



*Facultad
de
Ciencias*

A Deep Learning approach for the detection of point sources in the Cosmic Microwave Background

Trabajo de fin de Grado
para acceder al

GRADO EN FÍSICA

Author: David Balbás Gutiérrez
Director: Patricio Vielva Martínez
Co-Director: Biuse Casaponsa Galí

September - 2019

Contents

Acknowledgements	III
Abstract	V
Foreword	VII
1 Point sources in the Cosmic Microwave Background	1
1.1 The Cosmic Microwave Background	1
1.1.1 Anisotropies and power spectrum	2
1.1.2 Galactic foregrounds	3
1.1.3 Point sources	4
1.2 State of the art	5
1.2.1 The Planck Sky at different frequencies	5
1.2.2 Detection of point sources	6
2 Methodology	8
2.1 Simulations	8
2.1.1 Simulation of the CMB	9
2.1.2 Addition of point sources, foregrounds and noise	10
2.1.3 Projections onto the plane	11
2.2 Convolutional Neural Networks	12
2.2.1 Foundations	12
2.2.2 Architectures	16
2.3 Matched Filter	18
2.3.1 Sphere implementation	19
2.3.2 Local implementation	19
2.4 Object extraction	19
2.4.1 Detection criteria	20
2.4.2 Method evaluation	20
3 Detection in CMB and Gaussian noise simulations	22
3.1 Method characteristics	22
3.1.1 Initial tests	22
3.1.2 Simulations	23
3.1.3 CNN	23
3.1.4 Matched Filter	26
3.2 Method evaluation and comparison	26
3.2.1 CNN and spherical MF	27
3.2.2 CNN and real space MF	28
3.3 Discussion	29

4	Detection in realistic scenarios	33
4.1	Performance of previous methods	33
4.2	CNN characteristics and evaluation	35
4.2.1	Performance of the CNN at different training levels	37
4.3	Division of the sky in two regions	38
4.4	Application to the Planck data	42
4.5	Discussion	43
5	Conclusions and future work	46
	Bibliography	49

Acknowledgements

Este trabajo no solo representa para mí el final de mis estudios de Grado. Con el paso de los meses ha adquirido una fuerte carga personal, mayoritariamente por todas las pequeñas estrellas que han estado, de una forma u otra, involucradas en él.

En primer lugar, les estoy profundamente agradecido a los dos directores de este proyecto, Patricio y Biuse. Gracias por la inestimable experiencia, sabiduría y dedicación que habéis aportado, y también por vuestra paciencia y flexibilidad conmigo. Nada parecido a este trabajo hubiera sido posible sin vosotros. Además, quiero agradecerles a todas las personas del grupo de Cosmología del IFCA su hospitalidad y la posibilidad de trabajar con ellos. Particularmente a Diego, por su aporte al trabajo con el código del *matched filter* y por haberme contagiado el “gusanillo cosmológico” en sus clases.

En un principio, el trabajo pretendía ser una continuación de mi estancia en la Universidad de Brown. Aunque finalmente no fue así, le agradezco a mi tutor Savvas la oportunidad de involucrarme en su proyecto de investigación. Más importante aún ha sido mi pequeña familia americana: Elena, Petri y Jorge. A ellas dos, junto con Alberto, también les agradezco su ayuda en muchos momentos y el buen ambiente que ha desbordado siempre en nuestro rincón.

Por otro lado, le estoy agradecido a la Facultad de Ciencias por el Doble Grado en sí y por su flexibilidad, especialmente a Francisco y a Chema. También a la Universidad de Bergen y a quienes me acompañan desde entonces, entre los que destaco a Sara, Alejandro, Falk y Janina. Finalmente, a aquellos profesores, no únicamente universitarios, que han sabido convertir la ciencia en un arte y transmitirme su pasión por ella.

Por último, y aunque podría alargarme mucho más, reitero los agradecimientos de otro pequeño trabajo¹ que deposité dos meses atrás. En especial, a mis once compañeros de Doble Grado, a mi familia y a mi infatigable grupo de amigos por su apoyo incondicional en todo momento. También a las personas que me acompañan en mi nueva aventura escandinava; os prometo que a partir de este martes nada me frenará para darlo todo con vosotros.

¹BALBÁS GUTIÉRREZ, D. *Ataques al Criptosistema RSA*. Trabajo de Fin de Grado, Universidad de Cantabria, 2019.

Abstract / Resumen

The detection and subtraction of the point sources in the microwave sky has important implications in cosmology, since it allows to perform an accurate characterisation of the Cosmic Microwave Background and, thus, a measurement of the cosmological parameters. In this work, we present a new approach to the problem based on Deep Learning, where various convolutional neural networks are trained and tested on different scenarios. Besides, their performance is compared among them and to a two-dimensional matched filter.

Our models achieve a better result than the matched filter in the case of Gaussian noise and CMB simulations. Furthermore, they present a remarkable performance when applied to realistic maps which include Galactic foregrounds at the 143 GHz Planck channel, particularly in regions far from the Galactic plane. Finally, we test our method on real data from the Planck experiment, obtaining promising results.

Key words: CMB, point source detection, cosmology, Deep Learning, Convolutional Neural Networks.

Detectar y sustraer las fuentes puntuales en el cielo de microondas tiene importantes implicaciones en cosmología, ya que permite realizar una caracterización precisa del Fondo Cósmico de Microondas y, por tanto, una medición de los parámetros cosmológicos. En este trabajo, presentamos un nuevo método para la detección de fuentes puntuales basado en aprendizaje profundo, donde varias redes neuronales convolucionales son entrenadas y probadas en distintos escenarios. Además, su rendimiento es comparado entre sí y con un *matched filter* bidimensional.

Nuestros modelos consiguen un mejor resultado que el *matched filter* en el caso de simulaciones de ruido Gaussiano y de fondo cósmico. Por otro lado, presentan un rendimiento destacable cuando se aplican a mapas realistas que incluyen foregrounds Galácticos en el canal de 143 GHz de Planck, particularmente en regiones lejanas al plano Galáctico. Finalmente, nuestro método se prueba en datos reales del experimento Planck, obteniendo resultados prometedores.

Palabras clave: FCM, detección de fuentes puntuales, cosmología, aprendizaje profundo, redes neuronales convolucionales.

Foreword

This thesis is the result of a project which started in September 2018, when I was offered with the opportunity to carry out my *Trabajo de Fin de Grado* within the *Observational Cosmology and Instrumentation Group* at IFCA. The project, directed by Dr Patricio Vielva and Dr Biuse Casaponsa, has been partially supported by a *Beca de Colaboración del Ministerio de Educación, Cultura y Deporte* in the *Departamento de Física Moderna* at *Universidad de Cantabria*.

Our work aims to study the feasibility of a Deep Learning approach to detect point sources embedded in microwave sky maps. Throughout the project, several methodologies have been tried; from alternative map generations and projections to different network architectures and matched filter implementations, until accurate methods were found. The text focuses on the characteristics of the final simulations, the algorithms which build the detection pipeline, and particularly in the discussion of the methods and the results. We refer to some of the initial attempts when these provide an interesting insight into the discussion.

In Chapter 1, we introduce the fundamentals of the Cosmic Microwave Background and its main contaminants: the diffuse foregrounds from our Galaxy and the point sources. Besides, we survey the state of the art in the point source detection and characterisation, mainly the publications and catalogues of compact sources derived from the Planck Mission.

In Chapter 2, we explain the methodology and algorithmic foundations of our work. We describe the data simulation and generation, the Convolutional Neural Networks and their architecture, the matched filter, and the object detection algorithm and criteria.

In Chapter 3, we present the results of our method applied to simulated maps containing CMB, Gaussian noise and point sources. We mainly compare it to the matched filter, concluding that it exhibits outstanding performance, particularly in the low flux source range.

In Chapter 4, we apply our method to realistic simulations, including Galactic foregrounds, at a frequency of 143 GHz. The performance of the CNN at different levels of training is discussed and the sky is divided into regions, where we test different algorithms. Finally, we present a preliminary application of our methods to data from the Planck experiment, obtaining similar results to current detection methods with promising prospects.

The conclusions of our work are presented in Chapter 5. We also include a bundle of future directions in which we aim to continue our research. The main code can be found at <https://github.com/DavidBalbas/DeepSources>, including the necessary documentation for its usage.

Chapter 1

Point sources in the Cosmic Microwave Background

1.1 The Cosmic Microwave Background

Approximately 14 billion years ago, the Universe was a hot and dense clump of radiation and matter expanding vigorously. When it was 375 000 years old, the temperature was cold enough for electrons and protons to recombine into the first atoms. The photons were then no longer scattered by charged particles and freely continued their journey until reaching us today. The result is that the observation of the universe in the microwave range corresponds to an almost perfect blackbody radiation whose spectrum is characterised by a temperature of $T_0 = 2.725 \pm 0.001$ K [25].

Nowadays, the Cosmic Microwave Background is one of the main shreds of evidence supporting the (Λ CDM) standard cosmological model. The model is mainly built upon the geometrical description of Einstein's general relativity, the theory of the Big-Bang and the cosmological principle, this is, that the universe is homogeneous and isotropic at large scales and lacks of any privileged observer. In addition, it provides a strong argument in favour of cosmic inflation.

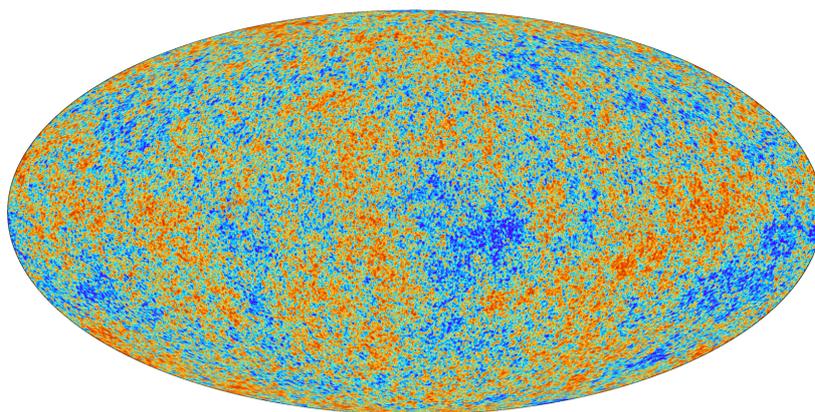


Figure 1.1: All-sky map of the CMB temperature anisotropies. The map represents the final data of the Planck mission [25].

Since the discovery of the CMB by Penzias and Wilson in 1964, cosmologists have strived for performing a precise characterization of this radiation. Even though the CMB is highly homogeneous, anisotropies of the order of 10^{-5} have been detected at different scales. Once

the dipole caused by the relative motion of the Sun is compensated, these anisotropies are the result of a combination of the initial inhomogeneities of the universe at the recombination time and several astrophysical and cosmological effects (see [7] or [11] for further details on CMB anisotropies). Besides a global picture from the earliest stages of our universe, a precise measurement of the CMB anisotropies provides with an accurate estimation of many cosmological parameters. For instance, the matter or radiation densities, the Hubble constant and the age of the universe [25]. Several missions and experiments in the last decades, such as COBE¹ (1989), WMAP² (2001) and PLANCK (2009, [25]), have measured the CMB with increasing precision, providing us with a deeper understanding of our universe.

1.1.1 Anisotropies and power spectrum

The temperature anisotropies of the CMB can be studied as angular fluctuations in the intensity of the CMB radiation from its average blackbody temperature $T = T_0$. To deal with these fluctuations in a statistical way, it is convenient to think of these anisotropies, seen from a certain point in a given time, as a scalar field $\frac{\Delta T}{T} : \mathbb{S}^2 \rightarrow \mathbb{R}$ defined on the sphere. This field can be developed in terms of spherical harmonics [7]:

$$\frac{\Delta T}{T}(\mathbf{n}) = \sum_{\ell=2}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} a_{\ell m} Y_{\ell m}(\mathbf{n}), \quad (1.1)$$

$$a_{\ell m} = \int_{\mathbb{S}^2} Y_{\ell m}^*(\mathbf{n}) \frac{\Delta T}{T}(\mathbf{n}) d\Omega, \quad (1.2)$$

where $\mathbf{n} \in \mathbb{S}^2$, the $Y_{\ell m}$ are the spherical harmonics, the $a_{\ell m}$ their coefficients, and Ω represents a solid angle. The parameter ℓ corresponds to a certain angular distance $\theta \approx \pi \ell^{-1}$. Thus, large values of ℓ represent great angular scales whereas low values of ℓ correspond to small angles. The index m represents the spatial orientation of the fluctuations.

As the universe does not present any privileged direction, the process generating the initial perturbations is statistically isotropic. This implies that there is no correlations among perturbations at a different m or ℓ , and it can be proven that the expectation value of the variance of two coefficients is $\langle a_{\ell m} a_{\ell' m'} \rangle = \delta_{\ell \ell'} \delta_{m m'} C_{\ell}$. The coefficients C_{ℓ} are the CMB angular power spectrum. The accuracy in the determination of the power spectrum is limited due to the cosmic variance, i.e., the uncertainty intrinsic to having a single observable realisation of the temperature field [17]. The average over all the m coefficients is the maximum likelihood estimator of C_{ℓ} for a certain ℓ ,

$$\hat{C}_{\ell} = \frac{1}{2\ell + 1} \sum_{m=-\ell}^{\ell} |a_{\ell m}|^2. \quad (1.3)$$

Assuming initial Gaussian fluctuations, as there are $(2\ell + 1)$ coefficients for each ℓ , the cosmic variance will be the dispersion of a χ^2 distribution with $2\ell + 1$ degrees of freedom,

$$\Delta \hat{C}_{\ell} = \sqrt{\frac{2}{2\ell + 1}} C_{\ell}. \quad (1.4)$$

The C_{ℓ} coefficients contain all the required information for the study of the temperature anisotropies. However, this intrinsic error source must be carefully taken into account and it is particularly notorious at small values of ℓ .

¹<https://lambda.gsfc.nasa.gov/product/cobe/>

²<https://map.gsfc.nasa.gov/>

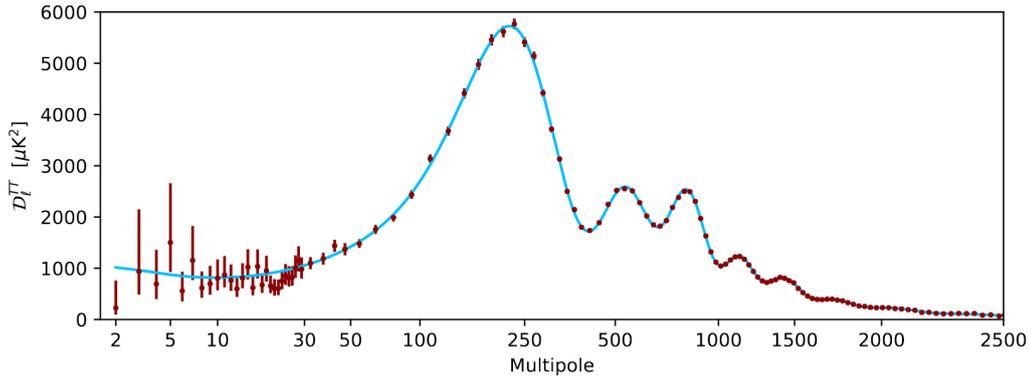


Figure 1.2: CMB power spectrum as observed by Planck. The $D_\ell = C_\ell \ell(\ell + 1)/2\pi$ coefficients are shown in the vertical axis, with respect to the multipole ℓ . The blue line is a best-fit model. Source: <https://www.cosmos.esa.int/web/planck/picture-gallery>

1.1.2 Galactic foregrounds

The CMB is not the only source of radiation at the microwave frequencies in our Universe, so not only photons from the CMB are collected when observing the sky. A complex processing pipeline is required to obtain a clean map of the background, which requires a deep understanding of the sources of contamination, known as foregrounds. There are two main types of foregrounds: Galactic, the ensemble of emissions from the Galaxy which mainly pollute the Galactic plane (Figure 1.3); and extragalactic, which appear roughly homogeneously across the sky. The main Galactic contaminants are the following [13]:

- **Synchrotron emission**, which is the radiation emitted by relativistic charged particles accelerated in magnetic fields. This radiation dominates at low frequencies, below 30 GHz.
- **Free-free emission**, produced by charged particles losing kinetic energy in electric fields produced by other particles. It dominates in a small range of frequencies around 50 GHz.
- **Spinning dust**, probably generated by rotational electric dipole emission. This pollutant is only relevant in the low-frequency range, below 40 GHz and, probably, in unlocalised regions of the Galaxy.
- **Thermal dust**, emitted by small grains of solid materials which behave as graybodies. This radiation clearly dominates above 90-100 GHz.
- **Molecular emission lines**, which is the spectrum of transitions in the energetic levels of molecules, such as CO, and appears only at very defined frequencies.

Looking to the sky far from the Milky Way, near the Galactic poles, seems the most obvious solution to avoid the above pollutants and precisely detect the CMB anisotropies. However, the large scale information would be compromised. Moreover, at small scales, the sky is full of bright objects of different nature whose angular size is smaller than the resolution of our experiments, thus resembling bright spots in our images. These objects, called point sources, highly affect the measurements of the CMB and are hard to deal with. Hence, they must be detected and masked or subtracted if a proper characterisation of the CMB anisotropies at small scales is to be obtained.

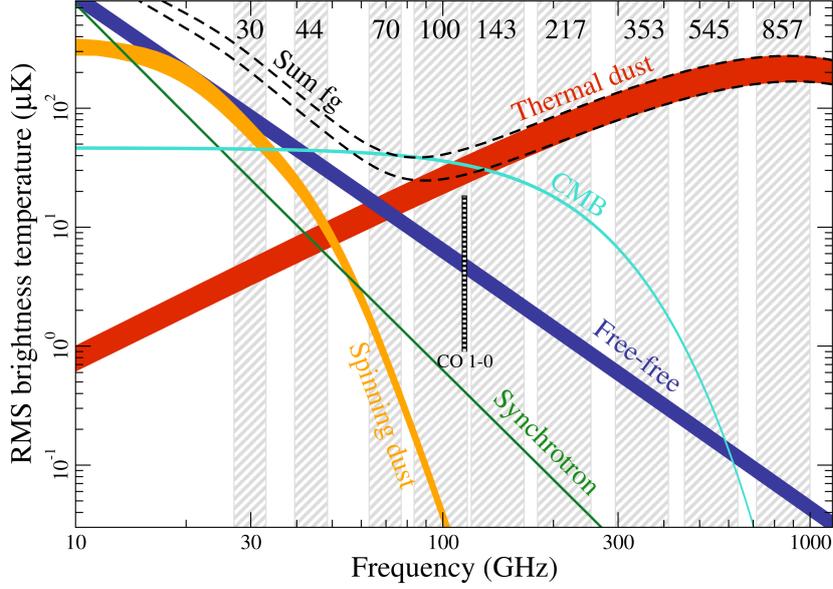


Figure 1.3: Intensity of the main Galactic foregrounds of the CMB at different frequency ranges. The shaded regions represent the frequency ranges observed by Planck. Source: <https://www.cosmos.esa.int/web/planck/picture-gallery>.

1.1.3 Point sources

The best way of considering a point source f in a temperature field is as a Dirac delta function $f(\mathbf{n}) = T_{ps}\delta(\mathbf{n})$, which is smoothed in the observation maps due to the finite resolution of the experiments. Normally, this is given as Gaussian point spread functions (PSF). Therefore, point sources are seen as small two-dimensional Gaussian peaks. Point sources are physical objects of different nature, typically galaxies that may be dominated from different physical processes than ours, so their emissions present a different frequency behaviour.

When one studies point sources at a certain frequency, the most relevant parameter is their flux S , which is linearly related to the variation in temperature we observe. The relation between the variation in intensity ΔI_{cmb} and in temperature ΔT_{cmb} is given by (see [10]):

$$\Delta I_{cmb}(\mathbf{n}, \nu) \approx \left. \frac{\partial B(\nu, T)}{\partial T} \right|_{T=T_0} \Delta T_{cmb}(\mathbf{n}), \quad (1.5)$$

where T_0 is the mean CMB temperature and $B(\nu, T)$ is the Planck Function. An accurate approximation of the conversion factor is

$$\left. \frac{\partial B(\nu, T)}{\partial T} \right|_{T=T_0} \approx 24.8 \left[\frac{x^2}{\sinh(x/2)} \right]^2 \text{ Jy sr}^{-1} \mu\text{K}^{-1}. \quad (1.6)$$

Here, $x \approx \nu/56.8$ GHz. To obtain the flux in Jy ($1 \text{ Jy} = 10^{-26} \text{ W m}^{-2} \text{ Hz}^{-1}$), one should integrate ΔI_{cmb} over the solid angle covered by the PSF.

Calculation of the flux of a point source from an image

As we mentioned above, point sources acquire (ideally) a Gaussian shape when appearing in images whose width is determined by the resolution of the instrument. More specifically, if an experiment presents a Gaussian beam characterised by its full width at half maximum

(FWHM), the FWHM of the instrument will correspond to the FWHM of the Gaussian-shaped point source. The conversion factor between the FWHM and the standard deviation σ of the Gaussian is $\text{FWHM} = 2\sqrt{2 \ln 2} \sigma \approx 2.355 \sigma$.

Let Ω be the angular resolution of the image pixels, σ_{pix} the standard deviation of the beam (both in pixels) and I_{max} the intensity of the source at the central pixel. We shall integrate the intensity of the point source in a unique pixel, obtaining a one-pixel intensity I_p ,

$$I_p = \int_{\mathbb{R}^2} I_{\text{max}} e^{\frac{-(x^2+y^2)}{2\sigma_{\text{pix}}^2}} dx dy = 2\pi\sigma_{\text{pix}}^2 I_{\text{max}}. \quad (1.7)$$

The flux S is the result of multiplying the one-pixel intensity by the angular resolution, $S = I_p \Omega$. The magnitude I_p (and its analogous in temperature, T_p) will acquire importance when carrying out simulations of the sky.

The distribution of the point sources in the sky has been studied in depth in the last years. Consequently, accurate modelling and simulation of point sources can be performed, such as the simulations produced by the Planck Sky Model [5] used in this work.

1.2 State of the art

As of 2019, the most accurate maps from the microwave sky and therefore from the CMB come from the ESA's Planck spacecraft. This satellite has scanned the microwave and submillimetre sky between 2009 and 2013, producing "*high resolution, all-sky maps in nine frequency bands from 30 to 875 GHz*" [25]. Planck has enabled the measurement of several cosmological parameters with high accuracy and a deeper understanding of the CMB, the Galactic foregrounds and the point sources, as compared to previous data from missions such as WMAP, COBE and ground-based experiments.

1.2.1 The Planck Sky at different frequencies

The aspect of the sky at each one of the Planck channels is completely different, as shown in Figure 1.4. The main reason is that different foregrounds dominate at each frequency, as we saw in Figure 1.3. Each channel has its own peculiarities, such as a different FWHM of the beam (resolution of the detector, Table 1.1), and the relative intensities of noise and foregrounds.

Channel (GHz)	30	44	70	100	143	217	353	545	857
Fitted FWHM (arcmin)	32.29	27.00	13.21	9.66	7.22	4.90	4.92	4.68	4.22
Instrument	LFI	LFI	LFI	HFI	HFI	HFI	HFI	HFI	HFI
Temp. sensitivity ($\mu\text{K deg}$)	2.5	2.7	3.5	1.29	0.55	0.78	2.56	0.78*	0.72*

Table 1.1: FWHM of the beam, detector and instrumental noise of each Planck channel. LFI/HFI are the acronyms for Low/High Frequency Instrument. The temperature sensitivity is the noise estimate scaled to 1° , assuming white noise. *Values in $\text{kJy sr}^{-1} \text{deg}$. Source: [24].

The 143 GHz channel

In regard to the characteristics of the different Planck channels, we carry out our study in the 143 GHz band. The main reason is that we aim to test the performance of a new method for detecting point sources, and a good starting point is to do so under the better possible conditions, where we know that current detection methods work accurately. According to

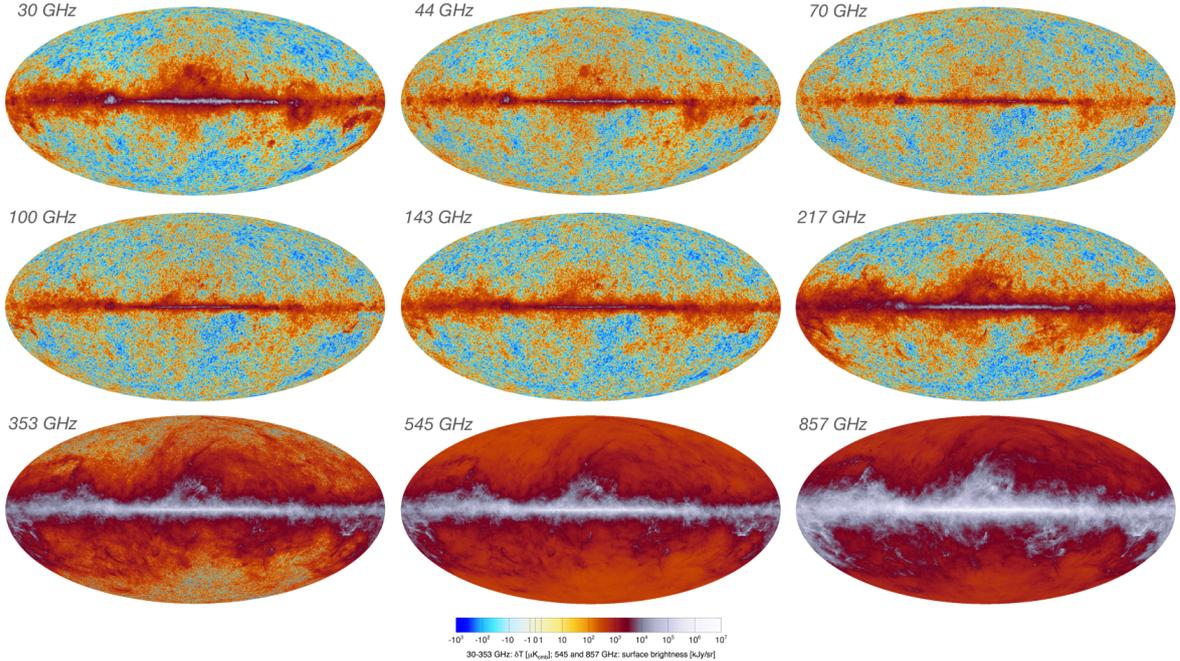


Figure 1.4: The microwave sky at the nine frequency channels, as observed by the Planck Mission at its final release in 2018. Source: <https://www.cosmos.esa.int/web/planck/picture-gallery>.

the Table 1.1 and the Figure 1.4, we find an optimal compromise among foreground intensity, detector resolution and noise sensitivity at this bandwidth. Lower frequencies imply lower resolutions whereas at higher frequencies the foreground intensity increases dramatically.

1.2.2 Detection of point sources

The most recent and complete work in point source detection in the microwave sky is the Planck Catalogue of Compact Sources (PCCS) [23]. The catalogue consists in a list of both Galactic and extragalactic compact sources detected all over the sky at each of the Planck frequency channels. A map including all the sources of the catalogue is shown in Figure 1.5.

There are three critical parameters aiming to evaluate the quality of a catalogue: the number of detected sources (N_{total}), the reliability (R) and the completeness curve ($C(S)$). The reliability is defined as $R = 1 - N_{\text{spur}}/N_{\text{total}}$, where N_{spur} is the number of spurious detections (i.e., the number of detections mistakenly catalogued as point sources). The completeness at a certain flux S is defined as the number of detected sources above the flux S over the total number of sources above that flux. High levels of reliability and completeness at low fluxes are difficult to achieve simultaneously; if we try to detect very faint sources it is likely to include statistical fluctuations from the background in our catalogue, whereas if we only trust bright spots we would miss most of the faint ones. In the PCCS, the reliability is set to 80%, obtaining a total of 5625 compact sources in the 143 GHz channel.

A further refinement of this work is the Second Planck Catalogue of Compact Sources [24], whose methodology is slightly different. The sky is now divided in two regions where detections are performed separately: an extragalactic area, and a strongly polluted Galactic area. The result is two complementary catalogues. First, the PCCS2 which includes the extragalactic area and therefore more reliable estimations of the source flux. Second, the PCCS2E, which encompasses the sources near the Galactic plane. Moreover, these catalogues contain information of the polarization of the sources. Their characteristics for the 143 GHz

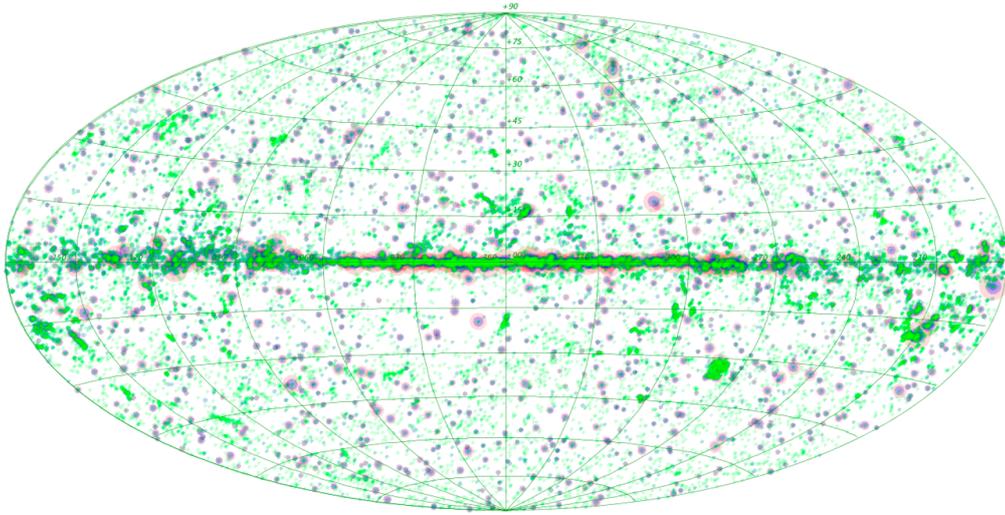


Figure 1.5: Distribution of the PCCS sources at three Planck channels. 30 GHz (pink), 143 GHz (magenta), and 857 GHz (green). Source: [23].

channel are shown in Table 1.2.

Catalogue	N_{total}	Covered sky	S_{90} / Jy
PCCS	5675	100%	0.190
PCCS2 (extragalactic)	2160	85.0%	0.177
PCCS2E (Galactic)	4139	15.0%	-

Table 1.2: Characteristics of the PCCS, PCCS2 and PCCS2E catalogues in the 143 GHz channel. S_{90} is the smallest flux S such that the completeness $C(S)$ is over 90% [23], [24].

The standard method for enhancing (and therefore detecting) point sources over a CMB, noise and foreground background is the application of tailored filters, among which the so-called matched filter is the optimal. This method, detailed in the next chapter, basically consists in performing a convolution, either in real or Fourier space, of the map and the “expected shape” of a point source. However, a major drawback is that it requires an accurate knowledge of the power spectrum of the map. In addition some technical difficulties normally arise when it is applied to real data [23]. The algorithm used for the construction of the catalogues is the Mexican Hat Wavelet 2 (MHW2) [16], which follows a similar procedure but is less sensitive to the issues that affect the matched filter.

We summarise that the detection of point sources and the estimation of their flux are two challenges of great relevance in modern cosmology. This work focuses on the first part, suggesting an alternative, deep learning based approach in contrast to the current methods and comparing it to a matched filter implementation. As we mentioned above, the study is carried out in the 143 GHz Planck channel, although a similar procedure may be employed to study different frequencies, as we discuss in our conclusions.

Chapter 2

Methodology

In this chapter, we present and describe the algorithms that we have developed for the detection of point sources in maps of the microwave sky. The first section deals with the simulation of the sky and its projection onto the plane, so that our detection methods can be fed with plane images. The second one provides the foundations of convolutional neural networks, in addition to the differences on their architecture depending on their purpose. The third section presents the matched filter and its implementation, required for comparison purposes. The fourth covers the methods for detection of compact objects in images, used after the maps have been processed by a filter or a neural network.

The combination of all those steps builds a processing pipeline allowing the analysis and the comparison of the performance of the different methods. All the computations have been performed using the Python programming language (Python Software Foundation, <https://www.python.org/>). Summarising, the underlying philosophy of our work is simple: generate realistic simulations of the microwave sky with known point sources, train a convolutional neural network (or produce an appropriate filter) for detecting them, and evaluate the quality of its predictions.

2.1 Simulations

The standard way of representing the microwave sky is using a spherical map. A map consists of a full pixelisation of the sphere, where each of the pixels has well defined coordinates (for example, Galactic latitude and longitude). In this work, in concordance to most of the work in cosmology nowadays, the HEALPix¹ pixelisation scheme has been used [9]. This scheme divides the sphere using a base-resolution of 12 quadrilateral pixels with equal area. The base pixels can be recursively divided into four subpixels until the desired resolution is achieved, as seen in Figure 2.1. The parameter N_{side} counts the number of divisions along the side of a base pixel and determines the resolution of a map. In addition, every pixel has a unique identifier, an integer in the range $[0, 12N_{\text{side}}^2 - 1]$ which we refer to as pixel number.

The resolution of all the maps we use throughout our work is $N_{\text{side}} = 2048$, corresponding to a total of $N_{\text{pix}} = 12 \times N_{\text{side}}^2 \approx 5 \times 10^7$ pixels. A simple calculation shows that the angular resolution of the map (the average side of a pixel) is 1.718 arcmin.

Other advantages of using HEALPix include native routines for transforming a map from and into spherical harmonics, projections, pixel localisation and calculations on the sphere. All the required functionalities are available in the Python library `healpy`².

¹<https://healpix.sourceforge.io>

²Read the documentation at <https://healpy.readthedocs.io/en/latest/index.html>

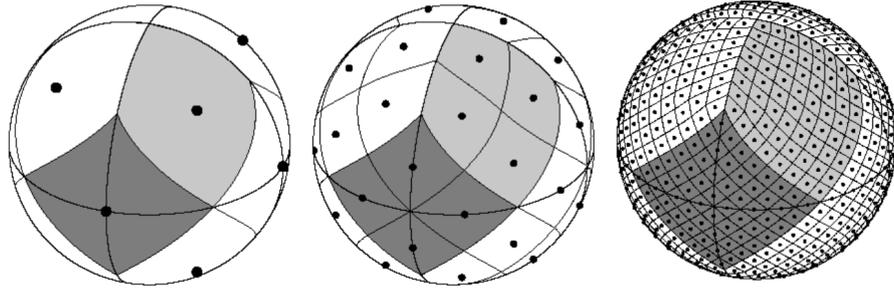


Figure 2.1: HEALPix pixelisation of the sphere at $N_{\text{side}} = 1, 2, 8$. Source: HEALPix documentation, <https://healpix.sourceforge.io/pdf/intro.pdf>.

As explained in the previous chapter, a microwave observation of the sky at a given frequency consists of several components. In simulations, these are produced independently and are later added to produce a realistic model of the sky. These components are the CMB, the Galactic foregrounds, the point sources and the instrumental noise, shown in Figure 2.2.

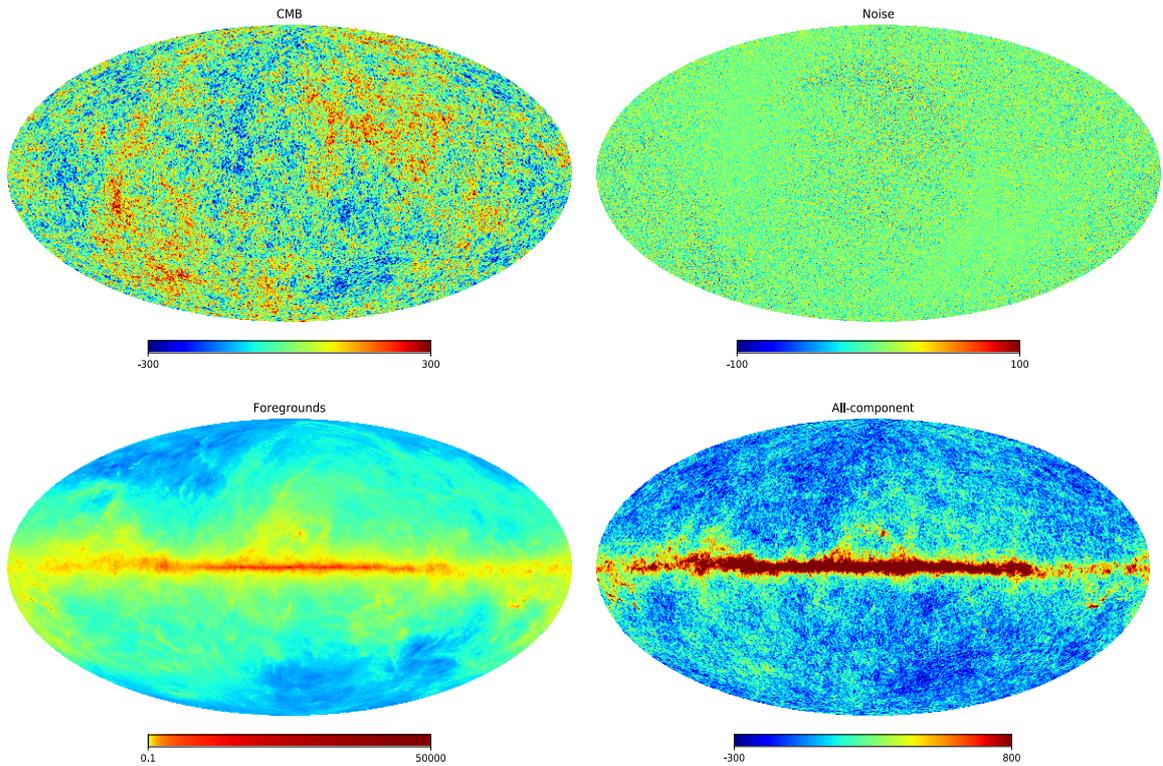


Figure 2.2: Mollweide projections of all-sky simulations of the CMB (top-left), instrumental noise (top-right), Galactic foregrounds (bottom-left, in logarithmic colour scale) and all the components adding point sources (bottom-right).

2.1.1 Simulation of the CMB

Obviously, the advantage of using simulations is that one can completely customise the maps, selecting only the desired components. For instance, a map with only point sources, CMB or foregrounds. In addition, simulations allow to work with many different realisations of the CMB, which is very useful for testing the statistical significance of the data.

In this work, we use the Planck best-fit Λ CMB power spectrum [25] to simulate the CMB. For this purpose, we apply the method `synfast` of the `healpy` library, which creates a map with random fluctuations based on the CMB angular power spectrum $(C_\ell)_2^{\ell_{max}}$ (see equation 1.3). In other words, it produces a set of spherical harmonics coefficients $a_{\ell m}$, $2 \leq \ell \leq \ell_{max}$, $-\ell \leq m \leq \ell$. The procedure is based in equation 1.3, which indicates that the power spectrum is an estimator of the variance of the $a_{\ell m}$ for a fixed ℓ . As there is no correlation between the $a_{\ell m}$, a realisation of the $a_{\ell m}$ can be simulated independently for each ℓ , as long as the property in equation 1.3 is maintained. Thus, we can randomly generate the $a_{\ell m}$ coefficients from a Gaussian distribution with zero mean and variance C_ℓ , for $2 \leq \ell \leq \ell_{max}$.

A brand new map is created in each simulation. Therefore, even if all the maps come from the same angular power spectrum, we work with a different realisation of the sky every time.

2.1.2 Addition of point sources, foregrounds and noise

We mainly deal with two kinds of simulations in our work: the first contains only CMB, point sources and instrumental noise, and the second also includes Galactic foregrounds. In the latter, we directly use the Planck Legacy Archive³ utility, and particularly the Planck Sky Model (PSM) [5], which provides us with foreground simulations that include synchrotron, free-free, spinning dust and thermal dust emissions. The foregrounds simulation is unique and independently generated.

The point sources are simulated separately of any existing facility. Every map contains between 75000 and 125000 point sources randomly distributed across the sky, i.e., we do not consider clustering of sources. We have extracted the distribution of the brightest point sources from point-source-only simulations of the PSM, shown in Figure 2.3, which we use to produce a realistic set of point sources.

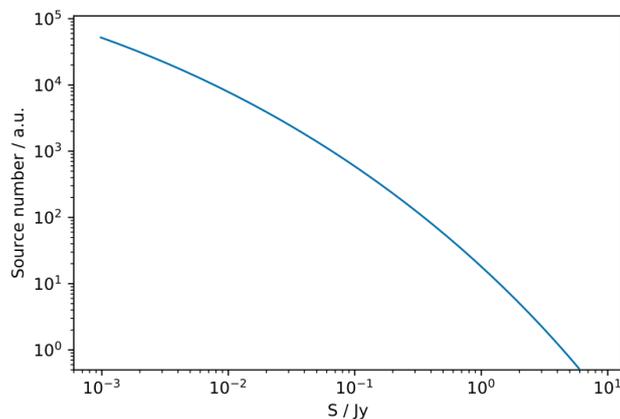


Figure 2.3: Distribution of bright ($S > 1$ mJy) point sources at each flux in the microwave sky. The data is extracted from PSM simulations of point sources only.

Only point sources with a flux $S > 2.97$ mJy (or equivalently, a one-pixel temperature $T_p > 31.6 \mu\text{K}$) are simulated, and there are also no sources above $T_p = 10$ K. These margins are set for practical reasons; sources below that flux are completely blinded by the background, basically impossible to detect with the resolution of our experiments, and the contamination

³Available at <http://pla.esac.esa.int/pla/#home>

they produce is negligible. Therefore, we do not aim to detect them with our methods. The upper limit is generous, simply to ensure that we have no sources above a certain level. If there were any, they would surely be detected by any of the methods we use, so they do not suppose a challenge.

Later, the point sources are smoothed using a Gaussian kernel filter and the map is saved separately. Having a map with only the point sources enables us to test the detections, as we explain in Section 2.4.

Finally, after smoothing the sum of the maps with a Gaussian kernel until the desired beam width is achieved (in this work, this will be 5 arcmin or 7.22 arcmin), we add a noise component to the previous map to simulate the actual Planck instrumental noise. This is carried out in one of the following two ways:

- Generate random and isotropic Gaussian maps, with average zero and standard deviation $\sigma = 19.4 \mu\text{K}$. This value is obtained from the Planck instrumental noise in Table 1.1, dividing by the pixel resolution 1.718 arcmin.
- Use realistic noise simulations from the PSM, which are neither totally isotropic nor perfectly Gaussian.

2.1.3 Projections onto the plane

Even though previous work has been done to translate convolutional neural networks into the sphere, and even into HEALPix pixelisations (see DeepSphere [20]), the size of our maps and the aim of our neural networks led us to project regions of the sky into 2D flat patches which are processed separately. In addition, current detection methods such as the matched filter and the MHW2 usually work on plane images too.

For this purpose, we divide the spherical map at $N_{\text{side}} = 2048$ into 3072 pixels, corresponding to a pixelisation at $N_{\text{side}} = 16$. Later, we obtain the coordinates of the central point of each of the pixels and we perform a gnomonic projection at a resolution of 1.7 arcmin, obtaining a 150×150 sized array. This consists in projecting the points in the sphere onto the tangent plane to the central point of the pixel, as seen in Figure 2.4. Our arrays are saved using the `.npy` format, which enables quick read and writing and while maintaining a small file size.

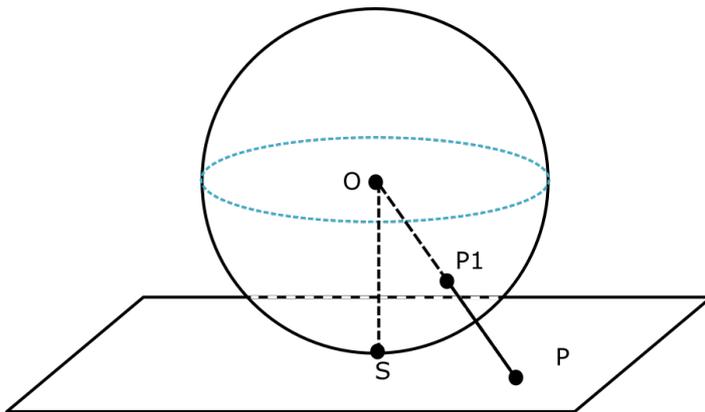


Figure 2.4: Gnomonic projection. The projection P of a point P_1 corresponds to the intersection with the tangent plane of the line passing through the centre of the sphere O and P_1 . Image from www.kisspng.com.

Originally, a pixel at $N_{\text{side}} = 16$ contains 128×128 pixels at $N_{\text{side}} = 2048$. The size of the projected image is slightly higher to ensure the sphere is fully covered by the 3072 projections,

and to enable us to discard the areas near the edges to avoid border effects (this fact will be discussed later). We prefer to have regions of the sky which may overlap in several images than lacking some parts of the map. This would involve missing point sources, deteriorating our results.

Finally, we require an algorithm to go back from the images to the sphere, this is, being able to transform the (x, y) pixel in the k image to its original point in the spherical map. This feature is especially relevant when working with real data, in order to remove duplicate detections and reconstruct the map with the location of the detected point sources. This is carried out in a very intuitive way. We simply create the projected images of a location map where the value of each pixel is exactly its pixel number. Then, the pixel number of the (x, y, k) pixel in the projected image is the value of the (x, y, k) pixel in the location image set.

One of the main advantages of using projections is that they permit a quicker data processing and the possibility of working in small portions of the sky. In addition, at some point in our work it will be useful to divide the sky in regions, which can be done easily by tracking the location of each of the patches. The principal inconvenient is the deformation suffered by the images, which in our case is minimised due to their small angular size (~ 4.3 deg each side).

2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are one of the most significant technological breakthroughs in the last decade despite being proposed 20 years ago by LeCun [14]. Since the mid-2010s, they have been responsible for major advances in computer vision, image recognition, natural language processing and several other fields of artificial intelligence. CNNs conform the core part of our work, as our aim is to compare their effectiveness for detecting point sources to current approaches.

For this work, we implement CNNs with the TensorFlow library for Deep Learning [1]. In addition, we make use of the Keras API [4].

2.2.1 Foundations

A CNN is a variation of a classic artificial neural network. The working procedure of the latter is fairly simple, so we will introduce it first. Globally, a neural network is a function $N : \mathcal{I} \rightarrow \mathcal{O}$, where \mathcal{I} is the input space and \mathcal{O} the output space, both of them real tensor (or array) spaces with fixed dimensionality. In classic neural networks, both the input and the output (also known as prediction) must be one-dimensional real arrays, but this restriction is no longer applicable to CNNs. The aim of neural networks is to classify, extract features or evaluate the input data. For instance, they can be trained to recognise handwritten digits or to distinguish whether an animal in a picture is a dog, a cat or a horse.

Neural networks are made of several layers, each composed by units named neurons. A typical structure of a neural network is seen in Figure 2.5.

The structure resembles a directed layered graph, where each neuron of layer i receives and processes the input of the neurons of the layer $i - 1$. Let $\mathbf{x} = (x_1, x_2, \dots, x_s)$ be the output vector of the neurons of the $i - 1$ -th layer. Then, the output y_k of the k -th neuron at the i -th layer is given by

$$y_k = f_i(\mathbf{x} \cdot \mathbf{w}_k + b_k) \quad (2.1)$$

where $\mathbf{w}_k \in \mathbb{R}^s$ is the weight vector, $b_k \in \mathbb{R}$ is the bias, and f_i is called the activation function. This function is usually non-linear. The most common activation functions, some of them

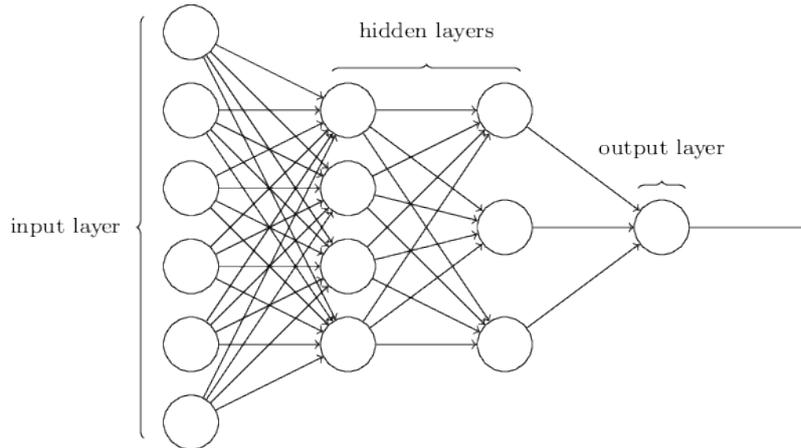


Figure 2.5: Typical structure of a simple feed-forward neural network. Source [19].

used in this work, are:

- The thresholding or **step** function. Given $t \in \mathbb{R}$, $f(x) = 0$ when $x < t$ and $f(x) = 1$ when $x \geq t$.
- The **ReLU** function, or Rectified Linear Unit. $f(x) = \max\{0, x\}$. This is probably the most used function nowadays in intermediate (hidden) layers [19].
- The **sigmoid** family of functions, which are bounded, monotonous and differentiable functions defined over the real numbers. An example is the logistic function $f(x) = 1/(1 + e^{-x})$, which is very useful for classification purposes.
- The **linear** function $f(x) = kx$, $k \in \mathbb{R}$, commonly used in the final layers for regression problems.

The main power of neural networks resides in their learning ability, this is, their capacity to adapt the weights of the neurons in order to achieve the desired output. Neural networks are a type of supervised learning algorithm, meaning that a labeled dataset is required for teaching the networks how to perform predictions.

Convolution layers

The only difference between neural networks and CNNs is the convolution operation. In a CNN, one or many neuron (dense) layers are replaced by convolution layers whose inputs and outputs are no longer restricted to be one-dimensional. In these layers, convolutions of filters (a weight tensor, also named kernel) are performed over the output of the previous layer. Each convolution layer contains a number k of filters, which result in k different outputs, also known as channels.

Convolutions may be carried out in any dimension, although the most common ones are the 1D, 2D and 3D convolutions. After a convolution, an activation function f is normally point-wise applied as in equation 2.1. We remark that dense 1D layers can also appear in CNNs (they commonly do so in classification models, for instance), but their input must be flattened previously. Next, we introduce the 2D convolution, which is the type used in our work.

Let $I \in \mathbb{R}^{n_x \times n_y}$ be an input array, and let $W_1, \dots, W_k \in \mathbb{R}^{m_x \times m_y}$, where $m_x < n_x, m_y < n_y$, be the filters of the 2D convolution layer (we assume that m_x and m_y are odd integers

for simplicity in our equations, although even values are perfectly valid). Let $s_x, s_y \in \mathbb{N}$ be the horizontal and vertical strides, this is, the spatial step between each convolution. Then, the output or feature map F of the layer will be a real tensor with dimensions $\lfloor n_x/s_x \rfloor \times \lfloor n_y/s_y \rfloor \times k$, given by

$$F_{a,b}^r = f \left(\sum_{i=0}^{m_x} \sum_{j=0}^{m_y} (W_r)_{i,j} \cdot I_{a \cdot s_x - (m_x - 1)/2 + i, b \cdot s_y - (m_y - 1)/2 + j} \right) \quad (2.2)$$

where a, b, r are the indices of an element in the tensor. It is easier to observe the tensor as a sequence of matrices of the same size, which are the outputs of each of the filters. Notice that the indices i, j may be negative or larger than the input size near the edges of the input. In this case, the convolution is carried out by padding with zeroes. If padding is eliminated, the products of the convolution will always have smaller dimensions than the input, but this is not convenient for our problem as we want to maintain the dimensionality.

Summarising, a convolution layer performs a scalar matrix product of the filter and a submatrix of the input, yielding the value of each of the points in the output. A visual representation of a simple convolution is found in Figure 2.6.

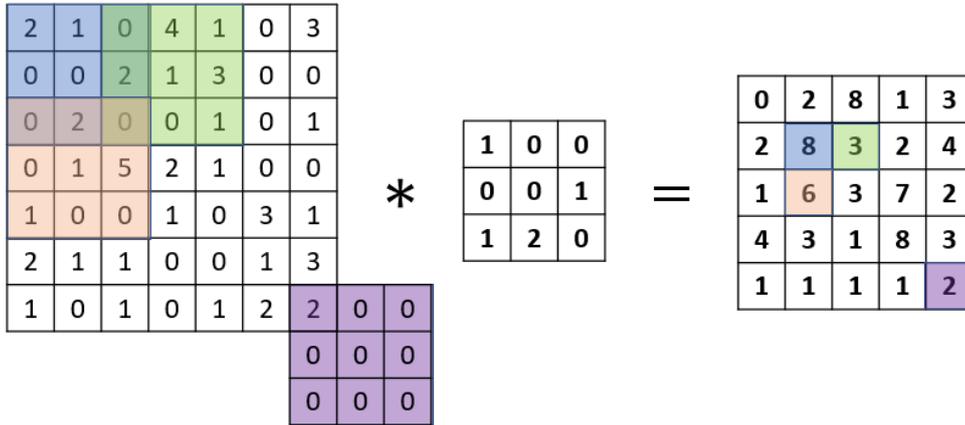


Figure 2.6: Representation of a one-channel 3×3 convolution with strides $s_x = s_y = 2$, on a 7×7 two-dimensional input, and the identity $f(x) = x$ activation function. Padding is applied to achieve a 5×5 output matrix.

The input for each of the filters of a convolution layer is also a weighted sum of the outputs of the previous layers, as in the case of the neurons. For this reason, it is important that all the filters of a convolution layer have the same size. The main benefit of convolution layers is their ability to detect both local and global features of the input, regardless of their location.

Training, loss and predictions

A neural network has to be taught how to make accurate predictions, as it is built of hundreds (or thousands) of parameters which are initially not suited for our data. The procedure to adapt the neural network to the data is called training. The model learns by updating the weights of the neurons towards decreasing a loss function, which basically represents the goodness-of-the-fit to the data. As the parameter space is very large, we also need large amounts of data to train the model. This data is called training set and is made of input-label pairs, where a label is the correct solution for an input.

A good example is our problem of point source detection. Here, our inputs are 2D real-valued arrays obtained as described in Section 2.1, which contain special areas we refer to as

point sources. The labels are a same-dimension 2D array of zeros and ones, where a region of ones indicate there is a point source in that position, and zeros represent the background. For every input, the model makes a prediction (its output). An example of input, label and prediction is shown in Figure 2.7.

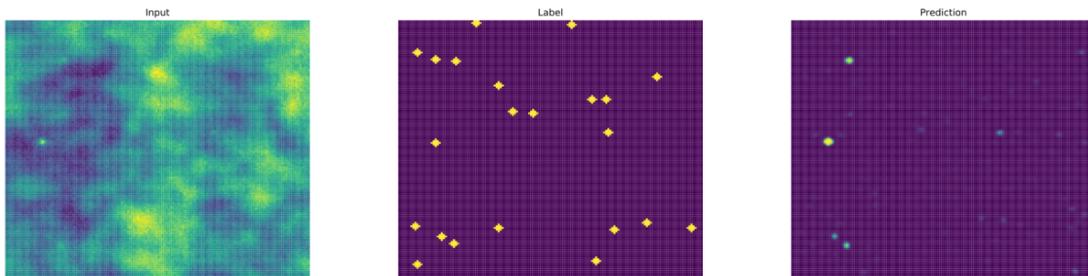


Figure 2.7: Input (left), label (centre) and prediction (right) of one of the neural networks in this work, trained for detection of point sources.

The loss function measures how accurate a prediction is in comparison to the label of the input. The lower the loss, the better the prediction. One may define different loss functions depending on the purpose of the neural network. Roughly, there are four types of purposes:

- **Binary classification:** Tell if the input belongs or not to a class. For instance, if there is a person in a picture.
- **Multi-class classification:** Tell to which class does the input belong. For instance, differentiate whether a person is a baby, teenager, adult or elder.
- **Regression:** Infer a continuous variable of the input, such as estimating the age of a person in the picture.
- **Segmentation:** Detect, localise and return the shape of objects of different classes. In other words, point-wise classification; for instance, signalling the areas of a picture where there is people.

In our work, we perform binary segmentation, as we detect if each position of an input image belongs to the class *point source* or *background*. Consequently, we use the binary cross-entropy as our loss function, which is the most common loss function for binary classifiers because it strongly penalties false predictions. Other functions have been tested, such as mean-squared errors or the Dice-Loss (see [18]), all of them yielding poorer results.

Suppose y_1, \dots, y_r are the values of the label at each output neuron (in our case, each pixel of the image), and O_1, \dots, O_r the predicted values. The binary cross entropy of a label-prediction pair in the output layer is then [19]:

$$H = -\frac{1}{r} \sum_j [y_j \ln O_j + (1 - y_j) \ln(1 - O_j)]. \quad (2.3)$$

Finally, it is convenient to use an accuracy indicator, i.e., a parameter which increases when the predictions are correct. In our work, we use the binary accuracy, which simply thresholds every pixel of the predictions at 0.5 and calculates the ratio of correctly classified pixels. As the performance of the method is thoroughly studied later, this parameter does

not have any relevance in our work, other than indicating that the model is being trained correctly.

Once all these parameters are defined, the model needs to learn from our labeled training set. The training process requires two steps: the evaluation of the training set and the recalculation of the weights. First, the training set is divided in batches, which means that instead of evaluating the whole training set at once, only a small part of it is used each time. This reduces both the training time and the memory requirements. Later, a prediction is performed on each batch of inputs, their loss function is computed and the weights of each of the layers are recalculated towards reducing this loss function. The recalculation is carried out by an optimiser, a tool which basically computes the gradient of the loss function with respect to the weights and updates them in the gradient descent direction [8]. In this work, we use the Adam optimiser [12].

The training process lasts for a prefixed number of epochs, which are the number of iterations over the whole dataset. After every epoch, the performance of the neural network is tested. A test is made by freezing the weights of the layers and performing predictions on a labeled validation dataset, which commonly consists of a small part of the training dataset that has not been used for training (this technique is common in many Machine Learning methods, see [22] for a detailed treatment). The predictions are then compared with the labels and the loss and accuracy functions are computed. As the test is carried out on data which has not been used to train the model, it produces a reliable estimate of how the network would perform in a real scenario.

Normally, the validation loss and accuracy are slightly worse than their training counterparts. If this difference is large, the model is overfitting, which means that it has learnt to recognise local patterns in the training dataset which cannot be extrapolated to general data. Overfitting is a common issue that may be avoided by improving the architecture of the network, modifying the training dataset or reducing the number of training epochs. A common indicator of overfitting is the convergence of the validation accuracy after several epochs while the training accuracy keeps rising, as seen in Figure 2.8, where training accuracy rises up to 1 whereas validation accuracy converges at 0.985.

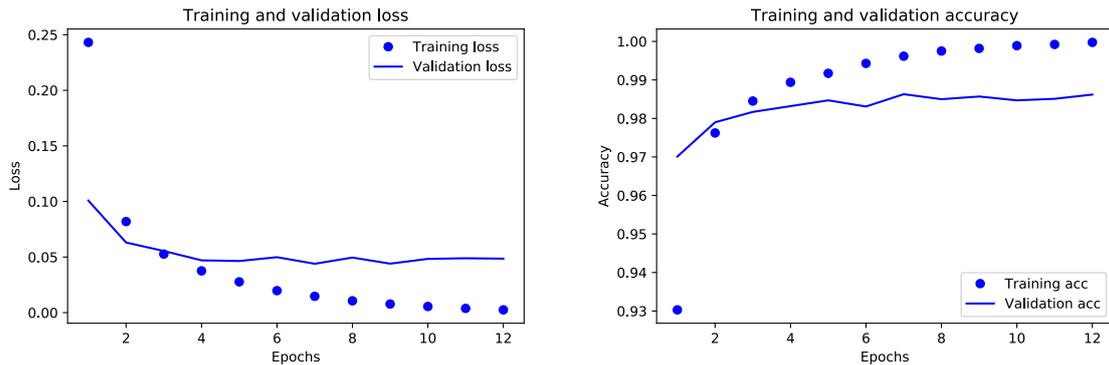


Figure 2.8: Validation against training accuracy (left) and loss (right) in an overfitting neural network.

2.2.2 Architectures

The way a neural network is built plays a key role in its performance. To begin with, there are two constraints that must be satisfied: that the input layer handles the corresponding input size, and that the output layer is adequate for the general purpose of the model. For example, in a binary classification problem, the output layer should be a single neuron whose

activation function ranges in $[0, 1]$, whereas in a regression problem it must have as many neurons as parameters to be inferred (as a rule of thumb).

Apart from dense and convolution layers, there are some others of great importance that we use in our work.

- **Max Pooling:** reduces the dimensionality of the input of a layer by keeping only the maximum values of local subarrays. For example, a 2×2 pooling layer will have the following effect in a 4×4 matrix:

$$\begin{pmatrix} 3 & 5 & -2 & 1 \\ 4 & 0 & 0 & 6 \\ -2 & -4 & 1 & 0 \\ 1 & 0 & -1 & 2 \end{pmatrix} \xrightarrow{\text{Max Pool}} \begin{pmatrix} 5 & 6 \\ 1 & 2 \end{pmatrix}.$$

Pooling layers are used to reduce the data size, thus decreasing the number of weights, preventing overfitting, accelerating the training and guaranteeing an earlier convergence. In addition, they help extracting the main features of the dataset. The main inconvenient is that an extensive use may incur in precision losses.

- **Batch normalisation:** for every batch in the training set, transforms the input of a layer to normalised data, with zero mean and unit variance. Its main effect is to avoid numerical instabilities and to allow training at higher learning rates.
- **Transpose convolution:** transforms the input in the opposite direction of a normal convolution, i.e., from an output-shaped result of a convolution to the shape of its input, maintaining a connectivity among them [4]. Instead of computing the scalar product of a subarray of the input and a kernel as in a convolution, it multiplies the points of the input by a kernel, producing kernel-sized results which are placed on a larger array. This layer is used to upsample the data, particularly after convolutions with strides and pooling layers.

The next step is putting a set of layers that fit together and are optimal for our problem. Unfortunately, one of the major drawbacks of neural networks is its variability and the difficulty of predicting whether a model will work or not without actually testing it. Therefore, hours of drafts, test and trial are required. Nevertheless, the study of previous works such as Long [15] and Ronneberger [21], and our own analysis of the problem derived in a series of guidelines that were later validated experimentally. These are the following:

- Input raw arrays of data, without previous normalisation. By this we attempt to avoid intense foregrounds and bright sources to completely dominate the image, trying to focus the network on the shape of the objects rather than their brightness.
- Use fully convolutional networks (no dense layers), since our aim is to detect localized small features in the images and performing segmentation. Introducing dense layers may diminish the precision in the positions.
- Use normalisation layers to help working with the large values of the arrays and speed the training up.
- The final layer must be a point-wise sigmoid activation function. This ensures that the output is a 150×150 array whose values range in $[0, 1]$. In this way, we can obtain a “*network probability*” of having a point source at a specific location.

We must note that the description of CNNs we give in this section, even if sufficient for describing our work, is not a complete picture of the state-of-the-art. The field is evolving quickly and there exist other architectures, such as encoders/decoders or recurrent networks which are out of the scope of this work.

2.3 Matched Filter

The matched filter is a classic method for enhancing objects with a known shape, such as point sources. As we already mentioned, we define it because it is a trusted method ideal for establishing comparisons with the CNN. The filter provides an optimal Signal-to-Noise Ratio (SNR), defined as the ratio between the expectation value of the maximum value of a signal $s(\mathbf{x})$ and the standard deviation of the background field σ_{map} ,

$$SNR = \frac{\langle \max s(\mathbf{x}) \rangle}{\sigma_{\text{map}}}. \quad (2.4)$$

Recall that in our problem, the profile of the point sources is a radial Gaussian profile $\tau(r) = \exp(-r^2/2\sigma^2)$, whose Fourier transform is $\tau(q) = \sigma^2 \exp(-(q\sigma)^2/2)$. If the point source has an amplitude A , the signal will be $s(r) = A\tau(r)$. The derivations that follow are based on the work by López-Caniego, ([16], [17]).

Let us first define a 2D filter $\psi(\mathbf{x}, R, \mathbf{b})$ where R is the scale of the filter, \mathbf{b} a translation vector (we assume the point source is centered in $\mathbf{b} = 0$) and \mathbf{x} the position. If we filter a field $f(\mathbf{x})$ with ψ , the result will be the convolution

$$w(R, \mathbf{b}) = \int f(\mathbf{x}) \psi(\mathbf{x}, R, \mathbf{b}) d\mathbf{x}. \quad (2.5)$$

The matched filter (MF) can now be obtained by imposing radial symmetry to the filter, $\psi(\mathbf{x}, R, \mathbf{b})$ and the conditions

- $\langle w(R_0, 0) \rangle = s(0) = A$. This means that $w(R_0, 0)$ is an unbiased estimator of the flux of the source.
- the variance of $w(R, \mathbf{b})$ has a minimum at the scale R_0 , so $w(R_0, \mathbf{b})$ is the minimum variance unbiased estimator of the source flux.

Such a filter is given by the equations

$$\psi_{MF} = \frac{1}{a} \frac{\tau(q)}{P(q)}, \quad a = 2\pi \int q \frac{\tau^2(q)}{P(q)} dq, \quad (2.6)$$

where $P(q)$ is the power spectrum of the unfiltered map. As the profile of the source is fully characterized by the FWHM of the map, the MF can be constructed accurately if $P(q)$ is known. The parameter a is a normalization factor which is only necessary when an estimation of the amplitude of the signals is required; as we focus on the detection of the sources, we set it to 1 in our work.

Despite its optimality under certain ideal conditions, the matched filter presents two main challenges when applied to real data. First, the background is not usually Gaussian, due to the foregrounds. Second, the estimation of the power spectrum of the image may yield a large error, particularly for the background inhomogeneity caused by foregrounds and, eventually, anisotropic instrumental noise. Nevertheless, the matched filter is very useful for comparison purposes in ideal or weak foreground scenarios. We suggest two implementations of the MF: a spherical one in harmonic space, and a local one.

2.3.1 Sphere implementation

The implementation of the matched filter in the sphere requires the knowledge of the theoretical angular power spectrum of the map, so it is only tested in ideal conditions: fully-characterised CMB spectrum, Gaussian noise and absence of Galactic foregrounds. The implementation is straightforward according to the previous derivations, taking into account that we must work using spherical harmonics. Let C_ℓ be the coefficients of the CMB angular power spectrum and $N_\ell = \sigma_{\text{noise}}^2 4\pi/N_{\text{pix}}$ the coefficients of the noise power spectrum (notice that these are constant as it is white noise). As the map is smoothed with a Gaussian beam of standard deviation σ (recall that $\sigma = \text{FWHM}/2.355$) and the convolution in real space becomes a product in harmonic space, the power spectrum of the map D_ℓ is

$$D_\ell = \omega_\ell^2 C_\ell + N_\ell, \quad \omega_\ell = \exp(-\ell(\ell+1)\sigma^2/2). \quad (2.7)$$

The ω_ℓ coefficients are the spherical harmonics coefficients of a Gaussian function. These also correspond to the point source profile $\tau(q)$, so from equation 2.6 we have that the harmonic coefficients of the matched filter ξ_ℓ are simply

$$\xi_\ell = \omega_\ell/D_\ell. \quad (2.8)$$

Again, the convolution of equation 2.5 becomes a product, so that $\hat{a}_{\ell m} = a_{\ell m} \xi_\ell$ are the spherical harmonics coefficients of the filtered map. The performance of this implementation will be discussed in Chapter 3.

2.3.2 Local implementation

The power spectrum of the background varies across the sky in reality. In the Galactic plane the foregrounds are very bright, whereas the CMB dominates otherwise. These two components present a very different power spectrum, so it is highly recommendable to generate independent filters for each region, which can be studied on flat patches via a spherical projection. For this reason, we suggest a local implementation of the MF. Now, the power spectrum of each patch $P(q)$ is not usually given by a theoretical model, and should be estimated from the data itself.

The algorithm we use applies a sigma clipping to remove the brightest peaks (frequently strong point sources) of the patch, in order to perform the estimation of the power spectrum. Later, it estimates the radial profile of the patch and applies equations 2.6 to process the input in Fourier space.

As the input patches have a small size, the estimation of the power spectrum is usually imprecise and the filter in the Fourier space does not output a clear result (spurious niches and bright points in the image are sometimes created). Hence, we suggest a real space implementation, which consists in performing a two dimensional convolution of the image and a kernel, obtained from the power spectrum estimation of the algorithm we just mentioned. In the real space implementation, we use a mirror padding at the edges of the input image to minimise the border effects of the convolution.

The real space matched filter will be our preferred choice, as we will not normally know the theoretical power spectrum of the map and the local evaluation is the best option available. We evaluate and compare the methods in Chapter 3.

2.4 Object extraction

Once the maps are processed and the point sources are amplified against the background, we need to label the point sources individually. This is called detection. We will consider that

an area of the image is a point source if the following criteria are satisfied:

- There is a group of brilliant pixels clearly resolved and differentiated from the background.
- There is a sufficient number of pixels in the group.
- The group is not located too close to the edge of the image (to avoid undesired effects of the convolution).

For this purpose, we use the software SExtractor [3]. This program builds a catalogue of objects from an image, including parameters of the objects such as their position, shape, brightness and size. In our work, we use the Python library `sep` [2], which implements the features of SExtractor.

Besides the object catalogue, the `sep` extraction method also provides us with a segmentation map. This map is an array of the same size as the input which sets all the background to zero and the detection regions to an integer different from zero. The segmentation map will be useful for evaluating the performance of the methods.

2.4.1 Detection criteria

The three ideas listed above need to be translated into objective criteria, which will be different for the matched filter and CNN according to the different characteristics of their respective outputs. The criteria are the following:

- **Matched Filter:** A minimum of 4 neighbour pixels above an $n\sigma$ threshold, where n is a fixed real number and σ is the standard deviation of the pixels of the filtered image. n usually ranges between 3 and 5.
- **CNN:** A minimum of 4 neighbour pixels above a fixed threshold $k \in (0, 1)$. The threshold is fixed and depends on the particular CNN, but not on the input.

In addition, the brightest pixel of the object must not be less than 5 pixels next to the nearest edge of the image.

Notice that `sep` library includes an implementation of the matched filter in real space. The filter can be used directly at the time of extracting the objects of the image if a window function is specified. However, after careful testing, we obtained far better results using our own implementations of the matched filter (most likely due to the individual estimation of the power spectrum of the images that we perform), so we decided to use a constant window function.

2.4.2 Method evaluation

Finally, we need to evaluate the performance and accuracy of our methods (CNN, MF). We do so by comparing the objects we obtain with SExtractor to the corresponding point-source-only image. Our aim is obtaining the completeness curve $C(S)$ and the reliability R of the methods using simulations, where we know the location of the point sources. Therefore, for a certain detection threshold T , our evaluation algorithm should not only return the number of detections but:

1. Count the sources in the point-source-only images which are above a flux S_{cut} .
2. Count the detections of the method which correspond to sources above S_{cut} .

3. Count the detections of the method which do not correspond to any source in the point-source-only image. These are called **spurious** detections.
4. Count the detections of the method which do correspond to some actual source. These are called **real** detections.

The algorithm is run on every image of the dataset at several fluxes S_{cut} so the completeness curve can be obtained. In the end, it is interesting to evaluate different thresholds to study how completeness and reliability are affected.

Our core algorithm, for single image detection, is presented below. The nomenclature we use is `output_image` for the processed image by the CNN or the MF, `ps_img` for the point-source-only image, `cat_` for the objects catalogue and `map_` for the segmentation map. The value T_{ps} is the threshold for the point-source only image; this is usually set very low so every source is included in the catalogue. The method `extract` refers to the routine in the `sep` library, which outputs the object catalogue and the segmentation map.

Algorithm 1 Single image evaluation method.

Require: `output_img`, `ps_img`, S_{cut} , T , T_{ps} .

Ensure: `spur_counts`, `true_counts`, `detected_above`, `undetected_above`, `detected_below`

```

1: cat_output, map_output = extract(output_img,  $T$ )
2: cat_ps, map_ps = extract(ps_img,  $T_{ps}$ )
3: spur_counts, true_counts, detected_above, undetected_above, detected_below = 0
4: for source  $P$  in cat_ps do
5:   if  $4 < P.x < 145$  and  $4 < P.y < 145$  {check source is far from edges} then
6:     if  $\max(\text{map\_output}[P.x - 2 : P.x + 3, P.y - 2 : P.y + 3]) > 0$  {tolerance margin} then
7:       if  $P.flux > S_{\text{cut}}$  then
8:         detected_above ++
9:       else
10:        detected_below ++
11:      else
12:        if  $P.flux > S_{\text{cut}}$  then
13:          undetected_above ++
14:        for source  $P$  in cat_output do
15:          if  $4 < P.x < 145$  and  $4 < P.y < 145$  then
16:            if  $\max(\text{map\_output}[P.x - 2 : P.x + 3, P.y - 2 : P.y + 3]) > 0$  then
17:              true_counts ++
18:            else
19:              spur_counts ++
20:          return spur_counts, true_counts, detected_above, undetected_above, detected_below

```

The actual implementation of the algorithm incorporates other functionalities to characterise the point sources, such as keeping track of their flux. When the algorithm is applied to the whole training set, the reliability R , completeness $C(S)$ and number of counts N are given by:

$$R = \frac{\sum \text{true_counts}}{\sum (\text{spur_counts} + \text{true_counts})}. \quad (2.9)$$

$$C(S) = \frac{\sum \text{detected_above}(S)}{\sum (\text{detected_above}(S) + \text{undetected_above}(S))}. \quad (2.10)$$

$$N = \sum (\text{detected_above} + \text{detected_below}). \quad (2.11)$$

Chapter 3

Detection in CMB and Gaussian noise simulations

In this chapter, we detail the behaviour and characteristics of our method and compare it to the matched filter in maps composed of CMB and Gaussian noise, without foregrounds. The motivation for our study is twofold. On the one hand, building and testing different CNN architectures in a simple scenario, where detections do not depend on the region of the sky we observe, allows us to make fast comparisons and retain the architectures that work better. On the other hand, we know that, in these conditions, the matched filter is optimal in the sense of providing a maximum amplification of the point sources. Therefore, a CNN whose performance is comparable to the detection after the application of the matched filter would indicate the suitability of CNNs for our problem. Such a performance is indeed a relevant result by itself.

Besides, a thorough analysis of the methods and the results in these conditions enabled us to build a robust pipeline and improve our code, before facing a more challenging scenario in Chapter 4.

3.1 Method characteristics

3.1.1 Initial tests

Prior to the presentation of the details and the performance of our methods, we believe it is valuable to include some conclusions that we extracted from testing the algorithms in the earliest stages of the project. From these tests we obtained information about how the actual algorithms that we use should be built and about their strengths and weaknesses.

Initially, we tested both shallower and deeper architectures of CNNs, with varied degrees of dimensionality reduction. The conclusions are the following:

- CNNs with only a few convolution-deconvolution layers are not able to clean the background completely, even if there is an enhancement of the point sources. Therefore, the number of spurious detections using these architectures is much higher.
- Reducing the dimensions of the data below 1/16 of the original input size in the intermediate layers generally causes a loss of precision in the predictions, probably due to an important loss of information. Mixed architectures, such as interconnected layers (see for instance [15]) have not been tried.
- Keeping the original dimensions of the image at all times slows down training and does not improve the final result.

A different type of test we have also made concerns the Gnomonic projections of the maps. The first procedure we used was directly using the division into 3072 HEALPix $N_{\text{side}} = 16$ patches. The main advantage of this method is that we do not need to care about lost or repeated point sources when performing the projections. However, the deformation of the point sources, especially near the galactic poles, prevented the implementations of the matched filter in the plane from functioning effectively. Besides, this deformation would probably require a specific training for the CNN in order to improve the detection near the Galactic poles.

Finally, we remark the importance of not taking into account the detections near the edges of the inputs, particularly with CNNs (which apply a zero padding) and the real space MF (symmetric padding). We found that a margin shorter than 5 pixels at each side decreased the completeness of our methods. As a result, we need to have overlapping projections in order to not miss any point sources.

3.1.2 Simulations

All the sky simulations in this chapter are the sum of three components. First, a realization of the CMB created from the Planck best-fit power spectrum, at a resolution (FWHM) of 5 arcmin. Second, the point sources, smoothed used a Gaussian kernel also with a FWHM of 5 arcmin. Third, a layer of Gaussian white noise. These components are created and added in the sphere, and the sum of them is later projected onto plane patches.

Separately, we tested simulations where the point sources were directly created on the plane (the pipeline can be found on our code). This means that the CMB has been simulated in the sphere, later projected, and the point sources and the noise have been simulated directly on the projected (plane) patches. This type of simulation was produced to test the real space matched filter under totally ideal conditions, without any kind of deformation of the point sources and no correlation in the noise. The results we obtain with this type of simulations are not presented in detail, although we will introduce some commentaries on them.

3.1.3 CNN

The architecture we have used for our CNN follows both the conclusions mentioned above and the prerequisites in Section 2.2, and is based on our previous experience from the initial tests, as we mentioned before. It encodes the input to a quarter of its original size in the MAX Pooling layer, upsampling it shortly afterwards in a Transpose Convolution layer. We use the ReLU activation function in all convolution layers but the last one, where we apply a sigmoid activation to obtain an output between 0 and 1 at each position. This architecture is shown in detail in Table 3.1.

The model has been trained with two full simulations of the CMB, noise and point sources. The simulations contained 85000 and 105000 point sources each (to introduce a larger variety in the number of objects per patch), both following the flux distribution in Figure 2.3. Each of them provides us with 3072 patches, which add up to a training set of 6144 images. The labels we use are binary images, where zeros indicate the background and there is a circle of ones of a 3-pixel radius (see Figure 2.7 for an example of input-label pair) at every point source. This radius has been chosen in regard to the FWHM of the maps, as it captures the brightest part of the sources. Among the 6144 images, we kept 344 for validation purposes; therefore, training is made with 5800 images at each epoch. We used a batch size of 50 and an optimiser learning rate of 0.01.

The evolution of the training loss and accuracy at training and validation is shown in Figure 3.1. As we can see, the loss and the accuracy converge rapidly after a few epochs.

No	Layer type	k	Input size	Output size	Filter size	s_x	s_y
1	2D Convolution	16	150×150	150×150	5×5	1	1
2	Batch Normalization	-	150×150	150×150			
3	ReLU Activation	-	150×150	150×150			
4	2D Convolution	32	150×150	150×150	5×5	1	1
5	Batch Normalization	-	150×150	150×150			
6	ReLU Activation	-	150×150	150×150			
7	MAX Pooling	-	150×150	75×75		2	2
8	ReLU Activation	-	75×75	75×75			
9	2D Transpose Conv.	64	75×75	150×150	3×3	2	2
10	Batch Normalization	-	150×150	150×150			
11	ReLU Activation	-	150×150	150×150			
12	Sigmoid (logistic)	-	150×150	150×150			

Table 3.1: Architecture of the CNN used for detection of point sources in CMB and Gaussian noise scenarios. k is the number of filters and s_x, s_y are the strides.

Nonetheless, these parameters are not useful to make a precise evaluation of the performance of our CNN, as they are not directly related to the completeness and the reliability in the detection problem.

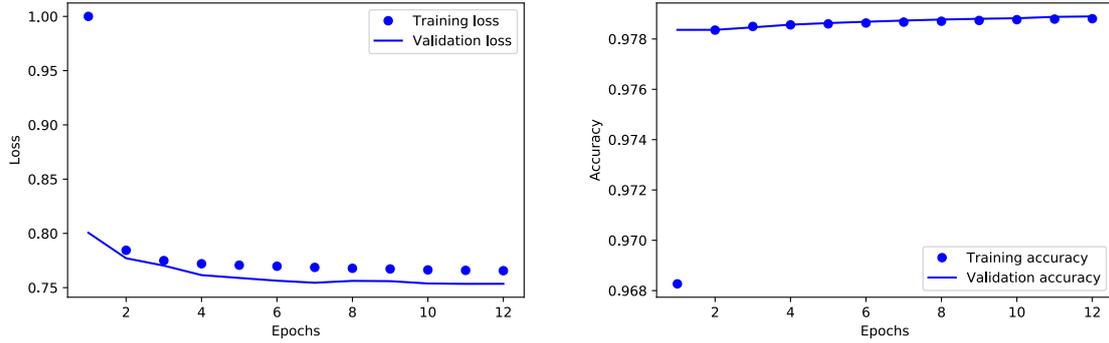


Figure 3.1: Evolution of the training and validation loss and binary accuracy of our network during 12 training epochs. The loss is normalized to the highest loss, in epoch 1.

The training time in our GPU (an integrated Intel UHD Graphics 620 unit) is approximately of 18 minutes per epoch. We found that this architecture in particular is very effective for detecting the point sources, as it outputs an array of values below 0.2 where there is only background, and values near 1 where there is a bright point source. Values in that interval are normally assigned to faint point sources, and some intense statistical background fluctuations.

We found that the CNN is a robust method that works accurately even if it is tested on images that differ slightly from the training conditions. This is the case when one works over simulations on the sphere or simulations on the plane; a CNN trained over any of the two of them performs equally well on both. Besides, as we show in Figure 3.2, when we make predictions of inputs with increasing instrumental noise, the model resists variable noise conditions up to a certain extent.

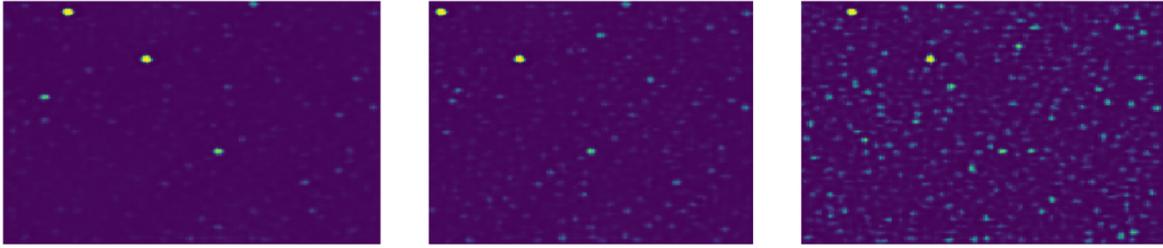


Figure 3.2: Prediction of the CNN (Table 3.1, Figure 3.1) on an image from the validation set (left), the same image adding a second layer of white noise (center), and the same adding the noise twice (right).

A detailed discussion on the improvement of the model performance at different training times is found in Chapter 4, as it is particularly relevant in realistic simulations for distinguishing bright compact objects from the foregrounds from actual point sources. For the simulations in this chapter, we found that training the model for 8 epochs is generally sufficient for achieving a very good result. We trained for 12 epochs to strive for a further refinement; we show a visual comparison of both levels of training in Figure 3.3, where it is seen that the difference is minimum although favourable for the 12 epoch CNN. However, fewer epochs are not sufficient for properly separating the background from the sources, as we will also show in Chapter 4.

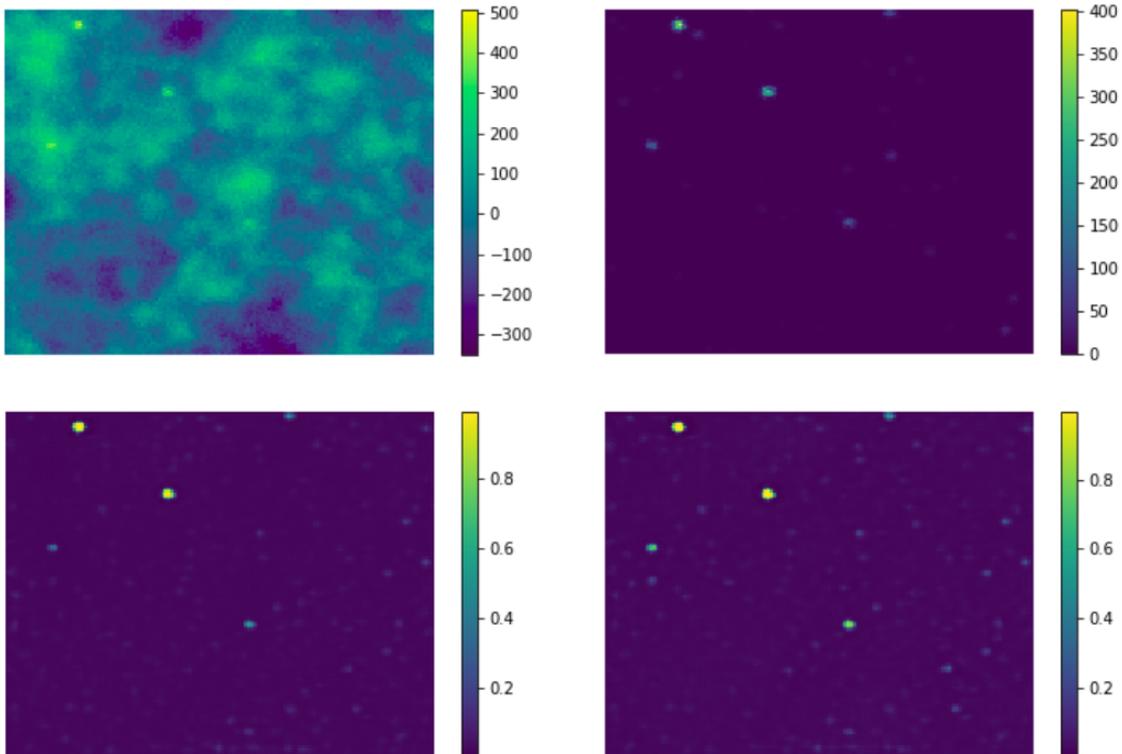


Figure 3.3: Example of a prediction of our CNN trained with 8 (bottom-left) and 12 (bottom-right) epochs, respectively. The top-left picture is the input and the top-right its corresponding point-source-only image.

3.1.4 Matched Filter

We introduced the matched filter in Section 2.3 as an optimal method for amplifying point sources embedded in a random and homogeneous Gaussian background, which are exactly the conditions in this chapter. The optimal implementation of the matched filter for this problem would be in the sphere, so we use it for comparison purposes. However, in realistic conditions (Galactic foregrounds) one must work on patches to perform local estimations of the angular power spectrum of the background.

Sphere

The sphere implementation of the matched filter yields optimal results, as we perfectly know the theoretical angular power spectrum of the background. The method does indeed enhance the point sources and cleans the background very effectively, although it produces rings (somehow resembling an Airy diffraction pattern) surrounding the point sources, as seen in Figure 3.4. These rings might be problematic, mainly for two reasons. First, the rings created by the filter near a bright source are strong and blind all the faint point sources next to it, preventing their detection. Second, sections of the closest ring are able to fool the object extraction algorithm, sometimes producing spurious sources.

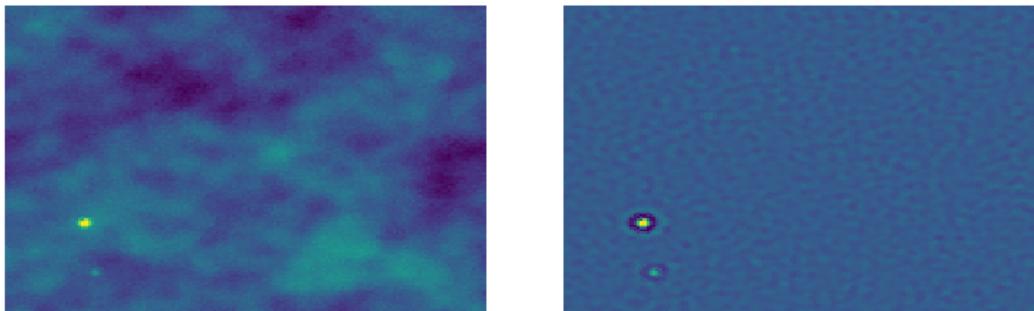


Figure 3.4: Input (left) and result (right) of the spherical MF in an input where a very bright point source is present.

Real space

The rings in the spherical implementation appear in the real space implementation too, although they acquire an also problematic diamond-like shape, as we will see in the comparison on Figure 3.8.

As we mentioned, we created specific simulations in the plane for studying the performance of this method in ideal conditions. However, its behaviour does not experience a large improvement in this case.

3.2 Method evaluation and comparison

As we discussed previously, the performance of the methods is evaluated by two main indicators: the completeness curve $C(S)$ and the reliability R . These parameters will tightly depend on the thresholds T we set for the output (sigmoid function) of the CNN and for the

MF (as number of σ of the filtered map). Therefore, the study is carried out for different thresholds in each of the methods.

The comparison is made by evaluating each of the methods on an all-sky simulation (3072 images). The simulation is different from the ones used for the training of the CNN, in order to maintain independence and avoid a possible bias introduced by overfitting. The time that the CNN requires to perform the all-sky prediction is of approximately 90 seconds, whereas the real space MF requires several minutes. The spherical MF is faster than any of the previous methods as it is directly provided with the theoretical power spectrum. It however requires to later project the map for performing detection, which slows down the process.

Next, we introduce our results for the spherical matched filter and for the real matched filter separately. The simulation we use is the same in both cases, and therefore the results we obtain with the CNN are identical. We recall that our CNN does not intend to estimate the amplitude of the point sources but only to detect their locations in the sky.

3.2.1 CNN and spherical MF

In Figure 3.5 we present the completeness curves for the spherical matched filter and the CNN at different thresholds.

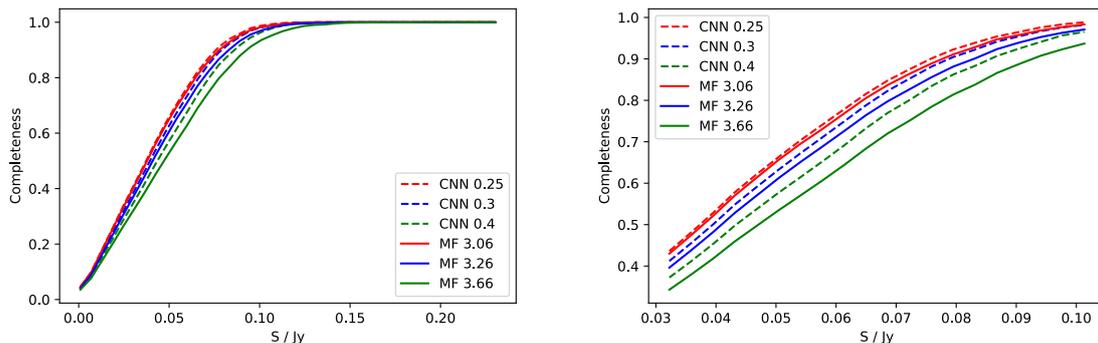


Figure 3.5: Completeness curves $C(S)$ of the evaluation of the spherical matched filter and the CNN over a complete all-sky simulation. Thresholds are chosen such that reliability factors $R = 0.90, 0.95, 0.99$ are obtained, for comparison purposes. Similar colours correspond to the same reliability. The plot in the right is a zoom at the low flux range.

We can see that both methods perform correctly, as they rapidly converge to full completeness. In comparison, the CNN performs better than the spherical MF at any of the reliability levels we have set, as its completeness curves for similar reliability stand always on top of the curves for the MF. The most relevant evaluation parameters (reliability, flux at 95% and 80% completeness) are shown in Table 3.2.

Matched Filter				CNN			
T / σ	S_{95} / Jy	S_{80} / Jy	R	T	S_{95} / Jy	S_{80} / Jy	R
3.06	0.091	0.064	0.895	0.25	0.088	0.063	0.897
3.26	0.097	0.069	0.951	0.30	0.093	0.066	0.949
3.56	0.113	0.078	0.990	0.40	0.099	0.072	0.989

Table 3.2: Summary of the parameters of the evaluation of the spherical matched filter and the CNN over a complete all-sky simulation. S_{95}, S_{80} are the flux at which a 95% and 80% completeness is obtained, respectively.

The behaviour of the CNN is particularly outstanding when we require a very high level of reliability, such as in the green curves in Figure 3.5. We remark that even if the difference in completeness at mid-range flux (0.05-0.10 Jy) may appear small, it is significant due to the larger number of point sources in that range in comparison to the number at a higher flux (see Figure 2.3).

3.2.2 CNN and real space MF

As we commented before, the spherical implementation of the MF is unfeasible in realistic scenarios because the power spectrum of the background needs to be estimated locally. Hence, we need an implementation that works on patches, such as the real space matched filter.

Next, we introduce the same evaluation plots (Figure 3.6) and summary of results (Table 3.3), using the same CNN as in the previous analysis and the real space matched filter.

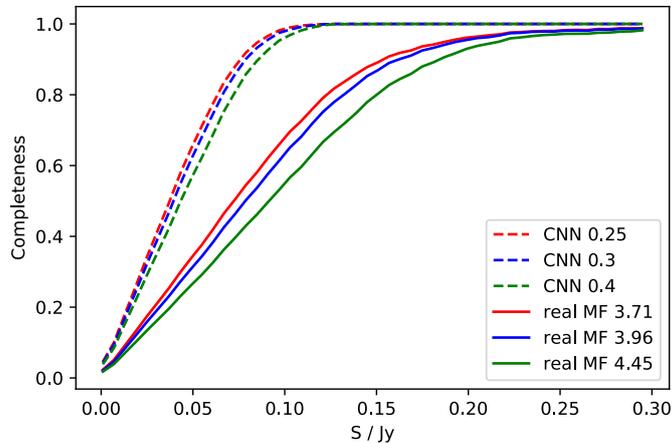


Figure 3.6: Completeness curves $C(S)$ of the evaluation of the real space matched filter and the CNN over a complete all-sky simulation. Thresholds are chosen such that reliability factors $R = 0.90, 0.95, 0.99$ are obtained, for comparison purposes. Similar colours correspond to the same reliability.

Matched Filter				CNN			
T / σ	S_{95} / Jy	S_{80} / Jy	R	T	S_{95} / Jy	S_{80} / Jy	R
3.71	0.180	0.122	0.901	0.25	0.088	0.063	0.897
3.96	0.188	0.131	0.951	0.30	0.093	0.066	0.949
4.45	0.213	0.149	0.989	0.40	0.099	0.072	0.989

Table 3.3: Summary of the parameters of the evaluation of the real space matched filter and the CNN over a complete all-sky simulation. S_{95}, S_{80} are the flux at which a 95% and 80% completeness is obtained, respectively.

The results show that the performance of the matched filter is not nearly as satisfactory as in the case of the spherical implementation, presumably due to the poor local estimation of the power spectrum due to the size of the images. Nevertheless, the correctness of the real space matched filter is clear, as its completeness eventually converges to 1 regardless of the pursued reliability.

Finally, in Figure 3.7 we show a map of the all-sky detections performed with the CNN, for a threshold $T = 0.25$ (and therefore a 90% reliability). In the figure, the points represent the different categories of sources we obtain from our evaluation methods: real sources, spurious

sources and undetected sources above a flux $S = 0.080$ Jy. Since the maps are isotropic and present no privileged directions, these points are randomly distributed in the sky as expected.

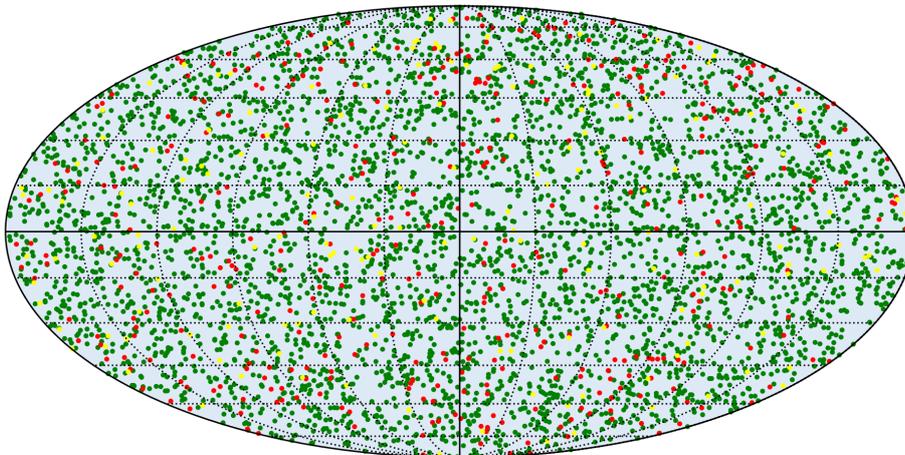


Figure 3.7: Mollweide all-sky projection indicating the detections of the CNN at threshold $T = 0.25$ ($T = 0.90$). Green points are correct detections, red points are spurious detections and yellow points are undetected sources with flux $S > 0.080$ Jy.

3.3 Discussion

There are diverse points to discuss following the results presented in this chapter. Therefore, we structure this section in three parts: a discussion on the performance and behaviour of the methods, on the capacity of the methods to estimate the flux of the sources, and a brief summary and conclusions.

Method performance

The first result we extract from our method evaluation is that our CNN performs better than both implementations of the matched filter for any of the reliability goals we have set. This result is remarkable and proves that CNNs are not only a valid but also a very effective approach to the problem of point source detection in the microwave sky. There is no doubt that one of the reasons for that performance is that CNN has been trained with maps who present very similar characteristics (recall that the maps are produced with the same noise and angular power spectrum) to the map we used for validation. In addition, as the maps are isotropic, patches are very similar among them. Nevertheless, we have seen that it is a robust method which can adapt to changes in the noise and to slight differences in the shape of the objects we aim to detect, which was seen by applying the same model over point source simulations in the plane without a new training. This adaptation capacity also suggests that our model would be able to detect point sources even if we are not able to simulate the exact beam of our instrument, which might be the case in a real observation. Therefore, there are arguments for being optimistic in our attempt to build a reliable model if we generate highly realistic simulations of the sky, and in particular, of the Galactic foregrounds.

The superior performance of the CNN also implies that there is room for the improvement of the current detection methods. The MWH2 method, currently used for the PCCS and PCCS2 (see Section 1.2) presents a very similar performance to the matched filter when applied under ideal conditions [16], which are the same conditions of CMB and Gaussian

noise that we test in this chapter. Therefore, surpassing an ideal (spherical) implementation of the matched filter in this case means that the CNN might also improve the MWH2. A thorough comparison between both methods is out of the scope of this project, but is an interesting topic of research for future work.

From the introduction of the spherical matched filter, we have frequently asserted that the method is optimal in the case we study. A natural arising question is then why does the CNN obtain a better result, if this is mathematically proven to be impossible. The answer may well be in the definition of optimality we introduced. In Chapter 2, we presented the matched filter as the minimum variance unbiased estimator of the flux of a source. Hence, any other method should yield poorer results than the matched filter when *estimating* the flux of the source, but not necessarily when *detecting* its presence. The aim of our CNN is not to estimate this flux, but only to signalise the zones of the sky where there is a point source. In fact, as we mentioned, the output of the CNN resembles a point source probability, as it is 1 or very close to 1 at bright point sources and close to 0 at the background.

A second point of interest is the comparison between both implementations of the matched filter. The spherical implementation offers a great result, but the real space performs far below our expectations. The main drawback of the spherical implementation is that we can only use it for comparison purposes in the ideal case, but it can not be used in the presence of foregrounds; therefore, we lack of a method which performs well for comparing to the CNN in a real scenario. Furthermore, the performance of the real space implementation does not improve considerably when the point sources are simulated on the plane, which in theory is a more favourable scenario for the filter.

The main reason why the real space matched filter exhibits such a low quality is a poor calculation of the power spectrum of the image. This is mainly because the inputs are evaluated individually and the size of the patches is not enough for allowing a proper estimation of all of the stochastic properties of the background. As a result, some of the predictions in which the power spectrum is correctly estimated are processed very accurately, whereas others result in more diffuse images, fact that we have experimentally verified. This method also induces errors in our algorithms by creating faint spurious sources next to the brightest ones, as we previously mentioned. In Figure 3.8, we show a comparison of how the three methods we use deal with the same input and of the further processing by the object extraction algorithm at a low threshold. If the threshold is increased, the problem with the spurious sources disappears, but in contrast many faint sources are undetected, reducing the completeness at a low flux. In particular, in Figure 3.8, the real space matched filter produces two additional spurious sources, one of them generated by a background fluctuation and the other by the ring of another point source.

A possible solution for the real space matched filter issues (which would probably improve the Fourier space matched filter as well) would be projecting larger images, although the spherical matched filter would not be beaten in any case. In Figure 3.8, one can also see the quality of the prediction of the CNN, as it is the only method which clears the background almost completely. Another possible improvement would be performing the convolution of the real space MF with the kernel that results from averaging the kernels for all the patches. The idea behind it is that the estimation of the power spectrum on a large set of images might be more accurate than on one of them only; however, we have attempted this with negative results.

In both implementations of the matched filter, there is a distortion created by the algorithm near the brightest sources which, as we see, may produce spurious detections. This does not occur in the predictions of the CNN, whose quality is remarkable taking into account that the model has been trained with individual images whereas the spherical matched

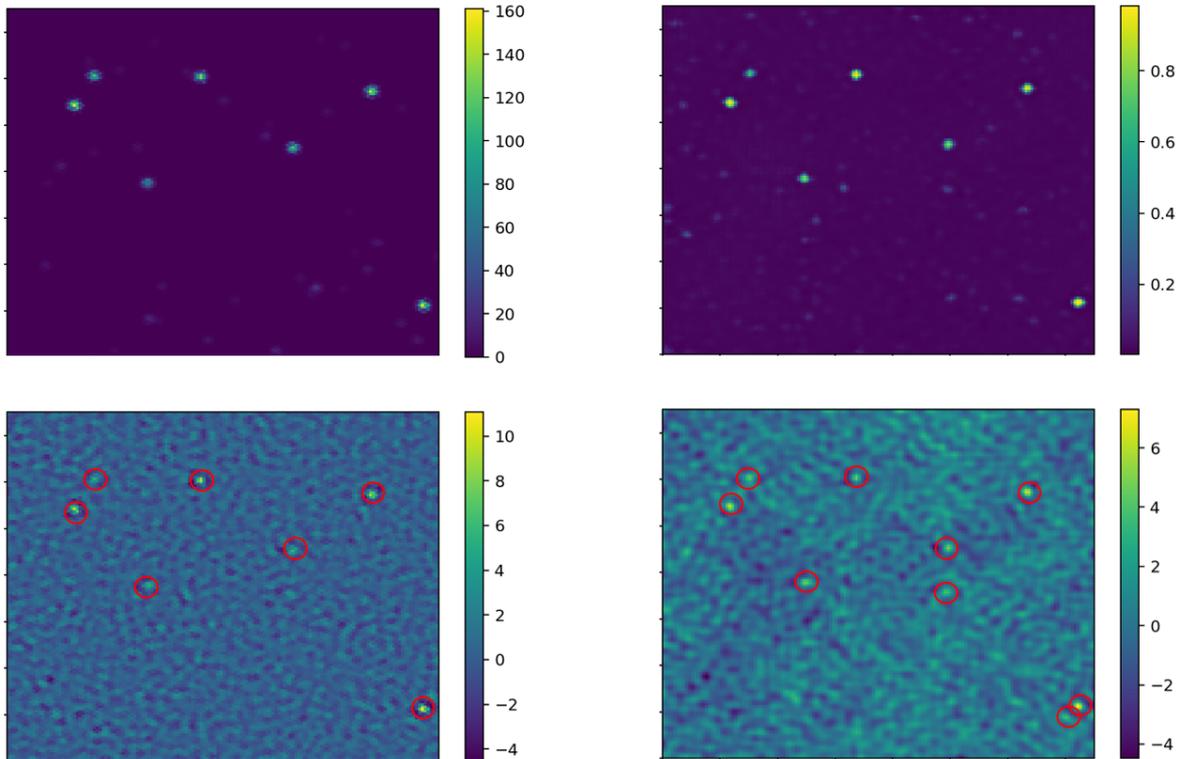


Figure 3.8: Comparison of the predictions of the CNN (top-right), spherical matched filter (bottom-left) and real space matched filter (bottom-right) for the same image. The top-left picture shows the point source only image. Both matched filter outputs are normalised by the standard deviation of the prediction σ . The red circles point the areas where the object extraction algorithm (see Section 2.4) detects objects, for a threshold $T = 2.8\sigma$ in both matched filters. The detections for the CNN output at a threshold $T = 0.20$ are the seven brightest spots in the image.

filter uses the exact power spectrum of the map. In particular, as we see in Figure 3.5, the CNN is able to perform predictions with a very high reliability without a very significant reduction of the completeness at a low flux. This reduction is greater in both matched filter implementations.

Flux estimation

We mentioned above that a probable reason for the superior performance of the CNN despite the proven optimality of the spherical matched filter is the definition of optimality itself. Therefore, it is interesting to analyse the performance of both methods at estimating the flux of the sources. In Figure 3.9, we show the outputs of the CNN and spherical MF compared to the actual flux of the point sources.

We can see that the CNN mainly lacks of the ability of performing flux estimations. It is indeed able to distinguish very bright sources, to which it associates the highest output. However, it is not possible to identify the flux of those sources as they are all identified by 1. Furthermore, the output for faint sources is variable and does not allow any reliable estimation of the flux of the source. On the other hand, a clear correlation among prediction and flux appears in the case of the MF, proving that it is indeed a good estimator of the flux of the sources.

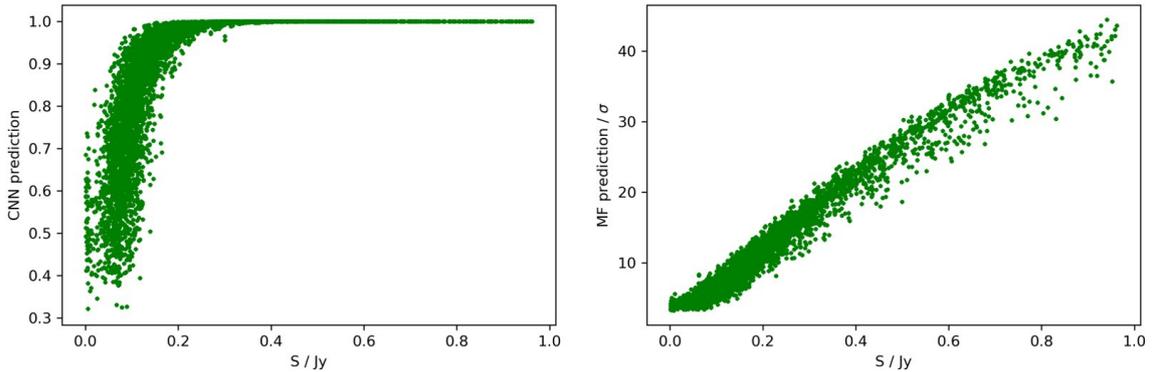


Figure 3.9: Scatter plot of the output value of the CNN (left) and spherical MF (right) against the real flux of the source, for every correct detection at the lowest thresholds in Figure 3.5 ($R = 0.90$).

Conclusions

The first conclusion we can extract from our study is that CNNs are a promising method for the problem of point source detection. CNNs present several positive characteristics both in general and in comparison to the matched filter beyond their accuracy and performance; we list these main features next. First, they clean the background very effectively, facilitating the task of detecting the sources from their predictions. Second, they are able to process the whole sky in a very short time. Third, they do not need to perform power spectrum estimations or use any assumptions on our data, other than a careful training. Fourth, they are robust and maintain a reliable performance in different conditions, such as in simulations of point sources in the plane or with variable noise. The main disadvantage that one may find is the need of producing training simulations which are very similar to the real data; otherwise, it would be challenging to evaluate their reliability. Thanks to experiments such as Planck, those kind of simulations are obtained easily.

Another conclusion is that, our CNN is better than the matched filter at detecting the point sources, but not at estimating their flux. Therefore, we suggest a double method approach when a full characterisation is required; use the CNN for detecting first and then estimate the flux in those areas by means of a matched filter, wavelets, a second regression-oriented CNN or other accurate methods.

Finally, we would like to remark that there is no proof that the architecture and training we present for our CNN are optimal for our problem, so that even better performances may be achieved by further refinements. We suggest two in particular, for future work. First, increase the number of faint point sources to reinforce the learning in the low flux range, where we achieve a lower completeness. One does not need to be totally realistic with the simulation of point sources, but perhaps should include a larger number of those objects for which the performance of the CNN is worse. Second, and oriented to applications on realistic sky simulations, use slightly more complex architectures trained with a larger variety of conditions, such as different CMB and noise intensities or even varied angular power spectra. A more robust model would probably be built without sacrificing a large part of its performance.

Chapter 4

Detection in realistic scenarios

The positive results we obtained in Chapter 3 indicate that CNNs are an effective approach for the detection of point sources in the microwave sky, at least in ideal conditions. In reality, however, the Galactic foregrounds and the anisotropic Planck instrumental noise conform a far more challenging scenario for this problem. In this chapter, we analyse the behaviour of our models in realistic conditions and suggest new directions, striving for a performance which is comparable to the methods which are currently used for detection.

The chapter is structured as follows. First, we analyse the performance of the previous CNN and the real space matched filter in maps with foregrounds. Second, we train and compare two different CNN architectures and evaluate the performance of the best of them at different training times. Third, we divide the sky in two regions attending to the foregrounds contamination, Galactic and extragalactic. We aim for a better performance of a CNN specialised in those regions, especially in the extragalactic one where the foregrounds are less intense. Finally, we obtain a preliminary result of the application of our models to the Planck data.

Our simulations consist of CMB, point sources and Galactic foregrounds at a FWHM of 7.22 arcmin (except for Section 4.1), which is the Planck Gaussian beam resolution for the 143 GHz channel (Table 1.1). Besides, we add an anisotropic instrumental noise component, as described in Section 2.1.

4.1 Performance of previous methods

Before training a new model in realistic maps with foregrounds, we study the performance of the CNN we trained in Chapter 3 (as defined in Table 3.1), while keeping the thresholds that offered a 95% and 99% reliability in Table 3.3. Besides, we also test the behaviour of the matched filter using a similar scheme. The simulations we use, in contrast to the ones in the following sections, are carried out at a map resolution (FWHM) of 5 arcmin.

As we see in Figure 4.1, where we show the results of the comparison, the behaviour of the CNN is apparently very good concerning only the completeness. Indeed, it maintains a very similar performance at low fluxes as in Section 3 and reaches almost 1 beyond 0.100 Jy. Nevertheless, the reliability decreases dramatically (46% and 54% for the lowest and highest threshold, respectively), becoming completely unacceptable for a detection algorithm. In the case of the real space matched filter, we also observe that the completeness is lower and converges at around a 90% for both thresholds, in addition to an important increase of the number of spurious detections.

When analysing the outputs of both algorithms in detail, they behave similarly as in the previous section near the Galactic poles, but start to perform inadequately when approaching

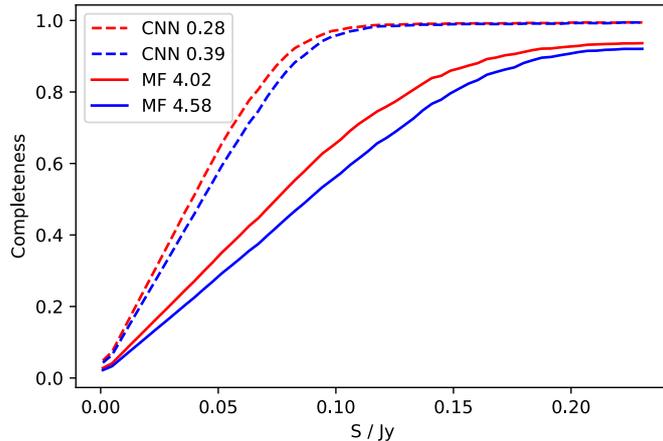


Figure 4.1: Completeness curves $C(S)$ obtained by the evaluation of the CNN defined and trained in Chapter 3 (dashed line) and the real space matched filter (solid line). The reliability at the studied thresholds $R(T)$ is the following. CNN: $R(0.28) = 0.46$, $R(0.39) = 0.54$. MF: $R(4.02) = 0.44$, $R(4.58) = 0.52$.

the equatorial plane. In regions dominated by foregrounds, the values of the output of both algorithms is much higher than usual, which also causes malfunctioning in the object extraction algorithm described in Section 2.4. In summary, neither the CNN trained without foregrounds nor the real space matched filter are valid algorithms in the sense that they provide a large number of spurious detections.

In Figure 4.2 we show the Mollweide projection of the all-sky map of detections of the CNN at the lowest threshold in Figure 4.1, $T = 0.28$. In contrast to the isotropy that we had in Chapter 3, in this case both the spurious (red points) and the undetected sources above a large flux (yellow points) are highly concentrated in the strong foreground regions, near the Galactic plane.

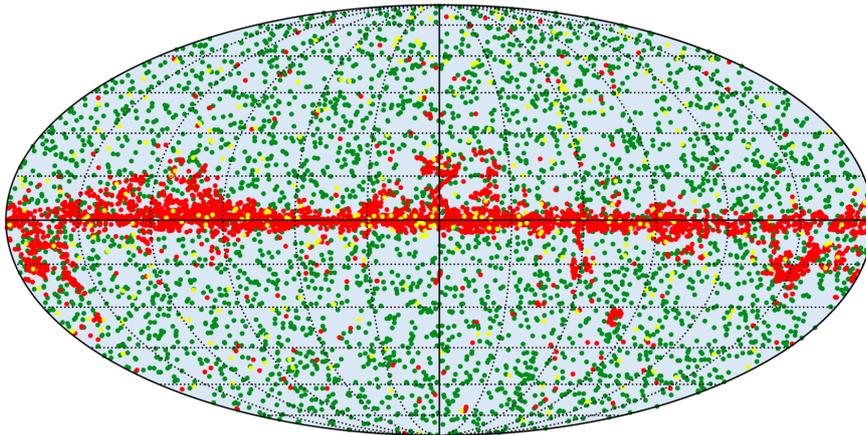


Figure 4.2: Mollweide all-sky projection including the detections of the CNN at threshold $T = 0.28$, trained as described in Chapter 3, on simulations with foregrounds at 5 arcmin. Green points are correct detections, red points are spurious detections and yellow points are undetected sources with flux $S > 0.080$ Jy.

In principle, the reasons behind the low reliability of both methods should be different. Whereas for the CNN the false detection could be associated to an inadequate training, in regions with large foreground emission, in the case of the MF this is related to high-emission compact objects that pass through the threshold limit. In this sense, the CNN still have

chances to improve its performance, whereas the MF is conditional to these objects. We will see whether this is possible in the next section.

4.2 CNN characteristics and evaluation

In the light of the previous results, it is required to train the model with realistic simulations to obtain a good performance in realistic scenarios. As the situation now is more challenging than working with the simulations in Chapter 3, we suggest to test a more complex architecture. We built a new CNN which is very similar to the one defined in Table 3.1, but using a larger filter size in the convolution and transpose convolution layers. The differences among both architectures are seen in Table 4.1. With the increase in the number of trainable parameters, we aim to obtain a better performance than with the previous (old) CNN.

No	Layer Type	k	Input size	Output Size	Filter Size	s_x	s_y
1	2D Convolution	16	150×150	150×150	7×7	1	1
4	2D Convolution	32	150×150	150×150	7×7	1	1
9	2D Transpose Conv.	64	75×75	150×150	5×5	2	2

Table 4.1: Differences in the architecture of the new CNN tested in this section and the CNN used in Chapter 3 (Table 3.1) for Gaussian noise and CMB simulations.

We trained both the old and the new CNN architectures with two all-sky simulations, in an identical procedure as in Chapter 3. Namely, we have used 5700 patches for training and 344 for validation, with a batch size of 50 and a learning rate set to 0.01. The labels corresponding to the inputs are defined as in Chapter 3 as well, these being binary images with ones at the position of the point sources and zeroes otherwise. We have trained for 12 epochs; the evolution of the loss and accuracy during training is seen in Figure 4.3 for both models. The plots show how both, the loss and the accuracy, converge rapidly after the first epochs, with almost identical values for both models. The training time is approximately of 18 minutes per epoch for the old CNN and 26 minutes for the new one.

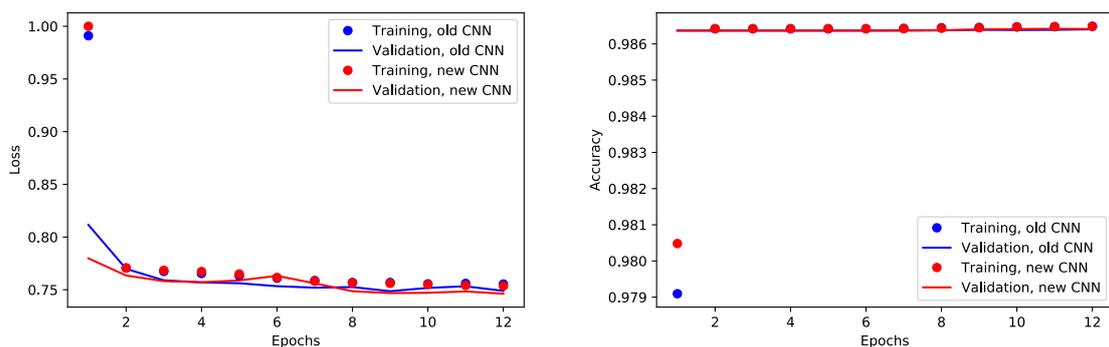


Figure 4.3: Evolution of the training and validation loss and accuracy of the CNN defined in Table 3.1 (blue) and of the new CNN defined in Table 4.1 (red), when both models are trained for 12 epochs with realistic simulations of the 143 GHz Planck channel. The loss is normalized to its highest value in epoch 1.

When studying the patches individually, the more evident difference we find on the behaviour of both CNNs is their output values. The new CNN generally associates lower values to its predictions, nevertheless, it is better at distinguishing point sources from other compact objects in the foregrounds or from statistical fluctuations. Such fact is also reflected in Figure

4.4, where we show the completeness curves for both CNNs at three different thresholds in each. These are chosen to achieve a reliability $R = 0.90, 0.95$ and 0.99 .

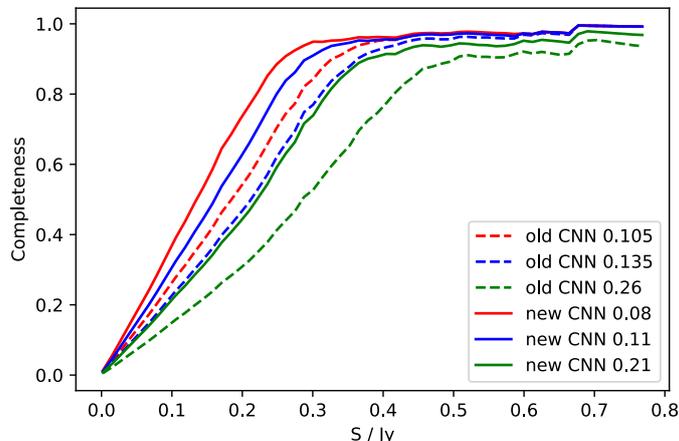


Figure 4.4: Completeness curves $C(S)$ obtained by the evaluation of the *old* CNN (Table 3.1, dashed line) and the *new* CNN (Table 4.1, solid line) on realistic simulations. Thresholds are chosen such that reliability factors $R = 0.90, 0.95, 0.99$ are obtained, for comparison purposes. Similar colours correspond to the same reliability.

In Figure 4.4 we see that the behaviour of both models is correct, reaching a completeness which is very close to 1 at large flux. However, the completeness of the new CNN is clearly above than the one of the old CNN at all similar reliability thresholds. We conclude that the new CNN is superior for our problem, and therefore we should retain its architecture in our study. In Table 4.2, we show the main parameters of the comparison of both CNNs.

<i>Old</i> CNN				<i>New</i> CNN			
T	S_{95} / Jy	S_{80} / Jy	R	T	S_{95} / Jy	S_{80} / Jy	R
0.105	0.413	0.284	0.900	0.08	0.358	0.221	0.897
0.135	0.455	0.312	0.953	0.11	0.409	0.253	0.950
0.26	-	0.411	0.988	0.21	0.671	0.320	0.990

Table 4.2: Summary of the parameters of the evaluation of both CNNs tested in this section. S_{95}, S_{80} are the flux at which a 95% and 80% completeness is obtained, respectively.

In contrast to the results we obtained in Chapter 3, the threshold has to be increased drastically to achieve a reliability of the 99%, dropping the completeness at the mid-range flux. This is due to strong spurious point sources, which are mostly produced by the foregrounds and are not easy to discriminate. In Figure 4.5 we show a projection of the sky indicating the location of the detections, spurious and undetected strong sources. The influence of the foregrounds in the spurious detections is clear, as almost all of them are particularly concentrated in mild to strong foreground-intense areas even for the lowest reliability we set, $R = 0.90$. We stand out the large improvement in performance compared to the model trained without foregrounds in Figure 4.1.

Notice that in the strongest foreground areas in Figure 4.5 the CNN is not able to detect at all, neither spurious nor real objects. In fact, there is a very high concentration of undetected sources above the limit flux $S = 0.250$ Jy. This is an interesting characteristic of the learning process; the strong foregrounds are a challenge for the model, so it protects itself from performing highly inaccurate predictions by not indicating any spot. The explanation

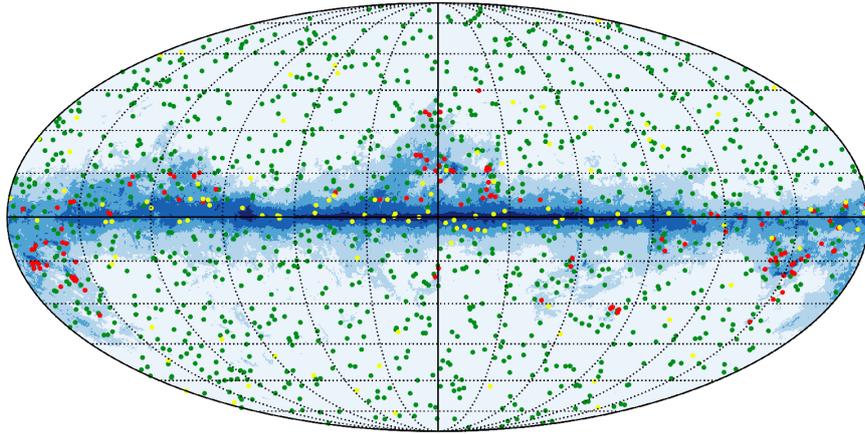


Figure 4.5: Mollweide all-sky projection including the detections of the *new* CNN (Table 4.1), on realistic simulations at threshold $T = 0.08$ ($R = 0.90$). Green points are correct detections, red points are spurious detections and yellow points are undetected sources with flux $S > 0.250$ Jy. The blue regions in the background represent the intensity of the foregrounds, where darker areas indicate high-intensity foreground zones.

for this behaviour resides in the definition of the loss function we use, which is the binary cross-entropy (equation 2.3). As we stated before, this function strongly penalizes incorrect predictions. The area covered by the point sources is much smaller than the background, then, the loss function is smaller if the prediction is a zero array than if it includes many spurious point sources. In general, the small values that the CNN assigns to the background and to faint point sources follow this argument too.

In Figure 4.6, we show the maximum value of the output of the CNN for each of the detections in Figure 4.5 (i.e., at threshold $T = 0.08$), against the temperature of the map in that point. The plot confirms the trend we discussed above, which is that the spurious point sources appear mostly in areas with strong Galactic foregrounds (high temperature values in the map).

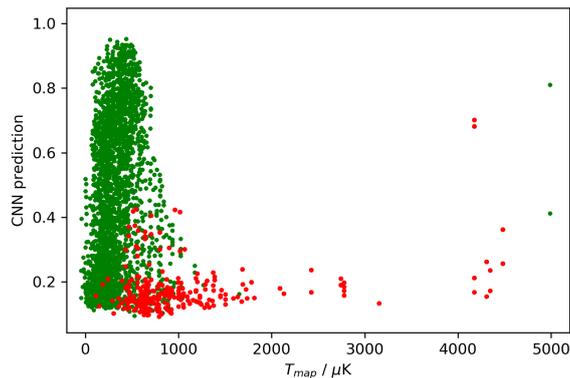


Figure 4.6: Scatter plot of the output value of the new CNN against the temperature of the map at the detection point (T_{map}), for every correct (green points) and spurious (red points) detections at threshold $T = 0.08$ in Figure 4.5.

4.2.1 Performance of the CNN at different training levels

The results both in the previous section and in Chapter 3 have been obtained with a CNN trained for 12 epochs. This choice is not arbitrary but rather justified by the performance of

the CNN at different training times. A qualitative comparison between a CNN with 8 and 12 epochs of training was done in Figure 3.3, where we saw that the performance at both stages is very similar.

In Figure 4.7, we present a quantitative analysis including the completeness curves for the new CNN (Table 4.1) after 4, 8 and 12 training epochs and at thresholds which achieve a 90% and 95% reliability. Notice that the irregular shape of the curves above a flux of 0.8 Jy is due to the reduced number of point sources at that flux. In this simulation in particular, one of the brightest sources was located on the Galactic plane, preventing the CNN from detecting it.

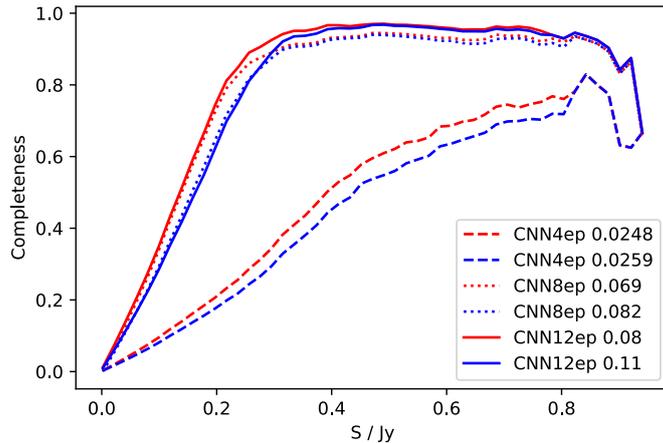


Figure 4.7: Completeness curves $C(S)$ obtained by the evaluation of the new CNN (4.1) trained for 4 (dashed), 8 (dotted) and 12 (solid) epochs on realistic simulations. Thresholds are chosen such that reliability factors $R = 0.90, 0.95$ are obtained, for comparison purposes. Similar colours correspond to the same reliability.

The analysis yields that a 4-epoch training is clearly insufficient, proving that small reductions in the loss function (recall its progression in Figure 4.3) can be translated to large differences in the predictions of the model. For a training of both 8 and 12 methods, the performance is very similar and almost makes no difference in the low and mid-range flux, although the 12-epoch model is slightly superior for strong sources near the Galactic plane, which justifies its usage. Finally, we remark the fact that the output values of the CNN tend to increase with the number of epochs. This effect may be counterproductive at a certain point, due to the fact that the spurious detections are also enhanced.

4.3 Division of the sky in two regions

According to the results of the training of the new CNN architecture, the Galactic foregrounds are preventing the CNN from improving its performance, not only in the Galactic plane but also in the extragalactic regions. Therefore, we carried out a division of the sky in two regions: an extragalactic region, which includes the sky which is less dominated by foregrounds, and a Galactic region. In each region, as a first approach we train a model with the architecture defined in Table 4.1. By doing this, we aim to obtain specialised models which address the following three issues:

- A unique model is not able to adapt its predictions to zones of the sky with very different characteristics.

- Almost no detections occur near the Galactic plane, neither true nor spurious.
- Most spurious sources appear in regions with a strong foreground component.

In addition, the procedure we follow is similar to the division in the PCCS2 catalogue that we described in Section 1.2. We developed and trained three different models:

- **85%-CNN**, aimed for detection in the 85% patches of the sky with weakest foregrounds. The model is trained on 2 full simulations of the 90% of the weakest-foreground sky, which include 2764 patches each. The model is trained for 12 epochs, using a batch size of 50 and a validation set of 150 images.
- **75%-CNN**, analogous to the 85%-CNN but in the 75% patches with weakest foregrounds. The model is trained on 2 full simulations of the 80% weakest-foreground sky, including 2457 patches each. We train for 14 epochs to compensate for the reduction of the training set.
- **Strong-CNN**, trained using the 30% brightest foreground patches of the sky and aimed for detecting in the Galactic region. The model is trained for 24 epochs and no validation set is used.

Notice that we include a larger area of the sky in the training set than the actual detection zone of the model. The reason is that we want to increase the robustness of the algorithm, providing the sufficient variation in the training set to have accurate detections also in the boundaries of the detection zones. This training procedure should be further tested and compared to training only in the application area.

The 85%-CNN and, especially, the 75%-CNN, present an outstanding performance in the extragalactic regions, which reminds to the results we obtained in Chapter 3. The completeness curves for thresholds at $R = 0.80, 0.90, 0.95$ and 0.98 are presented in Figure 4.8. We see that a total completeness is rapidly reached for all the thresholds we set. In addition, the completeness in Figure 4.8 rises much faster than in the case of the previous all-sky architectures, for instance in Figure 4.4. This is partially because the detections are now being performed only in weak foreground areas, but mainly because both models offer a much better performance since they have been constructed for specific regions. In Table 4.3, we present the summary of the performance indicators for both models at the four reliability goals we set.

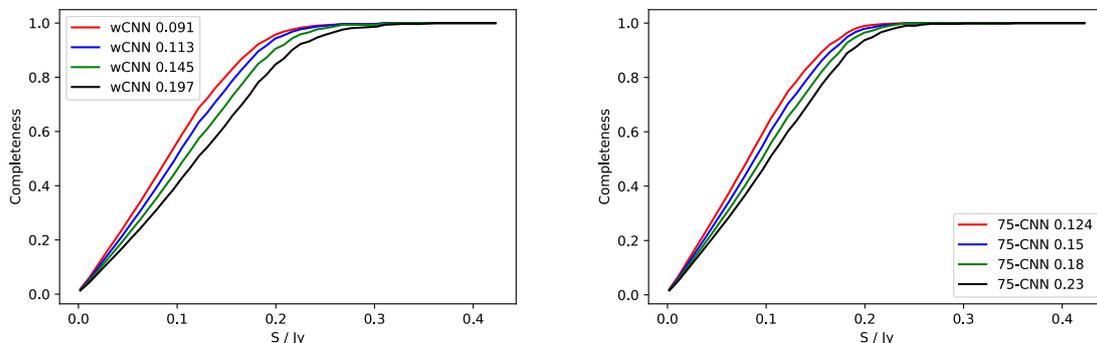


Figure 4.8: Evaluation of the CNNs for the 85% (left) and 75% (right) weakest foreground areas (85%-CNN and 75%-CNN models) in their corresponding regions. The completeness curves $C(S)$ are obtained by the evaluation at thresholds $T = 0.091, 0.113, 0.145, 0.197$ (left) and $T = 0.124, 0.150, 0.180, 0.230$ (right), whose corresponding reliability is respectively $R = 0.80, 0.90, 0.95, 0.98$ in both cases.

CNN 85%					CNN 75%				
T	S_{95}/Jy	S_{90}/Jy	S_{80}/Jy	R	T	S_{95}/Jy	S_{90}/Jy	S_{80}/Jy	R
0.091	0.197	0.175	0.149	0.800	0.124	0.180	0.157	0.132	0.802
0.113	0.207	0.182	0.160	0.899	0.150	0.185	0.167	0.143	0.899
0.145	0.223	0.199	0.170	0.950	0.180	0.194	0.175	0.152	0.950
0.197	0.248	0.214	0.187	0.980	0.230	0.212	0.186	0.162	0.978

Table 4.3: Summary of the parameters of the evaluation of both CNNs tested in this section. S_{95} , S_{90} and S_{80} are the flux at which a 95%, 90% and 80% completeness is obtained, respectively.

The results show that the 75%-CNN is better than the 85%-CNN by a 10-15% in terms of flux at which a certain completeness is achieved for all reliabilities. This variation cannot be explained only by the removal of some challenging areas of the detection in the case of the 75%-CNN. Since it performs indeed better in all the extragalactic region, the different training set is playing an important role. In Figures 4.9 and 4.10, we show the all-sky detection maps of both models. It is clear that, in contrast to previous detection maps where the spurious and the undetected sources were concentrated in the strongest foreground areas, these appear now at many other locations in the sphere. We also see a certain degree of clustering of the spurious detections, which indicates that there are regions where even though the foregrounds are not very strong, they have a certain structure which is able to fool the CNN. Moreover, the clustering appears in the same areas for both CNNs.

We recall that the matched filter was discarded due to its poor performance in the Galactic region, however, it could be applied in extragalactic regions and compared again to these specialised models.

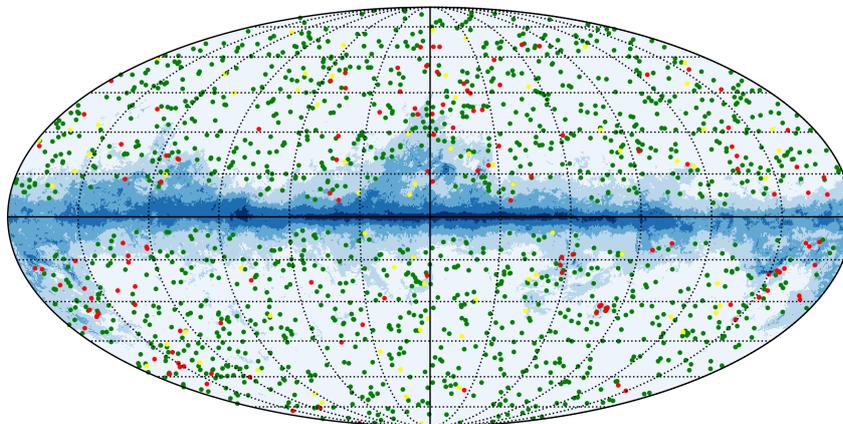


Figure 4.9: Mollweide all-sky projection including the detections of the 85%-CNN, on realistic simulations at threshold $T = 0.113$ ($R = 0.90$). As in previous figures, green points are correct detections, red points are spurious detections and yellow points are undetected sources with flux $S > 0.180$ Jy.

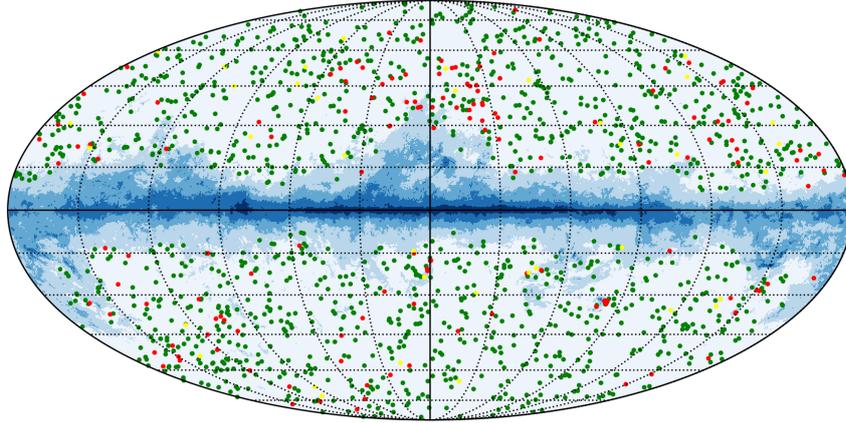


Figure 4.10: Mollweide all-sky projection including the detections of the 75%-CNN, on realistic simulations at threshold $T = 0.15$ ($R = 0.90$). As in previous figures, green points are correct detections, red points are spurious detections and yellow points are undetected sources with flux $S > 0.180$ Jy.

In Figure 4.11 we present the results of the evaluation of the Strong-CNN model on the 25% brightest sky, where one can clearly see that the model for strong foregrounds is not an accurate method for detections in the Galactic region. Its performance is very limited, achieving a best-case completeness of approximately 0.8 at high fluxes, but introducing a very large number of spurious sources at any threshold. In the right plot of Figure 4.11, we see a similar behaviour as in Figure 4.6; true detections are performed in the weakest foreground regions whereas spurious sources occur mostly in foreground-dominated areas. Here, obtaining an unsatisfactory performance is also common in other detection methods due to the heterogeneity of the foregrounds. The reasons of this performance will be discussed later, together with suggestions for its improvement.

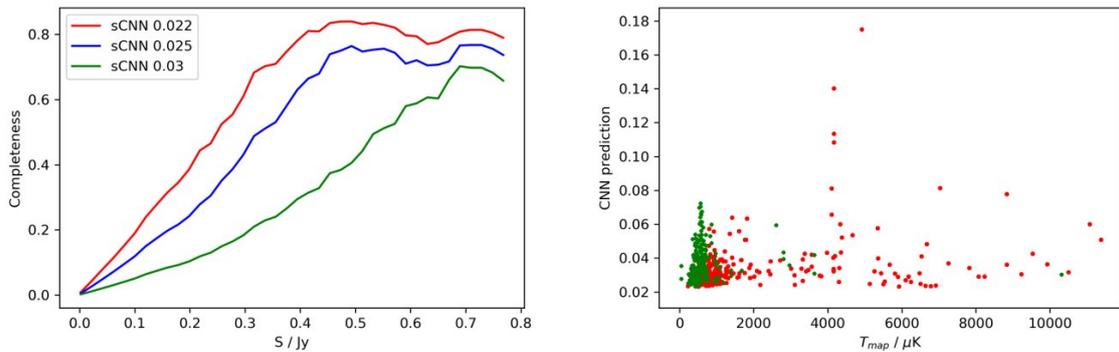


Figure 4.11: Evaluation of the Strong-CNN on the 25% strongest foreground area. Left, completeness curves $C(S)$ obtained by the evaluation at thresholds $T = 0.022, 0.025, 0.03$, whose corresponding reliability is respectively $R = 0.36, 0.55, 0.69$. Right, distribution of true and spurious detections against map temperature at $T = 0.025$.

We conclude that the division of the sky in zones is clearly a positive step towards the improvement of the quality of our detection pipeline and the production of accurate CNN models, even if the approach for the Galactic region should be modified.

4.4 Application to the Planck data

Finally, we apply the successful models we developed in the previous section (extragalactic CNNs in the 75% and 85% less contaminated regions of the sky) to the actual Planck mission data. Now, we completely lack of information on which of the detections of our algorithm are real sources and which ones are not, since we no longer generate the map ourselves. Hence, we define the reliability of our detections according to the results obtained in the previous study; as our simulations are realistic, we make the assumption that the reliability given by the thresholds in Table 4.3 translates into the same reliability for a real map.

We present the map of the detections in Figures 4.12 and 4.13 for the 85%-CNN and the 75%-CNN, respectively and in their corresponding regions. We have not applied the CNNs for the strong foreground areas, as they have not yielded a positive result in simulations. As expected, both figures show very similar results and most of the detections appear in both, fact easily in some point-source clustering regions. Even if this structure is the same, we appreciate a larger density of detections in the map of the 75%-CNN, also expected due to its better completeness.

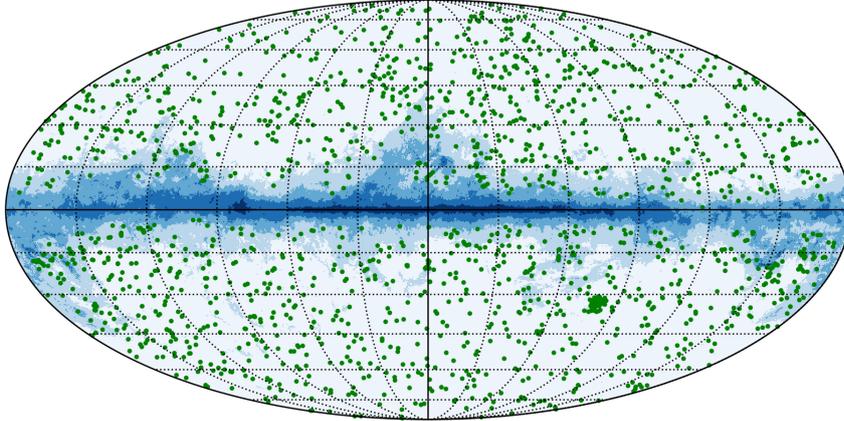


Figure 4.12: Mollweide all-sky projection including the detections of the 85%-CNN, on real Planck data at $T = 0.091$ ($R = 0.80$). Green points indicate detections. The total number of detections is $N_{\text{total}} = 1764$.

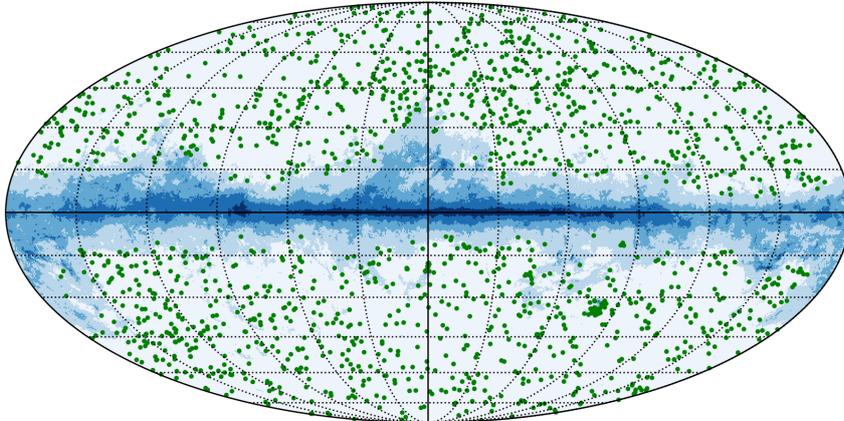


Figure 4.13: Mollweide all-sky projection including the detections of the 75%-CNN on real Planck data at $T = 0.124$ ($R = 0.80$). Green points indicate detections. The total number of detections is $N_{\text{total}} = 1682$.

The summary of our results, including the number of detections for each model at the four different reliability levels, is presented in Table 4.4. If we compare the flux at a 90% completeness (S_{90}) of the Planck Catalogues of Point Sources (see Table 1.2) and our results for realistic simulations in Table 4.3, we see that we should achieve very similar results with our 85%-CNN ($S_{90} = 0.175$ Jy for $R = 0.80$) than the PCCS2 in the extragalactic area ($S_{90} = 0.177$ Jy for $R = 0.80$), which also removes the 15% brightest regions of the sky for the 143 GHz channel.

However, our results in Table 4.4, are slightly worse and detect approximately 20% less point sources in the case of the PCCS2, which includes 2160 detections at 80% reliability. The results for the 75%-CNN are more difficult to compare, but still below than what the completeness curves in simulations would indicate. In the discussion, we present various hypothesis for this behaviour and several directions for improvement.

We remark that the results we obtain in this section are preliminary and should only be regarded as a proof of the suitability of our approach for the detection of point sources in the Planck sky. There are many aspects that can be improved and others that require further testing, as we will discuss later.

R	85%-CNN		75%-CNN	
	T	N_{total}	T	N_{total}
0.80	0.091	1764	0.124	1682
0.90	0.113	1398	0.150	1326
0.95	0.145	1138	0.180	1131
0.98	0.197	952	0.230	990

Table 4.4: Preliminary results of the application of the 85%-CNN and the 75%-CNN to the Planck data. The thresholds T correspond to the reliabilities $R = 0.80, 0.90, 0.95, 0.98$ of each of the rows in Table 4.3. N_{total} is the total number of detections in the indicated region.

4.5 Discussion

The scope of the results we present in this chapter is very wide, so their discussion is divided in two blocks. The first part analyses the results themselves, in addition to the behaviour and performance of the different models we present. The second part focuses on the application of our methods to real data and discusses possible improvements. A general overview, conclusions and future work directions are presented in Chapter 5.

Method performance

In Section 4.1, we clearly saw that specific simulations are required to achieve a good performance under specific conditions. The robustness we highlighted in Chapter 3 vanishes when there is an important change in the conditions of the input, such as the introduction of Galactic foregrounds in the map. Even if this is an expected result, it clearly is a drawback of using CNNs that should be extrapolated to other scenarios; one must make sure that the training set faithfully represents reality if a reliable model is to be produced.

When we trained our CNNs using maps with foregrounds in Section 4.2, we suggested a new architecture, which actually presented a much better performance, to compare to the previous one. We tried many architectures until finding one which was suitable for our problem in Chapter 3, and from there, we introduced a higher degree of complexity for realistic simulations. However, it is not possible to ensure that the model we suggest (Table 4.1) is the best improvement that can be made, despite having explored several options. Besides,

we used the same architecture in Section 4.3, when the sky was divided in two areas and we achieved our best performance in the extragalactic region, but probably more complex networks are required for the Galactic one.

In general, many different neural networks may present a good performance for a particular problem, but it is often possible to find a better model. This property makes CNNs a very powerful tool, but also a tedious one, since there is always a chance that a change in the model or in the training parameters improves or deteriorates the results. Regarding our problem, this means that surely there is a CNN which is able to perform better than any of the architectures or models we have trained, even if that improvement does not involve a significant qualitative change. Finding a model which achieves a good result is therefore a very promising indicator that those results can be further improved.

From the results of the division of the sky in two areas in Section 4.3 we can extract several conclusions. The first one is that strong foregrounds were *distracting* the CNNs we trained in Section 4.2. Including such parts of the sky in the training set not only affects the results on those areas, but also in other regions. We can see this clearly by comparing the completeness curves of the specialised 85%- or 75%-CNN (Figure 4.8) to the ones from the all-sky CNNs in Figure 4.4. If the latter were detecting with the same accuracy as the specialised ones outside the strong foreground region, as the point sources are randomly distributed across the sky, there would not be such a difference in the completeness, even when comparing thresholds for different reliability.

Another point is that we managed to fix the first of the three issues we intended to solve with the division of the sky (the adaptability of the models), but we did not solve the lack of detections near the Galactic plane, and only partially addressed the problem of the spurious sources concentrating in strong foreground areas. Regarding the detections in the Galactic plane, a possible reason for this failure is that the training set is not sufficiently extensive or that the training epochs are not enough. Moreover, there exist compact objects in the galaxy which are point sources from the point of view of the image that we have not included in our labels, confusing the training of the model. Clearly, the Galactic plane and its surroundings conform the area which presents a greater challenge and further research should be done, testing more complex architectures as well. Despite this, for performing analysis of the CMB, this region is commonly masked, so the importance of detecting point sources is much greater in the extragalactic region.

We say that the concentration of the spurious sources in foreground-dominated areas, that we saw clearly in Figure 4.5, is only partially addressed because of the clustering issue we mentioned above. Spurious sources appearing with our extragalactic models tend to concentrate in regions of the sky where foregrounds are not necessarily intense, but able to induce spurious detections. This is an important issue, because in case of detecting accurately in those regions, we would greatly increase the reliability of the model. A straightforward way to proceed would be either the training of specific models for each of those areas or the introduction of a larger number of patches from those areas in the training set. We are confident that any of the two approaches, properly implemented, will improve our results, so we plan to do this for future work.

Finally, in regard to the comparison between the two extragalactic models themselves, we see that the performance of the 75%-CNN is clearly superior, probably due to its higher level of specialization. We recall that the 85%-CNN has been trained with the 90% brightest patches of the sky. As there is already an important Galactic emission in several of those, the *distraction* effect we discussed above had an impact on this model. Presumably an even greater reduction of the covered area, such as a 60%-CNN, would provide better results; we also leave this open for future work.

Application to Planck data

The first reason why our results for real data do not correspond to what we expect may be the level of realism of our simulations. It is likely that we are not able to fully represent the reality in them, particularly in aspects such as the fine structure of the foregrounds (at small angular scales), the beam of the instrument, that we have represented as Gaussian, and the spatial distribution of point sources in the sky. If our simulations were more realistic, the completeness curves and reliability thresholds would be more similar to reality. Due to this effect, we might be setting thresholds which achieve a certain reliability in simulations and increase it in reality. For instance, it could be that the threshold $T = 0.124$ on Table 4.3, which yields $R = 0.80$ in simulations, actually offered a reliability of 0.90 or 0.95 in the real maps (but also lower, such as 0.70, which would have a negative effect). The main issue is that we have not found a way to evaluate this variability for real data, since we do not know where the sources are.

On the other hand, the completeness of our CNNs can be measured in real conditions, what would give us an idea of the accuracy of the simulations. The procedure is very simple: we create a set of point sources, add it to the real data, and then apply the detection algorithm to measure the completeness at each flux. This is one of the directions we will work on next to aim for a better understanding of our results. The above procedure is used for both evaluating the reliability and the completeness of the PCCS and PCCS2 catalogues. A different definition of reliability is therefore used, which may also contribute to the difference in the number of detections with respect to our models.

Apart from the possible issues with the simulations, there is a bundle of possibilities to try to improve our results, both in simulations and in real data. The first of them is to divide the sky in more regions and to train separate models for each of them, which can also help towards the clustering issue. Even if we had different completeness in the different areas, we would surely be able to build a larger source catalogue. Besides, as we discussed in Chapter 3, training using very realistic simulations of the whole sky is perhaps not the optimal way to proceed. An interesting approach would be to reinforce the learning in the areas where the CNN struggles and to increase the number of point sources in the mid-flux range.

One of our priorities in the continuation of our work is to increase the size of the patches, so that each patch covers a larger area of the sky. We believe that, as we have a safety margin for detections of 5 pixels at each side, there are some regions of the sky (particularly near the galactic poles where the deformation of the HEALPix pixels is larger) where point sources are always undetected or ignored. This issue, that we realised late in the project, might be responsible for the loss of a few percentage of the total amount of point sources in the real Planck maps.

Finally, notice that the results of the 85%-CNN are far from the number of detections of the catalogue, but the detections of the 75% are much closer. In a 13% larger portion of the sky, the difference between the 85%-CNN and the 75%-CNN is of less than a 5% in absolute number of detections, which is a large difference considering that the concentration of point sources near the Galactic plane is large (as seen in the PCCS2 map in [24] and also in Figure 4.12) and many point sources are located between the 75% and the 85% regions. Therefore, we have strong reasons to believe that smaller refinements over the 75%-CNN (or over further specialised model) and a detailed study of the completeness in real conditions will lead to the detection of many new point sources.

Chapter 5

Conclusions and future work

Throughout this work, we developed several Deep Learning-based models for the detection of point sources in the Cosmic Microwave Background. The results we obtained are very positive, both in realistic and ideal (without Galactic foregrounds) simulations, proving that Convolutional Neural Networks are an adequate, reliable and accurate approach for this problem. Our results are indeed scientifically relevant, as the approach that we suggest for the problem is original.

The good performance of our model in simulations of the CMB and Gaussian instrumental noise, particularly in comparison to an implementation of a matched filter in two dimensions, is a valuable result by itself. In fact, it proves that the current detection methods can be improved using CNNs. It also suggests that a combination of a CNN for detection and a matched filter (or other methods) for the estimation of the flux of the sources would yield very interesting results.

Besides, the performance of the different models we suggest for detection on realistic simulations of the Planck 143 GHz channel supports the previous idea. The number of detections in the preliminary results we obtain with their application to the Planck data is very close to the current point source catalogues, and we show that there is a large room for improving them.

Our work may also have implications for cosmology in general since CNNs are a versatile method which can be trained using simulations. In addition, we show that relatively deep and complex CNNs are able to process a HEALPix map at $N_{\text{side}} = 2048$ in less than a few minutes. Therefore, we encourage the application of CNNs to other cosmological problems such as inference of cosmological parameters, map delensing or the analysis of the CMB polarisation, where Deep Learning has not been tested yet (or only at very early stages). The fast growth of the Machine Learning methodologies themselves in recent years also opens the door for new learning-based insights.

In addition to its positive aspects, our approach also presents a series of drawbacks. Perhaps the most important one is proving the correctness of the predictions of a model in real data, since one must rely on the realism of their simulations. Furthermore, proving that a certain model, architecture or training approach is optimal is very hard, and so is achieving an adequate architecture.

Regarding the future development of this work, and additionally to the improvements and refinements we suggested in previous chapters, there are several directions of interest. The first is extending our methodology to the nine Planck channels, after tuning its performance for the current one. As each channel presents different characteristics and foreground intensities, we expect to obtain compelling conclusions on the behaviour of CNN-based models for the detection of compact objects in images.

Finally, we believe that our approach can be extended to the detection of point sources in polarisation maps of the CMB, suggesting a different path than previous work such as [6], based in filters. The analysis of the polarisation maps of the CMB is a hot research topic nowadays and such a contribution would also have relevant scientific implications.

In conclusion, the results we obtained are promising and very valuable, but there is a long road ahead. We aim to keep on our work in the suggested directions. Hence, further results should be expected in the near future, including a publication in early 2020.

Bibliography

- [1] ABADI, M., ET AL. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Software available from [tensorflow.org](https://www.tensorflow.org).
- [2] BARBARY, K. SEP: Source Extractor as a library. *The Journal of Open Source Software* 1, 6 (2016), 58.
- [3] BERTIN, E., AND ARNOUITS, S. SExtractor: Software for source extraction. *Astronomy and Astrophysics Supplement* 117 (1996), 393–404.
- [4] CHOLLET, F., ET AL. Keras. <https://keras.io>, 2015.
- [5] DELABROUILLE, J., ET AL. The pre-launch Planck Sky Model: a model of sky emission at submillimetre to centimetre wavelengths. *Astronomy & Astrophysics* 553 (2013), A96.
- [6] DIEGO PALAZUELOS, P. *Two Approaches to Modern Observational Cosmology: Detection of Point-like Sources in Polarization Maps of the Cosmic Microwave Background, and Estimation of Dark Matter Distribution in Galaxy Clusters*. Bachelor Thesis, Universidad de Cantabria, 2018.
- [7] DURRER, R. *The cosmic microwave background*, vol. 401. Cambridge University Press Cambridge, 2008.
- [8] GOODFELLOW, I., BENGIO, Y., AND COURVILLE, A. *Deep Learning*. MIT Press, 2016. Available at <http://www.deeplearningbook.org>.
- [9] GÓRSKI, K. M., ET AL. HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere. *The Astrophysical Journal* 622, 2 (2005).
- [10] HOBSON, M., JONES, A., LASENBY, A., AND BOUCHET, F. Foreground separation methods for satellite observations of the cosmic microwave background. *Monthly Notices of the Royal Astronomical Society* 300, 1 (1998), 1–29.
- [11] HU, W., AND DODELSON, S. Cosmic microwave background anisotropies. *Annual Review of Astronomy and Astrophysics* 40, 1 (2002), 171–216.
- [12] KINGMA, D. P., AND BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). Available at <https://arxiv.org/abs/1412.6980>.
- [13] LANZ OCA, L. F. *Detection of extragalactic sources in cosmic microwave background maps with multifrequency techniques*. PhD thesis, Universidad de Cantabria, 2016.
- [14] LECUN, Y., BOTTOU, L., BENGIO, Y., HAFFNER, P., ET AL. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324.

- [15] LONG, J., SHELFHAMER, E., AND DARRELL, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 3431–3440.
- [16] LÓPEZ-CANIEGO, M., HERRANZ, D., GONZÁLEZ-NUEVO, J., SANZ, J., BARREIRO, R., VIELVA, P., ARGÜESO, F., AND TOFFOLATTI, L. Comparison of filters for the detection of point sources in Planck simulations. *Monthly Notices of the Royal Astronomical Society* 370, 4 (2006), 2047–2063.
- [17] LÓPEZ-CANIEGO, M. *Detection of Point Sources in Maps of the Cosmic Microwave Background Radiation by means of Optimal Filters*. PhD thesis, Universidad de Cantabria, 2006.
- [18] MILLETARI, F., NAVAB, N., AND AHMADI, S.-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)* (2016), IEEE, pp. 565–571.
- [19] NIELSEN, M. A. *Neural Networks and Deep Learning*. Determination Press, 2005.
- [20] PERRAUDIN, N., DEFFERRARD, M., KACPRZAK, T., AND SGIER, R. DeepSphere: Efficient spherical convolutional neural network with HEALPix sampling for cosmological applications. *Astronomy and Computing* 27 (2019), 130–146.
- [21] RONNEBERGER, O., FISCHER, P., AND BROX, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (2015), Springer, pp. 234–241.
- [22] RUSSELL, S. J., AND NORVIG, P. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.
- [23] THE PLANCK COLLABORATION. Planck 2013 results. XXVIII. The Planck Catalogue of Compact Sources. *Astronomy & Astrophysics* 571 (2014), A28.
- [24] THE PLANCK COLLABORATION. Planck 2015 results. XXVI. The Second Planck Catalogue of Compact Sources. *Astronomy & Astrophysics* 594 (2016), A26.
- [25] THE PLANCK COLLABORATION. Planck 2018 results. I. Overview and the cosmological legacy of Planck. *arXiv preprint arXiv:1807.06205* (2018).