

POLITECNICO DI MILANO  
Corso di Laurea Magistrale in Ingegneria Informatica  
Dipartimento di Elettronica e Informazione



# A Deep Learning Approach to Sunspot Detection and Counting

AI & R Lab  
Laboratorio di Intelligenza Artificiale  
e Robotica del Politecnico di Milano

Relatore: Prof. Matteo Matteucci

Tesi di Laurea di:  
Enrico Fini, matricola 860761

Anno Accademico 2017-2018



*A papà e mamma*



# Sommario

La forte influenza del Sole sull'ambiente terrestre rende infatti necessario monitorare e prevedere la sua attività. Le macchie solari, manifestazioni di forti perturbazioni nel campo magnetico del Sole, sono una delle caratteristiche visibili che possono essere studiate per modellizzare i cicli solari. Finora, il conteggio delle macchie solari è stato per lo più eseguito da esseri umani e la comunità scientifica sembra riluttante all'utilizzo di algoritmi che, se introdotti, creerebbero discontinuità con i metodi di osservazione tradizionali. Lo scopo di questa tesi, che combina elementi di fisica solare osservativa ed informatica all'avanguardia, è dimostrare che, utilizzando il deep learning, è possibile costruire un programma che, se opportunamente addestrato, è in grado di apprendere da scienziati esperti ed eseguire automaticamente l'annotazione di immagini solari secondo criteri umani. Alcuni test sono stati progettati per valutare la qualità delle soluzioni proposte dal programma, rispetto alle prestazioni umane medie. I risultati sono promettenti e mostrano che l'algoritmo riesce a cogliere l'andamento del ciclo solare, rendendolo un valido strumento per la stima dell'attività del Sole.



# Abstract

The strong influence of the Sun on the environment of the Earth makes it necessary to monitor and predict its activity. Sunspots, manifestations of strong perturbations in the magnetic field of the Sun, are one of the visible features that can be studied in order to model solar activity cycles. So far, sunspot counting has been mostly done by humans and the scientific community seems reluctant to the introduction of algorithms because they would create a discontinuity with traditional observations. The purpose of this thesis, which lays at the intersection of observational solar physics and cutting-edge computer science, is to demonstrate that, using deep learning, it is possible to build a program capable of learning from expert scientists and performing solar image annotation automatically, according to human criteria. Test cases were designed to assess the quality of our solution with respect to average human performance. The results are promising and show that the algorithm can capture the progress of the solar cycle, making it a good tool for the estimation of the activity of the Sun.





# Ringraziamenti

Ringrazio i miei genitori per avermi dato tutto quello che potevano. Sono loro che mi hanno permesso di fare le esperienze all'estero che hanno reso questa laurea magistrale molto più divertente.

Ringrazio il Prof. Matteucci per avermi rassicurato e incoraggiato nei momenti in cui pensavo di non farcela. Ricorderò sempre le sue email alle 2 di notte che mi facevano dormire sonni più tranquilli.

Ringrazio gli amici che mi sono stati vicini, in particolare i PdN che hanno condiviso con me questo tortuoso percorso di studi, fatto di sacrifici, successi e tanta birra (tranne il Gingio che è astemio). Un pensiero speciale va anche a Marco e Fabio, compagni di epiche avventure in Spagna.

Un ringraziamento enorme va a Maria, che mi ha sopportato nei momenti di maggiore stress ed ha sofferto con me durante la scrittura della tesi.



# Contents

<b>Sommario</b>	<b>I</b>
<b>Abstract</b>	<b>III</b>
<b>Ringraziamenti</b>	<b>V</b>
<b>1 Introduzione</b>	<b>1</b>
<b>2 Background</b>	<b>5</b>
<b>3 State of the Art</b>	<b>15</b>
3.1 Automatic Sunspot Detection . . . . .	15
<b>4 Methodologies</b>	<b>23</b>
4.1 Basics . . . . .	23
4.2 Semantic Segmentation . . . . .	26
4.3 Clustering . . . . .	30
4.4 Representation Learning . . . . .	32
<b>5 Problem Statement</b>	<b>37</b>
<b>6 Dataset</b>	<b>41</b>
<b>7 The Model</b>	<b>47</b>
7.1 Training . . . . .	47
7.1.1 Full-Disk Image Segmentation . . . . .	47
7.1.2 Sunspot Representations and Classification . . . . .	52
7.2 Automatic Annotation Procedure . . . . .	56
7.3 Parameter Estimation . . . . .	58
<b>8 Results and Future Work</b>	<b>63</b>
8.1 Results . . . . .	63

8.2 Future Work . . . . .	66
<b>Bibliography</b>	<b>69</b>
<b>A Glossary</b>	<b>77</b>
<b>B McIntosh Classification Example Images</b>	<b>81</b>
<b>C Helios Processing Pipeline</b>	<b>83</b>
C.1 Preliminary Adjustments . . . . .	84
C.2 Feature Enhancement . . . . .	86
C.2.1 White-light . . . . .	86
C.2.2 H-alpha . . . . .	87
<b>D Master Control Program</b>	<b>89</b>

# Chapter 1

## Introduzione

This work lays at the intersection of observational solar physics and cutting-edge computer science. The activity of the Sun can be studied by observing the phenomena that take place on its surface, like sunspots. The purpose of this thesis is to detect and count sunspots using modern computer vision techniques, while also intending to completely remove human intervention in the process.

Traditionally, scientists identified solar features manually, sometimes with the aid of simple image processing algorithms. Recently, deep learning demonstrated that computers can achieve human performance in many challenging tasks. In spite of this, the scientific community seems reluctant in the adoption of new technology that eliminates the need for expert supervision. Nevertheless, during the last years, some encouraging signs of change in this trend were sent by the most important organizations of the field of science.

The European Organization for Nuclear Research (CERN) started applying deep neural networks to high energy physics simulations [1]. In astrophysics, the National Aeronautics and Space Administration (NASA) published an article in which machine learning is used for exoplanet detection [2]. In solar physics, the production of research studies using deep learning was stimulated by the publication of ready-to-use datasets [3] of the activity of the Sun. These attempts are very promising, but a real revolution in the field seems decades away. In this context, the work presented in this thesis wants to be a driving force for the modernization of the techniques that are used for scientific research.

The idea of creating a program that infers the activity of the Sun by counting sunspots on the disk came to my mind during the time spent at the

European Space Agency as an intern. For one year I worked at the Helios solar observatory, located in Madrid, Spain, where I learned all the basics of solar observation. My tasks ranged from practical telescope operation to the processing of the acquired images. In the process of developing the necessary tools for the automation of the whole daily observation routine, I also started to be interested in the physics that lays underneath the visible features of the Sun. The more I researched, the more I realized that there were great opportunities for machine learning to be applied to solar physics. Also, the abundance of data that modern instrumentations provide really encouraged me to experiment various solutions to the challenges that the study of the Sun proposes. The most successful of these experiments led to the realization of this thesis. The other part of the work that I did, but does not strictly relate to sunspot detection is left for the reader in appendix.

To be more precise about the meaning of “sunspot counting”, it is first necessary to understand the concept of sunspot. The Sun, the closest star to the Earth, is a sphere composed by hot ionized plasma that moves by convection, generating a strong magnetic field. One of the properties of magnetic field is that it tends to agglomerate into tubes that run across the volume of the sun and sometimes intersect with the surface. When this happens the outer layers of the star get perturbed and the convective mechanism that propagates the heat outwards gets locally interrupted, reducing the temperature of the interested area. The regions of reduced surface temperature caused by high concentrations of magnetic fields are called sunspots. Sunspots can be seen as manifestations of the internal dynamics of the Sun, therefore we expect high correlation with other similar phenomena.

During the last century, a series of studies proved several connections between the presence of sunspots and other indicators that can be detected from the Earth. Sunspots turned out to be strongly correlated with the repercussions of solar activity on the environment of the Earth. These interesting properties, together with the fact that they are observable with relatively cheap instrumentation, made the study of sunspots one of the most important aspects of solar physics.

In the early XVII century, solar observation pioneers started looking up at the Sun and counting the number of sunspots they could see. During more than 400 years, humanity kept recording and comparing the data using the relative sunspot number formula, ideated by R. Wolf in 1848, a simple expression that uses both the number of single spots and the concept of sunspot groups to obtain a unique time series. Nowadays, sunspot counting is still

done by humans, apart from some limited automatic solutions that work on specific datasets. The scientific community seems not to be intentioned to change the counting methodology in favor of automatic programs because that would generate a discontinuity in the annotation criteria. This thesis aims to show that modern computer vision models can learn the counting criteria from human experts, to be able to later produce an estimate of the number of sunspots that aligns with human results.

The algorithm proposed in this thesis uses only white-light images of the Sun for its computations and it is composed by two main components: the semantic segmentation component, that detects the areas of the image where sunspots are present, and the sunspot clustering component that uses representation learning techniques in combination with clustering in order to identify the number of groups. Both components use deep neural networks for their computations. The sunspot number time series produced by our algorithm is then compared with the international sunspot number (ISN) produced by SIDC-SILSO. The results show that, the algorithm is able to capture the trend of the solar activity pretty nicely, making it a good tool for its estimation.

A more detailed view of all the aspects of solar physics that are relevant for the understanding of this thesis is given in Chapter 2, together with a review of the history of solar observation.

Chapter 3 contains a description of the most important automatic programs for the annotation of images of the Sun. Advantages and disadvantages of each one are highlighted and the aspects of the solution presented in this thesis that represent an improvement are outlined.

Chapter 4 is divided in four sections. The first one gives the reader an introduction to the basics of machine learning, with particular focus on neural networks and how they learn. The following three sections review the most popular algorithms that perform respectively semantic segmentation, clustering and representation learning. The purpose of these sections is also to give the reader a gentle introduction on deep learning, making the rest of the thesis clearer.

Chapter 5 shows the formal setting of the problem we want to tackle and also provides visual intuition of the type of annotations that need to be performed from the algorithm.

Chapter 6 gives a detailed description of the data. The chapter describes how four datasets have been merged together to create a single, more comprehensive one to run the experiments. The challenges of the whole data preparation process are highlighted, and the splitting of data in train, validation and test set are touched on.

Chapter 7 is the core of the work. It contains an exhaustive explanation of the process of training both components of the algorithm. The procedure for predicting the sunspot number on unseen data is explained in the second section of this chapter, while the last section takes care the estimation of those parameters that are necessary for the algorithm to work and be compared with others.

Chapter 8 explains the reasons underlying the tests that have been carried out, the quantitative final results and how they can be refined in the future.

The first two appendices (A, B) serve as an aid to the reader respectively to navigate through the terminology of solar physics more rapidly and get a more visual intuition of the various forms the sunspots can take.

The last two appendices (C, D) explain some interesting parts of the work done at the European Space Agency, that are not completely on focus with the thesis but are still a valid tool to better understand the field of solar observation. In particular Appendix C explains the main processing techniques that have been used to enhance some features of the Sun, while Appendix D talks about how the automation of the Helios observatory was achieved during my internship.



## Chapter 2

# Background

The Sun is the closest star to the Earth and it sits at the heart of the Solar System. It is by far the largest object of our surroundings, in fact our planet can fit more than a million times in its volume [4], it holds 99.86% of the total mass of the Solar System [5], and its magnetic field reaches well past Pluto and Neptune [6]. The activity of the Sun has significant environmental influences on the Earth and therefore modeling its behaviour is fundamental. In order to do that it is necessary to understand its structure first, since a great deal of the phenomena that take place in the outer parts of a star are actually caused by some internal mechanism.

The Standard Solar Model (SSM) [7] is a mathematical formalization of the functioning of the Sun. It can be used to predict the internal observables (physical quantities that can be measured) through the resolution of the classical stellar equations and the knowledge of fundamental physics like nuclear reaction rates, screening, photon interaction and plasma physics [8]. In recent times, thanks to GOLF, MDI, and VIRGO instruments aboard the SOHO [9] spacecraft (ESA/NASA), it was possible, not only to shed light upon the internal mechanics, but also to validate the inferred structure of our star by using our knowledge of helioseismology (Seismic Solar Model - SeSM [10]).

The modern view of the interior of the Sun can therefore be summarized as (from innermost to outermost) [11]:

- **Core:** the innermost 20-25% of the radius, temperature and pressure are sufficient for nuclear fusion to occur;
- **Radiative zone:** between about 20-25% of the radius, and 70% of the radius, energy transfer occurs by means of radiation, no convection

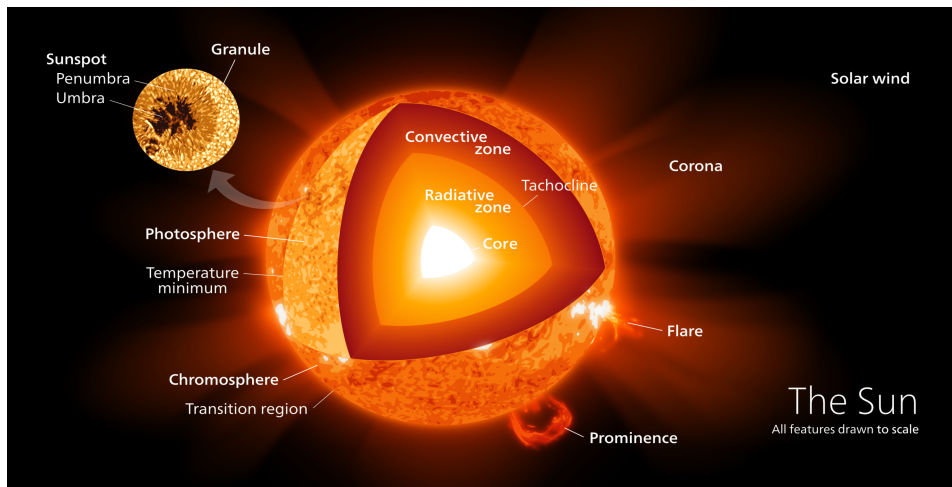


Figure 2.1: Visualization of the interior structure of the Sun. [12]

exists;

- **Convective zone:** Between about 70% of the radius and the visible surface, temperature is low and the particles diffuse enough for convection to occur;
- **Photosphere:** the deepest part of the Sun which we can directly observe with visible light. It can be regarded as essentially the solar *surface* that we see when we look at it, although the Sun, being a gaseous object, does not have a clearly-defined surface;
- **Atmosphere:** the surrounding gaseous *halo*, comprising: chromosphere, solar transition region, corona and heliosphere.

In this work we mainly focus on phenomena related to convection, hence occurring in the convective zone and impacting photosphere. Convection is the transfer of heat from one place to another by the movement of fluids. In particular, regarding the Sun, the temperature at the bottom of the convection zone is  $200,000^{\circ}$  K while at its outermost limit (surface of the Sun) it is being cooled by the creation of light and temperature is only about  $5,700^{\circ}$  K. This large difference triggers the plasma movement in order to propagate the heat outwards. Note, for instance, in Figure 2.2b the bright regions correspond to hot rising material, whereas the dark lanes are the location where the colder material falls down into the Sun [13]. Also, as the reader can verify from Figure 2.2b, the way convection cells organise on the surface is not regular but rather chaotic and turbulent.

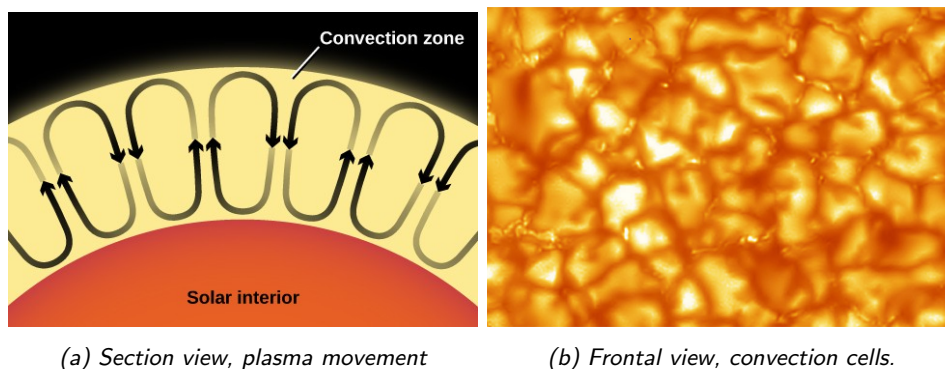


Figure 2.2: Convection

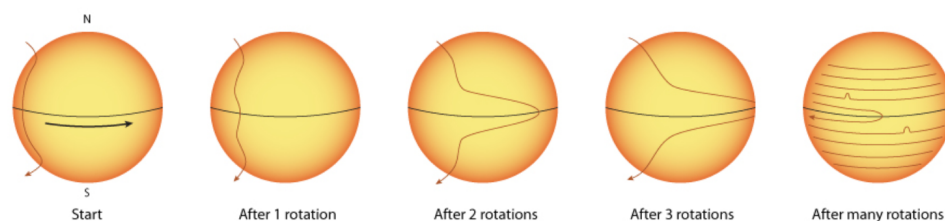


Figure 2.3: Visualization of differential rotation

Another interesting feature of the dynamics of the Sun is its rotation. In fact the Sun does not rotate uniformly, since it is not a rigid object (a solid body in which deformation is zero or very small). Our star is composed of gasses in the form of plasma and therefore the relative movement of its inner particles cannot be neglected. This results in a type of motion called differential rotation. It has been observed that the angular velocity of the particles changes in a way that depends on the latitude, in particular it is fastest on the solar equator and decreases as latitude increases [14].

From this notion, it follows that the rotation period is not constant, it takes 24.47 days at the equator and almost 38 days at the poles [15]. This behaviour has a critical importance for the understanding of this work for two reasons. First, the features that we study are located on the photosphere and move with the surface of the Sun, undergoing significant deformation. Second, differential rotation together with convective turbulent motions leads to the generation of electric currents and solar magnetic field. This phenomenon is called solar dynamo and is in some way similar to the dynamo effect that generates the magnetic field of the Earth. Moreover the generated magnetic field has the property that it tends to agglomerate into bundles called magnetic flux tubes.

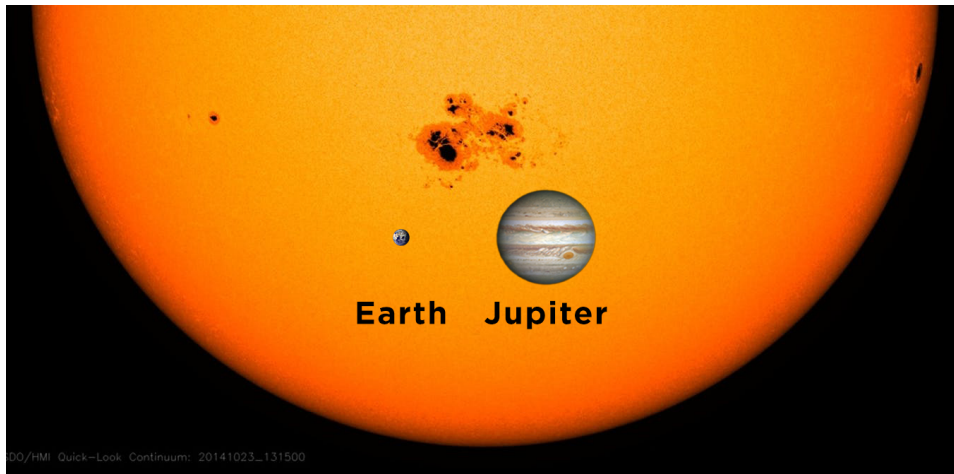


Figure 2.4: Relative size of AR12192, the Earth and Jupiter

When these tubes become strong enough to locally inhibit convection the heat coming from inside the Sun is not propagated upwards and the temperature of the surface decreases significantly. The local temperature drop makes the affected area look darker than the rest of the disk. These black patches, commonly named **sunspots**, can become very large and thus fairly easy to observe, even with amateur instrumentation. Their average size is comparable to the one of the Earth, but in some cases, when the magnetic perturbation is very strong, they can reach approximately the size of Jupiter or even more, as in the case of AR12192 (Figure 2.4), the largest group of the last solar cycle.

As shown in Figure 2.4 the intensity gap is not uniform, the darkest areas (*umbrae*) are located where the magnetic field is perpendicular to the surface, while on the periphery of the magnetic tube the field is slightly inclined and results in a lighter color (*penumbrae*).

It has been also shown that, since they indicate intense magnetic activity, sunspots accompany secondary phenomena such as coronal loops, prominences, flares and coronal mass ejections. For this reasons they have been widely observed and studied during the last 400 years.

Since the invention of the telescope [16] (early XVII Century) many astronomers started to notice these dark features, although they were not quite sure about their causes. Some thought they were shadows of undiscovered planets crossing the Sun, while others believed them to be dark clouds in the Sun's atmosphere. In 1843 an amateur German astronomer

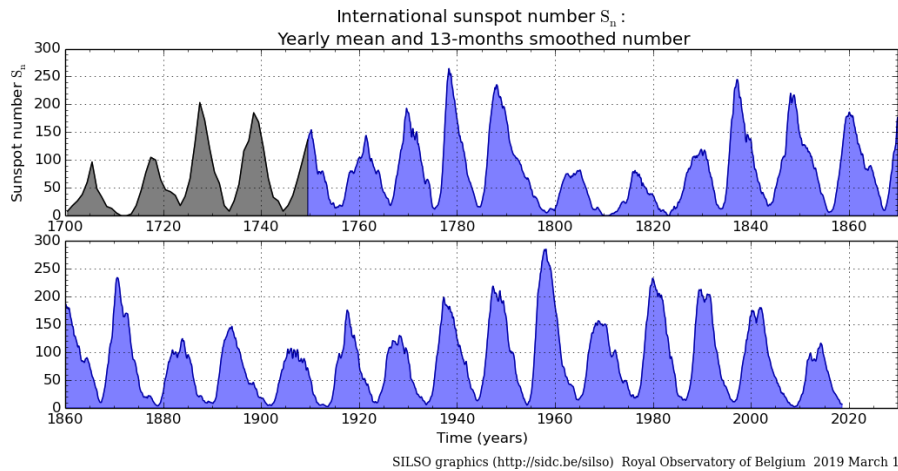


Figure 2.5: Yearly mean sunspot number (black) up to 1749 and monthly 13-month smoothed sunspot number (blue) from 1749 up to the present.[18]

named Samuel Schwabe discovered the rise and fall of yearly sunspot counts we now call the solar cycle [17]. Recent studies show that, more in general, the solar cycle is the nearly periodic 11-year change in the Sun’s activity that encompasses a multitude of phenomena, as, for example, the variable levels of emitted radiation and the ejection of material.

As already discussed, the change in magnitude of the activity of our star is also visible on the Earth; for instance large geomagnetic storms leading to auroras are most common during the peak of the cycle. Soon after Schwabe’s discovery, in 1848, Rudolf Wolf established a relative sunspot number formulation to compare the work of different astronomers using varying equipment and methodologies, known as the Wolf (or Zürich) sunspot number [19]. Such definition is still in use today and we show it later in this chapter.

Wolf succeeded in reliably reconstructing the variations in sunspot number as far as the 1755–1766 cycle, which is known conventionally as *Cycle 1*, with all subsequent cycles numbered consecutively thereafter; at the time of writing (early 2019), we are in the final phase of *Cycle 24*, although the first sunspot of *Cycle 25* may have appeared in early April 2018 [20][21] or even December 2016 [22].

Using Figure 2.5 and Figure 2.6 the reader can verify that the trend is indeed periodic. The graphics exhibit peaks and valleys; a peak in the sunspot count is referred to as a time of *solar maximum*, whereas a valley, a period when just few sunspots appear is called a *solar minimum*. Focusing

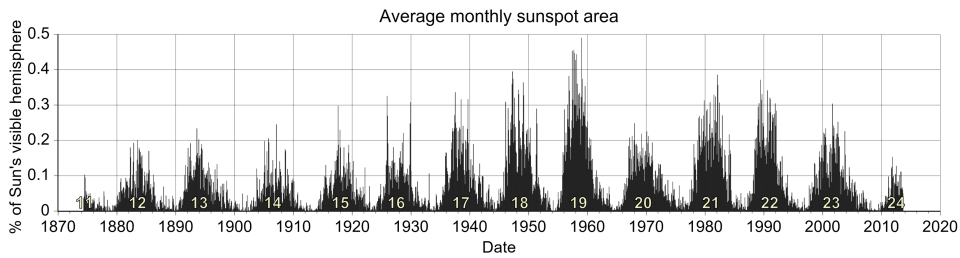


Figure 2.6: Diagram showing average monthly sunspot area from cycle 12 to cycle 24

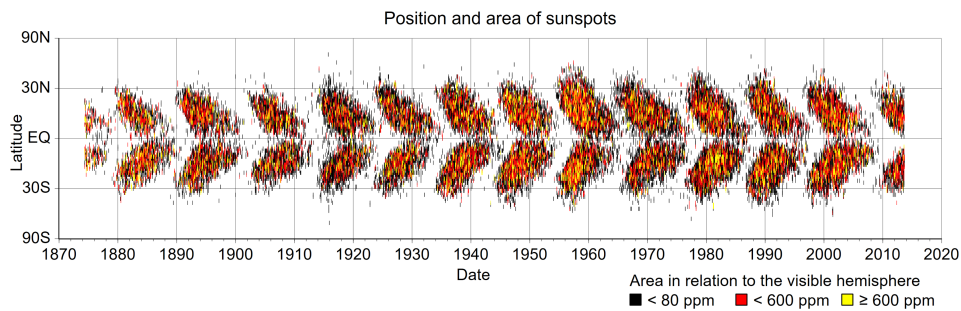


Figure 2.7: Butterfly diagram showing paired position and area of sunspots

on Figure 2.7 instead, we can discover one more property of the sunspot cycle: *Spörer's law* [23]. Spörer's law predicts the variation of sunspot position during a cycle by introducing the concept of *cycle latitude phase* (CLP), that is calculated from the average latitudes of each sunspot over a period of time. It turns out that sunspots tend to appear around  $30^\circ$  to  $45^\circ$  latitude on the Sun's surface. As the cycle progresses, sunspots appear at lower and lower latitudes, until they average  $15^\circ$  at solar maximum. The average latitude of sunspots then continues to decrease, down to about less than  $10^\circ$  and then while the old sunspot cycle fades, sunspots of the new cycle start appearing at high latitudes. The reason why this happens is still not completely understood by the scientists.

Another notion we need to introduce in order for the reader to fully understand this work is sunspot group classification. Indeed, several different categorization paradigms have appeared during the years, but the one that is most popular nowadays is the *McIntosh classification*, introduced in 1990 [24]. The McIntosh classification is constituted by three components, in which the nomenclature used for each sunspot group type is **Zpc**, where *Z* corresponds to the *Modified Zürich Classification* and the other two components, *p* and *c*, reflect the main sunspot characteristics: the type, size, and symmetry of the penumbra and umbra; and the degree of compactness of the

group [25]. We will focus on the **Z** component, that partitions the groups in the following categories[26] (refer to Appendix B for a visual intuition):

- **A**: a small single unipolar sunspot. Representing either the formative or final stage of evolution;
- **B**: a bipolar sunspot group with no penumbra on any of the spots;
- **C**: a bipolar sunspot group. One sunspot must have penumbra;
- **D**: a bipolar sunspot group with penumbra on both ends of the group. Longitudinal extent does not exceed  $10^\circ$
- **E**: a bipolar sunspot group with penumbra on both ends. Longitudinal extent exceeds  $10^\circ$  but not  $15^\circ$
- **F**: an elongated bipolar sunspot group with penumbra on both ends. Longitudinal extent of penumbra exceeds  $15^\circ$
- **H**: a unipolar sunspot group with penumbra.

Example images for each and every one of the classes can be found in Appendix B.

As already described, sunspots and therefore groups are highly correlated with perturbations of the magnetic field of the Sun. In fact, besides being physically close to each other, sunspots belonging to the same groups are usually manifestations of the same *active region*. These are areas of intense magnetic activity that can sometimes cause solar flares and coronal mass ejections. They are observable in several different bands of the spectrum emitted by the Sun, though normally detected using the high-energy ultraviolet band or line-of-sight (LOS) magnetograms. In general, there is a lot of research going on in multi-wavelength solar analysis, that is the field that investigates the Sun interpolating knowledge among different wavelengths. These studies are also pushed forward by the modern instrumentation we possess.

Nowadays spotting active regions is pretty easy and can be achieved with several different tools (refer to Chapter 2 for more information). Once every active region is recognised it is fairly easy to determine which sunspot belongs to which group. Still, the notion of sunspot group was born a long time before active regions were observed for the first time. To clarify this, it is sufficient to think that the relationship between sunspots and magnetic field is a relatively recent finding and magnetographs only appeared

at the beginning of 20th Century. However, this does not mean that all the studies that were carried out before the invention of sophisticated space telescopes should be dismissed as “outdated”. The complexity of modern instrumentation makes their lifespan limited and the dataset they produce quite unique. On the contrary, ground-based observatories have been pointing at the Sun during more than 400 years, making it one of the longest scientific experiment in the human history.

Longevity makes it possible to capture secular variability and therefore enable long-term analysis that would not be possible otherwise. Sure enough, ground observation has its own limitations as well. Despite being very stable in the long term it cannot be considered reliable in short periods of time. In fact, considering a single observatory as source of data, it is impossible to estimate the daily number of sunspots consistently. This happens because the quality of the data is strongly affected by the dynamics of the atmosphere. If the thick layer of air that sits inbetween the telescope and the Sun is turbulent or hazy the resulting images will loose definition, intruding a bias on the sunspot count. On the other hand, if the atmosphere is quiet and transparent the light will travel without trouble, making the observations almost as good as the ones of space telescopes. It is understood that if the solar disk is obscured by the clouds during the whole day the counting is impossible and it should be regarded as a missing data point.

Nowadays, to overcome these limitations the SIDC/SILSO [27], the most important authority in the field, uses an ensemble of hundreds of observatories to produce the sunspot number series [28]. Averaging over several heterogeneous detections has many benefits: mitigating the variance, improving accuracy and solving missing data problems. However, it also introduces a new layer of complexity in the computation of the final index, since it is not possible to perform a simple average considering that the available stations are different from day to day.

Every observatory set-up has its own, unique properties that should be taken into account in the calculation. This uniqueness is captured in the personal reduction coefficient ( $K$ ) included in the *relative sunspot number* formula by R. Wolf:

$$R = K \cdot (10 \cdot g + s) \tag{2.1}$$

The way this solar activity index is calculated has not changed much since its establishment, 400 years ago. What has changed, though, is our ability to resolve very small spots on the surface of the Sun and highlight them through post-processing.



An example of the processing techniques that can help scientists in the counting is *limb darkening correction*. This method consists in the reduction of the phenomenon of limb darkening through software. This phenomenon is an optical effect intrinsic to the physics of the observation, that causes an imbalance in the quantity of photons we receive from the limb of the disk compared to its center (refer to Figure 2.4 for a visual intuition). In practice, this can be pleasant to the eye because it gives the viewer the idea of the sphericity of the Sun, but it is also inconvenient when trying to detect small features on the edge.

The theoretical laws that govern the magnitude of the darkening for each point of the surface are very rich and interesting but they are out of the scope of this thesis. In any case, there are simpler solutions that are able to correct the images only leveraging on the intensity profile of the image (more details are given in Chapter 3). Other interesting post-processing techniques for feature enhancement will be illustrated in Appendix C, including for instance gradient sharpening, off-limb emission enhancement and mean intensity correction.

To conclude this brief introduction to solar physics, it is necessary to present some basic aspects of *solar coordinate systems*. Standardizing the way the position of solar features is encoded is vital for the creation of large scale datasets, but at the same time it is very difficult for several reasons. The first reason is that the axis of rotation of our star is tilted with respect to the ecliptic of the Earth. Therefore the solar north pole, as seen from the our planet, appears displaced. Clearly, the displacement is not constant, but rather periodic. As the Earth proceeds on its orbital path, the observed axial tilt changes in a range that goes from  $-7^\circ$  to  $7^\circ$ . The second reason that should be considered in the study of solar coordinates is that the Sun is a gaseous body, there are no fixed points of reference and this is made worse by differential rotation. Finally, it is also not perfectly spherical, although its oblateness is rather small and almost neglectable [29].

The last reason is represented by the fact that the revolution and rotation of the Earth itself influences our perspective of the Sun. For this reason, a coordinate should not really be considered complete if it does not include time. The development of sophisticated coordinate systems for solar image data allows to overcome, at least partially, these difficulties. Nowadays, solar coordinate systems can be divided into 3 categories [30]:

1. **Heliographic Coordinates:** latitude and longitude of a feature are

expressed on the solar surface, and can be extended to three dimensions by adding the radial distance from the center of the Sun. There are two basic variations on the heliographic system: **Stonyhurst** and **Carrington** heliographic. Both use the same solar rotational axis (based on the original work by R. Carrington in 1863), and differ only by an offset in the longitude definition.

2. **Heliocentric Coordinates:** any system of coordinates where the origin of the axes is located at the centre of the Sun. The system can be **Cartesian**, when the z axis is defined to be parallel to the observer-Sun line and the y axis points towards the north pole, or **radial** if it uses a position angle measured from the projection of the north pole and a radial distance.
3. **Helioprojective Coordinates:** observations are projected against the celestial sphere and all projective angles have origin at disk center, considered as the apparent disk center as seen by the observer without any corrections made for light travel time or aberrations.

## Chapter 3

# State of the Art

In this chapter the main challenges of automatic sunspot detection are analysed and several state of the art algorithms that overcome these challenges are presented, together with an assessment of their advantages and disadvantages. The reader will also understand why being able to automatically segment images of the Sun and perform sunspot clustering is so critically important.

### 3.1 Automatic Sunspot Detection

Manually drawing the contours of dark patches with white background can look trivial to the inexperienced eye, but centuries of disagreements among scientist on that matter demonstrate that this is actually not the case. In fact, irregularities in the shape of the sunspots and their variable intensity and contrast with the surroundings make their automated detection from digital images difficult [31]. Similarly, automatically clustering sunspots into groups, taking into account the properties of magnetic field, has revealed itself to be a rather complex task.

In the past, given the moderate quantity of data available to the scientists, it was quite easy to label all the images. During the years technological development progressively enhanced the quality of datasets at our disposal. From December 1995, when the SOHO mission was launched, the Sun has been under almost constant human surveillance from space. Since then, the space telescope has delivered a daily stream of 250MB of solar data back to researchers on Earth. With 12 instruments on board, probing every area of our star, SOHO has compiled a vast library of solar data. This solar work of art is beginning to be complemented by SDO, which is returning 1.5TB of

data per day, comprising constant streams of high-resolution images of the Sun over 10 spectral channels [32]. The volume of data produced by SDO has made it necessary to develop new ways of analysing it.

In the early stages of the development of this thesis traditional image processing approaches have been explored in order to tackle sunspot detection. Unfortunately methods like edge detection [33], or simple segmentation algorithms like watershed transform [34] or histogram clustering [35] are simply not powerful enough to yield acceptable performance on this task. On one hand both edge detection and watershed fail badly, tending to get confused by the noise introduced by convection cells, and ending up overdetecting. Hyperparameter tuning can possibly help in many cases but it is impossible to find the right values in order for the methods to generalize on unseen images. Even selecting ad-hoc parameters for every example the detection performance remains very poor. Such a strong evidence testifies that the above-mentioned approaches are not suitable solutions to the problem this work aims to tackle.

On the other hand, even though histogram clustering is a very naïve approach and yields poor performance on the general segmentation task, later on in this thesis it will be shown that very simple clustering algorithms are able to distinguish umbrae from penumbrae given the mask of the selected sunspot and its classification.

During the 1990s and the early 2000s, simple methods were regularly employed, most of them with human supervision. At that time, the existing techniques for sunspot detection could be divided into the three basic classes [36]:

- **Thresholding** methods: relying on disk intensity variation;
- **Border** methods [37]: using the gradient of the intensity of the image;
- **Bayesian Pattern Recognition** methods [38]: applying simple statistical analysis to data labeled with expert identification or unsupervised clustering.

In addition to being far from autonomous, all these methods are data specific, in the sense that they were developed for specific datasets and, hence, make a number of assumptions about the properties of the data, image resolution, average intensity and presence of image artefacts [36].

Around 2010, probably driven by the launch of SDO (February 11, 2010), automatic solar feature analysis saw a new wave of research, with novel, far more complex methods being published. The ones that had greater impact in the field are the following [39]: the **Automated Solar Activity Prediction (ASAP)** and the **Sunspot Tracking And Recognition Algorithm (STARA)** [40] to detect sunspots in white-light continuum images; the **Spatial Possibilistic Clustering Algorithm (SPoCA)** [41] that automatically segments solar EUV images into active regions, coronal holes and quiet Sun; and the **Solar Monitor Active Region Tracker (SMART)** [42] that detects active regions in line-of-sight magnetograms. The ones that are most relevant with respect to the work that is presented in the next chapters are definitely ASAP and STARA, since they both use data in the same spectrum band (white-light) as our algorithm. However, a direct comparison would not be fair, because they also use other wavelengths in addition to the one in common.

ASAP is actually a set of algorithms for sunspot, faculae, active-region detections [43], and solar-flare prediction [44]. Many versions of this tool exist, the one that best relates to this thesis was published in 2011 [45]. This version uses both continuum intensitygrams and magnetograms (SOHO/MDI) to produce segmented groups of sunspots, while previous versions also required quick look (in GIF or JPEG format) images as input. The main steps in this algorithm can be summarized as follows:

1. Pre-processing to detect the solar disk and remove limb darkening;
2. Coordinate conversion: detected solar disks are converted from heliocentric coordinates to Carrington heliographic coordinates;
3. intensity filtering threshold value  $T = \mu - (\sigma \times \alpha)$  is applied where  $\mu$  is the mean,  $\sigma$  is the standard deviation of the image, and  $\alpha$  is a constant.

In practice ASAP works pretty well [39] although the coordinate system change shows advantages and disadvantages. The main advantage of ASAP is that towards the limb of the Sun, on a two dimensional heliocentric image each degree is represented by fewer pixels while in a heliographic image each degree is represented with the same amount of pixels. Ideally, this trick corrects the distortions due to perspective on the original data and relieves the subsequent pipeline of models from the burden of dealing with it.

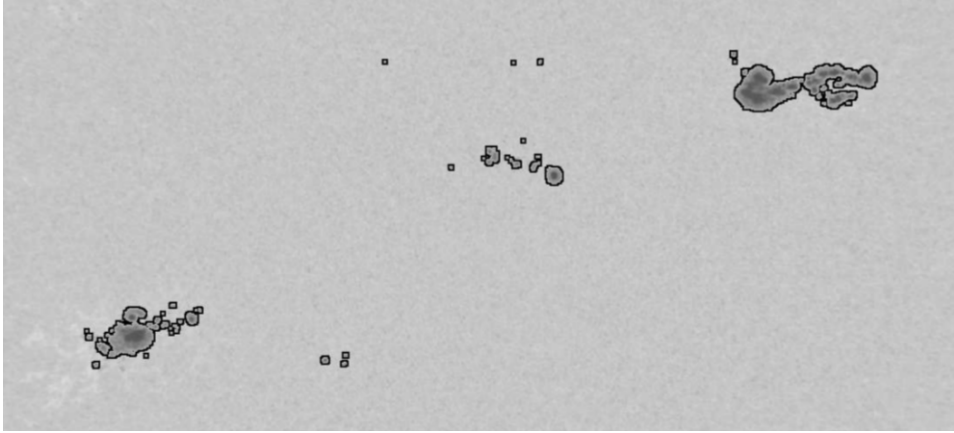
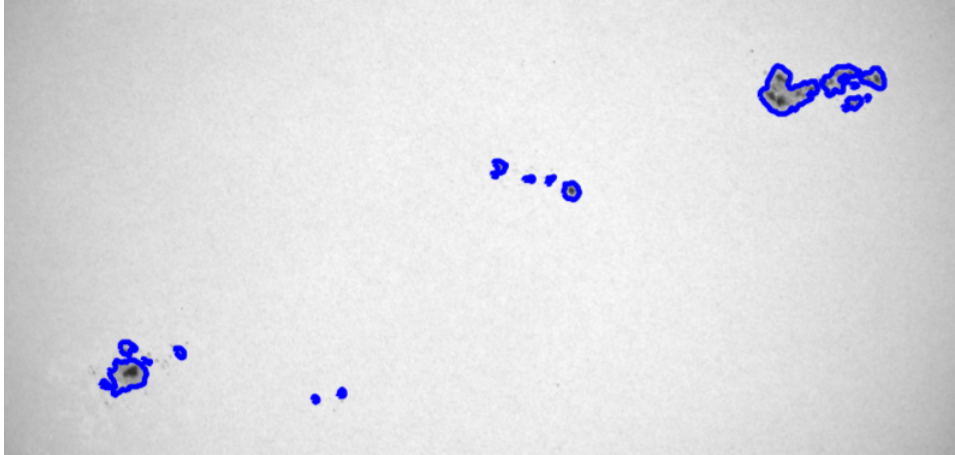


Figure 3.1: An example of ASAP detections from 10 June 2003.

Although coordinate conversion sounds like a very sensible approach, it actually introduces a bias in the produced image, because it needs to interpolate neighbouring pixels when the mapping is not one to one. The main problem with this approach is represented by the fact that the transformation is computationally expensive. For instance, the application of ASAP to SDO/HMI [46] images larger than 1024 by 1024 pixels shows that heliographic conversion algorithms must, in some way, be made more efficient to tackle larger images [39]. Also, like every other method that employs thresholding it is hardly generalizable on heterogeneous data and inherently less tolerant to noise. In any case, ASAP has been developed to work on images belonging to a well-established dataset produced by SOHO space telescope. This ensures an homogenous data distribution and a really low noise level (compared to telescopes observing from underneath the atmosphere of the Earth).

The STARA algorithm was originally developed to analyze the data generated by SOHO, but it has been improved to be able to take advantage of the new data recorded by SDO and, unlike ASAP, several ground-based instruments [47]. STARA makes use of morphological image processing techniques [48] to be able to efficiently process large datasets. Specifically, the images are inverted to leverage the top-hat transform that detects the intensity peaks in an image. This can be thought as moving a circle (structuring element) underneath the profile and marking the path of its center. As the structuring element is chosen to be larger than the width of the sunspot peaks, it cannot fit into those peaks and so their depth is reduced. This profile is then morphologically dilated (dual transform to erosion).

In formulas:  $T(f) = f - ((f \ominus s) \oplus s)$  where  $f$  is the original data,  $\ominus$  is



*Figure 3.2: An example of STARA detections from 10 June 2003.*

the erosion operation,  $\oplus$  is the dilation operation, and  $s$  is the structuring element. Doing this gives a very similar profile to that of the original image, but without the sunspot peaks present. Subtracting this profile from the original yields the final sunspot candidates. In practice, STARA demonstrated to work so consistently on space telescope data that NASA released a data catalogue with it and to ensure the detections were as accurate as possible a labeled dataset containing human sunspot detections was used as a ground truth in order for the optimal structuring element to be chosen.

Nonetheless STARA cannot yet be considered as fully autonomous when applied on ground-based telescope data. For instance, tests conducted in the Kodaikanal (India) observatory showed that human intervention is still needed in some cases [47]. Also, STARA is not able to perform sunspot clustering. Once it detects where the interesting features are located in the image it needs to be complemented by other algorithms like SPoCA or SMART to be able to partition the sunspots into groups.

Similar algorithms using mathematical morphology for image segmentation have been applied to data coming from ground observatories. An example is the pipeline that was set-up at the Ebro Observatory [49] located in Spain. It uses the top-hat transform to detect sunspots and then applies a threshold on the heliographic distance among them for grouping [50]. According to the statistics produced by the Ebro Observatory, the sunspots neighboring scale is 6 heliographic degrees, this means that sunspots which are separated less than  $6^\circ$  are part of the same solar group. This is a very simple method that works in some cases, but in the maximum of the solar cycle, when groups are

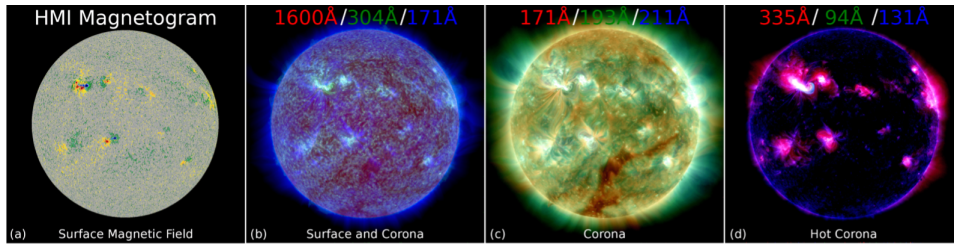


Figure 3.3: Data visualized with FlareNet: (a) The Helioseismic and Magnetic Imager (HMI) [46], colorized full-disk images of surface magnetic field; (b, c, d) The Atmospheric Imaging Assembly (AIA) [52], 8 channels ranging from UV to EUV composed into RGB images.

placed in a very close space, the operators who supervise the whole process at Ebro Observatory must manually redefine the groups [50].

More recent attempts to tackle the problem of sunspot detection and clustering are not well documented and it seems the research field has reached a plateau during the last years. Most ground-based observatories are using combinations of the simple methods described above and human supervision [51]. Nonetheless, until now, we are still very far away from having an autonomous agent that is able to accurately estimate the number of sunspots without external intervention. However, in this thesis we show that this is actually possible using modern Computer Vision techniques, like deep learning.

It is not the first time that advanced statistics is being utilized to study or predict the behaviour of the Sun. Around 2005, simple Machine Learning approaches have been explored for sunspot classification [53][54][55]. They mainly used decision trees on hand-crafted features in order to be able to predict the McIntosh class of a group of sunspots. More recently, with the raise of deep learning, deep neural networks started to be employed for space weather prediction, in particular to perform forecasting on solar flares, solar energetic particles, and coronal mass ejections. Thus, many studies have been published on this multi-disciplinary research field that NASA's 2017 Frontier Development Laboratory decided to develop FlareNet [3], a software framework for experimentation within these problems. FlareNet includes components for the downloading and management of SDO data, visualization, and rapid prototyping. The system architecture is built to enable collaboration between heliophysicists and machine learning researchers on the topics of image regression, image classification, and image segmentation.



In this context where artificial intelligence is becoming increasingly important in solar physics, our work seeks to create an agent that is capable of completely replacing the human expert in the process of sunspot counting.



## Chapter 4

# Methodologies

### 4.1 Basics

Machine learning is the discipline that studies how to provide computers with the ability to learn and improve their performance over time automatically. It is usually divided into three main subfields: supervised learning, unsupervised learning and reinforcement learning. However, since reinforcement learning is out of the scope of this thesis, we will focus on the former two.

The goal of supervised learning is to find the function that best approximates the desired outputs given some input data. What is interesting about this function approximation task is that it can be done without explicitly instructing the program (also called model) on how to do it, but instead letting it learn useful patterns from experience. This process is referred to as supervised because it uses the data, in the form of inputs and outputs, as a teacher for the model. In the learning of this mapping, we also expect the algorithm to generalize from the training to unseen situations in a reasonable way, not simply memorizing each and every data point it is fed (overfitting). Specifically, translating this idea to the real task of sunspot counting, it is desirable to build an agent that, after being taught by the experts on how to label images of the Sun, will be able to perform it automatically.

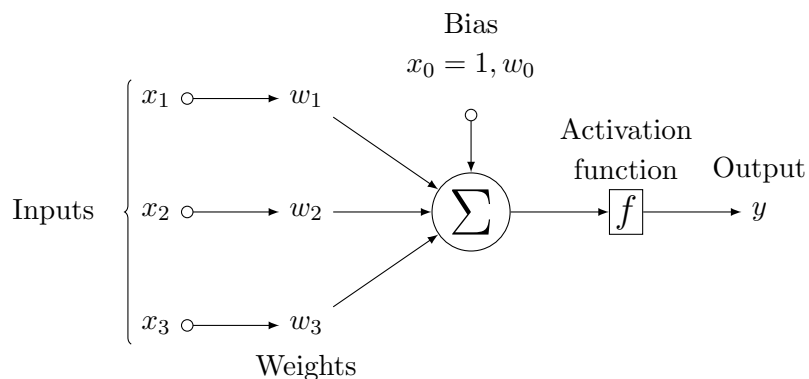
The second step of our algorithm tackles the problem of partitioning sunspot into groups. That's where unsupervised learning comes into play. The set-up of the problem is similar to the previous case, you have some data and want to teach a model to find interesting patterns in it, in order to perform some task. The crucial difference is that the data is not labeled. In other words, compared to the supervised setting, this means that the output data is

missing. Therefore, instead of learning a mapping, the goal for unsupervised learning is to model the underlying structure in the data to extract more insights about it. Examples of unsupervised learning techniques that will be treated in this thesis are clustering and representation learning.

Orthogonally to these macro-categories, machine learning encompasses several models, each one with its strengths and weaknesses. This work focuses on the type of model that is most popular at the time of writing: the Artificial Neural Network, a computing paradigm that was inspired by biological neural networks that constitute human brains. They consist of artificial neurons organized in layers that communicate with each other by sending signals. Each neuron is nothing more than a mathematical function with  $m$  inputs that computes the following:

$$y = f \left( \sum_{j=0}^m w_j x_j \right) \quad (4.1)$$

where  $f$  is an activation function and  $w$  is the vector containing the weight for every input  $x$ . Also,  $x_0$  is a special input, called *bias*, whose value is fixed to 1. This allows the activation function to be shifted to the left or to the right, to better fit the data. Changes to the weights alter the steepness of the activation function, while the bias offsets it. The blocks composing the artificial neuron can be schematized as below:



As the reader can verify, the final decision about the magnitude of the output signal is taken by the activation function. It is responsible to determine whether or not the input features are important enough to further contribute to the computation. The activation is a non-linear function, generally continuous and differentiable (except for very primitive formulations, like the

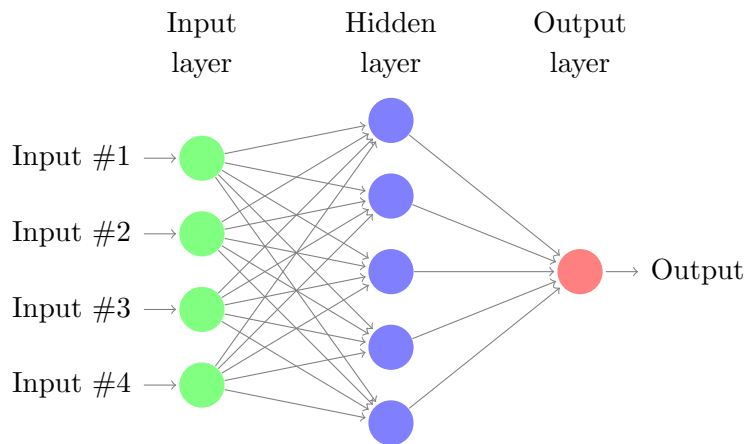


Figure 4.1: Multi-layer fully connected network architecture

perceptron [56]). These two properties are fundamental to build a multi-layered architecture. In fact, imagine to stack more than one layer of neurons as in Figure 4.1 where the output of the neurons are attached as inputs to the subsequent layer, then ideally, the larger the number of layers, the more “learning power” the network attains.

If the activation function was simply linear, regardless of how many layers are stacked, there would exist a single neuron that could approximate the same function, because the composition of linear functions is always just linear. Non-linearity solves this problem and enables non-linear approximations.

Artificial networks learn through trial and error, in a way that is arguably similar to the process of human learning. First, the inputs get propagated forward through the neurons until the output layer is reached and then the error is calculated with respect to the desired output value using a loss function. Intuitively, the loss represents the cost of the error, or a measure of the penalty the network will experience. Therefore, learning can be seen as an optimization problem that seeks to minimize the loss as objective function.

The last building block needed for the network to improve its performance over time is the weight update through backpropagation (short for “backward propagation of errors”). In fact, the weights of the neurons have a relationship with the error the network produces, thus, if the parameters make an adjustment in the right direction, the error decreases. The direction of the update is usually calculated by means of an optimization algorithm

called gradient descent, that is based on the observation that a function decreases fastest if one goes in the direction of the negative gradient. Hence, given a learning rate  $\gamma$  small enough, the loss  $\mathcal{L}$ , a weight vector  $\theta$  and the following update:

$$\theta_j \leftarrow \theta_j - \gamma \frac{\partial \mathcal{L}}{\partial \theta_j} \quad (4.2)$$

then the loss is guaranteed to decrease.

Neural networks are a very flexible tool that can be used for virtually every problem, yielding good performance. Neurons can be connected to each other in many ways, forming several architectures, in order to adapt them to different data types (images, sentences, etc.). Also, they can be used with good results in all the subfields of machine learning. The following two sections will explain how to build neural networks that are capable of processing images in both supervised and unsupervised settings.

## 4.2 Semantic Segmentation

Image segmentation is the partitioning of an image into several groups of pixels, also called segments, based on some criteria. The segments of an image are usually stored in a mask where each group is assigned a unique grayscale value or color to identify it. Image processing solutions to this problem, employing a wide range of criteria, have been deeply explored. Sometimes, it is posed as a graph partitioning problem [57], other times as a clustering task [58]. However, in general, these criteria do not make any attempts at understanding what the segments represent.

With the rise of Computer Vision the situation changed completely. Instead of just applying some transformations to the image, Computer Vision aims to extract knowledge from the scene. Pixels are no longer seen as mere elements in a matrix, but rather as a means of conveying meaning. Performing image segmentation based on the understanding of the content is called semantic segmentation.

Back in the 1990s, segmentation was carried out using probabilistic frameworks, like conditional random fields (CRFs) among the others. Their advantage is that they can model the relationship between pixels in order to be more accurate in the prediction of the label. Nowadays, CRFs are not really used anymore, except as post-processing step of neural network models. In fact, around 2006, a group of researchers brought together by the Canadian

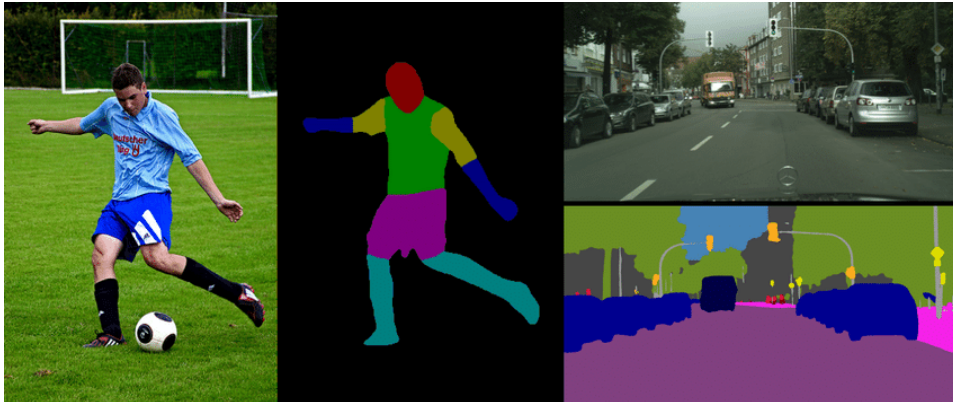


Figure 4.2: Examples of Semantic segmentation.

Institute for Advanced Research (CIFAR) showed that it is possible to train very deep artificial neural networks effectively [59] and that they outperform any other algorithm on computer vision tasks [60]. It was the dawn of the deep learning era. The following years saw a rapid increase in the quantity of scientific articles using deep neural networks, tackling all sorts of problems that, until then, seemed unsolvable.

In the context of semantic segmentation this revolution led to the invention of fully convolutional networks (FCNs - Long et al.[61]), a particular flavour of neural architecture that is able to produce pixel-level classification. The idea of using convolution-like operations to feed pixels into neurons first emerged in 1980 [62] and ways of training them through backpropagation were proposed around ten years later [63][64]. Convolutional Neural Networks (CNNs) take advantage of the fact that the input consists of images and they constrain the architecture in a more sensible way. In particular, unlike a regular neural network, the layers (or filters) of a CNN have neurons arranged in 3 dimensions: width, height and depth (Figure 4.3). The neurons in a layer are only connected to a small region of the layer that precedes them, instead of collecting the activations of all neurons in a fully connected manner [65]. Using this trick has 2 advantages, first, it decreases the number of parameters to be learned by a large margin, second, it forces the model to learn hierarchically, concentrating on smaller regions of the image to capture local information, but then merging them together to deduce more high-level concepts.

In practice, a CNN is able to encode the content of an image, automatically extracting features that are very effective at discriminating between classes,

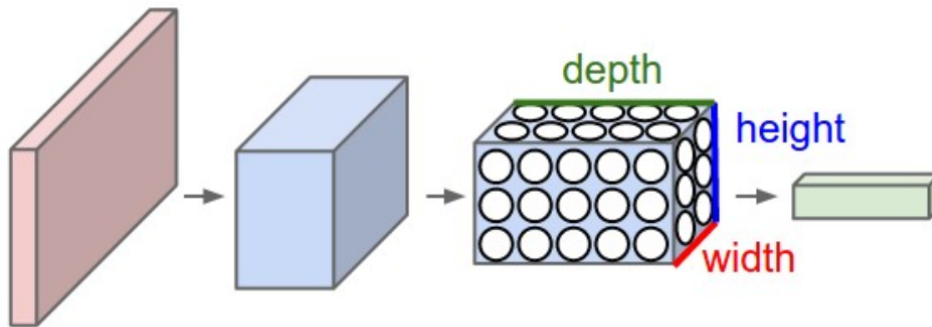


Figure 4.3: A visualization of a Convolutional Neural Network. In red the input image, in blue the convolutional layers, in green the output layer [65].

while at the same time preserving spacial information. This characteristic is fundamental because semantic segmentation faces an inherent tension between semantics and location: global information resolves “what” while local information resolves “where” [61].

A fully convolutional network (FCN) leverages the strenghts of this convolution-like operation to create an encoder-decoder architecture that is able to predict the location and the class of each object in the input. The encoder is a stack of convolutional, max pooling and batch normalization layers, where max pooling is a down-sampling operation that keeps the largest value, and batch normalization [66] handles the adjusting and scaling of the activations. The decoder is built similarly, but it up-samples the coarse feature maps into a full-resolution segmentation map using the transpose operation. The transpose convolutional filter does the opposite of a normal one, instead of performing a weighted sum of the inputs, it takes a single value, multiplies it by the weights of the filter and spreads the output values in the neighboring region of the next layer.

Long et al. also added “skip connections” that take activations from the encoder and sum them up to the up-sampled features of the decoder. The information extracted from earlier layers in the network (prior to a down-sampling operation) should provide the necessary detail in order to reconstruct accurate shapes for segmentation boundaries.

So, at training time, the network takes an image as input and propagates the values forward through the encoder and the decoder, until the output layer is reached. Subsequently, the error is calculated and the whole network is updated end-to-end using backpropagation. The loss function that leads



the training should be carefully chosen, depending on the architecture and the specific problem that is being dealt with. However, in general, the good old cross entropy loss over each pixel can be used for segmentation:

$$\mathcal{L}(p, y) = -y \log(p) - (1 - y) \log(1 - p) \quad (4.3)$$

where  $p$  is the predicted probability that a pixel belongs or not to the class and  $y$  is the ground truth. This measure is a solid choice with regard to image segmentation, because it returns very stable gradients to the network and helps it converge smoothly.

In some cases, cross entropy does not correlate well with human perception, especially when the distribution of the classes in the dataset is highly imbalanced. For this reason, other loss functions were introduced in the deep learning field. One of them is the Dice loss [67], based on the Dice coefficient ( $DC$ ) [68][69], in formulas:

$$DC(X, Y) = \frac{2|X \cap Y|}{|X \cup Y|} \quad (4.4)$$

where  $X$  and  $Y$  are respectively the set of pixels belonging to the predicted mask and the ground truth mask. Doing so, this coefficient quantifies the similarity (or overlap) between the real distribution of the classes and the one that was estimated by the network.

Among all the architectural proposals that followed the first FCN article [61], the one that most impacted the field was the U-Net [70] (Figure 4.4). Ronneberger et al. improve upon the vanilla version of the FCN primarily through improving the capacity of the decoder module of the network. More concretely, the strength of the U-Net consists in its symmetry, in other words the contracting path and the expanding path have roughly the same depth. In addition, the skip connections are revisited, in fact, in this architecture the features of the encoder are not summed to the ones of the decoder but rather concatenated. Thanks to these adjustments the U-Net produces much better results, being capable of distinguishing the details of objects from the background noise.

Over time, a wide range of more advanced atomic blocks got discovered and substituted to the original CNN blocks. This led to the publication of new versions of the U-Net. For example, in 2016, residual blocks [71] were added to the architecture [72] allowing faster convergence and deeper models to be trained. However, the simple but powerful ideas that laid the foundations for the U-Net still make it a very solid choice.

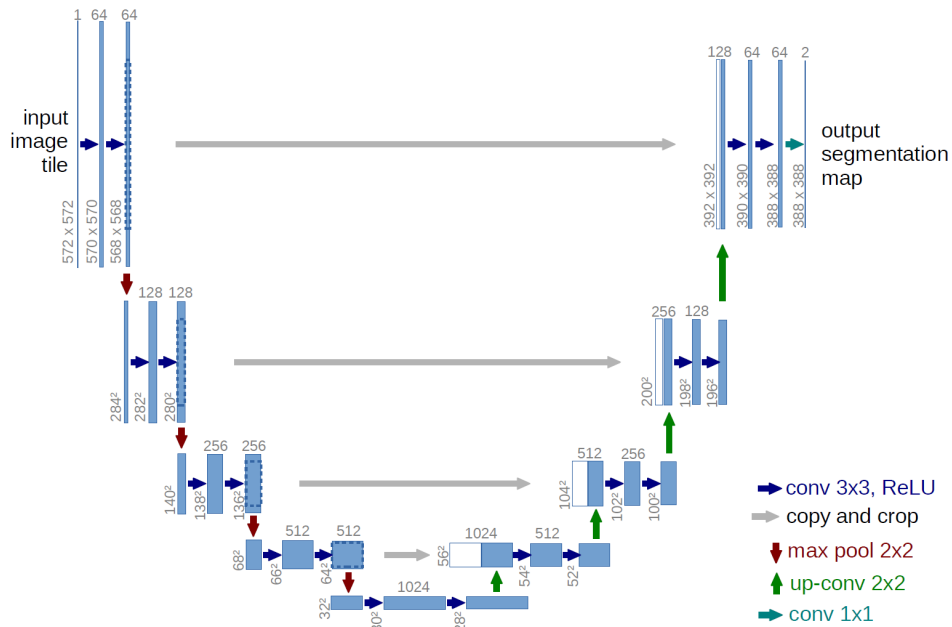


Figure 4.4: U-Net architecture [70].

### 4.3 Clustering

Clustering is the process of identifying natural groupings or clusters within multidimensional data, based on some similarity measures [73]. Clusters differ from classes because they are not known a priori but they are rather extracted from the data. Clustering can be achieved by various algorithms that differ significantly in the way they model the data. Despite this, all of its various realizations belong to the unsupervised learning family.

The purpose of this section is not to compile a review of each and every clustering algorithm. For that, the reader should refer to the extensive literature on the topic that can be found in bibliography [73][74]. This section is more of a brief overview on the methods that will be used in our main algorithm and the more subtle tweaks that make clustering work well.

Broadly speaking, clustering can be divided into two subgroups: those that require a priori knowledge of the number of clusters and those that are able to infer it. This distinction is particularly important in the context of sunspot detection because humans are very good at deducing the number of sunspot groups intuitively. An example of an algorithm that needs to know the number of clusters before running is k-means. Become very popular because of its simplicity and efficiency, k-means belongs to the type of algorithms

that model clusters as centroids. In fact, it initializes  $k$  randomly selected centroids which are used as the beginning points for every cluster, and then performs iterative calculations to finding the least-squares assignment to centroids. The algorithm terminates when the centroids have stabilized or alternatively when the maximum number of iteration has been reached. k-means, as most clustering algorithms, is not guaranteed to return the globally best solution. To mitigate this issue, the procedure can be restarted randomly for a fixed number of times. Although k-means is not able to determine the number of clusters automatically some methods have been developed in order to fix this, like the silhouette method [75] or the elbow method.

In the panorama of clustering algorithms that do not require a prior estimate of the number of clusters, DBSCAN [76] is one of the most famous. DBSCAN is based on a very different logic with respect to k-means. In fact it defines the clusters as dense connected regions in the data space. It takes 2 parameters as input:

- **eps** ( $\varepsilon$ ): the minimum distance between two points for them to be considered neighbors;
- **minPts**: the minimum number of points to form a dense region (core).

What happens at runtime (refer to Figure 4.5) is that a point is drawn from the dataset, DBSCAN forms an n-dimensional shape with radius  $\varepsilon$  around that data point, and then counts how many data points (neighbors) fall within that shape. If the number of neighbors is at least *minPts* then the selected point belongs to the core, otherwise it can be labeled as a border point or an outlier. Neighbors get expanded as well and the process continues until the density-connected cluster is completely found. Then, a new unvisited point is drawn and the same routine is repeated.

The resulting clustering has the property that all the clusters have at least density  $\varepsilon$ . Therefore, DBSCAN works well under the assumption that the density of the clusters is homogeneous. One of the advantages of using density is that it can be calculated using virtually any metric, while, by contrast, algorithms like k-means assume the points lay in an Euclidean space. In this thesis we feed such algorithms with data representations learned to optimize their distribution in a Euclidean space.

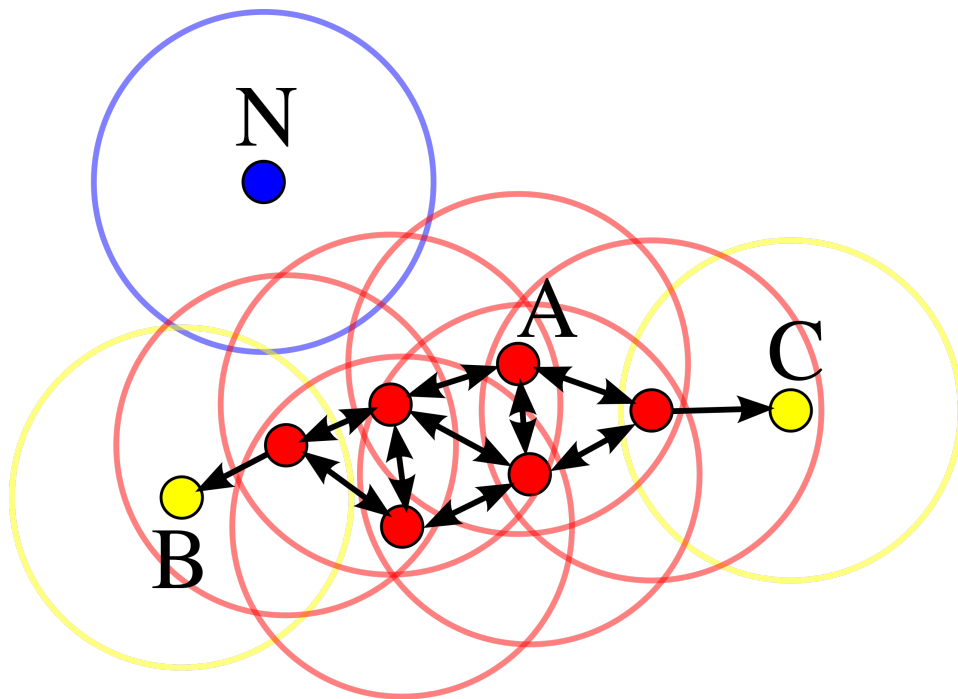


Figure 4.5: A visualization of DBSCAN running. Core points are highlighted in red, border points in yellow, outliers (noise) in blue

## 4.4 Representation Learning

Every day, when we read, hear a noise, see an object or, in other words, experience the world through senses our brains are flooded by information. One of the astonishing aspects of our intelligence is that we do not drown in this massive flow of data, but rather we are able to navigate through it. Much of the ability to draw the best from our perceptions is due to the way we represent information. In the human brain, sensory information is represented by neural activity. Recent studies proved that using the distribution of activation in the neural population generated by the sight of some object it is possible to reconstruct an image of the object itself [77]. This result suggests that the brain is constantly extracting salient features from perceptions.

Representation learning (also known as feature learning) is a fundamental step in almost any machine learning pipeline. Learning a representation means to find a model that maps a generic data point in a high-dimensional space to a set of latent variables in a low-dimensional space. Latent variables, as opposed to observable variables, are variables that are not directly

observed but rather inferred. This feature transformation technique can be very useful because depending on how data is represented, different information can be uncovered. Nonetheless, it is rarely used as a stand-alone procedure. To make the most out of it, representation learning should be coupled with other supervised or unsupervised techniques. In fact a good representation is one that makes a subsequent learning task easier [78, Chapter 15].

Many different approaches to representation learning are possible and some of them will be described here. The most straightforward way of discovering good descriptors for the data is through unsupervised learning. A statistical approach to representation learning is offered by the Principal Component Analysis (PCA) algorithm [79].

PCA is a dimensionality reduction algorithm that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of linearly uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. Representations can be extracted from the principal components and used as inputs to other models. PCA is very popular and effective in some situations, but suffers many limitations due to the assumptions that it makes. One of the most criticized aspects is that it can only find linear correlations, but the relationships among data are not always so simple. This assumption can be relaxed using models that are capable of non-linearity. Autoencoders are an example of this. They are particular types of neural networks that try to reconstruct their input. As in the case of image segmentation models, the architecture is composed by an encoder and a decoder, however, the interesting part here is not the output, but the bottleneck. The encoder is forced to produce a representation of the input that carries as much information as possible, in order for the decoder to be able to reconstruct the input. Features that are sampled at the bottleneck of the network can have several different properties, depending on the activation function and the loss that are used. Autoencoders can be used in combination with convolutional blocks as well, in order to extract features from images.

Clustering can also be considered an unsupervised feature learning algorithm. In fact, imagine to run, for instance, k-means on a dataset. The procedure will return  $k$  centroids, where  $k$  is the desired number of clusters. Features can be produced in several ways from the centroids. The simplest

is to create a  $k$ -dimensional one-hot vector for each data point, where the  $j$ -th element of that vector is 1 if the point belongs to the  $j$ -th cluster. It is also possible to use the distances to the clusters as features.

The situation changes when we have more information about the problem we want to solve. In fact we can train supervised learning models that look for the exact characteristics of the data that are most important to separate some predefined class. Imagine to have a dataset with 10 classes and to know the class associated to each record. Ideally, it is possible to train a model that is capable of taking two records as input and calculating their similarity. The model will be trained using the information about the classes as a supervision signal, for example the similarity can be defined to be 1 when both inputs belong to the same class and 0 otherwise. Once the model has converged, it can be used to predict the structure in unseen data.

There is plenty of clustering algorithms that are capable of taking a similarity matrix and find the clusters that optimize some measure. Hopefully then, each cluster will be trivially associated to the right class. This looks like a very far fetched approach in this case, because there are only 10 classes. With a low number of classes it is much easier to build a classifier that discerns them. But suppose to be working on Facebook's dataset of faces, where millions of people are catalogued. In this latter case, the classifier approach becomes unfeasible because it would have to decide among millions of classes, making complexity explode. On the other hand, for the similarity model this is not a problem, it just has more data to be trained on. In fact, pairwise similarity is still well defined and the large matrix that holds it does not have to be necessarily precomputed for the clustering phase. In practice, this model creates a different data representation by exploiting the relationships between the records. The new representation is indeed the similarity matrix.

Several new studies elaborated on the concept that relative knowledge is, sometimes, a very powerful source to be drawn from. In the field of computer vision the solution to the problem of learning similarities in the data is called siamese network [80] (Figure 4.6). The siamese network is a class of neural architectures that contain two or more identical subnetworks. Identical here means every aspect is shared, they have the same configuration, parameters and weights. In the case of two subnetworks, at training time each one takes one of the two input images and encodes it in a vector representation, also called embedding. The two outputs are then fed to a contrastive loss

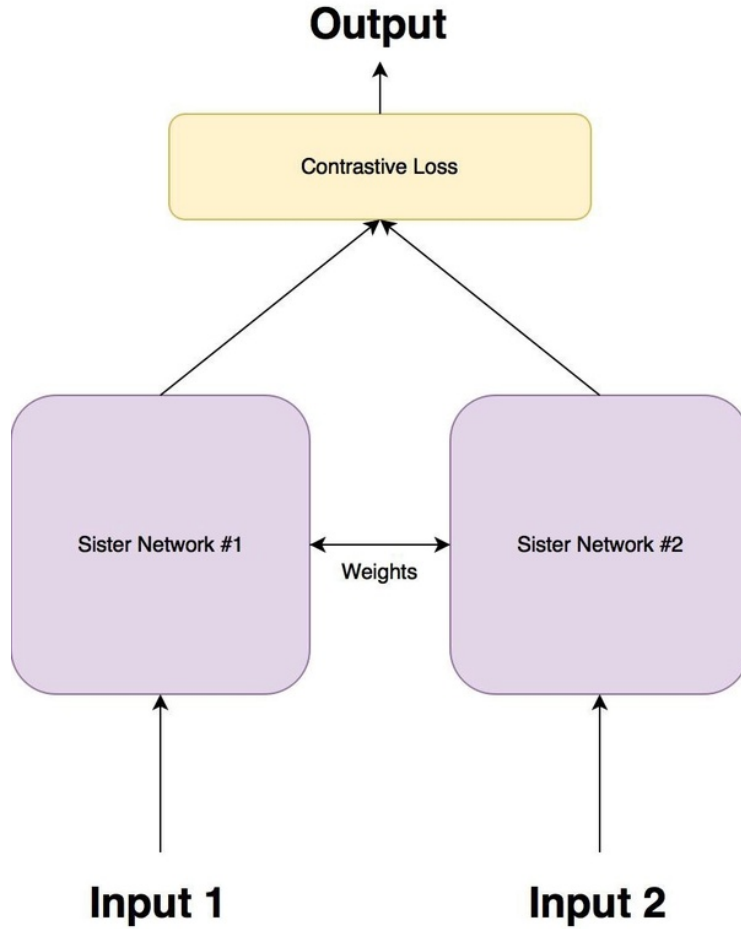


Figure 4.6: Siamese network architecture.

function [81] defined as follows:

$$\mathcal{CL}(D_{W_{12}}, y, m) = (y) \frac{1}{2} D_{W_{12}}^2 + (1 - y) \frac{1}{2} \max(0, m - D_{W_{12}})^2 \quad (4.5)$$

where  $D_{W_{12}}$  is the distance between embedding  $W_1$  and  $W_2$ ,  $y$  is the ground truth of the similarity and  $m$  is the margin hyperparameter. The loss is called contrastive because it forces the model to learn how to embed images in such a way that neighbors are pulled together while, by contrast, non-neighbors are pushed apart.

As the weights evolve during the training the updates are mirrored across both subnetworks. At testing time, all the images are embedded so that data becomes just a set of feature vectors. Any type of supervised or unsupervised learning algorithm can subsequently leverage the general repre-

representations generated by siamese networks. However, we will mainly focus on clustering methods that are able to group together all the realization of the same class in the test set. Representations can be learned in a way that optimizes clustering performance. It is sufficient to align the type of distance that is used by the contrastive loss and the clustering. So if, for instance, the distance  $D_W$  in the loss is Euclidean, then the clustering should be tuned accordingly to consider that the points lay in an Euclidean space. Moreover, if the number of classes in the test set is not known a priori knowing the expected distance between the clusters could be very useful. Here the parameter  $m$  comes into play. In fact,  $m$  represents the margin after which the network does not push two dissimilar points further apart. It is therefore reasonable to assume that the expected distance between clusters will be roughly  $m$ .

Nowadays, the state of the art results for siamese networks are achieved using a slightly different version of the contrastive loss: the triplet loss. While the contrastive loss aims at relatively minimizing and maximizing the distance between similar (positive) and dissimilar (negative) examples separately, the triplet formulation does both things at the same time. Formally:

$$\mathcal{TL}(D_{W_{ap}}, D_{W_{an}}, m) = \max(0, m + D_{W_{ap}} - D_{W_{an}}), \quad (4.6)$$

where  $D_{W_{ap}}$  is the distance between the anchor and the positive example,  $D_{W_{an}}$  between the anchor and the negative and  $m$  is, again, the margin. The advantage with respect to the normal contrastive formulation is that, updating on both positive and negative cases concurrently you can manage to be less greedy, or alternatively to take into account more context. That said, sampling heuristic in such pairwise distance learning setting may be as crucial as choosing the right loss, as some studies state [82].



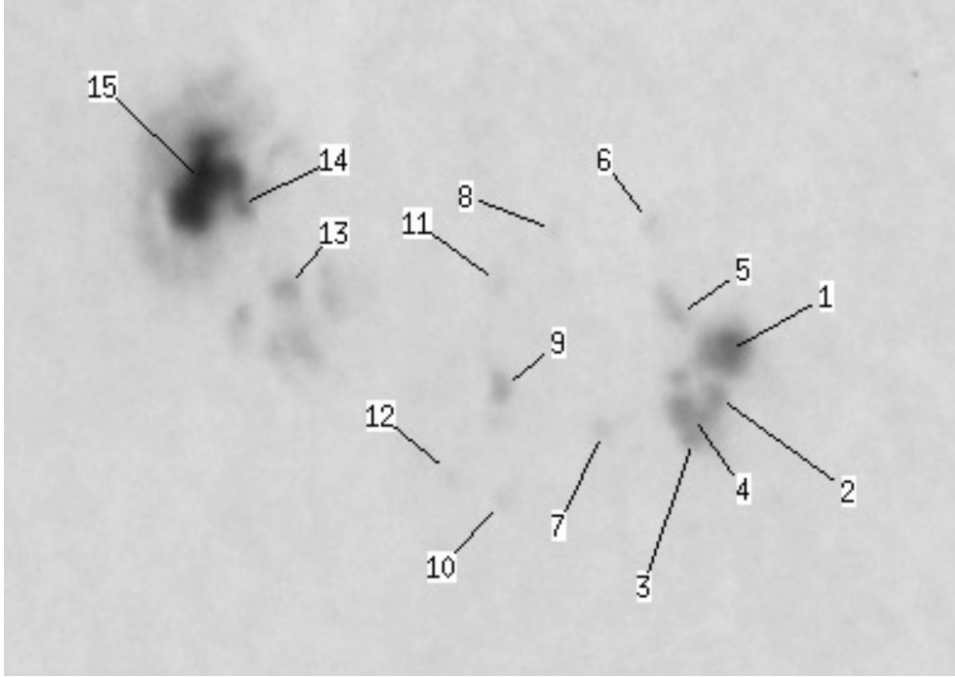
## Chapter 5

# Problem Statement

The amount of magnetic flux that rises up to the Sun's surface varies with the progress of the solar cycle. Visible sunspots on the disk are one of the manifestations of the perturbations of the magnetic field. The inner activity of the star and the flow of particles that get blasted out from the outer layers also depend on the intensity of these perturbations. It follows that, improving our estimate of the sunspot index would also make our space weather predictions more reliable.

One could think that determining the sunspot count univocally could be possible. Unfortunately, it turns out that, for several reasons, it is not possible. The personal reduction coefficient  $K$  in the relative sunspot number formula (2.1) tries to mitigate this exact problem. It captures both the variability due to the observation and the subjectivity of the observer. Ideally, if the person in charge of manually counting and grouping sunspots was the same in all the observatories of the world, then the variance generated by subjectivity would be neglectable. This is clearly not possible. But imagine having an algorithm that can learn from the experts and be distributed and run by every observatory. This would guarantee constant detection standards if the parameters of the program were identical in all the stations.

So far, the automatic methods that have been proposed show two main problems: first, they are not general, so they cannot be used on heterogeneous data without retuning their parameters; second, they are not reliable enough to work without human supervision. These drawbacks basically break the hypothesis that the algorithm is indeed consistent in the estimation of the number of sunspots. This happens because those methods are not based on the understanding of the scene, but rather they leverage very specific properties of the data. Advanced computer vision algorithms, instead, fo-



*Figure 5.1: Annotated group of sunspots.*

cus on the semantic of the image, making assumptions on the data much less important.

This thesis proposes a deep learning approach to sunspot counting and aims to be a step forward in the integration of computer vision into solar physics. Before diving deep into the practical details of this work, we need to define the problem in a more formal way.

The only theoretical physics device that is used is the relative sunspot number formula (2.1) that is shown here again for the sake of clarity:

$$R = K \cdot (10 \cdot g + s). \quad (5.1)$$

The three variables that we need to calculate are:

- the number of sunspots  $s$ ;
- the number of groups  $g$ ;
- the personal reduction coefficient  $K$  of the algorithm.

The identification of each one of these variables carries its own challenges. In Figure 5.1 the reader can appreciate a manual annotation of a group of

sunspots by an expert scientist. Each label refers to a single sunspot so the total number of sunspots in the image is equal to the number of labels that are present. Note that the annotation is not as straightforward as it seems, for example label 13 refers to more than one dark area while labels 2, 3, 4 all map to a single black spot. We do not go into detail about why labels have been assigned this way but the reader can assume that the expert annotator has a deep understanding of the behaviour of the magnetic field and therefore he is able to distinguish real sunspot instances from artifacts.

In the case of Figure 5.1 the sunspot group was found near the center of the disk, but there are cases where the sunspots appear much closer to the limb. In those cases the viewing angles are very high and the annotation is a lot harder, because of perspective deformations and the limb darkening effect.

Once all the sunspots in the image have been annotated they can be clustered into groups. Again, in Figure 5.2 the reader can get an impression of what it means to find sunspot groups. If the magnetogram of the solar disk is available, it is fairly easy to detect active regions first, and then map each sunspot to the closest one. For the reasons that were explained in Chapter 2 it is supposed that the magnetogram is not available. So, clustering has to be performed by inferring the magnetic link between each sunspot. Sunspot classification can be used as an aid for clustering. In fact, broadly determining the class while forming a group is important to make sure that the magnetic properties are respected. Nevertheless, when the cycle is near to the peak active regions tend to be very close to each other. This behaviour is showed in Figure 5.2, where groups 11516 (grey) and 11517 (red) are almost overlapping. In such a situation separating or not the two groups is a rather subjective matter that causes inconsistencies.

Finally, the personal reduction coefficient is usually estimated statistically. Since sunspot counting is done on a daily basis,  $K$  can be recalculated every day taking into account past observations. SILSO implements a similar procedure using a pilot station as a reference, and, for simplicity they recalculate once per month, instead of daily.

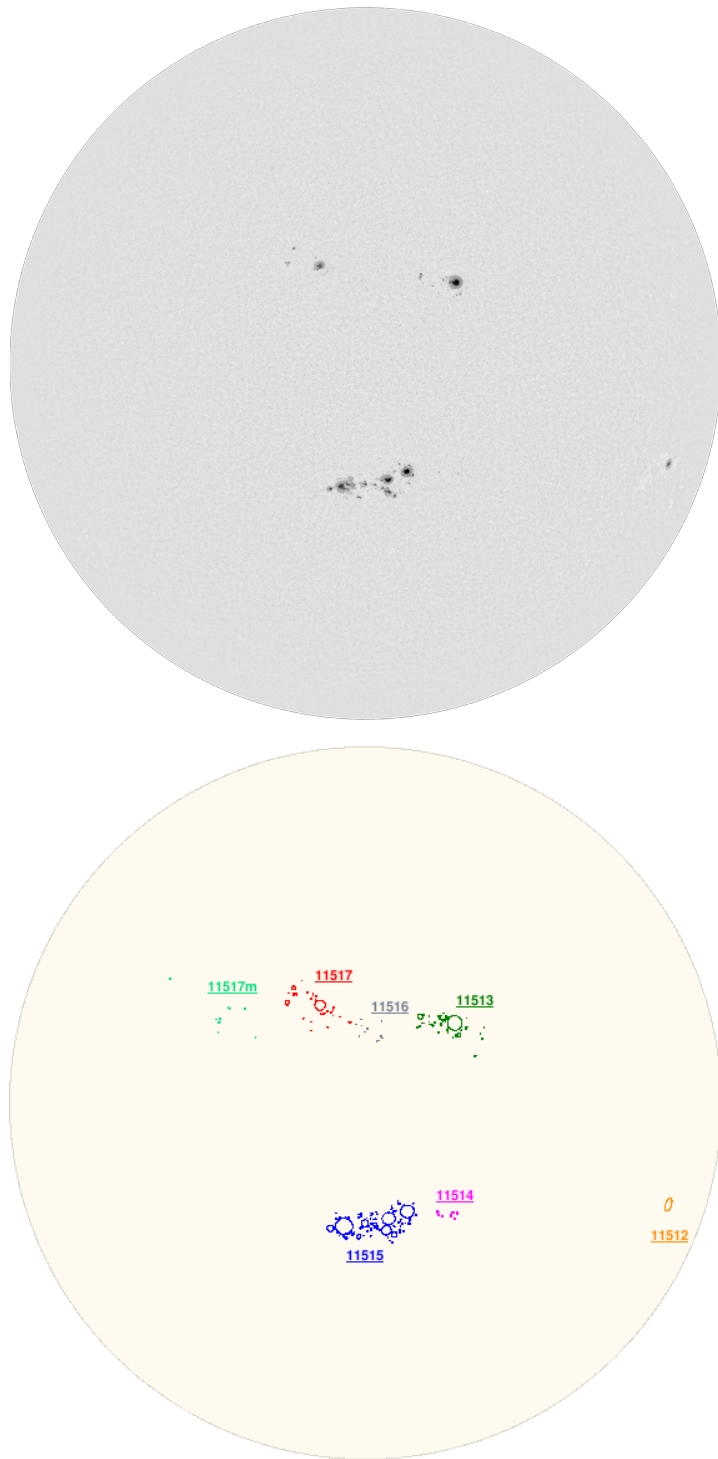


Figure 5.2: Complete solar observation, 03/07/2012. Top: full-disk white-light image, Bottom: annotated mask with colorized sunspot groups

# Chapter 6

## Dataset

The abundance of astronomical data provides great opportunities for machine learning. In fact, most of the data is free to use for anybody and labels are almost always available. Solar physics is no exception to that. Many public databases of solar events can be downloaded in textual form and then annotations can be reconstructed from them. However, the reconstruction is not trivial sometimes, since a good understanding of solar coordinate systems is required. In this work 4 datasets have been merged together:

- **Debrecen Photoheliographic Data (DPD) [83][84]:** a catalogue of positions and areas of sunspots from 1974 until present times, compiled by using white-light full-disk observations taken at the Helio-physical Observatory of the Hungarian Academy of Sciences (Debrecen, Hungary) and its Gyula Observing Station as well as at other observatories;
- **Sunspot Index and Long-term Solar Observations (SIDC-SILSO) Dataset [85]:** a time series of daily total sunspot number derived with the relative sunspot number formula (2.1), provided by the Royal Observatory of Belgium using a network of observers. This dataset is used for validation and testing purposes only;
- **Virtual Solar Observatory (VSO) [86]:** a tool for investigating the physics of the Sun, by searching and downloading existing databases for terrestrial and space-based observations;
- **US Air Force, Mount Wilson (USAF/MWL) Dataset:** a catalogue that provides a list of sunspot regions and their parameters, observed by the USAF solar observatories and the Mount Wilson ob-

servatory. In particular, it is famous for sunspot group classification data.

The last solar cycle was selected as a use case for the training and testing of the algorithm. Both ground and space telescope data were added up to form a quite large dataset. One full-disk observation was considered for each day, drawn and reconstructed from the DPD dataset. Space based data ranging from 2011 to 2014 was downloaded with a Python module called SunPy [87] that connects to VSO's open APIs, while ground based data from 2011 to 2013 was obtained directly from DPD's FTP access. Due to availability problems, the ground observations for 2014 were missing from the database.

Since the images came from different datasets some pre-processing was necessary to unify them. Ground based observations came in the form fits files with variable resolution (from 4096x4096 to 8192x8192), where limb darkening correction had already been applied but the disk was not centered in the the plate. So, first, the center, radius and axial tilt of the Sun were extracted from the header in order for the Sun to be aligned and cropped from the raw image. The coordinates of the sunspots were given in the DPD dataset in heliocentric-radial coordinates. From those, using simple trigonometry, it was easy to calculate the positions in pixel coordinates.

On the other hand, the preprocessing of space based observations needed a bit more work. For each day, sunpost instances were extracted from DPD, and SDO images were searched inside the VSO using date and time of the DPD observation. The temporally nearest image was selected and downloaded.

Despite the fact that SDO observations are very frequent, the average delay between the image and the annotated mask was around half an hour on average. This time shift is pretty large, considering that the Sun rotates and the Earth moves on the orbit (SDO can be considered solidal to the Earth in this case). The misalignment was big enough to make it necessary to rotate the positions of the sunspots to create very accurate masks. Luckily, DPD also provides Carrington heliographic latitude and longitude for each sunspot. These coordinates can easily be transformed into the helioprojective system. At this point, the images were loaded as a SunPy Map, a class that contains many useful methods to interact with solar data. One of these functions allows to calculate where an helioprojective coordinate maps to, after a time delta, taking into account the differential solar rotation profile.

This function works well when the sunspots are big because the magnetic tubes that create them usually have a lot of “inertia”, and therefore move pretty consistently with the rotation profile. Marginal misalignments still arise when sunspots are very small, because even a modest change in the magnetic perturbation causes erratic movements, making it difficult to predict their location. After these calculations, it was the turn of limb darkening correction. From the center of the disk, circles of increasing radius are created and the pixels that lay on the border of the circle are averaged, creating the limb darkening profile (blue points in Figure 6.1). A polynomial function is fitted on these data points and then, for every pixel, the right correction is applied to straighten the profile.

Once the coordinates of the sunspots are ready for both ground and space observations, the segmentation masks can be generated by a flooding procedure. For each sunspot a seed is initialized from its coordinates. From the seed, the procedure starts to explore the pixels around it in a breadth first fashion, so that for every iteration the pixels with lowest value are selected for expansion. By doing so, the algorithm floods the darkest areas of the image first, remaining trapped into the penumbra. Using the whole spot area provided by DPD, it is possible to calculate the exact number of pixels of each sunspot and therefore make the expansion stop exactly on the edge of the penumbra. All the pixels that have been selected at least once are then written on a black array that has the same shape of the original image, to create the segmentation mask.

The final mask also contains information about the groups and their classes in order to train the second part of the algorithm. The annotations created by this procedure are consistently good in practice, as it can be appreciated from Figure 6.2. The figure also shows how the precision of the mask increases with the size of sunspots. For instance, the contours of the small rightmost sunspots are not correctly captured in the mask, while the largest ones are almost perfectly represented.

Overall, data preparation was probably the hardest and most time consuming part of the work, but the results repay the effort. The final analysis on the dataset shows the following statistics:

- 2555 images with 4K resolution;
- around 21000 sunspot group instances;
- around 190000 single sunspots;

hmi\_20110101\_101910\_continuum.png - Intensity Profile

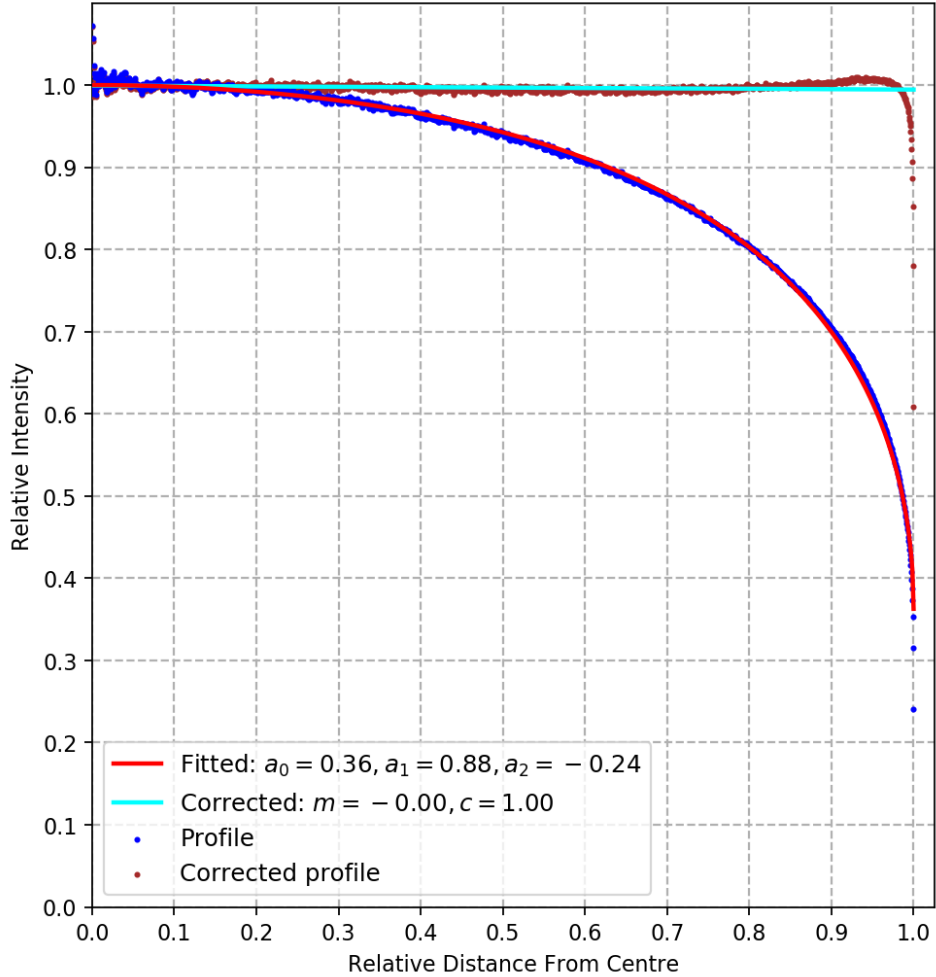
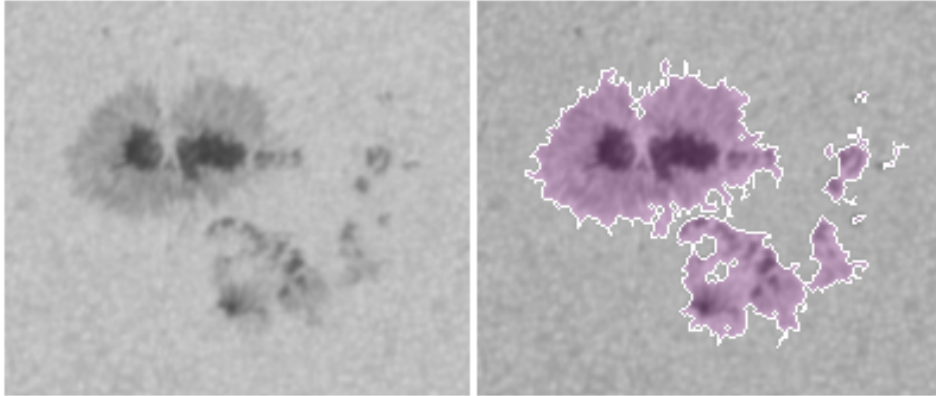


Figure 6.1: Limb darkening intensity profile created with SLDTk [88].

In order to train the algorithm and be able to assess its performance, it is necessary to divide the dataset into three subsets:

1. **training set:** the part of data the algorithm learns from;
2. **validation set:** the sample of data that is used to provide an unbiased evaluation of the model while tuning hyperparameters. In our case the validation set is particularly useful to estimate the personal reduction coefficient ( $k$ );
3. **test set:** the chunk of data that is used to estimate the final performance of the model once it is completely trained.





*Figure 6.2: Example of a sunspot group mask.*

The dataset was split accordingly trying to minimize the dependencies among the subsets. The initial idea was to split the dataset randomly, sampling observations for each month. This would have enabled us to estimate more precisely the average monthly sunspot number in the testing phase, because the observations would have been homogeneously distributed. Unfortunately, this is not a good approach because it introduces strong dependencies in the data, since sunspots can last even more than ten days on the disk and therefore the same sunspot could have appeared in both training and test data. A simple strategy to divide the dataset while reducing regularities is to sample observations consecutively for the validation and test sets. So for each month the data was partitioned as follows:

- days from the 1<sup>st</sup> to the 19<sup>th</sup> were assigned to the training set;
- days from the 20<sup>th</sup> to the 24<sup>th</sup> were assigned to the validation set;
- days from the 25<sup>th</sup> to the last one of the month were assigned to the test set;

This data split design does not completely solve the dependency problem. In fact, though harder, it is still possible that the same sunspot appears repeated among the three sets. However, the longer the sunspot survives on the disk the more deformation it undergoes, strongly mitigating the dependency issue. Also, this partitioning strategy is a good trade-off because it still enables us to make monthly activity charts, averaging the results over the test set.



# Chapter 7

## The Model

### 7.1 Training

#### 7.1.1 Full-Disk Image Segmentation

We argued that the problem of estimating the activity of the Sun ultimately reduces to the calculation of the relative sunspot number. A fundamental step for this calculation is finding the number of single sunspots in a full-disk image of the Sun. Semantic segmentation was selected as a tool to identify the areas of the disk that contain a sunspot, intended as all the pixels that fall inside the border of the penumbra, as opposed to the areas where the Sun is quiet.

In practice, we need a model that is able to take an image as input and return a binary segmentation mask, where all and only the pixels belonging to some sunspot have a predicted label equal to 1. Unfortunately, this is not sufficient to be able to count them. In fact, depending on the class of the group, two or more sunspots can be surrounded by the same penumbra. Therefore it becomes necessary to separate umbras from penumbras as well. This can be achieved with a mix of classification and clustering techniques applied on the results of the segmentation model. This further refinement of the counting routine was not performed at training time and therefore is addressed in Chapter 6.

Starting with the semantic segmentation step, the first component of the program that counts the sunspots is a deep neural network for image segmentation. Specifically, the architecture that was chosen in this case is the U-Net. Minor changes to its internal structure were made to adapt it to the particular problem, but the overall functioning is the same as explained

in Section 7.2. With respect to the original configuration, the network was made considerably more efficient by decreasing the number of convolutional filters in each layer, without though decreasing performance.

At training time input images of the disk are divided into an array of overlapping patches and for each one of them the number of non-zero pixels in the respective patch of the ground truth mask is calculated. The number of non-zero pixels is considered as a measure of the salience, or equivalently an indication of how much the network can learn from such patch. At the same time, the number of patches to be extracted ( $n_p$ ) is determined using a simple heuristic that takes into account the number of groups that are present in the image. Then,  $n_p$  patches are sampled from the array with probability proportional to the salience of the patch. This procedure enables the network to draw the best from each training episode, since the images are very large, while the sunspots are small and concentrated in the equatorial band of the disk.

Now that the training examples have been identified, it is the time of feeding them into the U-Net. The patches are organized in a batch and forwarded into the network. When the computation has finished, the output layer returns the predicted mask for each input example. These masks are compared to the real masks and the error is calculated using cross entropy loss. Subsequently, the error is backpropagated through the network so that the optimization of the parameters can be performed. The optimizer that was used on this network is called stochastic gradient descent (SGD) and it works by replacing the real gradient with a stochastic approximation. Instead of calculating the gradients for all of the training examples, it is sometimes more efficient to only use a random subset of these examples. By doing so, it increases the noise on the gradient, but at the same time it acts like a regularizer to the gradient estimation, removing the bias of the training set. Also, momentum was used in combination with SGD. With momentum the optimizer accumulates the gradient of the past steps to determine the direction to go, rather than using only the current step to guide the search. The combination of these two strategies helped the network converge and become very good at telling sunspots apart from quiet disk areas. Examples results on the training and validation sets are shown in Figure 7.1, while the reader can take a look at a predicted mask and compare it to the ground truth mask in Figure 7.2.

A detailed look at the results made it clear that the model has a quite good and qualitative understanding of the concept of sunspot, attaining good

performance, regardless of the fact that the sunspot is located on the center of the disk or on the limb. Moreover, the U-Net learns to overcome the limitations of the ground truth due to weak supervision. When sunspots are very small it looks like the network does a better job than the rotation routine at localizing them. This behaviour, together with the fact that the patches are sampled differently on each epoch, generates the high variance of the results (Figure 7.2). A summary of the whole training phase can be examined in Figure 7.3.

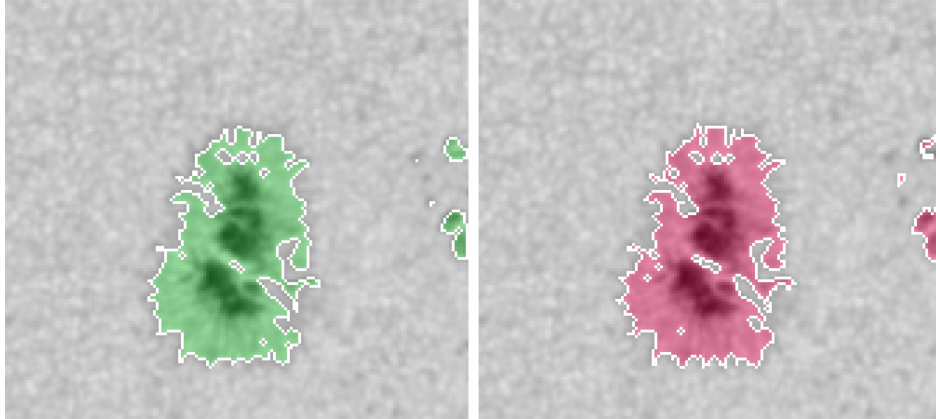


Figure 7.1: Comparison between the predicted mask (green) and the ground truth (red).

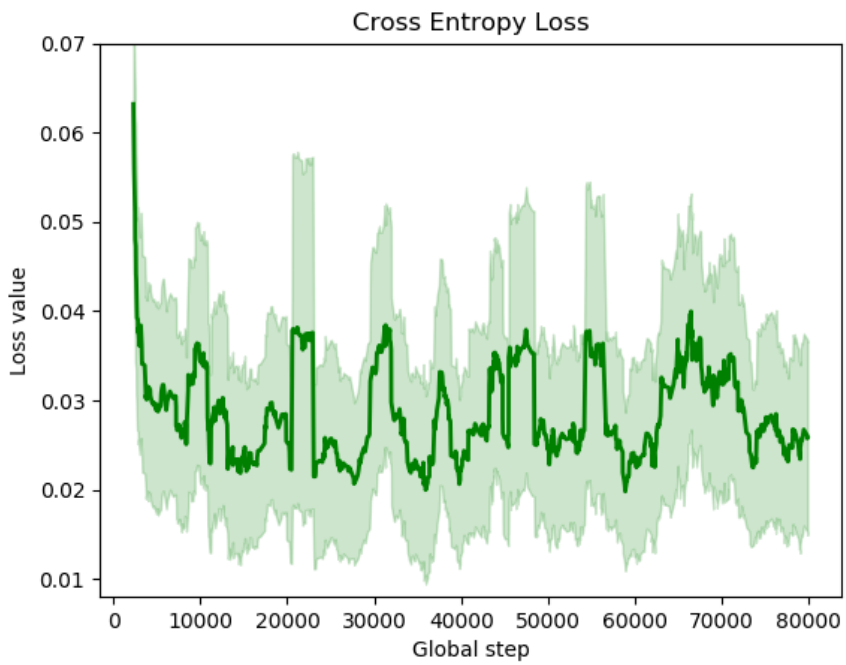


Figure 7.2: U-Net convergence curve on the training set.

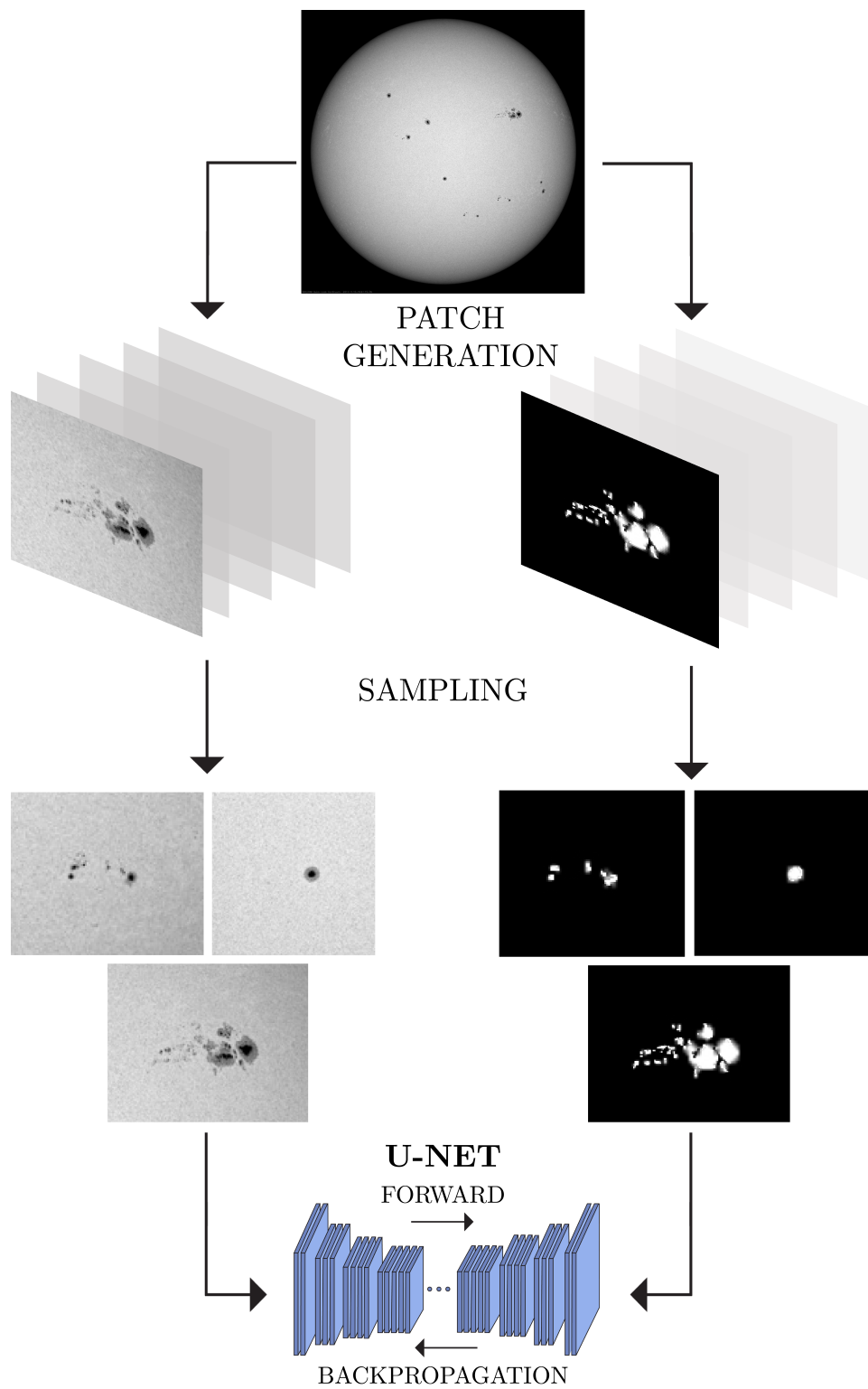


Figure 7.3: Summary of the training of the segmentation phase

### 7.1.2 Sunspot Representations and Classification

Probably, the part of the sunspot index estimation that requires most expert knowledge is the determination of the number of groups. The setting of the problem is mainly unsupervised, because groups change every day and there is no predefined category that generalizes from one day to another.

Clustering seems a very natural approach for this problem. It is indeed, but it is also desirable to leverage the datasets of annotated sunspot groups that are available online to make the algorithm mimic the way humans cluster sunspots. A sensible form of blending together a supervised model with clustering is by using the labels to create good representations of the data. Specifically, we want to take the image of a sunspot and remap it to a feature vector that optimizes the result of the subsequent step. This can be achieved by training a siamese network that embeds sunspots in a way that separates them according to the configuration of the groups. To guide the neural network in the search of the optimal features to extract, a contrastive loss is applied. After the mapping is performed, clustering can be employed to find the number of groups.

At each training episode a ground truth mask that also contains the instances of the clusters is loaded from the dataset. From every group that is present on the disk, an anchor sunspot is randomly chosen. If the sunspot is not alone in the group, a positive example is selected from the same group. Similarly, a negative example is chosen from the other clusters if they exist. A square patch is centered and cropped from the full-disk image for each example that has been sampled. Now, the visual appearance of the sunspot is certainly an important feature, but its position and its total area are crucial as well. So, how can we give this features as inputs to the siamese network? Recent studies have shown that coordinates can be given as input to convolutional neural networks to improve their performance [89]. Thus, the centroids of each sunspot in heliographic coordinates are appended as a channel to the input tensor (the coordinate value is repeated on the whole channel). The same is done with the value of the area, after having it normalized by the total area of the disk. The final input tensor to the convolutional layers has therefore 4 channels (as it can be seen in Figure 7.4) holding respectively:

- 1 channel for the patch of the image that contains the sunspot;
- 2 channels for the heliographic latitude and longitude;



- 1 channel for the normalized area of the sunspot.

After the tensor is propagated, the convolutional layers produce an intermediate representation that, in turn, passes through two fully connected layers, generating a 5-dimensional embedding as output. This procedure gets repeated for all the selected sunspots and for each one of them the loss tries to pull the positive matches together (low Euclidean distance) while pushing the negative ones apart (high Euclidean distance).

At the same time, the convolutional layers are repurposed for classification. The intermediate representation generated by the convolutional layers is also fed to a second, almost identical fully connected block that deals with classification. In this case, the desired output is the modified Zürich class (Z component of the McIntosh classification). Identifying the type of sunspot that is being processed is important because the presence of penumbra surrounding the umbra can be derived directly from the class. As always, the classification error (calculated with the standard cross entropy loss) is back-propagated through the layers. Note that the fully connected layers only receive the gradients coming from their own output while the shared convolutional layers accumulate the gradients coming from both the contrastive loss and the cross entropy loss.

This type of learning paradigm, where the network solves two or more problems at the same time, is called multitask learning [90]. It is based on the idea that what is learned for each task can help other tasks be learned better. This is indeed the case for sunspot clustering. In fact, even humans use group classification as an aid in the process of grouping them together.

The process of training this “two-headed” architecture was very hard, due to the fact that the way positive and negative examples are selected makes a very big impact on the final result. Nonetheless, finally, it was possible to make the network learn both tasks with very good performance. The training curves can be examined in Figure 7.5.

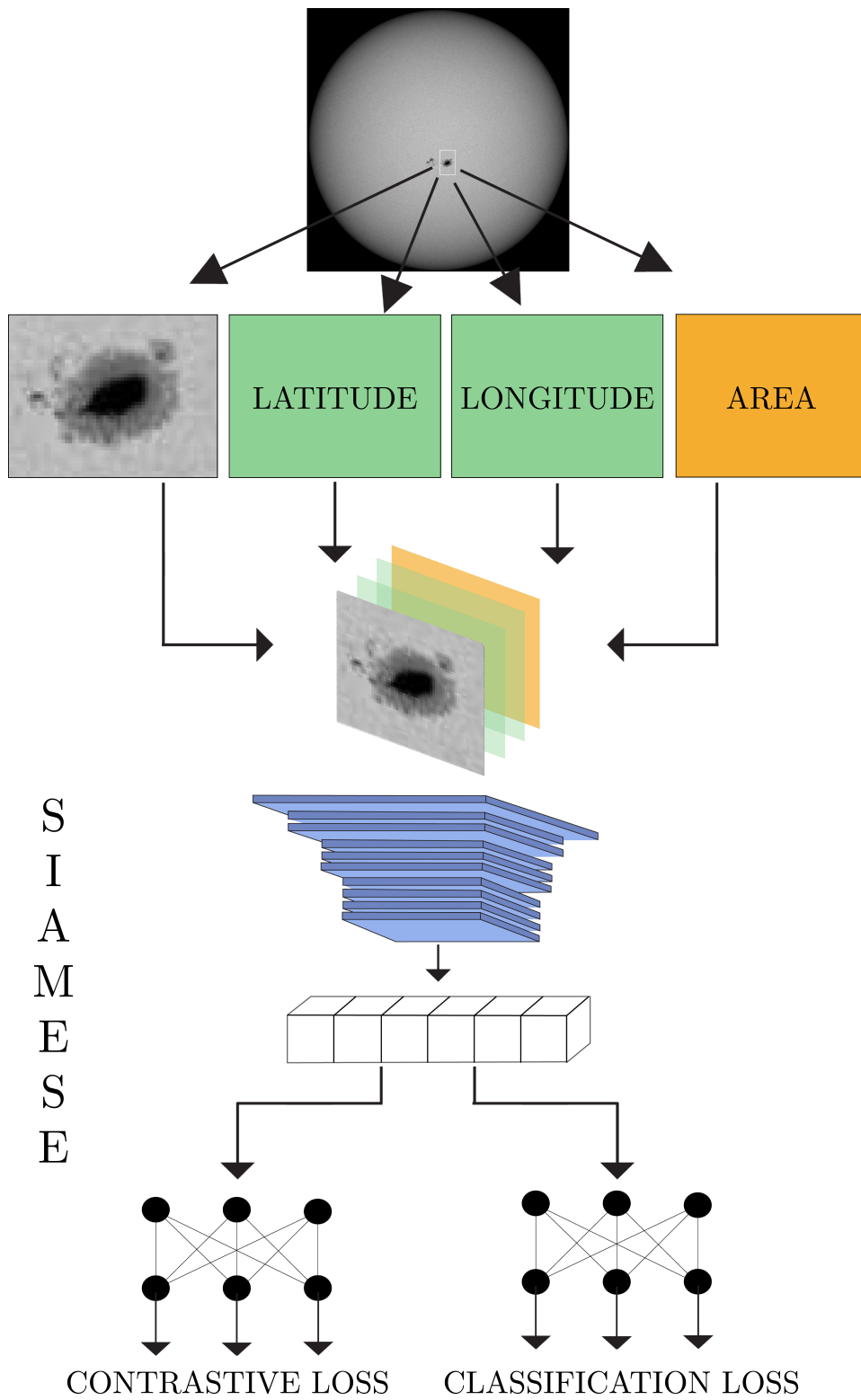


Figure 7.4: Summary of the training of the siamese network

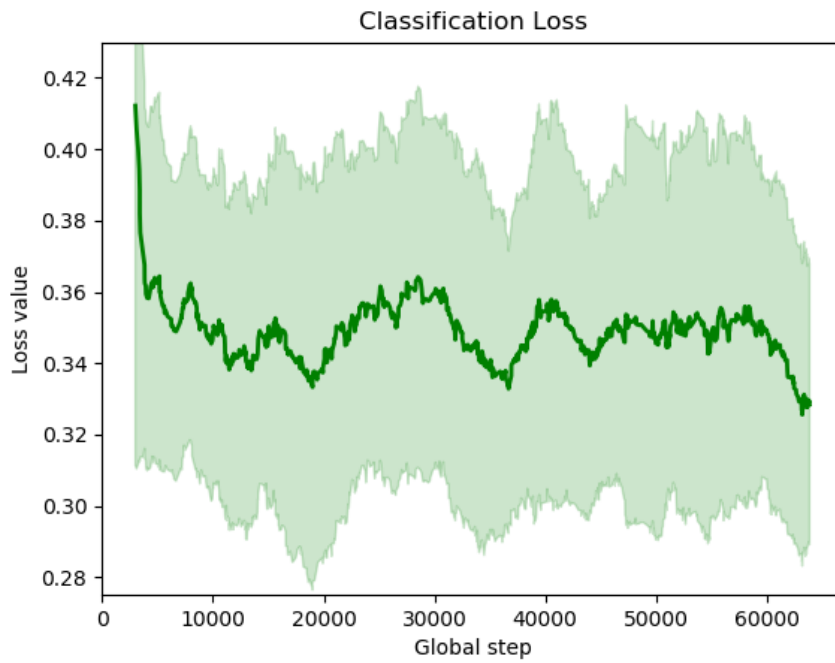
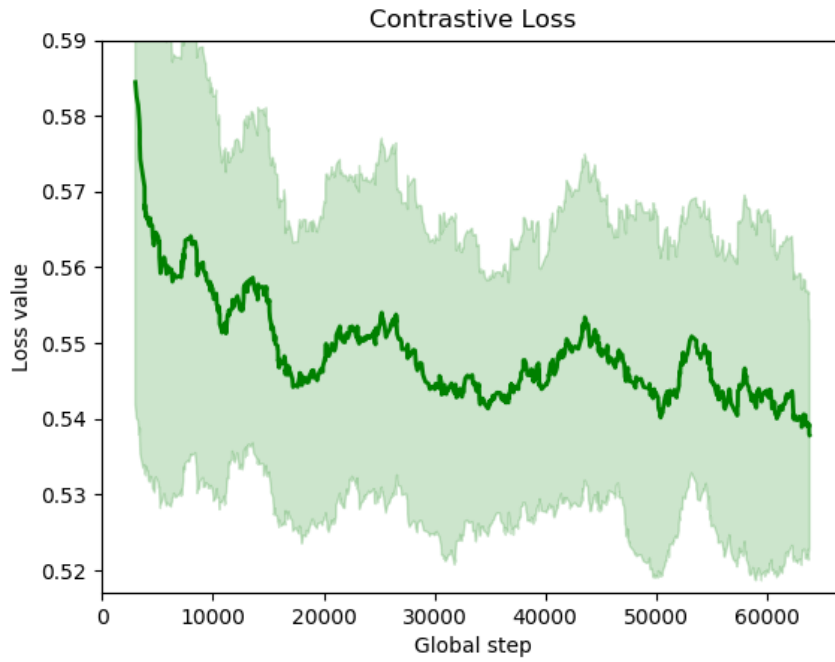


Figure 7.5: Multitask siamese network convergence curve. Note that the implementation of the contrastive loss that has been used constrains the minimum possible value to be 0.5

## 7.2 Automatic Annotation Procedure

After both components of the model have been trained, it is time to join them in order to make predictions on unseen data. However, just stacking the U-Net and the siamese network together is not sufficient and other sub-components are added. In this section we aim at describing in detail all the steps that are necessary to compose a successful annotation algorithm.

Contrary to the training phase, the loading routine is now very simple, since it just needs to fetch the image that is annotated and no mask is needed (it is produced automatically). So, the prediction routine starts loading the image and dividing it into patches, while the parameters of the two neural networks get restored from selected checkpoints. One by one, the patches get sent into the U-Net that takes care of the first segmentation stage. As soon as the whole image has been evaluated, the mask gets reconstructed from the patches. Note that, since the U-Net outputs probabilities, every pixel has a continuous value (between 0 and 1) that needs to be rounded with a threshold in order to obtain a mask where the background has value 0 and the sunspot areas have value 1.

At this point, we know with good confidence that the parts of the image that have been highlighted from the U-Net contain one or more sunspots each. To use this information for the subsequent steps we first need to convert it to a more usable form. The mask is scanned looking for connected components of pixels with value 1. At the same time, statistics are extracted for every component and the data is organized in an array. It is precisely from this information that the inputs for the siamese network are created. In fact, similarly to what was done during training time, the input channels are built using respectively a centered patch, the coordinates of the centroid and the area of the connected components. Coordinates are now in pixel space so they need to be translated to the heliographic system via trigonometry. When this is done, the data is forwarded through the convolutional layers of the siamese network and then the two fully connected heads are evaluated in parallel. The network outputs both the embeddings and the classes of each connected component.

From this moment on, the computation splits in two branches, one that seeks to refine the quality of the segmentation to yield the number of single sunspots, the other that finds the number of groups and assigns each sunspot to one of them. Refining the quality of the segmentation mask means, in this context, to be able to separate the umbra from the penumbra of each

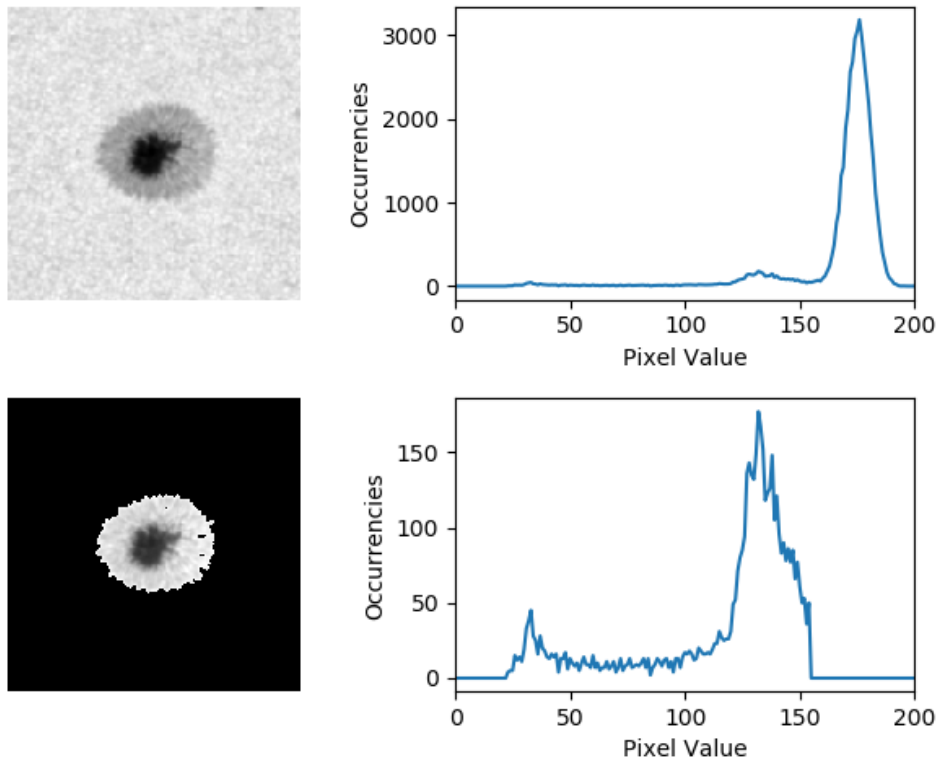


Figure 7.6: Comparison between histograms of the same patch, with or without the mask

connected component of pixels generated by the U-Net. This operation is important because more than one umbrae can be surrounded by the same penumbrae, affecting the sunspot count. First, it is necessary to understand if the sunspot exhibits or not the penumbra. This is easy considered that we can leverage the classification produced by the siamese network together with some simple heuristic concerning the area of the sunspot.

Classes **A** and **B** imply that the sunspots of the group do not have penumbra while classes **C** to **H** all expect penumbra. Small sunspots that usually locate at the periphery of groups are directly discarded with some threshold on the value of the area since we assume that they are very unlikely to carry more than one umbra. This knowledge enables us to segment the connected component using its histogram. In fact, since the division between umbra and penumbra creates a strong intensity discontinuity, the cut-off value is evident in the histogram if the mask of the whole sunspot is known a priori. Referring to Figure 7.6 as an aid to visualization, the two peaks that can be seen around pixel value 30 and 130 are respectively the umbra and the

penumbra. Given this knowledge of the problem we can proceed with two approaches, either fitting two gaussians and assigning each pixel to the most likely one, or simply performing clustering in one dimension fixing the number of clusters to the value 2. Both solutions were explored during the development of this work but in the end clustering was selected because, despite being simpler, it yields the same performance.

At the end, the refinement of the segmentation step reduces to running k-means on the mask generated by the U-Net. The two clusters of pixels are then used to create a new mask that only highlights umbras, and the number of connected components is recalculated from it.

The second branch of the procedure finds the number of groups and assigns each sunspot (connected component of pixels) to one of them, using the embeddings created by the siamese network. Those embeddings are created to optimize the positions of the connected components in a 5-dimensional Euclidean space. As already treated in the previous section, the optimization operated by the siamese network lies in the fact that sunspots belonging to the same group should be close together in the Euclidean space. This is, once again, a great opportunity to unleash the potential of clustering algorithms. However, the main difficulty here is that the number of clusters is not known beforehand, because it is indeed the value that we want to calculate.

DBSCAN seems like a very good match for this problem. In fact, it just needs two parameters: *eps* ( $\varepsilon$ ) which can be estimated statistically on the validation set (see next section) and *minPts* which is trivially set to 0 (otherwise groups composed by a lone sunspot would be interpreted as outliers). Given the parameters and the embeddings as inputs, DBSCAN reliably returns the number of clusters that is useful for later computations.

Now that the image has been annotated and the number of groups and single sunspots is known it is sufficient to find the personal reduction coefficient to be able to calculate the final value of the relative sunspot number. The estimation of the personal reduction coefficient and the *eps* ( $\varepsilon$ ) parameter for DBSCAN are addressed in the next section.

### 7.3 Parameter Estimation

The validation set provides an exceptional opportunity for the estimation of parameters. In fact, many instances of the algorithm can be run using the data that was held out from the training set, so the variation in performance

can be assessed. Various combinations of parameters can be evaluated and the best model can be then used to draw the final results on the test set. Among the others, two parameters have been studied in detail in this work:

- the *eps* ( $\varepsilon$ ) parameter for DBSCAN that represent the minimum distance between two points for them to be considered neighbors;
- the personal reduction coefficient ( $K$ ) of the algorithm, compared to the international sunspot number provided by SILSO.

These two parameters are not independent from each other and therefore they were tested in two consecutive phases, both on the whole validation set.

The first one that needed to be assessed was the value of *eps*. The number of clusters detected from DBSCAN dramatically depends on this parameter, so optimizing it is vital to improve the final performance. Although the variable that drives the performance is the number of clusters, we didn't use it as the measure to optimize *eps* because it is not informative enough. Instead, the Adjusted Random Index (ARI), a performance index that takes into account the configuration of points inside the clusters and compares it to the ground truth was used. ARI is defined as:

$$ARI = \frac{RI - \mathbb{E}[RI]}{\max(RI) - \mathbb{E}[RI]} \quad (7.1)$$

where *RI* (Rand Index) is the number of pairs of objects that are either in the same group or in different groups in both partitions divided by the total number of pairs of objects. The ARI performs a correction with respect to the Rand index in order to account for the variability of the expected value of two random partitions. Thus, the ARI has a value close to 0 for random labeling independently of the number of clusters and exactly 1 when the clusterings are identical.

We tested the algorithm 50 times on the whole validation set, each time with a different value for the *eps* parameter. The results are shown in Figure 7.7. After the estimation of the curve it is possible to take the value that maximizes the performance index and use it as input for the subsequent tests.

After *eps*, we proceeded to estimate the value of the reduction coefficient  $K$  using a similar procedure. This time, instead of maximizing some performance index, the task is to minimize the deviation of our calculation of the

sunspot number from the international sunspot index provided by SILSO. This can be achieved using the root-mean-square (RMS) error between the two calculations over the whole validation set. The procedure was repeated 100 times with values of  $K$  ranging from 0 to 2 (Figure 7.8). The optimal minimal error is attained when the  $K$  coefficient takes the value 0.58. Such a low value for  $K$  is explained by the fact that the images that have been used by our algorithm come from the best observatories in the world. Depending on the optics of the telescope, the amount of detail that can be resolved changes. So, in general, the better the optics, the more sunspots it captures. For this reason, the estimation our algorithm provides is high, and therefore it should be multiplied by a low value to align to the international sunspot number.



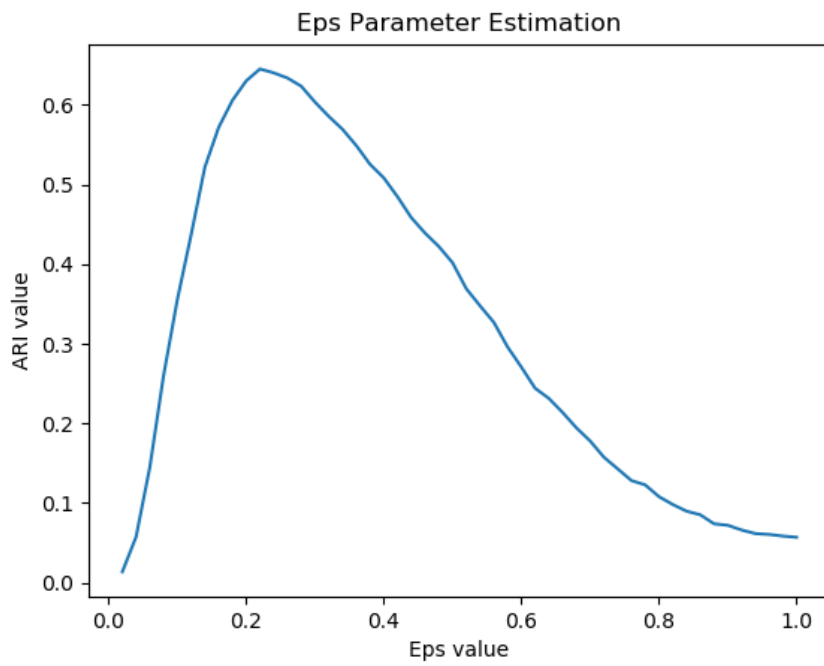


Figure 7.7: ARI value for varying eps values

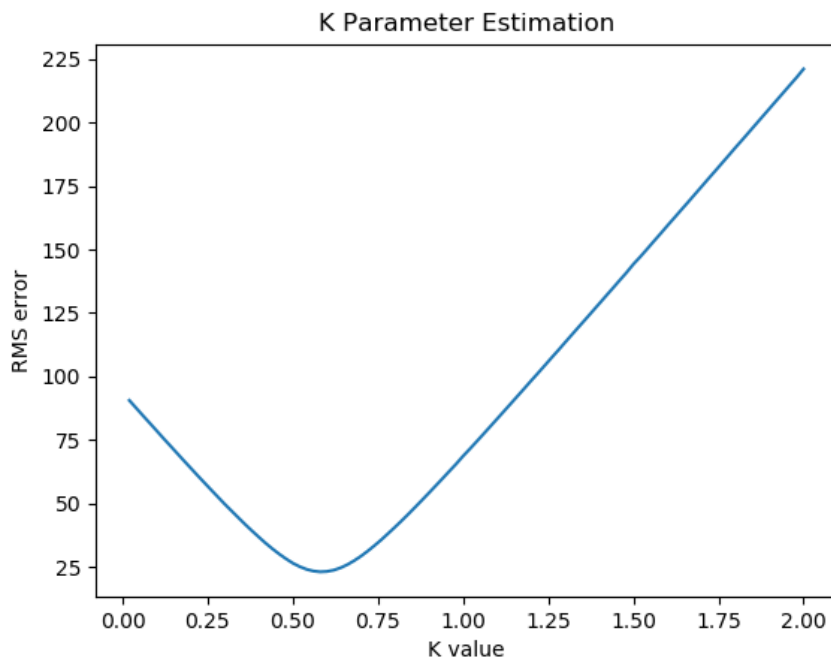


Figure 7.8: RMS error value for varying K value



## Chapter 8

# Results and Future Work

### 8.1 Results

After this long journey through solar physics and deep learning, we finally arrived to the results. But before considering the performance of the algorithm, let's have a closer look at the dataset that will be used as a baseline. Among all the sources that provide an estimation of the superficial activity of the Sun using white-light data, the International Sunspot Number (ISN), provided by SIDC-SILSO, is certainly the most used. Its popularity is due to the fact that it try to conjugate traditional observation techniques with modern standards of accuracy. This is achieved by using a network of stations composed by professionals and amateurs, located all around the world, although mostly in Europe. Also, what is interesting for us is that all annotations are done manually, directly from the eyepiece of the telescope or projecting the light on a white plate and then drawing on top of it.

Behind these choices there is the strong conviction that modern technology is, in some sense, still too limited and human supervision is required for robust estimations. This work proposes the idea that deep learning, being able to learn from human-produced experience, could be the tool that changes this trend. To demonstrate this point we need to be able to compare the results of our solution to the ones obtained by humans. Luckily, SIDC-SILSO includes in the dataset the value of the standard deviation as well as the number of data points included in the calculation (number of stations). This, besides allowing the assessment the precision of the value, enables us to compare the expected error of humans with ours.

The final results that will be shown in this chapter are all computed on the test set (containing both ground-based and space-based observations) and

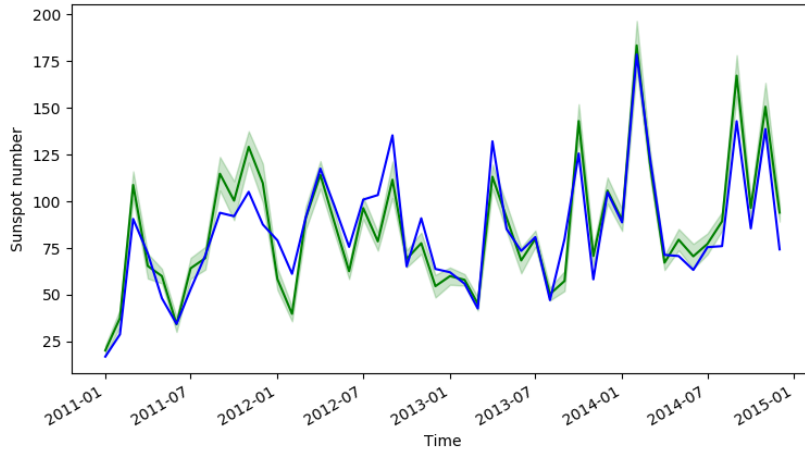


Figure 8.1: Comparison between monthly averages of our algorithm (blue) and the international sunspot number (green) with confidence interval.

employ the parameters that were estimated on the validation set. For the prediction of the sunspot number on the test images, the procedure explained in section 7.2 is used. Then, day by day, the detected number of sunspots is compared to the one found in the dataset, and the error is computed. To help visualization, the final value and the error are averaged monthly for both our results and the baseline. Although monthly smoothing works well to get an idea of the behaviour of the algorithm, we remind the reader that the average is computed only on the days that belong to the test set, and therefore it does not represent a good indicator of the overall trend of the solar cycle.

Figure 8.1 shows the sunspot number derived with the relative sunspot formula as calculated by our algorithm (blue) compared to the ISN provided by SIDC-SILSO. The plot also shows the confidence interval (standard deviation) for the ISN measurement. The algorithm seems to have learned to identify the tendency of the activity, in fact, the two lines follow roughly the same trend. Although, a closer inspection reveals that, in general, the deviation from the ISN is not neglectable.

Comparing the error generated by our algorithm and the standard error of the ISN (Figure 8.2), we discover that there is no strong evidence of correlation between the two errors. This means that the reasons why humans make mistakes are probably not the same as the ones that make our algorithm

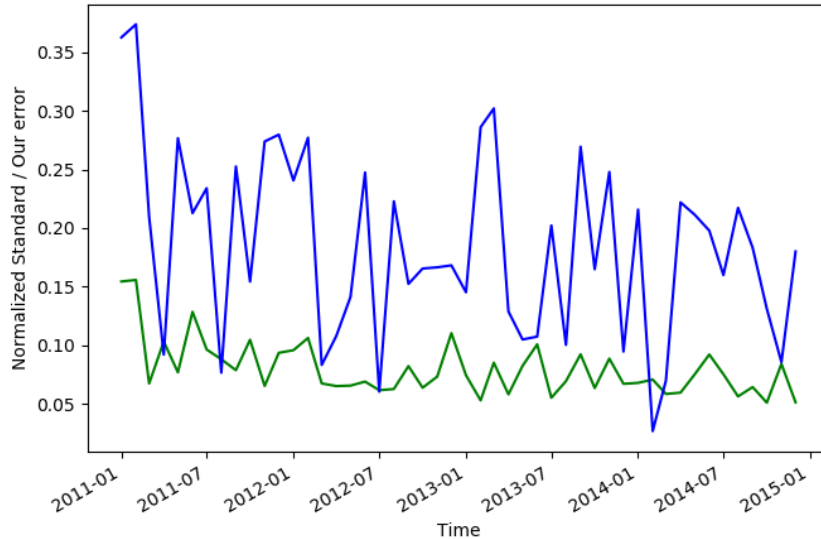


Figure 8.2: Comparison between normalized ISN standard error (green) and the normalized error of our algorithm (blue)

fail. Anyhow, the error produced by our solution seems to be larger than the average human error. Summing up all the errors together over the whole test set, it turns out that the average errors are:

- around 13 units (18%) for our algorithm;
- around 6 units (8%) for the the average station in the ISN network.

Since the SIDC-SILSO receives the pre-computed sunspot number from the station of the network, they do not need to calculate the number of groups and therefore they do not provide it in the data. This makes it a lot harder to understand the causes of the misalignment between the two measurements. Anyway we can make pretty confident hypotheses on why this happens. The first reason is related to the annotation method. In fact, the data (DPD dataset) provided to the algorithm for training and validation is annotated a posteriori from images of the disk, while the stations of the SIDC-SILSO network make live observations looking at the Sun. This problem is really difficult to overcome, because live observations are hard to turn into a digital format automatically and create a dataset with them.

Another solid reason for the deviation in the estimation of the sunspot number has its roots in the types of observatories that are used. On the one hand, we rely on the best observatories that are available, which are able to detect even small pores on the surface of the Sun. On the other hand, two thirds of the observatories of the SIDC-SILSO network use amateurial instrumentations that are not able to resolve all the sunspots on the disk. This hypotheses is also supported by the fact that the personal reduction coefficient of our algorithm is very low, meaning that it is able to detect a large amount of sunspots.

Nonetheless, despite the marginal problems that have been identified, we succeeded in the creation of an algorithm that can capture the fluctuations in the sunspot count and therefore it can be used to predict the activity of our star. As the title of this thesis suggests, this is an initial deep learning approach to the problem, thus we hope that many more similar algorithms will be explored in the future, building upon ours and increasing its performance.

## 8.2 Future Work

The procedure described in this thesis can be considered complete, in the sense that we proposed a solution that works from the beginning to the end without overlooking intermediate steps. Although this, there is still a lot of work to do in order to make the algorithm ready to be used for scientific purposes reliably.

Certainly, the part that needs most revision is the data. In fact, although the Decebren database (DPD) is a great place to start because it provides detailed annotations, the quantity of data could be increased. So far, only the peak years of the solar cycle 24 have been used to train the models, previous cycles should be added to increase the variability of the data. The deep learning models that were used would undoubtedly benefit from a wider set of examples to draw knowledge. From the work, we also learned that the annotations of the images should depend on the scope of application, so they need to be selected accordingly.

Another aspect that should be explored better is the minimization of dependencies among training, validation and test set. Even though we took them into account in the process of splitting of the dataset, we didn't succeed in removing them completely. For the splitting to be independent, it

would be required to check if each sunspot group appears in only one of the three chunks of data. Removing the images where the groups are repeated is always possible, at the cost of sensibly reducing the size of the dataset. Another approach is to use an algorithm that minimizes the groups that belong to the overlapping of the three sets. Since brute-forcing over thousands of groups is out of question, genetic algorithms may be used to find suboptimal solutions.

For what concerns the final results, more test cases can be designed. The comparison with the international sunspot number produced contrasting results. To clarify the situation, it is possible to analyze the deviation of our sunspot number with other datasets. For example, the American National Oceanic and Atmospheric Administration (NOAA) produces a similar time series, whose properties are, though, a bit different from the ISN. Besides that, other indicators can be analyzed. For instance, it is not unusual for the scientists to refer to the “backbone” number of groups for some studies, or even simply to the total area of sunspots on the disk. It is unclear which one of these measures is most suited as a base for more complex studies, but that is actually out of the scope of this work. It is sufficient for us to be able to compare our results with some human-produced ones. The drawback of these other datasets, that is also the reason why ISN was selected, is that they lack the measure of standard deviation because they are not computed ensembling human measurements.

On the other hand, for what concerns more structural and algorithmical aspects, even though our framework seems well established, some minor modifications can be made. For instance, the triplet loss can be used instead of the contrastive one for the training of the siamese network, to see if it improves the performance. The segmentation as well may be improved by moving the discovery of the umbra inside the U-Net. This can be achieved assigning different classes for umbra and penumbra, and then optimizing the network for the new multi-class problem. A dramatically different approach, instead, would be to perform the clustering phase inside the U-Net, together with semantic segmentation. This is called instance segmentation and it is known to be very hard to train, hence why it has been discarded in this work. The great advantage of being able to recognize instances would be the simplification of the workflow, since only one network would be evaluated to get both the number of clusters and single sunspots.

Some of these solutions will probably be explored in the future in the hope of improving this tool and making it publicly available online for scientific

and educational purposes.



# Bibliography

- [1] Luke de Oliveira, Michela Paganini, and Benjamin Nachman. Learning particle physics by example: location-aware generative adversarial networks for physics synthesis. *Computing and Software for Big Science*, 1(1):4, 2017.
- [2] Christopher J Shallue and Andrew Vanderburg. Identifying exoplanets with deep learning: A five-planet resonant chain around kepler-80 and an eighth planet around kepler-90. *The Astronomical Journal*, 155(2):94, 2018.
- [3] Sean McGregor, Dattaraj Dhuri, Anamaria Berea, and Andres Munoz-Jaramillo. FlareNet: A Deep Learning Framework for Solar Phenomena Prediction. In *NIPS Workshop on Deep Learning for Physical Sciences*, Long Beach, 2017.
- [4] F. Laclare, C. Delmas, J. P. Coin, and A. Irbah. Measurements and variations of the solar diameter. *Solar Physics*, 166(2):211–229, Jul 1996.
- [5] USNO. 2014 astronomical constants. [http://asa.usno.navy.mil/static/files/2014/Astronomical\\_Constants\\_2014.pdf](http://asa.usno.navy.mil/static/files/2014/Astronomical_Constants_2014.pdf).
- [6] NASA. Sun-earth connection. [https://www.nasa.gov/mission\\_pages/themis/auroras/sun\\_earth\\_connect.html](https://www.nasa.gov/mission_pages/themis/auroras/sun_earth_connect.html).
- [7] Guenther, D. B., Demarque, P., Kim, Y.-C., & Pinsonneault, and M. H. *Standard solar model*. The American Astronomical Society, March 1, 1992.
- [8] Sylvaine Turck-Chi ze. The standard solar model and beyond. *Journal of Physics: Conference Series*, 665:012078, 01 2016.
- [9] A. Poland B. Fleck, V. Domingo. The soho mission. *Springer Science + Business Media B. V.*, 1995.

- [10] Dziembowski, W. A., Goode, P. R., Pamyatnykh, A. A., & Sienkiewicz, and R. *Updated seismic solar model*. NASA, 1995.
- [11] Wikipedia. Interior structure of the sun. [https://en.wikipedia.org/wiki/Sun#Structure\\_and\\_fusion](https://en.wikipedia.org/wiki/Sun#Structure_and_fusion).
- [12] Wikipedia user:Kelvin13. Visualization of the interior structure of the sun. <https://commons.wikimedia.org/wiki/User:Kelvin13>.
- [13] NorthWest Research Associates. The convection zone. [https://www.cora.nwra.com/~werne/eos/text/convection\\_zone.html](https://www.cora.nwra.com/~werne/eos/text/convection_zone.html).
- [14] NASA Holly Zell. Solar rotation varies by latitude. [https://www.nasa.gov/mission\\_pages/sunearth/science/solar-rotation.html](https://www.nasa.gov/mission_pages/sunearth/science/solar-rotation.html).
- [15] J. G. Beck. A comparison of differential rotation measurements - (Invited Review). *Solar Physics*, 191:47–70, January 2000.
- [16] Henry C King. *The history of the telescope*. Courier Corporation, 2003.
- [17] Heinrich Schwabe. Solar observations during 1843. *Astronomische Nachrichten*, 20(495):234–235, 1843.
- [18] SILSO/SIDC. Sunspot number graphics. <http://www.sidc.be/silso/ssngraphics>.
- [19] José M Vaquero. Historical sunspot observations: a review. *Advances in Space Research*, 40(7):929–941, 2007.
- [20] SDO. Cycle 25 observations in sdo hmi imagery. [http://www.solen.info/solar/cycle25\\_spots.html](http://www.solen.info/solar/cycle25_spots.html).
- [21] Hugh Hudson. A sunspot from cycle 25 for sure. [http://sprg.ssl.berkeley.edu/~tohban/wiki/index.php/A\\_Sunspot\\_from\\_Cycle\\_25\\_for\\_sure](http://sprg.ssl.berkeley.edu/~tohban/wiki/index.php/A_Sunspot_from_Cycle_25_for_sure).
- [22] Tony Phillips. A sunspot from the next solar cycle. <https://spaceweatherarchive.com/2018/11/20/a-sunspot-from-the-next-solar-cycle/>.
- [23] VG Ivanov and EV Miletsky. Spörer’s law and relationship between the latitude and amplitude parameters of solar activity. *Geomagnetism and Aeronomy*, 54(7):907–914, 2014.
- [24] Patrick S McIntosh. The classification of sunspot groups. *Solar Physics*, 125(2):251–267, 1990.

- [25] VMS Carrasco, L Lefèvre, JM Vaquero, and MC Gallego. Equivalence relations between the cortie and zürich sunspot group morphological classifications. *Solar Physics*, 290(5):1445–1455, 2015.
- [26] SILSO/SIDC. Codes, terminology and classifications. <http://sidc.oma.be/educational/classification.php>.
- [27] SILSO/SIDC. Homepage of the sunspot index and long-term solar observations project. <http://sidc.be/silso/home>.
- [28] Frédéric Clette, Leif Svalgaard, José M. Vaquero, and Edward W. Cliver. *Revisiting the Sunspot Number*, pages 35–103. Springer New York, New York, NY, 2015.
- [29] Robert Henry Dicke and H Mark Goldenberg. The oblateness of the sun. *The Astrophysical Journal Supplement Series*, 27:131, 1974.
- [30] WT Thompson. Coordinate systems for solar image data. *Astronomy & Astrophysics*, 449(2):791–803, 2006.
- [31] JJ Curto, M Blanca, and E Martínez. Automatic sunspots detection on full-disk solar images using mathematical morphology. *Solar Physics*, 250(2):411–429, 2008.
- [32] ESA. Esa offers a new way of looking at the sun. <http://sci.esa.int/soho/48123-esa-offers-a-new-way-of-looking-at-the-sun/>.
- [33] John Canny. A computational approach to edge detection. In *Readings in computer vision*, pages 184–203. Elsevier, 1987.
- [34] Serge Beucher et al. The watershed transformation applied to image segmentation. *SCANNING MICROSCOPY-SUPPLEMENT-*, pages 299–299, 1992.
- [35] Jan Puzicha, Thomas Hofmann, and Joachim M Buhmann. Histogram clustering for unsupervised segmentation and image retrieval. *Pattern Recognition Letters*, 20(9):899–909, 1999.
- [36] et al. Zharkov, Sergei. Automated recognition of sunspots on the soho/mdi white light solar images. *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 2004.
- [37] T Pettauer and PN Brandt. On novel methods to determine areas of sunspots from photoheliograms. *Solar Physics*, 175(1):197–203, 1997.

- [38] M Turmon, JM Pap, and S Mukhtar. Statistical pattern recognition for labeling solar active regions: application to soho/mdi imagery. *The Astrophysical Journal*, 568(1):396, 2002.
- [39] Cis Verbeeck, Paul A Higgins, Tufan Colak, Fraser T Watson, Veronique Delouille, Benjamin Mampaey, and Rami Qahwaji. A multi-wavelength analysis of active regions and sunspots by comparison of automatic detection algorithms. *Solar Physics*, 283(1):67–95, 2013.
- [40] Fraser Watson, Lyndsay Fletcher, Silvia Dalla, and Stephen Marshall. Modelling the longitudinal asymmetry in sunspot emergence: the role of the wilson depression. *Solar Physics*, 260(1):5–19, 2009.
- [41] V Barra, V Delouille, Mathieu Kretzschmar, and J-F Hochedez. Fast and robust segmentation of solar euv images: algorithm and results for solar cycle 23. *Astronomy & Astrophysics*, 505(1):361–371, 2009.
- [42] Paul A Higgins, Peter T Gallagher, RT James McAteer, and D Shaun Bloomfield. Solar magnetic feature detection and tracking for space weather monitoring. *Advances in Space Research*, 47(12):2105–2117, 2011.
- [43] Tufan Colak and R Qahwaji. Automated mcintosh-based classification of sunspot groups using mdi images. *Solar Physics*, 248(2):277–296, 2008.
- [44] Tufan Colak and R Qahwaji. Automated solar activity prediction: a hybrid computer platform using machine learning and solar imaging for automated prediction of solar flares. *Space Weather*, 7(6), 2009.
- [45] Tufan Colak, Rami Qahwaji, Stan Ipson, and Hassan Ugail. Representation of solar features in 3d for creating visual solar catalogues. *Advances in Space Research*, 47(12):2092–2104, 2011.
- [46] J Schou, PH Scherrer, RI Bush, R Wachter, Savita Couvidat, MC Rabello-Soares, RS Bogart, JT Hoeksema, Y Liu, TL Duvall, et al. Design and ground calibration of the helioseismic and magnetic imager (hmi) instrument on the solar dynamics observatory (sdo). *Solar Physics*, 275(1-2):229–259, 2012.
- [47] B Ravindra, TG Priya, K Amareswari, M Priyal, AA Nazia, and D Banerjee. Digitized archive of the kodaikanal images: Representative results of solar cycle variation from sunspot area determination. *Astronomy & Astrophysics*, 550:A19, 2013.

- [48] Edward R Dougherty and Roberto A Lotufo. *Hands-on morphological image processing*, volume 59. SPIE press, 2003.
- [49] Homepage of the ebro observatory. <http://www.obsebre.es/en/>.
- [50] JJ Curto, M Blanca, JG Solé, and Observatorio del Ebro. Automatic detection of sunspots and group classification from white full disc images. In *Presentation at Solar Image Recognition Workshop. Brussels*, 2003.
- [51] Ragadeepika Pucha, KM Hiremath, and Shashanka R Gurumath. Development of a code to analyze the solar white-light images from the kodaikanal observatory: Detection of sunspots, computation of heliographic coordinates and area. *Journal of Astrophysics and Astronomy*, 37(1):3, 2016.
- [52] James R Lemen, David J Akin, Paul F Boerner, Catherine Chou, Jerry F Drake, Dexter W Duncan, Christopher G Edwards, Frank M Friedlaender, Gary F Heyman, Neal E Hurlburt, et al. The atmospheric imaging assembly (aia) on the solar dynamics observatory (sdo). In *The Solar Dynamics Observatory*, pages 17–40. Springer, 2011.
- [53] T Colak and R Qahwaji. Automatic sunspot classification for real-time forecasting of solar activities. In *2007 3rd International Conference on Recent Advances in Space Technologies*, pages 733–738. IEEE, 2007.
- [54] Trung Thanh Nguyen, Claire P Willis, Derek J Paddon, Sinh Hoa Nguyen, and Hung Son Nguyen. Learning sunspot classification. *Fundamenta Informaticae*, 72(1-3):295–309, 2006.
- [55] Sinh Hoa Nguyen, Trung Thanh Nguyen, and Hung Son Nguyen. Rough set approach to sunspot classification problem. In *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, pages 263–272. Springer, 2005.
- [56] Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, CORNELL AERONAUTICAL LAB INC BUFFALO NY, 1961.
- [57] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Departmental Papers (CIS)*, page 107, 2000.
- [58] Guy Barrett Coleman and Harry C Andrews. Image segmentation by clustering. *Proceedings of the IEEE*, 67(5):773–785, 1979.

- [59] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.
- [60] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.
- [61] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [62] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.
- [63] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme recognition using time-delay neural networks. *Backpropagation: Theory, Architectures and Applications*, pages 35–61, 1995.
- [64] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.
- [65] Stanford Andrej Karpathy, Justin Johnson. Convolutional neural networks for visual recognition. <https://cs231n.github.io/>.
- [66] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [67] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 240–248. Springer, 2017.
- [68] Thorvald Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biol. Skr.*, 5:1–34, 1948.

- [69] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
- [70] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [71] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [72] Michal Drozdal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury, and Chris Pal. The importance of skip connections in biomedical image segmentation. In *Deep Learning and Data Labeling for Medical Applications*, pages 179–187. Springer, 2016.
- [73] Mahamed GH Omran, Andries P Engelbrecht, and Ayed Salman. An overview of clustering methods. *Intelligent Data Analysis*, 11(6):583–605, 2007.
- [74] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.
- [75] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- [76] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.
- [77] Guohua Shen, Tomoyasu Horikawa, Kei Majima, and Yukiyasu Kamitani. Deep image reconstruction from human brain activity. *PLoS computational biology*, 15(1):e1006633, 2019.
- [78] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [79] Ian Jolliffe. *Principal component analysis*. Springer, 2011.
- [80] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

- 
- [81] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.
- [82] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848, 2017.
- [83] T Baranyi, L Gyóri, and A Ludmány. On-line tools for solar data compiled at the debrecen observatory and their extensions with the greenwich sunspot data. *Solar Physics*, 291(9-10):3081–3102, 2016.
- [84] L Gyóri, A Ludmány, and T Baranyi. Comparative analysis of debrecen sunspot catalogues. *Monthly Notices of the Royal Astronomical Society*, 465(2):1259–1273, 2016.
- [85] Frédéric Clette, Leif Svalgaard, José M Vaquero, and Edward W Cliver. Revisiting the sunspot number. *Space Science Reviews*, 186(1-4):35–103, 2014.
- [86] Frank Hill, Piet Martens, Keji Yoshimura, Joseph Gurman, Joseph Hourclé, George Dimitoglou, Igor Suárez-Solá, Steve Wampler, Kevin Reardon, Alisdair Davey, et al. The virtual solar observatory - a resource for international heliophysics research. *Earth, Moon, and Planets*, 104(1-4):315–330, 2009.
- [87] Stuart J Mumford, Steven Christe, David Pérez-Suárez, Jack Ireland, Albert Y Shih, Andrew R Inglis, Simon Liedtke, Russell J Hewett, Florian Mayer, Keith Hughitt, et al. Sunpy - python for solar physics. *Computational Science & Discovery*, 8(1):014009, 2015.
- [88] Sldtk, the solar limb darkening toolkit. <https://github.com/DonkeyShot21/SLDTk>.
- [89] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, pages 9628–9639, 2018.
- [90] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.
- [91] NOAA. Space weather glossary. <https://www.swpc.noaa.gov/content/space-weather-glossary>.



# Appendix A

## Glossary

The definitions in this glossary are taken from NOAA's space weather glossary [91].

### **Active Region:**

A localized, transient volume of the solar atmosphere in which plages, sunspots, faculae, flares, etc. may be observed.

### **Aurora:**

A faint visual (optical) phenomenon on the Earth associated with geomagnetic activity, which occurs mainly in the high-latitude night sky. Typical auroras are 100 to 250 km above the ground. The Aurora Borealis occurs in the northern hemisphere and the Aurora Australis occurs in the southern hemisphere.

### **Corona:**

The outermost layer of the solar atmosphere, characterized by low densities and extraordinarily high temperatures that extends to several solar radii. The heating of the corona is still a mystery. The shape of the corona is different at solar maximum and at solar minimum.

### **Coronal Hole:**

An extended region of the corona, exceptionally low in density (large open "gaps"), and associated with photospheric regions. Coronal holes are closely associated with those regions on the Sun that have an "open" magnetic geometry, that is, the magnetic field lines associated with them extend far outward into interplanetary space, rather than looping back to the photosphere. Ionized material can flow easily

along such a path, and this in turn aids the mechanism that causes high speed solar wind streams to develop.

**Coronal Mass Ejection:**

An observable change in coronal structure that occurs on a time scale between a few minutes and several hours, and involves the appearance of a new discrete, bright, white light feature in the coronagraph field of view, that displays a predominantly outward motion. The solar corona material is massive in size (they can occupy up to a quarter of the solar limb), and frequently accompanied by the remnants of an eruptive prominence, and less often by a strong solar flare. The leading edges of fast-moving CMEs drive giant shock waves before them through the solar wind at speeds up to 1200 km per second. Some astronomers believe that CMEs are the crucial link between a solar disturbance its propagation through the heliosphere, and the effects on the Earth.

**Differential Rotation:**

The change in solar rotation rate with latitude. Low latitudes rotate at a faster angular rate (approx. 14 degrees per day) than do high latitudes (approx. 12 degrees per day). For example, the equatorial rotation period is 27.7 days compared to 28.6 days at latitude 40 degrees.

**Disk:**

The visible surface of the Sun (or any heavenly body) projected against the sky.

**Eruptive:**

Solar activity levels with at least one radio event and several chromospheric events per day.

**Facula:**

A bright cloud-like feature located a few hundred km above the photosphere near sunspot groups, seen in white light. Facula are seldom visible except near the solar limb, although they occur all across the Sun. Facula are clouds of emission that occur where a strong magnetic field creates extra heat (about 300 degrees K above surrounding areas).

**Filament:**

A mass of gas suspended over the photosphere by magnetic fields and seen as dark lines threaded over the solar disk. A filament on the limb of the Sun seen in emission against the dark sky is called a prominence.

**Flux:**

The rate of flow of a physical quantity through a reference surface.

**Geomagnetic Field:**

The magnetic field observed in and around the Earth. The intensity of the magnetic field at the Earth's surface is approximately 0.32 gauss at the equator and 0.62 gauss at the north pole.

**H-alpha:**

This absorption line of neutral hydrogen falls in the red part of the visible spectrum and is convenient for solar observations. The H-alpha line is universally used for observations of solar flares and prominences.

**Helioseismology:**

A method for studying the Sun by utilizing waves that propagate throughout the star to measure its invisible internal structure and dynamics.

**Limb:**

The edge of the solar disk.

**Magnetogram:**

Solar magnetograms are a graphic representation of solar magnetic field strengths and polarity.

**Penumbra:**

The sunspot area that may surround the darker umbra or umbrae. It consists of linear bright and dark elements radial from the sunspot umbra.

**Plage:**

An extended emission feature of an active region that exists from the emergence of the first magnetic flux until the widely scattered remnant magnetic fields merge with the background. This bright feature is found in the vicinity of virtually all active sunspot groups and occurs on a larger scale and are brighter than facula. Plage is French for "beach," because each plage looks like light-colored sand against the darker structures around them.

**Plasma:**

Any ionized gas, that is, any gas containing ions and electrons.

**Prominence:**

A term identifying cloud-like features in the solar atmosphere. The features appear as bright structures in the corona above the solar limb and as dark filaments when seen projected against the solar disk.

**Quiet:**

Solar activity levels with less than one chromospheric event per day.

**Solar Cycle:**

The approximately 11-year quasi-periodic variation in frequency or number of solar active events.

**Solar Maximum:**

The month(s) during the solar cycle when the 12-month mean of monthly average sunspot numbers reaches a maximum.

**Solar Minimum:**

The month(s) during the solar cycle when the 12-month mean of monthly average sunspot numbers reaches a minimum.

**Umбра:**

The dark core or cores (umbrae) in a sunspot with penumbra, or a sunspot lacking penumbra.


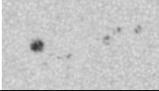
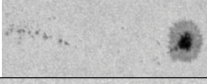
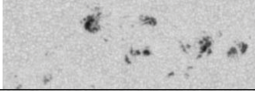
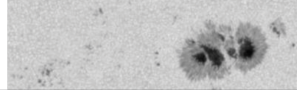
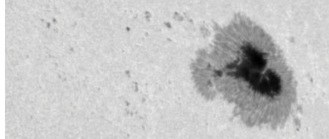
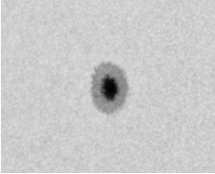
**White-Light:**

Sunlight integrated over the visible portion of the spectrum (4000 - 7000 angstroms) so that all colors are blended to appear white to the eye.

## Appendix B

# McIntosh Classification Example Images

Figure B.1 shows example images taken from SDO for each McIntosh class.

CLASS	EXAMPLE IMAGE
A	
B	
C	
D	
E	
F	
H	

*Figure B.1: Example images taken from SDO for each McIntosh class*



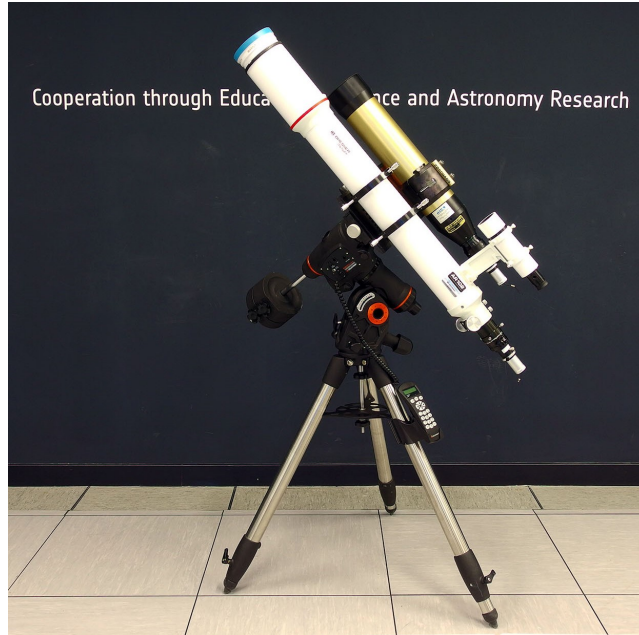
## Appendix C

# Helios Processing Pipeline

This appendix describes the first part of the work that was done at the Helios observatory of the European Space Agency. The observatory belongs to the CESAR (Cooperation through Education in Science and Astronomy Research) initiative, whose objective is to provide students from European secondary schools and universities with hands-on experience in Astronomy research in general and in Radio Astronomy and Optical Astronomy in particular. Helios was installed at ESAC (European Space Astronomy Centre) in 2012 and includes two telescopes (Figure C.1):

- Coronado Solarmax II 90, H-alpha, double stack, with specifications:
  - Aperture: 90mm
  - Focal Length: 800mm
  - Bandwidth:  $<0.5 \text{ \AA}$
- Bresser AR-102, visible (white-light), with specifications:
  - Aperture: 102mm
  - Focal Length: 1000mm
  - Solar Filter: BAADER AstroSolar Safety Filter

The two telescopes are mounted on the same robotic arm (Figure C.1) and are very different from each other. The main difference is in the type of filter they have on board. On the one hand, the filter that comes with the Coronado Solarmax isolates the H-alpha band, a deep-red visible spectral line. H-alpha is particularly useful in solar astronomy because it enables the observation of the atmosphere of the Sun. On the other hand, the filters that are used for white-light observation do not reduce the portion of



*Figure C.1: A picture of the two telescopes*

the spectrum that enters the scope, but rather decrease the intensity of the radiation. Therefore each pixel of the sensor, attached at the bottom of the scope, receives the whole visible spectrum and integrates it to obtain the total intensity.

Given these differences, dedicated processing techniques are used for each telescope. Nonetheless, the preliminary steps of the two pipelines are shared, although with different parameters, and they are therefore presented together. In fact, we can logically divide the pipelines in two phases (which are treated in the next sections):

- **Preliminary adjustments**, performed for both telescopes with different parameters;
- **Feature enhancement**, dedicated for each telescope.

## C.1 Preliminary Adjustments

The first phase of the processing starts with a set of quality control checks. For each image we need to make sure that they are not corrupted or completely black, since files could be damaged during the transfer from the observatory to the servers that are allocated for the processing. Then, a



second check, called cloud control, is performed to retain only the images where the Sun is perfectly visible, and discard the cloudy ones.

Once it is certain that the images are up to standard, the pipeline proceeds to perform some corrections that are necessary to remove the artifacts introduced by the sensor. In fact, even the best sensors suffer from variations in the pixel sensitivity of the detector and distortions in the optical path. Luckily, is possible to account for these errors using two techniques that are really popular in digital photography: **dark-frame subtraction** and **flat-field correction**. The former tries to minimize the noise due to defective pixels and dark currents, while the latter adjusts for the relative sensitivity of pixels. The dark-frame is created by taking long exposure images in complete darkness, or, in this case, when the lid of the telescope is on. The flat-field, instead, is generated attaching an omogeneous light source over the instrument, to measure how much each pixel reacts to it. Thus, two calibration images are used to adjust the images with the following formula:

$$C = \frac{(R - D)}{(F - D)} \quad (\text{C.1})$$

where  $C$  is the corrected image,  $R$  is the raw image, and  $F$  and  $D$  are the flat-field and the dark-frame respectively.

The third subphase of the preliminary adjustments phase deals with image centering and axial tilt correction. Although, as explained in Appendix D, the telescope itself and the master control program already account for the tracking of the Sun, the limited accuracy of mechanical arms makes it necessary to use image processing techniques to align and center the solar disk to the frame. Moreover, in this case, centering the image is made easier by the fact that the shape of the object we are looking for is known and its size can be calculated with simple geometry. In fact, first, the sun can be considered a sphere, since its oblateness is neglectable; and second, the size of the apparent radius can be determined from the Sun-Earth distance. For the latter calculation it is sufficient to know the radius in pixels at one point of the orbit (we used the perihelion), and remember that the apparent size of an object depends on its distance. Therefore the radius can be calculated as:

$$A_{pix} = \left( \frac{Ap_{pix}}{Ap_{rad}} \right) \arctan \left( \frac{r}{d} \right) \quad (\text{C.2})$$

where  $A_{pix}$  is the desired radius in pixels at distance  $d$ ,  $Ap_{pix}$  and  $Ap_{rad}$  are respectively the apparent radius at perihelion in pixels and radians,  $r$  is the absolute radius of the sun and  $d$  is the Sun-Earth distance at some point of the orbit extracted from a dataset.

Knowing the apparent radius for each day, it is possible to build a dynamic template of the disk, and template matching can be applied to find the center of the Sun in pixel coordinates. Also, the radius can be used to determine the size of the patch to be cropped in order to obtain an image in which the Sun is centered. The perfect alignment is achieved by rotating the resulting image by an angle that is equal to the axial tilt.

## C.2 Feature Enhancement

### C.2.1 White-light

For the white-light pipeline the feature enhancement phase starts with brightness correction. The brightness of the image depends on many factors, such as seeing, clouds, exposure time and gain. These factors also affect the depth of the intensity gap of sunspots. We desire to be able to account for different conditions and correct the image accordingly, while also trying to enhance sunspots.

An easy solution to this problem is to use the mean intensity of the disk to adjust brightness and the standard deviation to make the depth of sunspots roughly constant. In practice, the mean is subtracted from all the pixels of the disk, then the pixel counts are scaled to achieve the desired standard deviation value and finally the new mean is added to the pixels.

Depending on the purpose of the processing, whether scientific or visual, the limb darkening effect can be corrected or not. The procedure is the same as explained in section 7.2. Limb darkening corrections is important, in some cases, because it helps to identify those sunspots that are very close to the limb of the Sun.

The processing technique that most enhances the transitions between quiet disk background, penumbra and umbra is sharpening. Since in a scientific setting it is best to retain as much information as possible, sharpening techniques that involve lossy operations like blurring were discarded. The most satisfactory performances were achieved, using second order derivative sharpening.

The results of the processing can be appreciated in Figure D.1.

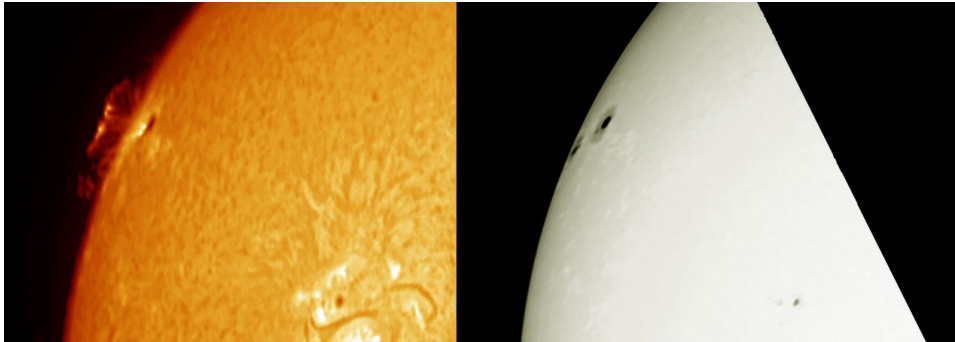


Figure C.2: Example images taken from the Helios observatory. H-alpha (right), visible (left)

### C.2.2 H-alpha

For what concerns the H-alpha telescope, this processing phase is very interesting, because it allows to uncover features that are not visible from the raw images. In fact, our H-alpha pipeline mainly focuses on bringing out those low intensity emissions that come from the atmosphere of the Sun. There are two phenomena that were analyzed: **prominences** and **filaments**.

Both prominences and filaments are large, bright feature extending outward from the Sun's surface. They are anchored to the Sun's surface in the photosphere, but their gaseous arms reach the corona. The only difference between them is the relative position with respect to the observer. Prominences extend orthogonally to the line of sight of the observer and cross the limb of the Sun outwards. This means that they are visible as bright bands emitting on a the black background (deep space). Filaments, instead, lay inbetween the observer and the Sun, therefore they appear as dark bands on the surrounding disk.

The process of revealing protuberances from raw images is called off-limb emission enhancement. At the Helios observatory two such techniques were explored:

- **thresholding** and **stretching**: the background is first cleaned from noise with a threshold and then the histograms stretched to increase contrast;
- **double exposure superposition**: two subsequent images with different exposure time are taken and then merged together. Off-limb

pixel values are taken from the image with longer exposure time, while the disk is copied from the other one. Even though the time difference between the two shots is minimal, it is usually necessary to align them before performing the merger.

Filaments, instead, are treated similarly to sunspots, since they both lay on the disk. The only processing that can be performed in this case is sharpening. Both gradient and laplacian sharpening have been applied for filament enhancement, with similar performance.

## Appendix D

# Master Control Program

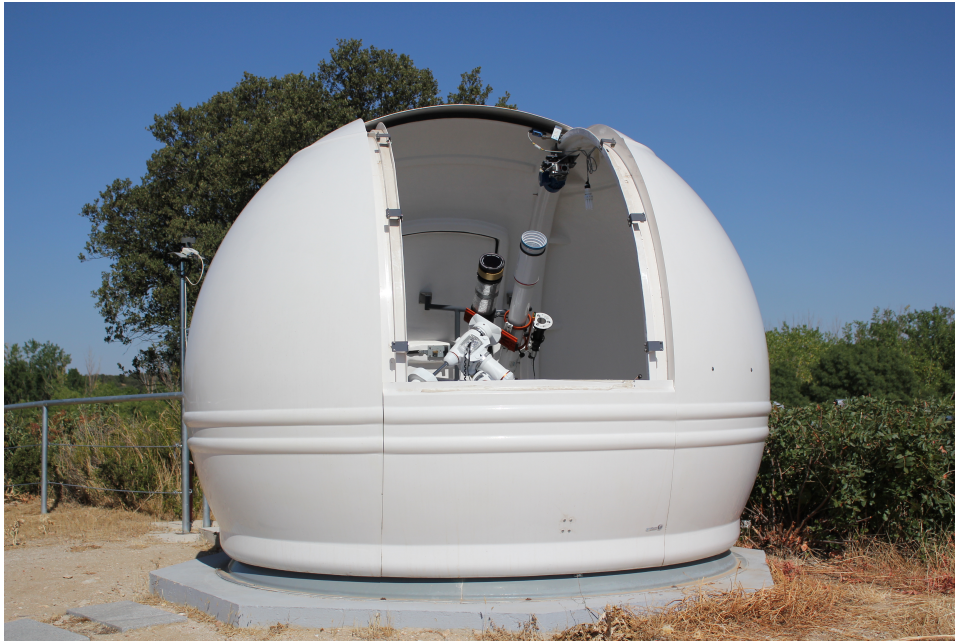
Apart from the two telescopes described in Appendix C, the Helios solar observatory is composed by a dome, a mount, a weather station and two camera sensors (one for each telescope). Here are the specifications:

- a Scopedome 3M dome;
- a Celestron CGEM mount;
- a AAG CloudWatcher weather station.
- two 2 QHY5-II-M planetary cameras with a 1/2 CMOS sensor;

All these components come with dedicated proprietary software programs to control them. The problem with these programs is that, apart from some predefined functionality, they are not able to work together. Therefore, in order to fully automate the observatory, it was necessary to build a dedicated control program.

All the hardware was selected according to its capability of being programmed. The protocol that was used in order for the master control program to interface with the hardware is called ASCOM (Astronomy Common Object Model). ASCOM establishes a set of vendor, language and platform independent interface standards for drivers that provide plug-and-play control of astronomical instruments and related devices.

The master control program is a very large project, that I started during 2018 and is still ongoing. During my internship I was able to program the connection to the components and a part of the user interface that lets the operator control the observatory manually. For what concerns the automation of the daily observation process, I focused on the improvement of the tracking of Sun.



*Figure D.1: An image of the Helios observatory performing daily observation*

Some tracking capabilities were already offered by the Celestron mount. In fact, once the telescope has slewed to some coordinates, it is able to follow that particular point in the sky, regardless of the rotation of the Earth. This functionality is called tracking, and it is included in any modern mount. The telescope moves thanks to a set of motors and gears that are embedded inside the robotic arm. Like any mechanism, it has imperfections and tolerances, that make the accuracy of the tracking is bounded.

The limitations of the mechanical tracking resulted in misalignments in the images, and in some cases in the Sun ending out of the field of view. To address this problem software can come to the aid of hardware. In fact, the two camera sensors stream images in real time from the scope to the control program. As shown in Appendix C, it is fairly easy to find the coordinates of the center of the disk. Hence, it is also possible to calculate the correction that should be applied by the mount to place it back to the center of the frame. In fact, using the same reasoning as in (C.2), we can compute how many radians correspond to each pixel of the image and, after some coordinate translation, instruct the telescope to slew back to the Sun.

The development of the master control program for is being continued by the interns that followed me at the Helios observatory.