Thesis Abstract

Optical wavemeters and spectrometers are devices that take light as input and are able to recognize the different frequency components in the input signal. These devices have been used in various applications including optical communication and food monitoring. The main issue is that they usually consist of grating structures, which make the devices bulky and expensive. By using nanophotonic technologies, it becomes possible to miniaturize these devices and make it possible to easily integrate them into optical systems. Therefore, there have been some trials using photonic crystals (PhC), but the operation usually relies on precisely controlled resonant structures which are difficult to achieve because of the presence of fabrication errors in such nanophotonic devices that often makes them difficult to put into practical use. Here, we describe a way to overcome this problem by combining light localization (caused by fabrication randomness) with deep learning. We use a simple chirped PhC waveguide (WG) in which light will exhibit different pattern depending on its frequency that will allow us to detect the frequency of the input light by recognizing the associated pattern. This structure also takes advantage of the random localization of light resulting from fabrication errors to increase the resolution of the device. We reconstruct the spectrum by feeding our algorithm with training images of the wavelength sensitive pattern formed by the scattering of light in the structure on which we perform learning. Using this approach, we can obtain a resolution even higher than the wavelength resolution of the fabricated device, which is limited by the resolution of the fabrication technique. By using deep learning, we show that we are not limited by the resolution of fabrication and that we can even use it at our own advantage to increase the resolution of our system.

Chapter 1 briefly introduces this work by describing the context and the objectives of this research.
Chapter 2 explains the theoretical background of photonic crystals and the photonic crystal structure that we are going to use for our design.
Chapter 3 explains the theoretical background regarding deep learning.
Chapter 4 describes the experimental procedure that has been made in order to acquire data.
Chapter 5 focuses on the main part of this work: how can we predict the frequency components of the input signal with deep learning.
Chatper 6 summarizes this work and shows the possible methods for improving the performances of the system.

Contents

Chapter 1

Introduction

1.1 Context

Optical communication was a huge breakthrough in communication technologies. The use of optical fiber as a media to transmit modulated light through the fiber presents undeniable advantages compare to other methods of communications as using copper wires: Low power losses which allows longer distance communication and high bandwith which allows the transmission of a bigger amount of data per unit time. It is therefore natural that this method for communication would be widespread in order to improve communication systems. Given the efficiency of optical communication, its application in communication systems is increasing more and more. For instance, applications using IoT (the connexion between physical objects) is expected to increase significantly and therefore the amount of data contained in communication will increase as well. Consequently, data traffic is significantly increasing by the year as it is shown in figure 1.1. This development in communication settles some new requirements for optical communication:

- Very-short distance optical communication technology to support large data capacitance
- Being able to achieve mass-production of devices
- Enabling the fabrication of small-footprint devices

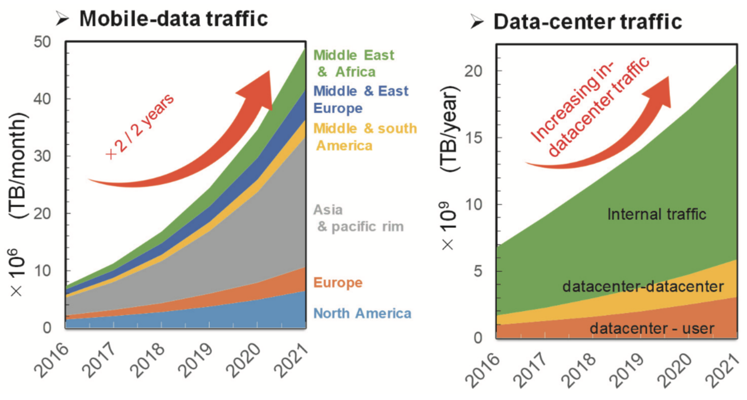

Figure (1.1): *Evolution of optical data capacity. Original data from cisco: [1]*

This dynamic conflicts with some conventional optical devices such as spectrometers as they have two major drawbacks: they are bulky and expensive because they consist of a combination of mirrors, lenses and grating structures as shown in figure 1.2. Furthermore, there is in such devices a trade-off between size and resolution, which means that to obtain higher resolution, an increase in the size of the device is needed. In a conventional optical spectrometer, the input light is reflected by a mirror onto a grating structure that will reflect light with a different angle for each wavelength component in order to separate them. They will then be focused by a mirror onto a detector.

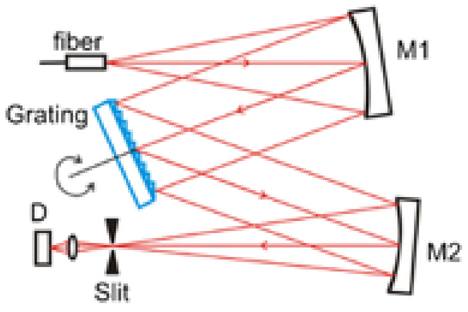

Figure (1.2): *Schematic of a conventional optical spectrometer*

The goal of reducing both size and cost of optical instruments doesn't only apply to devices such as transmitters, receivers, modulators, de-multiplexers, and splitters, but also for test and measurement instruments, such as spectrometers, that are essential for assuring good operation of the optical network. Consequently more and more researches have been made towards the significant reduce in size and cost of optical devices as well as the integration on chip of optical devices.

1.2 Spectrometer applications

A Spectrometer takes light as input and reconstruct the light's spectrum, in other words it detects the frequencies and power associated to each frequency in the optical signal. Spectrometers have a wide range of applications. One example is to be found in optical communication. There exist techniques of data transmission that allows the increase in data rate and to save physical space. One of these technique, called WDM (Wavelength Division Multiplexing) involves sending different data through the same media by modulating the different data on carriers having different frequencies. The data can be thus mixed into a single signal with the different data not interfering with one another because they are carried on different well separated frequencies. [38]

Another example is to be found in food industry where it is possible to use spectrometry to determine food quality and insure customers health through different methods. One being, for example, to shine light through a food sample and check on the output spectrum which wavelengths were absorbed by the food sample. [9]

1.3 Previous Researches

Previous researches have been made towards the on chip integration of optical spectrometers. What comes out of these researches is that silicon photonics is considered a good candidate to reduce both size and cost of the device. [13] In fact, nanofabrication techniques in silicon photonics are very well established and mastered making it very cheap and enabling the possibility of mass-productivity. Silicon also exhibits another advantage: It has low absorption in telecommunication wavelengths, which reduces the lost [22]. For these reasons, Silicon photonics appears as the ideal candidate for this application.

As a first example, *Andreas C. Liapis et al.* [31] conducted a research on the design of a spectrometer based on optical nano-cavities , i.e a structure that is able to confine light depending on its frequency. This paper proposes two different structures. In the first one, light is conducted into a certain number of resonators each tuned to a different wavelength. Wavelengths that cannot be confined in the structure will exit to the output. By monitoring the output to know from which cavity light exits, they are able to reconstruct the spectrum. These two structures are shown in figure 1.3. The obtained resolution is 0.02nm which is very high, nevertheless the working operation is in a 1.5nm wavelength range which is very narrow.

The second proposed structure has a unique photonic nano-cavity, the wavelength separation from the signal is performed by tuning the temperature of the cavity that will change its resonating wavelength. Since there is a relation between the temperature and the resonance wavelength of the cavity (and thus to the wavelength of the light exiting the cavity), it is possible to reconstruct the spectrum by monitoring the output to see at which temperature we detect light at the output. These two structures are limited by fabrication errors and thus require a very high fabrication precision such that it is difficult to achieve with current CMOS processes.

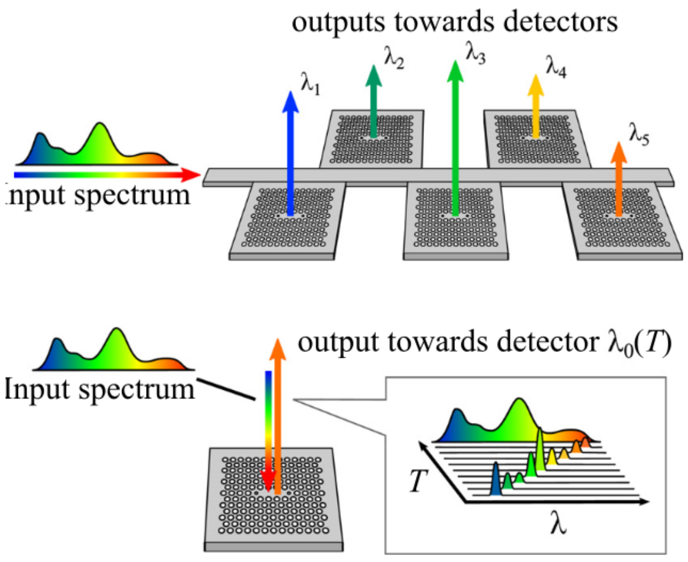

Figure (1.3): *Operating principle of a PhC cavity spectrometer using (a) an array of sequentially tuned cavities or (b) a single dynamically tunable cavity. Permission for the use of this figure was given by the author: Gao B. Siddiqui M. R. Shi Z. Liapis A. C. and R.WBoyd. "On-chip spectroscopy with thermally tuned high-Q photonic crystal cavities". In: Appl. Phys.Lett. 108, 12–16 (2006).*

To solve this problem, *B. Redding et al.*[15] used a random structure. After scattering randomly in the structure shown in figure 1.4, the input light reaches a detector array at the output. Spectrum reconstruction is enabled by recognizing the pattern of the light scattering at the detectors. This proposition overcomes the fabrication error problem since this structure is itself based fully on randomness. On the other hand, because of this feature it is difficult to obtain wavelength information intuitively. Using this method, they achieved a 0.75nm resolution within a 25nm wavelength range. If the wavelength range is higher than in the previous presented design, they didnt achieve a resolution nearly as good as they seems to be limited by a trade-off between resolution and wavelength range of operation.

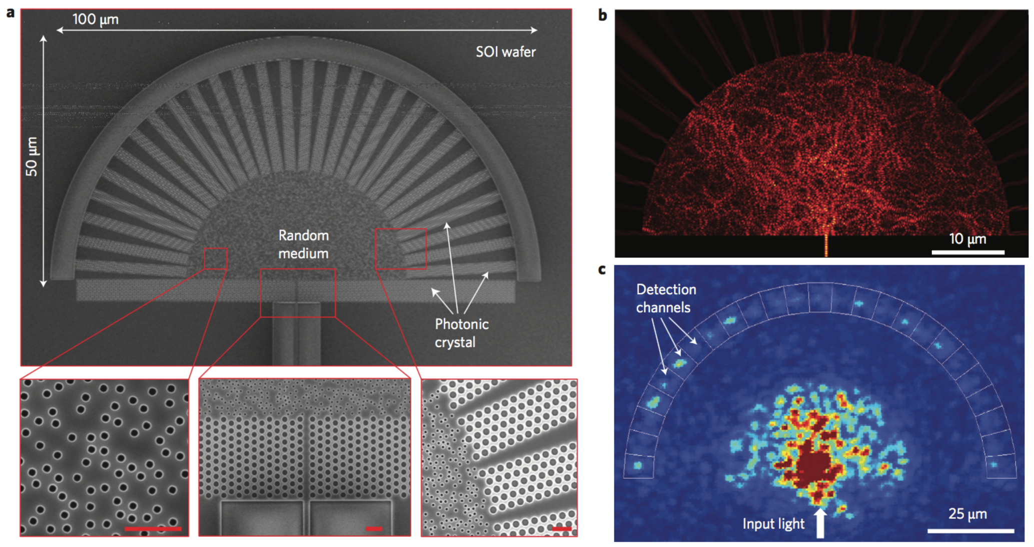

Figure (1.4): ***A chip-based spectrometer based on multiple scattering in a disordered photonic structure.*** ***a***, *SEM image of the fabricated spectrometer. The dispersive element is a semicircular array of randomly positioned air holes. The light diffuses through the random array via multiple scattering and eventually reaches the 25 defect waveguides around the circumference of the semicircle. These tapered waveguides will couple the signals to the detectors (not integrated). The distribution of intensities over the detectors is used to identify the input spectrum.* ***b***, *Numerical simulation of field pattern.* ***c***, *Experimental near-infrared optical image of the random spectrometer with a probe signal at* $\lambda = 1,500nm$. *The white boxes, labelled 'Detection channels', mark the positions of detectors at the end of 25 defect waveguides. To avoid the complexity of integrating the detectors, we estimated the intensity coupled into each output waveguide from the integrated intensity of scattered light within each white box. The out-of-plane scattering is caused by the semicircular groove, shown in* ***a***, *that terminates the waveguides at the location of the detectors. Permission to use this figure was given by the author: R. Sarma B. Redding S. F. Liew and H. Cao. "Compact spectrometer based on a disordered photonic chip". In: Nature Photonics, 7(9), 746–751 (2013).*

1.4 Motivation and objective of this study

Nevertheless, these researches have two major common points: First, they managed to propose a design of a cost effective optical spectrometer with low footprint. Second, they rely on photonic crystal structures which thus seems to be a promising candidate in order to achieve our goal.

Following the requirements and general trend of designing always smaller and cost effective optical devices with which we have to be able to achieve mass-productivity ,we therefore focused our research on using photonic crystal structures while proposing a structure that would solve both issue the previously introduced researches have experienced by using the strong points of both structures. We will use both order and disorder properties of a photonic crystal structure to enable compact but high resolution spectrometer with a wide wavelength range of operation.

This research will essentially focus on the spectrum reconstruction using deep learning. The objectives are:

- Combining deep learning with a small footprint photonic crystal waveguide to propose a compact spectrometer design
- Being able to detect single wavelength signal
- Being able to reconstruct multiple wavelength component signal
- Achieve high resolution on a wide wavelength range

The next chapter will thus concentrate on what is a photonic crystal structure and the structure we propose for our design.

Chapter 2

Photonic Crystals

In the last few decades, engineers in the optical field have been researching on ways to design structures in order to modify their optical properties. Some of these structures are already well implemented and widely used, for instance optical fibers, which simply guide lights with very low losses, were a breakthrough in the telecommunication industry. Complete control over light propagation in a material has been investigated in order to find structures that can allow us to achieve this task. To find the answer, we can perform an analogy with electronics. We managed to achieve control over electronic properties using crystals. A crystal is a material which shows periodicity in the arrangements of its atoms, causing electrons propagating in the crystal to see a periodic potential. This periodicity will cause the material to have gaps in the energy bands, which means that electrons with certain energy cannot propagate in the structure. It is then possible to control the electronic properties of the material by doping the material, which means we will insert foreign atoms in the structure and break the symmetry of the structure.
The idea behind photonic crystal is to mimic this concept and design optical materials such that when light is propagating, it sees a periodicity in the structure that affects its propagation.[53] This periodicity will be achieved by alternating materials in the structure to have a periodic variation of *permittivity*.

2.1 Concepts

Propagation of light is governed by Maxwell's equations given by:

$$\begin{aligned} \nabla \cdot \mathbf{B} = 0 \quad & \vec{\nabla} \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = 0 \\ \nabla \cdot \mathbf{D} = \rho \quad & \vec{\nabla} \times \mathbf{H} - \frac{\partial \mathbf{D}}{\partial t} = \mathbf{I} \end{aligned} \tag{2.1}$$

with $\mathbf{D}(\mathbf{r}) = \epsilon_0 \epsilon_r(\mathbf{r})\mathbf{E}(\mathbf{r})$ and $\mathbf{B}(\mathbf{r}) = \mu_0 \mu_r(\mathbf{r})\mathbf{H}(\mathbf{r})$,$\mathbf{E}$ being the electric field, $\mathbf{D}$ the displacement field, $\mathbf{H}$ the magnetic field, $\mathbf{B}$ the magnetic induction field, ρ the density of charges in the material and I the current in the material. μ_r is the relative permeability , μ_0 is the permeability in vacuum, ϵ_0 is the permittivity in vacuum. $\epsilon_r(\mathbf{r})$ is the relative permittivity and depends on the position $\mathbf{r}$ in the material because it is not constant since in a photonic crystal, there is a periodic variation of permittivity.
It can be shown that if we consider the following assumptions:

- The fields strengths are small enough to be in linear regime (no non linear effects)
- The material is macroscopic and isotropic
- We ignore material dispersion
- We consider the material as being transparent (i.e, $\epsilon(r)$ is purely real and positive)
- no current in the material , $I = 0$
- zero density of charges , $\rho = 0$

It can be shown that with 2.1 we can write:

$$\vec{\nabla} \times (\frac{1}{\epsilon(\mathbf{r})} \nabla \times \mathbf{H}(\mathbf{r})) = (\frac{\omega}{c})^2 \mathbf{H}(\mathbf{r}) \quad \text{with} \quad c = \frac{1}{\sqrt{\epsilon_0 \mu_0}} \tag{2.2}$$

This equation is the main equation that describe, using it together with 2.1, the propagation of $\mathbf{H}(\mathbf{r})$ for a given structure $\epsilon(\mathbf{r})$. If we define the operator $\hat{\Theta}$ as

$$\hat{\Theta}\mathbf{H}(\mathbf{r}) := \vec{\nabla} \times (\frac{1}{\epsilon(\mathbf{r})} \vec{\nabla} \times \mathbf{H}(\mathbf{r})) \tag{2.3}$$

We can rewrite 2.2 as

$$\hat{\Theta}\mathbf{H}(\mathbf{r}) = (\frac{\omega}{c})^2 \mathbf{H}(\mathbf{r}) \tag{2.4}$$

Equation 2.4 shows that, since the application of $\hat{\Theta}$ gives $\mathbf{H}(\mathbf{r})$ multiplied by a constant, $\hat{\Theta}$ is an eigen operator that has $(\frac{\omega}{c})^2$ as an eigenvalue and $\mathbf{H}(\mathbf{r})$ as an eigenvector and solving this equation is therefore the same as solving an eigenvalue problem. We will call, for a given $\epsilon_r(\mathbf{r})$ all the $\mathbf{H}(\mathbf{r})$ that satisfies this equation, a mode of the system. We can classify the different mode of a system by the particular values of their wave vector $\mathbf{k}$. This wave will be able to propagate in the structure given by $\epsilon_r(\mathbf{r})$ at frequency ω. But, for a given $\mathbf{H}(\mathbf{r})$ and $\epsilon_r(\mathbf{r})$, it is possible that there exist values of ω for which there are no solutions. In that case, an electromagnetic wave having frequency ω cannot propagate in the structure and the material exhibits a photonic *bandgap*. Analogously to electronic semiconductor physics where a semiconductor crystal can exhibit an energy bandgap, electrons having an energy in this bandgap not being able to propagate in the structure, photonic bandgap forbids electromagnetic waves having a frequency falling in the bandgap to propagate in the structure.[27] Since the relative permittivity ϵ_r is an intrinsic property of the material, we can conclude that it is possible to control the optical properties of materials by controlling $\epsilon_r(\mathbf{r})$, in other words the design of the structure. In that way it is possible to control the bandgap of the material and to control where in the structure, the light is allowed to propagate and other properties such as its propagation velocity.

The simplest photonic crystal (PhC) is a multilayer film which is shown in figure 2.1. [19] It has a periodic variation of its permittivity in only one direction. It corresponds to an alternation of layers of two different materials.

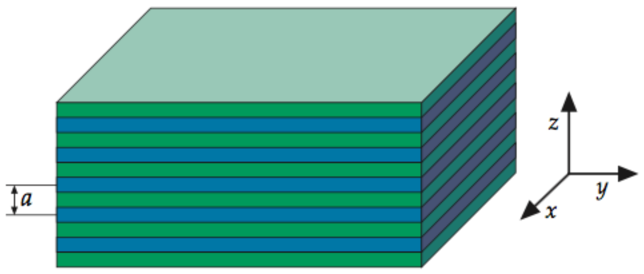

Figure (2.1): *1-Dimensional [John D. Joannopoulos, Steven G. Johnson, Joshua N. Winn, Robert D. Meade, Photonic Crystals: "Molding the Flow of Light". Princeton University Press,p45, 2007.]*

If the wavelength of the incident light is of the same order of magnitude as the periodicity of the crystal, the reflected and transmitted waves at each interface will interfere, leading to a band structure. Indeed for frequencies inside the bandgap, the overall interference will result in no transmission and corresponds to frequencies ω that don't respect equation 2.2. Assuming the 1D Photonic crystal being homogeneous in directions x and y and having a periodic variation of permittivity in direction z as indicated in figure 2.1, since the structure is periodic in z, we can use the *Bloch theorem* stating that we can rewrite $\mathbf{H}(\mathbf{r})$ as the product of a plane wave and a periodic envelope $\mathbf{u}$ having the same translational symmetry as the structure: $\mathbf{u}(z) = \mathbf{u}(z + R)$ with R being an integer multiple of a, the spatial period.

$$\mathbf{H}(\mathbf{r}) = e^{i\mathbf{k_{xy}}\cdot\rho}e^{ik_z z}\mathbf{u}(z) \tag{2.5}$$

$\mathbf{k}_{xy}$ represents the wave vector in the homogeneous plane (xy plane) and can have any value unlike $\mathbf{k}_z$ that is in the direction of permittivity variation. Since the structure is periodic of primitive lattice vector $a\hat{z}$, we can define the reciprocal primitive lattive vector as being $\frac{2\pi}{a}\hat{z}$ in the reciprocal space (space of wave vectors) and completely define the relation between ω and k_z in the interval $-\frac{\pi}{a} < k_z \leq \frac{\pi}{a}$ called the Brillouin zone.

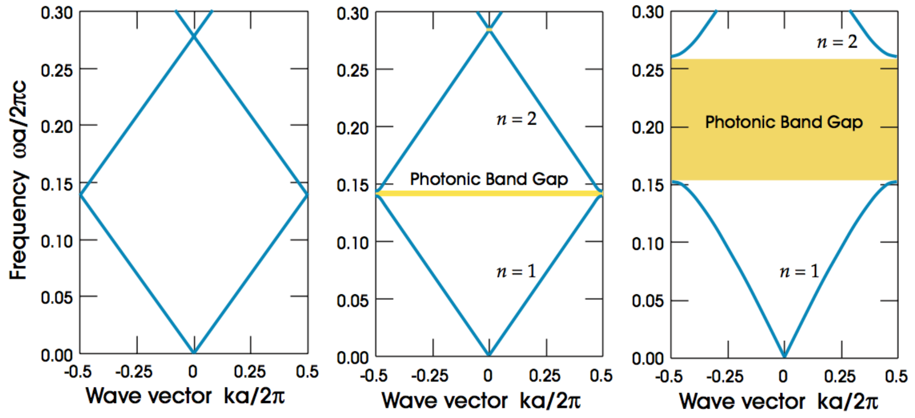

Figure (2.2): *The photonic band structures with increasing values of permittivity difference between the two materials [John D. Joannopoulos, Steven G. Johnson, Joshua N. Winn, Robert D. Meade, Photonic Crystals: "Molding the Flow of Light". Princeton University Press,p45, 2007.]*

We can see in figure 2.2 that the band diagram (i.e diagram showing the relation between frequency and wave vector) with no variation of permittivity given by the left most plot shows the linear relation between w and k given by $w(k) = \frac{ck}{\sqrt{\epsilon}}$. When a variation of permittivity is inserted in the structure, a bandgap starts appearing and increasing the difference of permittivity between the two materials composing the photonic crystal will increase the size of the bandgap .

2.1.1 Photonic Crystal waveguide

2 dimensional PhC structures have periodic variations of permittivity in two directions and are homogeneous in one direction. The propagation of light will thus be modified within a plane. In this case the structure will exhibit a bandgap in the 2 dimensions within this plane.[42, 46] The permittivity of the structure will exhibit periodicity such that $\epsilon(\mathbf{r}) = \epsilon(\mathbf{r} + \mathbf{R})$ with $\mathbf{R}$ being a linear combination of $a\hat{x}$ and $a\hat{y}$ considering a PhC heterogeneous in x and y directions. A 2D PhC will typically consist of a lattice of dielectric columns as shown in the left figure of 2.3. The lattice topology that is most commonly rectangular as in the right figure of 2.3 or triangular with one row of holes out of two shifted by half a period of the lattice as in the left figure of 2.3, but more complex lattices such as hexagonal lattices also exist.[56] The important parameters to consider are the lattice topology, the lattice period a and the radius of the holes r.

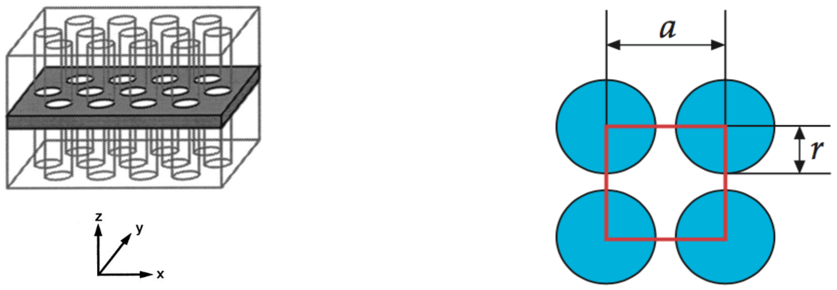

Figure (2.3): *2D PhC structure [Shanhui Fan Steven G. Johnson Pierre R. Villeneuve and J.D Joannopoulos. "Linear waveguides in photonic-crystals slabs". In: Physical Review B 62 (2000).]*

It is possible to modify the optical properties of the structure by including defects in the crystal. A defect consists in breaking the symmetry of the structure, this can be done for example by omitting some holes in the structure or changing the radius of the holes. By introducing these kind of defect along a line in a 2D PhC structure allow us to fabricate a *photonic crystal waveguide*. This type of structure is used to guide light from one point to another while being as confined as possible in the path on which light is guided. Different types of PhC waveguides are shown in figure 2.4. Omitting one or several lines of holes in the structure will change the bandgap structure and introduce possible modes within the bandgap, typically a *guided band* can be introduced inside the bandgap. These modes whose frequencies are in the bandgap will be confined in the defect line as it cannot propagate in the periodic parts of the structure. Indeed, since it is originally a mode that cannot propagate in the structure, it will be confined in the only part in which it can propagate which are the locations where symmetry has been broken.[29] These structure are used to guide light along this line of defect with low loss as the light can only propagate along the waveguide.

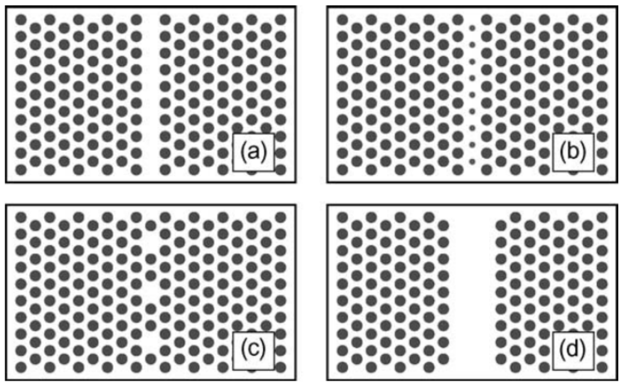

Figure (2.4): *Examples of PhC waveguides: (a) waveguide consisting of a row of missing holes, (b) waveguide consisting of a row of holes with smaller diameter, (c) coupled-cavity waveguide and (d) 3 holes wide waveguide [C. Jamois et al. "Silicon based two dimensional photonic crystal waveguides". In: Photonics and Nanostructures, Fundamentals and Applications 1 (2003).]*

PhC waveguides have the interesting property of only allowing light up to a certain wavelength propagate. This maximal wavelength value for which light can propagate in the waveguide, the *mode gap* wavelength, is sensitive to the width of the waveguide. Light having a wavelength above this mode gap wavelength cannot propagate in the waveguide. Waveguides with different widths will therefore exhibit different mode gap wavelength values. This relation is illustrated in figure 2.5 where we can see that, starting from a width W and reducing it to values $0.80W$ and $0.65W$, the guided band is shifted upwards increasing the photonic bandgap (PBG) therefore reducing the range of wavelengths being able to propagate in the structure by reducing the value of the mode gap wavelength [28].

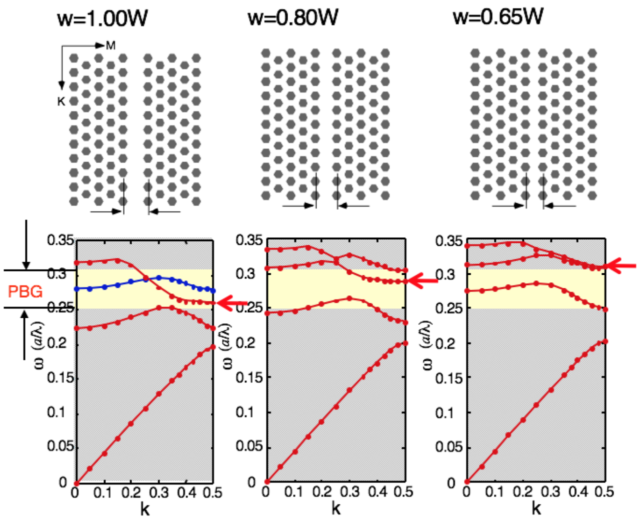

Figure (2.5): *Relation between width an Bandgap*

Figure 2.6 shows the transmission of a photonic crystal waveguide for different values of width, we can see that the transmission drops for lower values of wavelength when the width is smaller.

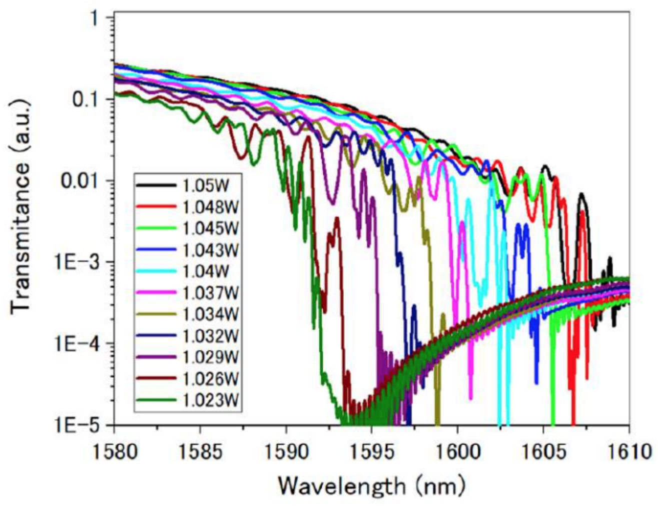

Figure (2.6): *MEEP simulation showing the transmission in PhC waveguide for different width values*

2.1.2 Anderson Localization

Photonic crystal structures are usually fabricated with lithography fabrication processes that have inherent resolution limits.[14] For example, the resolution of photolitography is limited by the wavelength of the shined light. These fabrication errors will introduce uncertainty and therefore induce randomness in the structure. In photonic crystal waveguides, we will thus have an uncertainty on the position of the holes, the radius of the holes and the width of the waveguide. In an ideal PhC waveguide, light propagation is described by bloch modes given by equation 2.5. In such ideal structures, the group velocity, given by:

$$v_g = \frac{d\omega}{dk} \tag{2.6}$$

and therefore corresponding to the derivative of the dispersion relation $\omega(k)$ illustrated by the curve of the band diagram, vanishes at the band edge if its wavelength is higher than the mode gap wavelength as it cannot propagate in the structure [36]. In the case of a non-ideal PhC waveguide, it has been shown experimentally that the group velocity v_g is very small when the wavelength is at the mode gap of the PhCW and can localize in the structure. Indeed, the mode gap wavelength will not be perfectly constant along the waveguide because the width itself is not perfectly constant due to random variations induced by the fabrication errors. Light having a wavelength above the mode gap wavelength (therefore a wavelength below the cut-off wavelength, making it possible to propagate in the structure) can be trapped in regions in the waveguide where the surrounding regions exhibit a mode gap wavelength higher than the ideal mode gap wavelength as it is indicated in figure 2.7.

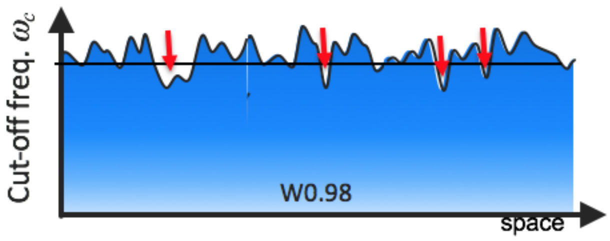

Figure (2.7): *Illustration of the variation of the cut-off frequency (corresponding to the mode gap wavelength but expressed in terms of frequency) induced by fabrication errors with respect to the position along the waveguide. To be able to propagate, light should have it's frequency above the cut-off frequency (which corresponds to its wavelength being below the mode gap wavelength). The black line represents the value of the cut-off frequency in the ideal case. The red arrows shows places in the waveguide where light can localize [55]*

This is a phenomenon called *anderson localization.*[26] The interference of the scattered light in the structure can lead to the formation of modes, called *localized modes*, that appear at random positions in the structure . Indeed, slow light is susceptible to backscattering which blocks the propagation of light if the wavelength is above the mode gap wavelength. The introduction of random variations in the structure creates localized states at the edge of the mode gap, creating resonant cavities in which light localizes along the waveguide. [24, 39] Figure 2.8 shows the localized states in the waveguide and figure 2.9 shows the possible randomness that can modify the transmission spectrum of the photonic crystal waveguide.

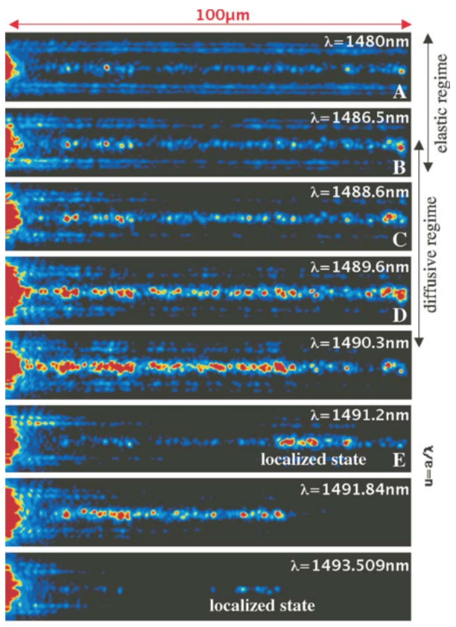

Figure (2.8): *example of localized state in a disordered photonic crystal structure. Light starts to localize in random positions in the waveguide when the wavelength is close to the mode gap [J. Jagerska V. Zabelin N. Le Thomas H. Zhang and R. Houdre. "Light transport regimes in slow light photonic crystal waveguides". In: Physical Review B 80 (2009).]*

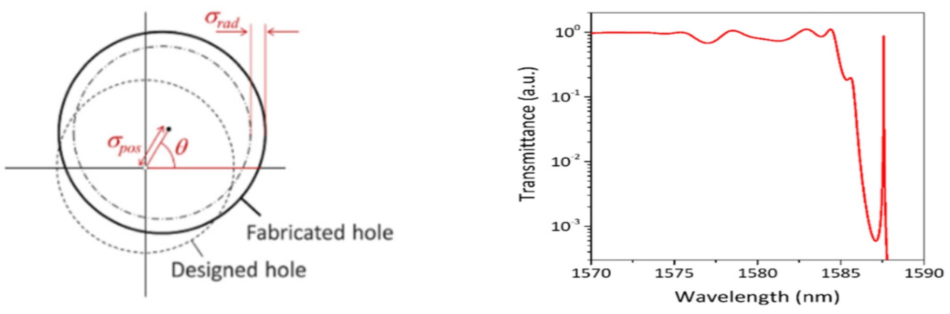

Figure (2.9): *Left: Random variations induced by fabrication errors, the holes can have an error on their radius or their position. Right: localized state above mode gap [Tomohiro Tetsumoto Yuta Ooka Nurul Ashikin Binti Daud and Takasumi Tanabe. "Compact resonant electro-optic modulator using randomness of a photonic crystal waveguide". In: Optical Society of America (2016)]*

2.2 Proposed Structure

Taking advantage of the property of photonic crystal waveguides having a mode gap wavelength depending on the width of the structure, we propose to use a chirped photonic crystal waveguide [35] which is a photonic crystal waveguide whose width is not constant and decreases along the waveguide. Due to the limitation in fabrication accuracy, the width does not decrease constantly but in steps, the chirped structure is therefore divided into sections of constant width, each section having a different width as illustrated in figure 2.10.

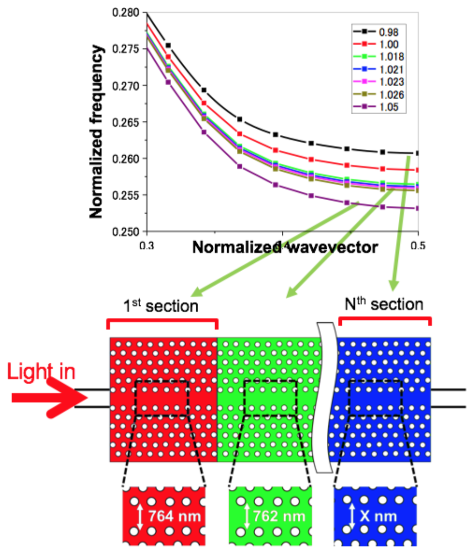

Figure (2.10): *Illustration of a Chirped Photonic crystal waveguide*

Since each section has a different width (the width decreasing from one section to another) the mode gap wavelength along the structure will decrease. This will cause light with different wavelengths to propagate in the waveguide, in fact light will propagate through the waveguide until it reaches the point where it's wavelength becomes higher than the mode gap wavelength of the section it is propagating in. When the input light comes into a section in which it cannot propagates, light will have to scatter backwards and upwards since it cannot propagate further nor scatter to the sides because it cannot propagate in the periodic structure (the 2D photonic crystal structure being designed to have its bandgap around the frequency of operation and therefore confining light in the waveguide). Ideally, this means that light with different frequency components will exhibit different patterns because each frequency component will be able to propagate in the waveguide up to different points therefore creating different patterns for different input light sources. By recognizing these different patterns, we would be able to tell which frequency components are present.

Nevertheless, the resolution in fabrication of the chirped structure will limit the resolution of our system because the decrease in width of the waveguide is not constant and is done in steps which divides the structure into section having the same mode gap. This means that rather than the mode gap variation being continuous, it is discrete. The difference in mode gap wavelength between two neighbouring sections will be the wavelength resolution of the structure, two input light having a wavelength differing by less than this wavelength resolution might therefore propagate in the waveguide up to the same point in which case we would not be able to tell the difference between these two wavelengths. The resolution of the whole system would be define by the wavelength resolution of the photonic structure, at least theoretically and not taking into account the fabrication errors. Indeed, as explained in the previous section, fabrication errors will cause light to randomly localize in the structure and these random localization are strongly frequency dependent. Furthermore, light will exhibit Anderson localization when its wavelength is closer to the mode gap. We therefore hope to take advantage of fabrication errors to go beyond the wavelength resolution imposed by the structure with the correct pattern recognition techniques. The pattern recognition process will therefore be based on two phenomena, a first that is predictable (we can have information on the wavelength of the input light by looking up to which section in the waveguides it propagates) and one that is random (detecting wavelength sensitive random localizations of light in the structure) that will help improving the resolution beyond the resolution imposed by the structure. Since we also rely on random phenomena, we choose to use deep learning for pattern recognition. The algorithm should be able to recognize the pattern taking the two phenomena explained above as feature of interest.

Following these reflexions, we decide to propose the following system: Light will input a chirped photonic crystal waveguide that will exhibit, looking from the top of the waveguide, different patterns depending on the frequency component present in the light signal. A camera at the top of the slab will take a picture and feed it to a model that will predict the wavelengths of the input light. This model will have been obtain by using deep learning, which means we will train our model with training images to make it able to perform pattern recognition. Figure 2.11 shows a schematics of the proposed design of the system.

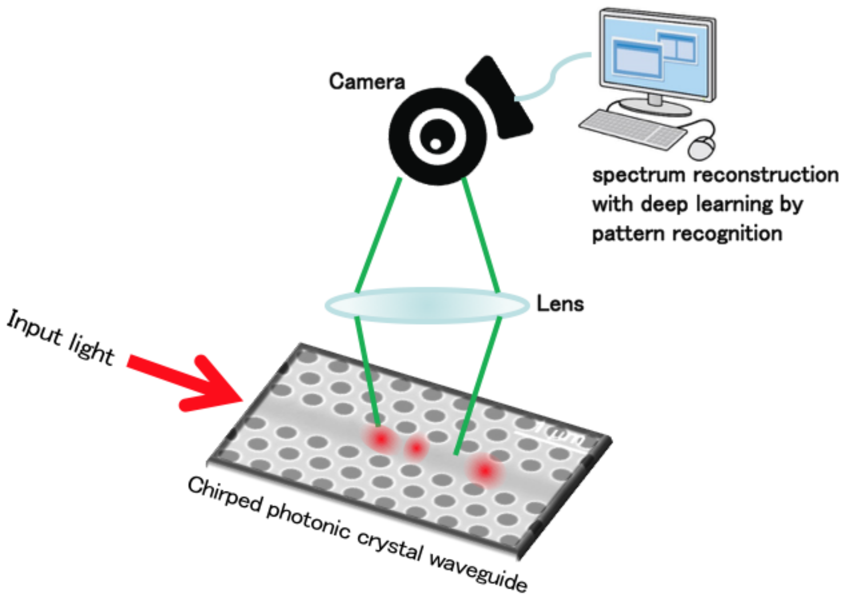

Figure (2.11): *Illustration of the proposed spectrometer system*

Chapter 3

Deep Learning

3.1 Basics

The application of machine learning to solve various cases of problems didn't stop increasing since its introduction in the 1950's. It refers to a mathematical and statistical approach of designing algorithms that allows the computer to increase its performances towards solving a problem through a self-learning process based on a provided dataset. Artificial neural networks aim to simulate the learning process in biological organisms. In such networks, neurons are connected together in order to perform computation. In artificial neural networks, each neuron is considered has a computation unit, which take several inputs and gives one output. [51] The n inputs $x_1, ..., x_n$ will be linearly combined by a set of weights $w_1, ..., w_n$ specific to each neuron of the network added to a bias b. This combination of the inputs will then be processed by a function z called activation function. A schematic representation of an artificial neuron is shown in figure 3.1. Each neuron will therefore act as a function that can be written has:

$$f(x_1, ...x_n) = g(\sum_{i=1}^{n} w_i x_i) + b \tag{3.1}$$

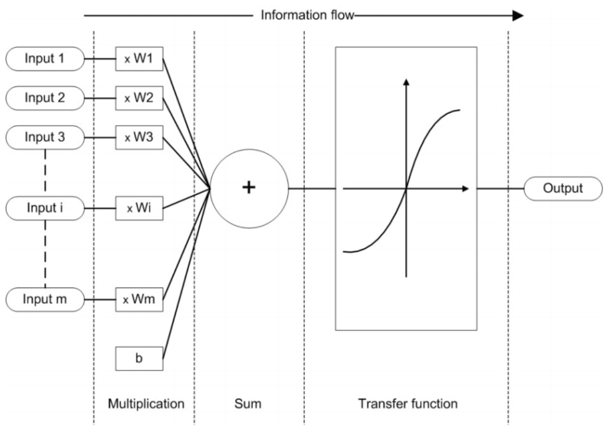

Figure (3.1): *Working principle of an artificial neuron [Kenji Suzuki,"Artificial Neural Networks - Methodological Advances And Biomedical Applications." InTech,p1, 2011]*

By combining these individual neurons together, we obtain a network that is able to take several input and perform complex computation.[40] This results in a simple neural network as shown in figure 3.2. It is important to note that each individual neuron in the network will have its own set of parameters: inputs and weights.

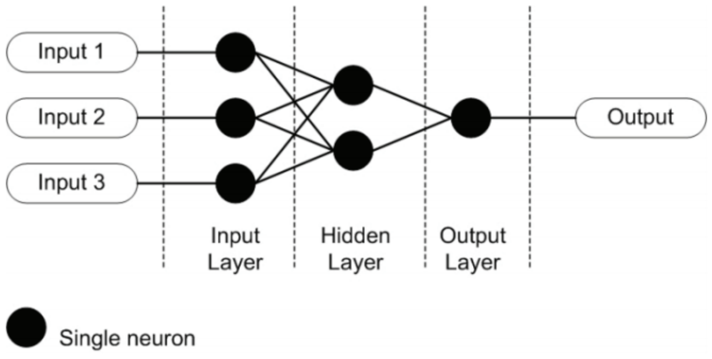

Figure (3.2): *Example of simple artificial neural network. [Kenji Suzuki,"Artificial Neural Networks - Methodological Advances And Biomedical Applications." InTech,p1, 2011]*

An artificial neural network is composed by layers that are a set of neurons that are taking as input the output of the previous layer. The first layer is called the input layer and the output is called the output layer. All layers in between are called hidden layers because the flow of information in these layers are completely hidden and it is not possible to know any intermediate state between input and output. The way the individual neurons are connected to each other is called the *architecture* of the neural network. The choice for the architecture will depend on the application, different architectures will be more or less efficient towards different applications. For example, we will choose different activation functions depending on the applications, it will be different if we want to predict a probability (for which we need a function that outputs a value between 0 et 1), if we need to output a real value without specific boundaries or if we need to output binary values.[12] The number of output nodes of a layer is also dependent to the activation function which in turn depends on the application. It is important to choose the appropriate activation function for the targeted application. Figure 3.3 shown in [12] shows the most common activation functions.

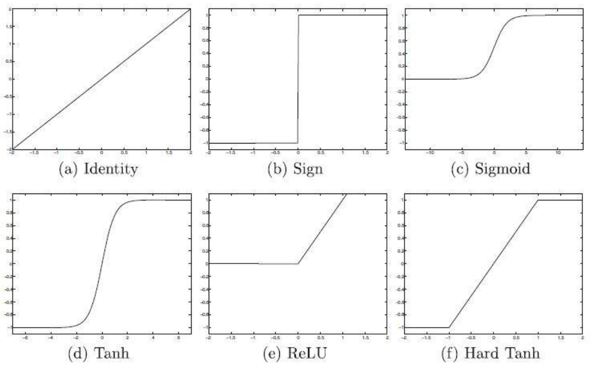

Figure (3.3): *Common activation functions [Charu C. Aggarwal, "Neural Networks and Deep Learning". Springer,p13, 2011.]*

3.2 Fully Connected Network and learning process

Fully connected Network (FCN) are multilayer neural network, they contain more than one computational layer.[21] These architecture are called feed-forward networks because the flow of computation only follows one direction from the input to the output. The term "fully connected" refers to the fact that each nodes from the previous unit are connected to all nodes of the next unit as it is the case in figure 3.2. Therefore, the architecture is almost fully defined once the number of layer and the number of nodes are determined.

First let's introduce the learning process with a larger view. We will focus in this work on a type of learning process called *supervised learning*, in which learning is performed through feeding the algorithm example data on which the model will try to fit in order to perform predictions on unknown data that is supposedly representative of the fed data. This method contrasts with *unsupervised learning* in which learning is performed with unlabelled data (unknown data).[33] Here, the input will be what can be referred to as training data. It will be data that is representative of data that we want to be able to predict. As a simple example (example that we will use across this chapter), if we want to build a program that is able to recognize images of cats, we will give as input our training data that will be images containing cats and images that don't. To each input x will be associated a label y that correspond to the output we want our program to associate with the input. In our example, we can see that the problem has a binary answer, either the image contains a cat or it doesn't. We can therefore associate a binary value to the labels: either 0 (if the image does not contain a cat) or 1 (if the image contains a cat).
After this learning process it is important to test the algorithm on a test set that must be different from the training set to check if the algorithm learned correctly and reaches satisfying accuracy. If it does, it means that the obtained model is suited for the targeted application.

Now that we saw the general view on the learning process, we can concentrate into how this process works in more details. To understand the mathematics behind

FCNs, we have first to take a look at some notations, including matrix notations that will allow us to write the equations in a more compact matricial form. Let's consider a FCN with k layers . The input layer is considered as layer 0 because no computation is performed, it just transmits data. The different notations are given as follows:

- We will write x_i the i_{th} input node.
- $\bar{x}$ is a vector containing the input nodes $\bar{x}_i$.
- h_{pi} is the i_{th} node of the p_{th} hidden layer.
- $\bar{h}_p$ is the vector containing the hidden layer nodes h_{pi} in the p_{th} hidden layer.
- W_i is the matrix containing the weight parameters connecting layers $i-1$ and i.
- b is the added bias.
- Φ_i corresponds to the activation function of he i_{th} hidden layer.
- We will write $y^{(i)}$ the i_{th} output node.
- $\bar{y}$ is a vector containing the output nodes $\bar{y}^{(i)}$ also called labels.

The set of equations showing the flow of computation in such networks and giving the relation between the input and the output can be given as follows:

$$\begin{aligned} &\bar{h}_1 = \Phi_1(W_1^T \bar{x}) \quad \text{Input to Hidden layer} \\ &\bar{h}_{p+1} = \Phi_p(W_{p+1}^T \bar{h}_p) \quad \forall p \in \{1 \ldots k-1\} \text{Hidden to Hidden Layer} \\ &\bar{y} = \Phi_{k-1}(W_{k+1}^T \bar{h}_k) \quad \text{Hidden to Output Layer} \end{aligned} \tag{3.2}$$

Having these equations,[48] we can now see how we can manage to obtain a learning process. Here we will discuss briefly on this method. This process is based on a technique called gradient descent which is an optimisation problem that is given as follows: we want to minimize a cost function $J(W, \bar{b})$ depending on the parameters W representing a matrix containing all the weights of the system and $\bar{b}$ a vector containing all the biases of the system. The cost function is given by

$$J(W, \bar{b}) = \frac{1}{m} \sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) \tag{3.3}$$

where $\hat{y}^{(i)}$ is the i_{th} output of the network (the prediction given by the network) and $y^{(i)}$ is the real value that we want to predict (in an ideal case, $\hat{y}^{(i)} = y^{(i)}$). $L(\hat{y}^{(i)}, y^{(i)})$ is called the loss function and computes the error for a single training example while the cost function is the average of the loss functions of the entire training set. $J(w, b)$ has the property of being a convex function and therefore possessing a minimum.

The learning process is the fact that we are going to perform several iterations of the computations while updating the parameters w and b in between each iterations in the following way:

$$\begin{aligned} w &:= w - \alpha \frac{\partial J(w,b)}{w} \\ b &:= b - \alpha \frac{\partial J(w,b)}{b} \end{aligned} \tag{3.4}$$

The algorithm will update the parameters in order to minimize the cost function. w and b are called *learning parameters* because they are the parameters on which the learning process is performed.[52, 43] α is the learning rate of the system and will have an influence on the sizes of the steps we use to reach to the minimum of the cost function. Setting it too low will increase the time we take to reach the minimum but setting it too high might give a situation where the steps are so big that we end up passing over the minimum and ending up to far from it.

Typical problems may arise during the learning process. One of them being what is called over-fitting, which is the fact that, even though we want our model to fit on the training set, we don't want it to fit too much. Over-fitting arises when the model doesn't generalize the problem enough. Taking our example problem of recognizing cats, if we only put images of white cats into the training set, the algorithm might over-fit and not recognize cats having other colors. The reason is that, during the learning process, the program will try to detect features that characterizes the input. Here the program might focus on the feature "white", but this is not a feature that represents a cat, in contrast to other elements like the shape of the body or the ears. This is why the training set must be as general as possible, so that the program concentrates on features that really represents the input.

3.3 Convolutional Neural Network

A convolutional neural network (CNN) is a type of neural network that strongly differs from FCNs. While FCN relies on the way nodes are arranged and connected together, CNNs rely essentially on filtering. The learning parameters of CNNs are the values of the filter. They are randomly initialized at the start of the process and will then be updated at each iteration of the algorithm in order to minimize the cost function as in FCNs. CNNs are usually used for applications were the input is an image, therefore a 3 dimensional input having *height*, *width* and *depth*. The last term refers to the number of *channels*. If we represent our input image as a matrix, there would be 3 channels corresponding to the RGB (amount of red, green and blue) values. The computation in CNN architectures are composed by convolutional layers that include 3 operations in order: *convolution*, *pooling* and *Relu*.

First, convolution is an operation where the input will be processed by a filter (also often called *kernel*). It is important to note here that we will refer to the terms "convolution" as *matricial convolution* which is defined a little bit differently than the conventional convolution operation. The matricial convolution operation will place the filter at every possible position in the image and perform a dot product. The results from all these dot products are placed in a matrix. The position of the filter on the input image determines the position of the obtained value in the obtained matrix. It is important to note that the output matrix will have different dimensions. In fact, if the filter has dimension $h \times w \times d$ and the image has dimensions of height and width $h_f \times w_f \times d$, then the obtained matrix will have dimensions corresponding to $h - h_f + 1 \times w - w_f + 1$. We can see here that the number of channels of the filter is the same as the number of channels of the image, which is a requirement to satisfy the applicability of matrix convolution. An example of matricial convolution is given in figure 3.4. During the convolution step, the input can be processed by more than one filters. The results of each filtering will each be a different channel of the ouput. In other words, if the input has dimensions $m \times m \times d$ and is processed by n filters of size $f \times f \times d$, the output will have dimensions $m - f + 1 \times m - f + 1 \times n$ as illustrated in the example given by figure 3.5.

The next operation in a convolutional layer is *pooling*. The idea behind this concept is to sample the data by taking a window of a certain size $P \times P$, and produce a matrix with the same depth but for each square of size $P \times P$, we select one element according to the pooling method. The most commonly used pooling method are *max-pooling* where we only select the maximum in each square and *average-pooling* in which we perform the average of each square. As a result of this sampling, the data will have its dimensions reduced by a factor P.

The last step of a convolutional layer is to apply an activation function which will be applied element-wise in the matrix. The most commonly used activation function is *ReLu* that is shown in figure 3.3 and whose equation is given by:

$$g(x) = max(x, 0) \tag{3.5}$$

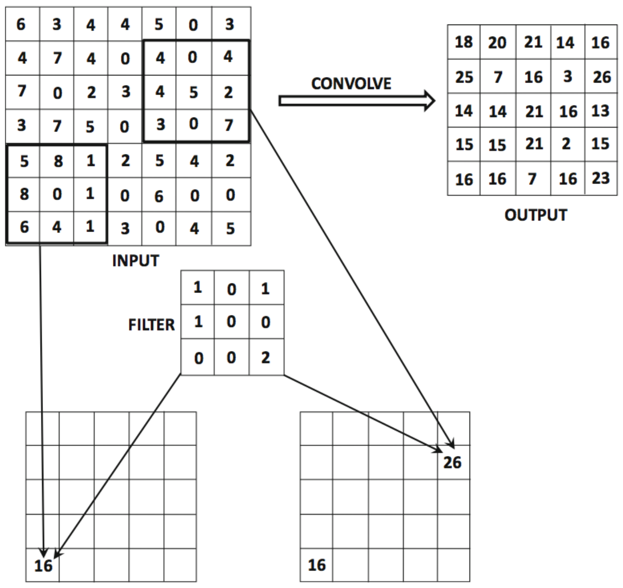

Figure (3.4): *Example of a convolution between a 7×7×1 input and a 3×3×1 filter [Charu C. Aggarwal, "Neural Networks and Deep Learning". Springer,p13, 2011.]*

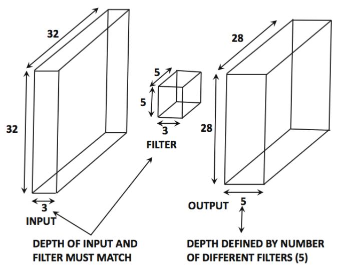
32
32
3
INPUT
5
5
3
FILTER
28
28
OUTPUT
5
DEPTH OF INPUT AND FILTER MUST MATCH
DEPTH DEFINED BY NUMBER OF DIFFERENT FILTERS (5)

Figure (3.5): *Example of a convolution step with 5 filters [Charu C. Aggarwal, "Neural Networks and Deep Learning". Springer,p13, 2011.]*

In order to build a CNN, we just have to stack convolutional layers one behind another in order to form a network that will have a general form: Convolution1 $\longrightarrow$ Pooling1 $\longrightarrow$ ReLu1 $\longrightarrow$ Convolution2 $\longrightarrow$ Pooling2 $\longrightarrow$ ReLu2 $\longrightarrow$ . . . . These architecture are then combined with a FCN that will output the prediction. Since the data in FCNs are vectors and not matrices as in CNNs, it is important to perform a flattening operation after the last convolutionnal layer. This flattening operation just converts a matrix $n \times m \times p$ in a vector containing nmp elements ordered following the rows of the matrix, then the columns and finally the depth, as shown in figure 3.6

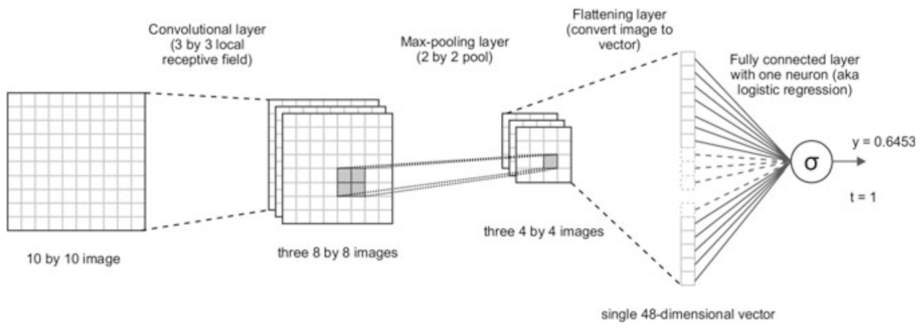
Convolutional layer (3 by 3 local receptive field)
Max-pooling layer (2 by 2 pool)
Flattening layer (convert image to vector)
Fully connected layer with one neuron (aka logistic regression)
y = 0.6453
t = 1
σ
10 by 10 image
three 8 by 8 images
three 4 by 4 images
single 48-dimensional vector

Figure (3.6): *A CNN with a convolutional layer, a max-pooling layer, a flattening layer and a fully connected layer with one neuron [Sandro Sanski,"Introduction to Deep Learning: From Logical Calculus to Artificial Intellingence." Springer,p126, 2011]*

3.4 Classification

3.4.1 Classification problem

Several categories of supervised learning problem exists, the most common being linear regression, where the goal is to predict a value and classification where the goal is to put the input into a category. In our case, we will focus on the latter as it will be the one we will use in our application. Taking our example of recognizing images of cats, it is a simple example of classification problem. Indeed the goal is to classify the data into two different categories: "cat" or "not a cat". Here we will also consider our problem as a classification problem As it will be seen in chapter 4. Fixing a wavelength resolution for our model and a wavelength range, we will have a finite number of classes, the goal will then be to classify the input image into one of these classes to determine the wavelength of the input signal. For example, if we want to have $1nm$ wavelength resolution on a $20nm$ wavelength range, we will have 20 different classes corresponding to the 20 wavelength we can detect and we will associate every input with these classes.

Several activation functions exist to be able to design a classifier, we will focus here on two of those. The first one is the *softmax* function.

$$\zeta(\mathbf{z}) := \frac{e^{z_j}}{\sum_{n=1}^{N} e^{z_k}} j = 1 \dots N \tag{3.6}$$

This function transforms a vector $\mathbf{z}$ with arbitrary real values into a vector with values ranging from 0 to 1. This is the reason why the softmax is used in the final layer of a FCN used for multiclass classification (classification with more than two classes) to get the output which can be a probability for the input to belong to the classes. The highest probability is then taken as the prediction. The probabilities given by softmax are dependant probabilities (their sum equals to 1).

The other activation function we will focus on is the *sigmoid* function shown in figure 3.7 and given by:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{3.7}$$

that also gives an output between 0 and 1.

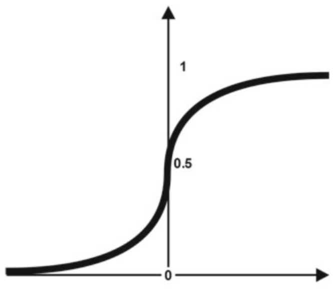

Figure (3.7): *Graph of sigmoid function [Sandro Sanski,"Introduction to Deep Learning: From Logical Calculus to Artificial Intellingence." Springer,p63, 2011]*

3.4.2 Metrics to evaluate classification

The evaluation of such models require to introduce four concepts: *True Positive* (TP),*True Negative* (TP),*False Positive* (FP) and *False Negative* (FN).
A True Positive arises when the classifier predicts X and it is truly an X (correct prediction). A True Negative arises when the classifiers predicts that the data is not X and it is truly not (correct prediction). A False Positive arises when the classifier predicts that the data is X when it is not (bad prediction). A False Negative arises when the classifiers predicts that the data is not X when it is X (bad prediction).
With these concepts introduce, we can now define the fundamental metrics to evaluate classifiers. The most fundamental one is the *accuracy* that is defined as the ratio of good prediction out of all predictions:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{3.8}$$

High accuracy means better model. But, even though the concept of accuracy seems trivial, care must be taken as it is a great measure taking the assumption that we have a symmetric dataset where the number of false positive and false negative are almost the same. The next metric is the *precision*, which is a measure of the rate of False Positives. Increasing precision means reducing the number of "false alarms", positive predictions turning out to be wrong, or in other words, the ratio of true positives out of all the positive predictions.

$$Precision = \frac{TP}{TP+FP} \tag{3.9}$$

The third metric is called *Recall* and measures how a classifier misses correct positive predictions out of all the predictions that turn out to be positive. It is used in the case we are concerned about "missing" true positives.

$$Recall = \frac{TP}{TP + FN} \tag{3.10}$$

Finally the last metric is called FScore and is a weighted average of precision and recall.

$$FScore = \frac{2Recall.Precision}{Recall + Precision} \tag{3.11}$$

Depending on the application and the objectives; one or several of these metrics can be used. [11]

Chapter 4

Experiments

4.1 Designed Photonic Structure

We decided to design our spectrometer for an application in the infrared part of the spectrum which corresponds to wavelengths in the range from 700 nm to 1 mm. Two photonic chips were designed: Chip A that will be used for single wavelength prediction and Chip B that will be used for the wavelength prediction of signals containing one or two wavelength components.

4.2 Experimental Setup

A Tunable laser diode (SANTEC TSL-510) with a resolution of $5pm$ and a power peak $\geq$ 13 dBm[2] was used to input the different signals into the structure while monitoring their wavelength and power . A power meter (Agilent 81634B, power range: -110 dBm to +10 dBm [3]) was used at the output of the structure to monitor the output power in order to maximize the transmission in the structure by matching the position of the laser and the input of the structure. In order to take images, an NIR (Near Infra-Red) camera (Goodrich SU320KTS-1.7RT/RS170) was placed at the top of the structure. A light focusing module with a focusing diameter of 2.8μ m, an extinction ratio $\geq$ 20 dB, an input loss $\leq$ 0.9 dB was used. The optical fiber conneting the laser to the input of the chip and the output to the power meter is a PMF fiber (Polarization-Maintaining optical Fiber). The experimental setup is shown in figure 4.1.

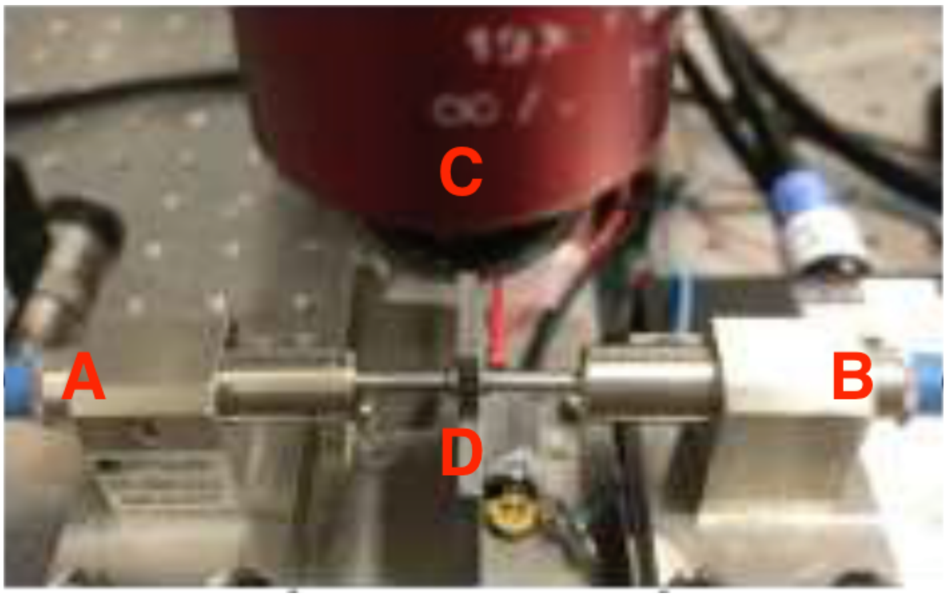

Figure (4.1): *Experimental setup: A) Input connected to the tunable laser diode. B) Output connected to the power meter. C) Infrared camera. D) Chip containing the photonic crystal structure.*

The same setup was used for chip B. To be able to produce multiple frequency component signals, a 50:50 splitter was used to mix the single frequency signal and produce an output signal which is the superposition of the two input signals.

4.3 Experimental data

4000 images were taken using chip A. The pictures were taken by doing a sweep through the wavelengths between 1565 nm and 1586.8 nm with a 0.1 nm step, which corresponds 200 different frequencies. This sweep was realised at 20 different power levels (from -10 dBm to 9 dBm) which results in a total of $20 \times 200 = 4000$ images. It is possible to approximate the values of the mode gap of the different sections by doing a sweep across the different wavelengths and checking the wavelength for which we observe Anderson localization because, as it was explain in Chapter 2, Anderson localization occurs when the input wavelength is close to the mode gap. Figure 4.2 shows an example of experimental data taken for an input wavelengths of 1573.8 nm 1574 nm.

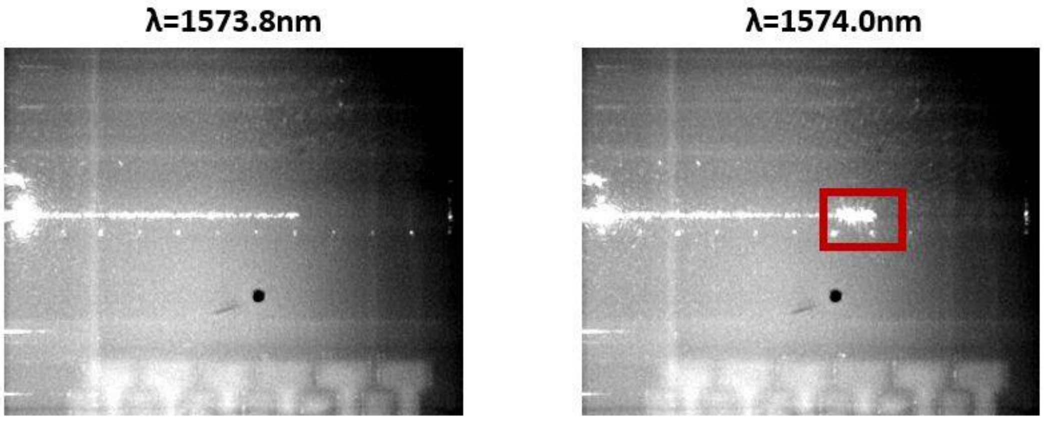

Figure (4.2): *Images of signals at wavelength* 1573.8*nm and* 1574*nm. They propagate until the same section in the waveguide which shows that there difference in wavelength is smaller than the wavelength resolution of the structure but the signal at wavelength* 1574*nm experiences Anderson localization because its wavelength is very close to the mode gap*

Figure 4.3 show the approximate mode gap wavelengths for each sections are respectively given by 1566.9*nm*,1568.9*nm* 1570.3 nm, 1571.9 nm, 1574.0 nm, 1575.8 nm, 1577.5 nm, 1578.5 nm, 1580.3 nm and 1581.2 nm.

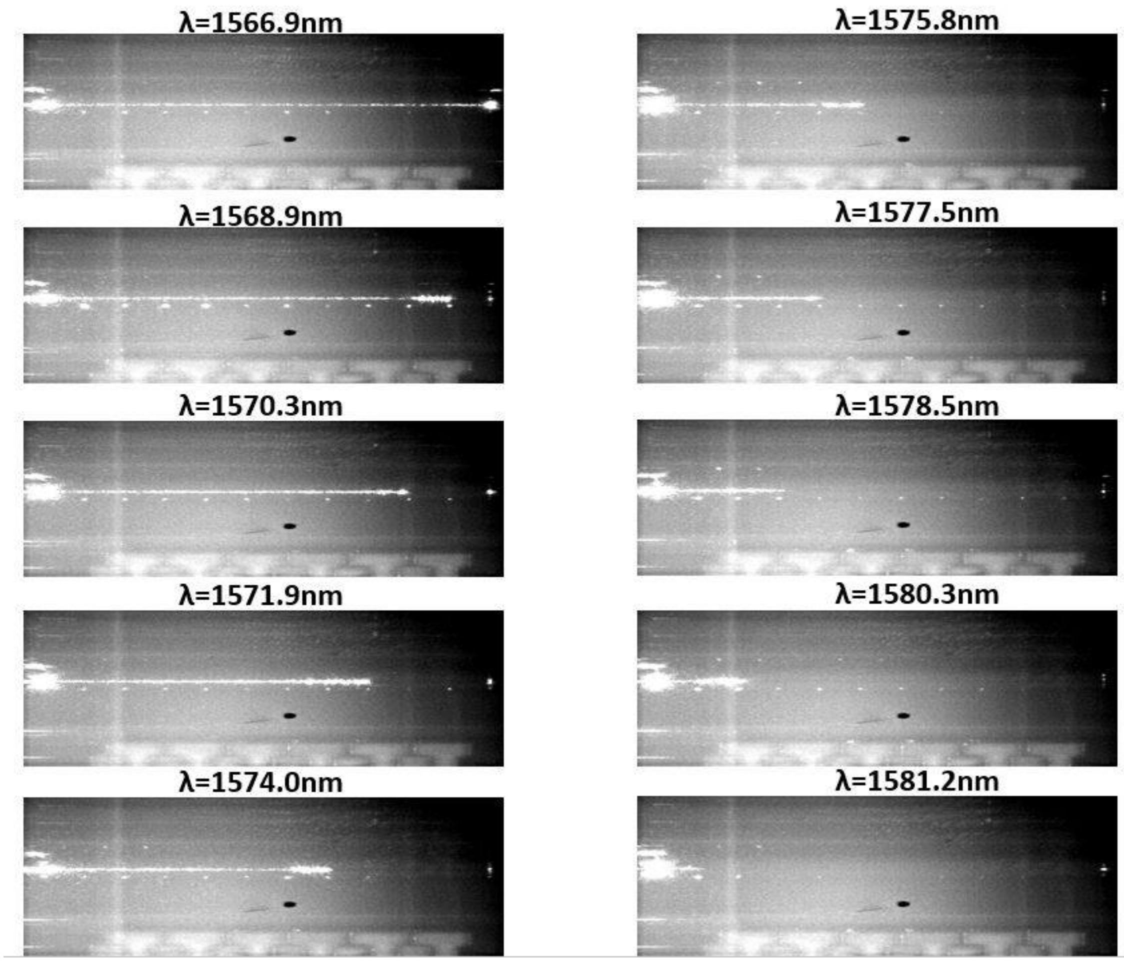

Figure (4.3): *Different signals at localization wavelengths (wavelength near mode gaps) for chip A*

Regarding chip B, 28000 images of single frequency signals were taken by doing a sweep between 1568nm and 1581.99 nm with 0.01 nm step repeated on 20 different power levels from −9 dBm to 10 dBm. 1680 Images of multiple frequency component signals were taken by fixing the first frequency component to a power of −7 dBm and respectively wavelengths equal to 1571.88 nm, 1573.93 nm, 1575.33 nm and 1579.17 nm. The second frequency component is swept from 1568 nm to 1581.9 nm with a 0.1 nm step. The different approximate values for the mode gap of each section was found through the same method as for chip A and were found to be respectively equal to 1568.17 nm,1570.22 nm,1571.74 nm,1573.20 nm, 1575.16 nm, 1576.64 nm,1578.32 nm,1579.67 nm,1580.42 nm and 1581.28 nm. Signals at the different mode gap wavelengths are shown in figure 4.4.

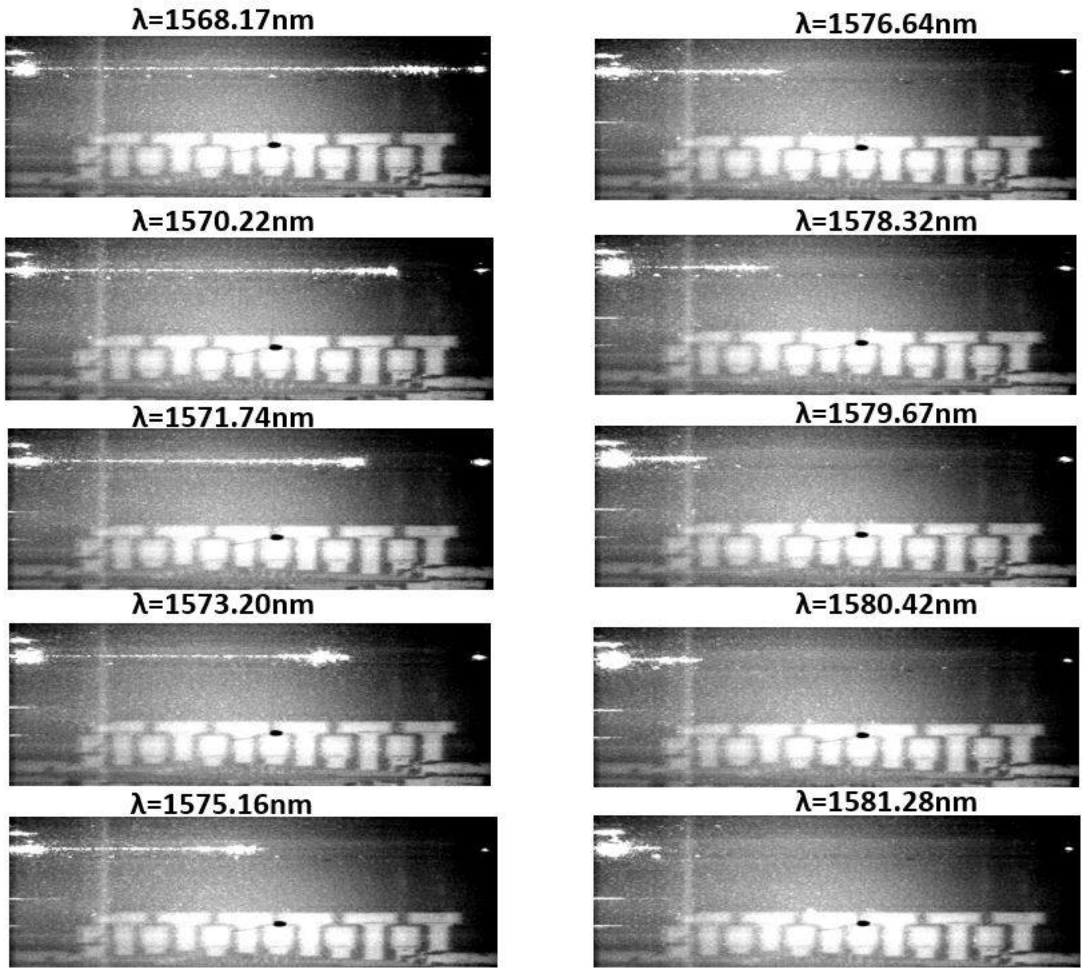

Figure (4.4): *Different signals at localization wavelengths (wavelength near mode gaps) for chip B*

Chapter 5

Wavelength And Power Detection: Methods, Results and Discussion

This chapter will talk about the methods used to perform wavelength prediction and power prediction given images of light scattering in our designed chirped PhC waveguide that were shown in the previous chapter. The final goal of this project is to perform spectrum reconstruction (i.e being able to predict all the wavelength component in a signal and their associated power), in order to progress towards that goal, we will first focus on a simpler version of this problem: Performing wavelength prediction for monochromatic light (composed of only one wavelength component). We will then see how we can perform power detection on monochromatic light before tackling the prediction of multiple wavelength component signal.

As it was explain in Chapter 4, we will base our wavelength prediction system on supervised learning, which means we will train our algorithm using training data represented by the collected data shown in the previous chapter. There is mainly two methods of tackling a problem using supervised learning: Classification or Regression. The main difference between both methods is how we are going to treat the data. In classification, the data has discrete values, we have a finite number of classes in which we will categorize the data. In regression, there are no class and the algorithm will assign to the input a continuous value. Since here we are working with a structure having discrete properties (due to the resolution limit of fabrication) and with data having discrete values (we work on a certain range of wavelengths with a step that defines the number of wavelengths we are going to work on),our data appears as discrete values data and we will therefore treat this problem as a classification problem. This work as been done using the "keras" library [4] in python that offer to design deep learning architectures in an easy and intuitive way.

5.1 Single wavelength detection

5.1.1 Pre-processing

Image cutting

Before starting to feed the data to the network for learning, it is important to pre-process the data to optimize the learning process. The first step is to cut the image as in figure 5.1 and 5.2 to isolate the part that is important, in our case, the important information is the position at which light stop propagating in the waveguide. Everything that is located outside the waveguide is thus useless information for the learning process. We therefore define the region of interest as being 7 pixels around the waveguide.

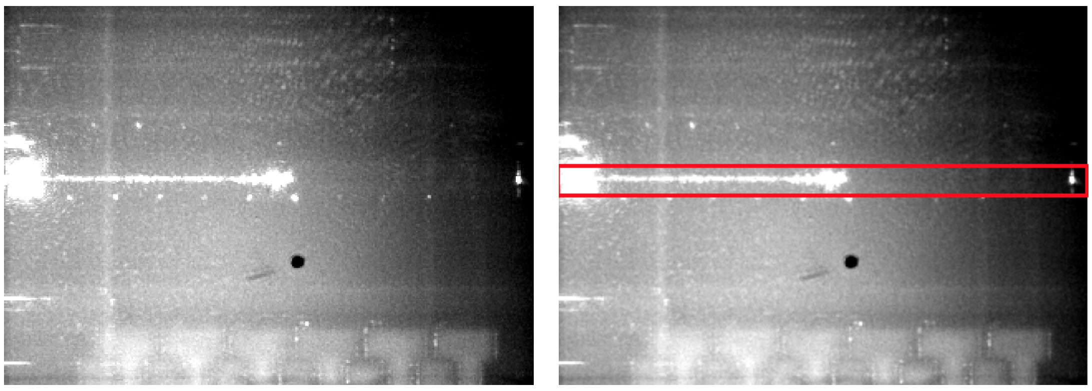

Figure (5.1): *Isolation of the region of interest for the image pattern of a* $-2dBm$ *power and* $1575nm$ *wavelength input light*

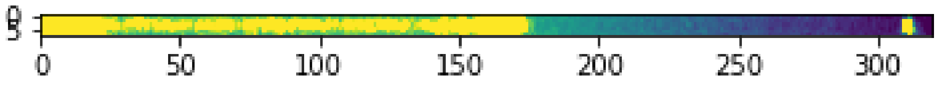

Figure (5.2): *Image obtained after cutting the input image around the waveguide, only keeping the region of interest*

Using only the region of interest for learning will allow the algorithm to learn faster since it directly focuses on the important features and also because the size of the data is significantly smaller (we go from a 256×320 image size to a 7×320 image size).

Removing background noise

Another step that will help optimizing the learning is to remove the noise from the background of the image. Indeed, we are not interested in the background but only on the pattern of the light in the waveguide. Since the pattern of the light in the waveguide is significantly brighter than the background, we will remove the background just by applying a filter that will only allow certain level of brightness in the image. By analyzing the value of the pixels in the image, we see that the pixels corresponding to the light pattern all have a value above 0.05 and the pixels from the background have a value way lower. We will thus apply a filter that will apply for each pixel value v of the image the following:

$$v = \begin{cases} v, & \text{if } v \geq 0.05 \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

The result after removing the background is shown in figure 5.3.

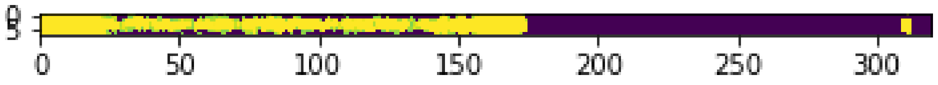

Figure (5.3): *Result after removing background*

This will be the data with which we will create our dataset.

5.1.2 Creating the dataset

After pre-processing, the data is organized into datasets using *.h5* files, which are files in the HDF5 format.[5] These files are binary data format files in which it is able to store big amounts of data and to manipulate it easily. It is possible to establish hierarchy and to organize data that allows us to regroup different datasets and to organize them all in a single file that is easily accessible in widely used programming languages like python.

The collected data is separated randomly into a *training set* (on which we will train our algorithm) and a *test set* (on which we will evaluate the performance of the algorithm) following a common rule of thumb regarding the proportions when we split our dataset: around 70% of the data will be used for training and 30% for testing [32]. We will respect this rule by taking a 75% and 25% ratio. In each dataset, the data is classified in an array with its corresponding value of wavelength and power as shown in figure 5.4.

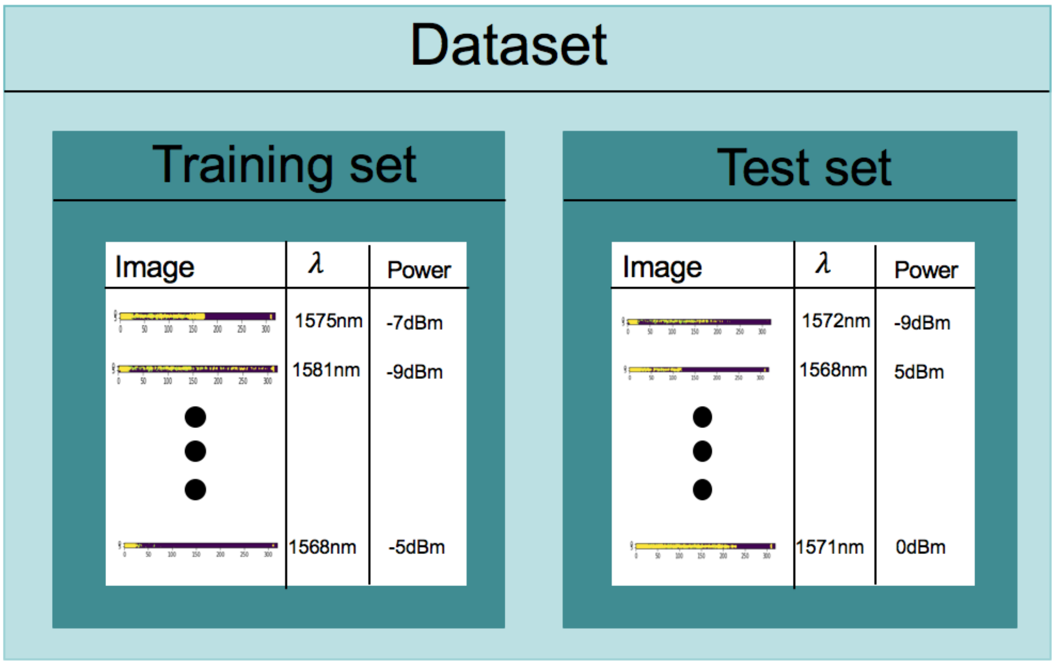

Figure (5.4): *Organization of the dataset*

Architecture and choice of parameters

We first have to decide on a deep learning architecture that we will use. Since we are going to process images, we choose to use a convolutional neural network as it will be more cost-effective computationally and it is the common solution when training on images. We start with the simplest architecture for our case: one convolutional layer followed by one fully connected layer. The activation function at the output will be a softmax function as it will give us the probabilities for the input to belong into each class. We will start by fixing the resolution of the system to 0.2nm. The resolution here is defined as the ratio between the wavelength range on which we operate and the number of wavelength we can detect. Here we will start by considering 100 different wavelength on a 20nm wavelength range which fixes the resolution to 0.2nm. Since we have 100 classes, the size of the output fully connected layer is 100 nodes. Starting with a simple architecture and progressing towards more complex architecture is the best way to find a simple architecture for which we can obtain good performance.
We choose as starting parameters: the learning rate $lr = 0.001$, the number of filters $N_f = 2$, the size of the filter $L = 3 \times 3$, the size of the pooling window $P = 2 \times 2$, the number of epochs is fixed at 500. Finding the right parameter combination that will give good performance is done by parameter tuning. Parameters are tuned one at a time, selecting the values that give the best results. Parameters were tuned, comparing the learning process by analysing the *learning curve*. The learning curve is the evolution of the accuracy on the training set through each iterations of the program on the training set. The accuracy on the training set is defined as the ratio between the number of good prediction in this set on the total number of predictions,

the accuracy on the test set will be defined in the same way but applied to the test set. This curve exhibits the learning behaviour of the model and constitute the main metric to evaluate how a model learns.

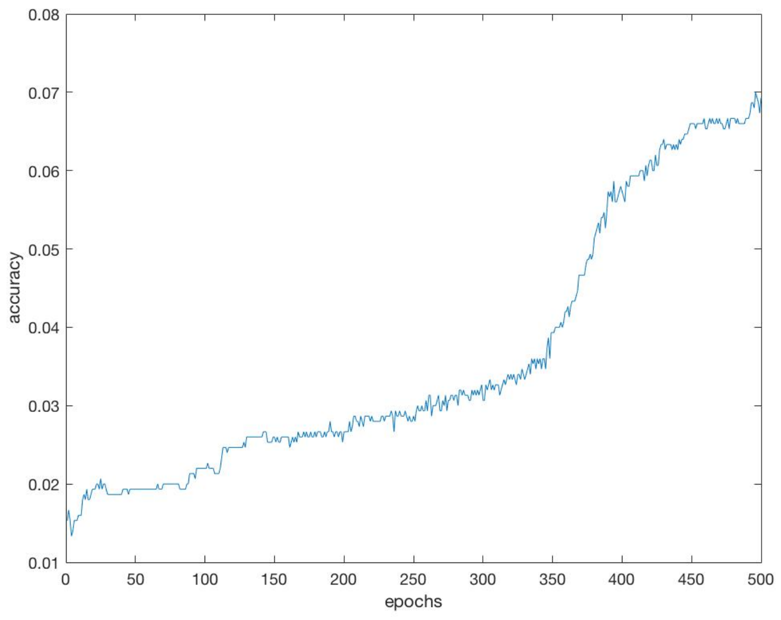

Figure (5.5): *Learning curve with starting parameter values*

Figure 5.5 shows the learning curve for the model with the starting parameters. We can see that the accuracy after 500 epochs is very low, even though we can observe that the algorithm is learning because we do observe an increase in the accuracy. To solve this problem we can try to increase the value of the learning rate to obtain faster learning process.

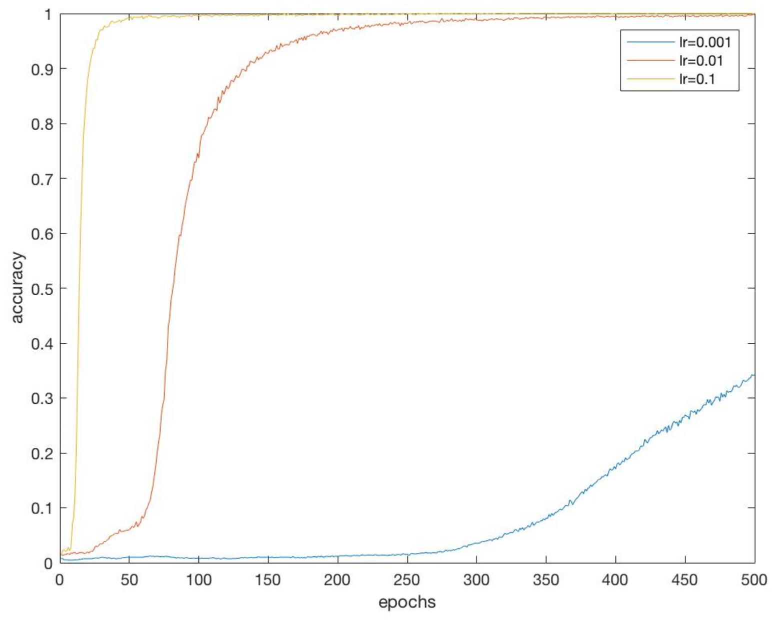

Figure (5.6): *Learning curve for learning rate values* $lr = 0.001$, $lr = 0.01$,$lr = 0.1$

Figure 5.6 shows the learning curve for different values of the learning rate. We can see that the low accuracy after 500 epochs in 5.5 is due to a low learning rate that results in very slow learning. Increasing lr speeds up the learning process and we are able to obtain rapidly satisfying values of accuracy on the training set. We will therefore fix $lr = 0.1$.

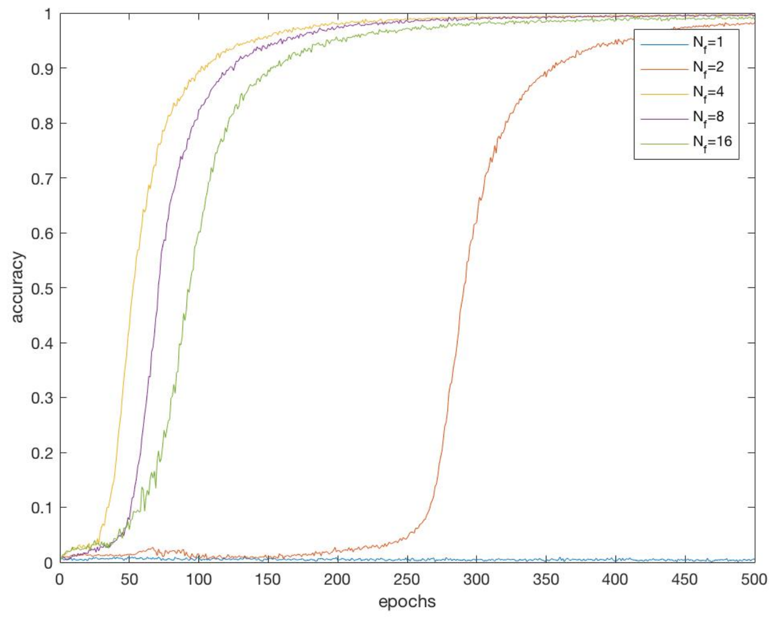

Figure (5.7): *Learning curve for number of filters* $N_f = 1$,$N_f = 2$,$N_f = 4$,$N_f = 8$,$N_f = 16$

The influence of the number of filter on the learning process is shown in figure 5.7. We will want to choose the smallest number of filter for which we will obtain satisfying training accuracy. We can observe that there is no learning for a filter size $N_f = 1$, the filter is too small to be able to observe any learning. For $N_f = 2$, the algorithm manages to learn properly after a certain amount of time (learning starts around 250 epochs), which is way later than the 3 other cases on which we will thus focus on. The three most interesting cases are for $N_f = 4$, $N_f = 8$, $N_f = 16$ which shows similar learning process. It is a bit faster for $N_f = 4$ but we will choose this value mainly to reduce the complexity of the system since it is twice and 4 times less filtering computations than for $N_f = 8$ and $N_f = 16$.

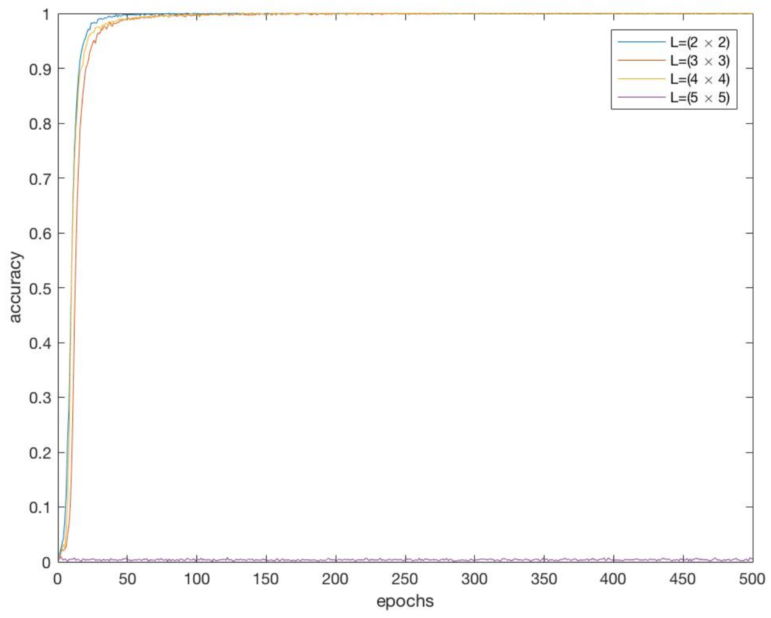

Figure (5.8): *Learning curve depending on filter size*

The impact of the filter size is represented in figure 5.8. We can see that as we increase the filter size, the algorithm is progressively less performant until the algorithm doesn't learn anymore for filters of size (5×5). This can be explained by the fact that smaller filter will highlight smaller features in the image and the algorithm will therefore be able to focus on smaller details. When the size of the filter becomes too big, the filtering process will not be able to highlight these small features. Filtering with bigger filter therefore leads to a loss of information that will cause the loss of features in the image that are important for the learning process. Since a smaller filters leads to more computations, we would want to choose the biggest filter for which we have satisfying results. Because the learning performances seem identical for sizes (2×2) , (3×3) and (4×4) , our choice for the filter size will therefore be filters of size (4×4).
Figure 5.9 shows the influence of the size of the Pooling filter. This figure shows that

this parameter has the same influence has the filter size. For the same reasons cited for the previous parameter, we will choose a pooling filter size of (4×4).

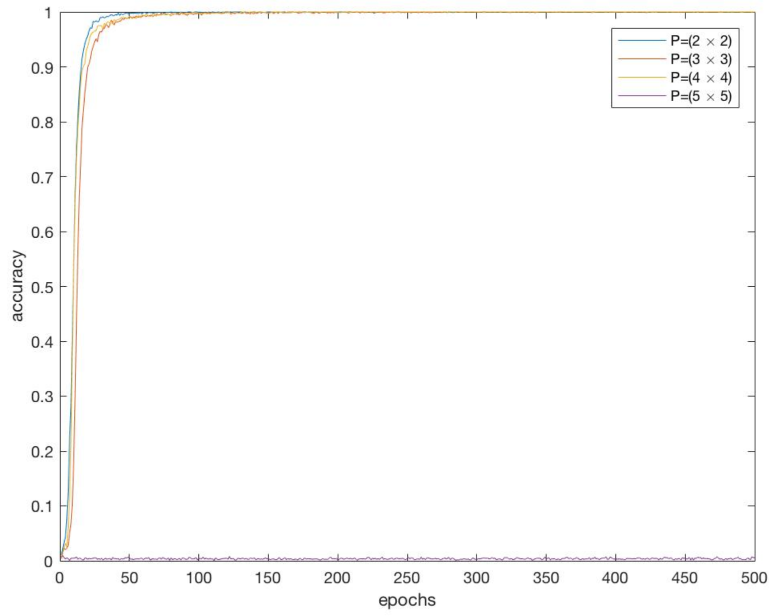

Figure (5.9): *Learning curve depending on the pooling filter size*

Proposed Architecture

After parameter tuning, the proposed architecture is thus as follows: $lr = 0.1$, $N_f = 4$, $L = (4 \times 4)$, $P = (4 \times 4)$.

The obtained learning curve shown in figure 5.10 shows that the model can achieve proper learning with 100 epochs.

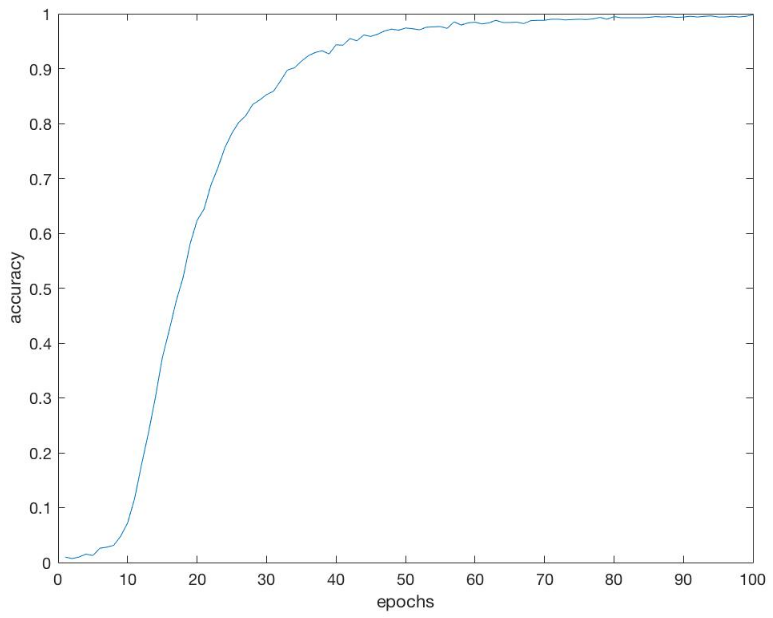

Figure (5.10): *Learning curve of the designed model*

This model was then evaluated on the test set and achieved an accuracy of 99% of correct prediction. This model is therefore validated for the prediction of single wavelength data in a range between 1565*nm* and 1584.8*nm* wavelength. This is an important result as it shows that we managed to obtain a very high accuracy with 0.2*nm* resolution. The resolution of fabrication of the PhC waveguide having a wavelength resolution of 1.5*nm*, we can see that we managed to go beyond this resolution limit that would, theoretically, not allow us to achieve higher resolution. This shows that it is possible, using deep learning, to take advantage of the intrinsic randomness of the PhC waveguide induced by the limitations in the fabrication process to increase the resolution of the system.

It is even possible to go beyond 0.2*nm* resolution. Indeed this model managed to achieved 97.7% accuracy on the test set on a 0.1*nm* resolution, which is increasing the resolution beyond the resolution limit imposed by the PhC structure by 15 folds. Even though there is a small drop in the accuracy on the test set due to the increase in the number of classes in the deep learning model (going from 100 classes to 200 classes), the accuracy is still very high and the model is applicable. The learning curve of this model is shown in figure 5.11.

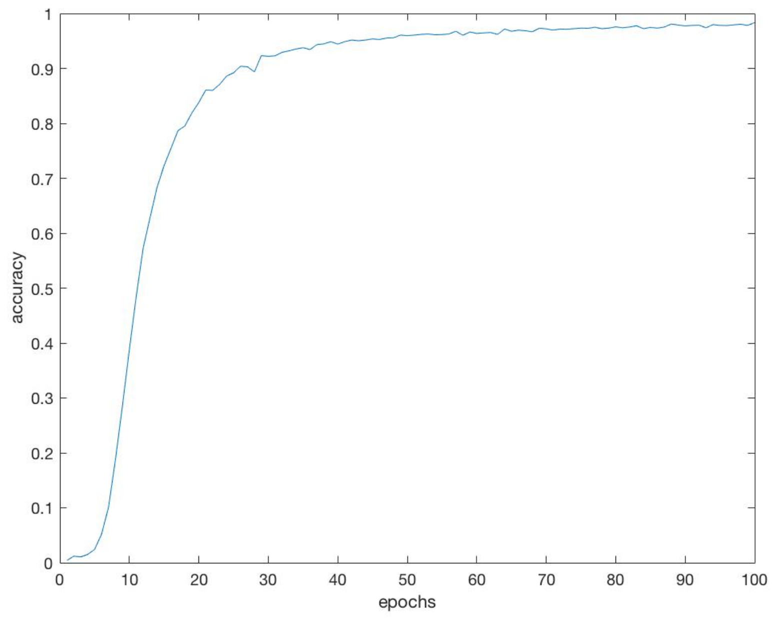

Figure (5.11): *Learning curve of the designed model on a* 0.1*nm resolution dataset*

5.1.3 Learning on 1 Dimensional data

Even though the model presented above presents impressive performances, it is interesting to find a way to simplify the model even further. The learning process can be very heavy in terms of computations. The computational complexity of a single layer can be estimated by looking at the number of operations performed in a single layer. The main operation used in a convolutional neural network layer is the matrix convolution which is given by:

$$y_{r,x,y} = \sum_{q=1}^{Q} \sum_{u=1}^{M_x} \sum_{v=1}^{N_w} w_{r,q,u,v} x_{x,q+u,y+v} \tag{5.2}$$

Where there are Q input X $(X_1 \dots X_q)$ of size $M_x \times N_x$ and R outputs Y $(Y_1 \dots Y_r)$ of size $M_y \times N_y$. [54] The number of operation in a matrix convolution is therefore given by QM_xN_w. The overall computational complexity of a convolutional neural network is , using the big O notation, given by

$$O(\sum_{l=1}^{d} n_{l-1} s_l^2 n_l m_l^2) \tag{5.3}$$

[23] with l the index of a convolutional layer, d the number of convolutional layers, n_l the number of filters in the l_{th} layer (n_{l-1} is also the number of input channels of the l_{th} layer), s_l is the size of the filter and m_l the size of the output. This means that the total execution time will asymptotically (i.e for large inputs) be proportional to $\sum_{l=1}^{d} n_{l-1} s_l^2 n_l m_l^2$. For large inputs the computational complexity can become very

large and it might be interesting to see if we can reduce the complexity while increasing the accuracy of the model. One method of reducing the complexity of the model is to perform learning on 1 dimensional inputs and thus using 1 a dimensional convolution network. In order to do, so we add more operations in the processing stage in order to convert the 2D data into 1D data.

filtering

The filtering stage includes two operations and is added in the pre-processing stage: The process of the image by an edge detection filter and vertical averaging.The role of the edge detection filter is to highlight features to help the learning process. Vertical averaging is done right after to reduce the data into 1D data. The input size image before vertical averaging is 7×320 and the output will be a vector having a size of 320. The vertical averaging process can be written has:

$$A_o[j] = \frac{1}{7}\sum_{i=0}^{7} A_i[i][j] \tag{5.4}$$

Where A_o is the output vector after averaging, A_i is the image just before averaging. Because we transform our 2 dimensional data into 1 dimensional data, there is an unavoidable loss in information. It is in order to compensate this that the edge detection filter is important because it helps highlighting the important features in the image in order to make the learning process more efficient.

An edge detection filter is a filter that highlights edges in an image, in other words strong variations between two regions in the image. In our case, it will highlight the places where light localizes importantly in the image which is the feature that interest us the most and that helps us improving the resolution beyond the resolution limit of the structure.

A common edge detection filter is the Sobel filter, this edge detection Sobel filter computes the gradient accross the image to exhibit variations in pixel values according to their position. It works on the assumption that the edges in the image occurs where there is a discontinuity in the pixel values or a very steep intensity gradient in the image. Using this assumption, it is possible to take the derivative of the pixel values across the image and find points where the derivative is maximum, which is where an edge will be located. The Sobel operator computes an approximation of the gradient of the image which corresponds to convolving the image with two filters: M_x that will calculate the x component gradient (which means it will detect vertical edges) and M_y that will calculate the y component gradient (which mean it will detect horizontal edges). [25] The expressions of these filters are given by:

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{5.5}$$

$$M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{5.6}$$

The sum of the outputs given by these two filters will highlight the edges of the input image. The result of edge detection filtering is shown in figures 5.12, 5.13 and 5.14.

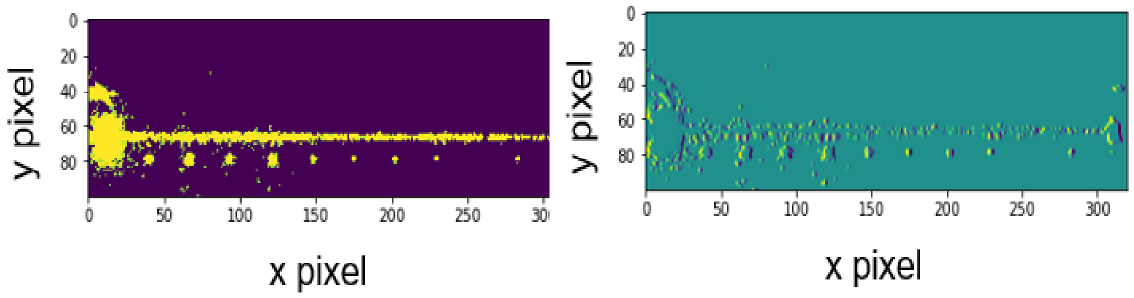

Figure (5.12): *Image for wavelength* 1568*nm before (left image) and after filtering (right image)*

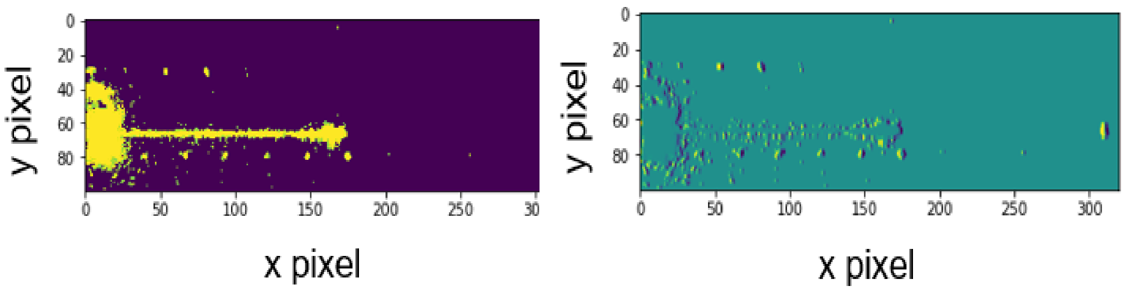

Figure (5.13): *Image for wavelength* 1575*nm before (left image) and after filtering (right image)*

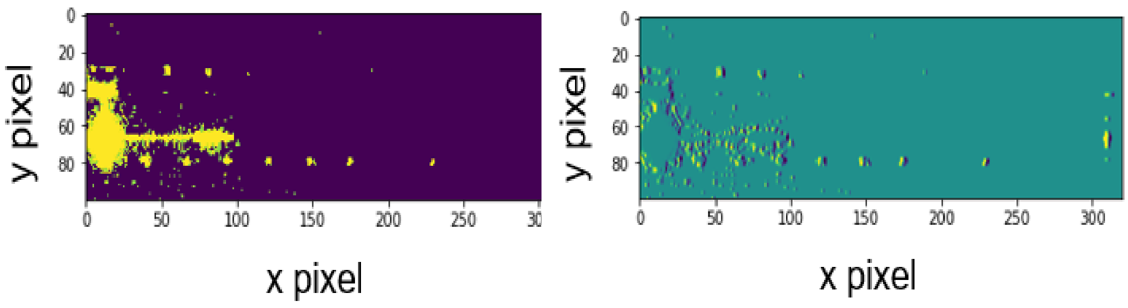

Figure (5.14): *Image for wavelength* 1579*nm before (left image) and after filtering (right image)*

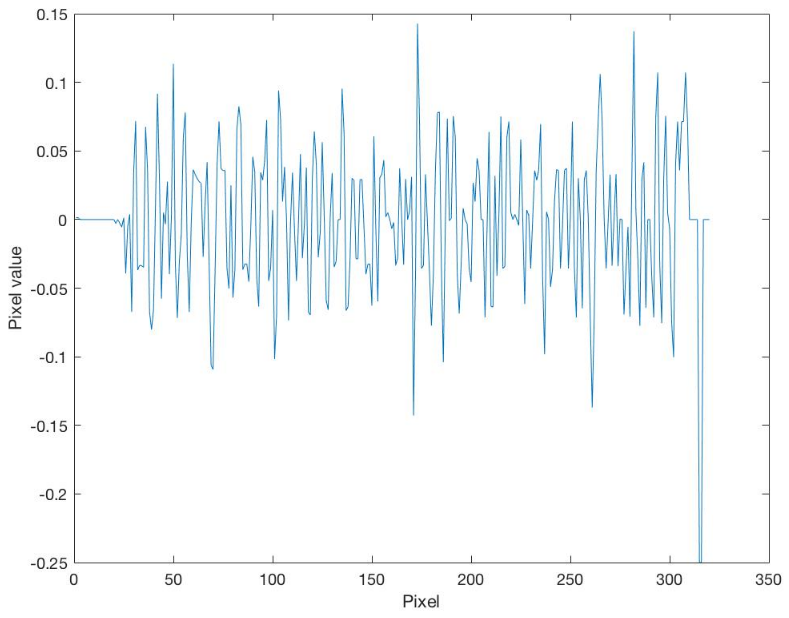

Figure (5.15): *Image for wavelength* 1568*nm after vertical averaging*

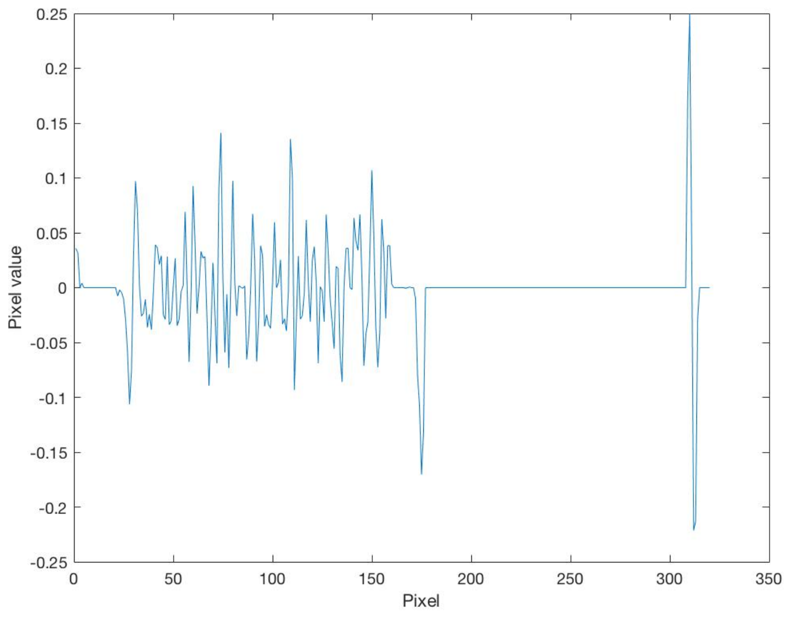

Figure (5.16): *Image for wavelength* 1575*nm after vertical averaging*

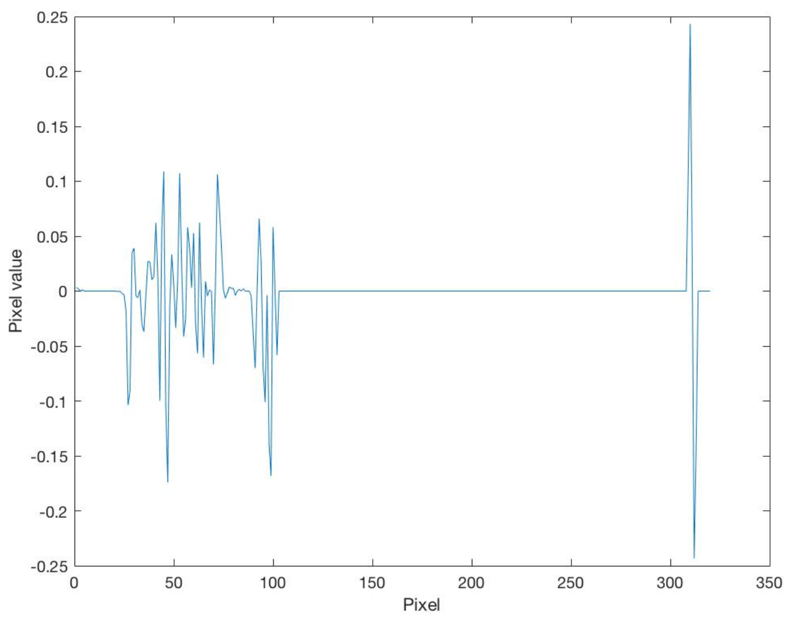

Figure (5.17): *Image for wavelength* 1579*nm after vertical averaging*

We can see from figures 5.15,5.16 and 5.17 that vertical averaging keeps information on the wavelength as we can spot the pixel at which light stops propagating. We can also notice that the peak around pixel 320 corresponds to light scattering at the output of the structure that we can see in the 2 dimensional images such as figure 5.1.
Figure 5.18 and 5.19 compares the learning process performance of the 1 dimensional data when a sobel filter is applied and when it is not. We can see that the model manages to learn even without an edge detection filter, but that the learning process is significantly enhanced when applying the sobel filter. This is also reflected in the accuracy on the test set. The model that doesn't involve any sobel filtering has an accuracy respectively of 90% and 88% on the test set for a wavelength resolution of 0.2*nm* and 0.1*nm* while the model that applies sobel filtering has a resolution of respectively 98% and 96%. This clearly shows that sobel filtering helps compensating for the drop in accuracy due to the loss of information induced by the conversion from 2 dimensional data to 1 dimensional data.

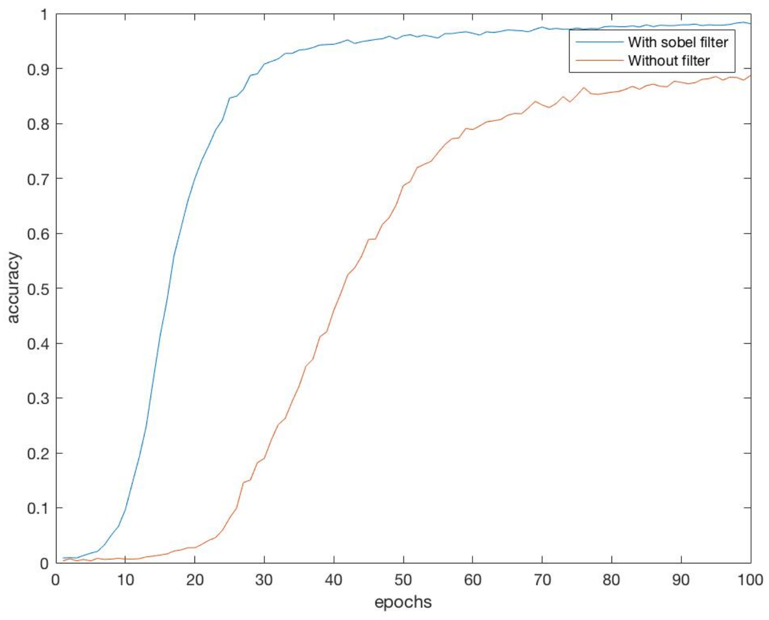

Figure (5.18): *Learning curve when applying sobel filter for a resolution of* 0.2*nm*

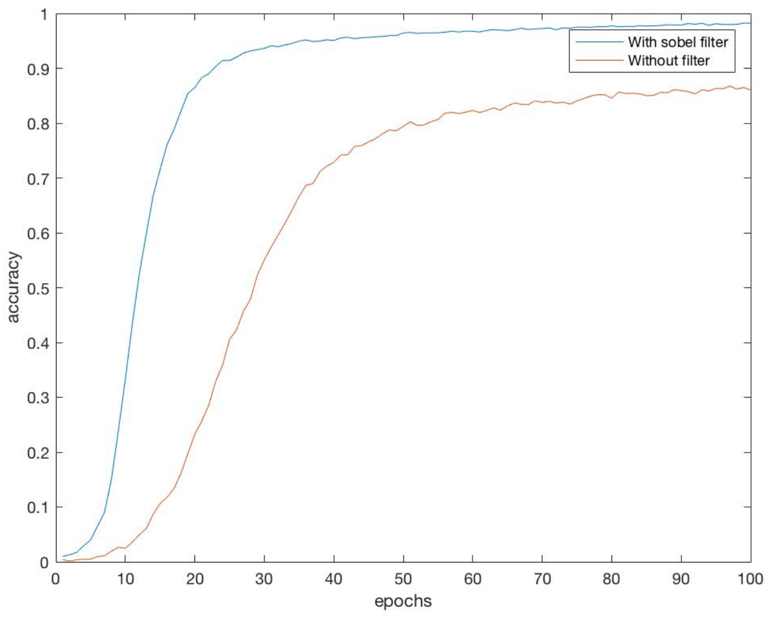

Figure (5.19): *Learning curve when applying sobel filter for a resolution of* 0.1*nm*

Comparison with 2 Dimensional Learning

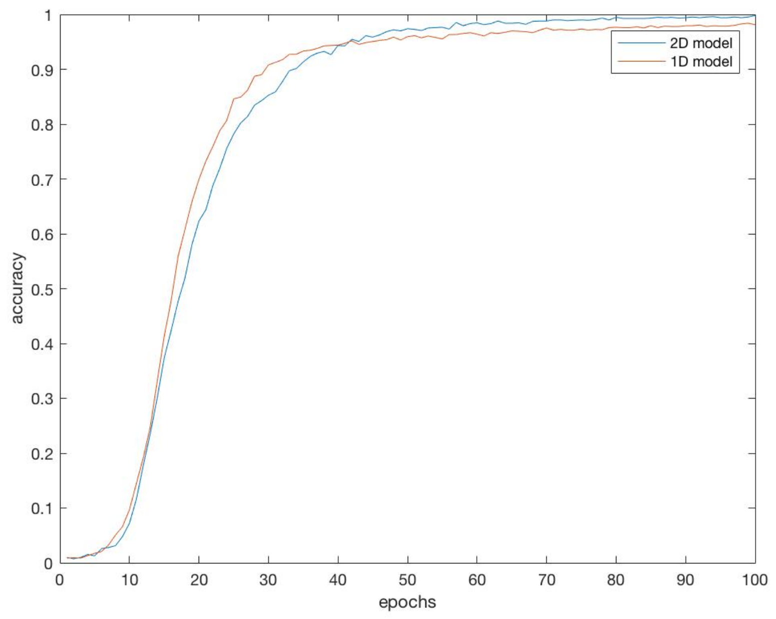

Figure (5.20): *Learning curve for 2D model and 1D model for a resolution of 2nm*

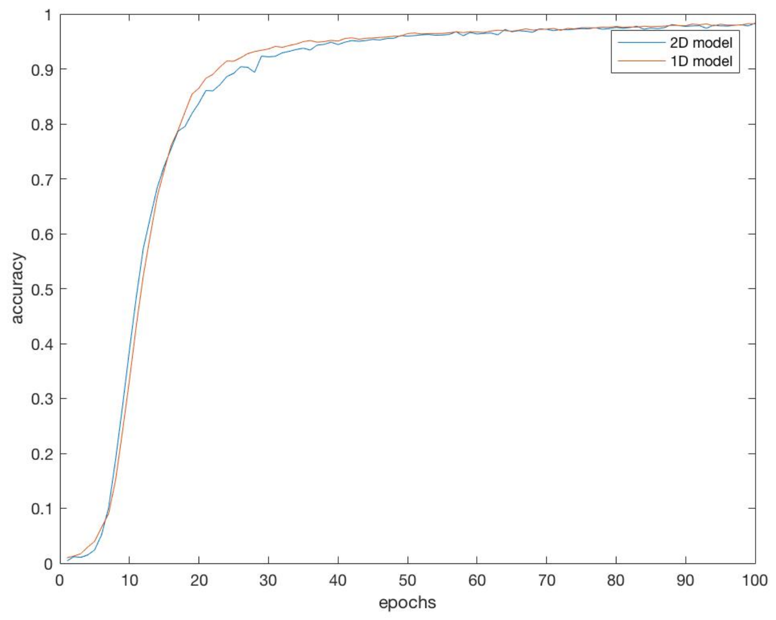

Figure (5.21): *Learning curve for 2D model and 1D model for a resolution of 1nm*

We can see from figure 5.20 and 5.21 that the learning process are similar for the 1 dimensional and the 2 dimensional case both 1nm and 2nm resolution . Either can therefore be used for application as they show good learning and satisfying accuracy on the test set. Using 1 dimensional data offer several advantages such as: lower memory requirements (each image becomes 7 times smaller) and faster program execution. Indeed the execution time will not be proportional to $\sum_{l=1}^{d} n_{l-1}s_l^2 n_l m_l^2$ as shown for the 2 dimensional case in equation 5.3 but will become:

$$O(\sum_{l=1}^{d} n_{l-1}s_l n_l m_l) \tag{5.7}$$

since s_l and m_l become vectors instead of square matrices of sizes $s_l \times s_l$ and $m_l \times m_l$.

5.2 Power detection for single wavelength input

A spectrometer is also supposed to be able to detect the power associated with each wavelength. While this research focuses more on the detection of wavelengths, it is interesting to see if it is possible, through the same method, to make predictions on the power of the input signal. We are going to concentrate in this section on the power prediction of a monochromatic signal (i.e a signal with only one wavelength component). The power detection will be applied to the same wavelength range as for the wavelength detection, which correspond to wavelengths between $1568nm$ and $1584nm$ with a $0.1nm$ step. The power levels are between $-9dBm$ and $10dBm$ with a $1dBm$ step (and therefore $1dBm$ resolution). Figures 5.22, 5.23 and 5.24 show examples of patterns for input signals with the same wavelength but different powers.

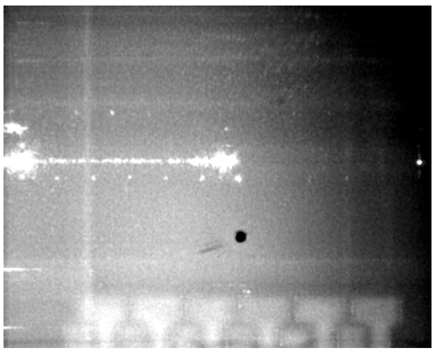

Figure (5.22): *$1575nm$ input light with $-5dbm$ power*

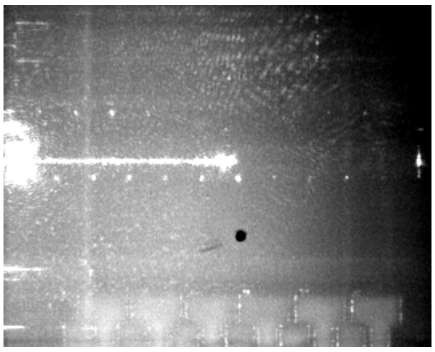

Figure (5.23): 1575*nm input light with* 0*dbm power*

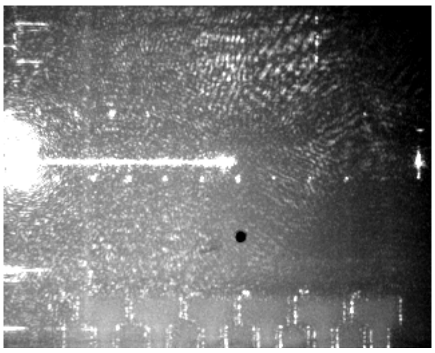

Figure (5.24): 1575*nm input light with* 5*dbm power*

We first try to use the same cut as for the wavelength detection (cutting around the waveguide).

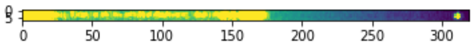

Figure (5.25): *Image for* 1575*nm input light with* −5*dbm power cut around the waveguide*

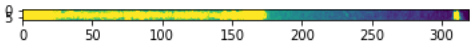

Figure (5.26): *Image for* 1575*nm input light with* 0*dbm power cut around the waveguide*

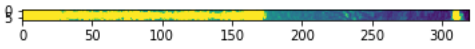

Figure (5.27): *Image for* 1575*nm input light with* 5*dbm power cut around the waveguide*

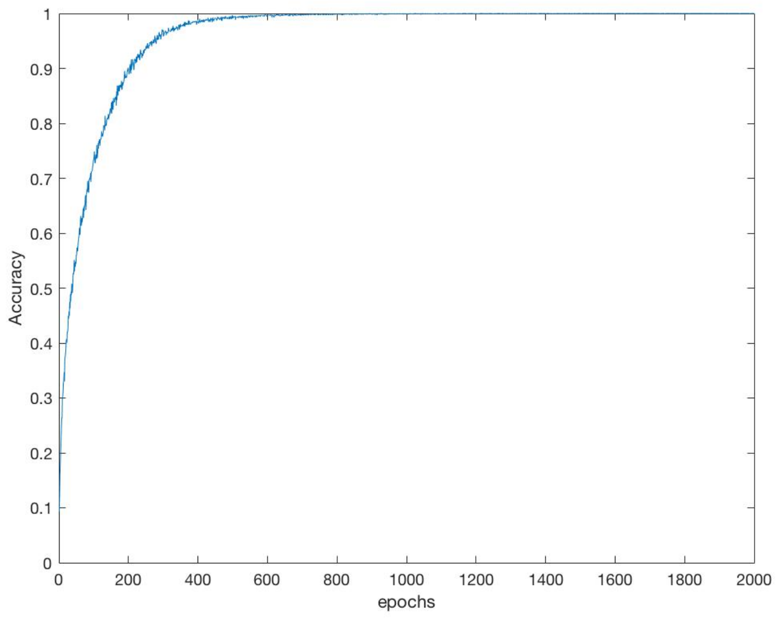

Figure (5.28): *Learning curve for power detection when the image is cut around the waveguide*

We can see from figure 5.28 that the algorithm manages to learn correctly and reaches a value of accuracy on the training set equal to 100%. Nevertheless, the accuracy on the test set is 42% which is very low. This big difference between accuracies of test set and training set can be interpreted as the model *overfitting* on the training set. The poor performances can be understood by the fact that the data for different powers are pretty similar as we can see in figures 5.25, 5.26 and 5.27. It seems more suitable to cut the image around the input of the structure as shown in figures5.29, 5.30 and 5.31 and taking advantage that light scatters at the input, the scattering being stronger for higher powers.

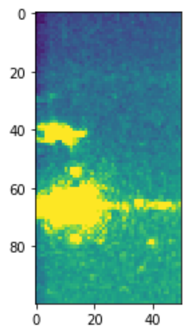

Figure (5.29): *Image for* 1575*nm input light with* −5*dbm power cut around the input*

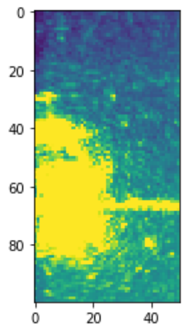

Figure (5.30): *Image for* 1575*nm input light with* 0*dbm power cut around the input*

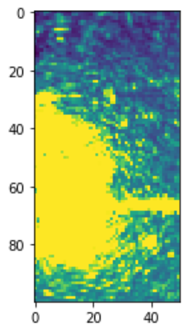

Figure (5.31): *Image for* 1575*nm input light with* 5*dbm power cut around the input*

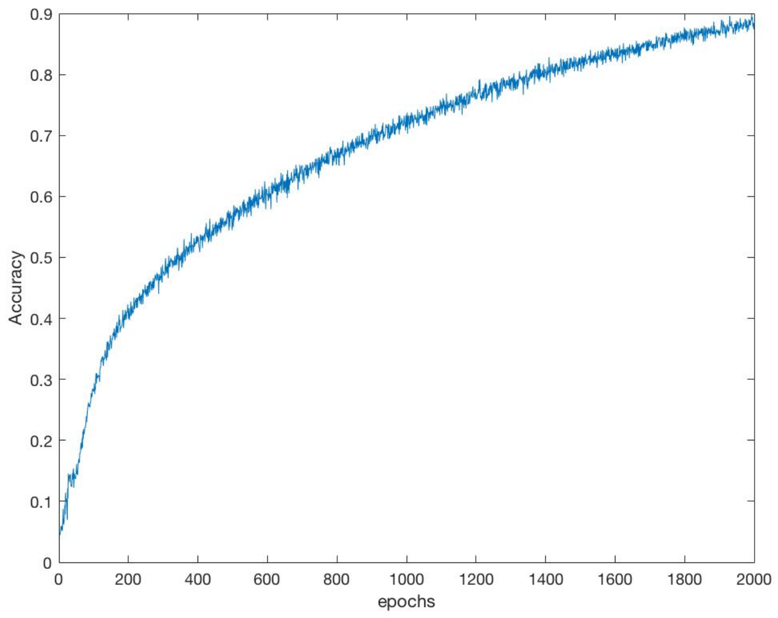

Figure (5.32): *Learning curve of the power detection for a 1dbm resolution*

We can see in figure 5.32 that the learning process seems very slow because the curve is still increasing after 2000 epochs and does not seem to saturate yet. The obtained accuracy on the test set was 80% which is way better than when cutting around the waveguide, but still not sufficient to be able to use it confidently in application.

5.3 Multiple wavelength detection

5.3.1 Modification of the architecture

There is one fundamental modification that has to be done in our architecture to be able to apply it to our architecture. For single wavelength data, we used *softmax* as an activation function, which is the most suited activation function for single label classification. When dealing with multiple wavelength detection, we have to be able to predict all the wavelengths in a signal, without knowing beforehand how many wavelength components are present in this signal. If we are predicting wavelengths in a range between wavelengths λ_1 and λ_2 with a resolution of n, the only information we have on the number of wavelength components in the signal is that it is located between 1 and $\frac{\lambda_2-\lambda_1}{n}$. We are therefore facing a *multi label classification problem* [49, 50] which means that, contrary to the single wavelength prediction case, one data can belong to multiple classes at the same time. For example, one signal having wavelength components λ_1,λ_2 and λ_3 will belong to these three classes. The first impact of this modification in the nature of the problem itself is that we have to change the activation function. Softmax outputs dependent probabilities between each classes and we have to select the highest probability as being the prediction, which makes it suitable for single label classification, but for multi label classification we need an activation function that would output independent probabilities since there are no dependence between the different classes in the way one piece of data would belong to different classes at the same time. We therefore choose to use the *sigmoid* activation function that outputs a confidence coefficient between 0 and 1 for each classes for the data to belong to these classes.

5.3.2 Selecting the number of wavelengths in the prediction

An important point to consider is: how do we select the number of wavelengths for our prediction since we don't know beforehand how many wavelength components are present in the signal? We have to decide for a threshold to be able to select the wavelengths that we will choose for our prediction. The naive method would be to set the threshold to 0.5 because the sigmoid function outputs for each possible class a confident coefficient between 0 and 1. But nothing indicates that this method is optimal or will give accurate predictions. Instead, we decide to use the metrics defined in chapter 4. The idea is to train our model with a training set and test it with a test set as for the single wavelength prediction. The difference is that afterwards we will compute for each test prediction a metric for different values of threshold, and choose the threshold corresponding to the highest value of this metric.

Hot encoded representation of label

Four metrics were introduced previously to evaluate a prediction: The accuracy, The precision, the recall and the FScore. These metrics are computed by counting the number of *true positives* (TP), *true negatives* (TN), *fake positives* (FP) and *fake negatives* (FN) of a prediction. This means that we have to represent our label as *hot encoded* values. Hot encoded values are values represented as a set of 0's and 1's. We therefore have to find a way to represent a vector of wavelength values into a vector of 0's and 1's. The way to proceed is to create a vector whose size is the same as the total number of classes (the total number of wavelength we can predict). Each position in this vector will correspond to a wavelength. All 1's in the vector will be located at a position where the corresponding wavelength component is present in the signal, the position of the 0's will indicate that the corresponding wavelengths are not in the signal. Let's take the example where we detect wavelengths between $1568nm$ and $1581nm$ with $1nm$ step, which means we can detect 14 wavelengths. This means that there are 14 different classes and our vector representing the label for this data will have 14 elements. The first of this vector will correspond to wavelength $1568nm$, the second $1569nm$ and so on up to the fourteenth position corresponding to $1581nm$. If our input signal has two wavelengths components at $1570nm$ and $1573nm$, this means that it will have a 1 at positions 3 and 6 and 0's at the other position.

$$\left[1570nm, 1573nm\right] \rightarrow \left[0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0\right]$$

$$\left[1568nm, 1581nm\right] \rightarrow \left[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1\right]$$

$$\left[1574nm\right] \rightarrow \left[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0\right]$$

Once the label of the prediction is transformed into a hot encoded label, we can compute the number of TP, TN, FP and FN for the prediction. Here, the number of TP will be the number of 1's correctly predicted, the number of TN will be the number of 0's correctly predicted, the number of FP will be the number of 1's badly predicted and the number of FN will be the number of 0's badly predicted. For example, if a signal has label $A = \left[1568nm, 1575nm\right]$ and we perform prediction $\hat{A} = \left[1568nm, 1574nm\right]$ (one of the predicted value is wrong), we obtain the following hot encoded representation:

$$A = \left[1568nm, 1575nm\right] \rightarrow \left[1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0\right]$$

$$\hat{A} = \left[1568nm, 1575nm\right] \rightarrow \left[1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0\right]$$

In this example, one 1 is correctly predicted, eleven 0's are correctly predicted, one 1 is badly predicted and one 0 is badly predicted, which gives the following values:

$$TP = 1$$

$$TN = 11$$

$$FP = 1$$

$$FN = 1$$

Using equations 3.8, 3.9, 3.10 and 3.11 we can compute the value for the four metrics:

$$Accuracy = 0.857$$

$$Precision = 0.5$$

$$Recall = 0.5$$

$$FScore = 0.5$$

Choice of metric to be used

We need to choose one metric that will help us selecting a threshold for our predictions. As explained previously, the Accuracy is suited in the case were there are similar amount of 0's and 1's in the hot encoded values, otherwise the accuracy will always reach very high values and will mislead us during our evaluation of the performance. In our case we will typically have a significantly bigger amount of 0's than 1's.
Our focus goes towards minimizing the number of mistakes rather than maximizing the number of correctly predicted values. We prefer for our application to miss a correct value, rather than predicting wrong values. Therefore, our priority goes towards minimizing the number of FP's and FN's. This can be done looking at equations 3.9 and 3.10 by maximizing the Precision and Recall. It is possible to maximize both by maximizing the FScore (equation 3.11 that takes both Precision and Recall into account).

In order to find a threshold, we will therefore compute the average FScore on the dedicated dataset for values of threshold from 0 to 1 with a 0.005 step and select the threshold giving the maximum FScore.

5.3.3 2nm resolution

We will first concentrate on the prediction of multi wavelength component signal with 2*nm* resolution. The same method of parameter tuning was used as for the prediction of single wavelength data, we will present here three models that were considered. Model A possess a similar architecture as for single wavelength detection:

1 convolutional layer with 4 filters of size 4×4 and a pooling layer of size 4×4., the learning rate is $lr = 0.1$ for model A and will be fixed to $lr = 0.05$ for models B and C. The different models are illustrated in figures 5.33, 5.34 and 5.35.

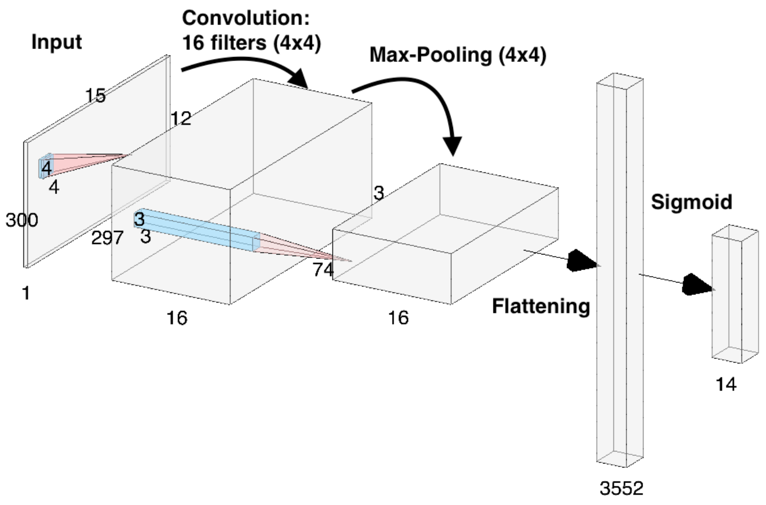

Figure (5.33): *Schematic of model A, learning rate is fixed to 0.1*

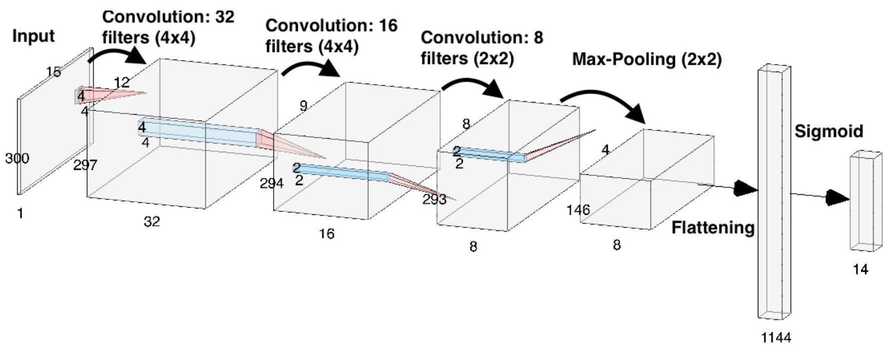

Figure (5.34): *Schematic of model B, learning rate is fixed to 0.05*

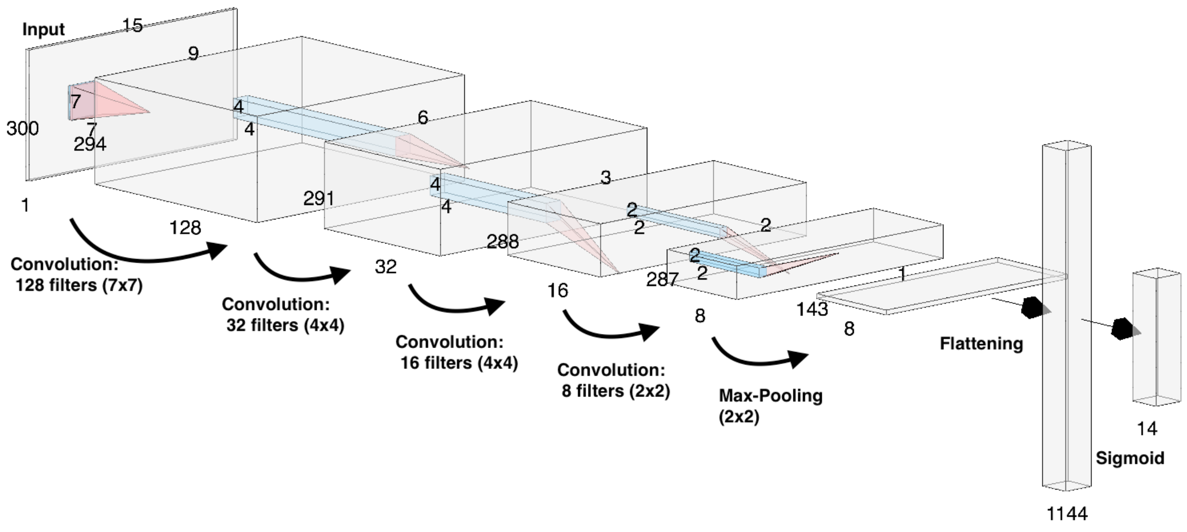
Input
15
9
7
300
7
294
1
128
Convolution: 128 filters (7x7)
4
4
6
291
Convolution: 32 filters (4x4)
32
4
4
3
288
Convolution: 16 filters (4x4)
16
2
2
2
287
Convolution: 8 filters (2x2)
8
2
2
143
Max-Pooling (2x2)
8
1
Flattening
1144
14
Sigmoid

Figure (5.35): *Schematic of model C, learning rate is fixed to 0.05*

All 3 models achieved learning in 100 epochs, the learnin process is illustrated in figures 5.36.

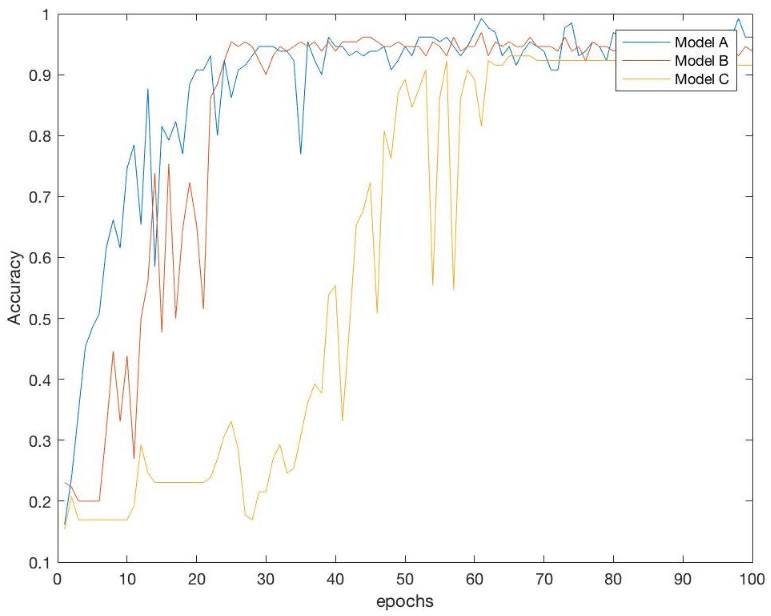

Figure (5.36): *Learning curve for models A,B and C*

As explained previously, we are going to compare the performance of the different models by looking at the average FScore on the test set for values of threshold between 0 and 1 and select the model that manages to achieve the highest average FScore. We are then going to choose as a threshold the value corresponding to the maximum FScore of the chosen model. The maximum average FScore and the corresponding maximum for models A, B and C are summarized in table 5.1.

Model	Maximum average FScore	Threshold corresponding to maximum
A	0.35	0.05
B	0.46	0.05
C	0.52	0.05

Table (5.1): *Performance of the different models.*

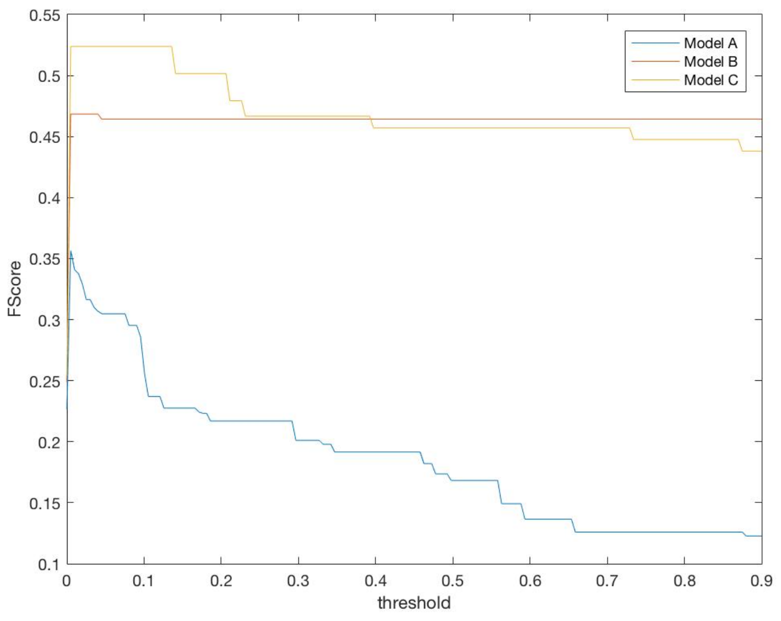

Figure (5.37): *Average FScore on validation set in function of the threshold for models A,B and C*

We can see that the model showing the best performance is model C because it has the highest maximum out of all three models as shown in figure 5.37. The corresponding average FScore is 0.52 and corresponds to a threshold of 0.05. Figure 5.38 shows the average maximum FScore on the dataset with regard to the threshold at the output of the network. We select as the threshold one which corresponds to the maximum of the curve. Figure 5.39 shows one example of the output of the model for a prediction of a signal containing wavelength components 1572nm and 1576nm, illustrating the method used to perform prediction using the computed threshold.

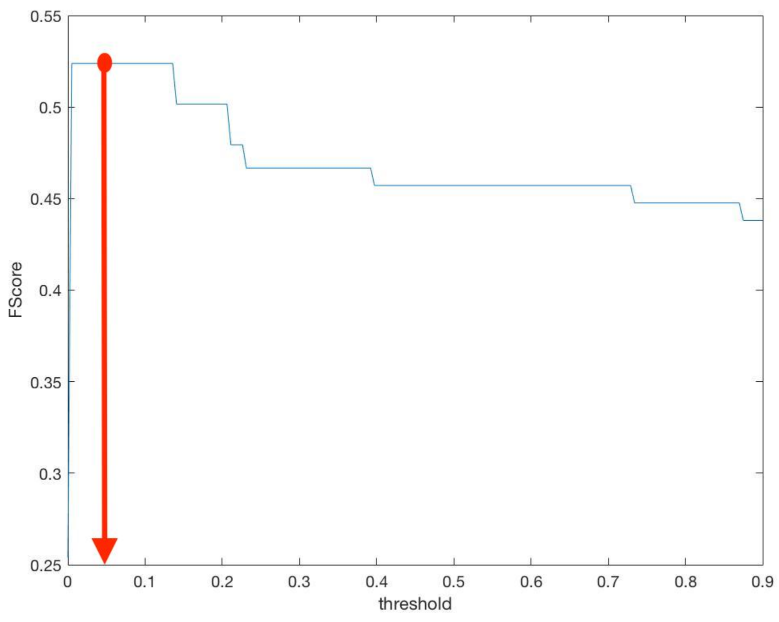

Figure (5.38): *Average FScore on validation set in function of the threshold for models C, we choose the threshold corresponding to the maximum of the curve*

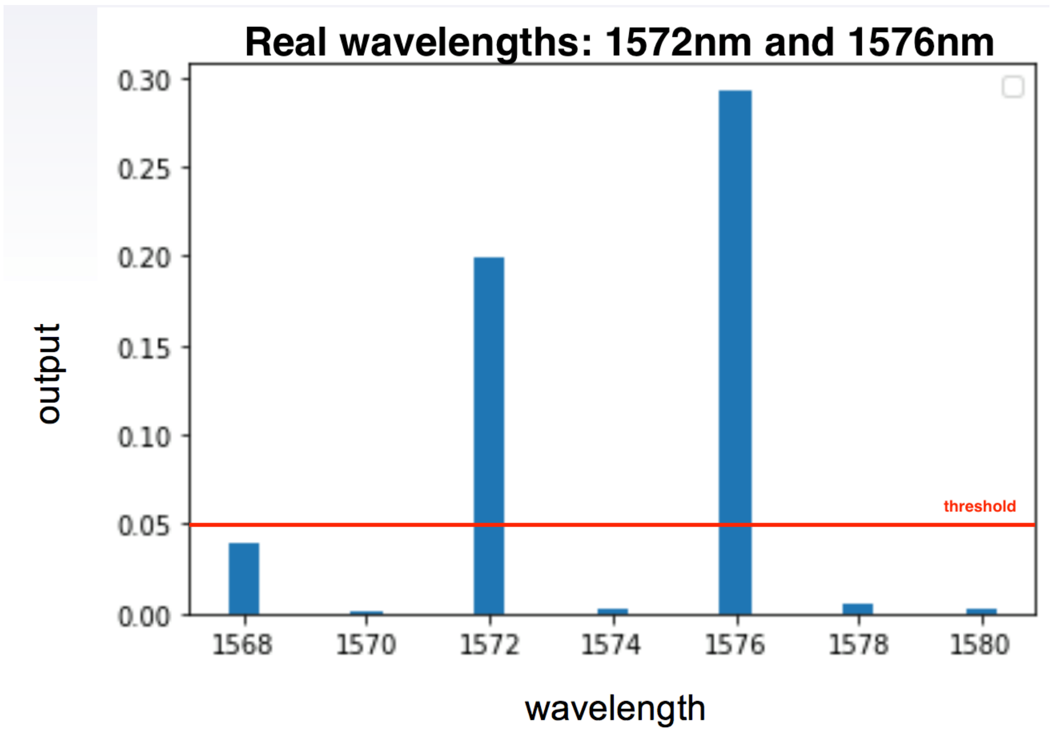

Figure (5.39): *Output of the network for a prediction of a signal containing wavelength components 1572nm and 1576nm. We can see that using the threshold previously found we can detect correctly the two wavelength components.*

	Correct Predictions	error on number of components	No wavelength predicted
Model A	61.3%	0.4	36%
Model B	0%	0.91	0%
Model C	88.6%	0.13	11%

Table (5.2): *Performance of the different models. The first column represents the number of correct prediction (correct wavelengths values and correct number of wavelength components), the second column represents the average error on the number of wavelength components and the last column represents the percentage of cases were no wavelengths are predicted.*

Using this model, we will therefore select as predictions all wavelengths that show a confident coefficient given by the sigmoid function higher than 0.05.

As we can see in table 5.2, the low FScore of model A can be explain by the very low percentage of correct predictions. Model B has a high number of True Positives (it predicts the wavelength values correctly) which his shown by the highest FScore than model B but the number of correct predictions is very low because it predicts most of the time too many wavelengths components, which is shown by its high average error on the number of wavelength components. Model C clearly stands out as the best model out of the three models. Considering the performances of model C, we will define a satisfying model as being a model with an FScore of at least 0.5.

When trying to improve the resolution up to 1nm using the same model, we obtain the learning curve and FScore-threshold curves given respectively by figure 5.40 and 5.41. The obtained threshold is 0.05 and the corresponding maximum FScore is 0.34.

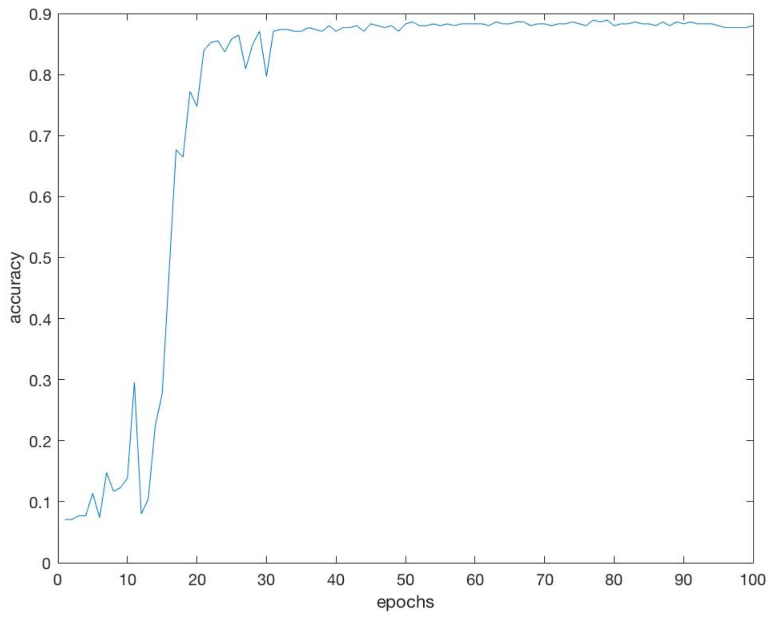

Figure (5.40): *Learning curve for a resolution of* 1*nm*

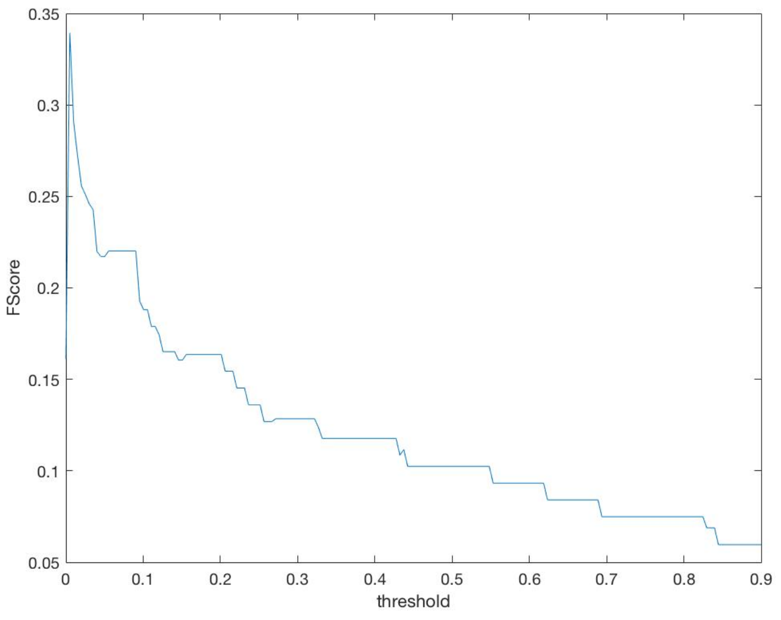

Figure (5.41): *Average FScore on validation set in function of the threshold for models C for a resolution of* 1*nm*

Keeping the resolution to 1nm, if we train the model on multi-wavelength component data whose wavelengths are separated by at least 2nm, the FScore significantly rises. The corresponding learning curve and FScore-threshold curve are respectively given by 5.49 and 5.43.

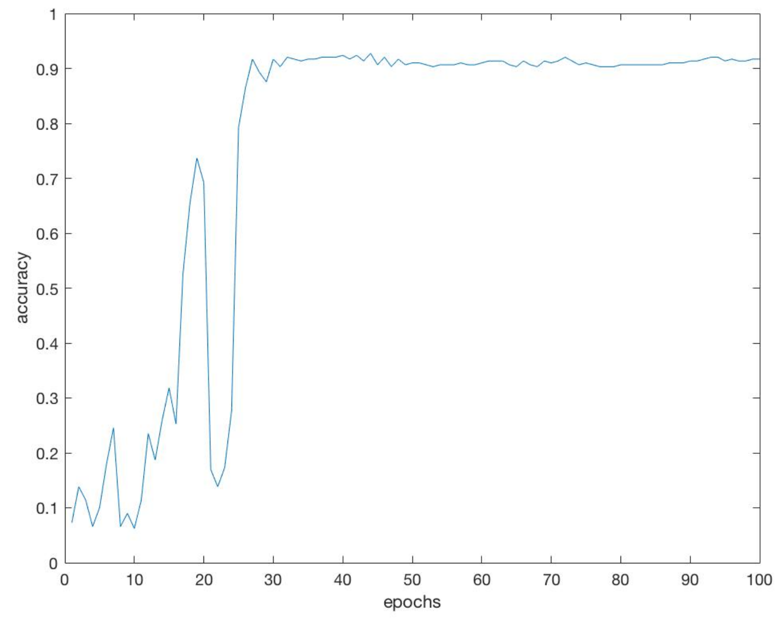

Figure (5.42): *Learning curve for model C after training on artificial data with a resolution of* 1*nm*

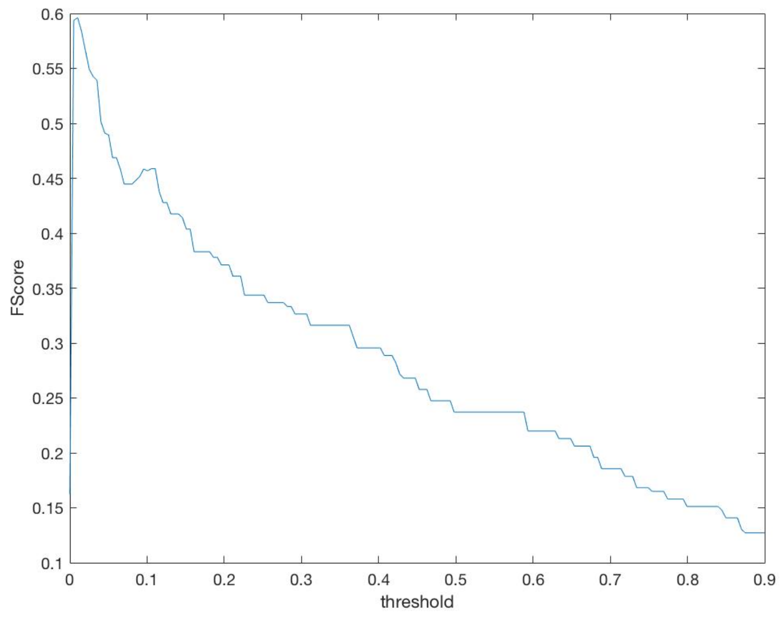

Figure (5.43): *FScore for 1nm resolution when trained and test with multi-wavelength component data whose wavelengths are separated by at least 2nm*

This can be explained by the fact that multiple wavelength data with wavelength components that are very close have their wavelength component that localizes in the same section of the chirped waveguide and makes the learning process more difficult as the resulted pattern is not just a linear superposition anymore.

As we did with the prediction of single wavelength component data, we can try to learn on 1 dimensional data by performing an additional filtering step in the preprocessing stage. We notice that the program doesn't manage to learn and mainly stays around an accuracy on the training set of 22%. With this model we are thus restrained to using 2D data for the prediction of multiple wavelength components data. The corresponding learning curve is given by figure 5.44.

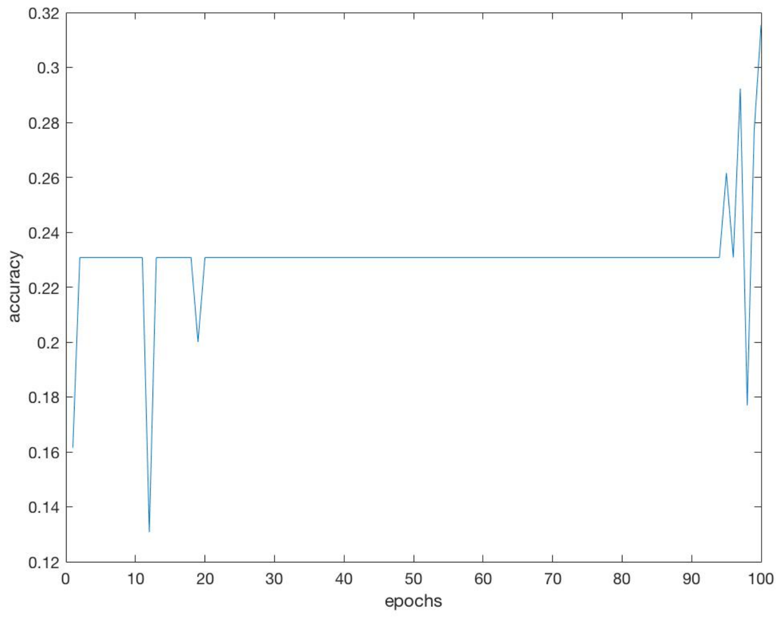

Figure (5.44): *Learning curve of model C applied to the 1 dimensional data*

5.3.4 Artificial Data

As it can be seen previously, the dataset for the prediction on multiple wavelength component signals is quite small, and it is strongly believed that increasing the size of the dataset would lead to an increase in the accuracy of the model because the dataset would be more representative of the data it could encounter. Nevertheless, collecting a large dataset of multiple wavelength component data can be quite difficult. Indeed, the number of combination between all the possible classes is very high. The number of combinations of pairs between signals for n wavelength on each of the k power levels taken is given by equation 5.8.

$$k^2\frac{(n-1)n}{2} + n\frac{(k-1)k}{2} \tag{5.8}$$

which for a dataset composed of 14 wavelengths over 19 powers corresponds to a dataset of 35245 elements. We can try to reduce this number by considering the fact that, since we will combine this dataset with single wavelength data we can ignore the combinations between elements that have the same wavelength component but different power, which leads to the expression in equation 5.9.

$$k^2\frac{(n-1)n}{2} \tag{5.9}$$

which for a dataset composed of 14 wavelengths over 19 powers corresponds to a dataset of 32851 elements, which still makes it unrealistic to collect this amount of data experimentally. The details of the computations to find equations 5.8and 5.9 can be found in the appendix. Furthermore, the number of possible combination would explode if we want to be able to perform prediction for signals containing more than 2 wavelength components. It is thus worth considering the generation of *simulated multiple wavelength data*, which is multiple wavelength data that we will be generated artificially based on the single wavelength component dataset. We thus generate multiple wavelength component data by simply superposing two images by adding them. An example of generation of multiple wavelength component data is represented in figure 5.45. The features are conserved compared to the corresponding data taken experimentally as shown in figure 5.46.

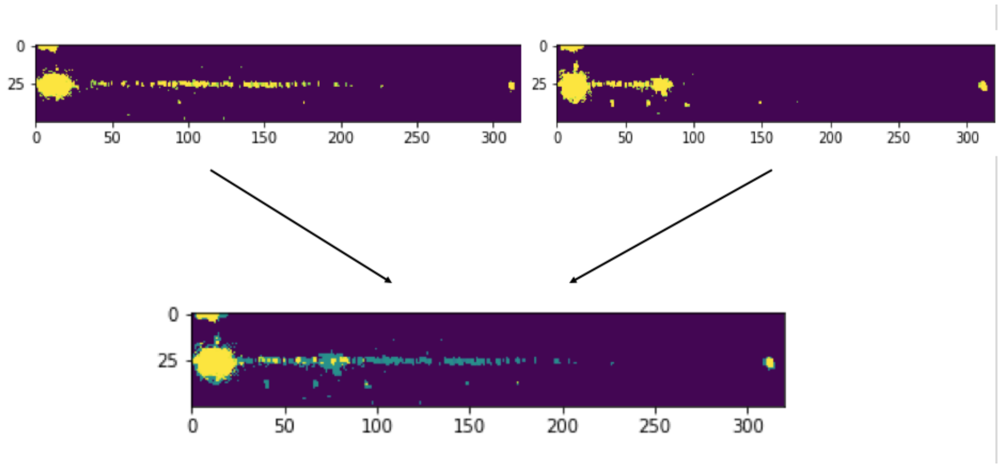

Figure (5.45): *Addition of images of two single wavelength data images to obtain computationally a representation of multi wavelength data*

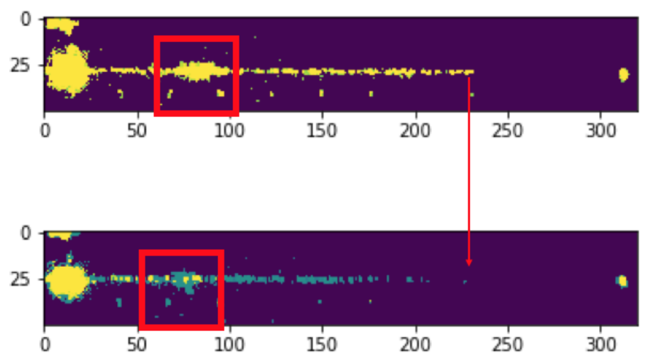

Figure (5.46): *Comparison between an image obtain experimentally by combining wavelengths around* 1579*nm at* −7*dBm and* 1571*nm at* −7*dBm and an image obtain computationally by addition between single wavelength data corresponding to these two wavelengths. We can see that we can observe important features that are similar for the two pieces of data: the light localization observed for the wavelength component around* 1579.17*nm and the spot where light stops localizing for the component at* 1571*nm*

By using this method, we can easily generate a dataset containing all possible combination of 2 wavelength components. The dataset that is used is composed of the computationally obtained multi wavelength data and the experimental single wavelength data. We manage to obtain a good learning process with a simpler model than the one we used with the experimental data as it is only composed of 2 convolutional layers as illustrated in figure 5.47.

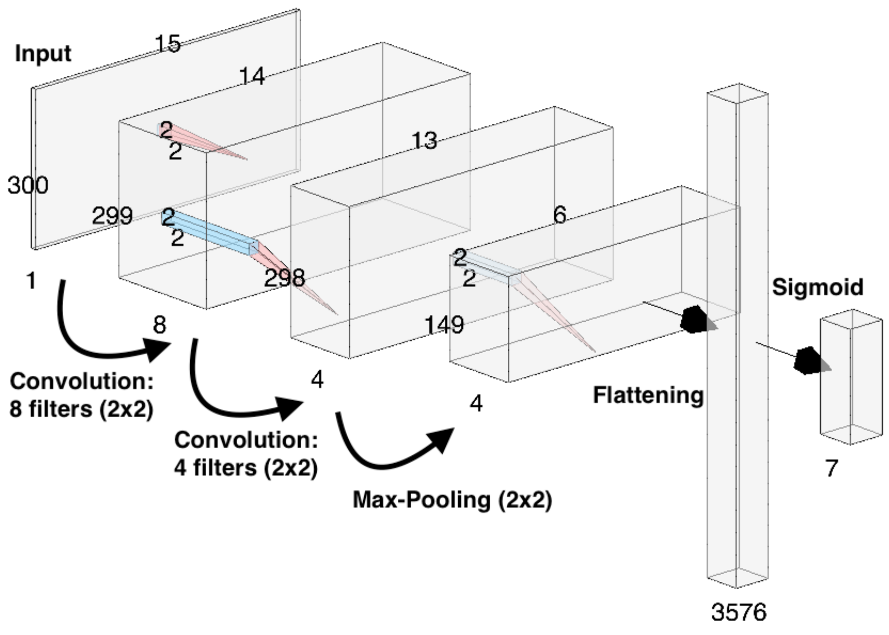

Figure (5.47): *Architecture used to train on artificial data*

We obtain a maximum FScore of which is comparable to the FScore previously obtained with experimental data. The FScore-threshold curve of the corresponding model for 1nm and 2nm resolution are given respectively in figure 5.48 and 5.48.

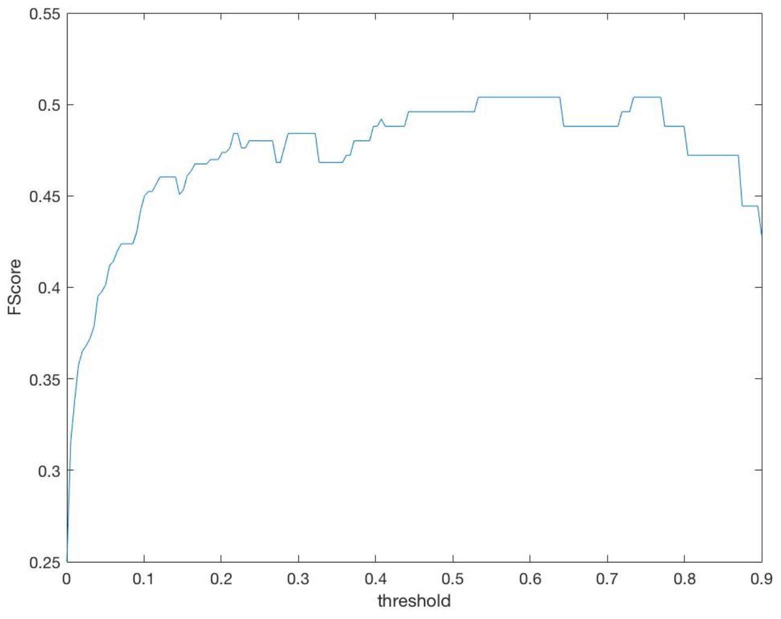

Figure (5.48): *Average FScore on validation set in function of the threshold for model C after training on artificial data with a resolution of 2nm*

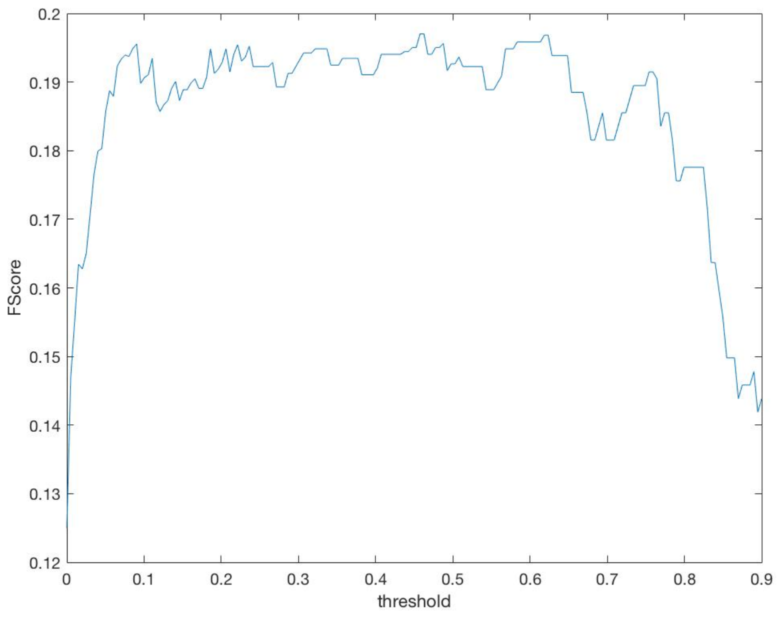

Figure (5.49): *Average FScore on validation set in function of the threshold for model C after training on artificial data with a resolution of 1nm*

Chapter 6

Summary and possible improvements

Several researches were made towards using photonic crystals to design an optical spectrometer in order to reduce significantly the size and cost of these devices. The main problems with the devices that were proposed was that they are either limited by fabrication errors or they are structures that fully rely on randomness in the structure which makes the pattern recognition more complicated and less intuitive.

In this work, we presented a structure that is meant to overcome both this two issues by relying on both order and disorder properties. Using a chirped photonic crystal waveguide we demonstrated that we can obtain high wavelength resolution by using deep learning. With this structure we managed to detect wavelengths between 1565 nm and 1585 nm with an outstanding 0.1 nm resolution, which is much higher (15 folds) than the structure resolution limited by the fabrication. Considering the very high obtained accuracy, it might even be possible to increase even further the resolution. This demonstration shows not only that fabrication randomness can be overcome, but also that it can even be actively used to enhance the performance of nanophotonic devices if they are combined with a good data processing approach such as deep learning. The accuracy obtain for this system is 97% which showed that we can achieve very high accuracy with a very simple Convolutional Neural Network (CNN) model having only one convolutional layer. By using more complex CNN models, it was possible to perform prediction on multiple wavelength components signal having up to 2 components with a rate of correct predictions on both the number of frequency components and correct frequency values of 88.7 % . We even showed that it was possible to increase the size of the dataset for multiple wavelength components prediction by computationally generate multiple wavelength components data based on the experimental single wavelength component data and perform prediction with performances similar as using experimental data, which makes the process of data acquisition easier since we just have to acquire single wavelength component data.

On the other hand, some limitations can be pointed out for such a system. First, given that the performances significantly decreased when going from single to multiple wavelength prediction (the resolution going from 0.1nm to more than 1nm and the rate of correct predictions dropping) we can imagine that performing predictions for more than 2 wavelength components will become very hard as we will always need bigger datasets and more complex models. Even though we can create bigger datasets by generating multiple wavelength data computationally, the size of the dataset will considerably increase when increasing the number of wavelength components, which is shown by equation 5.9.

Another possible limitation that was not discussed in this work is the influence of temperature on this device. The refractive index of a material changes with temperature [17]. A variation of temperature will therefore induce a variation of refractive index that could potentially modify the pattern that the input light exhibits in the structure. If the pattern are too temperature sensitive, it could be the case that a change in temperature would change the pattern and make it too different compared to the patterns the model learned from. It is thus worth to complete this work by investigating this possible issue.

Nevertheless, this innovative design of an optical spectrometer shows promising signs towards the design of high resolution, small footprint and low cost optical spectrometer devices.

Appendix A

Proof of the size of artificial dataset

In this appendix, we will demonstrate how to obtain equations 5.8 and 5.9. Let's assume a dataset composed of n frequencies for each of k power levels. We will note a_{ij} an element of this dataset having frequency i and power j. We can arrange this dataset in a matrix as follows:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ \vdots & \ddots & & \vdots \\ a_{n1} & \dots & \dots & a_{nk} \end{bmatrix} \tag{A.1}$$

Where every elements on the same row has the same frequency and every elements on the same column has the same power. We will first demonstrate equation 5.8 which results as answer teh following question: How many pairs (a_{ij}, a_{xy}) can we form such that $i \neq x$ and $j \neq y$.
We can observe that the first element a_{11} can combine with the $k-1$ elements on its own row and the $(n-1)k$ other elements in the matrix, it can therefore combine with $(n-1)k + (k-1)$ elements. The second element will combine with the $(n-1)k$ other elements in the matrix and the remaining $k-2$ elements on its row which corresponds to $(n-1)k + (k-2)$ pairs. By repeating the same process for all the elements on the first row we obtain a number of combinations containing an element of the first row equal to:

$$\sum_{i=1}^{k}((n-1)k + (i-1)) = (n-1)k^2 + \frac{k(k-1)}{2} \tag{A.2}$$

We can repeat the same process with the elements of row 2 that will combine with the elements on its own row and the remaining $n-2$ rows. The total number of combination containing at least one element of row 1 and 2 will therefore be $(n-1)k^2 + \frac{k(k-1)}{2} + (n-2)k^2 + \frac{k(k-1)}{2}$. Repeating this process for every rows, we obtain the number of distinct pairs we can make which is given by:

$$\sum_{i=1}^{n}((i-1)k^2 + \frac{k(k-1)}{2}) = k^2\frac{(n-1)n}{2} + n\frac{k(k-1)}{2} \tag{A.3}$$

To prove equation 5.9 we follow the same process but since we have to ignore the pairs having the same frequency, which means pairs of elements on the same line of matrix A, we have to ignore the term of combination between elements on the same line which correspond to the second term of equation A.3 which gives the following expression:

$$\sum_{i=1}^{n}(i-1)k^2 + \frac{k(k-1)}{2} = k^2\frac{(n-1)n}{2} \tag{A.4}$$

Bibliography

[1] 2020. URL: https://www.cisco.com.

[2] 2020. URL: http://www.santec.com.cn/.

[3] 2020. URL: https://www.keysight.com/en/pd-1279686-pn-81634B/low-polarization-dependence-optical-power-sensor?cc=JP&lc=jpn.

[4] 2020. URL: https://keras.io/api/.

[5] 2020. URL: https://www.h5py.org/.

[6] 2020. URL: https://numpy.org/doc/1.18/reference/index.html.

[7] 2020. URL: https://docs.scipy.org/doc/scipy/reference/.

[8] 2020. URL: http://www.avantes.ru/en/spectrometer/opticbanch/.

[9] *Advances in Agrophysical Research, p 347*. Intechopen, 2013.

[10] C. Jamois et al. "Silicon based two dimensional photonic crystal waveguides". In: *Photonics and Nanostructures, Fundamentals and Applications* 1 (2003).

[11] Bei Yang Heng Weng Aihua Ou Huixiao Hong Zhaoxian Zhou Ping Gong Andrew Maxwell Runzhi Li and Chaoyang Zhang. "Deep learning architectures for multi-label classification of intelligent health risk prediction". In: *BMC Bioinformatics* 18:523 (2017).

[12] *Artificial Neural Networks - Methodological Advances And Biomedical Applications*. InTech, 2011.

[13] Mehdi Asghari. "Silicon Photonics: A Low Cost Integration Platform for Datacom and Telecom Applications". In: *OSA publishing,National Fiber Optic Engineers Conference 2008* (2008).

[14] Michel Gadonna Azar Maalouf and Dominique Bosc. "An improvement in standard photolithography resolution based on Kirchhoff diffraction studies". In: *Journal of Physics D: Applied Physics* 42 (2009).

[15] R. Sarma B. Redding S. F. Liew and H. Cao. "Compact spectrometer based on a disordered photonic chip". In: *Nature Photonics, 7(9), 746–751* (2013).

[16] Hamza Kurt Berkay Neseli Emre Bor and Mirbek Turduev. "Transmission Enhanced Wavelength Demultiplexer Design Based on Photonic Crystal Waveguide with Gradually Varied Width". In: *ICTON 2019* 113 (2019).

[17] Timothy J. Madison Bradley J. Frey Douglas B. Leviton. “Temperature-dependent refractive index of silicon and germanium”. In: *Proc. SPIE 6273, Optomechanical Technologies for Astronomy, 62732J* (2006).

[18] John Canny. “A Computational Approach to Edge Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* Volume: PAMI-8 , Issue: 6 (1986).

[19] D. N. Chigrin and C. M. Sotomayor Torres. “Periodic thin-film interference filters as one-dimensional photonic crystals”. In: *Optics and Spectroscopy* 91, pages484–489 (2001).

[20] Stephen Bonner Daniel Justus John Brennan and Andrew Stephen McGough. “Predicting the Computational Cost of Deep Learning Models”. In: *IEEE International Conference on Big Data (Big Data)* (2018).

[21] Rosenblatt F. “The perceptron: a probabilistic model for information storage and organization in the brain”. In: *Psychol Rev* 65:386–408 (1958).

[22] Q. Dai H. Butt and T. D. Wilkinson. “Photonic Crystals and Metamaterial Filters Based on 2D Arrays of Silicon Nanopillars”. In: *Progress In Electromagnetics Research* 113 (2011).

[23] Kaiming He and Jian Sun. “Convolutional Neural Networks at Constrained Time Cost”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

[24] B. Ilic J. Topolancik and F. Vollmer. “Experimental Observation of Strong Photon Localization in Disordered Photonic Crystal Waveguides”. In: *Physical Review Letters* 99 (2007).

[25] O. Folorunso JO. R. Vincent. “A Descriptive Algorithm for Sobel Image Edge Detection”. In: *Proceedings of Informing Science and IT Education Conference (InSITE)* (2009).

[26] Sajeev John. “Strong localization of photons in certain disordered dielectric superlattices”. In: *Phys. Rev. Lett* 58, 2486 (1987).

[27] C. T. Chan K. M. Ho and C. M. Soukoulis. “Existence of a photonic gap in periodic dielectric structure”. In: *Phys. Rev.* Lett. 65, 3152 (1990).

[28] Controllable tune of the cutoff frequencies in a photonic crystal waveguide with hexagonal lattice. “ZHU KongTao, DENG TianSong, SUN Yan, ZHANG QiFeng and WU JinLei”. In: *SCIENCE CHINA Physics, Mechanics and Astronomy* 56 (2013).

[29] et al. L.C. Andreani. “Photonic bands and gap maps in a photonic crystal slab”. In: *IEEE Journal of Quantum Electronics* 38 , Issue: 7 (2002).

[30] Lin Y Zhao H Zhang C Li R Liu W. “An ensemble multilabel classification for disease risk prediction.” In: *Healthcare Engineering* (2017).

[31] Gao B. Siddiqui M. R. Shi Z. Liapis A. C. and R. W Boyd. “On-chip spectroscopy with thermally tuned high-Q photonic crystal cavities”. In: *Appl. Phys.Lett.* 108, 12–16 (2006). URL: https://aip.scitation.org/doi/abs/10.1063/1.4939659.

[32] Han Liu and Mihaela Cocea. “Semi-random partitioning of data into training and test sets in granular computing context”. In: *Granular Computing* 2 (2017).

[33] Bradley C. Love. “Comparing supervised and unsupervised category learning”. In: *Psychonomic Bulletin and Review* 9, 829–835 (2002).

[34] Francesco G.B. De Nataleby Marco Accame. “Edge detection by point classification of Canny filtered images”. In: *Signal Processing* Volume: 60, Issue: 1 (1997).

[35] D. Mori and T. Baba. “Chirped photonic crystal waveguides”. In: *Conference on Lasers and Electro-Optics, 2004. (CLEO)* vol.1, pp. 2 (2004).

[36] J. Jagerska V. Zabelin N. Le Thomas H. Zhang and R. Houdre. “Light transport regimes in slow light photonic crystal waveguides”. In: *Physical Review B* 80 (2009).

[37] *Neural Networks and Deep Learning*. Springer, 2011.

[38] C.R. Davidson N.S. Bergano . “Wavelength division multiplexing in long-haul transmission systems”. In: *IEEE, Journal of Lightwave Technology* 14 , Issue: 6 (1996).

[39] S. Stobbe P. D. Garcia S. Smolka and P. Lodahl. “Density of states controls Anderson localization in disordered photonic crystal waveguides”. In: *Physical Review B* 82 (2010).

[40] Sankar K. Pal and Sushmita Mitra. “Multilayer Perceptron, Fuzzy sets and Classification”. In: *IEEE transactions on neural networks* vol.3 No 5 (1992).

[41] *Photonic Crystals: Molding the Flow of Light*. Princeton University Press, 2007.

[42] et al. R.D. Meade K.D. Brommer. “Existence of a photonic band gap in two dimensions”. In: *Appl. Phys. Lett.* 61, 495. (1992).

[43] Williams RJ Rumelhart DE Hinton GE. “Learning internal representations by error propagation”. In: *Parallel Distrib. Process. Explor. Microstruct. Cogn* 1 (1986).

[44] Ashish Verma Rupika Rana. “Comparison and Enhancement of Digital Image by Using Canny Filter and Sobel Filter”. In: *IOSR Journal of Computer Engineering (IOSR-JCE)* Volume 16, Issue 1 (2014).

[45] M. I. Faruk N. K. Dankadai S. Babani A. A. Bature. “Comparative Study Between Fiber Optic And Copper In Communication Link”. In: *International Journal of Technical Research and Applications* 2, Issue 2 (2014).

[46] Pierre R. Villeneuve J. D. Joannopoulos Steven G. Johnson Shanhui Fan and L. A. Kolodziejski. “Guided modes in photonic crystal slabs”. In: *Phys. Rev. B* 60, 5751 (1999).

[47] Shanhui Fan Steven G. Johnson Pierre R. Villeneuve and J.D Joannopoulos. "Linear waveguides in photonic-crystals slabs". In: *Physical Review B* 62 (2000).

[48] S. Tamura and M. Tateishi. "Capabilities of a four-layered feedforward neural network: four layers versus three". In: *IEEE Transactions on Neural Networks* vol. 8, no. 2, pp. 251-255 (1997).

[49] Katakis I Tsoumakas G. "Multi-label classification: an overview". In: *Int J Data Warehous Min* 3:1-13 (2007).

[50] Vlahavas I Tsoumakas G Katakis I. "Random k-labelsets: an ensemble method for multilabel classification". In: *Mach. Learn* (2007).

[51] B. Widrow and M. A. Lehr. "30 years of adaptive neural networks: perceptron, Madaline, and backpropagation". In: *Proceedings of the IEEE* vol. 78, no. 9, pp. 1415-1442, (1990).

[52] Y. Wang F. Lombardi Y. Liu S. Liu and J. Han. "A Stochastic Computational Multi-Layer Perceptron with Backward Propagation". In: *IEEE Transactions on Computers* vol. 67, no. 9, pp. 1273-1286 (2018).

[53] E. Yablonovitch. "Inhibited spontaneous emission in solid-state physics and electronics". In: *Phys. Rev.* Lett. 58, 2950 (1987).

[54] Leiou Wang Yulin Zhao Donghui Wang and Peng Liu. "A Faster Algorithm for Reducing the Computational Complexity of Convolutional Neural Networks". In: *Algorithms* 11 (2018).

[55] Tomohiro Tetsumoto Yuta Ooka Nurul Ashikin Binti Daud and Takasumi Tanabe. "Compact resonant electro-optic modulator using randomness of a photonic crystal waveguide". In: *Optical Society of America* (2016).

[56] Lin Zhang and Changxi Yang. "Photonic crystal fibers with squeezed hexagonal lattice". In: *Optics Express* 12, Issue 11, pp. 2371-2376 (2004).