



THE UNIVERSITY
OF QUEENSLAND
AUSTRALIA

HONOURS THESIS

Deep Learning for the Spectral Classification of Transient Astronomical Objects

Author:

Daniel MUTHUKRISHNA

Supervisors:

Dr. David PARKINSON

Dr. Brad TUCKER

*A thesis submitted in partial fulfillment of the requirements
for the degree of the Bachelor of Engineering
in the discipline of Electrical and Aerospace Engineering*

November 25, 2016

Daniel Muthukrishna
daniel.muthukrishna@uqconnect.edu.au
49 Foxmont Drive
Carina, QLD 4152

November 25, 2016

Prof Michael Brünig
Head of School
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia, QLD 4072

Dear Professor Brünig,

In accordance with the requirements of the Degree of Bachelor of Engineering in the School of Information Technology and Electrical Engineering, I submit the following thesis entitled:

"Deep Learning for the Spectral Classification of Transient Astronomical Objects"

The thesis was performed under the supervision of Dr David Parkinson. I declare that the work submitted in the thesis is my own, except as acknowledged in the text and footnotes, and that it has not previously been submitted for a degree at the University of Queensland or any other institution.

Yours sincerely,

Daniel Muthukrishna

Acknowledgments

Firstly, I would like to thank my primary supervisor, Dr David Parkinson of the University of Queensland, for his continuous help, assistance, and guidance throughout this thesis. I would also like to thank my second supervisor, Dr Brad Tucker of the Australian National University, for his regular contact over email and video calls that guided the goals and direction of the project. I am indebted to both David and Brad for their generous support which expertly went above and beyond in responding to all of my questions over the course of the year.

I am also very grateful to Mr Samuel Hinton, Prof Tamara Davis, and Dr Chris Lidman for their expertise and suggestions during this thesis. Their support has provided invaluable direction in the outcomes of the project. Finally, I'd like to thank several other members of the OzDES community including Dr Alex Kim, Dr Richard Kessler, Dr Jeffrey Silverman, Dr Anais Moller, Prof Mark Sullivan, Dr Michael Childress, and Prof Bob Nichol for their feedback and suggestions on the front-end requirements of the project.

Abstract

This thesis details the creation of a novel supernova spectral classification tool, DASH, that has been developed primarily for the OzDES collaboration as a replacement to current classification tools. The main aim was to improve upon the speed and ease of classification while not compromising accuracy. DASH has used a completely new approach that does not rely on iterative template matching techniques like all previous software, but instead classifies based on the features of each supernova type and age bin. It has achieved this by employing a deep neural network to train a matching algorithm. This has enabled DASH to be over 100 times faster than Superfit while also being just as accurate. This has been tested using the latest OzDES ATEL data, where DASH has accurately classified each spectrum with a higher degree of certainty.

The deep learning model was developed using Tensorflow, and has involved defining 306 different classification bins made up of 17 supernova subtypes and 18 age bins. The model was trained using the Obelix Supercomputer at UQ and made use of nearly 4000 supernova templates from SNID and the Berkeley SN Ia Program. The trained model is independent of the number of templates, which allows for DASH's unprecedented speed. Two user interfaces available on GitHub and PyPI have been developed. These include a graphical interface for easy visual classification and analysis of supernovae, and a python library for the autonomous and quick classification of several supernova spectra.

The speed, accuracy, user-friendliness, and versatility of DASH presents an advancement to existing spectral classification tools, and is a viable alternative for the astronomy community.

Contents

Acknowledgments	v
Abstract	vii
List of Figures	xv
List of Tables	xix
1 Introduction	1
1.1 Motivation	1
1.2 Scope	2
1.2.1 Wider Context	3
2 Background	5
2.1 Astronomical Objects	5
2.1.1 Flux, Luminosity, and Magnitude System	5
2.1.2 Supernovae	6
Types	6
Age	8
2.1.3 Galaxies	8
2.2 Astronomical Spectra	9
2.2.1 Spectroscopic Features	10
2.2.2 Redshift	10
2.2.3 Distortions	11
Interstellar Dust	12

Sky lines	13
Noise	13
2.2.4 Equipment Miscalibration	14
2.3 OzDES	14
2.3.1 Anglo-Australian Telescope	15
3 Prior Methods	17
3.1 Current OzDES Approach	17
3.2 Prior Software	17
3.2.1 SNID	18
3.2.2 MARZ	18
3.2.3 AUTOZ	18
3.2.4 Superfit	19
3.2.5 Summary of Previous Tools	19
4 Requirements and Tool Decisions	21
4.1 Speed and Accuracy	21
4.2 Installation and Updating	22
4.3 Operating System	22
4.4 Online vs Offline	23
4.5 Language Decision	24
4.6 Integration	26
4.7 Libraries and Tools Used	26
4.7.1 Libraries	26
4.7.2 Software Tools	27
Git and GitHub	27
Resources	27
5 Data Description	29
5.1 Data Collection	29
5.1.1 SNID Templates	29

Templates-2.0	30
5.1.2 Superfit	30
5.1.3 Increased SNID Templates	31
Liu & Modjaz	31
BSNIP	31
5.1.4 WISeREP	32
5.2 Deleting Templates	33
5.2.1 Unknown Ages	33
5.2.2 Ic-broad bias	33
5.3 Need for more templates	34
5.3.1 Bias in all Supernova Classification Tools	34
5.4 Templates Description	34
5.4.1 Types and Ages	35
5.4.2 Number of Templates	36
Accounting for SNIa Bias	36
5.5 Plot of Templates	37
5.6 File Types	38
5.6.1 Inw File	38
5.6.2 ASCII, data, Two-column Text Files	40
5.6.3 FITS File	40
6 Initial Design and Major Decisions	43
6.1 Initial Design	43
6.1.1 Host Galaxy Subtraction	45
6.1.2 Initial Classification Method	45
6.2 Changing Classification Method	46
6.3 Deep Learning	48
6.3.1 Overview	48
6.3.2 Machine Learning in Astronomy	49
6.3.3 Tensorflow	50

7	Implementation	51
7.1	Processing Method	51
7.1.1	Filtering	52
7.1.2	Normalising and De-redshifting	53
7.1.3	Log-wavelength Binning	54
7.1.4	Continuum Modelling with Spline Interpolation	55
7.1.5	Cosine tapering edges	56
7.2	Preparing Training Data	57
7.2.1	Label and Image Data	58
7.2.2	Template sample bias	58
7.3	Deep Learning Training	59
7.3.1	Softmax Regressions	59
7.3.2	Building Layers	61
7.4	Trained Models	61
7.4.1	Trained at zero redshift	61
7.4.2	Agnostic Redshift	63
7.5	Interfaces	64
7.5.1	Graphical User Interface	64
	Label Guide:	66
7.5.2	DASH Python Library	69
8	Performance	71
8.1	Validation Set	71
8.1.1	Alternative Models	73
8.2	Results with OzDES data	74
9	Project Evaluation	79
9.1	Completion	79
9.2	Comparison to Current Methods	80
9.2.1	Speed	80

9.2.2	More specific classification	81
9.2.3	Accuracy	81
9.2.4	Installation and Ease of Use	82
9.3	Use in Astronomy Community	82
9.4	Disadvantages of DASH	83
9.5	Future Improvements	83
9.5.1	Redshifting	83
9.5.2	Identify Host Galaxy	83
9.5.3	False Positives	84
9.5.4	Python Library Functions	84
9.5.5	GUI Features	85
10	Conclusion	87
	Bibliography	89

List of Figures

2.1	The different supernovae types are classified based on the presence or absence of certain chemical features in their spectrum. [42]	7
2.2	Example of the bolometric luminosity of a supernovae over time. The time at peak luminosity corresponds to an age of zero days. [10]	9
2.3	Spectrum of the Type-1a supernova '2002bo' at zero redshift. each of the peaks and troughs correspond to the absorption and emission lines caused by different chemical compounds present in the supernova.	10
2.4	A plot of the extinction against inverse wavelength. The amount of extinction increases with decreasing wavelength.	13
2.5	Example of a dichroic jump visible at 5800 Å. This caused by miscalibrated CCDs and can be misinterpreted as a spectral feature. The graph has been taken from MARZ [15].	15
4.1	Ranking of the most used programming languages subdivided by career stage [27]. Python is the most popular language, and appears to be even more popular for the younger upcoming scientists.	25
5.1	Plot of each of the different subtypes used in this project at an age between -2 to 2 days (approximately maximum brightness). Note that Ib-pec, IIL, and Ibn have not been plotted because they do not have any templates in this age bin (see Table 5.3). The spectra have been preprocessed using the method outlined in section 7.1.	41
5.2	Plot of the Ia-norm subtype at 17 different ages from -18 to 50 days. The spectra have been preprocessed using the method outlined in section 7.1.	42

6.1	Process flow diagram indicating the possible outcomes of an input spectrum into my software. This was developed as an initial design.	44
6.2	A representative diagram illustrating the layers of a deep neural network.	49
7.1	A plot of an example supernova spectrum from OzDES, DES16C2ma observed on 16 September 2016. The blue line shows the raw data spectrum, while the red line shows the result after a low-pass median filter has been applied. Note that the units of flux are not important, but only the relative contributions of each feature.	53
7.2	A plot of the filtered data (DES16C2ma) from Figure 7.1 shown in red. The spectrum has then been de-redshifted and normalised and is shown in blue.	54
7.3	A plot of the binned wavelength (blue), the modelled continuum (green) and the continuum subtracted spectrum (red). These steps have been applied to the same DES16C2ma spectrum that has been used in Figures 7.1 and 7.2.	56
7.4	A plot of the cosine-tapered spectrum (green) which removes the edge discontinuities from the previous spectrum (blue). The red line is a normalised version of the green spectrum. It ensures that the flux is between 0 and 1. This process has been applied to the same DES16C2ma spectrum used in Figures 7.1 to 7.3.	57
7.5	A plot illustrating the issues with determining redshift using the zero-redshift model. The plot represents the probability of each redshift for a particular arbitrary classification bin. The redshift with the highest probability is $z = 0.076$	63
7.6	The initial landing page displayed to a user after they open DASH.	65
7.7	After clicking "Browse" on the landing page, the user is prompted to select a spectrum file. This is an example of how the user will be prompted. Note that the examples use an Ubuntu theme.	66
7.8	Message prompting the user to select a redshift before continuing.	66
7.9	A labelled example of a spectral file (DES16C2ma) being correctly classified.	67

8.1 Plot of the accuracy against training epoch for the two different trained models.
The training epoch refers to the relative amount of training that each model re-
ceived. The Agnostic Redshift model takes a longer time to train and crashes
after too much training. 74

List of Tables

4.1	Comparison of an online interface vs an offline interface on four different criteria. Online interfaces do not require an installation, however, they require an internet connection, cannot easily be integrated into projects, and their functionality is limited to a graphical interface.	23
5.1	A break down of the quantity of each subtype of supernova available in the WISeREP repository as provided by Ofer Yaron.	32
5.2	List of the 18 age bins that each of the supernova subtypes are separated into. They have been labelled with letters from A to R. These labels correspond to the headings in Table 5.3.	36
5.3	The number of templates for each subtype (rows) and each corresponding age (columns) are listed in the table. The letters A to R correspond to the age bin labels from Table 5.2.	37
6.1	A comparison of three different approaches to classifying supernova spectra. . .	47
8.1	Performance of DASH on the validation set based on four different criteria. Type refers to the broad type (i.e. Ia, Ib, Ic, II), while subtype refers to one of the 17 different types defined in chapter 5. Subtype and age refers to the algorithm choosing exactly the correct classification bin with the age correct. Type and age refers to choosing the correct broad type and age.	72

8.2 Classification of supernovae released in the most recent ATELS by OzDES [40, 18, 30, 29]. The first column is the name of the observed object, the second column is the redshift determined by MARZ. The third column is the classification given in the ATEL by OzDES. It details the type followed by the age in brackets. A (+) indicates that it is after the maximum brightness, (-) indicates that it is pre-max, while (0) represents that it is at maximum. A question mark indicates that the ATEL is not certain on the classification. Most of these measurements were taken by Superfit or SNID. The fourth column is the classification from DASH. The fifth column states the softmax regression probability provided by DASH. The sixth column has a tick if the ATEL and DASH agree on the type of the supernova, and a cross if they disagree. 77

Chapter 1

Introduction

1.1 Motivation

While the notion of an expanding universe is not a new phenomenon, the 1998 discovery [31, 34, 37] that this expansion rate is accelerating and not slowing down was a major challenge to our understanding of the composition of the universe. The discovery led to the 2011 Nobel Prize in physics, but the cause of the acceleration remains unknown. The known luminosity of Type Ia Supernovae (SNIa) have provided some of the most compelling evidence for this discovery, and continues to be a major focus of study.

The Dark Energy Survey in Australia (OzDES) is currently in a mature stage of its life span - being in its fourth year of its five-year spectroscopic survey on the 3.9m Anglo-Australian Telescope (AAT) [11, 20]. The goal of the survey is to understand the cause of the universe's acceleration to determine whether theories supporting "dark energy" in the form of Einstein's cosmological constant, or theories replacing General Relativity with a modified theory of gravity on cosmic scales can be ruled out. To this end, the survey is observing tens of thousands of objects, with the hope of measuring approximately 2500 Type Ia supernova host galaxies [45].

Currently, the process of classifying these astronomical objects is enormously time-consuming, and prone to human-bias and human-error. The development of a software that can automate the classification process is of high interest to the supernova analysis community.

1.2 Scope

One of OzDES's key aims is to measure the redshifts and luminosities of approximately 2500 Type-Ia supernova host galaxies [19, 20]. To prevent contamination from other supernova types, OzDES must be able to classify the targets. This project will aim to develop a software which can minimise the human-time involved in classification, while also limiting human bias and error so that spectra from the AAT can be objectively, quickly and accurately classified. The observed objects will predominantly be supernovae, however, the program should be able to broadly determine the transient type. Within this, the software should accurately classify the exact supernova type, its redshift, and age since maximum light.

The AAT spectra of supernova events will also be intermixed with a lot of light from the host galaxy, and may contain a lot of background noise and high-frequency features due to the various issues outlined in Chapter 2. The software should be able to distinguish the type of supernova spectrum from the intermixed host-galaxy, and account for some low signal-to-noise spectra.

The goal is to build upon previous software, to design a program capable of automating the classifying process, while also aiming to improve upon the speed and accuracy of classification compared to previous work. Ideally, it should be capable of making humans as obsolete as possible in this process, so that scientist's time can be better used, and so that human error can be minimised. Overall, the project should be accurate, fast, and user-friendly.

One of the primary innovations of this thesis is the use of a machine learning technique called deep neural networks (or deep learning). Previous spectral classification tools all make use of statistical classification techniques such as cross-correlations [7, 15, 4, 36] or chi-squared minimisation approaches [16] based on matching an input with a set of templates. In this thesis, I have utilised a novel approach to the problem that has significantly improved upon the speed and objectiveness of current methods (see Chapter 9).

1.2.1 Wider Context

There has been a significant amount of interest in my classification tool from not only OzDES, but also from a lot of scientists in the wider international DES (Dark Energy Survey) community in the United States and Europe. For this reason, while the main aim of the project is to develop a classification tool specifically for the OzDES collaboration, I have written the software so that it can be used for spectral data in a range of different file types from any worldwide telescope. As such, the software has been made open-source and available on PyPI and GitHub for the worldwide astronomy community.

Moreover, to enable easy uptake of this tool over current tools, *DASH* (name of my classification tool) has been made available in two different interfaces. Firstly, an easy to use graphical interface has been developed for visualisation and easy classification of spectral data files. Secondly, it has also been made available as an importable python library available on PyPI, so that classifications can be incorporated into the user's work, and so that several spectra can be classified iteratively. To this end, unlike previous tools, my software has been written in the Python programming language for the reason that it is one of the most popularly used languages among astronomers.

Chapter 2

Background

2.1 Astronomical Objects

The targets being observed by the AAT primarily consist of transient astronomical objects, which refer to phenomenon whose duration lasts from seconds to days, weeks or years. The different astronomical objects that will need to be classified include supernovae, galaxies, Active Galactic Nuclei (AGNs), Luminous Blue Variables (LBVs), and M-Stars. DASH should be able to broadly identify the type of object being observed, however, an accurate classification of the non-supernovae objects is outside the scope of the project.

2.1.1 Flux, Luminosity, and Magnitude System

In astronomy, the amount of light detected by a telescope is called 'flux'. This is analogous to how bright a star looks to us from Earth. However, due to the fact that objects further away appear as being dimmer, astronomers use a consistent measurement of the actual energy output from an astronomical body called the 'luminosity'. This is a measure of the total amount of energy emitted by an object per unit time. In SI units, luminosity is measured in Joules per second or Watts. The relationship between Flux and Luminosity is shown in equation 2.1,

$$L = 4\pi d^2 F, \tag{2.1}$$

where L is the luminosity, F is the Flux, and d is the distance to the object. Because these numbers are generally very large, astronomers find it more convenient to use a unit-less log scale convention for both luminosity and flux called absolute and apparent magnitude, respectively. The apparent magnitude of two astronomical objects A and B is related to the flux of the objects by the following equation,

$$m_A - m_B = -2.5 \log_{10} \left(\frac{F_A}{F_B} \right) \quad (2.2)$$

where m is the apparent magnitude. Similarly, the absolute magnitude is related to the luminosity. Equation 2.3 shows the relationship between the absolute magnitude, M , to the apparent magnitude, and distance to the object in mega-parsecs.

$$m - M = -5 + \log_{10}(d). \quad (2.3)$$

2.1.2 Supernovae

The primary targets of OzDES are supernovae. Supernovae are the result of the core-collapse of massive stars or the thermonuclear explosions of white dwarfs. They are classified based on the presence of certain features in their optical spectrum taken near maximum light as illustrated in Figure 2.1.

Types

Supernovae are classified into four broad types Type-Ia, Type-Ib, Type-Ic, Type-II.

Type-Ia supernovae (or SNIa) are the most important targets of the survey. Unlike the other types, SNIa's are caused by the thermonuclear reaction of a binary star system consisting of a white-dwarf accreting matter from a companion star. The white-dwarf eventually accretes so much mass that its core reaches a critical density that causes an uncontrolled fusion of carbon and oxygen, releasing a consistent amount of energy. The amount of energy released at peak luminosity is standardisable among all Type-Ia supernovae, and it has thus been used as a standard candle in the universe to measure cosmic distances.

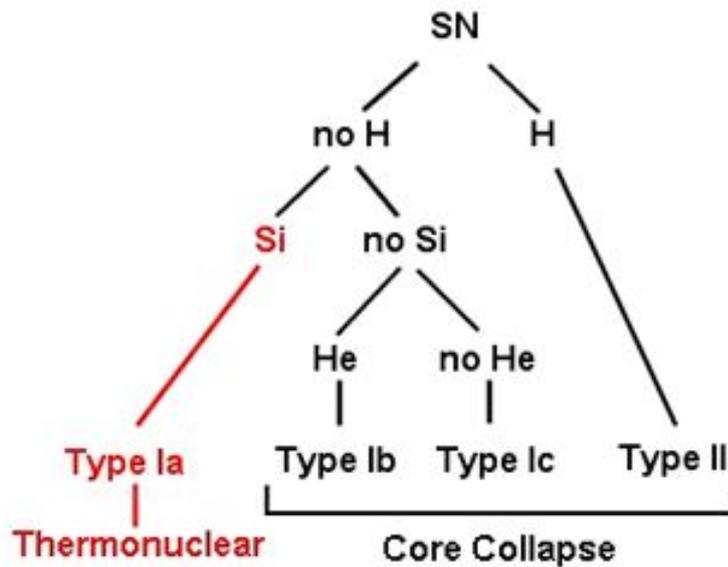


FIGURE 2.1: The different supernovae types are classified based on the presence or absence of certain chemical features in their spectrum. [42]

Type-Ib supernovae (or SNIb) are stellar explosions caused by the core collapse of massive stars. They are identified by the fact that they contain Helium and not Silicon in their spectra.

Type-Ic supernovae (or SNIc) are similar to Type-Ib except that they lack the Helium absorption line in their spectra.

Type-II supernovae (or SNII) are caused by the core collapse of massive stars and are identified by the presence of Hydrogen in their spectra.

Within each of these broad types, several more specific subtypes have been defined by astronomers due to a range of different characteristics in the observed spectra. In fact, supernova types are defined by variations in their spectral appearance rather than the physical mechanism that causes them. The reason for this is often due to the mechanisms not being well understood [42].

As outlined in Chapter 5, the supernova types used for classification in this thesis are based on the types defined in [38, 7, 25]. These 17 subtypes are listed below next to their broad type.

SN Ia: Ia-norm, Ia-91T, Ia-91bg, Ia-02cx, Ia-csm, Ia-pec

SN Ib: Ib-norm, Ibn, Iib, Ib-pec

SN Ic: Ic-norm, Ic-broad, Ic-pec

SN II: IIP, IIL, IIn, II-pec

The unprecedented number of supernovae being collected by OzDES means that it is possible that the survey will discover supernovae that do not fit into any of these categories, and perhaps require new physics to be able to be understood. One such example is the recently discovered superluminous supernovae (SLSN) [39]. These SLSN's have been categorised as a Ia-pec. If newer supernova types are discovered, it is expected that the software will flag that the spectrum cannot be identified accurately.

Age

Unlike many other astronomical objects, supernovae are transients, whereby their spectra changes with time and are only visible for a few weeks. The age of a supernova is defined as the number of days after it emits its maximum light. Figure 2.2 illustrates the bolometric light curve of an example supernova. The supernova builds to a maximum luminosity very quickly before decreasing in intensity. Because of the rapid change in the mechanics of the supernova event over time, the spectrum changes as time passes. Thus, each supernova object must not only be classified as a specific type, but its age must also be determined.

The 17 different supernova types listed above are plotted in Figure 5.1 shown in Chapter 5. However, each of these 17 types change with time (or age). One of the supernova types (Ia-norm) is plotted at various different ages in Figure 5.2.

2.1.3 Galaxies

Each of the supernovae observed by the AAT will include light from the host galaxy intermixed in the spectra. The software in this project aims to classify the supernova from the intermixed

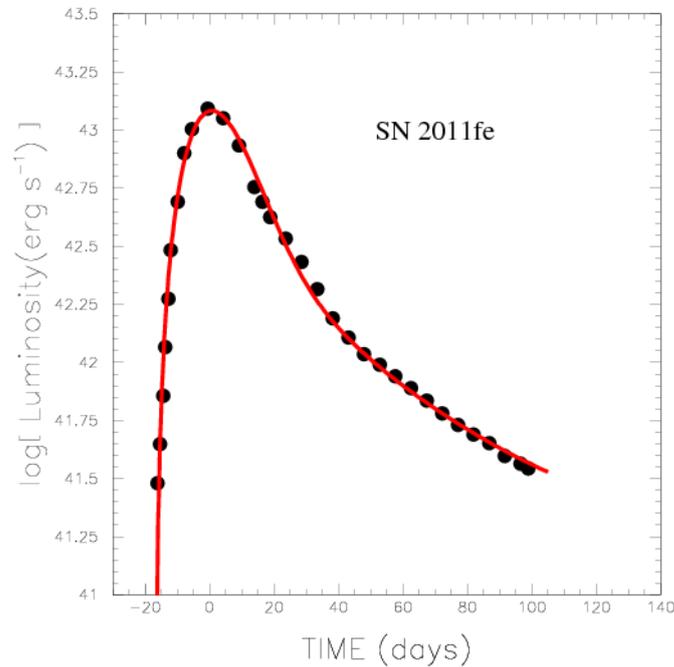


FIGURE 2.2: Example of the bolometric luminosity of a supernovae over time. The time at peak luminosity corresponds to an age of zero days. [10]

spectrum.

There are three main types of galaxies: Elliptical, Spiral, and Irregular. Within these types, the spectra can change greatly depending on the star formation rates, the relative composition of star types and gas, and its structure. While it would be convenient if a spectrum of the galaxy before the supernova event was available so that the galaxy light can be subtracted from the combined spectrum, it is often unavailable due to the limited telescope time.

2.2 Astronomical Spectra

The spectrographs on the AAT enable the flux from the observed object to be separated into its constituent wavelengths. A plot of the flux at different wavelengths can be used to determine a lot of information about an astronomical object. An example spectrum of a type-1a supernova at zero redshift is illustrated in figure 2.3.

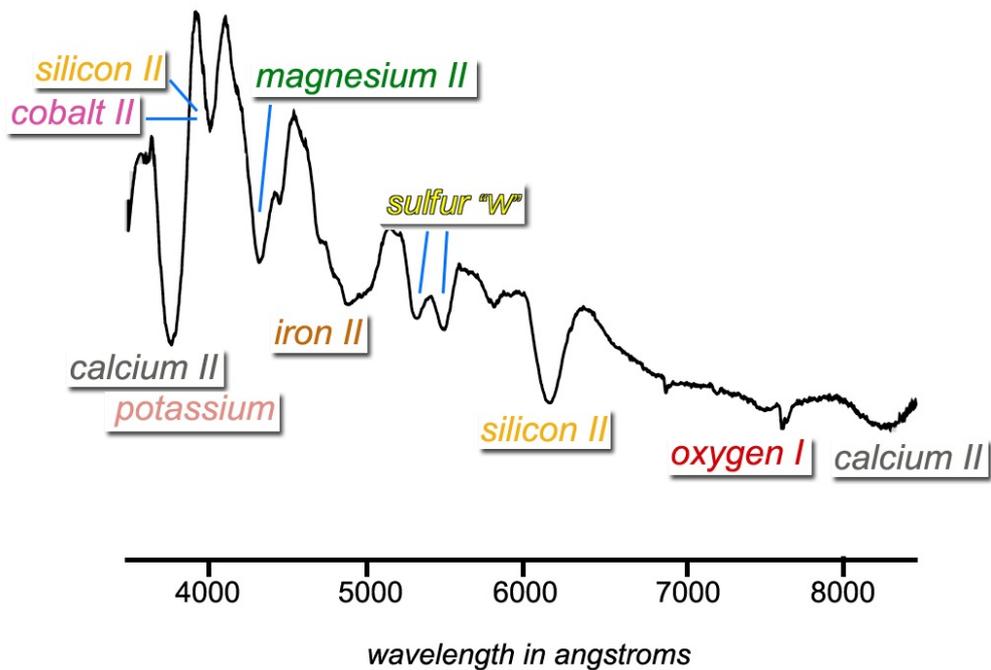


FIGURE 2.3: Spectrum of the Type-1a supernova '2002bo' at zero redshift. each of the peaks and troughs correspond to the absorption and emission lines caused by different chemical compounds present in the supernova.

2.2.1 Spectroscopic Features

Different chemical compounds (such as Hydrogen, Helium, Silicon etc.) emit or absorb light at known specific frequencies due to their atomic transitions. If a spectrum has a high enough signal-to-noise ratio, the peaks and troughs (which correspond to emission and absorption lines, respectively) in the spectrum can be used to determine the elements contained in the object (Figure 2.3). Based on the specific chemical features in the spectrum, the type of astronomical object can be determined.

2.2.2 Redshift

The expansion of the universe causes electromagnetic radiation to be stretched in proportion to the distance travelled by a photon. This stretching is called cosmological redshift, and is observed as an increase in the wavelength compared to the emitted wavelength. The relationship between the cosmological redshift, z , and the observed and emitted wavelengths is detailed in

the following equation,

$$z = \frac{\lambda_{observed}}{\lambda_{emitted}} - 1. \quad (2.4)$$

Thus, if the wavelength has doubled in size when it was observed compared to when it was emitted, the redshift is $z = 1$, whereas objects that have not been redshifted and hence, are very close to us, have a redshift $z = 0$. The cosmological redshift can be thence determined by comparing a spectrum to an zero-redshift template.

Moreover, objects that are further away, or were emitted at an earlier point in time will have travelled through more expanding space, and would thus have a higher redshift. The rate that space is expanding is given by Hubble's law which states that the recession velocity caused by expanding space is faster as the distance increases. This is formalised in the following equation,

$$D = \frac{H}{v} \quad (2.5)$$

where D is the co-moving distance to the object, v is the apparent recession velocity, and H is Hubble's constant. Due to the fact that the exact value of Hubble's constant, and the exact distance is dependent on the cosmological model used - which is still being actively considered [12, 28] - astronomers prefer to use redshift instead of time or distance.

This project will use translations in the spectra to determine cosmological redshift. However, redshift estimates can also be obtained from photometric data. Whilst photometric data has the advantage of being able to redshift fainter objects, photometric redshifts are not as accurate as spectroscopic measurements [9]. Moreover, the OzDES team is making use of the AAT with the AAOmega spectrograph [45], and thus only require spectroscopic redshifting.

2.2.3 Distortions

The spectra from the AAT is subject to distortions caused by Interstellar dust, and sky lines in the earth's atmosphere due to the fact that the AAT is a ground-based telescope. The signal may also be contaminated with background noise.

Interstellar Dust

Dust grains in the interstellar medium are solid, macroscopic particles composed of dielectric and refractory materials. The dust particles can cause wavelength-specific extinction of light, polarization of light by scattering, absorption of starlight, and infrared emission from heated grains [33]. These absorption features can remove useful lines used for matching spectra, or can appear as features in the spectrum that are not a part of the object.

Interstellar extinction refers to the dust particles extinguishing starlight passing through them. The amount of selective extinction is given by,

$$E_{B-V}(\lambda) = \frac{A_V(\lambda)}{R_V} \quad (2.6)$$

where $A_V(\lambda)$ is the total extinction at a specific wavelength, and R_V is an empirically found parameter that indicates the ratio of the total to selective extinction [33]. Observationally, R_V ranges between 2 and 6, but previous work usually adopts a value of $R_V = 3.1$ for diffuse interstellar medium, and $R_V = 5$ for dense molecular clouds.

Extinction tends to increase with decreasing wavelength, and hence, R_V is a measure of the relative slope of the extinction curve. The simple one-parameter model tends to be reasonably accurate for wavelengths between 3000Å and 7000Å. Above 7000Å, however, the extinction law is essentially independent of R_V , while below 3000Å, multiple parameters are required to adequately fit the observed extinction. In the UV (ultraviolet) to NIR (near-infrared) wavelengths, the interstellar extinction law is $A_\lambda \propto \lambda^{-1}$ [33]. A plot of the extinction against wavelength is illustrated in Figure 2.4.

This project will deal with extinction by subtracting the continuum from the spectra. Continuum removal places more emphasis on the spectral features instead of the colour information which changes due to interstellar dust.

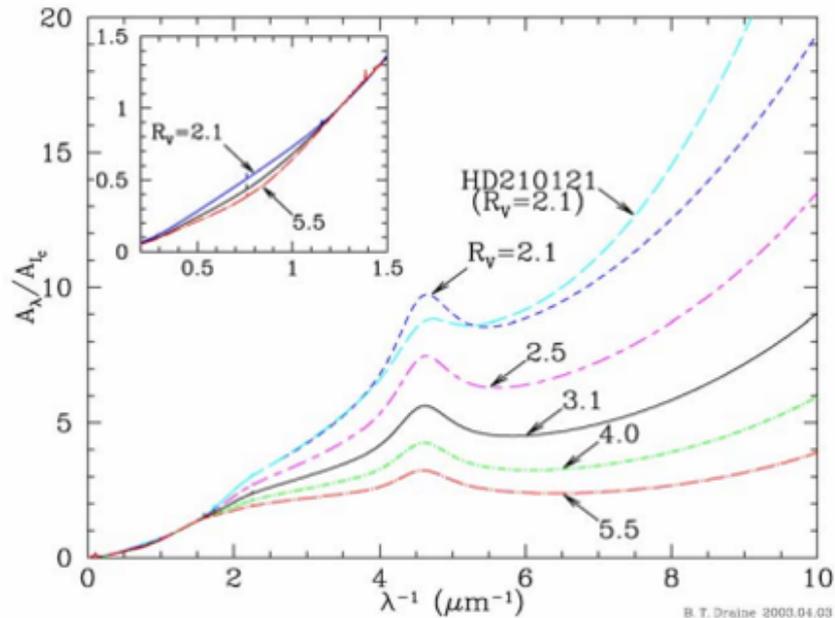


FIGURE 2.4: A plot of the extinction against inverse wavelength. The amount of extinction increases with decreasing wavelength.

Sky lines

Given that the AAT is a ground-based telescope, it is subject to the emission and absorption lines caused by particles in the Earth's atmosphere. The AAT will generally limit this effect by observing the night sky with and without the target, so that the night sky can be subtracted from the target spectrum. This process can sometimes be imperfect, however, due to spectrograph errors, optical distortions, and variability in the sky's emission and absorption spectrum [17].

Noise

For any measurement tool, there will always be an unavoidable level of random statistical noise mixed in with the signal. This can be caused by thermal noise, electronic noise from surrounding circuitry, dark noise (which is not normally an issue for Charge-Coupled Devices (CCDs)), or shot noise from the target source or the atmosphere [8]. While several precautions

are taken by the OzDES astronomers at the AAT to reduce the noise level, it is not possible to remove all of the noise.

The amount of target signal compared to the level of noise (S/N ratio) will vary for each target from the AAT. While it will obviously be easier to classify targets with a high S/N ratio, several targets with a low S/N may not be able to be classified. This project will have to evaluate the types of sources that cannot be classified, and determine whether there are similarities in the types of these sources. If there are similarities, such as a particular type of host galaxy or supernovae being classified less reliably, then this will add a bias to final supernova dataset and will need to be considered.

2.2.4 Equipment Miscalibration

The AAOmega spectrograph used on the AAT features both a red and a blue arm in its spectrograph before the light reaches the CCDs [2]. Since CCDs vary in their sensitivity to different wavelength ranges of light, the spectral arms with different CCDs are occasionally miscalibrated. A miscalibration in the two arms is often observed as a dichroic jump in the spectrum which can easily be mistaken for an emission line. This dichroic jump will occur at the boundary of the red and blue arms which is usually at approximately 5800\AA . An example of this dichroic jump is illustrated in Figure 2.5.

2.3 OzDES

The Australian Dark Energy Survey (OzDES) is a five-year 100-night spectroscopic survey on the Anglo-Australian Telescope, whose primary aim is to follow up objects selected by the international Dark Energy Survey collaboration (DES). One of the four key science areas of DES is the supernova survey, where OzDES plays a key role in measuring the redshifts and luminosities of approximately 2500 Type-Ia Supernovae host galaxies [45]. Type-Ia supernovae (SNIa) play a particular importance in measuring cosmic expansion because they act as standard candles that can provide accurate distance scales in the universe (see section 2.1.2). The overall

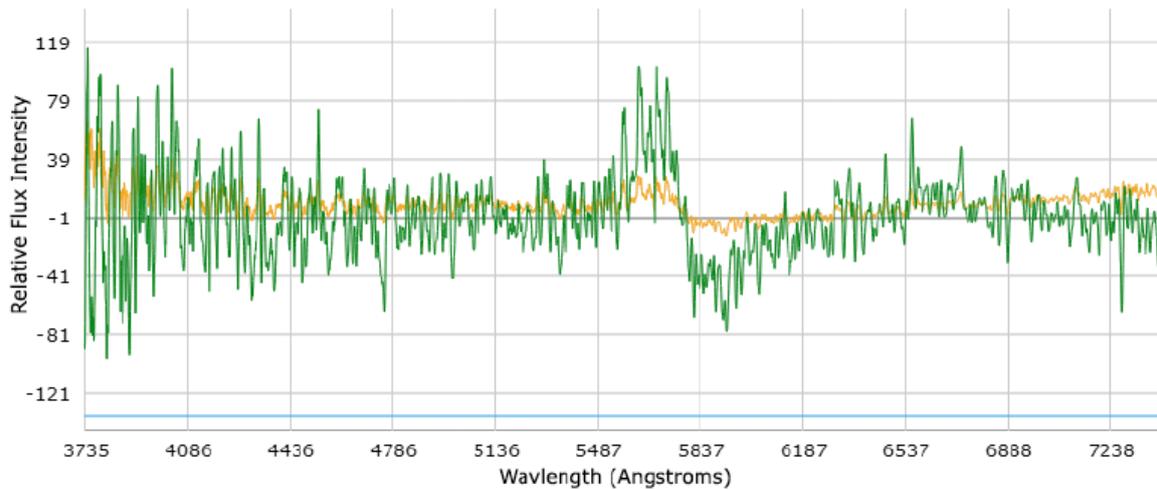


FIGURE 2.5: Example of a dichroic jump visible at 5800 Å. This caused by miscalibrated CCDs and can be misinterpreted as a spectral feature. The graph has been taken from MARZ [15].

goal of the survey is to increase the current supernovae dataset, in order to understand the nature of dark energy and ultimately to rule out existing theories of cosmic acceleration including modified gravity theories, and theories involving Einstein’s cosmological constant.

The spectral targets have been selected by the Dark Energy Survey (DES). However, not all of these targets are necessarily SNIa host galaxies, and may include other types of Supernovae or other Transient Astronomical Objects. This project aims to classify the type of objects being observed, as well as to provide specific classifications on the redshift, age, and host galaxies being observed.

2.3.1 Anglo-Australian Telescope

The survey is being conducted on the 3.9m Anglo-Australian Telescope (AAT). This telescope has recently been upgraded to make use of a unique Two Degree Field (2dF) facility that is able to use 400 individual optical fibres to collect the light from up to 400 stars or galaxies from a two degree field of view [3]. The light from each of the optical fibres is then directed to a spectrograph, where it can be separated into its constituent wavelengths before being detected by a CCD for analysis. OzDES makes use of this unique new technology to observe thousands

of objects over 100 nights. Each object has a unique spectrum consisting of a measured light intensity at each wavelength, that the software in this project will analyse.

Chapter 3

Prior Methods

3.1 Current OzDES Approach

Currently, the process of classifying supernovae is very slow, and labour-intensive, with the classification process for a single supernova taking up to a few hours. Due to the fact that thousands of supernovae will be need to be classified by OzDES, this process must be automated. Because most supernovae are intermixed with their host galaxy, programs like SNID (see section 3.2) are not suitable. The main program used by astronomers for supernova classification is Superfit. This program takes several minutes per supernova, often has several difficulties when separating the host galaxy spectra, and is only accurate after a few different runs of the software that involve an experienced astronomer having to add in prior information about the object. This process involves a lot of constant human involvement, and the goal of this project is to limit that current requirement.

3.2 Prior Software

There are several different redshifting tools used for the classification of non-supernova objects, and only a few tools used for supernova spectral classification. This section will review the main software packages used by large cosmology surveys.

3.2.1 SNID

SNID is a very fast typing tool written in Fortran that is used to classify supernova spectra. It is based on the algorithms developed by [43]. While the speed of classification is very high, it is only accurate for pure supernova spectra and is not able to accurately classify spectra that are intermixed with their host galaxy, or spectra with a low S/N ratios. In the astronomy community SNID's primary purpose has been to distinguish between type-Ia and type-Ib/c supernovae. To do this, it uses a method of cross-correlation and pre-processing similar to that outlined in section 7.1 to determine the redshift, and compare the input spectrum with a set of template spectra.

It formulates a parameter called 'rlap', which gives an indication of how well an input spectrum matches a particular template spectrum [7]. The value is computed using the cross-correlation peaks, as well as the overlap of an input and template spectrum.

3.2.2 MARZ

MARZ is a very user friendly redshifting tool written in Javascript that has recently become widely accepted by the OzDES community. It surpasses previous redshifting tools such as RUNZ and AUTOZ due to its speed, web-based user friendly platform, and accuracy [15]. The software uses a modified version of the AUTOZ [4] cross-correlation algorithm to match input spectra against a variety of stellar and galaxy templates. Like SNID and AUTOZ, MARZ preprocesses its spectra using a similar method to that outlined in section 7.1. Because of its high accuracy and speed, DASH will make use of the redshifts determined by MARZ as a prior input for supernova classification.

3.2.3 AUTOZ

AUTOZ is a redshifting tool written in IDL that was previously used by the astronomy community [4]. It uses a very similar preprocessing and cross-correlation matching algorithm to that outlined in section 7.1. As the MARZ software analysed and improved upon the techniques

used by AUTOZ, this project will primarily focus on the redshifting techniques employed by MARZ.

3.2.4 Superfit

Superfit is a supernova classifying tool written by Andy Howell in IDL that is currently the main software used by a large part of the astronomy community [16]. Unlike the previous three software tools, Superfit makes use of a chi-squared minimisation approach to classifying spectra. Its advantage over SNID, and other tools is that it can subtract the spectra of host galaxies (selecting from eleven templates), and therefore deal with combined supernova and galaxy input spectra. As outlined in section 3.1, Superfit is limited by the amount of time it takes, and the amount of continuous user involvement that is required. The other downfall of superfit is that it can only classify into the four broad types, and cannot classify a spectrum into its subtype. Moreover, due to its very small set of supernova templates, it is not able to give accurate age estimates either.

3.2.5 Summary of Previous Tools

All of the prior software makes use of template matching techniques that involve either cross-correlation or chi-squared minimisations to ascertain a best matching template. However, using this approach means that the total computation time increases linearly with the number of templates. While cross-correlations are relatively fast, chi-squared minimisations are slow, and are the reason for the small number of templates available in Superfit.

Both SNID and Superfit's reliance on their template set mean that they cannot accurately gauge the specific subtype and age of an input spectrum. This is because instead of using the aggregate feature of a particular class of supernova, they can only compare with one template at a time. DASH significantly changes this framework by classifying based on features instead of templates.

Chapter 4

Requirements and Tool Decisions

After consultation with several members of the OzDES collaboration, a set of usability and functional requirements were developed. These requirements were shaped such that if they were met, DASH would be considered a viable and welcomed replacement to current tools. This chapter details these requirements, and also outlines some key tools used and decisions made.

4.1 Speed and Accuracy

The AAT is observing an unprecedented number of transient objects that need to be classified. Existing tools are either too slow and require a lot of human involvement (Superfit), or are inaccurate for low signal-to-noise or galaxy-contaminated spectra (SNID). As such, one of the key outcomes of my project is that it needs to minimise the human-involvement by being fast and as autonomous as possible. At the same time, it must also match or improve upon the accuracy of current tools.

4.2 Installation and Updating

One of the biggest issues with many astronomy tools is the difficulty in the initial installation of the software package. In particular, tools such as RUNZ and Superfit are notoriously difficult to get started with, often requiring the laborious and non-user-friendly tasks of installing several dependencies and setting system paths. As such, in order to enable widespread uptake of my software, OzDES requires that the installation process be simple and user-friendly. Ideally, installation should be possible with a nearly self-contained installation package with all dependencies either being installed automatically or with minimal difficulty.

Furthermore, the maintenance of the software should not require much effort from the end-user. Updates should occur automatically or with very basic and easy to follow instructions. In addition, OzDES expects that the user will not need to manually update any dependencies or to reconfigure any files or content.

4.3 Operating System

While being cross-platform is not a strict requirement for the OzDES team, most previous applications require a Mac or Linux Distribution. Moreover, most members of the astronomy community have requested that the software be easily available on the Mac operating system.

As discussed in Chapter 7, while DASH has been developed to be cross-platform, one of the main dependencies is currently only available on Mac and Linux. As such, at this stage, DASH has been optimised to run on both Mac and Linux distributions, with untested support on the Windows OS. However, as most astronomy applications are only suited to the Mac OS, there is no current need to be operable on other operating systems.

4.4 Online vs Offline

A recent tool developed by Sam Hinton, MARZ [15], is the first redshifting tool to run completely on an online interface. To do this, it was written nearly exclusively in JavaScript and HTML, and does not require a large server to host and relay data. Instead, it operates exclusively within the browser. Running completely online has eliminated the previous installation and cross-platform issues that surrounded previous software. Due to MARZ's effectiveness and the fact that no installation is required, it has widely overtaken previous software, such as RUNZ, as the primary redshifting tool.

However, MARZ is a very lightweight tool that requires minimal computational effort: making use of only 11 different templates instead of the over 4000 used in this project. MARZ works purely as a redshifting tool rather than a classification tool, and therefore does not require as much computational effort. Thus, in order to enable DASH to be online, a dedicated server would need to be setup to act as a host that processes the large amounts of data.

To make this decision about the front-end interface, a qualitative poll from the future users of this software from the OzDES collaboration was conducted ¹. The key advantages and disadvantages of each approach are illustrated in the table below.

	Online Interface	Offline Interface
Installation	None	Required
Internet access	Required	Not needed
Integration	Not easy	Possible
Functionality	Fixed	More options

TABLE 4.1: Comparison of an online interface vs an offline interface on four different criteria. Online interfaces do not require an installation, however, they require an internet connection, cannot easily be integrated into projects, and their functionality is limited to a graphical interface.

¹The astronomers who provided feedback on this decision have been acknowledged in the Acknowledgements section at the beginning of this report

As illustrated in Table 4.1, the main advantage of an online interface over an offline one is that there is no installation required. However, one of the main comments from many members of the OzDES collaboration were that classification of supernovae often occur at locations where internet access is very limited. Anecdotally, supernova classification can often occur at telescopes where internet access can be very poor, and also at airports and airplanes as astronomers travel. For this reason, the main sentiment was that provided that installation is relatively easy, an offline interface is definitely preferred.

What's more, an offline tool also has the advantage of more easily enabling library functions that can be integrated into the classification pipeline, or into the work of scientists without requiring human inspection. The ability to have a customisable functionality, which is more readily available on an offline importable package is also an additional benefit.

Ultimately, an offline tool is widely preferred, provided that it is easily installable, and is also easier to implement due to the difficulties in managing a dedicated server.

4.5 Language Decision

The main languages that I considered for this project were Python, C++, IDL, Fortran, Javascript, and Java. However, one of the main guiding factors of this decision were to use a language that is familiar to astronomers who would be using it. This is an important requirement not only so that the software can be maintained by others in the future, but also so that it can be more easily customised to suit their work.

According to a survey by [27], the most popular language used by the worldwide astronomical community is Python, with IDL, C++, and Fortran coming next. This is illustrated in Figure 4.1.

While Javascript would have been useful for an online tool, this rules out Java and Javascript which are used by less than 3% of the community. The figure highlights that Python is the most popular, and is significantly more popular among graduate students and postdocs. This suggests that Python usage is the fastest growing among astronomers.

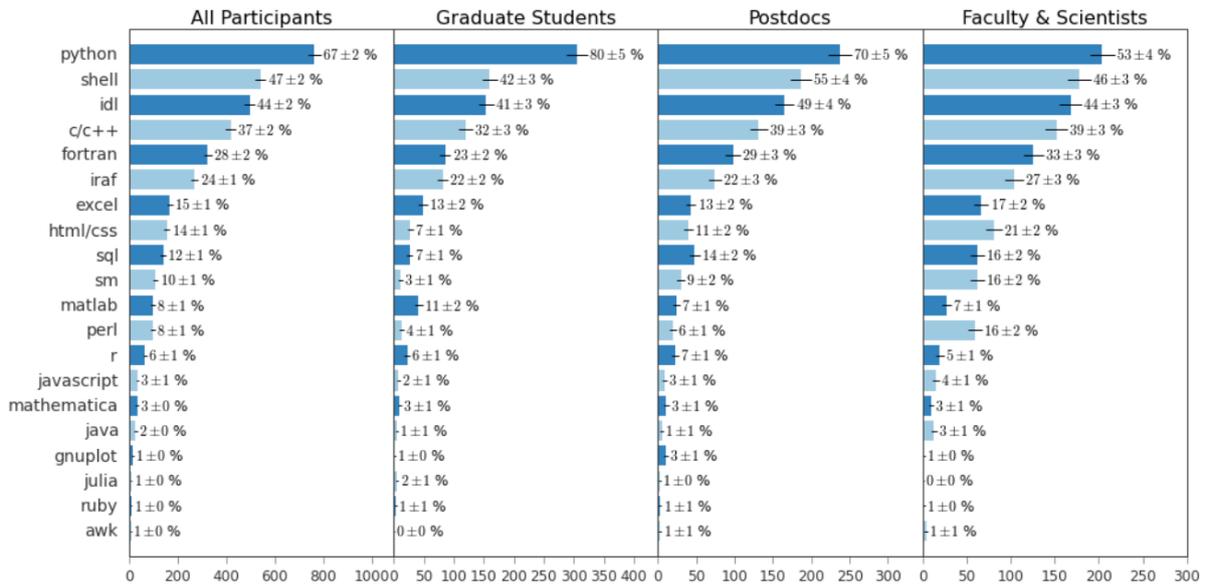


FIGURE 4.1: Ranking of the most used programming languages subdivided by career stage [27]. Python is the most popular language, and appears to be even more popular for the younger upcoming scientists.

Previous tools such as Superfit and SNID have been written in IDL and Fortran respectively. While IDL and Fortran both have external library support, they are disadvantaged by their difficult readability in their syntax, and more importantly that they are both procedural languages with scant GUI features. Moreover, IDL is not freely available and requires an annual cost to install onto personal computers.

C++ is a low-level compiled language that is known for its high-speed execution. However, one of its main disadvantages is that given its lower popularity in astronomy, there are much fewer modern libraries available, and the libraries that do exist are often platform dependent.

On the other hand, Python is not only popular, but it is also available cross-platform, has highly readable syntax, and has a vast range of external libraries. One of its disadvantages, however, is that its speed does not rival that of faster low-level languages like C and C++. However, some key scientific libraries including NumPy, and Tensorflow [1] (both used in this project) are written with a C++ back-end that allows for the easy-programming of Python with the efficiency and speed of C++. Overall, this all makes Python the clear choice for this project.

4.6 Integration

For the purposes of the OzDES project, a standalone supernova classification tool is all that is required. However, while this is not a strict requirement, some members of the OzDES collaboration have mentioned that beyond OzDES, it would be useful to have a tool that can be integrated into their code. This would involve being able to classify without needing to look at a graphical interface, and instead using DASH as a python library function. One use case would be to import the DASH package into a python script, and then call classification functions to be run on some input data file. The advantage of this is that it enables users to iteratively classify several spectra, and to use the results of the classification in their work. It also allows for an automated classification that does not rely on visual human checks.

For this reason, DASH has been made available on the Python Package Index (PyPI) such that it can be easily installed with 'pip'. Users are then able to import DASH and use the available library functions (see Chapter 7).

4.7 Libraries and Tools Used

4.7.1 Libraries

Numpy: This project makes heavy use of Numpy for the manipulation of the large arrays used in the project. Numpy makes use of a C++ backend that enable it to provide very fast matrix and array calculations.

Scipy: Scipy is a common statistics library that is used in this project for the spline fit continuum removal (see section 7.1).

Specutils: This library is used primarily for reading a FITS files, a particularly common file type in astronomy.

PyQt4: This is used primarily for it easy, efficient, and vast GUI libraries. The GUI as well as the graphing tools were implemented using this library.

Tensorflow [1]: This was heavily used in this project for its powerful machine learning libraries that enable the development of multi-layer neural networks.

4.7.2 Software Tools

Git and GitHub

Git is an open-source and intelligent version control software used by several developers and companies around the world. It has been used in this project to both control the several files that make up the project, and to easily share my code with my supervisors, and the members of the OzDES community who have tested the project. I believe in open-sourcing my code, and have used GitHub as a public repository. The code is available at https://github.com/daniel-muthukrishna/SNClassifying_Pre-alpha.

Resources

Some of the main external resources used in this project include PyCharm and Obelix.

- I have used PyCharm as the main integrated development environment (IDE) to develop my software and to upkeep a local git repository.
- I have also made use of the Obelix Supercomputer at the University of Queensland Physics Department. This was used heavily in the training of the deep learning model, which used 10 to 40 cores on the cluster and took several hours or days of computation.

Chapter 5

Data Description

This chapter describes the types of data that were used to train the model, details how the data was collected, and outlines the decisions made that led to the final dataset.

5.1 Data Collection

To analyse and classify supernova spectra, a large set of templates from a range of different types of supernovae at varying ages were collected.

5.1.1 SNID Templates

SNID [7] makes use of 1515 spectral templates from 111 different supernovae that have been observed by various astronomical surveys between the years 1979 to 2006. Each of the 111 supernovae were observed on several different days (or ages) which is how we get 1515 templates. Each of these templates have been de-redshifted back to redshift zero so that they can all be equivalently compared.

SNID uses the Supernova subtypes defined by John Tonry and Marc Davis [43]. There are 14 different subtypes they classify into. These are listed here:

Ia-norm, Ia-91T, Ia-91bg, Ia-csm, Ia-pec, Ib-norm, IIb, Ib-pec, Ic-norm, Ic-broad, IIP,
III, IIn, II-pec.

SNID processes all their templates in the same way, whereby they smooth the spectra with a low pass filter, remove the continuum with a spline fit, bin the spectra into 1024 points and apodize the edges. This template processing method is outlined in section 7.1.

Templates-2.0

In 2012, the number of SNID templates was significantly increased [6] to consist of 3754 spectra across 349 different supernovae. This new set added several more supernovae that were obtained between 1993 to 2008 through the Center for Astrophysics Supernova Program. However, several of the templates did not have the correct age information. The templates without age information were not made clear in [6], but the ages in the files were all set to -99 days to represent that the age was unknown. These templates were iteratively deleted from the template set used for training in this project by writing a short script to find and delete files where the ages were set to -99 days.

5.1.2 Superfit

Superfit [16] makes use of a total of 306 spectral templates, which is much less than the number used in SNID. The main reason for the lower number of templates used is that the chi-squared matching process employed by Superfit is significantly slower than the cross-correlation method used by SNID. The computational matching time of both methods increases linearly with the number of templates used [7].

The main problem with the templates used by Superfit is that they only distinguish the Supernova types into the four main types (Ia, Ib, Ic, II), and do not separate them into subtypes. This makes them incompatible for use with SNID templates. Additionally, all of the supernova templates used in Superfit are also used in the original SNID templates or the Modjaz and Liu [25, 22] templates outlined in the following sections.

5.1.3 Increased SNID Templates

Liu & Modjaz

In 2014-2016, Yuqian Liu and Maryam Modjaz released a few papers [25, 24, 22, 21] that updated the Templates-2.0 set from SNID to correct several of the Ib/c templates which had the incorrect subtype or age information. In addition, they introduced two new subtypes called Ib-n and Ic-pec to account for variations in particular features of the spectra. Finally, they added 70 new supernovae to the template set. All of these updated templates including the original SNID templates are available on a GitHub¹ repository that released while this thesis was being written.

BSNIP

In 2012 Jeffrey Silverman et al. [38] added 4015 new spectra from approximately 350 different supernovae as part of the Berkeley SN Ia Program (BSNIP). Many of these were, however, also part of the SNID templates from Blondin, Liu and Modjaz. The BSNIP release also created two new subtypes called Ia-02cx and Ia-99aa. The BSNIP v7.0 templates were used in this thesis and were downloaded from the following link https://people.lam.fr/blondin.stephane/software/snid/faq.html#bsnip_v7.

During this thesis, I emailed Jeffrey Silverman about what supernova types he thought were important to include in my new software. He stated that although he formed a new subtype Ia-99aa in his 2012 paper [38], he believed that Ia-99aa could be a subset of Ia-91T, and may not need its own category. Based on this information, and the fact that there were not enough Ia-99aa templates to form its own subtype, I made the decision to let all Ia-99aa's fall under the Ia-91T subtype.

¹ GitHub repository with updated SNID templates are available here: <https://github.com/nyusngroup/SESNTemplate/tree/master/SNIDtemplates>

5.1.4 WISeREP

WISeREP [44] is an online database of supernovae. It consists of 13381 different spectra, nearly encompassing all the supernovae that have ever been observed. I contacted the author of the repository, Ofer Yaron, and he provided me with all of the spectra broken down into 29 different supernova subtypes. These are listed in the table below.

Subtype	Quantity	Subtype	Quantity	Subtype	Quantity
SLSN-I	147	Ia-91T	134	Ic-pec	34
SLSN-II	75	Ia-pec	241	II	730
SLSN-R	9	Ia-SC	87	II-pec	64
SN	63	Ib	421	IIb	577
I	20	Ib-Ca-rich	13	IIIL	68
I-faint	15	Ib-pec	11	IIIn	856
I-rapid	7	Ib/c	210	IIIn-pec	52
Ia	7823	Ibn	172	IIP	586
Ia-02cx	69	Ic	656	SN imposter	29
Ia-91bg	97	Ic-BL	115		

TABLE 5.1: A break down of the quantity of each subtype of supernova available in the WISeREP repository as provided by Ofer Yaron.

WISeREP clearly provides a large range of data, while introducing a lot of new subtypes. There is a significant amount of bias in the dataset, whereby over half of all templates are Type Ia supernovae. This is due to the usual preference of observing SNIa's due to their scientific use. Adding this many templates would be very useful for training a machine learning model. However, the problem with this database is that the ages of the supernova spectra are not provided. In fact, the only way to find them would be to individually trace the observation dates and the assumed dates of maximum light for each supernova from some external papers and databases. However, after spending sometime searching for the ages of these supernovae, I determined that there is no easy way to find the ages. The reason for this is that either the date of maximum light is not known by astronomers, or the date of maximum light is only

given individually in the paper that discovered the particular supernova. Given that there are thousands of spectra, finding the age of each supernova is obviously not a task that can feasibly be completed in this thesis.

5.2 Deleting Templates

5.2.1 Unknown Ages

Many of the collected SNID templates listed in the previous section had unknown ages. Since age is an important characteristic for training in my model, I iteratively deleted any of the supernova spectra in my template set that had an unknown age.

5.2.2 Ic-broad bias

After testing my software, I found that there was a bias towards the Ic-broad subtype. Upon further investigation, I found that many of the templates that were marked as Ic-broad had a low signal-to-noise ratio and did not appear to match the majority of other Ic-broad spectra. As such, seven of the Ic-broad supernovae were deleted. These include:

- 2010ma
- 2010bh
- PTF10qts
- 2013cq
- 2013cq
- 2003dh
- 2013dX
- 2012bz

I have listed these here because my claim that they are not Ic-broad's could be significant to some astronomers who are interested in this subtype. As such, I believe that some further investigation into the above listed supernovae may be needed to clarify their type.

5.3 Need for more templates

The data that is used in DASH from Modjaz, Liu, Silverman, and Blondin as outlined in the previous sections have proven to be sufficient for effective classification (see Chapter 9). However, it is expected that if a wider and deeper range of templates is added to the training set, the accuracy of the model will improve. Perhaps in the future, the data from WISEREP or some other database with ages may be collated to increase the current dataset, and improve the machine learning model.

5.3.1 Bias in all Supernova Classification Tools

One of the main flaws in DASH, and all supernova classification tools including Superfit and SNID is that there are several gaps in the types of templates that have been recorded. As seen in Table 5.3, there are a lot of supernova types where we don't have any templates at many of their ages. This flaw is mainly due to the lack of interest by observational cosmologists who observe non Type-Ia supernova only once, and do not go ahead and collect more data for more ages of the supernovae.

5.4 Templates Description

The templates used in this thesis are the same as the updated SNID templates outlined in the above sections. Overall this gives a total of 3936 spectra across 515 different supernovae. In order to be consistent with current work, and to ensure that the work that has gone into improving the SNID template list is not lost if DASH begins to be adopted by the astronomical

community, I have chosen to use the same template processing format as SNID. As such, templates that are added for training in DASH should all be log-wavelength binned just as the SNID templates (see section 7.1 for an outline of this processing method).

5.4.1 Types and Ages

The 3936 templates used for this thesis have been separated into 17 different subtypes. These subtypes are listed below:

SN Ia: Ia-norm, Ia-91T, Ia-91bg, Ia-02cx, Ia-csm, Ia-pec

SN Ib: Ib-norm, Ibn, IIb, Ib-pec

SN Ic: Ic-norm, Ic-broad, Ic-pec

SN II: IIP, IIL, IIn, II-pec

Within each of these subtypes, there are a range of different possible ages. In general, supernovae are only bright enough to be noticed 20 days prior to the date of maximum light. Similarly, most supernovae are no longer of interest after around 50 days after their date of maximum light. While some supernovae are observed up to 100 days past their maximum, these are usually very dim, and the spectra are mostly dominated by their host galaxy light. For this reason, in this project we are only considering ages between the range of -20 days to +50 days, where the date of maximum light is defined as an age of 0 days.

In order to group the spectra into bins that can be trained on for the machine learning algorithm, I have chosen to split up the ages into 4 day intervals. As such for each supernova subtype, there are 18 age bins. These age bins are listed below:

Combining the number of subtypes and the number of ages means that we have a total of $17 \times 18 = 306$ different bins to separate all of the templates.

A: -20 to -18 days	G: -2 to 2 days	L: 18 to 22 days	Q: 38 to 42 days
B: -18 to -14 days	H: 2 to 6 days	M: 22 to 26 days	R: 42 to 46 days
D: -14 to -10 days	I: 6 to 10 days	N: 26 to 30 days	S: 46 to 50 days
E: -10 to -6 days	J: 10 to 14 days	O: 30 to 34 days	
F: -6 to -2 days	K: 14 to 18 days	P: 34 to 38 days	

TABLE 5.2: List of the 18 age bins that each of the supernova subtypes are separated into. They have been labelled with letters from A to R. These labels correspond to the headings in Table 5.3.

5.4.2 Number of Templates

The total number of templates that are available for each subtype and corresponding age is illustrated in Table 5.3. The columns are labelled with letters from A to R. These correspond to the age bin labels listed in Table 5.2.

From Table 5.3 we can see that there are significantly more Type-Ia supernova templates than the rest. This is due to the bias in cosmological surveys which tend to observe more SNIa's due to their scientific significance. We can also note that there are obviously very few templates in columns A and B since these represent ages between -20 to -14 days where supernovae are often too faint to notice.

Moreover, there are several gaps (cells with 0), where we don't have any templates. This means that there is no way that we can ever classify into these groups. This is a problem with all classification tools including SNID, Superfit, and DASH.

Accounting for SNIa Bias

The fact that there are a much higher number of Type Ia supernovae can lead to a bias in the classification process if it is not properly accounted for. To eliminate this bias, each of the templates in the 306 different bins were repeated in the training set, until each cell had the same number of templates as the largest bin (Ia-norm at 2 to 6 days). This process is equivalent to adding an extra weight to bins with a low template count.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Ia-norm	0	2	54	179	231	274	286	229	191	158	121	96	91	79	52	60	54	29
Ia-91T	0	0	18	51	54	32	19	25	20	27	26	21	19	12	13	9	11	15
Ia-91bg	0	0	0	10	24	25	39	23	24	17	21	10	10	13	2	2	6	5
Ia-csm	0	0	2	0	0	2	2	4	2	2	0	0	2	2	0	6	4	2
Ia-02cx	0	0	0	8	11	2	1	0	2	3	3	5	2	0	1	1	2	0
Ia-pec	0	0	5	15	18	20	13	17	10	9	8	12	8	8	7	7	4	1
Ib-norm	1	4	8	13	18	17	8	10	15	11	7	4	4	4	5	5	2	6
Ibn	0	0	0	0	0	0	3	6	3	2	3	1	1	3	1	1	2	1
IIf	4	12	15	7	13	6	13	12	15	11	8	5	3	4	7	6	3	2
Ib-pec	0	0	2	1	2	0	0	0	1	4	1	0	0	0	0	0	1	0
Ic-norm	0	1	1	11	18	18	15	9	7	12	9	12	5	10	3	4	3	8
Ic-broad	0	1	7	6	21	16	18	17	13	10	13	10	13	6	5	3	3	11
Ic-pec	0	0	0	0	3	9	7	0	1	0	4	3	0	2	0	0	0	2
IIP	0	0	0	1	12	30	23	22	12	10	4	11	10	5	13	5	5	3
III	0	0	0	0	0	0	0	3	4	0	0	0	0	0	0	1	2	0
IIn	0	0	4	0	0	8	2	4	6	2	0	0	0	6	0	4	6	2
II-pec	1	3	2	3	3	1	2	2	0	0	0	0	0	0	0	0	0	0

TABLE 5.3: The number of templates for each subtype (rows) and each corresponding age (columns) are listed in the table. The letters A to R correspond to the age bin labels from Table 5.2.

5.5 Plot of Templates

One template from each subtype of supernova observed at maximum brightness (-2 to 2 days) is plotted in Figure 5.1. However, as illustrated in Table 5.3, a few of the subtypes do not have any templates in this age bin. These templates which are also not included in the figure are Ib-pec, III and Ibn. The templates that have been plotted are a representative sample of the other datasets, and the particular supernova plotted was chosen at random.

These plots represent the spectra after they have been processed onto a 1024-point log-wavelength scale, continuum-subtracted, and cosine-tapered edges as outlined in the section 7.1. We can

see that many of the broad types share similar characteristics. While it may be useful to show the spectra of each of these subtypes at a range of different ages, this would take too much space in this report given that there are 306 different bins. Instead, Figure 5.2 plots the subtype Ia-norm at 17 different age bins from -18 days to 50 days.

From Figure 5.2, we can see that all Ia-norm's share very similar features. The only difference with age is that some of the features broaden or become relatively smaller or larger with changes in the age.

Figures 5.1 and 5.2 give an illustration of the variation in the data that the machine learning model must train on. Each of the spectra have been recorded at varying wavelength ranges due to the spectrograph restrictions at the telescope that observed the supernovae. This is illustrated on the figures by the spectra starting at different wavelengths. It should also be noted that all of the plotted spectra have been pre-processed, smoothed and de-redshifted to zero. Input data, on the other hand, is expected to have a much lower signal-to-noise, have a non-zero redshift, and be partially contaminated with host galaxy light. An example of an un-processed input spectrum is illustrated in Figure 7.1.

5.6 File Types

DASH has been customised to accept several different file types that contain supernova data. These are listed in the following subsections.

5.6.1 Inw File

The data templates from SNID are in a file type defined in [7] with a .lnw extension. These files include a lot of information about a single supernova. The header consists of information about the supernova and information about the spline fit that has been divided from the spectrum.

The first line is made of tab separated information that in order detail the following:

- Number of spectra in the file. These spectra correspond to the the number of ages that have been recorded for the particular supernova.
- Number of points that the spectra has been binned into (All are set to 1024 points).
- Minimum wavelength. All templates have been set to 2500 Angstroms.
- Maximum wavelength. All templates have been set to 10000 Angstroms.
- The maximum number of spline points used to fit the continuum of all the spectra in the file.
- The name of the template (e.g. SN2002ap).
- dta: Usually set to -9.99, but the significance of this is unknown and appears to be irrelevant.
- Template subtype (e.g. Ic-norm).
- Index of type (irrelevant for use in DASH).
- Index of subtype (irrelevant for use in DASH).

There are then a number of indented lines which give information about the continuum that has been divided from the spectra. This information gives the points used to model the continuum with a spline fit.

The next line gives several tab separated values indicating the ages of the spectral information in the columns below. Following this, there are several tab separated columns which give information about the flux of the spectra. Each column has 1024 rows to represent the log-wavelength binned and continuum divided spectra (see section 7.1). The first column gives wavelengths from 2500 to 10000 Angstroms. The following columns give normalised fluxes from 0 to 1 for each of the ages of the supernovae in the file.

5.6.2 ASCII, data, Two-column Text Files

Most of the raw data from the AAT are uploaded as .DAT files which consists of a two column tab separated file. The first column is the wavelength in Angstroms, and the second column is the flux. Any two-column data file can be input into DASH for classification.

5.6.3 FITS File

A very common file type used in astronomy is a FITS file. This file can be customised to contain a lot of header information about a particular astronomical object. For the purposes of DASH, it is only important to extract the wavelength and flux from the files. Both the wavelength and flux are identified by their names as keys in the file, and can be extracted using astropy's (python library) FITS file reader. I have made use of a python package called 'specutils' which enables easy reading of FITS files. (See the class ReadSpectrumFile in preprocessing.py on the GitHub repository.)

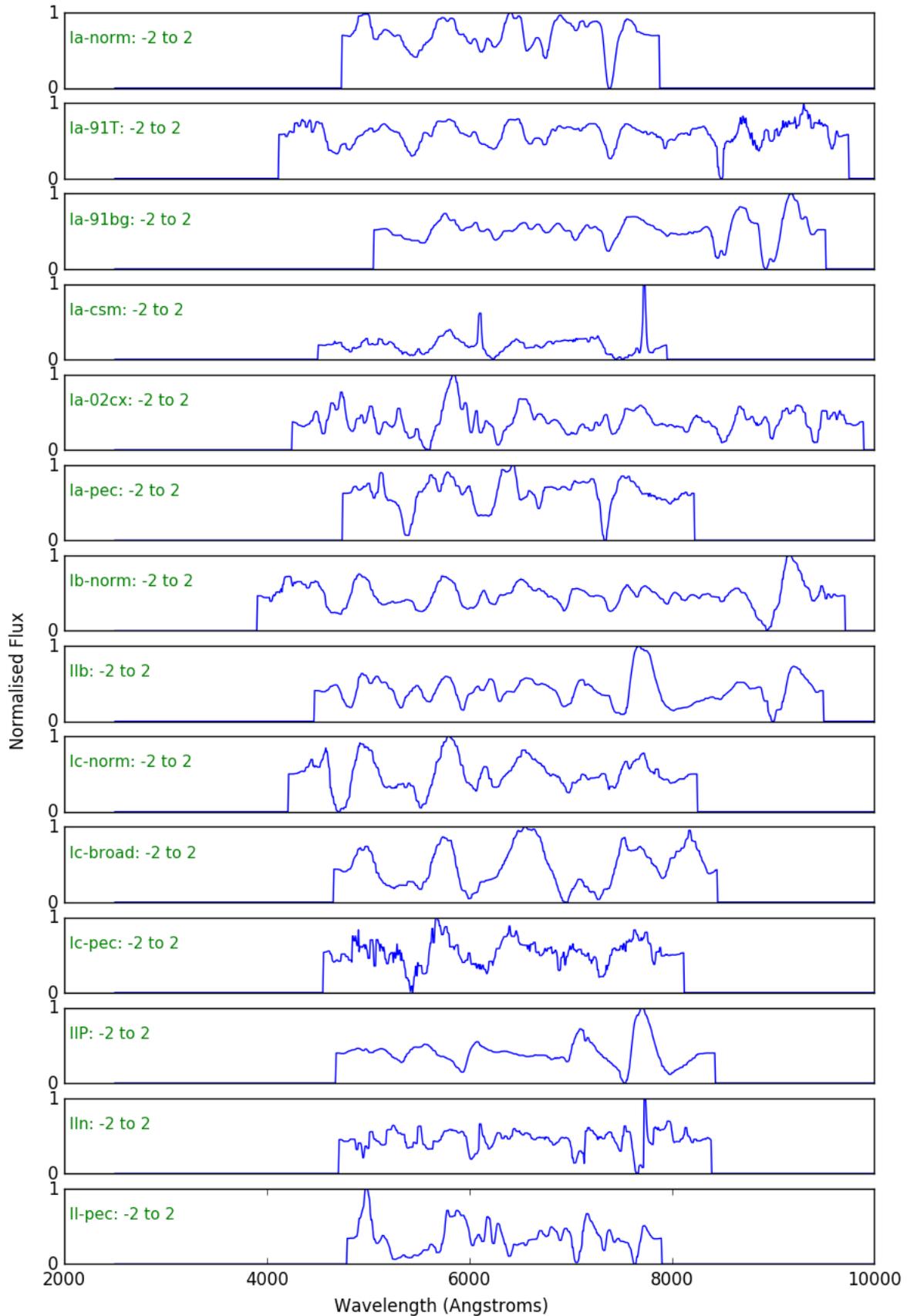


FIGURE 5.1: Plot of each of the different subtypes used in this project at an age between -2 to 2 days (approximately maximum brightness). Note that Ib-pec, IIp, and IIn have not been plotted because they do not have any templates in this age bin (see Table 5.3). The spectra have been preprocessed using the method outlined in section 7.1.

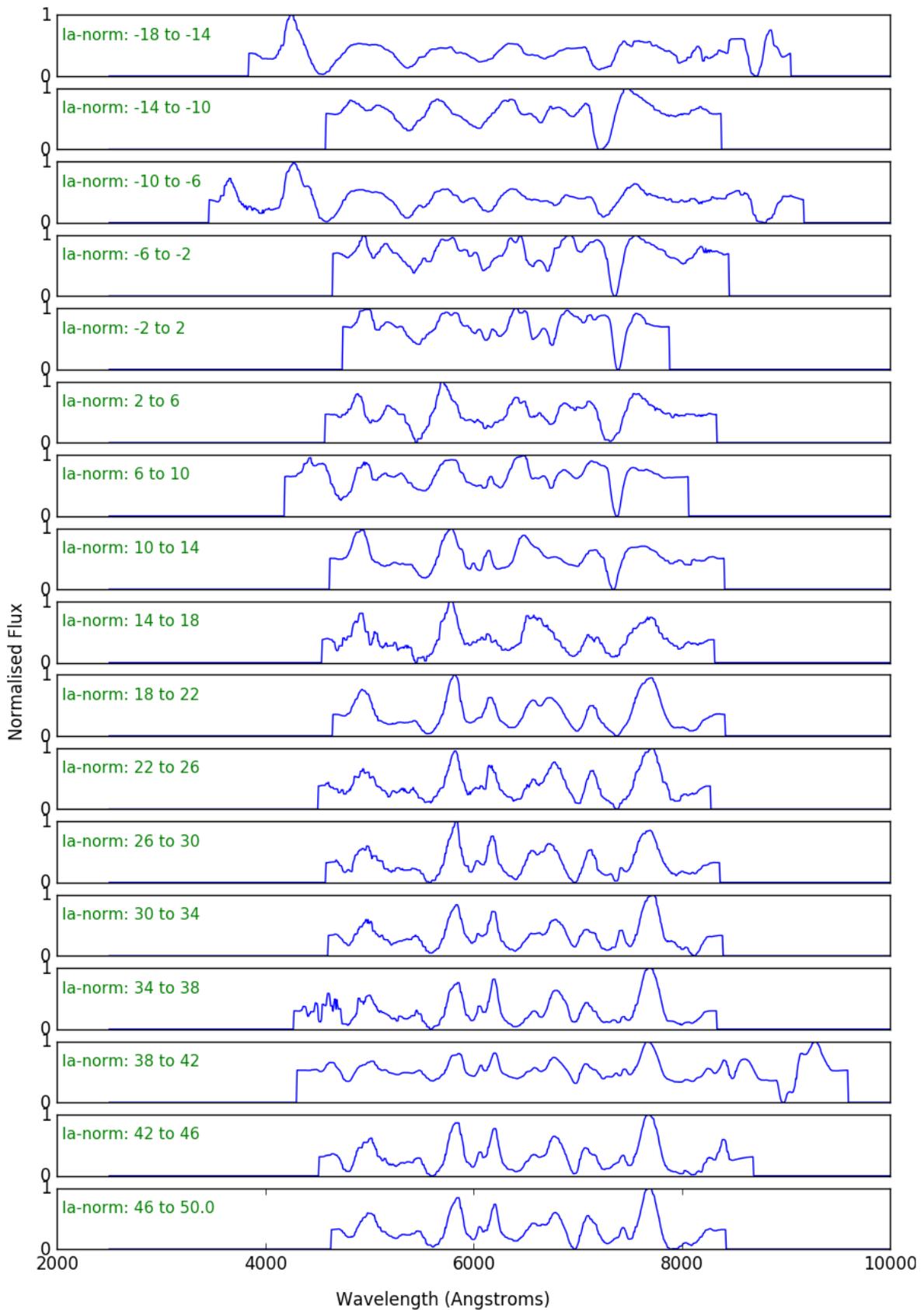


FIGURE 5.2: Plot of the Ia-norm subtype at 17 different ages from -18 to 50 days. The spectra have been preprocessed using the method outlined in section 7.1.

7.1.

Chapter 6

Initial Design and Major Decisions

6.1 Initial Design

At the start of this thesis I originally planned to approach the classification problem with a cross-correlation template matching technique similar to that used by SNID, MARZ and many other redshifting tools. This problem was thought to be too difficult for standard machine learning approaches, and deep learning had never been used for spectral classification as far as we knew, and was thus not considered as an option. I also had no familiarity with the topic, so wasn't able to determine whether machine learning could be a viable option. As such, we turned to the existing software: Superfit and SNID, and aimed to improve upon them. In fact, the original plan by the consulted members of OzDES was to make my software an extension of Sam Hinton's MARZ redshifting program. MARZ's main advancement is its convenience of being online, its very user-friendly appearance, and processing and cross-correlation accuracy compared to previous redshifting tools. A supernova classification tool, however, is a much more involved problem for the reason that there are so many more factors involved in supernova classification than redshifting. In particular, the sharp features in galaxy spectra make redshifting a much more obvious process than the redshifting of supernovae. Furthermore, the degeneracies in type, age, redshift and host contamination make supernova classification a very complicated process.

The main advancements in my software over previous tools was aimed at improving the speed

and to minimise human involvement in classification. The way we planned to achieve this was to recreate Superfit’s ability to subtract galaxies and classify supernovae using a Python framework instead of IDL. Speed improvements were thought possible by enabling more efficient multi-core processing. We developed a flowchart illustrating the top-level design of how my software would deal with the different possible cases of an input spectrum. This is illustrated in Figure 6.1.

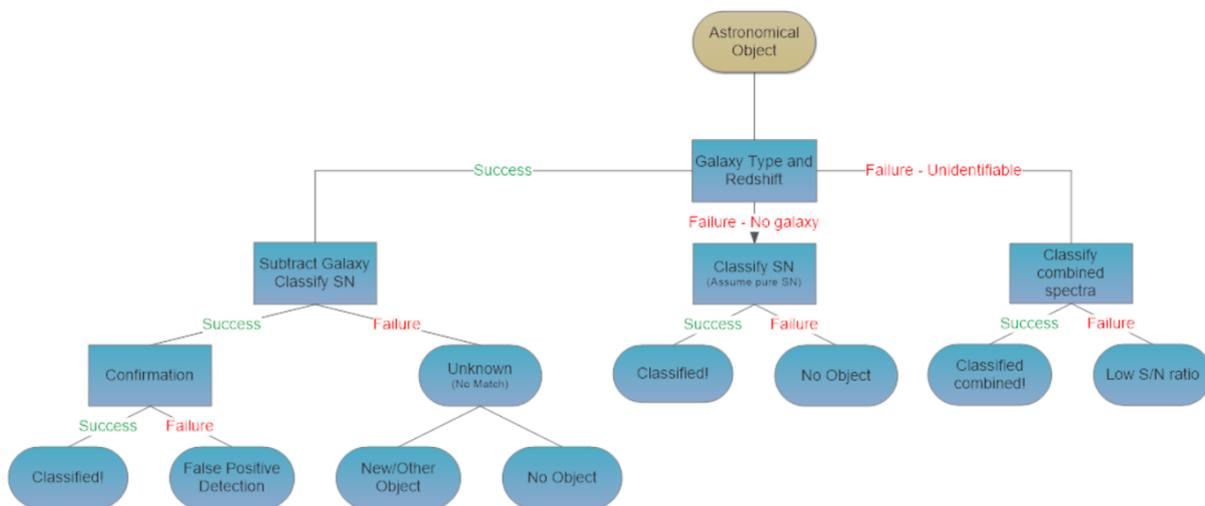


FIGURE 6.1: Process flow diagram indicating the possible outcomes of an input spectrum into my software. This was developed as an initial design.

The flowchart illustrates how an input spectrum would be dealt with in my software. Initially, we would try to determine the galaxy type and the redshift. This is essentially what MARZ does at the moment, so it was not considered a top priority for my software. If the galaxy was successfully determined (left branch) then it would be subtracted from the spectrum, and the supernova would be attempted to be classified. Ideally, this would work, and the program would finish, but we considered that three other possibility could occur. The first possibility was that the supernova classification would fail because there was no supernova in the spectrum; second, some new undiscovered supernova was being input; third the flowchart could move down the middle branch whereby the galaxy light was feint compared to the supernova and thus it could be classified directly. Alternatively, the galaxy and supernova were combined such that the galaxy could not easily be subtracted away. In this case, a set of templates which had both galaxies and supernovae would be need to compare with the input.

6.1.1 Host Galaxy Subtraction

The process explained in the previous section is similar to how superfit deals with classification. The main difficulty with this approach is being able to subtract the galaxy light. This is a difficult problem because it is not known how much galaxy light compared to supernova light is in each wavelength bin of the spectrum. We considered a way to estimate the galaxy light proportion by estimating that the combined spectrum was made up of a linear combination of the host galaxy (Host) and supernova (SN) as in the following equation:

$$\text{Combined} = \alpha(\text{SN}) + \beta(\text{Host}) \quad (6.1)$$

where $\alpha + \beta = 1$ and α and β represent the percentage contribution of the Supernova and Host galaxy spectra to the combined spectrum, respectively. In order to determine α and β so that the host galaxy could be subtracted from the combined spectrum, we would vary the α and β parameter space until we found a combination that matched closest to the combined spectrum. The closest match would be defined by either the combination that maximised the cross-correlation, or the combination that minimised the total chi-squared probability.

6.1.2 Initial Classification Method

For the first couple of months of my thesis I worked to recreate SNID and Superfit. In this time I successfully recreated SNID in Python, and was broadly able to classify supernovae using a cross-correlation and overlap maximising process (see section 3.2.1). However, the time it took my program to classify a single supernova was approximately 80 seconds, compared to SNID which was about 15 seconds. The main reason for the longer time was because of the hundreds of files my python program had to iteratively open and read. Python is notoriously slow at doing this, so I would have had to store the files in memory before classification begun to improve this speed.

The process I used to classify the spectra was as follows. I first processed the input spectrum using the method outlined in section 7.1. Next, the input spectrum was iteratively cross-correlated with each of few thousand templates to determine the redshift and the best matching templates. In order to simplify the computation time when calculating the cross correlation, the convolution theorem was used:

$$s(\lambda) * t(\lambda) = \int_{-\infty}^{\infty} s(\Delta\lambda)t(\lambda - \Delta\lambda) \quad (6.2)$$

$$= \mathcal{F}[S(k)T(k)] \quad (6.3)$$

where $s(\lambda)$ and $t(\lambda)$ are the input and template spectra respectively. λ is the wavelength, k is the wave-number, \mathcal{F} represents the Fourier Transform, and the capitalised $S(k)$ and $T(k)$ represent the input and template spectra in Fourier space. Here, the cross-correlation can be computed with Fourier transforms and multiplications rather than computationally expensive integrals.

The cross-correlation was also multiplied by a low pass filter to remove high frequency noise, and hence improve the signal-to-noise ratio.

6.2 Changing Classification Method

Ultimately, my initial method was successful at classifying supernovae, provided that there was a reasonably high signal-to-noise, and that there was minimal host galaxy contamination. This was an okay result, but I had not at all improved on previous methods. As such, my next goal was to recreate Superfit so that I could deal with host galaxy light being intermixed in the signal. Superfit makes use of a chi-squared minimisation approach, but also has to iteratively compare the input with several templates.

At this point, I also had the novel idea of using a machine learning approach to solve the problem. One of the main advantages of machine learning was that the computation time did not increase linearly with the number of templates, and it had the potential to be much

faster. I considered both supervised and unsupervised learning techniques, but upon research, Deep Learning appeared to be solving a whole new range of Big Data problems. In particular, it has had a huge amount of success in image classification problems, and I thought that this supernova classification problem could be re-defined as a one-dimensional image classification problem. The table below provides a comparison of the different classification methods I was considering.

	Deep Learning	Cross-correlation matching	Chi-squared matching
Classification technique	Matches based on the combined 'features' of all templates	Iteratively compares to templates	Iteratively compares to templates
Speed	Very Fast (no change in speed with templates)	Fast (but increases linearly with number of templates)	Slow (increases linearly with number of templates)
Noise	Can train with noise	Cannot classify low S/N	OK with low S/N
Redshifting	Redshifting is unreliable	Very good at redshifting	OK redshifting
Goodness of Fit	Relative	Absolute	Absolute

TABLE 6.1: A comparison of three different approaches to classifying supernova spectra.

Deep Learning has several advantages compared to the cross-correlation matching used by SNID and many other tools, and the chi-squared matching approach used by Superfit. In particular, Deep Learning uses a very different classification technique. While the other two statistical methods have to iteratively compare an input to a large set of templates, Deep Learning does not need to. It's biggest advantage is that the training process is separate to the testing process. Once a model has been trained with as many templates as possible, the matching is based on the output model, and does not require the original templates at all. As such, it has the possibility of being significantly faster, because the number of templates does not affect the speed like it does for the other methods. In addition to this, as with many image classification programs, deep learning can be trained to recognise a signal from a lot of background noise.

One of the main disadvantages, however, are that cross-correlations are extremely effective at redshifting, whereas deep learning is often position-invariant. This position invariance means that the classification can often find a sub-image no matter where it appears in the larger outline. The other disadvantage, is that unlike the statistical methods of cross-correlation and chi-squared matching which can give an absolute measurement of how good the classification is, deep learning can only give a relative measurement of the goodness of fit by comparing to the other possible fits. This is discussed further in Chapter 9.

Overall, deep learning is new, has never been tried before in this field, has had enormous success in other fields, and appears to be the only way that my software has the potential to be significantly better than alternative methods.

6.3 Deep Learning

6.3.1 Overview

Deep learning is a branch of machine learning that has recently gained a lot of popularity for its success in a range of different applications including image, speech, and language recognition. Advancements in computers have enabled neural networks to solve these more complicated problems in reasonable amounts of time.

Figure 6.2 illustrates a visual representation of a deep neural network. Each layer is in the form of a set of nodes or neurons that represent the data. For example, in this thesis, the input layer is made of 1024 neurons representing the fluxes of an input spectrum. Additional layers of neurons on the right of the input signal are built to ensure that each new layer captures a more abstract representation of the original input layer. Each new hidden layer identifies new features by forming linear and non-linear combinations of the of the previous layer [14]. For example, the hidden layers in the spectral classification in this thesis represent abstract constructions of the input flux. The final output layer will then simply represent 306 different neurons corresponding to the 306 different classification bins of supernova types and ages.

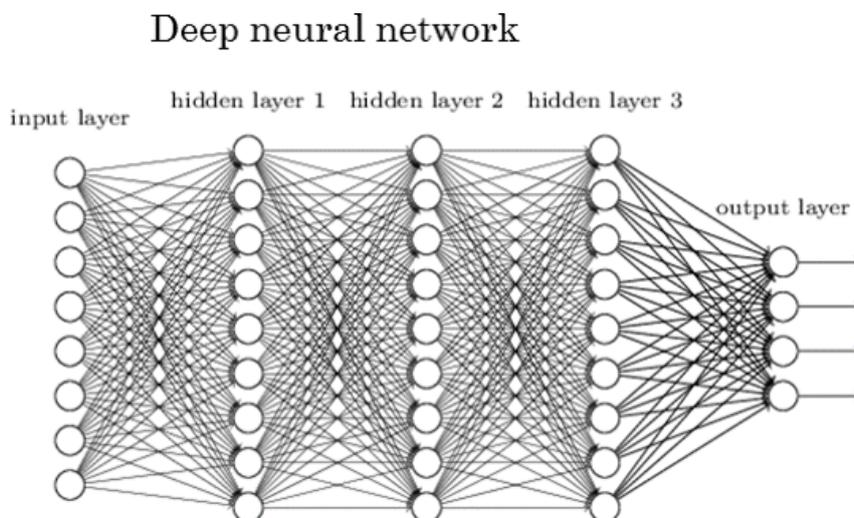


FIGURE 6.2: A representative diagram illustrating the layers of a deep neural network.

6.3.2 Machine Learning in Astronomy

Several successful attempts at machine learning in astronomy have been made [5, 23, 26]. In particular, they have even been applied to supernova classification [23, 26], but have only been successful for photometric data. Spectral classification, on the other hand, is viewed as a problem that is too difficult for standard machine learning algorithms. For photometric classification, the approaches have primarily used supervised algorithms such as Random Forest and Boosted Decision Trees. Deep neural networks have seldom been applied. However, due to the relatively recent advancements in computers, spectral classification is finally a problem that may be solvable by neural networks.

As such this project is very novel in its decision to use deep learning for spectral classification. However, while this thesis was being written, a recent paper by Sasdelli et al. [35] has applied deep learning to supernova spectra. This work has not been applied to supernova classification, but has instead aimed to explore spectroscopic diversity in particular Type-Ia supernovae.

Moreover, a recent Masters thesis by [13] has applied deep learning for the spectral classification of quasars, stars and galaxies. This is a very similar problem to the one in this thesis, and I have thus made use of the lessons learned in this work. This being said, supernovae are inherently more complicated than quasars and many other astronomical objects because of

the fact that they vary with age, and thus have many degeneracies with their type, age and redshift. Moreover, they have the additional problem of being distorted by the light from their host galaxy.

Nonetheless, by combining the techniques and lesson learned by both [35] and [13], I have been able to create a successful Supernova classification tool.

6.3.3 Tensorflow

A few different machine learning libraries exist for Python. These include Theano, Caffe, Lasagne, Keras, Tensorflow, and many others. Tensorflow is the newest neural network library having been released within the last year. It has received widespread praise for its high-level library that avoids low-level details. It allows a programmer to focus on designing the neural network, enables a flexible architecture, has a very fast performance - making use of a C++ backend, and was developed for use in Google products. As such, while many alternatives exist, I have made use of Tensorflow to build the neural networks in this thesis primarily because I was keen to be one of the first programmers involved in the latest deep learning hype.

Using Tensorflow has proven to be a very good decision, as it has been able to classify supernova spectra with unprecedented speed, and very good reliability (see Chapter 9).

Tensorflow relies on a highly efficient C++ backend to do its computation [1]. The connection to the backend is called a session. I have made use of a two layer neural network with a Softmax regression model. The exact methodology is explained further in section 7.3.

Chapter 7

Implementation

This chapter details how the final project has been implemented. The project consists of over 23 python files and several thousands of lines of code which cannot be completely explained. Instead this section provides a detailed overview of how the project has been carried out. It begins by outlining the processing techniques used to prepare the spectral data files, and then explains how the data was prepared for the machine learning. The development of the neural network is then detailed before the training and testing method is explained. Finally, the design of the graphical user interface and the created python library is described.

7.1 Processing Method

SNID, MARZ, and AUTOZ all pre-process their spectra in a similar way before they are ready for cross-correlation and matching to the templates. There are several steps to the processing algorithm used. These consist of the following:

1. Low pass median filtering.
2. Normalising and De-redshifting.
3. Log-wavelength binning.
4. Continuum modelling with spline interpolation.
5. Continuum subtraction.

6. Cosine tapering edges.

Once the data has been read from the .DAT or FITS file, the above method is used for processing so that the input spectrum can be matched to the template data. A real OzDES data file observed at the AAT on 16-September-2016 named DES16C2ma is used as an example for the purpose of illustrating this algorithm. The following subsections detail the process.

7.1.1 Filtering

The first step of the processing algorithm involves low-pass filtering the data to remove the high-frequency noise. Three possible methods were considered to apply the low-pass filter:

1. Blackman Window function
2. Moving-average filter
3. Low-pass median filter

The first method involves first Fourier-transforming the data before multiplying it by a window function with a defined cut-off frequency. The second method involves averaging every n -points along the input vector. The final method is similar, but involves taking the median of every n -points. While the first method is very effective, it is a little computationally expensive, so it was thought of as a last-resort if one of the other two methods were not possible.

One of the main differences between a median filter and an averaging filter is how they deal with very high frequency points that appears as sharp spikes in the spectra. An averaging filter will take the mean, and will thus be affected by a large spike. On the other hand, a median filter ignores sharp spikes by taking the median of n -points in the vector. this makes a low-pass median filter the obvious choice. A plot of the DES16C2ma raw data spectra and the filtered signal are illustrated in Figure 7.1.

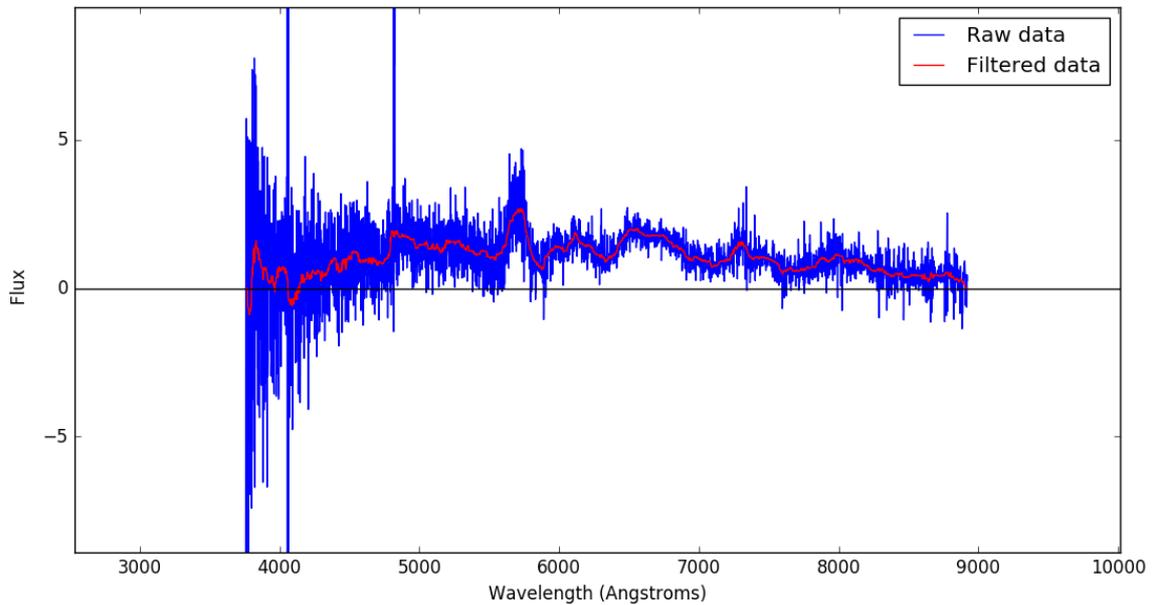


FIGURE 7.1: A plot of an example supernova spectrum from OzDES, DES16C2ma observed on 16 September 2016. The blue line shows the raw data spectrum, while the red line shows the result after a low-pass median filter has been applied. Note that the units of flux are not important, but only the relative contributions of each feature.

7.1.2 Normalising and De-redshifting

The next stage involves de-redshifting the spectrum. This is an optional stage depending on whether the redshift agnostic or zero-redshift model is used (see section 7.4). The data is de-redshifted to account for how much the light was stretched due to the universe's expansion. In order to de-redshift the spectrum the following equation was applied:

$$\lambda_{emitted} = \frac{\lambda_{observed}}{z + 1}. \quad (7.1)$$

My software deals with redshift by either iteratively adjusting the redshift estimate, or by enabling the user to input a redshift. For the purposes of this explanation, the redshift was determined by using MARZ on the spectrum, and was determined to be $z = 0.24$.

After de-redshifting, the spectrum is normalised so that the flux is vertically shifted up to the positive range, and then divided by the maximum flux range so that the normalised flux is

between 0 and 1. This normalisation was achieved using the following equation,

$$\text{flux}_{\text{normalised}} = \frac{\text{flux} - \text{flux}_{\text{min}}}{\text{flux}_{\text{max}} - \text{flux}_{\text{min}}}. \quad (7.2)$$

The filtered data from the previous subsection is plotted along with the de-redshifted and normalised spectrum in Figure 7.2.

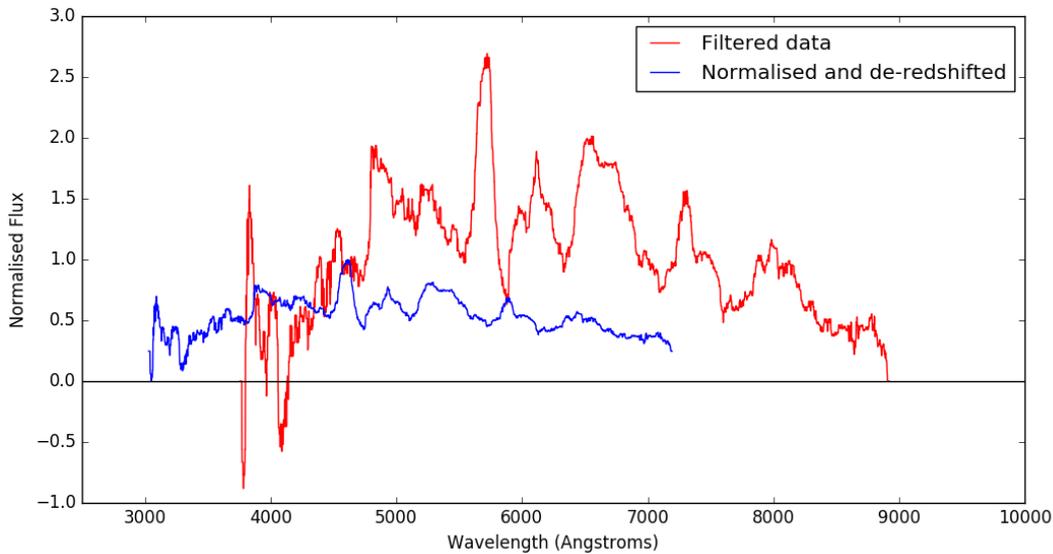


FIGURE 7.2: A plot of the filtered data (DES16C2ma) from Figure 7.1 shown in red. The spectrum has then been de-redshifted and normalised and is shown in blue.

7.1.3 Log-wavelength Binning

The spectrum is then binned onto a log-wavelength scale with a fixed number of points (bins) within 2500 to 10000 Angstroms. This step is important for a few reasons. Firstly, ensuring that each spectrum is a vector of exactly the same length within the same wavelength range makes comparison much easier. Secondly, the primary reason for the log-wavelength scale is so that it is consistent with the SNID templates. Moreover, it also makes redshifting less computationally intensive since multiplying the signal by the redshift ($1 + z$) is equivalent to adding a $\ln(1 + z)$ shift to the logarithmic wavelength axis [7]. The binning process used in this project follows the method outlined in [7]; some of the key steps are shown here.

First, the log-wavelength axis, $l_{\log,n}$, is defined as:

$$l_{\log,n} = l_0 \ln e^{n \times dl_{\log}}, \quad (7.3)$$

where $l_0 = 2500$ is the minimum wavelength, $l_1 = 10000$ is the maximum wavelength, $N = 1024$ is the number of bins, n runs from 0 to N and

$$dl_{\log} = \ln(l_1/l_0)/N \quad (7.4)$$

is the size of a logarithmic wavelength bin. The binned wavelength can then be translated from the normal wavelength with the following relationship,

$$\text{binnedwave} = A \ln l_{\log,n} + B \quad (7.5)$$

where $A = N/\ln(l_1/l_0)$ and $B = -N \ln l_0/\ln(l_1/l_0)$.

Using this method, the input and template spectra were binned onto this scale. The binned spectrum is illustrated as the blue line in Figure 7.3.

7.1.4 Continuum Modelling with Spline Interpolation

The next step in preparing the spectra involves subtracting the continuum. For galaxy spectra, the continuum is well defined and is easily removed using a least-squares polynomial fit. In supernova spectra, however, the apparent continuum is ill-defined due to the domination of bound-bound transitions in the total opacity [32]. For this reason, a 13-point cubic spline interpolation is used to model continuum. The 13 points was thought to be sufficient to interpolate the spectrum. From the 13 points, a cubic spline is then extrapolated such that the continuum is modelled. This is illustrated as the green line on Figure 7.3.

This continuum is then subtracted from the spectrum (red line). This step removes any spectral colour information (including flux miscalibrations), and enables the correlation to rely purely

on the relative shape and strength of spectral features in the spectra. According to [7], the loss of colour information has very little impact on the redshift and age determination.

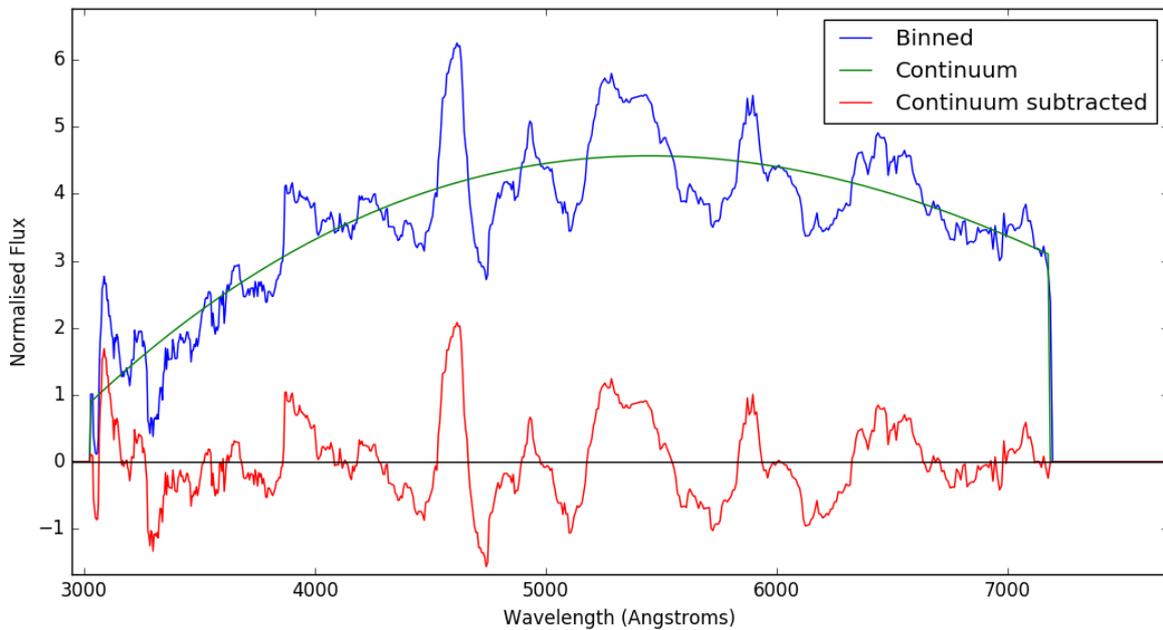


FIGURE 7.3: A plot of the binned wavelength (blue), the modelled continuum (green) and the continuum subtracted spectrum (red). These steps have been applied to the same DES16C2ma spectrum that has been used in Figures 7.1 and 7.2.

7.1.5 Cosine tapering edges

While the discontinuities at each end of the spectrum are limited by the continuum subtraction, further discontinuities are removed by apodizing the spectrum with a cosine bell. This involves multiplying 5% of each end of the spectrum by a cosine, to remove sharp spikes. This is illustrated as the green line in Figure 7.4. Finally, the spectrum is renormalised to positive values between 0 and 1 (red line).

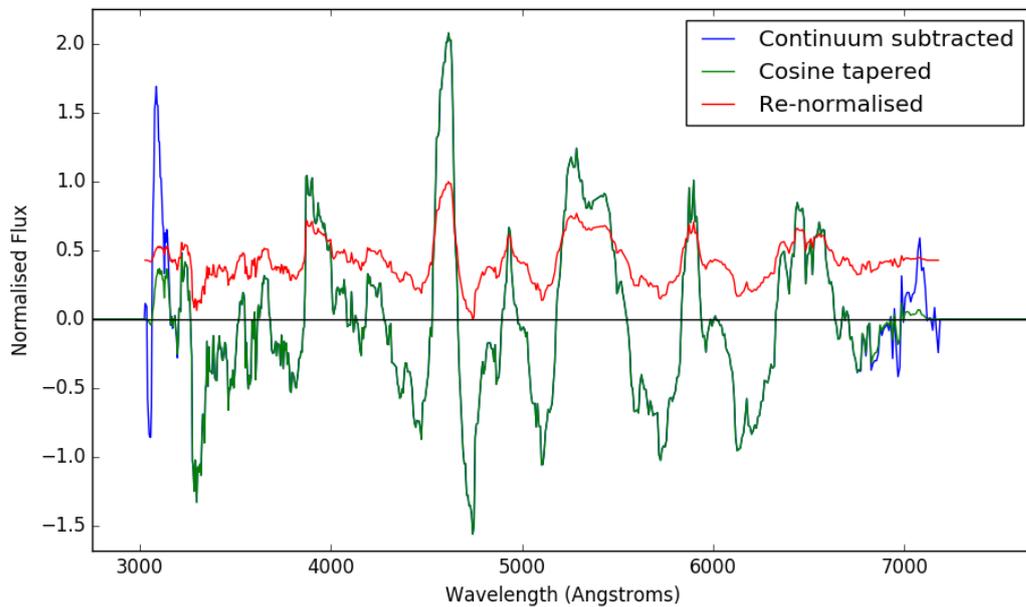


FIGURE 7.4: A plot of the cosine-tapered spectrum (green) which removes the edge discontinuities from the previous spectrum (blue). The red line is a normalised version of the green spectrum. It ensures that the flux is between 0 and 1. This process has been applied to the same DES16C2ma spectrum used in Figures 7.1 to 7.3.

7.2 Preparing Training Data

One of the most influential factors in an effective learning algorithm is ensuring that the training set is of a good quality and are consistent [13]. As such a lot of the software effort in this project has been in ensuring that the data is well prepared before it is sent for training using Tensorflow. When applying a machine learning tool, it is important that the problem that needs to be solved is well defined. I have phrased this spectral classification problem as a standard one-dimensional image classification problem, where the fluxes correspond to pixel intensities. This has allowed me to use a very similar method to that used to solve the common MNIST classification problem.

7.2.1 Label and Image Data

As outlined in section 5.4.2 and Table 5.3, 306 different classification bins have been defined. These correspond to the 17 different subtypes and each of their 18 corresponding age bins. Each data template has two important properties: its label and image data. The image data is simply made up of a 1024 point vector that corresponds to its normalised flux values. The labels correspond to one of the 306 different classification bins. Instead of defining the label as a string, it has been defined as a one-hot vector. This one-hot vector is 306 elements long, and filled with zeros except for one entry which has a one (hence, the name). Each cell represents a different classification bin, and the cell with a one indicates that the image data corresponds to that bin. The advantage of using one-hot vectors is so that matrix multiplication can be used when training the data.

7.2.2 Template sample bias

One common issue in classification problems is that some bins are much more populated than others. This is an extremely significant problem in this thesis project as illustrated in Table 5.3. If this is not dealt with before training, then the machine learning algorithm will favour the more populated bins because it has a higher chance of being correct relative to rest of the training set.

There are three common ways to deal with this problem:

1. Undersampling
2. Oversampling
3. Adding weights

The first method involves undersampling or reducing the number of templates in the bin that is over populated. This is only a good solution when each bin is already well populated. However, in this project, the main problem is a lack of data, so removing data from an already small

dataset by undersampling is not ideal. The alternatives are to keep the whole dataset but oversample the bins that are under-populated by repeating them several times, or enough times to match the most populated bin. The other possibility is to add an additional scaling weight to the underpopulated bins so that the machine learning model is not biased towards particular bins. This would be a valid method, but I have chosen to oversample instead because it can be applied before training has even started.

However, while oversampling, it was important that the overall arrays that contained the image and label data were shuffled to a random order before passing them into the training model. This is so that the repeated spectra were not trained during the same training epoch.

7.3 Deep Learning Training

The multilayer neural network was developed using Tensorflow's python library [1]. The process of building the neural network is outlined in the following subsections.

7.3.1 Softmax Regressions

The deep learning model is trained on 306 different classification bins (see Chapter 5). In order for the program to be able to rank each of these options for a given input spectrum, we need to be able to define probabilities for each of the 306 bins. A Softmax regression enables this by generalising a logistic regression to the case where it can handle multiple classes [41]. It involves two steps, first it adds up the evidence of an input spectrum being in a particular classification bin, and then converts that evidence into probabilities. In order to add the evidence for each bin, a weighted sum of the 1024-point input vector was prepared following the method outlined in the previous section. This is illustrated in the following equation

$$\text{evidence}_i = \sum_{j=1}^{1024} W_{i,j} x_j + b_i \quad (7.6)$$

where W_i is the weights, x is the input layer, i is a number from 1 to 306 indicating the classification bin, j runs from 1 to 1024 to sum over the entire input vector, and b_i is a bias that is added to allow some points in the vector be more independent of the input. Then, the evidence tallies for each classification bin can be converted into Softmax regression probabilities, y , by applying a Softmax function:

$$y = \text{Softmax}(\text{evidence}) \quad (7.7)$$

where the Softmax function is defined as

$$\text{Softmax}(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}}. \quad (7.8)$$

This function effectively normalises the evidence so that the total probabilities of all the classification bins sums to 1. These Softmax probabilities are very important in this thesis because they provide a ranking of all the best matching classification bins. However, it is important to note that these probabilities only state the relative chance that the particular classification bin is one of the 306 options. As such, it does not provide an absolute measurement of whether the object is a good fit, but only that it fits a particular bin more than the others.

The weights and biases are free variables that are computed by Tensorflow during the training process. However, in order to train the model, a 'cost-function' is used to define what it means for the model to have an accurate prediction. A common cost-function used in machine learning is the cross-entropy, defined as

$$H'_y(y) = - \sum_{i=1}^{306} y'_i \log(y_i) \quad (7.9)$$

where y is the model's prediction of the probability distribution, and y' is the true distribution. Effectively, y' is a 306-point one-hot vector with zeros in all entries, except for one which is filled by a 1. This entry represents the classification bin of the input spectrum. y is a 306-point vector that represents the model's prediction of the distribution. The sum of all the entries adds to 1, and ideally, the entry with the highest probability would be the same as the entry with a 1

in y' . Thus, the cross-entropy equation measures how inefficient the predictions are compared to the truth.

Equation 7.9 is minimised during the training using a Gradient Descent Optimizer called Adam Optimiser.

7.3.2 Building Layers

I have developed a neural network with two hidden layers, one input layer, and one output layer. After some testing I found that two hidden layers gave a significantly better accuracy than a single hidden layer (up to 10% better). I also tried using a 3-layer model but found that there was relatively no improvement. As such two hidden layers were used in the model.

The layers were built by defining weights and biases for each layer based on the previous layer. The input and two hidden layers were convolved after their weights and biases were defined, and the Softmax regression probabilities were calculated after minimising the total cross-entropy using an Adam Optimiser function.

7.4 Trained Models

A few different types of models were trained during this thesis. The first was the most successful one which involved building the training set from all templates de-redshifted back to $z = 0$. The second involved building a training set from a template set with a range of different redshifts.

7.4.1 Trained at zero redshift

This model involved using the templates described in Chapter 5. The 306 classification bins were used, and in each bin, the templates were all kept at a redshift of $z = 0$. This means that the deep learning model was only trained to match an input spectrum if it was at redshift zero.

As such, any input spectrum needs to be de-redshifted back to zero before it can be passed into the matching algorithm. There are two ways to deal with this. Firstly, the user would have to know the redshift beforehand and input it as a prior so that the input spectrum can be de-redshifted before it is compared to the model.

Secondly, the algorithm could iteratively de-redshift the spectrum by varying amounts and pass several versions of the input spectrum at varying redshifts. For example, the input spectrum would be de-redshifted by $z = 0.1, 0.2, 0.3, 0.4, 0.5$, and the five differently redshifted spectra would be passed into the matching algorithm. For each of these five spectra, there would be a separate list for the best matching classification bins. The issue then is how to rank these differently redshifted spectra. I chose to combine the five lists, and rank them based on their Softmax probabilities. Each classification bin would appear multiple times on the list, and I chose to only use the highest Softmax probability for each classification bin in the Best Matches List. A plot of the probability vs redshift for one of these classification bins is illustrated in Figure 7.5. In that figure I used 5000 different redshifts, instead of the 5 used in the example I have been discussing.

Based on the plot, there is one major issue with this method. That issue is that we are trying to compare Softmax probabilities from the different lists. This is not a reasonable comparison because the Softmax probabilities are relative to the other probabilities in each of the five lists.

The method used for redshifting with this model is very complicated, and may not easily be understood by the above explanation given. Nonetheless, the main takeaway message, is that this method cannot be considered a valid approach to redshifting. As such a better method is needed, and one is given in the following subsection. For this reason, this model is only used if the redshift is known and input as a prior by the user.

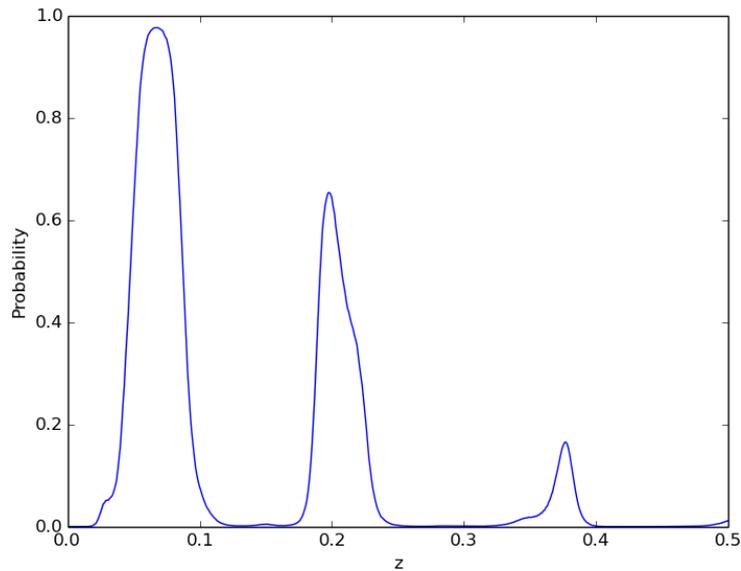


FIGURE 7.5: A plot illustrating the issues with determining redshift using the zero-redshift model. The plot represents the probability of each redshift for a particular arbitrary classification bin. The redshift with the highest probability is $z = 0.076$.

7.4.2 Agnostic Redshift

A second approach to redshifting involves using a training set that includes template spectra at a range of different redshifts. This would make the model invariant or agnostic to the redshift of an input spectrum. Ideally, an input spectrum could be classified into one of the 306 classification bins without needing to know the redshift.

This model has been trained by using the template spectra described in Chapter 5, and iteratively redshifting each of the templates from $z = 0$ to $z = 0.5$ in $\Delta z = 0.01$ intervals. We still make use of the 306 classification bins, but now in each of those bins there are a 500 times as many spectral templates due to the fact that they have been shifted to 500 different redshifts.

While in theory this should work very well to make the classification algorithm invariant to changes in redshift, several problems have been encountered that have not enabled this method to be completely functional at this stage. One of the largest problems is that the training data has a significantly larger dataset. As such, my laptop is not able to finish training without

running out of memory. For this reason, I have made use of the Obelix Supercomputer at UQ and have made use of 20 cores with a lot more RAM available. I have also reduced the size of the dataset to have a lower redshift precision so that it is only 10-20 times larger than the zero-redshift model. This has had some success, as illustrated in Chapter 8 but trains at a significantly slower rate. In fact, training can take up to a few days to converge to a stable accuracy. (Note that while training may take days, the classification process can complete in a few seconds (see section 6.2)).

Nonetheless, a framework has been developed to determine redshift once this model is appropriately trained. This model is able to classify a spectrum into one of the 306 classification bins without knowing the redshift. However, once the type and age of the supernova has been determined, the redshift can then be calculated by cross-correlating the input spectrum with the best matching template. Similar to the method discussed in section 6.1.2 the redshift that gives rise to the highest cross-correlation will be the redshift of the input spectrum.

7.5 Interfaces

In accordance with the requirements and requests from members of the OzDES collaboration (see section 4.6), two different interfaces have been provided in the project. The main one for the purposes of OzDES is the developed graphical user interface. A python library available on PyPI has also been made available so that users can more easily incorporate the results of the classification into their work.

7.5.1 Graphical User Interface

I aimed to make the user interface as user-friendly as possible, with minimal clutter. The initial landing page when a user first runs DASH is illustrated in Figure 7.6.

Instructions detailing how to start using the GUI have been provided in the README.md file provided with the software. However, even without reading the file, several prompts have

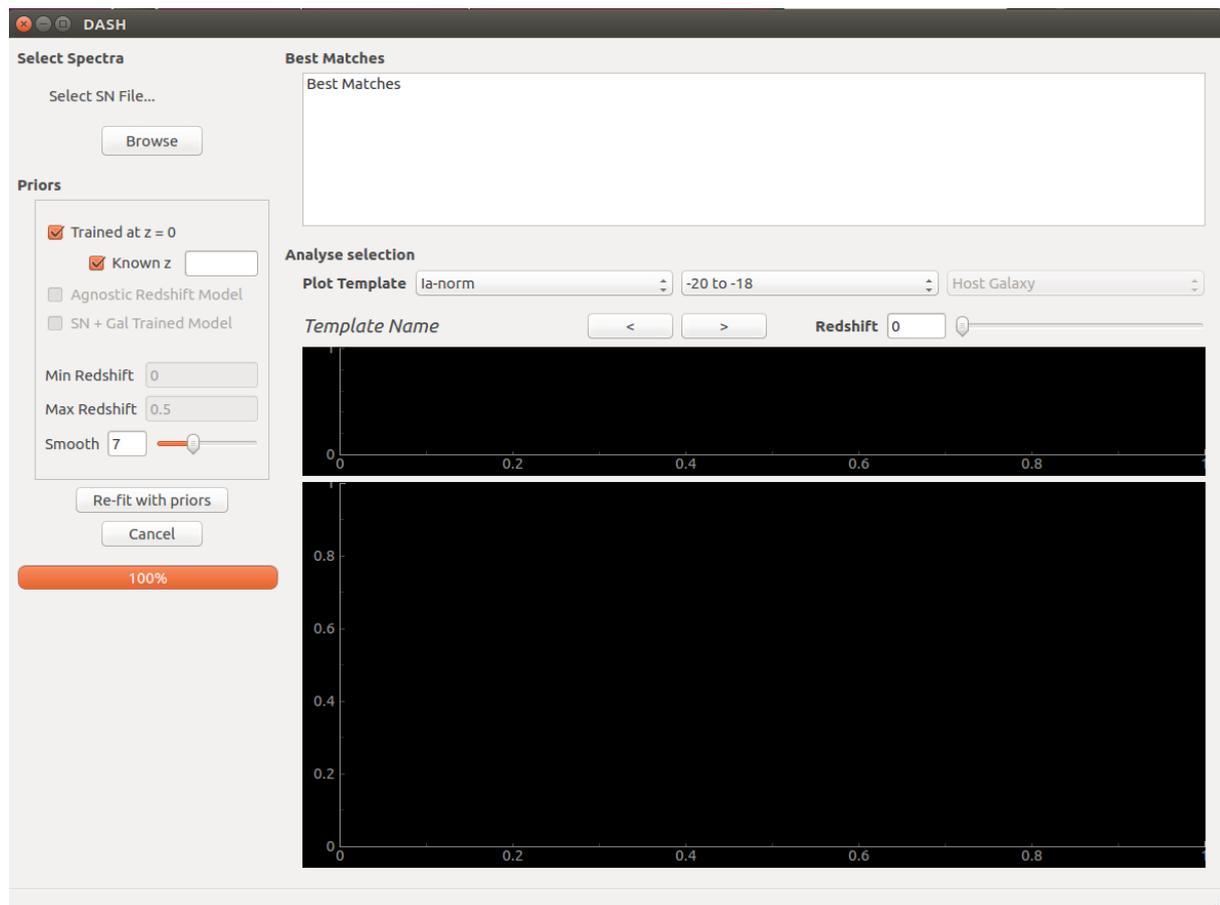


FIGURE 7.6: The initial landing page displayed to a user after they open DASH.

been designed to guide the user. In the top left, the user is asked to select a supernova data file. Clicking "Browse" leads them to a directory explorer where they are prompted to select a spectrum file. An example on an Ubuntu theme is illustrated in Figure 7.7.

Once the user has selected a file, they are able to set the "Priors". If they continue without setting a redshift, they are prompted to select a redshift before continuing, this is illustrated in Figure 7.8.

After a file and redshift has been selected by the user, the right side of the GUI populates with a list of best matches, template information, and graphs. Figure 7.9 provides an example of the final interface. Each of the important components have been labelled and their corresponding functions are described below.

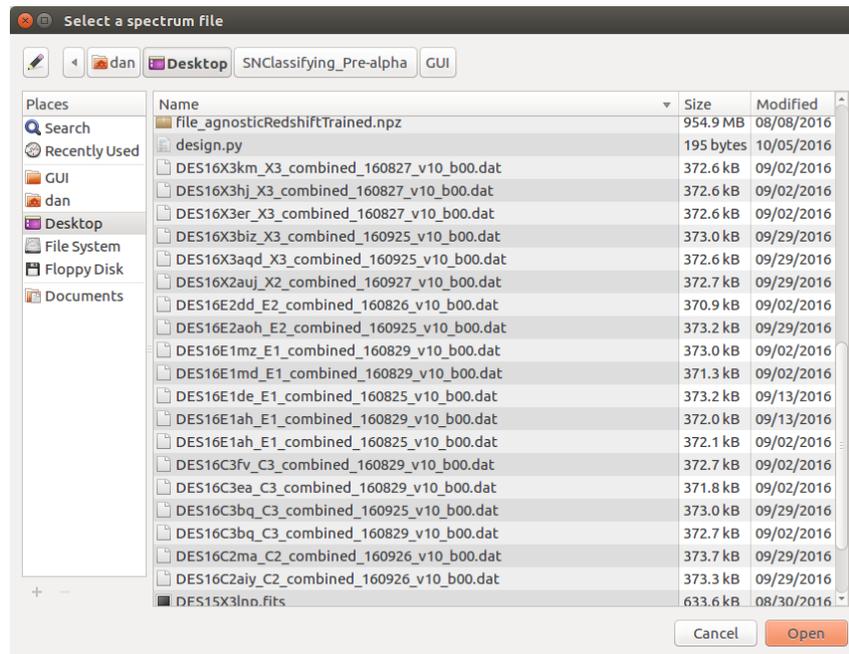


FIGURE 7.7: After clicking "Browse" on the landing page, the user is prompted to select a spectrum file. This is an example of how the user will be prompted. Note that the examples use an Ubuntu theme.

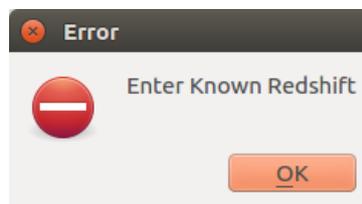


FIGURE 7.8: Message prompting the user to select a redshift before continuing.

Label Guide:

1. **Select a spectrum:** The user is prompted to browse for a file, and the file name is labelled above the button.
2. **Select input redshift:** The type of model can be selected by the user. In this case the model: "Trained at $z=0$ " has been selected, and the user is prompted to select the known redshift of the input spectrum.
3. **Select model:** The user can select the Agnostic Redshift Model or the model trained with the combined supernova and galaxy spectra. If the Agnostic Model is chosen, the user is also prompted to select the known redshift range of the input supernova. These options

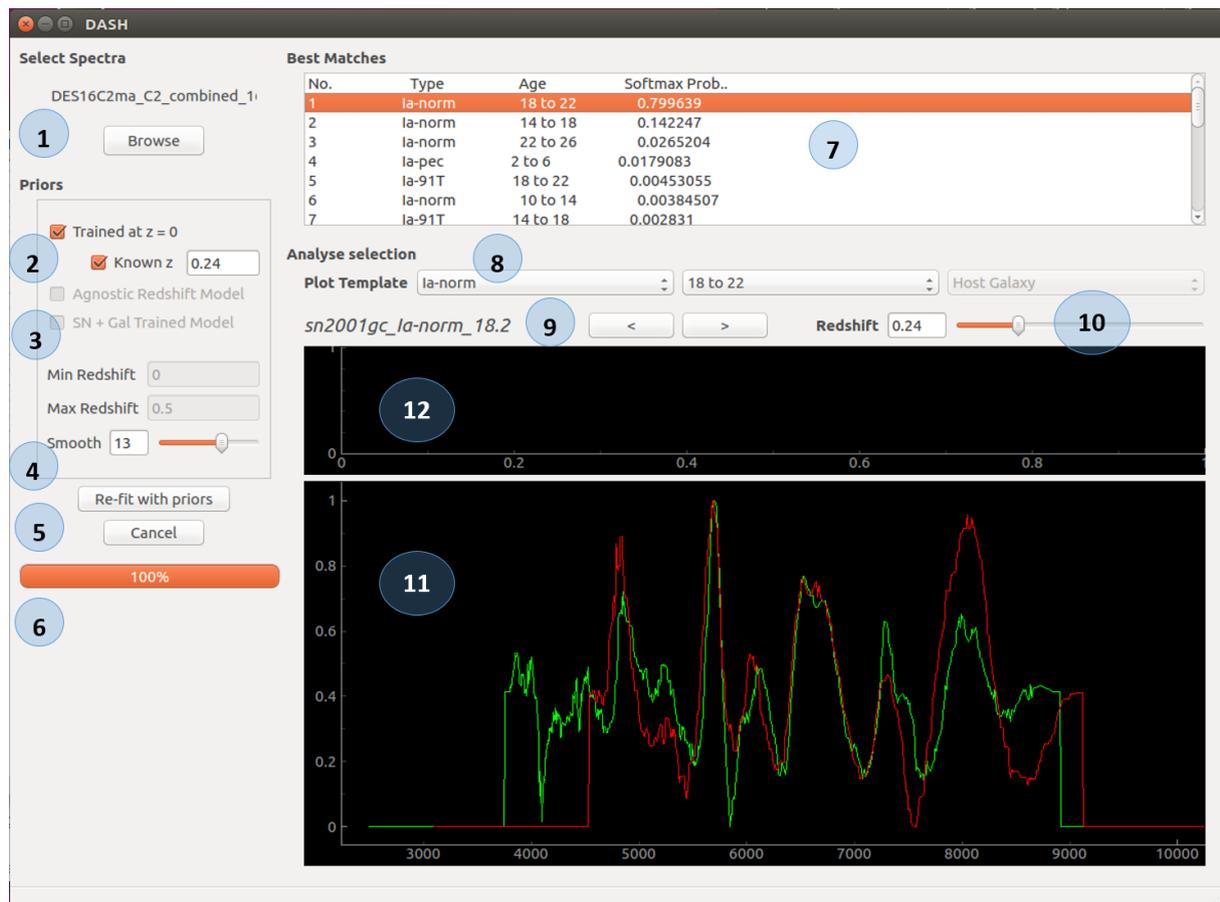


FIGURE 7.9: A labelled example of a spectral file (DES16C2ma) being correctly classified.

have been disabled because the models do not provide accurate results at the time of writing.

4. **Smoothing:** The level of smoothing can be changed with the slider or text box. No smoothing corresponds to zero, and would give the raw processed spectrum, while higher smoothing uses a low-pass median filter with a lower frequency cut-off.
5. **Re-fit:** Once the redshift, smoothing or and other priors have been updated, the user can click "Re-fit with priors" to recalculate the best matches. An option to cancel is also available.
6. **Progress bar:** A loading progress bar provides the status of the matching process.
7. **Best matches:** A list of the best matching supernova classification bins are listed in this

space. The columns represent in order: ranking of the best match, type of supernova, age of supernova, probability of this classification bin as provided by the deep learning Softmax regression analysis. The user can highlight different options to quickly update which of the classification templates are illustrated in the graphing section below.

8. **Select plotting template:** If the user wants to test other classification bins instead of the ones ranked in the Best Matches section above, they can use the drop-down menus to select the supernova type, age. They can also add a host galaxy template to view on the graphing pane below.
9. **Select other supernova templates:** The user can use the left and right arrow buttons provided to select the other templates within the selected classification bin. This allows the user to view what other templates within this bin the algorithm has trained on. The name of the supernova, its type, and age are labelled on the left of the button separated by underscores.
10. **Chang redshift:** This redshift is automatically updated to match the user's input redshift selected in the Priors. However, the user can use the slider or the text box to change the redshift so that they can analyse whether a different match is possible.
11. **Spectral graphing:** This is the main graphing pane. It illustrates the input spectrum (green) and the selected template spectrum (red). The horizontal axis is the wavelength in Angstroms, while the vertical axis is the normalised flux. The graphed spectra have been pre-processed using the method outlined in section 7.1.
12. **Best matching redshifts:** If the agnostic redshift model, or a known redshift has not been selected for the input spectrum, then DASH tries to calculate the redshift. This graphing pane plots the probability of the possible redshifts. The horizontal axis is the possible redshifts, while the vertical axis is the probability that the input spectrum is at that redshift.

7.5.2 DASH Python Library

While the GUI was the main interface developed for OzDES, a secondary pythonic interface has also been made available. I have uploaded the DASH project to the Python Packaging Index (PyPI) so that it can be easily installed with 'pip'. Typing the following into the command line will install DASH onto a machine.

```
pip install deepstars
```

Note that I plan to replace "deepstars" with "DASH" in the near future. The software can also be downloaded straight from the GitHub repository instead. The advantage of using 'pip install' is that the main dependencies including numpy, scipy, and specutils are automatically packaged and installed. However, the user will need to install Tensorflow and PyQt independently.

Once the library has been installed, it can be imported into any python script with

```
import deepstars
```

While the library is still in progress upon the request of other members of OzDES, a few library functions are available to quickly classify an input spectrum with a known redshift. An example usage is illustrated below:

```
classification = deepstars.SNClassify(filename, redshift)
print(classification.best_matches())
```

In the above python code example, 'classification' is an object of 'deepstars'. The class 'SNClassify' takes the file name of a input spectrum and a corresponding redshift. It then returns the 'classification' object, where the list of best matching templates can be printed using the next line. In the second line, 'best_matches' is a function that returns a list of the best matching classification bins with their corresponding Softmax regression probabilities.

Chapter 8

Performance

This chapter details the final results of the developed matching algorithm. The algorithm is first tested against the validation set of SNID templates, and is then test on actual data taken at the AAT for OzDES within the last two months. In this second case, DASH can be directly compared to outputs of SNID and Superfit.

8.1 Validation Set

From the total number of templates described in chapter 5, 80% were used for training the deep learning algorithm and 20% were left for testing and validation. The training was deemed to have finished once the training set was correctly classified 100% of the time by the algorithm, and the validation set appeared to have plateaued.

The performance of the project on the validation set against a few different criteria is listed in Table 8.1. The criteria are defined as follows:

Type: Correct broad type (i.e. Ia, Ib, Ic, II) identified by the matching algorithm.

Type (Ignoring Ib/c mismatches): Correct broad type identified by the matching algorithm if Ib and Ic types being misclassified with each other are ignored. Another way of phrasing this, is that this criterion defines three broad types: Ia, Ib/c, and II, instead of four.

Subtype: Correct subtype (i.e. Ia-norm, Ib-pec, Ib-norm, etc.) identified by the matching algorithm.

Type and Age: Correct broad type and the correct age bin identified by the matching algorithm.

Type: Correct subtype and the correct age bin identified by the matching algorithm.

In the above criteria, a correctly identified match is strict in that it only refers to the classification bin with the highest Softmax probability (i.e. the highest ranking match in the Best Matches List). The percentage of each criteria that was correctly classified in the validation set is shown in the table below.

Criteria	Correctly Classified
Type	98%
Type (Ignoring Ib/c mismatches)	100%
Subtype	93%
Type and Age	91%
Subtype and Age	87%

TABLE 8.1: Performance of DASH on the validation set based on four different criteria. Type refers to the broad type (i.e. Ia, Ib, Ic, II), while subtype refers to one of the 17 different types defined in chapter 5. Subtype and age refers to the algorithm choosing exactly the correct classification bin with the age correct. Type and age refers to choosing the correct broad type and age.

For the purposes of OzDES the most important criteria is being able to distinguish the broad type of the supernova. For the validation set, DASH gets this correct in 97.5% of the templates tested. However, analysing this closer, the only mismatches of the type in the validation set were type Ib and type Ic misclassifications with each other. These two type are often seen to be very similar by astronomers [25, 21, 7], and they are often combined into a single broad type Ib/c. DASH correctly classifies 100% of the validation templates under this criterion.

For the purposes of cosmological surveys, the primary goal is to confirm whether the observed object is a Type-Ia supernova or not. Under this definition, DASH is 100% effective for this validation set.

8.1.1 Alternative Models

The above validation results refer to the training of the model trained with all templates de-redshifted to $z = 0$. However, as outlined in section 7.4, a second redshift agnostic model was also attempted in order to allow DASH to predict redshift. The main issue with this, however, was that the agnostic model had a significantly larger and more complicated dataset which meant that it took a lot longer to converge. In fact, using the Obelix Supercomputer at UQ with 20 cores still meant that the algorithm would take a few days to a week to complete an accurately trained model.

Figure 8.1 plots the correctly classified percentage of the supernova type against the training epoch. The training epoch is an arbitrary scale that indicates the amount of training that each model received. The figure indicates that the zero-redshift model converged and plateaued to 97.5% for the percentage of correctly classified types. However, the Agnostic Redshift model is much slower to converge, and also crashes after a lot of training

Hence, one of the main reasons that I was not able to complete the redshift agnostic model within this thesis time frame was that the Tensorflow model would crash after too much training. Upon research, it appears that this is a bug in Tensorflow, and can often be fixed by providing more memory to the program. The size of the training data is around 10 gigabytes. While this is not too large for the Obelix Supercomputer, I think that the Tensorflow model stores a lot of extra memory in RAM without deletion while it trains. Obelix only enables a user to set the number of cores used, and the RAM provided is proportional to this. To train the model illustrated in Figure 8.1 I made use of 40 cores. However, this may not have been enough due to the amount of memory that Tensorflow stores during its training. As such, I think future work will need to provide the training model with a significantly larger amount of memory so that it can train for a longer period of time without crashing.

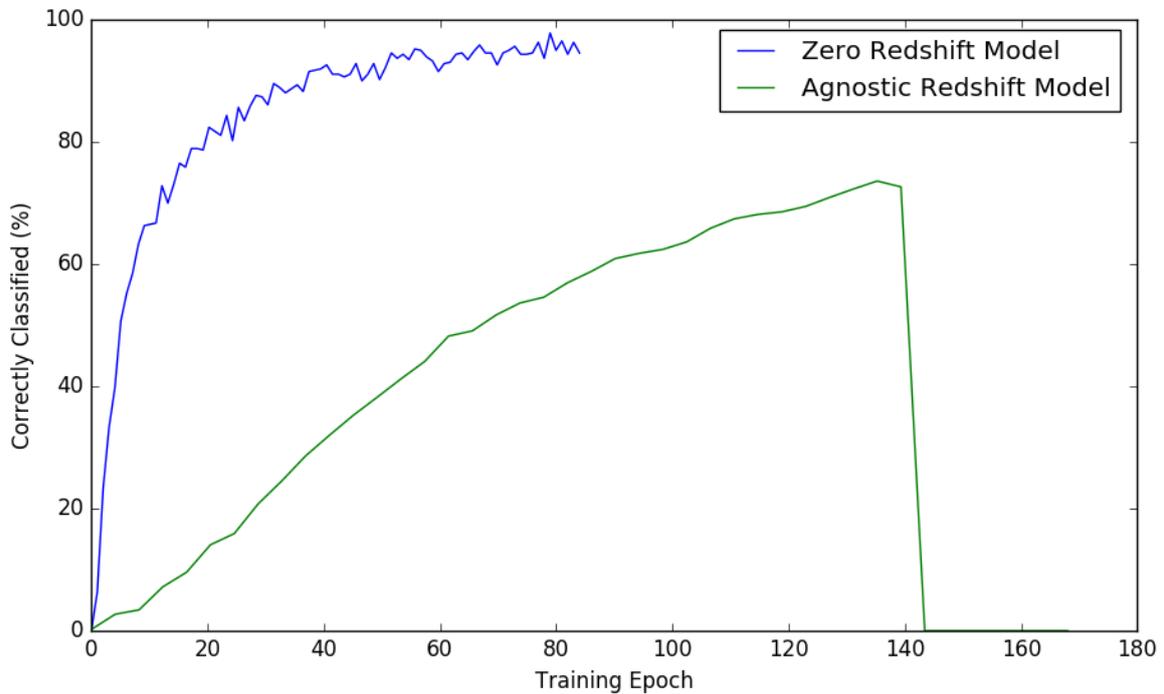


FIGURE 8.1: Plot of the accuracy against training epoch for the two different trained models. The training epoch refers to the relative amount of training that each model received. The Agnostic Redshift model takes a longer time to train and crashes after too much training.

8.2 Results with OzDES data

Testing with the validation set which is made up of clean supernova templates does not give a very good indication of how DASH will function under real data. As such, I have used 23 spectra from the most recent results (Run024 and Run025) of the OzDES survey. This enables DASH to be directly compared to both Superfit and SNID. I have used the same redshift obtained by MARZ as an input. Table 8.2 is a list of the latest Astronomical Telegrams (ATELs) released by OzDES [40, 18, 30, 29]. These have been classified by Dr Chris Lidman and Dr Brad Tucker using a combination of Superfit and SNID.

Table 8.2 compares the classification stated on the ATEL with the classification by DASH. While

accuracy is important, it should be noted that the speed of classification of DASH is significantly better than SNID and Superfit. While the ATEL had to classify each spectrum individually, the DASH python library enabled me to classify all 23 spectra with two lines of code. This DASH classification took approximately one minute to complete.

The last column in Table 8.2 confirms whether or not the DASH type classification matched the ATEL type classification. Before analysing this, it is important to note that DASH is able to accurately state both the subtype and the age of the astronomical object. Superfit classifications, however, have only stated the broad type and whether or not the age is positive or negative.

Out of the 23 objects from the ATEL, DASH agrees with 20 of the classifications with very high certainties. All of the correctly classified templates have softmax probabilities above 84%, which is very high given that there are 306 different classification bins. The ATEL has placed question marks on the classifications that are not certain. Two of the classifications that differ from DASH have a question mark, and thus the ATEL classification cannot be considered reliable. The three incorrect cases are discussed below:

DES16X3es: Superfit has classified this as a Ia but has marked that it is uncertain on this classification. DASH, however, states that the object has a 92% chance of being a IIP with an age of 22 to 26 days. In fact, after visually inspecting this spectrum, I think that the ATEL classification is incorrect.

DES16X3jj: Superfit has again marked that it is not confident on classifying this spectrum as a type II supernova. DASH places a very low probability on the spectrum being a Ic-pec. However, given the low probability, the classification should not be considered reliable.

DES16E1ah: Superfit has stated that this is a type-II supernova, however DASH places a 75% probability that the spectrum is a Ia-91T. It should be noted that the second option by DASH was a type-II.

Overall, given the question marks by Superfit, DASH has correctly determined at least 21 out of the 23 spectra from the ATEL. However, for the classifications that DASH and Superfit agreed, DASH was able to provide a much stronger confidence on the classification than Superfit. It

did this by not simply stating the broad type and post or pre-max age, but by accurately stating the subtype as well as specific age bracket. This exemplifies that the DASH classifications are significantly more detailed. Furthermore, the classification by DASH are orders of magnitude faster than Superfit. All 23 classification took less than 60 seconds using DASH, but consumed several hours of computation on Superfit.

Name	Redshift	ATEL Classification	DASH		Match?
			Classification	Probability	
DES16E1de	0.292	Ia? (+2)	Ia-pec (+2 to +10)	91%	✓
DES16E2dd	0.0746	Ia (+3)	Ia-norm (+2 to +6)	89%	✓
DES16X3km	0.0542	II (+)	IIP (+6 to +10)	99.7%	✓
DES16X3er	0.167	Ia (+2)	Ia-91T (-2 to +6)	86%	✓
DES16X3hj	0.308	Ia (0)	Ia-norm (-2 to +2)	90%	✓
DES16X3es	0.554	Ia? (0)	IIP (+22 to +26)	92%	x
DES16X3jj	0.238	II? (+)	Ic-pec (-2 to 2)	37%	x
DES16C3fv	0.322	Ia (-6)	Ia-norm (-10 to +2)	99.8%	✓
DES16C3bq	0.241	Ia (+0)	Ia-norm (-2 to +6)	99.6%	✓
DES16E1md	0.178	Ia (0)	Ia-norm (-6 to +2)	99%	✓
DES16E1ah	0.149	II (+)	Ia-91T (+14 to +22)	75%	x
DES16C3ea	0.217	Ia (+)	Ia-norm (+10 to +26)	88%	✓
DES16X1ey	0.076	II (+)	IIb (+2 to +6)	38%	✓
DES16C3bq	0.237	Ia (+)	Ia-norm (-2 to +6)	97%	✓
DES16E2aoh	0.403	Ia (+)	Ia-norm (-2 to +6)	88%	✓
DES16X3aqqd	0.033	IIP (+)	IIb (-6 to +2)	99%	✓
DES16X3biz	0.24	Ia (-)	Ia-norm (-14 to +2)	98%	✓
DES16C2aiy	0.182	Ia (+)	Ia-norm (-2 to +6)	99.99%	✓
DES16C2ma	0.24	Ia (+)	Ia-norm (+14 to +22)	99.2%	✓
DES16X1ge	0.25	Ia (+)	Ia-norm (+14 to +22)	99.7%	✓
DES16X2auj	0.144	Ia (0)	Ia-norm (-6 to +6)	84%	✓
DES16E2bkg	0.478	Ia (0)	Ia-norm (-2 to +6)	99%	✓
DES16E2bht	0.392	Ia (+3)	Ia-norm (-6 to +2)	58%	✓

TABLE 8.2: Classification of supernovae released in the most recent ATELS by OzDES [40, 18, 30, 29]. The first column is the name of the observed object, the second column is the redshift determined by MARZ. The third column is the classification given in the ATEL by OzDES. It details the type followed by the age in brackets. A (+) indicates that it is after the maximum brightness, (-) indicates that it is pre-max, while (0) represents that it is at maximum. A question mark indicates that the ATEL is not certain on the classification. Most of these measurements were taken by Superfit or SNID. The fourth column is the classification from DASH. The fifth column states the softmax regression probability provided by DASH. The sixth column has a tick if the ATEL and DASH agree on the type of the supernova, and a cross if they disagree.

Chapter 9

Project Evaluation

9.1 Completion

The original project proposal for this thesis aimed to build upon previous software to design a program capable of automating the classification process, while improving upon the speed of previous work. I made the decision to try a completely new approach to solving this problem that was very different to previous tools. The decision to use deep learning was a big risk, for the reason that it had never been used for this type of spectral classification before, and therefore had the potential to not be effective at all. I believe that this risk has definitely paid off, and has enabled me to create a tool that is in many ways significantly better than previous tools.

Overall, I have produced a very novel supernova classification tool, that has met or exceeded the main objectives of the project. The tool is relatively user-friendly offering two different interfaces. The GUI is not overly complicated, and allows for easy operation and classification of spectra from a range of different file types. One of the programs main advantages over current tools is its speed, which is orders of magnitude better than Superfit, and faster than SNID. This is especially true when the user tries to classify several spectra at once. The second interface offered is a python library that enables quick classifications without human inspection. This advantage enables DASH to be a nearly completely autonomous program.

The matching accuracy is very good at determining the correct type of the supernova, reaching 100% reliability on the validation set used. The degeneracies in the age and subtypes mean that a classification on the age and subtype are approximately 87%. However, given that OzDES was most interested in being able to determine the subtype, DASH has definitely met this requirement. DASH works extremely well on signals with a high supernova signal-to-noise, and is able to operate at least just as accurately as Superfit and SNID.

9.2 Comparison to Current Methods

Overall I think DASH is superior to the currently used classification tools for four important reasons:

1. Speed
2. More specific classification
3. Accuracy
4. Installation and ease of use

9.2.1 Speed

The main improvement of DASH over current tools is the significant speed increase. The primary reason for the increase in speed is that machine learning does not iteratively compare with templates, and instead classifies based on features in the spectrum. Thus, unlike SNID and Superfit which increase their computation time linearly with the number of templates, DASH is able to separate the training and testing stages. The classification of a single supernova takes only a few seconds in DASH, but can take several tens of minutes in Superfit.

Moreover, while SNID is already a fast program, DASH is even faster, and this is particularly true when classifying several spectra at once. By making use of the DASH library functions available on PyPI, a user is able to quickly iteratively classify several spectra. By making use

of Tensorflow's C++ back-end, all of the different spectra can be passed into its background session and quickly classified.

9.2.2 More specific classification

Inspection of Table 8.2 clearly highlights that DASH is able to provide a much more detailed and specific classification than Superfit. Superfit is only able to classify into the four broad types: Ia, Ib, Ic, II, and cannot specify the subtypes. Moreover, the age classification is also much less precise as it usually can only state whether the spectra is post or pre maximum. On the other hand, DASH is able to classify the subtype and the age bracket with a very high accuracy (or softmax regression probability).

9.2.3 Accuracy

Based on the test comparison in Table 8.2, DASH can be stated to be at least as accurate as SNID and Superfit. Nearly all of the spectral classifications were consistent with SNID and Superfit, and the three that weren't may in fact have been correct because Superfit identified that its classifications were not certain.

It is possible that DASH has the potential have a higher accuracy than Superfit and SNID, but this is difficult to test because the correct classification is not known for spectra that Superfit could not classify.

One of the other main reasons why DASH's accuracy has the potential to be superior is that it matches based on identified features of the learning algorithm instead of iteratively searching through templates. The advantage of this is that a classification is always made based on the entire set of templates within a particular classification bin, rather than a single spectrum. This reduces the impact of templates with incorrect classifications or unrepresentative spectra.

9.2.4 Installation and Ease of Use

SNID and Superfit already have relatively intuitive and user friendly interfaces, and DASH has matched this expectation. However, one of the main issues with Superfit is that it is very difficult to install. This is partially due to the fact that it is written in IDL which is not easily accessible to a personal computer due to its expensive License fee. On the other hand, DASH has been written in Python, which is the most common language used among astronomers, is freely available, and thus allows users to manipulate it as they wish. Moreover, DASH is very simple to install, being available on GitHub and PyPI. It only has two dependencies that are not included in the pip installation including Tensorflow and PyQt. Both of these are not complicated to install, and thus make the installation of DASH a relatively straight-forward process.

9.3 Use in Astronomy Community

This software was developed primarily for use by OzDES. At the time of writing, it is currently being tested by members of the OzDES community including the director, Chris Lidman. Given its impressive speed improvements, and the fact that it appears to match the accuracy of Superfit, it is expected that DASH will soon replace current tools once some improvements (listed in the next section) are made.

While the software was made for OzDES, DASH has been made versatile enough so that it can be used by anyone and in particular any astronomer who is interested in classification. Some members of the OzDES community from California and the UK have been interested in the tool, and a python library function that enables quick classification without human-inspection. The developed DASH python library allows for this, and may be adopted as a classification tool in the near future.

9.4 Disadvantages of DASH

Two disadvantages of DASH compared to some current tools is its inaccuracy with redshifting, and the fact that it cannot identify host galaxies. Both Superfit and SNID are able to redshift based on an input spectrum, and Superfit is somewhat able to identify and subtract the host galaxy from the spectra. Both of these things can already be done by MARZ and can be input as a prior into DASH. However, future work will aim to incorporate this into the program as discussed in sections 9.5.1 and 9.5.2.

9.5 Future Improvements

There are still several missing features that have been requested by some OzDES members and that would be very useful for a more advanced supernova classification tool.

9.5.1 Redshifting

While DASH has been shown to exceed previous software in the four listed criteria in section 9.2, one down-fall is that it is not able to provide an accurate estimate of the redshift of a spectrum. Instead, at this stage, it requires an input redshift by a user. However, given that several good redshifting tools already exist (such as MARZ), this is not detrimental to the use of DASH. Future work will aim to train a redshift agnostic model, that will classify an input spectrum independent of redshift and then identify redshift based on a cross-correlation with the classified spectrum (see section 7.4).

9.5.2 Identify Host Galaxy

The current classification tools cannot easily classify a supernova from a spectra that has a bright host. A future improvement of DASH should aim to train a new model that includes an extra dimension of classification bins. While there are currently 306 bins for the 17 subtypes

and 18 age bins, there could be an extra dimension for the 6 different galaxy types. This would lead to around 1500 classification bins. The current spectra would be added to varying levels of each of the 6 galaxy types in order to train each classification bin.

If the host galaxy and supernova can be classified together, then DASH will be able to subtract the galaxy from the overall spectrum to provide the user with a separate supernova and host galaxy. This has never been done before, but would be of enormous use to many members of the astronomy community. As such, future work should aim to test the feasibility of this model.

9.5.3 False Positives

DASH provides the user with a softmax regression probability for each of the classification bins. The problem with this, however, is that all of the classifications only state the probability relative to all of the other classification bins. If the input spectrum does not match any of the supernova types, it can lead to false-positive classifications. This is because the program does not give an absolute measurement of whether the fit is good or not, but only states where it is good relative to the other options. Cross-correlation methods on the other hand, can give an absolute measurement. As such, future work should cross-correlate the input with the best matches to determine whether the fit is above some absolute threshold, and can therefore be considered to be reliable.

9.5.4 Python Library Functions

Currently the python library only consists of a single function to provide the user with the best matching classification bins for a particular input spectrum. Future work should aim to extend this library to include plotting comparisons, statistics of the classifications and other functions in consultation with users of the software.

9.5.5 GUI Features

Some extra GUI features can also be added to improve the experience of a DASH user. Firstly, a pie chart or list that quickly sums the Softmax probabilities to identify the overall probability for each of the four broad types has been a requested feature.

Secondly, an option to quickly turn on and off the continuum of the spectra that are being graphed, instead of just subtracting it away will be a useful feature for users who wish to more closely visually inspect the plots.

Finally, the GUI should enable a user to input several spectra at once from a FITS file. Currently, the GUI only enables one file at a time, but future work should aim to enable several at once.

Chapter 10

Conclusion

This thesis report has detailed the development of DASH, a novel supernova spectral classification tool. The program has been made for the OzDES collaboration, and aimed to improve upon current methods. Some of the main advancements of DASH have been its speed, accuracy in the specific details of the supernova, and its ease of use.

A review of the prior literature revealed that most classification tools have made use of template matching algorithms with cross-correlation or chi-squared approaches. While I began my thesis following this procedure, I soon realised that a novel approach was the only way that my software could make significant advancements from current work. I have made use of a Deep Learning algorithm which has enabled DASH to be orders of magnitude faster than the two main previous tools: Superfit and SNID. The main drawback of these programs is that they are either too slow and accurate, or fast but inaccurate. DASH has been able to be both fast and accurate due to its very different approach. Instead of matching based on an iterative comparison with templates, machine learning allows the software to train on the features that make each supernova type. DASH has been tested on the latest OzDES data, and has proven to be significantly faster and at least just as accurate as Superfit and SNID.

Thousands of supernova templates from the Berkeley SN Ia program and SNID have been used to train the model. As more data becomes available, it is expected that DASH can be trained to be even more reliable without sacrificing speed.

Overall, this thesis has created two new interfaces for the astronomy community. First, a user-friendly graphical interface has been developed to allow for visual inspection and analysis of each classification. Second, a python library for quick classification of several spectra has also been made available with easy installation with pip or from GitHub. Ultimately, the speed, accuracy, user-friendliness and versatility of DASH presents an advancement to existing spectral classification tools. As such, DASH is a viable alternative for the astronomy community.

Bibliography

- [1] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2015. URL: <https://www.tensorflow.org/> (visited on 11/04/2016).
- [2] Australian Astronomical Observatory. *The AAOmega Spectrograph*. URL: <https://www.aao.gov.au/science/instruments/current/AAOmega> (visited on 04/07/2016).
- [3] Australian Astronomical Observatory. *The Anglo-Australian Telescope*. URL: <https://www.aao.gov.au/about-us/anglo-australian-telescope> (visited on 04/07/2016).
- [4] I K Baldry et al. “Galaxy And Mass Assembly (GAMA): AUTOZ spectral redshift measurements, confidence and errors”. In: *Mon. Not. R. Astron. Soc* 000 (2014), pp. 1–13. arXiv: arXiv:1404.2626v1.
- [5] Nicholas M. Ball and Robert J. Brunner. “Data Mining and Machine Learning in Astronomy”. In: (2009). DOI: 10.1142/S0218271810017160. arXiv: 0906.2173. URL: <http://arxiv.org/abs/0906.2173><http://dx.doi.org/10.1142/S0218271810017160>.
- [6] S. Blondin et al. “The Spectroscopic Diversity of Type Ia Supernovae”. In: *The Astronomical Journal* 143.5 (2012). DOI: 10.1088/0004-6256/143/5/126. arXiv: 1203.4832. URL: <http://arxiv.org/abs/1203.4832><http://arxiv.org/abs/1203.4832>.
- [7] Stéphane Blondin and John L Tonry. “DETERMINING THE TYPE, REDSHIFT, AND AGE OF A SUPERNOVA SPECTRUM”. In: *The Astrophysical Journal* 666.666 (2007). arXiv: arXiv:0709.4488v1.
- [8] Mike Bolte. *Signal-to-Noise in Optical Astronomy*. 2006. URL: http://www.ucolick.org/~bolte/AY257/s{_}n.pdf.
- [9] Micol Bolzonella, Joan-Marc Miralles, and Roser Pelló. “Photometric Redshifts based on standard SED fitting procedures”. In: (2000). arXiv: 0003380v2 [arXiv:astro-ph].

- [10] Shlomo Dado and Arnon Dar. "ANALYTICAL EXPRESSIONS FOR LIGHT CURVES OF ORDINARY AND SUPERLUMINOUS TYPE Ia SUPERNOVAE". In: *The Astrophysical Journal* 809.1 (2015), p. 32. ISSN: 1538-4357. DOI: 10.1088/0004-637X/809/1/32. URL: <http://stacks.iop.org/0004-637X/809/i=1/a=32?key=crossref.bb89c8ec5b40714e0947788b5d381878>.
- [11] Christopher D'Andrea and OzDES. "OzDES: 100 Nights of AAT Spectroscopy on DES Sources". In: *American Astronomical Society, AAS Meeting #223, id.254.15 223* (2014).
- [12] T. M. Davis et al. "Scrutinizing Exotic Cosmological Models Using ESSENCE Supernova Data Combined with Other Cosmological Probes". In: *The Astrophysical Journal* 666.2 (2007), pp. 716–725. ISSN: 0004-637X. DOI: 10.1086/519988. URL: <http://stacks.iop.org/0004-637X/666/i=2/a=716>.
- [13] Bc Pavel Hála. "Master's thesis Spectral classification using convolutional neural networks". In: (2014). arXiv: arXiv:1412.8341v1.
- [14] G. E. Hinton and R. R. Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks". In: *Science* 313.5786 (2006).
- [15] S R Hinton et al. "Marz: Manual and Automatic Redshifting Software". In: *Astronomy and Computing* 15 (2016). DOI: 10.1016/j.ascom.2016.03.001.
- [16] D A Howell et al. "GEMINI SPECTROSCOPY OF SUPERNOVAE FROM THE SUPERNOVA LEGACY SURVEY: IMPROVING HIGH-REDSHIFT SUPERNOVA SELECTION AND CLASSIFICATION". In: *The Astrophysical Journal* 634.2 (2005). DOI: 10.1086/497119.
- [17] Daniel D. Kelson. "Optimal Techniques in Two dimensional Spectroscopy: Background Subtraction for the 21st Century". In: *Publications of the Astronomical Society of the Pacific* 115.808 (2003), pp. 688–699. ISSN: 0004-6280. DOI: 10.1086/375502. URL: <http://iopscience.iop.org/article/10.1086/375502>.
- [18] A. King et al. "Classification of 8 DES Supernova with OzDES". In: *The Astronomer's Telegram* 9570 (2016). URL: <http://adsabs.harvard.edu/abs/2016ATel.9570...1K>.

- [19] Kyler Kuehn et al. "OzDES and the Dark Energy Survey". In: *AAO Observer* (2014), pp. 4–7. URL: <https://www.aao.gov.au/files/201402---Issue-125---February-2014.pdf>.
- [20] Chris Lidman et al. "OzDES reaches the half way mark". In: *AAO Observer* 128 (2016), pp. 4–8. URL: <https://www.aao.gov.au/files/201603---Issue-129---March-2016.pdf>.
- [21] Yu-Qian Liu et al. "ANALYZING THE LARGEST SPECTROSCOPIC DATASET OF STRIPPED SUPERNOVAE TO IMPROVE THEIR IDENTIFICATIONS AND CONSTRAIN THEIR PROGENITORS". In: ().
- [22] Yuqian Liu and Maryam Modjaz. "SUPERNOVA IDENTIFICATION SPECTRAL TEMPLATES OF 70 STRIPPED-ENVELOPE CORE-COLLAPSE SUPERNOVAE". In: (2015). arXiv: arXiv:1405.1437v3.
- [23] Michelle Lochner et al. "Photometric Supernova Classification With Machine Learning". In: (2016). DOI: 10.3847/0067-0049/225/2/31. arXiv: 1603.00882. URL: <http://arxiv.org/abs/1603.00882><http://dx.doi.org/10.3847/0067-0049/225/2/31>.
- [24] M Modjaz et al. "OPTICAL SPECTRA OF 73 STRIPPED-ENVELOPE CORE-COLLAPSE SUPERNOVAE". In: *The Astronomical Journal* 147.17pp (2014). DOI: 10.1088/0004-6256/147/5/99.
- [25] Maryam Modjaz et al. "THE SPECTRAL SN-GRB CONNECTION: SYSTEMATIC SPECTRAL COMPARISONS BETWEEN TYPE IC SUPERNOVAE, AND BROAD-LINED TYPE IC SUPERNOVAE WITH AND WITHOUT GAMMA-RAY BURSTS". In: ().
- [26] A. Möller et al. "Photometric classification of type Ia supernovae in the SuperNova Legacy Survey with supervised learning". In: (2016). arXiv: 1608.05423. URL: <http://arxiv.org/abs/1608.05423>.
- [27] Ivelina Momcheva and Erik Tollerud. "SOFTWARE USE IN ASTRONOMY: AN INFORMAL SURVEY". In: (2015).

- [28] Daniel Muthukrishna and David Parkinson. "A cosmographic analysis of the transition to acceleration using SN-Ia and BAO". In: (2016). arXiv: 1607.01884. URL: <http://arxiv.org/abs/1607.01884>.
- [29] C. R. O'Neill et al. "Classification of 2 DES supernova with OzDES". In: *The Astronomer's Telegram* 9636 (2016). URL: <http://adsabs.harvard.edu/abs/2016ATel.9636...10>.
- [30] C. R. O'Neill et al. "Correction to ATel #9636 - Missing Table". In: *The Astronomer's Telegram* 9637 (2016). URL: <http://adsabs.harvard.edu/abs/2016ATel.9637...10>.
- [31] S. Perlmutter et al. "Measurements of Omega and Lambda from 42 High-Redshift Supernovae". In: *The Astrophysical Journal* 517.2 (1998), pp. 565–586. ISSN: 0004-637X. DOI: 10.1086/307221. arXiv: 9812133 [astro-ph]. URL: <http://arxiv.org/abs/astro-ph/9812133><http://dx.doi.org/10.1086/307221>.
- [32] Philip A Pinto and Ronald G Eastman. "THE TYPE IA SUPERNOVA WIDTH-LUMINOSITY RELATION". In: (2000). arXiv: 0006171v1 [arXiv:astro-ph].
- [33] Richard Pogge. *IV Interstellar Dust*. 2011. URL: <http://www.astronomy.ohio-state.edu/~pogge/Ast871/Notes/Dust.pdf>.
- [34] Adam G. Riess et al. "Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant". In: *The Astronomical Journal* 116.3 (1998), pp. 1009–1038. ISSN: 00046256. DOI: 10.1086/300499. URL: <http://stacks.iop.org/1538-3881/116/i=3/a=1009>.
- [35] M Sasdelli et al. "Exploring the spectroscopic diversity of type Ia supernovae with DRACULA: a machine learning approach for the COIN collaboration". In: *MNRAS* 000 (2015), pp. 1–16.
- [36] Will Saunders, Russell Cannon, and Will Sutherland. "Improvements to runz". In: *Anglo-Australian Observatory Epping Newsletter* (2004), p. 16.
- [37] Brian P. Schmidt et al. "The High Z Supernova Search: Measuring Cosmic Deceleration and Global Curvature of the Universe Using Type Ia Supernovae". In: *The Astrophysical*

- Journal* 507.1 (1998), pp. 46–63. ISSN: 0004-637X. DOI: 10.1086/306308. arXiv: 9805200 [astro-ph]. URL: <http://arxiv.org/abs/astro-ph/9805200>.
- [38] Jeffrey M Silverman et al. “Berkeley Supernova Ia Program I: Observations, Data Reduction, and Spectroscopic Sample of 582 Low-Redshift Type Ia Supernovae”. In: *Mon. Not. R. Astron. Soc* 000 (2012), pp. 1–34. arXiv: arXiv:1202.2128v3.
- [39] M. Smith et al. “DES14X3taz: A TYPE I SUPERLUMINOUS SUPERNOVA SHOWING A LUMINOUS, RAPIDLY COOLING INITIAL PRE-PEAK BUMP”. In: *The Astrophysical Journal* 818.1 (2016), p. L8. ISSN: 2041-8213. DOI: 10.3847/2041-8205/818/1/L8. URL: <http://stacks.iop.org/2041-8205/818/i=1/a=L8?key=crossref.2dc01b3b45b4b5a45e56f2d9b7faa908>.
- [40] N. Sommer et al. “Classification of 13 DES supernova with OzDES”. In: *The Astronomer’s Telegram* 9504 (2016). URL: <http://adsabs.harvard.edu/abs/2016ATel.9504....1S>.
- [41] Stanford University. *Softmax Regression - Ufldl*. URL: http://ufldl.stanford.edu/wiki/index.php/Softmax{_}Regression (visited on 07/30/2016).
- [42] Swinburne University of Technology. *Supernova Classification*. URL: <http://astronomy.swin.edu.au/cosmos/S/supernova+classification> (visited on 04/07/2016).
- [43] John Tonry and Marc Davis. “A survey of galaxy redshifts. I - Data reduction techniques”. In: *Astronomical Journal* 84 (1979), pp. 1511–1525. DOI: 10.1086/112569.
- [44] Ofer Yaron and Avishay Gal-Yam. “WiSeREP - An Interactive Supernova Data Repository”. In: *Publications of the Astronomical Society of Pacific, Volume 124, Issue 917, pp. 668-681 (2012)*. 124 (2012), pp. 668–681. ISSN: 0004-6280. DOI: 10.1086/666656. arXiv: 1204.1891. URL: <http://arxiv.org/abs/1204.1891><http://dx.doi.org/10.1086/666656>.
- [45] Fang Yuan et al. “OzDES multifibre spectroscopy for the Dark Energy Survey: first-year operation and results”. In: *Monthly Notices of the Royal Astronomical Society* 452.3 (2015), pp. 3047–3063. ISSN: 0035-8711. DOI: 10.1093/mnras/stv1507. URL: <http://mnras.oxfordjournals.org/lookup/doi/10.1093/mnras/stv1507>.