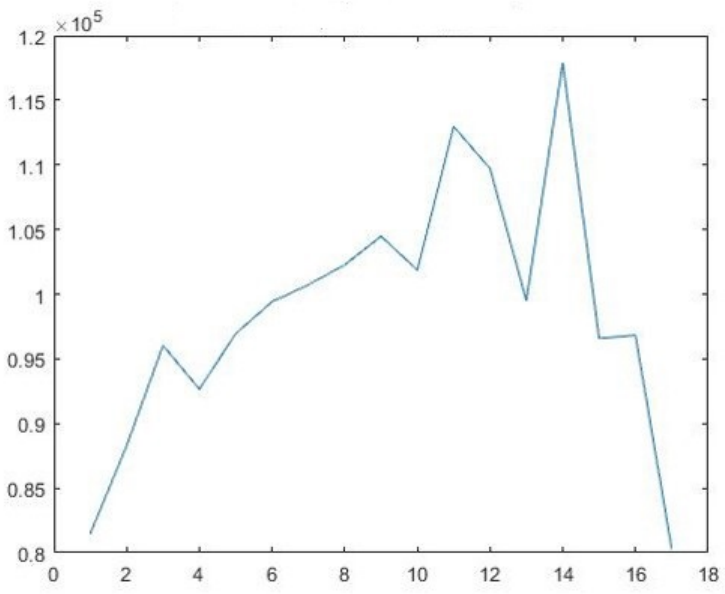


Figure 4.6: Differential signal of a single perturbation, obtained perturbing a type-1 fiber through the linear stepper motor.

cases be 'messy', that is they can be composed of several smaller peaks, instead of a larger single one. For example, this depends on the type of interaction between the fiber hitter and the POF. If the fiber slips uniformly on the fiber hitter, the perturbation is represented by a single peak. Otherwise, if the movement of the fiber is jerky, the related peak will be fragmented into smaller ones in the differential signal. This last case is frequent for the perturbation produced manually.

Despite these difficulties, it is almost always possible to detect the peak position into the video sequence, through the use of some numerical calculations. Figure 4.5 provides an example of the result of these computations, where the red circles represent the detected positions of the peaks. As we can see, the reference position of each peak is approximately associated with the central frame of each perturbation. This detail will become fundamental in the next step.

Partitioning of the video At this point, we know the positions of all the perturbations (i.e. the peaks) in the differential video. Thus, we can isolate the differential frames relative to each perturbation. Denoting as $t_i \in \{1, \dots, K-1\}$ the reference position in time of the ith peak in the differential video, the frames containing the evolution of the ith perturbation are those belonging to the reference set $T_i = \{D^{t_i-(w-1)/2}, \dots, D^{t_i}, \dots, D^{t_i+(w-1)/2}\}$, where w is the width of peaks, supposed odd and constant over time. The frames that do not belong to a reference set T_i can be discarded after this step.

Sum of the frames In the fourth and final step, we compress the amount of data related to a single perturbation. This operation is necessary for two reasons. Firstly, it allows saving more storage space, enabling the recording of a greater number of videos. Secondly, it is difficult to manage the information provided by the whole evolution of the perturbation. Furthermore, the analysis of such amount of data

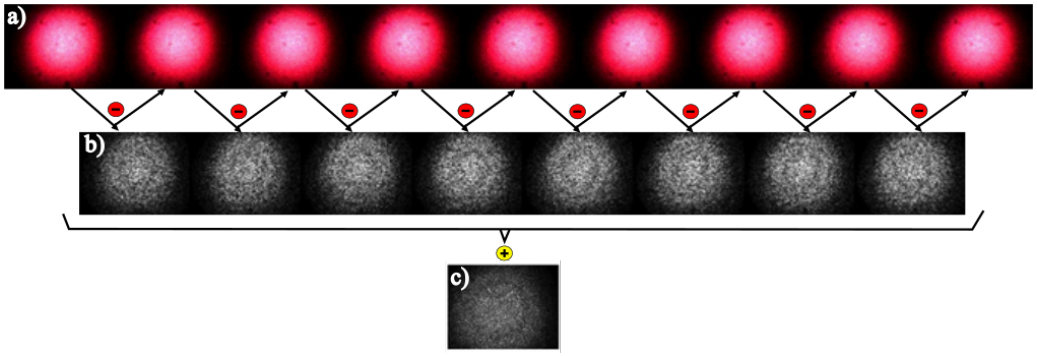


Figure 4.7: Representation of the steps of the compressing process. Figure (a) shows a sequence of frames related to a perturbation induced on a type-1 fiber. Figure (b) represents the differential video, where each frame is computed as defined in Equation (4.1). Figure (c) shows the corresponding reference image of the analyzed perturbation.

requires a much more complex ANN, in order to fully exploit all the possible relations between the pixels in time. This means that it is necessary to train the network over a much more larger dataset to obtain a suitable model. All these considerations lead to an increase of time and cost, going beyond the purposes of our work.

In order to compress the data related to the ith perturbation, we sum all the frames belonging to the reference set T_i as a function of time. The result will be an image S^i of dimensions $N \times M$, which summaries all the information related to the ith perturbation. We can express the values of the pixels of S^i as:

$$S^i_{n,m} = \sum_{D \in T_i} D_{n,m} \tag{4.3}$$

where $S^i_{n,m}$, called reference image, represent the value of the pixel located in position (n, m), $n \in \{1, \ldots, N\}$, $m \in \{1, \ldots, M\}$, in the resulting image. An example of reference image is reported in Figure 4.7 (b).

Thus, at the end of the compression process, we own a set of images S^i, which constitute our dataset. The next phase concern the normalization of the actual dataset. This procedure aims to ensure that the classification is based solely on the variations of the mode's configuration. For example, if two perturbations are induced in the same position along the fiber, but with a different intensity, the strongest perturbation will be related to a brighter reference image. Then, since we want to classify the perturbations only in relation to the position, the images need to be normalized.

Furthermore, if the perturbations related to a specific position are systematically more intense than the others, this characteristic would constitute a distinctive feature among the speckle pattern categories. Thus, the machine learning algorithms could detect it to distinguish between the specklegrams, leading to the improper identification of the speckle pattern's features.

In order to define this procedure, we can denote as R^i the result of the normalization process applied to the reference image S^i. This operation consists of dividing each pixel

of S^i by the sum of all the pixel's values in the image. Mathematically:

$$R^i = \frac{S^i}{\sum_{n=1,m=1}^{N,M} S^i_{n,m}}. \tag{4.4}$$

When this process is concluded, we can move to the final one, i.e. the labeling. This operation aims to divide the dataset of normalized images R^i according to the position where the perturbations were induced along the fiber. In this way, the data acquired and processed through these phases can be used to train the ANNs, or to test the accuracy of the model obtained.

The labeling can be executed simply storing the normalized images R^i in the same order of acquisition. Since the interactions between the POF and the fiber hitter are fully predictable and periodic, the labeling is a straightforward operation. However, in order to execute correct training of the ANNs, it is important to generate an equally divided dataset, i.e. each category has to contain the same number of images. If it is not, the performances of the supervised learning algorithm are suboptimal, leading to lower accuracy of the obtained model.

4.4 Structure and Training of the ANNs

Once obtained one or more normalized and labeled datasets, it is possible to proceed with the creation of the model for the classification pattern task. As already said, this operation is performed automatically using supervised machine learning techniques. In particular, we used two different types of neural networks, exploiting the tools provided by Matlab®R2016a.

4.4.1 Feedforward Neural Network

The first analysis was realized training a feedforward neural network (see Section 3.2.4). This ANN consisted of two layers of neurons, plus the initial input layer, as depicted in Figure 4.7. The number of neurons belonging to each layer varied depending on different conditions.

Firstly, the number of inputs was equal to the number of pixels contained into each image R^i; as can be seen from Figure 4.7, each input variable is associated to a pixel of the normalized image. The number of pixels that make up an input image depended on the actual dimensions of the specklegram on the CCD sensor, i.e. on the type of POF and on the distance between the fiber and the sensor.

On the other hand, the number of output neurons is equal to the number of categories that we aim to characterize, that is the number of positions where the fiber can be perturbed. Thus, this parameter expresses the spatial resolution wanted from the model, that was equal to three or ten in our case.

Finally, the hidden layer, which connects the inputs to the output neurons, had a dimension between 10 and 20 neurons. The exact value of this parameter was determined for each training dataset, in order to obtain the best model possible, i.e. the one which provides the higher accuracy. Anyhow, the results showed that the dimension of the

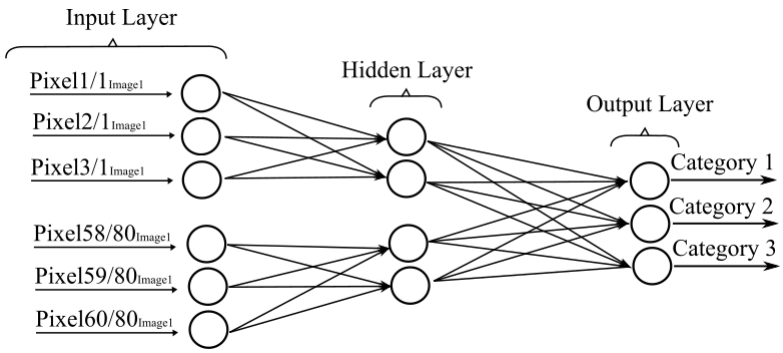


Figure 4.8: Structure of the ANN used in the experiments for the classification of three position of perturbation along the fiber.

hidden layer affects only marginally the performance of the system. For our purpose, the implementation of this kind of ANN with approximately 12 neurons leads to a fast training phase and a good accuracy.

To train the network described above, we used the scaled conjugate gradient back-propagation algorithm provided by Matlab®Neural Network Toolbox. This training procedure was proposed by Moller [33], as already mentioned in Section 3.3.3. In this case, the corrections of the weights of the network are based on the conjugate directions, which decrease the complexity of the algorithm and provide better convergence capabilities.

The optimization of the weights continue until any of these conditions is met:

- The entire training set is used as input for a maximum number of times;
- The execution of the algorithm requires more than a certain time;
- The network obtained from the training procedure yields performances below a fixed goal;
- The performance provided by the updating of the weights decreases below a specified value;
- The performance provided using only the validating set increases more that a certain value respect to the last time it decreased.

All the values used in the conditions above can be set depending on the needs of the specific application. In our case, we run the learning algorithm using the default values.

After the training set (i.e., when the BP algorithm stops), the network is ready to be tested on whatever dataset or single perturbation. The output neuron which is associated with the maximum output value indicates the corresponding categorization of the given input. The test of the network on a dataset different from the training one permits to determine the level of generalization achieved during the training phase, that is the quality of the model obtained.

4.4.2 Convolutional Neural Network

The second type of analysis regards the use of a convolutional neural network (see Section 3.2.5), which structure is reported in Figure 4.9. As we can see, the network

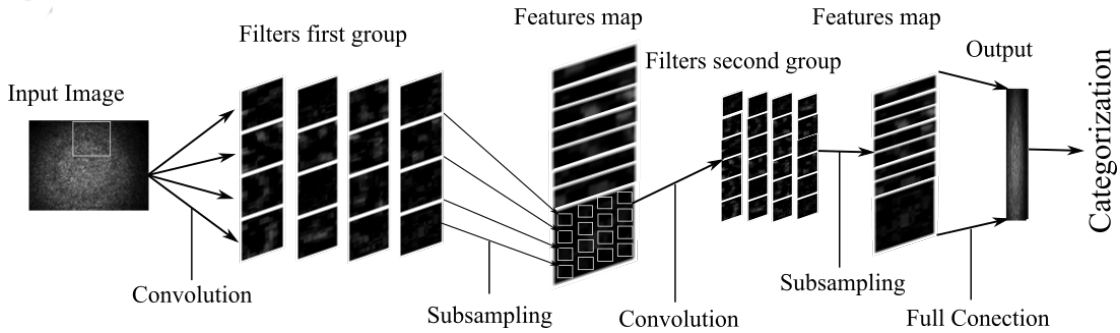


Figure 4.9: Structure of the CNN used in the experiments.

is composed of two convolutional layers, both followed by a subsampling layer. Also in this case, the implementation of such a structure has been realized and trained using the Matlab®Neural Network Toolbox.

Differently from the ANN's case, the input of our CNN it is not considered just as a set of variables. On the contrary, it is recognized as a grayscale image, where each pixel expresses an intensity value related to a specific position. This approach enables the application of the local receptive fields from the convolutional layer.

More specifically, this layer is composed of ten feature maps, also called *filters*, by analogy with the term used in the convolution context. Each of these maps defines a set of weights, that is shared over all the regions of the image. In the first convolutional layer, the local field related to each unit is defined by a window with a height and width of 15 inputs.

The output of the convolutional layer is analyzed by a "Rectifier Linear Unit" (ReLU), that is simply an activation function which performs a threshold operation to each element, without changing the number of them. In this type of network, the threshold value is set to zero, that is

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{4.5}$$

where x is the value of the considered unit, and the $f(x)$ is the value related to that unit at the output of the ReLU layer.

At this point, the network down-sample the representation of the data coming out from the ReLU layer. In order to do this, a max pooling layer is used, which permits to reduce the number of parameters by dividing the input into non-overlapping subregions and taking the maximum element to represent the entire subregion at the output. In our implementation, we set the size of the subregions of the first max pooling layer at 3×3 units.

Now, the matrix built in this way pass through another set of the layers described above, characterized by a different set of dimensions and parameters. For example, in the second convolutional layer, the height and width of the local receptive fields decrease to 5, while the number of feature maps increases up to 16. In addition, also the subregions of the second max pooling layer become smaller respect to the first one; in our implementation, we set a 2×2 size units.

Then, in the final part of the network, the data obtained through all the previous layers becomes the input of a fully connected layer, which expresses an output of 256

units. After another ReLU layer, the data is downsized using a new fully connected layer, which has a number of outputs equal to the number of categories that we aim to characterize, i.e. three or ten.

Finally, the very last operation that has to be executed on the data concerns the use of the softmax function, that can be considered as the multiclass generalization of the logistic sigmoid function (see Section 3.2.2). In the end, the data obtained conveys the class scores in the classification layer.

The training of the network was performed using the tools available in Matlab®R2016a. In particular, we exploited an algorithm based on stochastic gradient descent, as presented in Section 3.3.3. The trained network obtained with this procedure provided the classification of a given input assigning the maximum of the values at the output layer to the corresponding output neuron.

5

Experimental tests

In this chapter are reported the tests carried out in order to determine the accuracy provided by the trained networks. In particular, we will highlight the conditions that affect the results. As we will see, besides the dependency on the type of fiber used and the neural network employed as sensing medium, an important role is done by the training and testing datasets. In general, if these are produced through the same procedure, the accuracy results higher, since the higher degree of similarity enables a better match between the features discovered and those present in the inputs. Anyhow, the accuracy provided by the manual datasets indicates a good level of generalization of the model design by the learning algorithms.

5.1 Preliminary settings

At the beginning of our study, we tried to understand what were the best configuration for the system reported in Section 4.2.2. One of the main aspects was related to the choice of a suitable optical fiber, and therefore to the number of modes that it should have been supported by it. Initially, we decided to use the type-1 fiber, because it provides a greater sensibility and it is covered by a jacket. This detail allows a slower impairment of the POF due to the interaction with fiber hitter, enabling the recording of larger datasets. In addition, this type of fiber resulted in the best choice for vibration detection, as demonstrated in previous works [11] [15] [16]. Therefore, this work enables the release new functionality for these sensors, in order to allow further applications.

Secondly, we implemented our system using type-2 fiber. This choice was made to test the performances of the proposed configuration with a POF that express a lower noise level. In fact, the specklegrams at the output of type-1 fiber were continually changing, even if the POF was not perturbed by the system. Although these movements principally concerned only the higher-order modes, and a perturbation was detectable to the naked eye, this high noise level could affect the performance of the system. Moreover, the output specklegrams of the type-2 fiber can be considered more discernible between each other, because they are composed of larger speckles (see Figure 4.1). Furthermore, they are numerically fewer than those in type-1 fiber's speckles.

Another parameter that we tested at the beginning of our study concerned to the resolution of the video recording. Initially, we used the maximum resolution possible, that was 640×480 pixels per frame (the typical resolution of an economical webcam). The reason was related to the dimension of the speckles in type-1 fiber output. We believed that, in order to detect them, was necessary to fully exploit the CCD sensor's capabilities; otherwise the image's pixels would have represented the mean of the speckles

in a specific region.

This approach created two practical problems: the videos were difficult to manage, due to their considerable size, and the training phase of the ANNs required a non-negligible amount of time. Thus, we decided to decrease the resolution of the CCD sensor to 80×60 pixels per frame. Unexpectedly, the results improved significantly, probably because this operation reduced the speckle's changes due to the noise.

Other settings analyzed in our preliminary tests concerned the distance between two perturbed points, that was fixed at two meters, and the size of each dataset, which mainly depended on the perturbation method, which could be automatic or manual. The acquisition of a dataset was often complicated by technical problems, as misalignment of the specklegram respect to the CCD sensor, or the entanglement of the POF in the fiber hitter. For these reasons, the size of the datasets was not constant, and it was sometimes necessary to delete or merge some of these in order to properly train the networks.

Finally, we also tested alternative algorithms to extract the most important features from the speckle patterns. For example, we used a Sequential Floating Forward Selection (SFFS) algorithm to determine the correlation between the pixels located in different positions in the images. Another test was performed using the Matlab®'s implementation of an autoencoder neural network, provided into the Neural Network Toolbox. Both these techniques aim to provide a compressed representation of the input data through the extraction of the main features coded into the analyzed data. Despite the interesting approach to the problem, both these methods did not provide significant results.

Then, in the end, we design two experiments based on the ANN and the CNN reported in the previous chapter. In Figure 5.1 are illustrated the preliminary results obtained with the settings described above, and using a limited dataset acquired automatically. They are expressed through confusion matrices, that allow a clear overview on the model's behavior. In particular, the columns represent the category to which the specklegrams belong, while the rows show the classification provided by the network. We aim to obtain results as close as possible to 100% on the diagonal, highlighted in green, that implies a correct detection of the position of perturbation. The items out of the diagonal express the misclassifications accomplished by the model, and are marked in red. As we can see, these results are encouraging in most cases, especially those related to the type-2 fiber.

Based on these outcomes, we decided to train and test the networks using larger datasets; furthermore, we checked the performance concerning the classification of perturbations produced manually, in order to verify the level of generalization achievable by the system.

5.2 Experiments

After the preliminary tests, we conducted two main experiments to determine the performances of our system. These tests were executed through two main experiments. In the first one, the system was set to induce perturbations in three different positions; in the second one, we incremented the number of perturbed point up to ten. In the next sections, we will report the results concerning these two system's configurations.

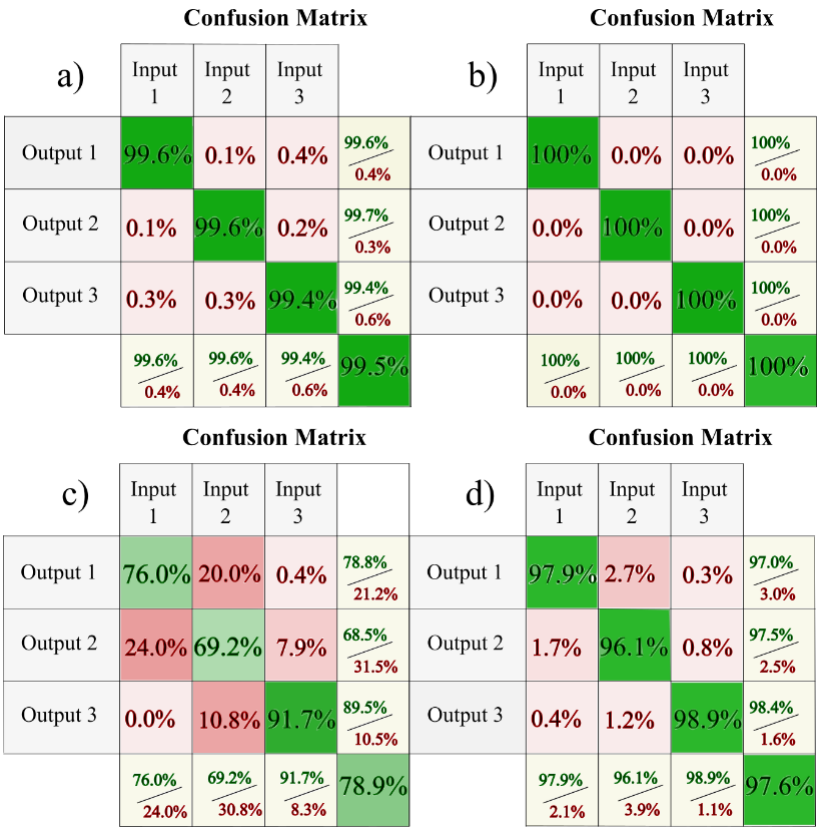


Figure 5.1: Preliminary results obtained after the preprocessing, expressed by confusion matrices. The data refers to (a) type-2 fiber using CNN, (b) type-2 fiber using ANN, (c) type-1 fiber using CNN, and (d) type-1 fiber using ANN.

5.2.1 Experiment 1

As we previously mentioned in this chapter, the performances of the machine learning algorithms are affected by the datasets used to train and test the networks. In our analysis, firstly we employed the images belonging to the same dataset, in order to accomplish both these operations. That is, we randomly divided the whole of specklegrams into two disjointed sets, taking care to allocate the same number of images for each category in both ones. In this case, we assign 75% of the specklegrams to the training set, and the remaining 25% of the dataset to the testing one. The results are reported in Table 5.1 and show a satisfactory accuracy in almost all the cases.

We can start our analysis from the most controlled situation, that is the first one. In this case, the classification is nearly perfect using both the tested types of neural network. Moreover, we also obtained a notable accuracy employing the manual dataset; this result indicates that the model does not categorize the specklegrams based on the particular type of perturbation, because they are completely different between each other, in this specific case. Instead, the machine learning algorithms successfully extracted and analyzed the most characterising features related to the positions, enabling an accuracy slightly lower than 80%.

As we can see, the performance of ANN and CNN are quite similar in this case, and the first technique yields slightly better results, in particular. This fact can be unexpected, given the complexity of the convolutional neural network used to execute

the experiment. On the other hand, it is important to note that CNNs were design to deal with a very different kind of images, composed of shapes and objects. Thus, the process of subsampling used to summarize the features and obtain the high-order characteristics of the specklegrams can be deleterious, leading to an inappropriate representation of the input data.

In addition, the local receptive fields mechanism can be useful to detect the speckle pattern's features independently on the exact position of the whole specklegram respect to the CCD sensor. In principle, this characteristic should enable a less-frequent calibration of the system, being able to manage the displacement of the optical fiber due to the induced perturbations over time. On the other hand, this property can lead to detecting a particular shape produced by the interaction of some modes independently on other features' position inside the speckle pattern. This behavior appears to be a loss of information, which reduces the accuracy of the model.

Finally, we can compare the accuracy provided by the two different kinds of POFs used in the experiment. Table 5.1 shows results mostly similar between the tests conducted on manual datasets; on the other side, perturbations induced automatically by the linear stepper yield higher accuracy using type-2 fiber. Furthermore, type-1 POF exhibits almost the same performance among the two different perturbing methods. Since the conditions provided by the automatic perturbations are quite different respect to those related to the manual ones, we can speculate that the neural networks are not able to adequately analyze the information provided by the speckle patterns.

This limitation can be due to several reasons, which can jointly play a part in this. For example, the higher complexity of the specklegrams, which is the result of the larger number of modes supported by type-1 fiber, could be too high for this type of analysis. Another possible reason can be related to the resolution of the CCD sensor used in the experiment, which may be not able to detect a sufficiently-accurate representation of the modes' configuration at the output end of the fiber. Finally, the two different kinds of POFs are characterized by a very different level of noise, which can affect the performances also under the best conditions provided by automatic perturbations.

Fiber	Training Dataset	Testing Dataset	ANN's Accuracy	CNN's Accuracy
Type 1	43,020 [L]	14,340 [L]	79.1%	80.1%
	14,550 [M]	4,850 [M]	78.7%	72.3%
Type 2	112,500 [L]	37,500 [L]	99.8%	99.7%
	17,050 [M]	5,680 [M]	79.1%	78.3%

Table 5.1: Results of the overall accuracy when using parts of the same datasets for training and for testing. The columns 'Train Data' and 'Test Data' report the number of perturbation used in the respective datasets. The notes next to these numbers indicate how the perturbations were induced: using the linear stepper motor [L] or manually [M].

In Table 5.2 are reported the performance of the same system configuration, but we introduced a substantial difference respect to the results analyzed so far. In this case, we used different datasets for the training and the testing phases, keeping unchanged

the number of specklegrams belonging to each dataset respect to Table 5.1.

As we can note, the performances are lower, in general. The type-2 fiber perturbed automatically by the linear stepper provides acceptable results, while in the manual case the accuracy shows insufficient classification capabilities. For this specific task, type-1 POF based on ANN analysis yields good results, even if lower than the correspondent case in Table 5.1.

This observation indicates a strong dependency of the model provided by the neural networks to the specific state of the system. We tried to mitigate this effect applying some transformations to the speckle patterns, before using them in the training phase. In particular, we detected the center of the specklegrams belonging to various datasets and we displaced the non-aligned ones to a common center, in order to obtain calibration as a post-process. Unfortunately, this procedure did not produce substantial results.

Moreover, we tried to enhance the performances of the networks using other techniques. For example, we reduced the noise level that affects the optical fibers, establishing a threshold that a pixel has to exceed to be considered useful. That is, all the pixels below this value were set to zero, excluding them from the analysis process.

Another approach that we investigated was based on the 'artificial' extension of the datasets. In contrast with the post-process centering of the specklegram tested initially, we increased the number of speckle patterns creating a series of similar copies, each one slightly displaced respect the others. This operation aimed to reduce the overfitting by providing a greater number of training images that contain the same information, presented in a somewhat different way. This procedure should have enabled the design of a better model, but our tests did not show appreciable results.

Fiber	Training Dataset	Testing Dataset	ANN's Accuracy	CNN's Accuracy
Type 1	43,020 [L]	14,340 [L]	73.7%	71%
	14,550 [M]	4,850 [M]	75.6%	66.1%
Type 2	112,500 [L]	37,500 [L]	96.9%	90.3%
	17,050 [M]	5,680 [M]	53.4%	48.9%

Table 5.2: Results of the overall accuracy when using different datasets for training and for testing. The columns 'Train Data' and 'Test Data' report the number of perturbation used in the respective datasets. The notes next to these numbers indicate how the perturbations were induced: using the linear stepper motor [L] or manually [M].

5.2.2 Experiment 2

Based on the results obtained in the first experiment, we extended our system in order to test its performance with a more challenging task. In particular, we increased up to ten the number of position where the fiber was perturbed, as represented in Figure 4.2. In this case, we also expanded the whole length of the fiber up to twelve meters, enhancing the spatial resolution required to the model to one meter.

The preprocessing phase was the same used in the previous experiment, as well as the structure of the neural networks. We made this choice because the task required to the

system is substantially identical to the last case, that is based on the same information coded into the images.

In this case, the stability of the system is more difficult to achieve, since the overall fiber is perturbed more times respect to Experiment 1. For example, it is more complicated to ensure that the POF remains in an appropriate configuration over time for all the positions. Furthermore, the highest number of perturbations leads to a quicker displacement of the output end of the fiber respect to the CCD sensor. For these reasons, we decided to use only the type-1 POF, even if it provided lower results in the first experiment. These mechanical properties are due to the physical characteristics of this kind of fiber, which is covered by a jacket that increases the weight and the thickness respect to type-2 POF.

In this case, we decided to evaluate the performances of the model only using images from the same dataset for the training and the testing phases. This choice has been made because we have already demonstrated the high dependency of the results from the specific datasets used. Otherwise, in Experiment 2 we want to focus our attention on the capacity of the system to manage a more significant number of positions. In particular, the primary goal is to determine the amount of misclassifications of such a complicated setup and find out between which categories these problems occur.

In Figure 5.2 and 5.3 are reported the results concerning Experiment 2. The first figure presents the confusion matrix relative to the model obtained through the artificial neural network, while the second figure illustrates the confusion matrix provided by the convolutional neural network. As already seen in the previous sections, in a confusion matrix the columns represent the real categories which the inputs belong, while the rows show the classification provided by the model. It is important to note that the first category concerns the most distant position from the CCD sensor, while the tenth one represents the closest perturbation point to the camera. Finally, the correct classification is identified by the diagonal elements (highlighted in green), while the others represent misclassifications (highlighted in red).

If we compare the two figures, we can note that the model designed using the ANN yields better results respect to that based on CNN, following the performances computed in the previous experiment. As can we see, the speckle patterns related to the closes positions to the CCD sensor are correctly detected in almost all the cases. Otherwise, the perturbations induced at the beginning and in the middle of the fiber are characterized by a lower accuracy, even if it the misclassification is almost always below 33% (in the ANN case).

The deterioration of the performances going through the fiber from its end to the first position can be explained considering the noise levels. The speckle patterns associated with the last category are mostly affected by the noise from the final position to the CCD sensor, i.e. just one meter of POF. Thus, the noise level can be considered low, enabling the correct detection of the relevant feature by the learning algorithm. Otherwise, the data produced by the closest positions to the laser, i.e. at the input of the fiber, will have to deal with a much more long distance, that is a higher amount of noise induced by about ten meters of POF. Then, the specklegrams related to those positions will be corrupted through the fiber, and the information coded into them will result damaged.

In other words, the points of the fiber that are not affected by perturbations should be

Confusion Matrix

	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8	Input 9	Input 10	
Output 1	66.2%	43.1%	23.3%	2.6%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	48.9% / 51.1%
Output 2	22.7%	33.4%	5.9%	4.8%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	49.8% / 50.2%
Output 3	5.1%	1.6%	69.2%	0.0%	0.0%	0.0%	10.2%	0.0%	0.0%	0.0%	80.2% / 19.8%
Output 4	0.0%	20.8%	0.2%	67.8%	4.9%	5.4%	0.0%	0.4%	0.0%	0.0%	64.5% / 35.5%
Output 5	0.2%	0.2%	0.0%	11.9%	75.1%	58.1%	0.0%	5.2%	0.0%	0.0%	49.7% / 50.3%
Output 6	0.2%	0.5%	0.0%	12.6%	12.0%	35.1%	0.0%	0.0%	0.0%	0.0%	58.4% / 41.6%
Output 7	0.0%	0.4%	1.1%	0.2%	0.0%	0.0%	89.7%	0.6%	0.0%	0.0%	97.4% / 2.6%
Output 8	0.0%	0.0%	0.0%	0.2%	8.0%	1.5%	0.0%	82.3%	6.2%	0.0%	83.8% / 16.2%
Output 9	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	11.5%	93.8%	0.0%	89.0% / 11.0%
Output 10	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100%	100% / 0.0%
	66.2% / 33.8%	33.4% / 66.6%	69.2% / 30.8%	67.8% / 32.2%	75.1% / 24.9%	35.1% / 64.9%	89.7% / 10.3%	82.2% / 17.8%	93.8% / 6.2%	100% / 0,0%	71.3%

Figure 5.2: Confusion matrix of the Experiment 2, using ANN.

Confusion Matrix

	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8	Input 9	Input 10	
Output 1	44.9%	35.9%	12.6%	6.3%	0.0%	0.5%	0.0%	0.0%	0.0%	0.0%	44.8% / 55.19%
Output 2	31.2%	38.6%	6.2%	20.4%	1.0%	1.9%	0.6%	0.0%	0.0%	0.0%	38.64% / 61.36%
Output 3	17.3%	10.8%	67.7%	1.2%	0.0%	0.1%	2.7%	0.0%	0.0%	0.0%	67.84% / 32.16%
Output 4	8.7%	16.6%	0.7%	41.9%	13.3%	16.5%	0.6%	1.5%	0.0%	0.0%	42.0% / 58.0%
Output 5	0.1%	0.2%	0.0%	9.3%	45.5%	28.5%	0.0%	16.1%	0.2%	0.6%	44.0% / 56.0%
Output 6	0.6%	1.2%	0.0%	15.0%	30.9%	49.7%	0.0%	2.7%	0.0%	0.0%	49.7% / 50.3%
Output 7	0.8%	1.2%	2.2%	1.8%	3.5%	0.0%	89.8%	2.7%	1.1%	0.0%	87.1% / 12.9%
Output 8	0.0%	0.0%	0.0%	1.8%	12.0%	4.9%	3.3%	61.9%	16.1%	0.0%	61.9% / 38.1%
Output 9	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.4%	11.6%	87.8%	0.2%	87.8% / 12.2%
Output 10	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.1%	0.1%	0.2%	99.4%	99.5% / 0.05%
	43.3% / 56.6%	37.0% / 63.0%	75.7% / 24.3%	42.9% / 57.1%	42.8% / 57.2%	48.3% / 56.6%	92.1% / 7.9%	64.1% / 35.9%	83.3% / 16,6%	99,2% / 0,8%	63.1%

Figure 5.3: Confusion matrix of the Experiment 2, using CNN.

considered just as a medium to transmit the light to the CCD sensor. They should be characterized by an invariant set of parameters, denoting the absence of perturbations. But, in the real world, they are fully-fledged sensing points affected by the dynamics of the environment, which change the set of parameters related to them.

In addition, the lower results regarding the classification of the perturbations occurred at the input of the fiber have been explained in [1]. In this work, Spillman *et al.* discovered that the perturbations in a multimode optical fiber last for a certain length through it. Then, as much as the perturbation is induced in a position far away from the detector, as much as the perturbation will result weakened for it.

However, regarding Figure 5.2, it is important to note that the worst accuracy is not associated to the furthest point from the CCD sensor. Instead, it concerns the second and the sixth points of perturbation. In these cases, it is clear that the system is not able to learn or detect the set of features that allow the distinction between these positions with those closest to them.

We can extend almost all the considerations reported above to Figure 5.3, even if the performances are significantly lower and unsatisfactory.

6

Conclusions

6.1 Overview

In this thesis, we described a system able to turn an FSS into a spatially resolved sensor. That is, our approach enables the automatic classification of the speckle patterns according to the position where the perturbation occurred.

In the first part of this work, we reported the theoretical concepts related to the specklegrams, analyzing the current state of the art. In particular, we presented the main techniques used to detect the perturbations through the speckle patterns, and we described the major implementations of this kind of optical fiber sensor.

In the third chapter, we showed the characteristics of the artificial neural networks, from their biological origins to the most recent approaches. We analyzed the different configurations that can be adopted to build a classification model, exploiting the machine learning algorithms.

Then, in Chapter 4 we described the experimental setup that we used to induce the perturbations along the fiber. Furthermore, we reported the preprocessing method that we applied to train the neural networks and to test their accuracy.

Finally, in Chapter 5, we showed the results of our experiments, highlighting the positive aspects and the possible reasons for misclassifications.

As we have seen, our system provides a high accuracy using data produced automatically, but the performances result lower if the perturbations are induced manually; in particular, the categorization of three points using an ANN yield up to 99.8% and 79.1%, respectively.

However, we observed a high dependency of the results on the specific conditions of the system, leading to a higher amount of misclassifications if the testing data does not belong to the same dataset used to train the network. This is an important aspect that has to be taken into consideration in a real-world implementation.

In addition, we analyzed the performances of the system for a more challenging task concerning the classification of ten perturbation points, within one meter of each other. We obtained a lower accuracy (about 71%) mainly due to two categories, which were mistaken for the closest ones.

This experiment also showed an interesting behavior of our method, related to the distribution of the misclassification through the fiber length. In general, the accuracy is higher for the categories closest to the sensor at the output end of the POF, while it decreases for those on the other side of the fiber. This characteristic can be explained considering the different noise levels and the diffusion of the light through the POF.

Finally, despite the proposed system represents a very basic setup, it allowed us to

prove that this kind of classification is possible, even if it is necessary to develop more advanced techniques in order to produce a real-world sensing system. We will briefly discuss this topic in the next section.

6.2 Future works and possible implementations

One of the main drawbacks of the method described in this thesis is the high dependency of the result to the specific configuration of the sensing system. This problem makes impracticable the realization of a real-world implementation of our solution, since it is impossible to train the network and obtain a sufficiently general classification model.

The primary reason for this drawback is the displacement of the fiber respect to the CCD sensor. We tried to overcome this problem exploiting CNN properties and centering the specklegram to a determined point in the image, but these techniques did not result useful. Another approach that can be tested concerns the use of a different system to fix the output end of the fiber to a specific position of the sensor. For example, the use of silicone foam can stabilize the POF, leading to a more robust setup. Furthermore, it could be used for the entire length of the fiber, ensuring the return to a basic configuration of the POF; on the other hand, this technique strongly reduces the sensitivity of the FSS, which cause a decrease of the general performance of the system.

Another problem that should be addressed to develop a real-world implementation of this system concerns the lower accuracy at the fiber input and the presence of a great number of misclassifications for some perturbed positions. This aspect has to be further investigated, in order to better determine its causes and their specific effects. However, it could be interesting to use two optical fibers in the same configuration jointly, but inverting the position of the laser and the CCD sensor. The information provided by such a system should have a more uniform performance along the fiber length.

Finally, it could be useful to test the proposed approach with different machine learning techniques or focusing the analysis on some specific regions of the specklegrams.

Bibliography

[1] W. Spillman, B. Kline, L. Maurice, P. Fuhr, "Statistical-mode sensor for fiber optic vibration sensing uses," *Appl Opt.*, 1989,28(15):3166-76.

[2] K. Pan, C.-M. Uang, F. Cheng, F. T. S. Yu, "Multimode fiber sensing by using mean-absolute speckle-intensity variation," *Appl. Opt.*, vol. 33, no. 10, pp. 2095–2098, 1994.

[3] G. Kaiser, "Optical Fiber Communications," chapter 2, *McGraw-Hill*, NY, second edition, 1991.

[4] F. T. S. Yu, M. Wen, S. Yin, C. Uang, "Submicrometer displacement sensing using inner-product multimode fiber speckle fields," *Appl. Opt.*, vol. 32, no. 25, pp. 4685–4689, 1993.

[5] H. S. Efendioglu, "A review of fiber-optic modal modulated sensors: Specklegram and modal power distribution sensing," *IEEE Sensors Journal*, 17(7), 2055-2064, 2017.

[6] H. S. Efendioglu, T. Yildirim, O. Toker, K. Fidanboylu, "New statistical features for the design of fiber optic statistical mode sensors," *Opt. Fiber Technol.*, vol. 19, no. 4, pp. 279–284, Aug. 2013.

[7] H. S. Efendioglu, T. Yildirim, O. Toker, "Advanced image processing and artificial intelligence based approaches to fiber optic statistical mode sensor design," *Proc. SPIE*, vol. 7982, p. 79820T, Apr. 2011.

[8] H. S. Efendioglu, T. Yildirim, O. Toker, K. Fidanboylu, "New statistical features for the design of fiber optic statistical mode sensors," *Opt. Fiber Technol.*, vol. 19, no. 4, pp. 279–284, Aug. 2013.

[9] A. Rodriguez-Cuevas, L. Rodriguez-Cobo, M. Lomer, J. M. Lopez-Higuera, "Safe and private pedestrian detection by a low-cost fiber-optic specklegram," *Optical Fiber Sensors Conference (OFS)*, 2017 25th (pp. 1-4). IEEE, Apr. 2017.

[10] A. Dhall, J. K. Chhabra, N. S. Aulakh, "Intrusion detection system based on speckle pattern analysis," *Experim. Techn.*, vol. 29, no. 1, pp. 25–31, Jan. 2005.

[11] L. Rodriguez Cobo, "Sensores ópticos para estructuras inteligentes," Ph. D. thesis, Departamento de Tecnología Electrónica, Ingeniería de Sistemas y Automatica, Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación, Universidad de Cantabria, Santander, Spain, 2013.

[12] Z. Zhang, F. Ansari, "Fiber-optic laser speckle-intensity crack sensor for embedment in concrete. Sensors and Actuators," *A: Physical*, 126(1):107–111, 2006.

[13] J. S. Leng, A. Asundi, "NDE of smart structures using multimode fiber optic vibration sensor," *NDT&E Int.*, vol. 35, no. 1, pp. 45–51, Jan. 2002.

[14] W. B. Spillman, M. Mayer, J. Bennett, J. Gong, K. E. Meissner, B. Davis, R. O. Claus, A. A. Muelenaer, X. Xu, "A 'smart' bed for non-intrusive monitoring of patient physiological factors," *Meas. Sci. Technol.*, vol. 15, no. 8, pp. 1614–1620, 2004.

[15] M. Lomer, L. Rodriguez-Cobo, P. Revilla, G. Herrero, F. Madruga, J. M. Lopez-Higuera, "Speckle POF sensor for detecting vital signs of patients," *Proc. SPIE*, vol. 9157, p. 91572I, Jun. 2014.

[16] A. Rodriguez-Cuevas, E. R. Pena, L. Rodriguez-Cobo, M. Lomer, J. M. Higuera, "Low-cost fiber specklegram sensor for noncontact continuous patient monitoring," *J Biomed Opt.*, 1;22(3):37001, Mar. 2017.

[17] A. Kamshilin, T. Jaaskelainen, Y. Kulchin, "Adaptive correlation filter for stabilization of interference-fiber-optic sensors," *Applied Physics Letters*, 73 (6) (1998)705–707.

[18] J. Gomez, A. Salazar, "Self-correlation fiber specklegram sensor using volumecharacteristics of speckle patterns," *Optics and Lasers in Engineering*, 49 (3)(2011) 473–480.

[19] L. Rodriguez-Cobo, M. Lomer, A. Cobo, and J.-M. Lopez-Higuera, "Optical fiber strain sensor with extended dynamic range based on specklegrams," *Sens. Actuators A*, Phys., vol. 203, pp. 341–345, Dec. 2013.

[20] L. Rodriguez-Cobo, M. Lomer, J-M. Lopez-Higuera, "Fiber specklegam-multiplexed sensor," *J. Lightwave Technol.*, vol. 33, pp. 2591-2597, Jun. 2015.

[21] K. P. Murphy, "Machine Learning a probabilistic perspective," *Mit press*, Cambridge, 2012.

[22] I. Goodfellow, Y. Bengio, A. Courville, "Deep learning," *Mit press*, Cambridge, MA, 2016.

[23] R. Hecht-Nielsen, "Neurocomputing," *Addison-Wesley*, Reading, MA, 1990.

[24] I.A Basheer, M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of Microbiological Methods*, 43(1):3 - 31, 2000, Neural Computting in Micrbiology.

[25] P. K. Simpson, "Artificial neural systems foundations, paradigms, application and implementations," *New York Pergamon*, 1990.

[26] Y. H. Hu, J. N. Hwang, "Handbook of neural network signal processing," *CRC*, Boca Raton, 2002.

[27] A. K. Jain, J. Mao, K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, 29.3 (1996): 31-44.

[28] S. S. Haykin, "Neural networks a comprehensive foundation," *Prentice Hall*, Upper Saddle River, N.J., 1999.

[29] Y. LeCun, Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, 3361.10 (1995): 1995.

[30] C. M. Bishop, "Pattern recognition and machine learning," *Springer*, New York, 2006.

[31] T. M. Mitchell, "Machine Learning," *McGraw-Hill*, New York, 1997.

[32] E. M. Johansson, E. U. Dowla, D. M. Goodman, "Backpropagation learning for multi-layer feed-forward neural networks using the conjugate gradient method," *International Journal of Neural Systems*, 2(4), 291-302, 1991.

[33] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural networks*, 6.4 (1993): 525-533.

[34] Y. LeCun, L. Botton, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 86.11 (1998): 2278-2324.

[35] M. Plöschner, T. Tyc, T. Čižmár, "Seeing through chaos in multimode fibres," *Nature Photonics*, 9(8):529-35, 2015.

[36] E. Fujiwara, Y. T. Wu, M F. Marques dos Santos, E. A. Schenkel, C. K. Suzuki, "Development of a tactile sensor based on optical fiber specklegram analysis and sensor data fusion technique," *Sensors and Actuators A*, Physical, Volume 263, Pages 677-686, 2017.

[37] O. Kotov, I. Chapalo, "Mode-mode fiber interferometer with impact localization ability," *Proc. SPIE*, 9899, Optical Sensing and Detection IV, 98992J, Apr. 2016.

[38] F. Berghmans, H. Thienpont, "Plastic optical fibers for sensing applications," *Optical Fiber Communication Conference*, Optical Society of America, 2014.