Deep Learning Techniques in Signal Processing

Radamanthys Stivaktakis

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science and Engineering

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Advisor: Professor *Panagiotis Tsakalides*

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.

This work has been supported by the Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science (ICS) and partly funded through the DEDALE project, within the H2020 Framework Program of the European Commission under contract number 665044.

University of Crete
Computer Science Department

Deep Learning Techniques in Signal Processing

Thesis submitted by
Radamanthys Stivaktakis
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: ______________________
Radamanthys Stivaktakis

Committee approvals: ______________________
Panagiotis Tsakalides
Professor, Thesis Supervisor

____
Maria Papadopouli
Professor, Committee Member

____
George Tzagkarakis
Principal Researcher, Committee Member

Departmental approval: ______________________
Antonios Argyros
Professor, Director of Graduate Studies

Heraklion, February 2019

Deep Learning Techniques in Signal Processing

Abstract

Deep learning architectures have revolutionized research in numerous scientific domains and triggered a paradigm shift from traditional machine learning methodologies and feature engineering to architecture design and the so-called "end-to-end" training. While the efficacy of deep learning networks can be strongly attributed to their vigorous capacity of extracting aggregated knowledge, as the size of the available data increases, at the same time they exhibit an underwhelming performance, when trained with a limited amount of annotated examples. Our main aim, in this thesis, is to explore the impact of the utilization of state-of-the-art deep learning methodologies, in both cases of data abundance and data deficiency, in two major research topics in the fields of cosmology and remote sensing.

In the first case study, we address the problem of spectroscopic redshift estimation in astronomy. Due to the expansion of the Universe and its statistical homogeneity and isotropy, galaxies recede from each other on average. This movement causes the emitted electromagnetic waves to shift from the blue part of the spectrum to the red part, due to the Doppler effect. This redshift is one of the most important observables in astronomy and cosmology, allowing the measurement of galaxy distances. Several sources of astrophysical and instrumental noise render the estimation process far from trivial, especially in the low signal-to-noise regime of many astrophysical observations. In recent years, new approaches for a reliable and automated methodology of the redshift evaluation have been sought out, in order to minimize our reliance on currently popular techniques that heavily involve human intervention. The fulfillment of this task has evolved into a grave necessity, in conjunction with the insatiable generation of immense amounts of astronomical data, falling into the category of the so-called Big Data. We propose an alternative approach that transforms the issue at hand from a regression problem to a multi-class classification task, opening the field for the deployment of a currently dominating deep learning classifier, commonly known as Deep Convolutional Neural Networks. This approach is extensively evaluated on a spectroscopic dataset of full spectral energy galaxy distributions, modelled after the upcoming Euclid satellite galaxy survey. Experimental analysis on observations of idealistic and realistic conditions demonstrate the potent capabilities of the proposed scheme.

In the second case study, we examine a flourishing research topic in the field of remote sensing, namely land cover classification. Conventional methodologies mainly focus either on the simplified single-label scenario or on pixel-based approaches that cannot efficiently handle high resolution images. On the other hand, the problem of multi-label land cover scene categorization remains, to this day, fairly unexplored. While deep learning and Convolutional Neural Networks have demonstrated an astounding capacity at handling challenging image classification tasks, they significantly underperform when trained on limited in size datasets. To overcome this issue, we propose an online data augmentation technique that can

drastically increase the size of a smaller dataset to copious amounts. Our experiments on a multi-label variation of the UC Merced Land Use dataset demonstrates the potential of the proposed methodology, which outperforms the current state-of-the-art by more than 6% in terms of the F-score metric.

Τεχνικές Βαθιάς Μάθησης στην Επεξεργασία Σημάτων

Περίληψη

Οι αρχιτεκτονικές βαθιάς μάθησης έχουν επιφέρει θεμελιώδεις αλλαγές στην έρευνα πολυάριθμων επιστημονικών πεδίων και έχουν πυροδοτήσει τη μεταστροφή από τις πατροπαράδοτες μεθοδολογίες μηχανικής μάθησης και την κατασκευή γνωρισμάτων, στη σχεδίαση αρχιτεκτονικών και στην επονομαζόμενη εκπαίδευση “από άκρη σε άκρη”. Ενώ η αποτελεσματικότητα των δικτύων βαθιάς μάθησης μπορεί να αποδοθεί στη σθεναρή ικανότητά τους να εξάγουν συναθροισμένη γνώση, καθώς το πλήθος των διαθέσιμων δεδομένων αυξάνεται, συγχρόνως αποδίδουν απογοητευτικά, όταν εκπαιδεύονται με τη χρήση περιορισμένων σε πλήθος σημειωμένων δεδομένων. Η βασική επιδίωξη αυτής της εργασίας είναι η εξερεύνηση του αντίκτυπου σύγχρονων μεθοδολογιών βαθιάς μάθησης, σε περιπτώσεις έλλειψης ή αφθονίας δεδομένων, σε δύο σημαντικά ερευνητικά θέματα που άπτονται των πεδίων της κοσμολογίας και της τηλεπισκόπησης.

Στην πρώτη περιπτωσιολογική μελέτη θέτουμε επί τάπητος το πρόβλημα εκτίμησης της φασματοσκοπικής ερυθρής μετατόπισης στην αστρονομία. Εξαιτίας της διαστολής του Σύμπαντος και της στατιστικής του ομοιογένειας και ισοτροπίας, οι γαλαξίες απομακρύνονται, κατά μέσο όρο, μεταξύ τους. Αυτή η κίνηση αναγκάζει τα εκπεμπόμενα ηλεκτρομαγνητικά κύματα να μετατοπιστούν από το μπλε τμήμα του φάσματος στο κόκκινο, σύμφωνα με το φαινόμενο Doppler. Η ερυθρή μετατόπιση είναι από τα πιο σημαντικά φαινόμενα στην αστρονομία και την κοσμολογία, καθιστώντας δυνατή τη μέτρηση των αποστάσεων των γαλαξιών. Αρκετές πηγές θορύβου, είτε αστροφυσικής προέλευσης είτε από τα όργανα μετρήσεων, καθιστούν τη διαδικασία εκτίμησης μη-τετριμμένη, ειδικά σε περιπτώσεις αστροφυσικών παρατηρήσεων υψηλού θορύβου. Τα τελευταία χρόνια, έχουν αναζητηθεί νέες προσεγγίσεις για την αξιόπιστη και αυτοματοποιημένη εκτίμηση της ερυθρής μετατόπισης, με σκοπό να ελαχιστοποιηθεί η εξάρτησή μας από τις υπάρχουσες δημοφιλείς τεχνικές που βασίζονται ισχυρά στην ανθρώπινη παρέμβαση. Η εκπλήρωση αυτής της αναζήτησης αποτελεί σοβαρή αναγκαιότητα, σε συνδυασμό με την ακόρεστη παραγωγή απέραντων, σε πλήθος, αστρονομικών δεδομένων που υπάγονται στην κατηγορία των Μεγάλων Δεδομένων. Προτείνουμε μία εναλλακτική προσέγγιση, που μετασχηματίζει το ζητούμενο πρόβλημα από ένα ζήτημα παλινδρόμησης σε ένα πρόβλημα ταξινόμησης πολλαπλών κλάσεων. Έτσι, ανοίγει ο δρόμος για την αξιοποίηση ενός υπερισχύοντος ταξινομητή βαθιάς μάθησης, κοινώς γνωστός ως Βαθιά Νευρωνικά Δίκτυα Συνέλιξης. Αποτιμούμε εκτενώς την προσέγγιση αυτή, με τη χρήση φασματοσκοπικών δεδομένων που αποτελούνται από γαλαξιακές κατανομές φασματικής ενέργειας, μοντελοποιημένα σύμφωνα με την επερχόμενη επισκόπηση του δορυφόρου Ευκλείδη. Η πειραματική ανάλυση σε ιδεαλιστικές και ρεαλιστικές παρατηρήσεις επιδεικνύει τις ισχυρές δυνατότητες του προτεινόμενου σχεδίου.

Στη δεύτερη μελέτη περίπτωσης εξετάζουμε ένα ακμάζον ερευνητικό θέμα στο πεδίο της τηλεπισκόπησης, και συγκεκριμένα την ταξινόμηση κάλυψης γης. Οι συμβατικές μεθοδολογίες που έχουν χρησιμοποιηθεί στο παρελθόν εστιάζουν είτε στην απλοποιημένη περίπτωση του προβλήματος μοναδικών ετικετών, είτε σε προσεγγίσεις βασισμένες σε εικονοστοιχεία, οι οποίες δεν μπορούν να διαχειριστούν αποδοτικά εικόνες υψηλής ανάλυσης. Αντίθετα, το πρόβλημα της ταξινόμησης κάλυψης γης με τη χρήση πολλαπλών ετικετών, παραμένει μέχρι και σήμερα σχετικά ανεξερεύνητο. Παρά του ότι οι μέθοδοι βαθιάς μάθησης και τα Νευρωνικά Δίκτυα Συνέλιξης έχουν επιδείξει μία εκπληκτική ικανότητα στην αντιμετώπιση απαιτητικών προβλημάτων ταξινόμησης εικόνων, όμως δεν καταφέρνουν να ανταποκριθούν στις προσδοκίες σε περιπτώσεις που η εκπαίδευσή τους πραγματοποιείται με τη χρήση περιορισμένων δεδομένων στο πλήθος. Για να υπερνικήσουμε το συγκεκριμένο ζήτημα, προτείνουμε μία δυναμική τεχνική επέκτασης των δεδομένων που είναι ικανή να αυξήσει το μέγεθος μικρών συνόλων δεδομένων σε άφθονες ποσότητες. Τα πειράματά μας σε μία παραλλαγή των δεδομένων UC Merced Land Use, με πολλαπλές ετικέτες, επιδεικνύουν τις δυνατότητες του προτεινόμενου πλαισίου, που ξεπερνάει την προϋπάρχουσα αποτελεσματικότερη μέθοδο, κατά ένα ποσοστό της τάξης του 6% στη μετρική του F-score.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω με τον πιο ειλικρινή τρόπο τον επόπτη μου, καθηγητή και αντιπρύτανη κύριο Παναγιώτη Τσακαλίδη, που από την πρώτη στιγμή πίστεψε σε εμένα και μου εμπιστεύτηκε μία θέση στην ερευνητική του ομάδα. Η εκτίμηση που τρέφω για την ερευνητική και ακαδημαϊκή του υπόσταση είναι πολύ ισχυρή και αποτελεί πηγή έμπνευσης και οδηγό για τη συνεχή μου προσπάθεια προσωπικής εξέλιξης.

Επίσης, θα ήθελα να εκφράσω δημόσια ένα μεγάλο ευχαριστώ προς τον μεταδιδακτορικό ερευνητή και άμεσο επιβλέποντά μου, δόκτωρ Γρηγόριο Τσαγκατάκη, για τη συνεχή υποστήριξη και την ανεκτίμητη βοήθεια που μου προσέφερε τα τρία τελευταία χρόνια, τόσο στα πλαίσια της πτυχιακής όσο και της μεταπτυχιακής μου εργασίας.

Δεν μπορώ να μην αναφερθώ στα δύο συγκινονούντα δοχεία, το Πανεπιστήμιο Κρήτης και το Ινστιτούτο Τεχνολογίας και Έρευνας, για όλους τους πόρους (εκπαιδευτικούς, ερευνητικούς και οικονομικούς) που μου παρείχαν τα τελευταία 9 χρόνια, χωρίς τη συμβολή των οποίων δε θα είχα καταφέρει όσα έχω επιτύχει μέχρι σήμερα. Αποτελούν και τα δύο πρότυπο ακαδημαϊκής και ερευνητικής ευημερίας και ήταν μεγάλη μου χαρά και τιμή που αποτέλεσα κομμάτι τους όλα αυτά τα χρόνια.

Ακόμη, θα ήθελα να αναφερθώ στους ανθρώπους, που αν και η επαφή τους με την έως τώρα ακαδημαϊκή και ερευνητική μου πορεία δεν ήταν ακριβώς άμεση, την καθόρισαν, παρ'ολα αυτά, με τον πιο καίριο και ουσιαστικό τρόπο. Θέλω να πω ένα απέραντο ευχαριστώ στη θεία μου Ασπασία, για όλα όσα έχει κάνει για εμένα, και στους γονείς μου Αντώνη και Μαρία, για την ανιδιοτελή και άνευ όρων βοήθεια, υποστήριξη και αγάπη τους. Τέλος, καμία λέξη δεν είναι αρκετή για να μπορέσω να εκφράσω τη βαθιά ευγνωμοσύνη και αγάπη μου προς τη σύζυγό μου, Ελένη. Η καλοσύνη της, το θάρρος και η στήριξή της αποτέλεσαν φάρο που φώτισε τα δύσκολα ώστε να φαίνονται απλά. Χωρίς εσένα δεν θα είχα καταφέρει τίποτα.

στο Φως της ζωής μου, Ελένη

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The rise of the "golden age" of Deep Learning [1] has fundamentally changed the way we handle and apprehend raw, unprocessed data. While existing machine learning models heavily rely on the development of efficient feature extractors, a non-trivial and very challenging task, deep learning architectures are able to single-handedly derive complex representations concealed in the data by learning intermediate representations and by structuring different levels of abstraction, essentially modelling the way the human brain works.

The origins of the deep learning paradigm (also known as Artificial Neural Networks - ANNs) lie in the midst of the 20$^{\text{th}}$ century. Specifically, Ivakhnenko and Lapa [2] described for the first time, in 1965, the first working learning algorithm for supervised deep feed-forward multi-layer perceptrons. For the coming decades, the concepts surrounding deep learning methodologies evolved and expanded, but deep learning itself waxed and waned in popularity mostly because of its insatiable need for immoderate amounts of computational and memory resources. However, the arrival of the current decade (2010s) marked the end of an era of controversy and dispute regarding the reliability and viability of ANN architectures and established a new order for deep learning, an idea that was once sort of an "ugly duckling" which blossomed to become "the belle of the ball" [3]. The rapid technological advancements in the computer processing power and memory-storage means and the drastic increase in the amount of available data had a profound contribution in the performance of various deep learning architectures, which were established as the state-of-the-art in many research fields.

One such an architecture subsists in Convolutional Neural Networks (CNNs) [4], namely a sequentially structured model that utilizes a combination of convolutional, non-linear, pooling and fully-connected layers. The inspiration behind CNNs resides in the concept of visual receptive fields [5], i.e. the region in the visual sensory periphery where stimuli can modify the response of a neuron. This is the main reason that CNNs initially found application in image classification, by learning to recognize images by experience in the same perception that a human being can gradually learn to distinguish different image stimuli from one another.

In this work, we build upon the state-of-the-art methodology of Convolutional Neural Networks and we focus on its application on two challenging research topics in the fields of cosmology and remote sensing. Specifically:

- We consider the problem of spectroscopic redshift estimation on realistic and idealistic galaxy spectral profiles.
- We address the problem of multi-label land cover scene categorization on the modified, from a multi-label perspective, UC Merced Land Use Dataset [6].

1.1 Big Data as the primary fuel of deep learning methodologies

The emergence of the concept of Big Data [7] has signaled a major shift from the narrow availability of exploitable data to data overflow and under-utilization, essentially cultivating the need for new approaches on data processing and management. The acquisition of data in vast amounts and from various sources has opened new horizons in scientific research and in the way we apprehend and utilize existing or newly developed methodologies. The aformentioned monumental success of deep learning architectures in the recent years, can be strongly attributed to their interminable capacity to harness the power of Big Data and has been significantly enhanced by fully exploiting emerging, cutting-edge hardware technologies.

On the other hand, the dependency of these type of architectures on attainable unlabeled or human-annotated observations constitutes, at the same time, their major drawback leading to severe under-performance issues in cases where the availability of said data is limited. The more parameters we want the model to learn or as complex as the issue at hand gets, so does the data required for training increase. The risk of these networks overfitting to small training datasets is highly apparent considering that most of their variations train on parameters that often exceed the order of magnitude of a million, or even tens of millions. Even though their potential ability to extract high-level, complex abstractions and data representations is unparalleled compared to other methodologies, nevertheless this ability can greatly deteriorate with datasets of inadequate size.

1.1.1 Alternative techniques for data deficiency

Dealing with deficient datasets can prove to be a challenging issue, having in mind that at first we need to identify the scarcity itself as the source of the network's under-performance and, at the same time, to effectively address the problem with potential solutions. While cross-validation can significantly help in dealing with the bias-variance dilemma (i.e. the trade-off between underfitting and overfitting), a series of simple, yet powerful techniques can be also employed to address the problem of overfitting itself. This set of techniques, commonly known as *regularizers*, can be broadly divided into two characteristic categories, either a model-centric or a data-centric. In the former case, we interfere with the network's structure and functionality with the goal to reduce its complexity and enhance its generalization capabilities. Commonly used techniques associated with this case, include weight decay, early stopping, dropout [8] and transfer learning [9]. In the latter case, we intervene on the data themselves, either by artificially augmenting their size (data augmentation) [10] or by consistently normalizing their intermediate representations in each layer (batch normalization) [11].

In the scope of this thesis, we emphasize on three of the aformentioned techniques, thoroughly described in Chapter 3. Specifically, we examine the cases of dropout, data augmentation and batch normalization.

1.2 Challenges in remote sensing

The broader scientific area of remote sensing refers to the use of satellite- or aircraft-based technologies, for the acquisition of information about an object or phenomenon without any physical contact. There are several scientific fields pertinent to the remote sensing archetype and the challenges that it poses. In this thesis, we consider the problems of spectroscopic redshift estimation in cosmology and the multi-label scene categorization in land cover classification.

1.2.1 Spectroscopic redshift estimation on galaxy spectral profiles

Modern cosmological and astrophysical research seeks answers to questions such as "what is the distribution of dark matter and dark energy in the Universe?" [12, 13], or "how can we quantify transient phenomena, like exoplanets orbiting distant stars?" [14]. To answer such questions, a large number of deep space observation platforms has been deployed. Spaceborne instruments, such as the Planck Satellite[1] [15], the Kepler Space Observatory[2] [16] and the upcoming Euclid mission[3] [17], seek to address these questions with unprecedented accuracy, since they avoid the deleterious effects of Earth's atmosphere, a strong limiting factor to all their observational strategies. Meanwhile, ground-based telescopes like the LSST[4] [18] will be able to acquire massive amounts of data through high frequency full-sky surveys, providing complementary observations. The analysis of huge numbers of observations from a variety of different sources has paved the way for new methodologies in various research fields, and astronomy is an indicative scenario where observations propel the data-driven scientific research [19].

One particular long-standing problem in astrophysics is the ability to derive precise estimates to galaxy redshifts. According to the Big Bang model, due to the expansion of the Universe and its statistical homogeneity and isotropy, galaxies move away from each other and any given observation point. A result of this motion

[1] http://www.esa.int/Our_Activities/Space_Science/Planck

[2] http://kepler.nasa.gov/

[3] http://sci.esa.int/euclid/

[4] https://www.lsst.org

is that light emitted from galaxies is shifted towards larger wavelengths through the Doppler effect, a process termed *redshifting*. Redshift estimation has been an integral part of observational cosmology, since it is the principal way in which we can measure galaxies' radial distances and hence their 3-dimensional position in the Universe. This information is fundamental for several observational probes in cosmology, such as the rate of expansion of the Universe and the gravitational lensing of light by the matter distribution - which is used to infer the total dark matter density - among other methods [20, 21].

A variety of photometric redshift estimation techniques have been widely used due to the fact that photometric analysis is substantially less costly and time consuming contrary to the spectroscopic case. However, the limited wavelength resolution of photometry, compared to spectroscopy, introduces a higher level of uncertainty to the given procedures. In spectroscopy, the dispersion of the light into a wider spectral band can lead to better discrimination between different wavelengths making the whole process far more accurate. By observing the full spectral energy distribution (SED) of a galaxy, one can theoretically detect distinctive emission and absorption lines that can lead to a fairly trivial redshift estimation by measuring the wavelength shift of these spectral characteristics from the rest-frame (frame of reference where the galaxy is at rest, i.e. at zero redshift).

In practice, estimation of redshift from spectroscopic observations is far from straightforward. There are several sources of astrophysical and instrumental errors, such as readout noise from CCDs, contaminating light from dust enveloping our own galaxy, Poisson noise from photon counts, and more. Furthermore, due to the need of obtaining large amounts of spectra, astronomers are forced to limit the time of integration for any given galaxy, resulting in low signal-to-noise measurements. As a consequence, not only it becomes difficult to confidently measure specific spectral features for secure redshift estimation, but we also incur the risk of misidentifying features - e.g. confusing a hydrogen line for an oxygen line - which results in so-called catastrophic outliers. Human evaluation mitigates a lot of these problems with current, relatively small, data sets. However, the spectral observations we will consider are particularly challenging, existing in very low signal-to-noise regimes and detected in massive amounts, forcing us to develop automated methods capable of achieving high accuracy and necessitating minimal human intervention. Chapter 4 proposes our novel approach to the problem of spectroscopic redshift estimation on both clean, idealistic spectral profiles and on noisy, realistic observations.

1.2.2 Multi-label, land cover scene categorization on top-view image data

High-resolution imaging sensors aboard miniaturized satellite and aerial vehicles acquire large amounts of high-resolution imagery, which mandates the development of automated and sophisticated algorithms for reliably processing and deriving meaningful information from the image content. This can become more apparent in time-sensitive situations [22], [23] or in cases where the frequency of arrival of incoming data can be at a daily or even hourly basis.

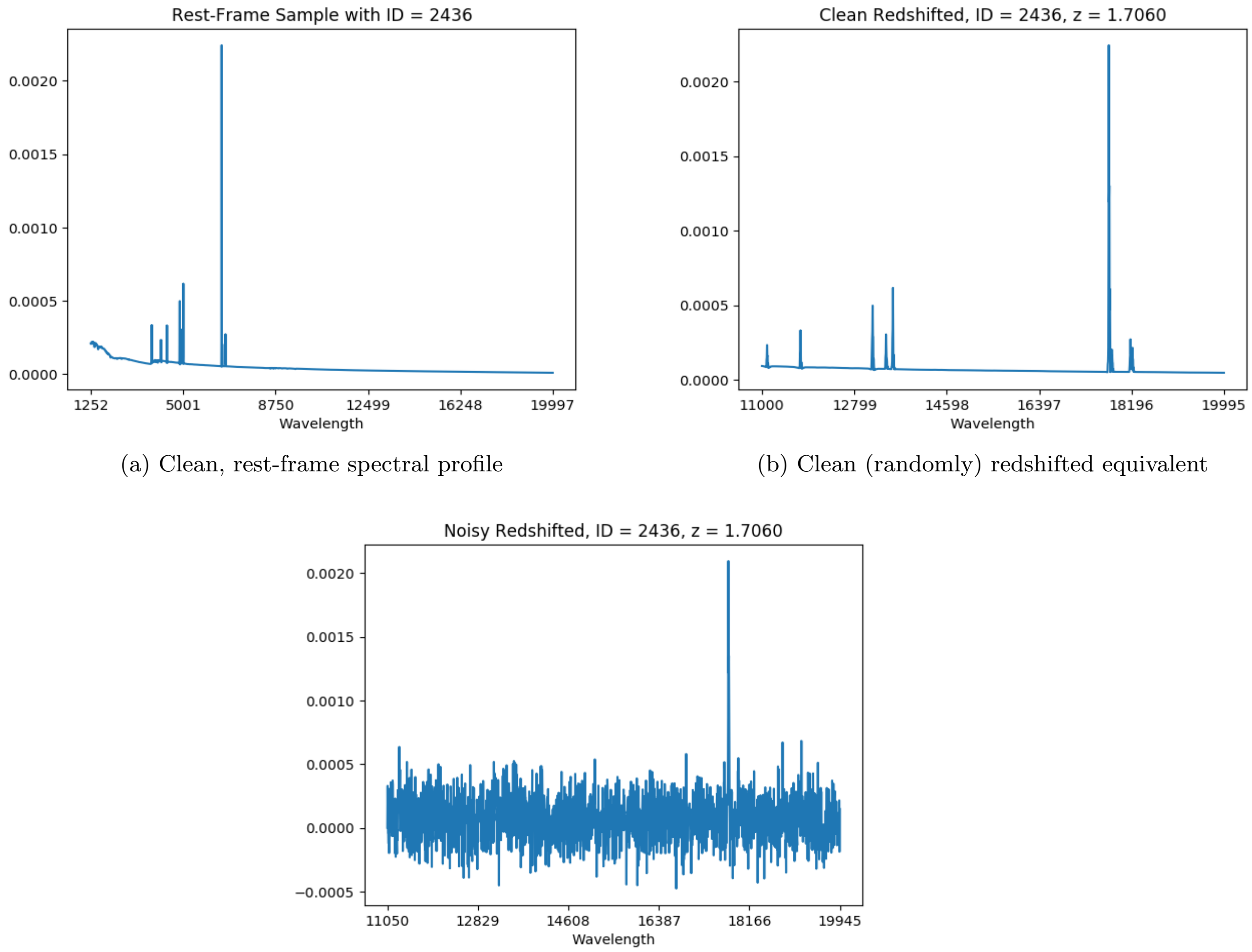
Land cover classification remains one of the biggest challenges in the remote sensing discipline and a crucial component in monitoring physical and anthropogenic phenomena in the large scale. Semantic segmentation of satellite images has been widely applied, in a pixel-wise manner, but as denoted in [24] there are certain limitations to that approach when dealing with higher-resolution images. Meanwhile, in higher level feature-based approaches each image is processed as a whole, with a subsequent goal to be associated with a descriptive label of the scene content. Most existing works concentrate on the multi-class scenario, where every image is categorized to one-of-many different labels, an assumption that oversimplifies the issue at hand given that a certain scene can depict more than one of the primary classes. This is generally known as a multi-label classification approach and is a very underdeveloped research topic, in the case of land cover scene categorization.

In Chapter 5, we approach the problem of multi-label land cover scene categorization using a conventional deep learning-based methodology, uniquely modified for multi-label classification. Given that most existing multi-label land cover datasets are very small, we employ a novel technique that artificially increases the size of any given image dataset in copious amounts, commonly known as *data augmentation* [10].

1.3 Motivation and objectives

The aformentioned challenges motivated us to use a state-of-the-art deep learning model as a baseline, namely Convolutional Neural Networks, attuned and customized to their specific requirements. Both the examined research problems remain fairly unexplored and lack of generalized, automated and robust methodologies, that can address them in a reliable way.

In the case of spectroscopic redshift estimation, the estimation of galaxy redshifts is perceived as a regression task in general, still, a classification approach can be formulated without the loss of essential information. The robustness of the proposed model will be examined in two different data variations, as depicted in the example of Figure 1.1. In the first case (b), we deploy randomly redshifted variations of the original rest-frame spectral profiles (a) of the dataset used. These redshifted equivalents, effectively result from linear translations of the rest-frame, in logarithmic scale. This case can be considered as an idealistic scenario, as it



(a) Clean, rest-frame spectral profile

(b) Clean (randomly) redshifted equivalent

(c) Noisy redshifted equivalent

Figure 1.1: Example of the utilized spectral profiles. From the initially available rest-frame samples, randomly redshifted equivalents are produced, in clean and noisy forms. The x-axis corresponds to the emitted wavelengths, while the y-axis resembles the normalized spectral density flux value.

ignores the interference of noise or presumes the existence of a reliable denoising technique. On the other hand, a more realistic scenario is considered (c), with the available redshifted observations subjected to noise of realistic conditions.

In the case of multi-label land cover scene categorization, the lack of adequately sized datasets can seriously cripple the performance of the adopted deep learning model, necessitating the need for alternative solutions. To that end, we exploit the CNNs' transformation invariance property, based on the fact that a CNN must be able to robustly classify inputs (in our case images), regardless of small, possible alterations of their content. The employed technique, termed data augmentation earlier in this thesis, has proven to be significantly effective in image classification. Data augmentation proposes a simple, yet powerful, framework where the size of a small labeled dataset, derived in a limited set of conditions, can be artificially increased through a series of potential transformations (translations, flips, rescaling etc.). As previous studies in the single-label case have shown [22, 25, 26], CNN classification with data augmentation can have a substantial impact in multiple remote sensing scenarios, demonstrating its potent capabilities.

1.4 Contribution

The main contributions of our work are referenced below:

- We use a deep learning architecture for the case of spectroscopic redshift estimation, never used before for the issue at hand. To achieve that we need to convert the problem from a regression task, as engaged in general, to a classification task, as encountered in this novel approach.

- We utilize Big Data and evaluate the impact of a significant increase of the employed observations in the overall performance of the proposed methodology. The dataset used is modelled after one of the biggest upcoming spectroscopic surveys, the Euclid Mission [17].

- We employ the cutting edge methodology of Convolutional Neural Networks with dynamic data augmentation, tailored for multi-label land cover scene classification. The proposed method marks a clear departure for existing techniques, such as the current state-of-the-art graph-theoretic semi-supervised approach [27], or a recent work [28] that exploits different types of features, either hand-crafted or derived via transfer learning, to calculate image distances and to obtain corresponding similarities. Our method, apart from being the first to employ a fully trainable, end-to-end deep learning model for the task at hand, it manages, at the same time, to significantly outperform the state-of-the-art on a redefined version, from a multi-label perspective, of the UC Merced Land Use Dataset [6].

1.5 Related publications

The experimental efforts of this thesis have been summarized in the following two original publications:

1. R. Stivaktakis, G. Tsagkatakis, B. Moraes, F. Abdalla, J-L Starck, P. Tsakalides, “Convolutional Neural Networks for Spectroscopic Redshift Estimation on Euclid Data”, IEEE Transactions on Big Data (Special Issue on Big Data From Space) - under review

2. R. Stivaktakis, G. Tsagkatakis, P. Tsakalides, “Deep Learning for Multi-Label Land Cover Scene Categorization Using Data Augmentation”, IEEE Geoscience and Remote Sensing Letters - accepted

1.6 Roadmap

The remainder of this thesis is structured as follows. In Chapter 2, we present a brief outline of the relevant literature concerning the utilized methodologies and the accompanying challenges, in astronomy and land cover classification. A detailed overview of the existing theoretical background, models and techniques adopted in this work is provided in Chapter 3. In Chapters 4 and 5, we focus on the two examined case studies, regarding the spectroscopic redshift estimation on galaxy spectral profiles modelled after the Euclid survey and the multi-label land cover scene categorization on the modified UC-Merced Dataset. Specifically, we present the datasets used, analyze the proposed frameworks, demonstrate our experimental efforts and evaluate our findings in comparison with other methodologies. Finally, in Chapter 6 we conclude, presenting possible directions for future work.

Chapter 2

Related work

Most existing deep learning models have largely benefited from the dawn of the Big Data era, demonstrating impressive results that can match, or even exceed, human performance [29]. Despite the fact that training a deep artificial neural network can be fairly computationally demanding, even more so while we increase its complexity and the data it needs to process, nevertheless, the continuous evolution of computational means and memory storage capacity have rendered feasible such a task. At the same time, and in contrast to the training process, the evaluation phase for a test dataset can be exceptionally fast, with a negligible execution time, regardless of its size. Currently, deep learning is considered to be the state-of-the-art in various research domains, such as image classification, natural language processing and robotic control, with models like Convolutional Neural Networks [4], Long-Short Term Memory (LSTM) networks [30], and Recurrent Neural Networks [31], dominating the research trends.

The main idea behind Convolutional Neural Networks materialized for the first time with the concept of "Neocognitron", a hierarchical neural network capable of performing visual pattern recognition [32], and evolved into LeNet-5, by Yann LeCun et al. [4], in the following years. The massive breakthrough of CNNs (and deep learning in general) transpired in 2012, in the ImageNet competition [33], where the CNN of Alex Krizhevsky et al. [34] managed to reduce the classification error record by ~10%, an astounding improvement at the time. CNNs have been considered in numerous applications, including image classification [34, 35] and processing [36], video analytics [37, 38], spectral imaging [39] and remote sensing [40, 41, 42] confirming their dominance and ubiquity in contemporary scientific research.

In recent years, the practice of CNNs in astrophysical data analysis has led to new breakthroughs, among others, in the study of galaxy morphological measurements and structural profiling through their surface's brightness [43, 44], the classification of radio galaxies [45], astrophysical transients [46] and star-galaxy seperation [47], and the statistical analysis of matter distribution for the detection of massive galaxy clusters, known as strong gravitational lenses [48, 49]. The exponential increase of incoming data, for future and ongoing surveys, has led to a compelling need for the deployment of automated methods for large-scale galaxy decomposition and feature extraction, negating the commitment on human visual inspection and hand-made user-defined parameter setup.

The problem of estimating galaxy redshifts has been examined in greater depths in the case of photometry, contrary to the spectroscopic equivalent. Concerning the former, popular methods used, adapted for this kind of problem, include a Bayesian estimation with predefined spectral templates [50] and a variety of machine learning-based models, such as the Multilayer Perceptron [51, 52] and Boosted Decision Trees [52, 53]. On the other hand, due to heavily noisy observations in spectroscopy, the main redshift estimation techniques involve cross-correlating the SED with predefined spectral templates [54] or PCA decompositions of a template library. Noisy conditions and potential errors, due to the choice of templates, are the main reasons that most reliable spectroscopic redshift estimation methods heavily depend on human judgment and experience to validate automated results.

Regarding land cover classification, a variety of pixel-based approaches have been proposed [55, 56, 57, 58, 59, 60, 61, 62], but in cases where datasets of high resolutions are adopted, feature-based approaches that designate labels to the image content in its entirety, are generally preferred [24, 25, 42, 63, 64]. The extraction of meaningful and descriptive features from remote sensing imagery has been a critical step in the design of automated and sophisticated machine learning algorithms. In the domains of computer vision and image processing, the development of highly effective methodologies like Scale Invariant Feature Transform (SIFT) [65] and Histogram of Oriented Gradients (HOG) [66] incorporate a reliable approach to various tasks, suffering nonetheless from an over-reliance on many heuristic optimizations and, in general, on human intervention. At the same time, remote sensing tailored features like NDVI (Normalized Difference Vegetation Index) [67] are too closely coupled with particular types of observations. On the other hand, deep learning architectures have proven to be potent feature extractors [34, 38, 68, 69, 70], in a more generalized way, by learning hierarchical intermediate representations and structuring complex and deep levels of abstraction.

Finally, in the case of multi-label scene categorization, previous works in the natural image classification literature [71, 72, 73] expose the principal challenges met and propose potential solutions. In the special case of the remote sensing, content-based image retrieval (CBIR), for multi-label, satellite or aerial image scene categorization, which we examine in this thesis, the only relevant works that exist in the literature are the previously mentioned works in [27] and [28], which, however, are not based on deep learning.

Chapter 3

Theoretical background

In this chapter, we provide a detailed overview of the theory, methods and techniques used throughout this thesis. First, we present the fundamentals of the supervised classification in machine learning and then we focus on the theory behind Convolutional Neural Networks and the specifics regarding their structure, functionality and regularization. A basic knowledge of conventional Artificial Neural Network (ANN) methodologies is assumed for the reader.

3.1 Classification fundamentals

The problem of classification (Figure 3.1) is one of the most significant topics in the field of machine learning. Given a collection of objects, whose members each belong to one of a number of different sets or classes, a classification or prediction rule is the process where for each observation in the collection, a prediction is made concerning the real class it belongs to. The prediction rule is usually derived by an

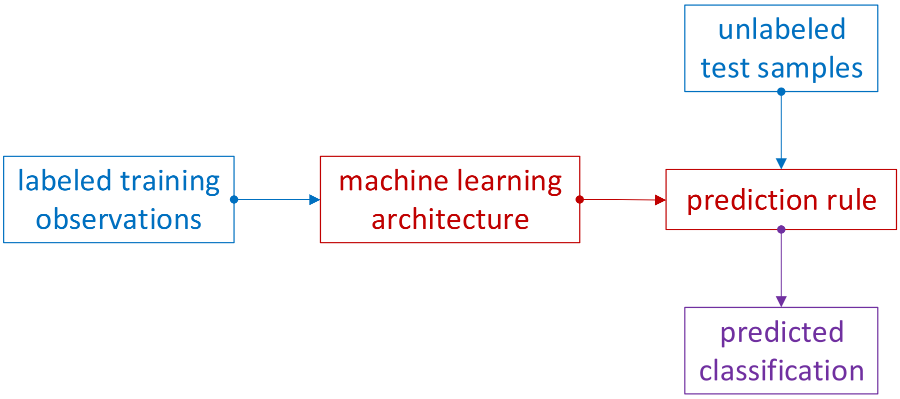


Figure 3.1: Block-diagram of the classification problem, in its abstract form.

automated machine learning architecture, which is trained on labeled observations, following a “programming-by-example” paradigm. The procedure of using labeled examples for the training of the machine learning model is commonly known as supervised learning. The prediction of new, unlabeled observations by the trained model constitutes the testing phase. In this phase, the evaluation of the model’s performance takes place.

In a classification scenario, the various classes that characterize the objects of the studied collection can be of different statistical data types, based on the formulated problem. They can either be binary (positive or negative, true or false, success or failure), numerical (integer-valued or real-valued) or categorical (on the basis of some qualitative property). Categorical classes can be furtherly divided into ordinal and nominal classes. In the first case, there is a natural ordering between the different categories, however the distance between consecutive classes is considered unknown and not equally distributed. On the other hand, in the latter case, there exists no form of ranking between the classes.

The three most common types of a classification problem are the binary, the multi-class and the multi-label classification. In the following sub-sections, we will examine the basic principles of each type, along with their main differences.

3.1.1 Binary classification

Binary categorization is generally considered as the easiest form of classification and is defined as the problem, where the output attribute of each observation (i.e., its label) can be categorized as one of two outcomes. Typical examples of binary classification tasks include pass-or-fail situations (e.g. the final exams of a course),

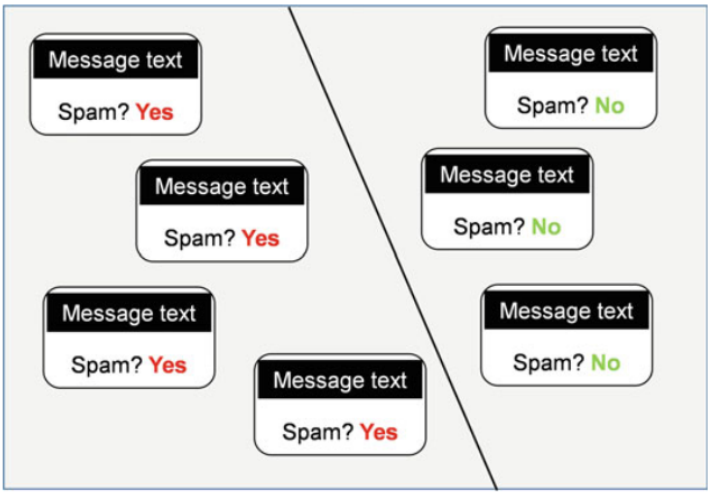


Figure 3.2: Example of a binary classification problem with a linear decision boundary. The trained classifier can either categorize a message as spam or as not spam (a relevant message).

medical evaluations (e.g. a patient suffering from a certain disease or not), spam filtering (Figure 3.2, [74]) and so on so forth. The main objective of a binary classifier is to find a robust decision boundary that will reliably dichotomize the feature space into two groups, one for the data points that belong to the positive class and another for the points of the negative class. The simplest form of a decision surface is linear, in cases where the observed data are linearly separable. In all other cases, the derived boundary can either be linear or more complex in form.

One of the main challenges met in the case of binary classification is class imbalance, meaning that the proportion of data examples that correspond to one of the two binary classes is considerably larger compared to the alternative. For example, in the case of spam filtering, the number of messages attributed as spam is generally smaller compared to the number of relevant messages. At the same time, misclassifying spam messages as normal (false negative) is not of the same significance as falsely appointing that a normal message, coming from a legitimate source, is actually spam (false positive). Certain solutions that can successfully address the problem of class imbalance include data re-sampling, generation of synthetic data, utilization of class weighted models and gradient boosting [75, 76].

3.1.2 Multi-class classification

A generalized version of the binary categorization problem can be found in the case of multi-class classification, in the context that in both scenarios each data sample is associated with a single output attribute, though in the latter case, that attribute is not restricted to binary values. In a multi-class classification problem, the label

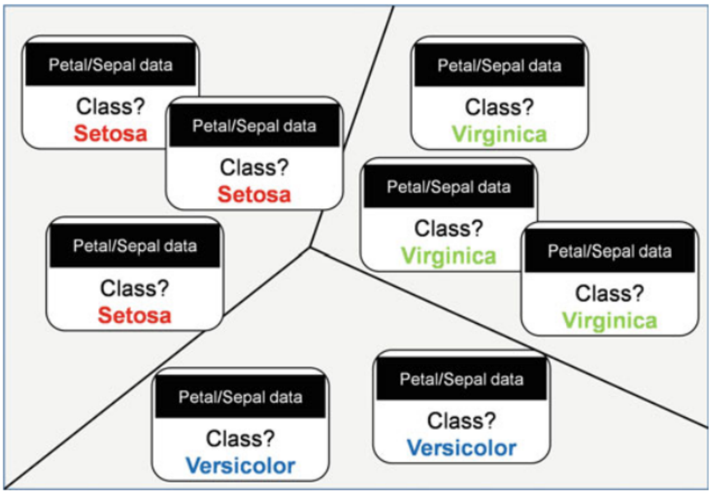


Figure 3.3: Example of a multi-class classification problem on Fisher's Iris dataset. The number of distinct classes is 3 (Setosa, Versicolor, Virginica). Each data sample can be categorized in either of the 3 classes, yet exclusively in one.

of each observation can hold any value from a finite set of predefined values, based on the utilized application, with all values, however, being mutually exclusive. Typical examples of multi-class classification tasks include face recognition [77, 78], handwriting classification [4, 10], Fisher's linear discriminant analysis on the Iris flower dataset [79] (Figure 3.3, [74]) and so on so forth.

Since many classification methodologies have been developed specifically for the binary classification scenario, multi-class classification often requires the use of alternative techniques, that will divide the issue at hand into easier, binary sub-problems. To that end, the concepts of one-vs-all and one-vs-one reduction can be employed. Let us consider a multi-class classification scenario with C different classes. In the one-vs-all strategy, and for each class, a new classifier is trained that regards all the samples of that class as positive examples, and all the remaining observations that belong to the rest of the classes as negative examples. In this case, we need to train as many classifiers as the number of classes, hence C classifiers. On the other hand, in the one-vs-one reduction we adopt a slightly different strategy. For all possible pairs of classes, we retain all the observations that belong to the two classes and at the same time we discard all the remaining samples. Then, for each pair, we train a new classifier based on the preserved samples. In this case, the number of different classifiers that we need to train is $\frac{C(C-1)}{2}$, hence, as many as all the different combinations of pairs. The main drawback of both presented reduction techniques lies in the fact that both one-vs-all and one-vs-one methodologies can't easily handle problems where the number of unique classes is very large, due to the inevitable increase in complexity. The utilization of these strategies is considered redundant in the case of Convolutional Neural Networks, given that CNNs adopt algorithm adaption techniques which can easily extend their functionality, from a binary to a multi-class classifier.

The investigated case study in Chapter 4, concerning the problem of spectroscopic redshift estimation, pertains to the multi-class classification problem.

3.1.3 Multi-label classification

The case of multi-label categorization is the most difficult among the examined classification problems, as it combines properties from both binary and multi-class classification tasks. Formally, multi-label classification is the problem of training an automated model that is able to map input data samples to binary output vectors, assigning a value of 0 or 1 for each element in the output vector. The elements of the output vector correspond to the distinct labels of the formulated problem, with no mutual exclusivity between the labels. This means, that each data observation can be directly associated with more than one labels, without any restrictions on their number. Typical examples of multi-label categorization applications include sentiment analysis [80, 81], recommender systems [82], automatic media tagging [83], categorization of natural image scenery (Figure 3.4, [74]), land cover classification [27] etc.

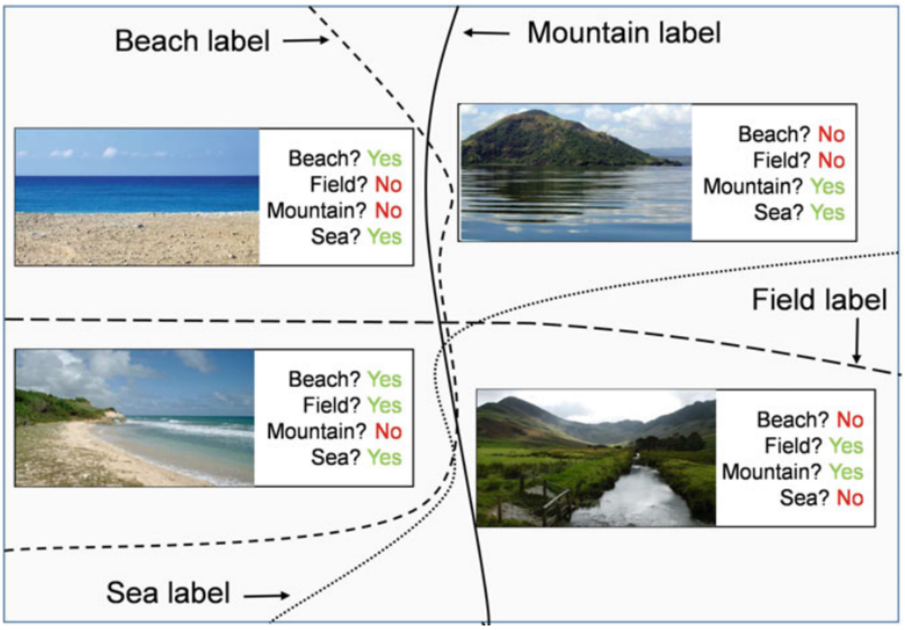


Figure 3.4: Example of a multi-label classification problem on natural image scenery. The problem consists of a set of non-mutually exclusive labels (beach, field, mountain, sea), meaning that each image can be associated with more than one of the labels of the utilized labelset.

The fact that in multi-label classification each sample can be associated with more than one classes, prohibits the utilization of a variety of popular machine learning methodologies which have been developed with the problem of multi-class classification as a reference point. Various approaches have been considered, that convert the problem of multi-label classification into simpler multi-class or binary tasks, having as a major drawback that label independence is assumed, thus not preserving the cross-correlations between labels of the same sample. A characteristic example of these approaches is sample unfolding [84], where each observation is duplicated in as many instances as the number of active labels it is associated with, assigning a single label to each duplicate. Hence, the multi-label classification problem is reduced to a mutli-class problem, with a considerably increased dataset size, which grows in magnitude as the label cardinality rises. Another example, lies in the calculation of the power set of the existing labelset, associating each data observation exclusively with one entry in the newly formed power set, based on its active labels. The major drawback of this technique manifests in the fact that the size of the produced power set, increases exponentially with each increase of the size of the original labelset (if C is the initial number of distinct labels, then the size of the power set will be 2^C). Once more, the utilization of these techniques in the case of Convolutional Neural Networks can be omitted, given that CNNs can be adjusted to adopt algorithm adaption techniques and, therefore, be extended from a multi-class to a multi-label classifier.

The investigated case study in Chapter 5, regarding the problem of land cover scene categorization, concerns the multi-label classification scenario.

3.2 Convolutional Neural Networks

A Convolutional Neural Network is a particular type of Artificial Neural Network, which comprises of neuronal inputs, outputs and intermediate representations, along with their respective connections that encode the learnable weights of the network. One of the key differences between CNNs and other neural architectures, like Multilayer Perceptron [85], is that in typical ANNs, each neuron of any given layer connects with all neurons of its respective previous and following layers (fully-connected layers). On the contrary, CNNs are structured in a locally-connected manner, exhibiting the spatial correlations of the given input, under the assumption that neighboring regions of each observation are more likely to be related than regions that are farther away. By reducing the number of total connections, CNNs can successfully manage to drastically decrease the number of trainable parameters, rendering the network less prone to overfitting. At the same time, CNNs can be administered in the use of various types of data, with more or less complicated dimensional structures, with the pivotal property of maintaining their spatial correlations without the need to collapse higher dimensional matrices into flattened vectors. Based on this property, and in association with the structural dimensions of the input data, a CNN can either be one-dimensional, two-dimensional, three-dimensional or, as a matter of fact, n-dimensional.

In sub-sections 3.2.1 and 3.2.2, we present the basic components of a typical CNN, which demonstrate its structural and functional properties. The first part of a CNN architecture consists of a combination of convolutional, non-linear and downsampling (pooling) layers and is commonly known as the feature extraction module. The main responsibility of this module is to derive informative and relevant characteristics from the input observations, starting from abstract representations in its shallower layers, and culminating to concrete and detailed features, as the depth of the network increases. The procured features of the feature extractor are adopted by the classification module of the CNN, which in turn is tasked to perform valid and reliable predictions for the input data. An example illustration of the above pipeline is provided in Figure 3.5 [86]. As a final step, a detailed overview concerning the utilized regularizing techniques is presented in sub-section 3.2.3.

3.2.1 Feature extraction module

3.2.1.1 Convolutional layers and non-linearities

The foundational layer of a CNN, the *Convolutional Layer*, encodes the spatial correlations of the given input, by identifying appropriate n-dimensional filters.

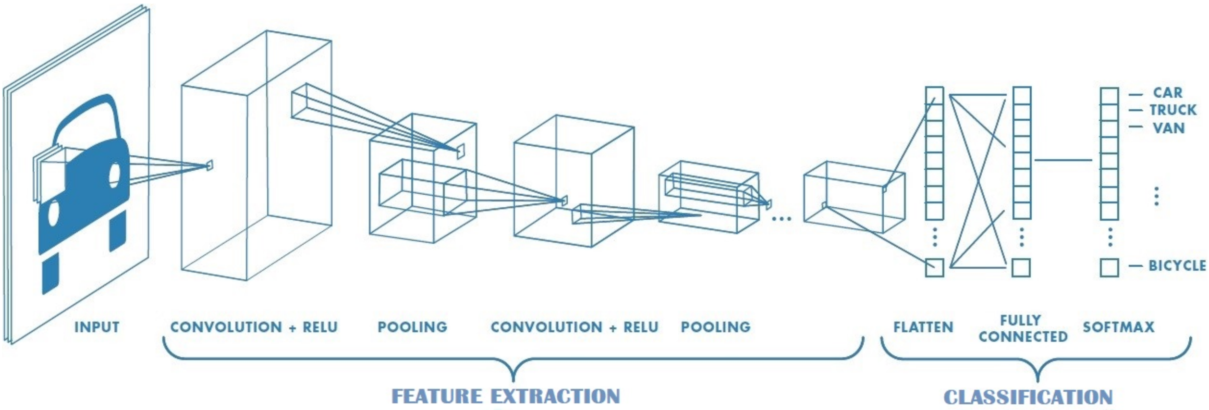
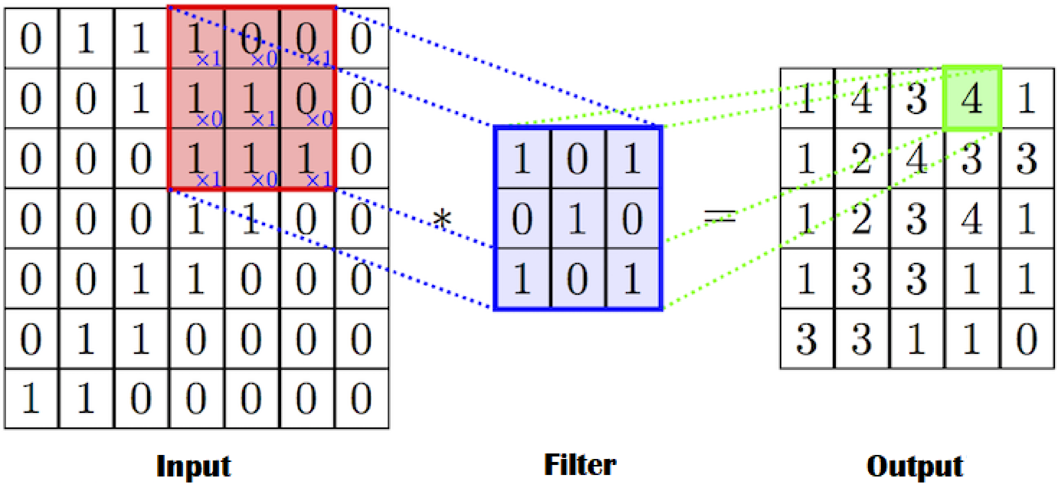


Figure 3.5: Example architecture of a simple, two-dimensional CNN. In the module of feature extraction, a convolutional + ReLU layer extracts primary features from the input image, with a subsequent downsampling via a pooling layer. The aformentioned pipeline can be repeated as many times as needed to procure higher-order features. In the classification module, the derived features of the final pooling layer are flattened into a one-dimensional vector and are given as input to a sequence of fully-connected layers. Finally, a probabilistic softmax layer is adopted, which is responsible for the output predictions.

These trainable filters essentially map local, possibly overlapping, regions of the preceding layer to units of the succeeding layer, resulting in local connectivity patterns. The filter incorporates the learnable parameters of the network, which at first are random [87] and, therefore, totally unreliable, but as the training of the network advances, through the process of backpropagation [68], these parameters are optimized and are able to capture interesting features from the given inputs. The parameters (i.e. weights) of the filter are considered to be shared [88], in the aspect that the same weights can be utilized throughout the convolution of the entirety of the input, with the alternative being, having different weights for each convolutional step. The assumption of weight sharing is based on the fact that for a particular filter we want it to be able to detect a certain kind of features, in all possible positions of the given input. This, can consequently lead to a drastical decrease in the number of weights, lowering the complexity of the network and enhancing its ability to generalize, thus adding to its total robustness against overfitting.

Using the two-dimensional case as a reference point, we can furtherly examine the convolution of an input observation of size $W \times H$ with a trainable filter of size $K \times L$. The resulting output will be of size $W' \times H'$, where $W' = W - K + 1$ and $H' = H - L + 1$. The values of W' and H' may vary based on the stride of the operation of convolution, meaning that with a minimum stride of 1 the filter is slid over the input vector one cell at a time, thus generating a longer output. On the other hand, with a bigger stride value, the filter "jumps" to more distant



0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	1	1	0	0	0	0
1	1	0	0	0	0	0

1	0	1
0	1	0
1	0	1

1	4	3	4	1
1	2	4	3	3
1	2	3	4	1
1	3	3	1	1
3	3	1	1	0

Figure 3.6: A simple example of a two-dimensional convolutional layer.

cells after each step of the convolution, resulting in a smaller output. An example of the convolution of an input observation with a simple filter, and a stride of 1, is illustrated in Figure 3.6. Even though the aformentioned example corresponds to the two-dimensional case, it can be easily generalized to the n-dimensional case.

For each particular convolutional layer, more than one filters can be trained, each associated with the acquisition of different features. Each of the n-dimensional filters, act independently on the input, generating an n-dimensional output structure per filter. The generated outputs are then stacked together over a new channel dimension, with a length equal to the number of different filters trained in the given convolutional layer. For a subsequent convolutional layer, that will need to operate on this higher, $(n + 1)$-dimensional structure, the properties of the convolutional procedure will not change, in the aspect that each filter of this layer will be convolved with each of the channels separately. For the current filter and for each channel, a new n-dimensional output will be produced and, ultimately, all these derived outputs will be summed up and combined into one final n-dimensional structure.

When addressing challenging problems, the use of shallow CNN architectures is insufficient, given their limited capacity to form deeper and complex representations of the input data. The development of deeper models, able to derive informative and detailed characteristics, becomes a necessity. The claim that an effective expansion of the CNN can be achieved by introducing more convolutional layers, one on top of another, is actually invalid. Given the linear property of the convolutional operation, the sequential stacking of all these convolutional layers could actually be accounted for as one merged linear transformation over the input data, thus rendering the formed architecture as shallow. To be able to effectively form deeper, more complex CNN models, a non-linearity needs to be introduced directly after each convolutional layer, enabling the network to act as a universal function approximator [89]. Typical choices for the non-linear function (also known as activation function) include the logistic (sigmoid) function, the hyperbolic tangent (tanh) and the Rectified Linear Unit (ReLU), represented by the following formulae:

$$f(x) = \frac{1}{1+e^{-x}}\,, \qquad \text{(logistic function - sigmoid)} \tag{3.1}$$

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}\,, \qquad \text{(hyperbolic tangent - tanh)} \tag{3.2}$$

$$f(x) = max(0,x)\,. \qquad \text{(rectifier - ReLU)} \tag{3.3}$$

In CNNs, the most common choice is ReLU and its variations [90]. Compared to the cases of the sigmoid and hyperbolic tangent functions, the rectifier possesses the advantage that it is easier to compute (as well as its gradient) and is generally more resistant to saturation conditions [34], rendering the training process faster and less likely to suffer from the problem of vanishing gradients [91].

3.2.1.2 Pooling layers

Pooling Layers are usually introduced between subsequent convolutional + ReLU layers. Though optional, the use of pooling can have a very impactful role in the successful performance of the model, by introducing desired properties like scale invariance through a form of non-linear downsampling. The main intuition behind pooling lies in the fact that the exact location and orientation of a detected feature is less significant than its relative position to other features. Therefore, the network can be rendered invariant to small changes of the initial input, that don't tamper with its original content. With pooling, the processed input is commonly split into small, evenly sized and non-overlapping local regions, and for each region a given operation is executed (e.g. maximum, average, L2-norm etc.). Thus, the most relevant information is preserved, leading at the same time to a substantial reduction of the data dimensonality and, consequently, to an increased robustness against overfitting. A simple example of a max-pooling operation is presented in Figure 3.7 [92].

3.2.2 Classification module

3.2.2.1 Fully-connected layers and the softmax classifier

The classification segment of a Convolutional Neural Network is responsible for the credible deduction of valid predictions for the input observations. While the feature extractor is in charge of the production of high-quality features, at the same time, the classifier is responsible of distilling meaningful knowledge by taking into account the entirety of those features. This can be effectively achieved by utilizing

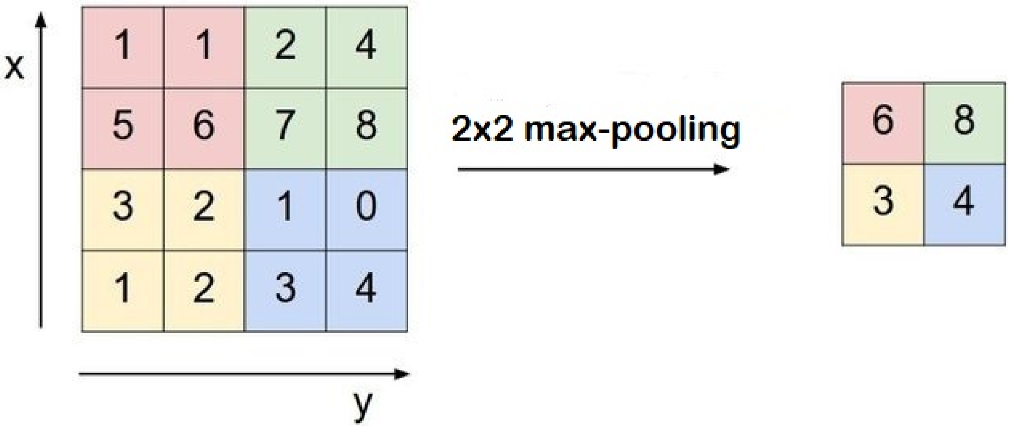


Figure 3.7: A simple example of a two-dimensional max-pooling layer.

Fully-Connected Layers, commonly known as dense layers, where, as the name implies, all the neuronal nodes of a given layer are directly connected with all nodes of its subsequent layer. Multiple dense layers (typically with a ReLU activation function) can be stacked together, to compose even deeper architectures.

The final classification step is performed via a dense layer, with as many output units as the number of unique classes of the addressed problem. In this final layer, a probabilistic activation function must be employed, typically in the form of the multi-class generalization of logistic regression, commonly known as *Softmax Regression*. Softmax regression is based on the exploitation of the probabilistic characteristics of the normalized exponential (softmax) function defined as:

$$h_\theta(x)_j = \frac{e^{\theta_j^T x}}{\sum_{k=1}^{C} e^{\theta_k^T x}}, \tag{3.4}$$

where x is the input of the fully-connected layer, θ_j are the parameters that correspond to a certain class c_j and C is the total number of the distinct classes related to the task at hand. It is fairly obvious that the softmax function reflects an estimation of the normalized probability of each class c_j to be predicted as the correct class. As deduced from the previous equation, each of these probabilities can take values in the range of [0,1] and, at the same time, they all need to add-up to the value of 1. This property, makes softmax an exceptional choice for the problem of multi-class classification, however in the multi-label case it displays a problematic behaviour, as justified in Chapter 5.

3.2.3 Regularizing techniques

The risk of overfitting the training dataset remains imminent in the case of Convolutional Neural Networks, considering their high complexity and their parameter-heavy functionality. As previously mentioned, the local-connectivity patterns formed in the convolutional layers and the concept of weight sharing can seriously reduce the total number of adopted parameters, however the utilization of these methodologies does not suffice, given the fact that the most parameter-intense segment of a CNN is its classification module with all the densely connected layers. The employment of effective techniques that can reliably address the problem of overfitting is a grievous matter, considering that the trained model can easily adapt to the particular dataset it was trained on, failing to effectively generalize on new, unseen data.

Apart from the trivial solution of gathering more data to experiment with, having a theoretical complete negation of the effects overfitting when the number of training observations tend to infinity, a variety of simple techniques can be effectively used to enhance the generalization capabilities of the trained network. In the scope of this thesis, we are going to demonstrate the effects of data augmentation, dropout and batch normalization.

3.2.3.1 Data augmentation

In most cases, the collection of more data can prove to be a complicated and expensive procedure. The difficulty of this task becomes even more challenging, when the data collected must also be labeled. A possible solution can be formulated by manipulating the adopted dataset to appear as if it was more diverse. To that end, we exploit the aformentioned CNNs' transformation invariance property, based on the fact that a CNN must be able to robustly classify inputs, regardless of small, possible alterations of their content. Specifically, we employ a technique that has proven to be significantly effective, especially in the case of image classification, known as data augmentation [28]. Data augmentation proposes a simple framework, where the size of a small labeled dataset, derived in a limited set of conditions, can be artificially increased through a series of potential transformations (e.g. translations). In Chapter 5, we will thoroughly examine the impact of the utilization of data augmentation, in a multi-label image classification and remote sensing scenario.

3.2.3.2 Dropout

Dropout [8] is one of the most popular techniques, used with CNNs, that can help narrow down the effects of overfitting. With dropout, a simple, yet very powerful trick can be used to temporarily decrease the total parameters of the network at each training iteration. All the neurons in the network are associated with a probability value p (subject to hyper-parameter tuning) and each neuron, independently from the others, can be temporarily dropped from the network (along with all incoming and outgoing connections) with that probability. This is an iterative process, meaning that for each training sample of a training batch, a random portion of the entirety of the original network is dropped, leading to "thinner" and more degenerated variations of its initial structure, as the value of the probability p grows bigger. Each layer can be associated with a different p value, meaning that dropout can be considered as a per-layer operation, with some layers discarding neurons in a higher percentage, while others dropping neurons in a lower rate or not at all. In the testing phase, the entirety of the network is used, meaning that dropout is not applied at all. Given that the weights of this final version of the network are scaled-down, compared to the originally trained weights, then for each neuron, all its associated outgoing weights are multiplied by the probability value $1 - p$, equal to the probability that the said neuron was retained in the network during the training phase.

3.2.3.3 Batch normalization

Contrary to dropout, batch normalization can be accounted for, more as a normalizer, but previous studies [11] have shown that it can work very effectively as a regularizer as well. Batch normalization is, in fact, a local (per layer) normalizer, that operates on the neuronal activations in a similar way to the initial normalizing technique, optionally applied to the input in the pre-processing step. The primary goal is to enforce a zero mean and a standard deviation of one, for all activations of the given layer and for each mini-batch. The main intuition behind batch normalization lies in the fact that, as the neural network deepens, it becomes more probable that the neuronal activations of intermediate layers might diverge significantly from desirable values and might tend to saturation. This is known as Internal Covariate Shift [11] and batch normalization can play a crucial role on mitigating its effects. Consequently, it can actuate the gradient descent operation to a faster convergence, but also it can lead to an overall highest accuracy and, as stated before, render the network stronger and more robust against overfitting.

Chapter 4

Case study: Convolutional Neural Networks for spectroscopic redshift estimation on Euclid data

In this chapter, we explore the problem of accurate redshift estimation from realistic and idealistic spectroscopic observations, modeled after the Euclid spectroscopic survey. Redshift estimation is generally considered as a regression task, given the fact that a galaxy redshift (z) can be measured as a non-negative, real-valued number. Considering the specifications of the Euclid space telescope, we can focus our study in its redshift range of detectable galaxies. Consequently, we can restrict the precision of each of our estimations to match the resolution of the spectroscopic instrument, meaning that we can split the chosen redshift range into evenly sized slots, equal to Euclid's required resolution. Hence, we can transform the problem at hand from a regression task to a multi-class classification task, using a set of ordinal classes, with each class corresponding to a different slot. Our main goal, in this Euclid case study, is the utilization, validation and evaluation of a cutting-edge classification model, namely Convolutional Neural Networks, tailored to the modified problem of spectroscopic redshift estimation. In the next sections, we first explore the case of the Euclid space telescope and we present the specifications of the adopted dataset. Next, we examine the proposed framework and the utilized methodologies. Finally, we demonstrate our experimental findings and we discuss on the inferred results.

4.1 The Euclid space telescope

The Euclid mission [17] aims to measure the global properties of the Universe to an unprecedented accuracy, with emphasis on a better understanding of the nature of Dark Energy. It will collect photometric data with broadband optical and near-infrared filters and spectroscopic data with a near-infrared slitless spectrograph. The latter will be one of the biggest upcoming spectroscopic surveys and will help us determine the details of cosmic acceleration through measurements of the distribution of matter in cosmic structures. In particular, it will measure the characteristic distance scale imprinted by primordial plasma oscillations in the galaxy distribution. The projected launch date is set for 2020 and throughout its 6-year mission, Euclid will gather of the order of 50 million galaxy spectral profiles, originating from wide and deep sub-surveys. A top-priority issue associated with Euclid is the efficient processing and management of these enormous amounts of data, with scientific specialists from both astrophysical and engineering backgrounds contributing to the ongoing research. To successfully achieve this purpose, we need to ensure that realistically simulated data will be available, strictly modeled after the real observations coming from Euclid in terms of quality, veracity and volume.

4.2 Dataset

When generating a large, realistic, simulated spectroscopic dataset, we need to ensure that it is representative of the expected quality of the Euclid data. A first requirement is to have a realistic distribution of galaxies in several photometric observational parameters. We want the simulated data to follow representative redshift, color, magnitude and spectral type distributions. These quantities depend on each other in intricate ways, and correctly capturing the correlations is important if we want to have a realistic assessment of the accuracy of our proposed method. To that end, we define a master catalog for the analyses with the COSMOSSNAP simulation pipeline [93], which calibrates property distributions with real data from the COSMOS survey [94]. The generated COSMOS Mock Catalog (CMC) is based on the 30-band COSMOS photometric redshift catalogue with magnitudes, colors, shapes and photometric redshifts for 538,000 galaxies on an effective area of 1.24 deg^2 in the sky, down to an i-band magnitude of ~ 24.5 [95]. The idea behind the simulation is to convert these real properties into simulated properties. Based on the fluxes of each galaxy, it is possible to select the best-matching SED from a library of predefined spectroscopic templates. With a “true” redshift and an SED associated to each galaxy, any of their observational properties can then be forward-simulated, ensuring that their properties correspond to what is observed in the real Universe.

For the specific purposes of this analysis, we require realistic SEDs and emission line strengths. Euclid will observe approximately 50 million spectra in the wavelength (λ) range $11000 - 20000\,\text{Å}$ with a mean resolution $R = 250$, where $R = \frac{\lambda}{\Delta\lambda}$. To obtain realistic spectral templates, we start by selecting a 50% random subset of the galaxies that are below redshift $z = 1$ with Hα flux above $10^{-16}\,erg\,cm^{-2}\,s^{-1}$, and bring them to rest-frame values ($z = 0$). We then resample and integrate the flux of the best-fit SEDs at a resolution of $\Delta\lambda = 5\text{Å}$. This corresponds to $R = \frac{\lambda}{\Delta\lambda} = 250$ at an observed wavelength of $11000\,\text{Å}$, if interpreted in rest-frame wavelength at $z = 2$. For the purpose of our analysis, we will retain this choice, even though it implies higher resolution at larger wavelengths. Lastly, we redshift the SEDs to the expected Euclid range. In the particular case where we wish to vary the number of training samples, we generate more than one copy per rest-frame SED at different random redshifts. We will refer to the resampled, integrated, redshifted SEDs as "clean" spectra for the rest of the analysis.

For each clean spectrum above, we generate a matched noisy SED. The required sensitivity of the observations is defined in terms of the significance of the detection of the $H\alpha$ Balmer transition line: an unresolved (i.e. sub-resolution) $H\alpha$ line of spectral density flux $3\times10^{-16}erg\,cm^{-2}s^{-1}$ is to be detected at 3.5σ above the noise in the measurement. These requirements imply a detection rate that depends on magnitude and redshift, and Euclid will mostly detect galaxies in the redshift range $0.7 < z < 2.0$. We create the noisy dataset, by adding white Gaussian noise such that the significance of the faintest detectable $H\alpha$ line, according to the criteria above, is 1σ. This does not include all potential sources of noise and contamination in Euclid observations, such as dust emission from the galaxy and line confusion from overlapping objects. We do not include these effects as they depend on sky position and galaxy clustering, which are not relevant to the assessment of the efficiency and accuracy of redshift estimation. Our choice of Gaussian noise models other realistic effects of the observations, including noise from sources such as the detector read-out, photon counts and intrinsic galaxy flux variations.

4.3 Proposed framework

4.3.1 One-dimensional CNN

A typical CNN architecture, as described in Chapter 3, has been adopted for the problem of spectroscopic redshift estimation. Given the one-dimensional structural form of the utilized spectra, this CNN architecture must be based on one-dimensional convolutional operations. In Figure 4.1, we present a simple example of such a network. The use of pooling has been excluded from the pipeline, considering its aformentioned property that renders the network invariant to small changes of the initial input. Even though it is a significant property in the case of image classification, at the same time, this is the reason why we can't use pooling in our designated problem, given that these transformations of the original rest-frame SEDs, define the different redshifted states. By using pooling we "cripple" the network's ability to identify each different redshift, considering the suppression of these transformations.

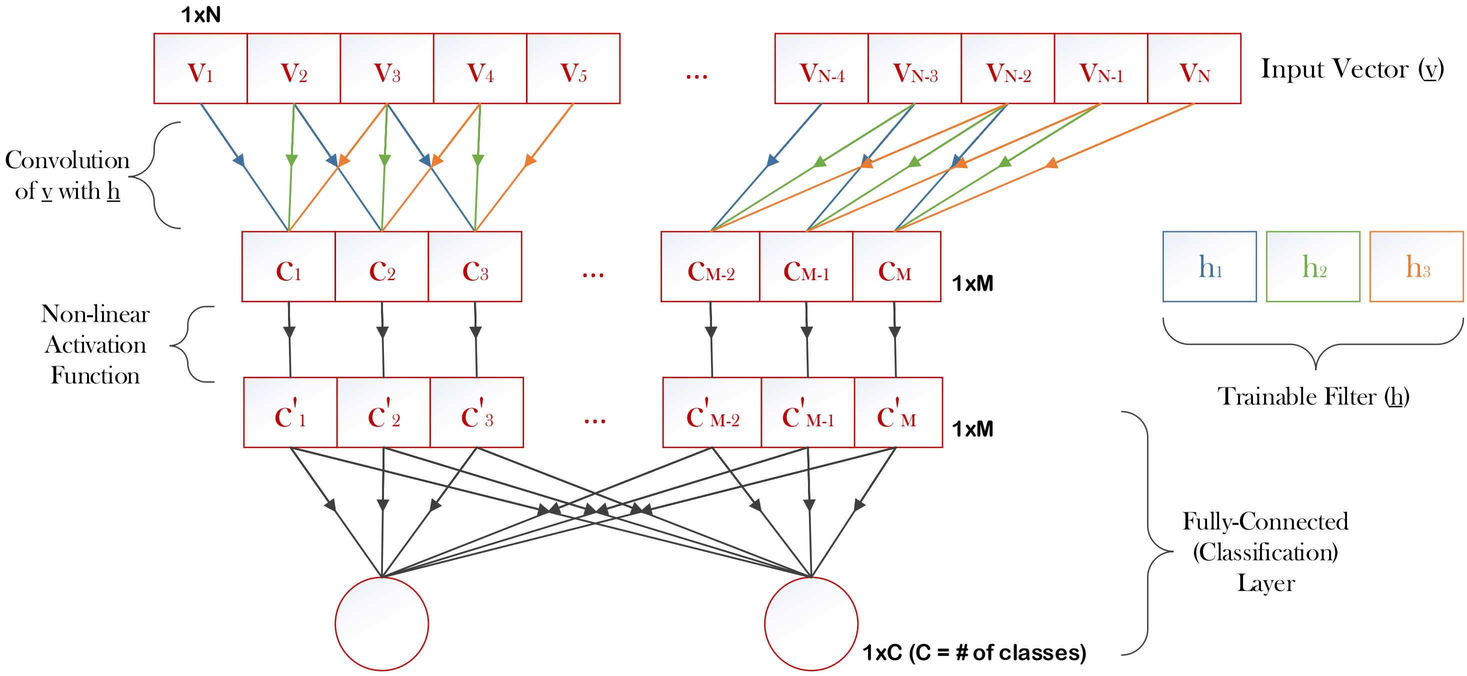


Figure 4.1: Simple one-dimensional CNN. The input vector v is convolved with a trainable filter h (with a stride equal to 1), resulting in an output vector of size $M = N - 2$. Subsequently, a non-linear transfer function (typically ReLU) is applied, element-wise, on the output vector without altering its original size. The use of pooling is excluded. Finally, a fully-connected, supervised, softmax layer is used for the task of classification. The number of the output neurons (C) is equal to the number of the distinct classes of the formulated problem (800 classes in our case). More than one fully-connected layers can be optionally applied, right before the softmax-classification layer.

4.3.2 Sample transformation: From rest-frame to redshifted spectra

Each of the clean, rest-frame spectral profiles of our initial dataset consist of 3750 wavelength-related bins. These bins correspond to the spectral density flux value of each observation, for that certain wavelength range ($\Delta\lambda = 5\mathring{A}$, $\lambda = [1252.5, 20002.5]\mathring{A}$). To create valid redshifted variations from their rest-frame equivalents, we can use the formula:

$$log(1+z) = log(\lambda_{obs}) - log(\lambda_{emit}) \Leftrightarrow 1+z = \frac{\lambda_{obs}}{\lambda_{emit}}, \qquad (4.1)$$

where λ_{emit} is the original, rest-frame wavelength, z is the redshift we want to apply and λ_{obs} is the wavelength that will ultimately be observed, for the given redshift value. This formula is linear on logarithmic scale. For the conduction of our experiments, we work on the redshift range of $z = [1, 1.8)$, which is very similar to what Euclid is expected to detect. Also, to avoid redundant operations and to establish a simpler and faster network we use a subset of the wavelength range of each redshifted example (instead of the entirety of the available spectrum), based on Euclid's spectroscopic specifications ($1.1 - 2.0\mu m \Leftrightarrow 11000 - 20000\mathring{A}$). That means that all the training and testing spectra will be of equal size $\frac{20000-11000}{\Delta\lambda} =$ 1800 bins. Finally, in both realistic and idealistic observations, a simple normalization method has been performed, such that would ensure that the structure of the data would remain unchanged, establishing at the same time numerical compatibility with the trained CNN. In the following equation, X_{max} corresponds to the maximum spectral density flux value encountered in all examples (in absolute terms, given the noisy case) and $X_{original}$ is the initial value for each feature:

$$X_{normalized} = \frac{X_{original}}{2 * X_{max}} \tag{4.2}$$

4.3.3 Label transformation: From regression to classification

For the "regression-to-classification" transition, our working redshift range of [1, 1.8) has been split into 800 non-overlapping, equally-sized slots resulting in a resolution of 0.001, consistent with the Euclid expectations. Each slot corresponds to an associated ordinal class (from 0 to 799), which in turn must be converted into the one-hot encoding format to match the final predictions procured by the softmax layer of the CNN. A certain real-valued redshift of a random spectral profile will be essentially associated with the ordinal class that corresponds to the redshift slot it belongs to. This transformed equivalent of the task of spectroscopic redshift estimation can be essentially categorized, as already mentioned in Chapter 3, as a multi-class classification problem.

4.4 Experimental evaluation and discussion

The pipeline of the proposed methodology can be briefly summarized in the block diagram of Figure 4.2. Shallow and deeper variations of the CNN have been considered, with 1,2 and 3 convolutional (+ ReLU) layers. Cross-validation has been conducted throughout the entirety of the experimental evaluation. As initial pre-experiments have shown, desirable values for the network's different hyperparameters are a kernel size of 8, a number of filters equal to 16 (per convolutional layer) and a stride equal to 1. Additionally, the adagrad optimizer [96] has been employed for the optimization step, considering its adaptable learning rate capability, that grants the network a bigger flexibility in the learning process. Finally, the use of the categorical cross-entropy loss has been preferred, among other commonly used choices.

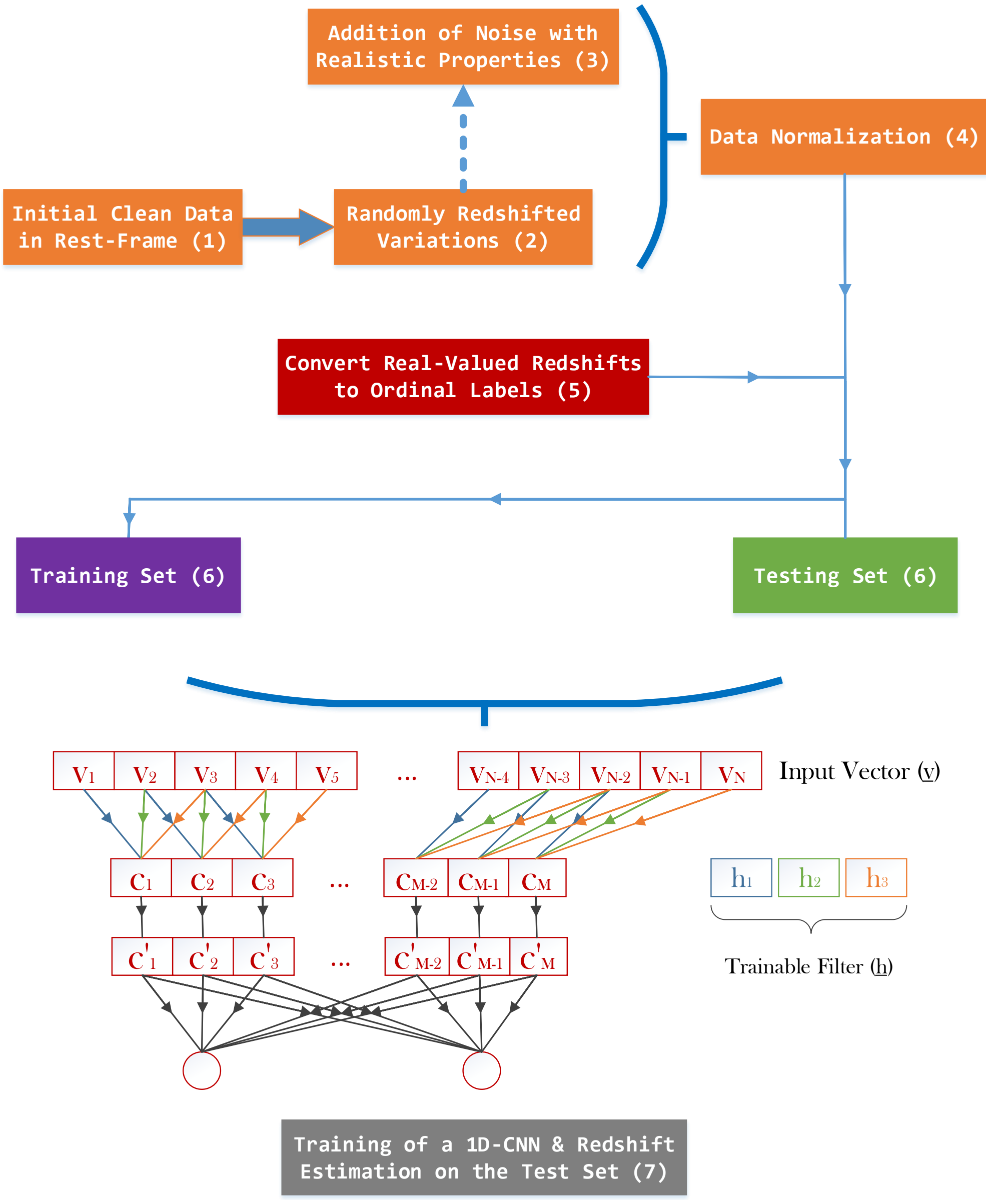

Addition of Noise with Realistic Properties (3)
Initial Clean Data in Rest-Frame (1)
Randomly Redshifted Variations (2)
Data Normalization (4)
Convert Real-Valued Redshifts to Ordinal Labels (5)
Training Set (6)
Testing Set (6)
V1
V2
V3
V4
V5
...
VN-4
VN-3
VN-2
VN-1
VN
Input Vector (v)
C1
C2
C3
...
CM-2
CM-1
CM
C'1
C'2
C'3
...
C'M-2
C'M-1
C'M
h1
h2
h3
Trainable Filter (h)
Training of a 1D-CNN & Redshift Estimation on the Test Set (7)

Figure 4.2: The flow diagram of the proposed approach. Our initial, clean rest-frame dataset (1) is used for the generation of randomly redshifted examples (2) in the redshift range of $z = [1, 1.8)$, which corresponds to that of Euclid. The data are (optionally) corrupted with noise of realistic properties (3) and are, then, normalized (4). The redshift value of each galaxy profile is categorized into its corresponding ordinal class (5) and subsequently the dataset (pairs of galaxy profiles and redshift labels) is split into the training set (6) and the testing set (6). The first is utilized by a deep Convolutional Neural Network, which is trained for the task at hand (7). The latter is used in the testing process (7), where the validity of the predicted labels (i. e. the estimated redshifts) is evaluated.

4.4.1 Idealistic observations

4.4.1.1 Impact of the network's depth

Our initial experiments revolve around the depth of the Convolutional Neural Network. We have used a fixed number of 400,000 training examples, 10,000 validation and 10,000 testing examples. Our aim is to examine the impact of increasing the depth of the model, on the final predictive outcome. Specifically, we have trained and evaluated CNNs with 1,2 and 3 convolutional layers. In all cases, a single fully-connected layer with a softmax activation function and 800 output neurons has been used for classification.

Accuracy is the basic metric that can be used to measure the performance of a trained classifier, during and after the training process. As the training goes by, we expect that the parameters of the network will start to adapt to the problem at hand, thus decreasing the total loss, and, consequently, improving the accuracy percentage. In Figure 4.3, we support this presumption by demonstrating the accuracy's rate of change over the number of training epochs. It can be easily derived that as a CNN becomes deeper, it is clearly more capable to converge on a satisfying solution. Both 2- and 3-layered networks converge very fast and

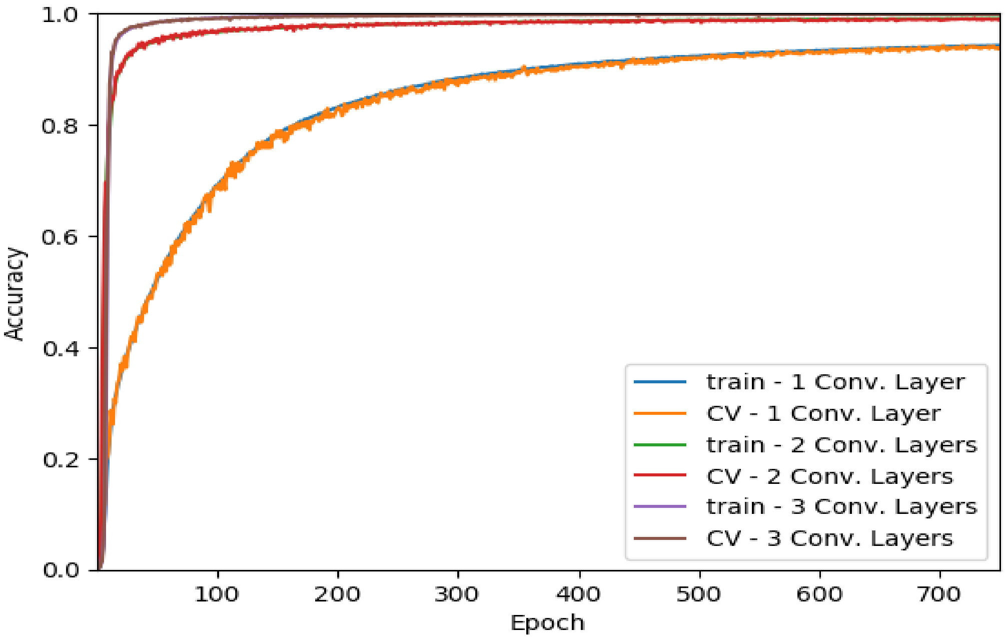


Figure 4.3: Accuracy plot for the training and cross-validation sets, for 1,2 and 3 convolutional layers. The x-axis corresponds to the number of executed epochs. In all cases we used the same 400,000 training examples.

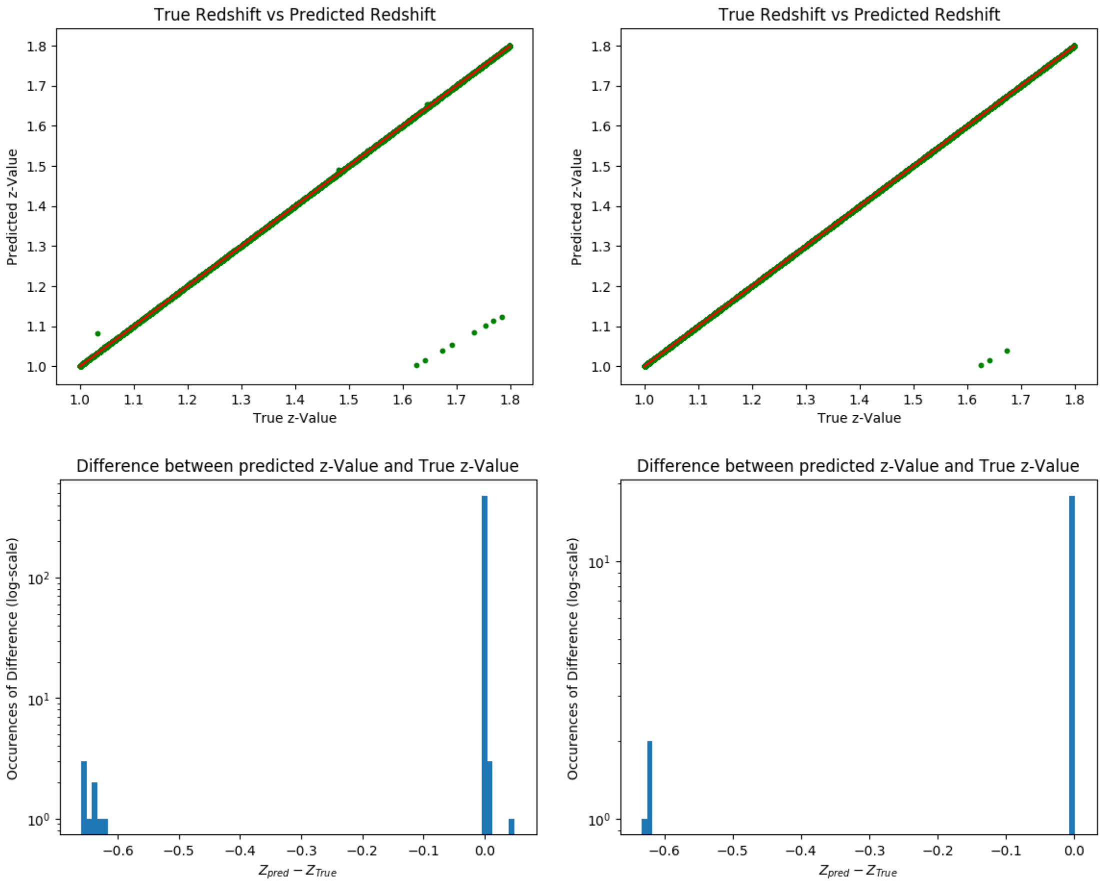


Figure 4.4: Classification scatter plots and histograms for the 1 convolutional-layer case (1^{st} column) and the 3 convolutional-layer case (2^{nd} column). The scatter plots illustrate points in 2D space that correspond to the true redshift value for each testing observation versus the predicted outcome of the given classifier, for that observation. The bar plots, on the other hand, depict the difference (in value) between the state-of-nature and the prediction, for the misclassified cases.

very close to the optimal case, with the latter, narrowly resulting in the best accuracy. On the other hand, the shallowest network is very slow and significantly underperforms compared to the deeper architectures.

More information can be deduced in Figure 4.4 (1^{st} row), where we compare, for the shallowest and for the deepest case, and per testing example, the predicted redshift value outputed by the trained classifier versus the state-of-nature. Ideally, we want all the green dots depicted in each plot to fall upon the diagonal red line that splits the plane in half, meaning that all predicted outcomes coincide with the true values. As the green dots move farther away from the diagonal, the impact of the faulty predictions becomes more significant leading to the so called catastrophic outliers. A good estimator is characterized, not only by its ability to procure the best accuracy, but also by its capacity to diminish such irregularities. At the same time, in the 2^{nd} row of the aforementioned Figure, the depicted histograms represent the actual difference in distance (positive or negative) between misclassified estimated values and their corresponding ground-truth value versus the frequency of occurence, in logarithmic scale, for each case. Negative values of difference correspond to outliers that exist in the lower right half of the scatter plot and positive values correspond to outliers that exist in its upper left part. Once again, the deeper network not only leads to a significantly smaller number of errors, compared to the 1-layered case, but also to a more limited amount of catastrophic failures.

4.4.1.2 Data-driven analysis

In this setting, we will explore the significance of broad data availability in the overall performance of the proposed model. As mentioned before, Big Data have revolutionized the way Artificial Neural Networks perform [29], serving as the

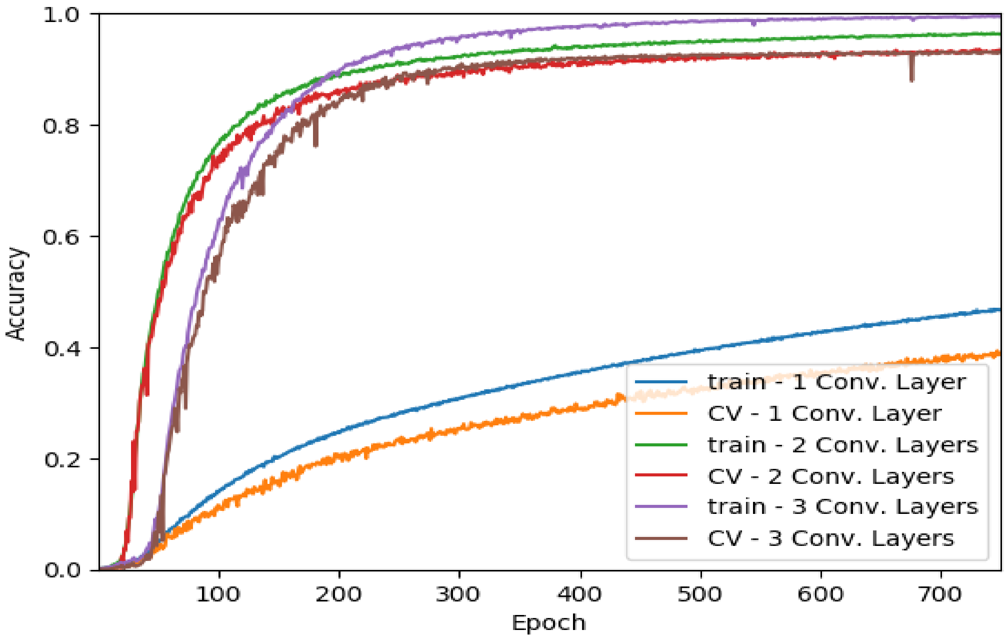


Figure 4.5: Training and cross-validation accuracy, for 1,2 and 3 convolutional layers, using a significantly decreased amount of training observations (40,000). Overfitting is introduced, to various extents, based on each case.

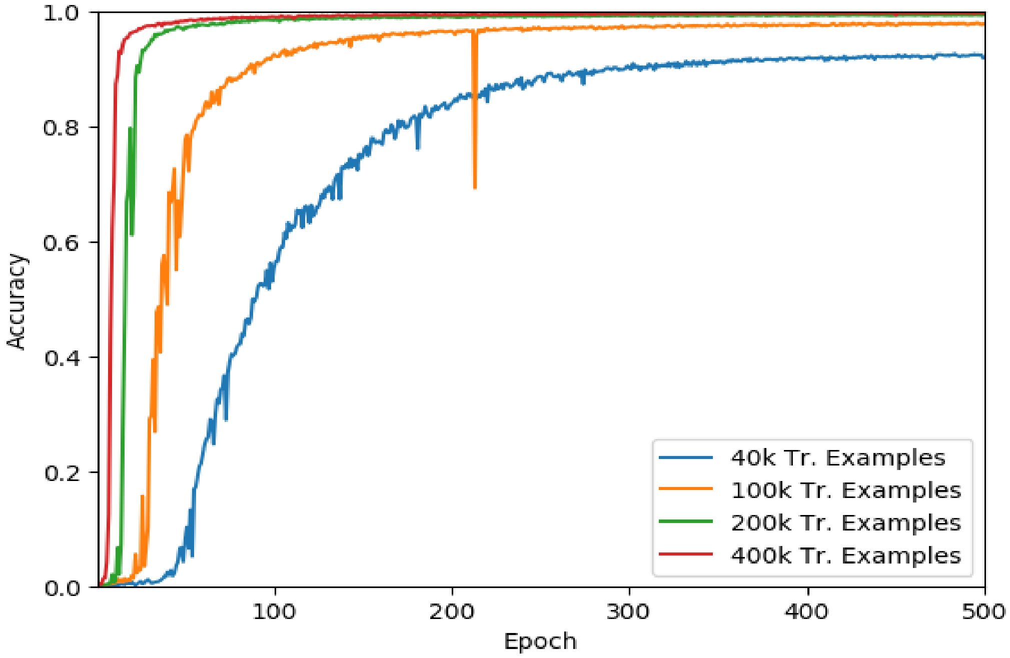


Figure 4.6: Validation performance of a 3-layered network, using larger and more limited in size datasets. In all cases the training accuracy (not depicted here) can asymptotically reach 100% accuracy, after enough epochs.

main fuel for their conspicuous achievements. Figure 4.5 illustrates the behavior of the same network variations as in previous experiments (1,2 and 3 convolutional layers), using this time a notably more limited, in size, training set of observations compared to the previous case. Specifically, we have lowered the number of training examples from 400,000 to 40,000, namely to one-tenth. Compared to the results we have previously examined in Figure 4.3, we can evidently identify a huge gap between the performance of corresponding models with copious vs more limited amounts of data. Also, it is adequately obvious that in all three cases overfitting is introduced, to various extents, with overoptimistic models that perform well in the training set, developing a decaying performance on the validation and the testing examples.

As a second step, we want to preserve the network's structural and hyper-parametric characteristics immutable, whereas altering the amount of training observations utilized in each experimental recurrence. We have deployed a scaling number of training examples beginning from 40,000 observations, then to 100,000 and finally to 200,000 and 400,000 observations for the training of a 3-layered CNN (3 convolutional + 1 fully-connected layer). As shown in Figure 4.6, while we increase the exploited amount of data, the curve of the validation accuracy also increases in a smoother and steeper pace, until convergence. On the contrary, when we use less data, the line becomes more unstable, with a delayed convergence and a poorer final performance. It is very important to state that despite the fact that the training accuracy can asymptotically reach in all cases 100% accuracy, after enough epochs, the same doesn't apply for the validation accuracy (and respectively for the testing accuracy) with the phenomenon of overfitting taking its toll, mostly in the cases where the volume of the training data is not enough to handle the complexity of the network, failing to generalize in the long term. As we will observe in more detail in the noisy-data case, regularizing techniques, such as dropout, can significantly help battle this phenomenon, but not in a way that the difference between the training and the validation performance will be completely commensurated.

4.4.1.3 Tolerance on extreme cases

Before advancing to noise-afflicted spectral profiles it is worthsome to investigate some extreme cases, concerning two astrophysical-related aspects of the data. As presented before, one of our main novelties is the realization of the redshift estimation task as a classification task, guided by the specific redshift resolution that Euclid can achieve, and leading to the categorization of all possible detectable redshifts into 1 of 800 possible classes. As a first approach, we want to extend our working resolution to a double precision, specifically from 0.001 to 0.0005, meaning

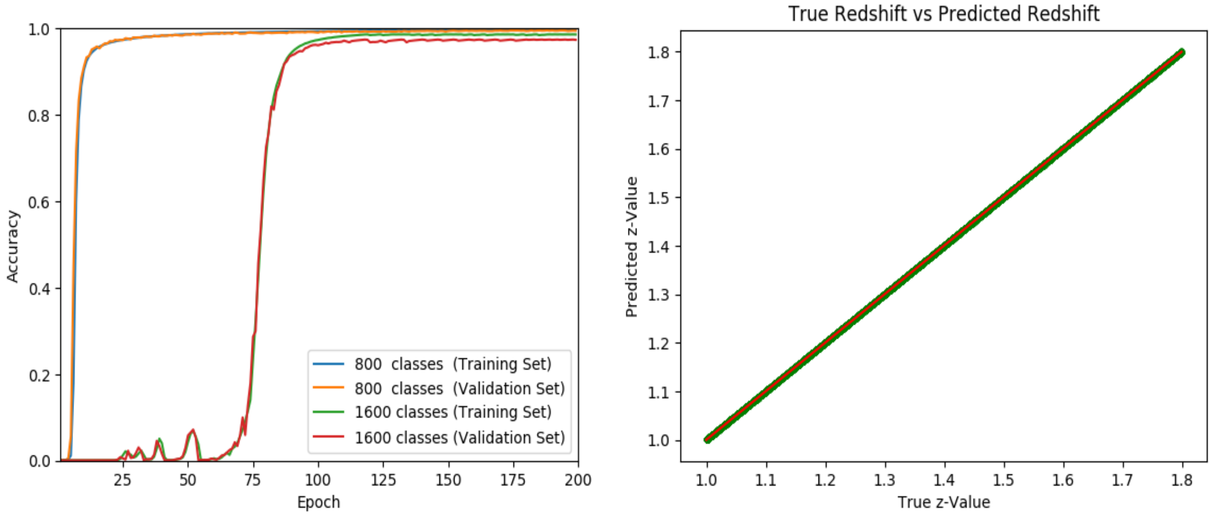


Figure 4.7: Performance of a 3-layered network trained with 400,000 training examples. In the first plot we compare the cases where the redshift estimation problem is transformed into a classification task, with the use of 800 versus 1600 classes. In the second plot, we present the scatter plot of the predicted result versus the state-of-nature of the testing samples, only for the case of 1600 total classes.

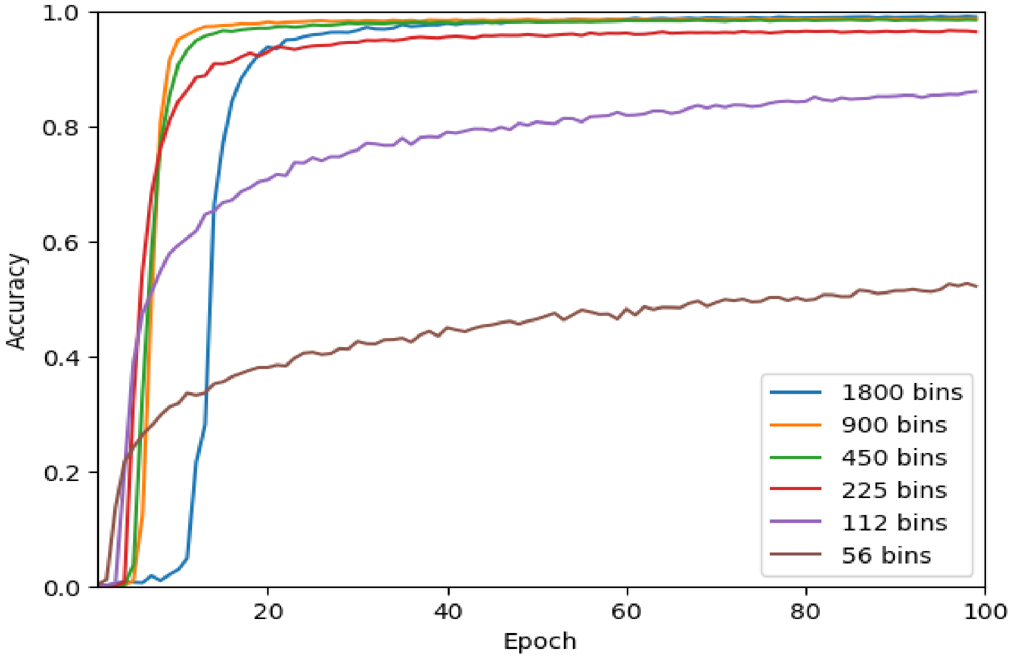


Figure 4.8: Validation performance of a 3-layered network trained with 400,000 training examples. We want to examine the behavior of the model, when trained with data of reduced dimensionalities.

that the existing redshift range of [1, 1.8) will be split into 1600 classes instead of 800.

As observed in Figure 4.7, doubling the total number of possible classes has a non-critical impact in the predictive capabilities of our approach, given the fact that at convergence, the model produces a similar outcome for the two cases. Despite the fact that doubling the classes leads to a slower convergence, a behavior that can be attributed to the drastical increase of the parameters of the fully-connected layer, the network is still able to estimate successfully, in the long term, the redshift of new observations. Furthermore, as depicted in the scatter plot of the same figure, we can deduce that increasing the predictive resolution of the CNN can lead to an increase in the total robustness of the model against catastrophic outliers, given the fact that none of the misclassified observations in the testing set exist far from the diagonal red line, namely the optimal error-free case.

In our second approach, we want to challenge the network's predictive capabilities, when presented with lower-dimensional data, and to essentially define where is the turning point where the abstraction of information becomes more of a strain, rather than a benefit. Having to deal with data that exist in high-dimensional spaces (like in the case of Euclid), can become more of a burden, rather than a blessing, as described by Richard Bellman [97], with the introduction of the very well-known term, of the “curse of dimensionality”. In our case, data dimensionality can be derived by splitting the operating wavelength of the deployed instrument into bins, where each bin corresponds to the spectral density flux value of the wavelength range it describes. Euclid operates in the range of $1.1 - 2.0\,\mu m$ with a bin size of $\Delta\lambda = 5\mathring{A}$, which implies 1800 different bins per observation. To reduce that number, we need to increase the wavelength range per bin by merging each bin with neighboring cells, namely by adding together their corresponding spectral density flux values. Thus, we can assert that by lowering the dimensionality of data using this methodology, we can accomplish the concentration of existing information in cells of compressed knowledge, rather than discarding redundant information.

In Figure 4.8, we can conclude that when dealing with clean data, the reduction of the number of total wavelength bins into more manageable numbers can result, not only in a congruent performance compared to the initial model, but also into a faster convergence. On the other hand, oversimplifying the model can be deemed inefficacious, if we take into account the decline of the achieved accuracy in the three lower-dimensional cases. A moderate decline in the performance becomes

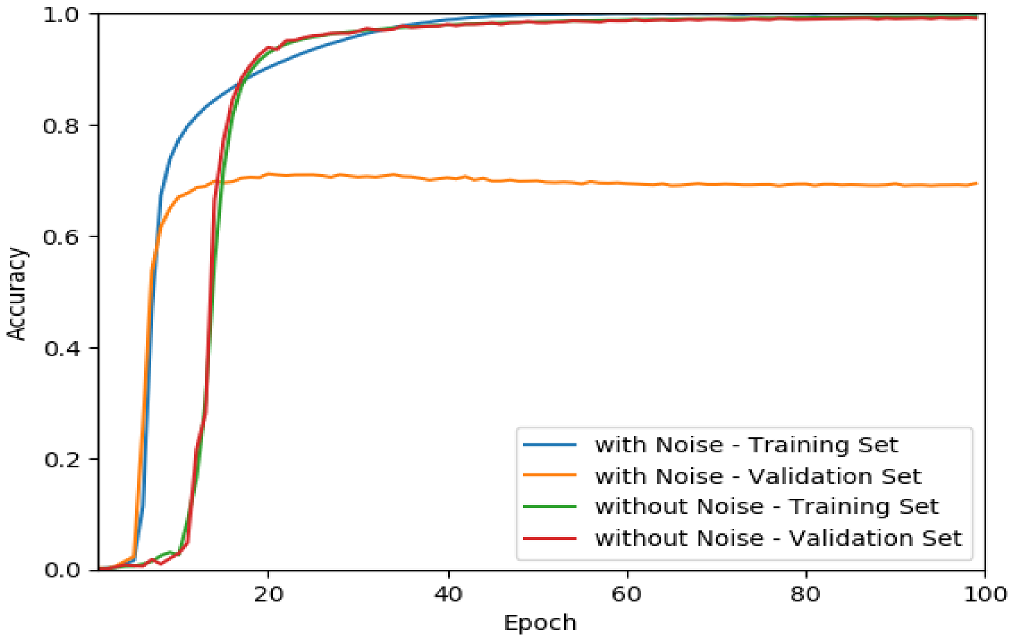


Figure 4.9: Comparison of the model’s performance, trained with clean and with noisy data (400,000 in both cases). The 3-layered neural network utilizes the same hyperparameters, in both cases, without any form of regularization.

visible in the case of 225 bins, with a more aggressive degeneration of the model occurring in the two remaining cases.

4.4.2 Realistic observations

The availability of idealistic data presumes the ambitious scenario of a reliable denoising technique for the spectra, prior to the estimation phase. Although successful methods have been developed in the past [98], [99], our main aim is to integrate implicitly the denoising operation in the training of the CNN, meaning that it should be able to distinguish the noise from the relevant information by itself, without depending on an intermediate party. This way, an autonomous system can be established, with a considerable robustness against noise, a strong feature extractor and essentially a reliable predictive competence. To that end, we have directly used the noisy observations described in section 4.2, as the input of the adopted CNNs.

A comparison between the idealistic and the realistic scenarios constitutes the first step that will lead to an initial realization of the difficulty of our newly set objective. In Figure 4.9, we observe that training a noise-based model with a number of observations that has been previously proven to be sufficient in the clean-based

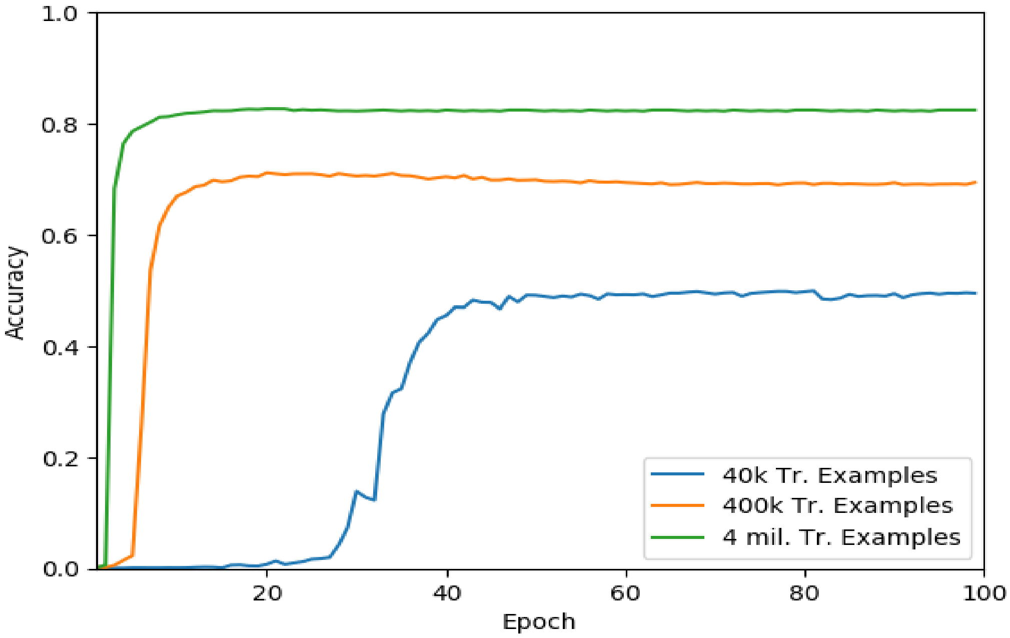


Figure 4.10: Accuracy on the validation set (noisy dataset), for different sizes of the training set. No regularization has been used.

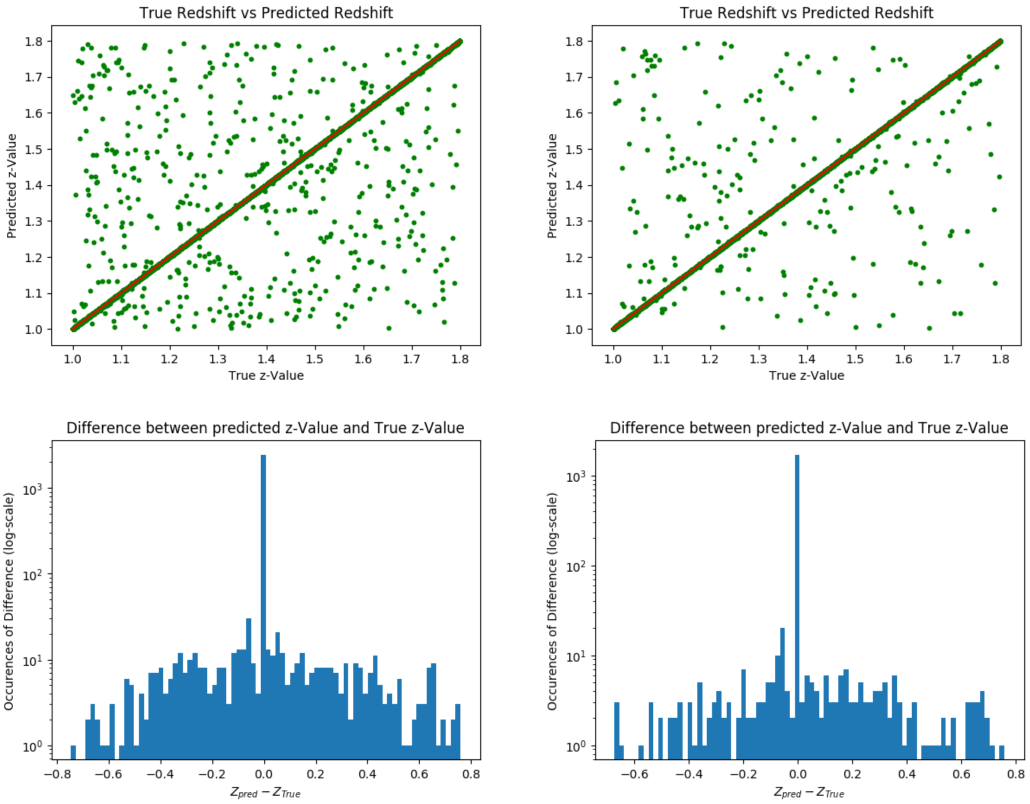


Figure 4.11: Classification scatter plots and histograms for the realistic case, for 3-layered networks trained with 400,000 training examples (column a) & 4,000,000 training examples (column b).

case, leads to an exaggerated performance during the training process that doesn't apply to newly observed spectra, hence leading to overfitting. Clean data are notably simpler than their noisy counterparts, which in turn are excessively diverge, meaning that generalization in the latter case is seemingly more difficult. The main intuition to battle this phenomenon lies in drastically increasing the spectral observations used in training. Feeding the network with bigger volumes of data can mitigate the effects of overfitting, given the fact that the observed set of spectra tends to become so large that it befits the general case. The above intuition is strongly supported by Figure 4.10, where we compare the performance of similar models when trained with different-sized sets. Preserving constant hyperparameters and not utilizing any form of regularization, we can conclude that, just by increasing in bulk the total amount of data, the network's generalization capabilities can also increase in a scalable way. As a final remark, the increased predictive difficulty established by the noisy scenario is also demonstrated in Figure 4.11.

The drastical increase in the number of misclassified samples is more than obvious, compared to the previously examined case in Figure 4.4, leading to an abrupt rise in the amount and variety of the different catastrophic outliers. Nevertheless, the faulty predictions that lie approximate to the corresponding ground-truths constitute the majority of the mispredictions, as verified by the highly populated green mass around the diagonal red line (scatter plots) and the highest histogram column bordering the origin, in the case of the histograms.

4.4.2.1 Impact of regularization

The effects of regularization are illustrated in Figure 4.12, in two different settings, one with a training set of 400,000 examples and another with a training set of 4,000,000 examples. In the case of batch normalization, we inserted an extra batch normalization layer after each convolutional layer and after ReLU. Although in literature [11], the use of batch normalization is proposed before the non-linearity, in our case extensive experimental results suggested otherwise. Dropout was introduced only in the fully-connected layer, with a value of p equal to 0.5, which appeared to yield the best results compared to other choices. It is important to note that the use of dropout can be also applied in the case of the convolutional layers, however without an out-of-the-ordinary change in the final performance.

As we can see in both examined cases, dropout can clearly help enhance the

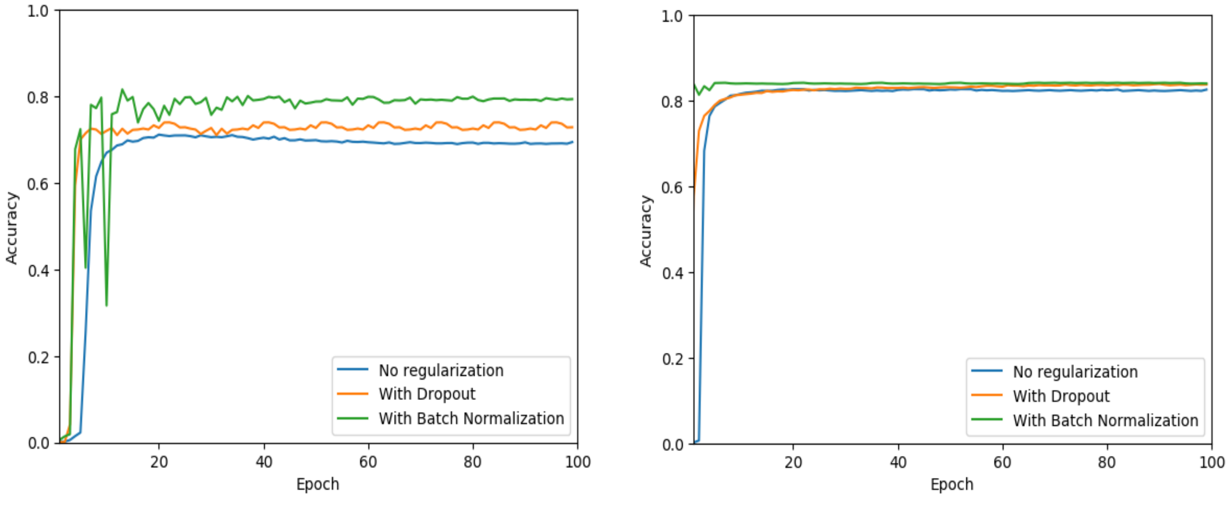


Figure 4.12: Impact of regularization, in regard with the size of the training set. In the left plot, a network trained with 400,000 observations is illustrated, while in the right plot 4,000,000 training examples have been utilized. The reported accuracy is associated with the validation set.

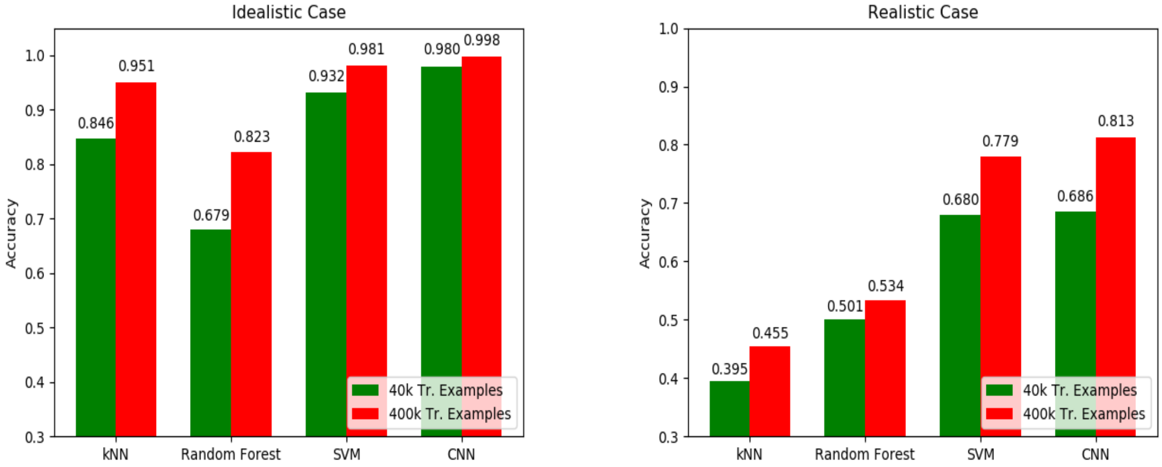


Figure 4.13: Comparison bar plots for the k Nearest Neighbours, Random Forest, Support Vector Machine and Convolutional Neural Networks algorithms. We present the best case performance on the test set, for each classifier, in the idealistic and the realistic case, with a limited and an increased amount of training data.

network's performance in the validation set, leading to an increase in the accuracy by ~0.5% in the worst case and ~1.5% in the best case. This is not a groundbreaking increase per se, but it is worth mentioning nonetheless. On the other hand, batch normalization appears to have a bigger regularizing effect in improving the accuracy of the model, yielding a tremendous increase by almost 10% in the case of 400,000 training examples, and a significantly lower gain of ~2% when trained with 4,000,000 observations. In this final case, even though batch normalization still leads to the best performance, its difference compared to dropout is almost negligible.

4.4.3 Comparison with other classifiers

In this sub-section, we want to compare the best-case performance of the proposed model on the task of spectroscopic redshift estimation, against the performance of other popular classifiers, namely k Nearest Neighbours [100], Random Forest [101] and Support Vector Machine [102]. The bar plots in Figure 4.13 corroborate the claim that Convolutional Neural Networks reign supreme as the most effective algorithm, in all examined cases. The main competitor, in both idealistic and realistic scenarios, stands in the case of the Gaussian-kernel Support Vector Machine, which in our problem is inexpedient to use given the fact that SVMs are most effective in binary classification scenarios or in cases where the total amount of unique classes is limited. With 800 possible classes to predict, both techniques of one-vs-all and one-vs-one multi-class classification require the training of a large amount of individual classifiers, namely 800 and $(800 * 799 / 2) = 319,600$ accordingly. On the other hand, k Nearest Neighbours and the Random Forest methodologies significantly underperform, failing to cope with the noisy variations of the data, even with an increased amount of training examples.

4.4.4 Levels of confidence

One of the benefits of the transformation of the redshift estimation problem to a classification procedure manifests in the association of each estimation, with a level of confidence of the network's certainty that the predicted outcome corresponds to the true redshift value. Using the probabilities produced by the softmax function, we can extract valuable information about the network's reliability, as illustrated in Figure 4.14, where we examine the derived confidence of the best-case trained networks for both idealistic and realistic datasets. In the idealistic scenario, we can observe that the trained model is generally very confident about the validity of its predictions leading to a very steep cumulative curve in the transition from the 90% to 100% . As also verified by the corresponding histogram, most of the predictions are associated with a very high probability that lies in the range of (0.9, 1], with a decreased frequency of occurrence as the levels of confidence decrease. This is a very desirable property, given the fact that we want the network to be certain about its designated choice, leading to concrete estimations that are not subject to dispute. In the realistic scenario, although the total confidence of the trained network clearly drops, as expected, still the high confidence choices remain dominant in quantity, compared to the lower cases which mostly correspond to the misclassified observations.

4.4.5 Intermediate representations

In this final paragraph, we will briefly examine the undergoing transformation of the input testing observations, as they flow deeper into the trained network. Specifically, we will comment on the derived intermediate representations of randomly chosen filters of different layers. As previously discussed, Convolutional Neural Networks are excellent feature extractors and can successfully distill important knowledge from raw data, even when afflicted with high levels of noise. In the case of the clean spectra, Figure 4.15 outlines that the salient effect of randomly chosen filters is the gradual removal of the continuum of the derived intermediate representations, preserving only the characteristic emission and absorption lines of the given galaxy profiles (most importantly the $H\alpha$ line). Removing the continuum is one of the key steps that any spectroscopic analysis requires, while at the same time, distinguishing these lines constitutes a key characteristic to a better discrimination of the different redshift classes. The introduction of mirror amplitudes in the negative half-plane is not of specific importance, given their

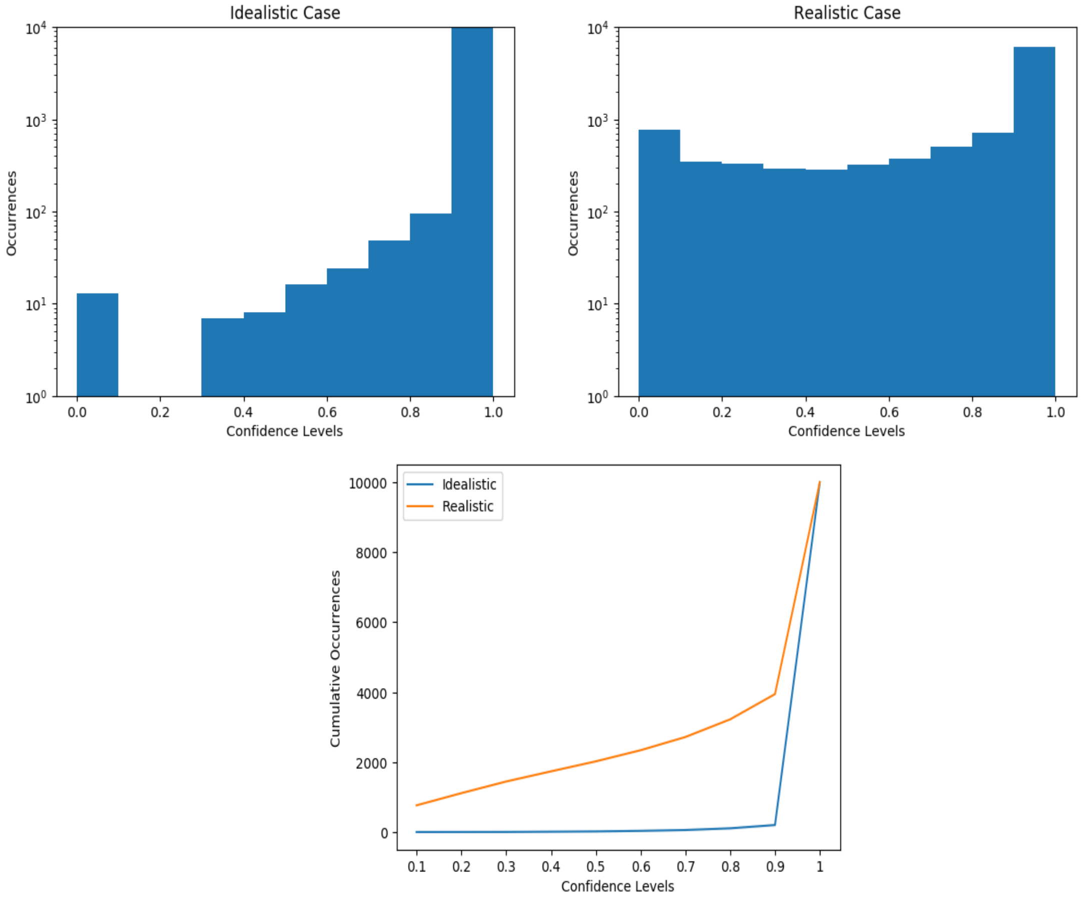
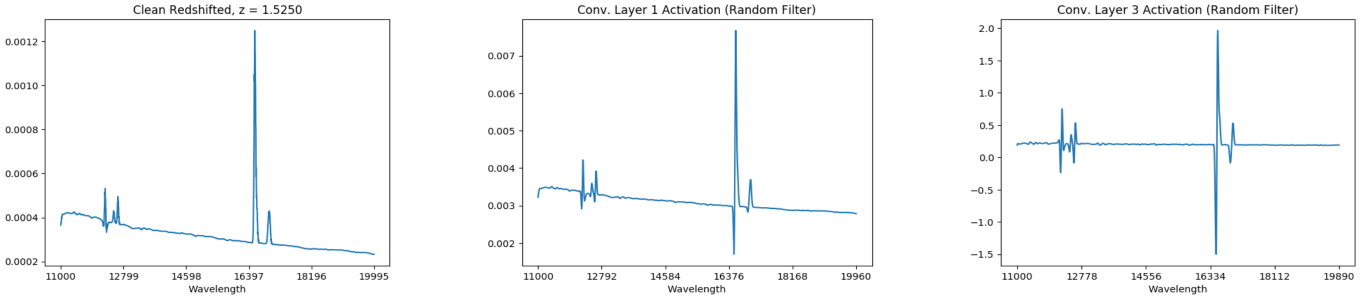


Figure 4.14: Levels of confidence derived by softmax in the testing set. The lower plot depicts the cumulative occurrences per level of confidence, for both examined cases. For example, the y-axis value that corresponds to the x-value of 0.4 represents the number of testing observations that obtain a predictive output with a confidence that is less than or equal to 0.4. The upper left (idealistic case) and upper right (realistic case) histograms, exhibit a similar scenario, but not in a cumulative form (and in logarithmic scale).

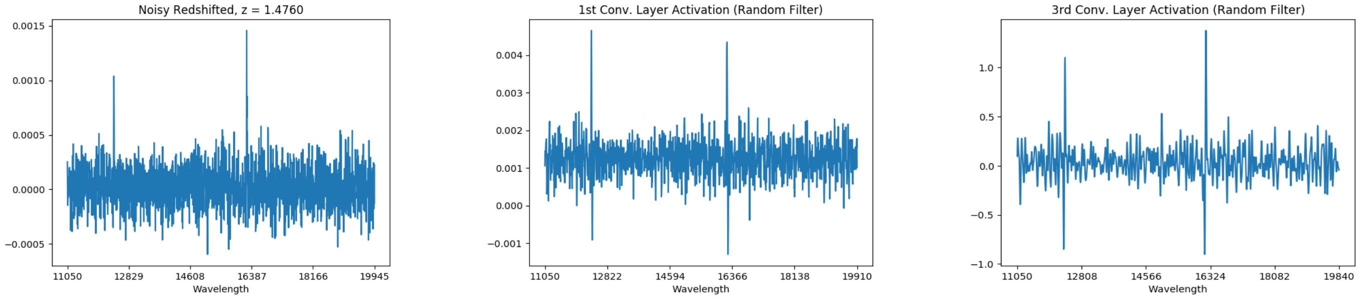


(a) Clean redshifted spectral profile

(b) Activation of 1^{st} Conv. Layer

(c) Activation of 3^{rd} Conv. Layer

Figure 4.15: A random testing example (clean clase) and the corresponding activations of the 1^{st} and the 3^{rd} convolutional layers.



(a) Noisy redshifted spectral profile

(b) Activation of 1^{st} Conv. Layer

(c) Activation of 3^{rd} Conv. Layer

Figure 4.16: A random testing example (noisy case) and the corresponding activations of the 1^{st} and the 3^{rd} convolutional layers.

immediate nullification by the succeeding ReLUs. Furthermore, in the case of the realistic observations in Figure 4.16, even though the outright removal of irrelevant information may not be easily achievable, given the low signal-to-noise ratio of the observed spectrum, essentially the network is able to perform a partial denoising of the examined profile, gradually isolating the desired peaks from the faulty discontinuities.

Chapter 5

Case study: Deep learning for multi-label land cover scene categorization on the modified UC-Merced dataset

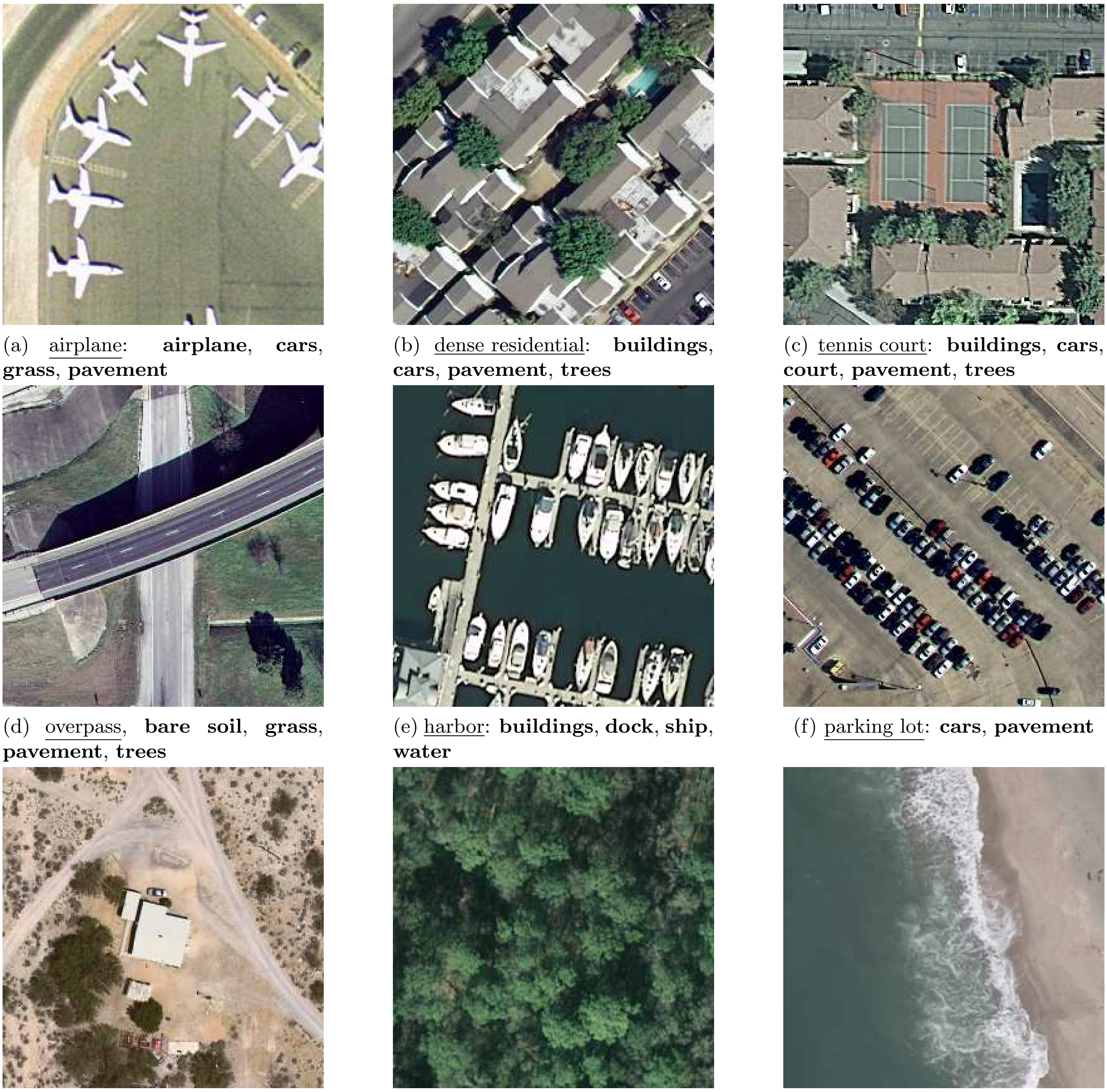
In this case study, we concentrate on the problem of multi-label land cover scene categorization, on aerial high-resolution images. At first, we introduce the UC Merced Land Use Dataset in both its original, multi-class configuration and in its modified, multi-label version. Then, we examine the proposed framework, namely the needed adjustments we need to perform on the CNN's architecture to conform with the problem of multi-label classification. Moreover, we propose an online variation of the image data augmentation technique, given the limited size of the adopted dataset, and we present the applied cross-validation methodology, the hyperparameters and the evaluation metrics used. Finally, in the last section of this chapter, we demonstrate our experimental findings and we comment on the deduced results.

5.1 The UC Merced land use dataset

The UC Merced Land Use Dataset[1] (UC-Merced) [6] includes aerial images extracted from larger images of the USGS National Map Urban Area Imagery Collection[2] and it has been widely used in various remote sensing applications. It is a high-resolution dataset that contains 2100 different images (of 256x256 pixels) evenly split among 21 mutually exclusive, unique classes (Table 5.1). UC-Merced has been considered in many different land cover categorization methodologies that concentrate on the single-label scenario. For the purpose of our experiments,

[1] http://weegee.vision.ucmerced.edu/datasets/landuse.html

[2] https://nationalmap.gov/ortho.html



(a) airplane: **airplane**, **cars**, **grass**, **pavement**

(b) dense residential: **buildings**, **cars**, **pavement**, **trees**

(c) tennis court: **buildings**, **cars**, **court**, **pavement**, **trees**

(d) overpass, **bare soil**, **grass**, **pavement**, **trees**

(e) harbor: **buildings**, **dock**, **ship**, **water**

(f) parking lot: **cars**, **pavement**

(g) sparse residential: **buildings**, **cars**, **chaparral**, **pavement**, **sand**, **trees**

(h) forest: **trees**

(i) beach: **sand**, **sea**

Figure 5.1: Example images of the UC Merced dataset. The underlined annotations indicate the multi-class association for each image. On the other hand, the labels marked with **bold** are the corresponding multi-label annotations, examined in this case study.

we have utilized UC-Merced with a completely re-imagined labelset, more suited for the multi-label case. Specifically, the new labelset consists of 17 different labels, listed in Table 5.2, and each image in UC-Merced can be categorized to one through seven of those labels, in accordance with its content. The labelset modification has been conducted by the authors in [27], in the context of the BigEarth[3] research project. An example set of UC-Merced images is illustrated in Figure 5.1, where each image is tagged with both its multi-class and multi-label annotations.

5.2 Proposed framework

In this section the deployed methodologies and the accompanying concepts will be shortly presented. The pipeline of the proposed approach is briefly described in the block diagram in Figure 5.2.

5.2.1 Modification of the typical CNN architecture

5.2.1.1 Sigmoid outputs and thresholding

The first modification that we need to perform on a Convolutional Neural Network to address a multi-label task is the substitution of the output softmax layer. Given that softmax is normalized to strictly output probabilities that will always add up to one, it is considered an ideal choice for a single-label multi-class scenario, where all classes are mutually exclusive, albeit not as good of a choice for the multi-label case. With softmax, as the trained system's confidence for the prediction of a specific class increases, there is a need to enhance the probability score of that certain class and concurrently to decrease the respective probabilities of the remaining

Table 5.1: The original labelset of UC-Merced (single-label case) and the number of samples associated with each label-class.

label	# of samples
agricultural	100
airplane	100
baseball diamond	100
beach	100
buildings	100
chaparral	100
dense residential	100
forest	100
freeway	100
golf course	100
harbor	100
intersection	100
medium residential	100
mobile home park	100
overpass	100
parking lot	100
river	100
runway	100
sparse residential	100
storage tanks	100
tennis court	100

[3] http://bigearth.eu/index.html

Table 5.2: The modified labelset of UC-Merced (with a multi-label perspective) and the number of samples associated with each label.

label	# of samples
airplane	100
bare soil	633
buildings	696
cars	884
chaparral	119
court	105
dock	100
field	106
grass	977
mobile home	102
pavement	1305
sand	389
sea	100
ship	102
tanks	100
trees	1015
water	203

classes. This is an undesirable property for the multi-label approach, given that in most cases more than one labels must be associated with each sample. Instead of selecting the single label with the maximum probability score, we need to select all those labels with a score large enough that has rendered them active. To that

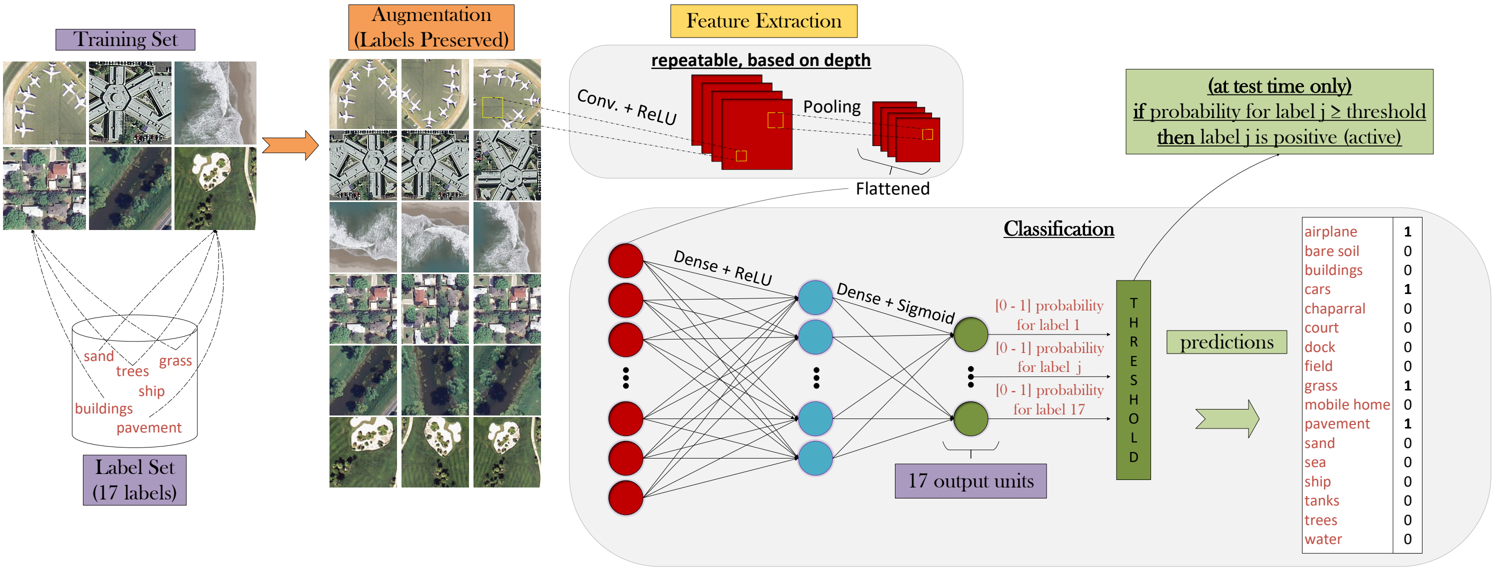


Figure 5.2: The pipeline of the proposed methodology. Each batch of the training set is dynamically and randomly augmented at every iteration of the training process. The augmented data are fed into a deep CNN with sigmoid output units, which is trained with backpropagation based on a chosen loss function. Probability thresholding is used for the predictions only at test time. The test set is excluded from the data augmentation methodology.

end, for each individual output unit of the CNN, we must be able to efficiently transition from its predicted score, to the binary decision of designating a label as active or rejected. Considering that the number of active labels can be different for each observation, there are no guarantees that a sufficiently high softmax probability score for a certain label, for a given sample, will also be regarded as high for another sample of the dataset. As a result, in our method we choose to employ the following sigmoid output activation function, yielding probability scores without constraints concerning their sum:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{5.1}$$

During inference, translating the probabilities associated with each output node into a binary prediction for each label, requires the utilization of an appropriately defined threshold such that a label is considered active if the associated score exceeds the threshold. However, it is important to note that using thresholding during training is not suggested. Given that the threshold operator converts the predicted probability score values to constant numbers (0 for a rejected label and 1 for an active label), it leads to a zero gradient calculation for all output units, causing the backpropagation process to malfunction.

5.2.1.2 Pertinent loss function choices

A second adjustment that we need to undertake, in order to adapt the network to a multi-label problem is the adoption of pertinent to the task at hand loss function choices. The use of popular losses, such as the mean squared reconstruction error and the categorical cross entropy, may be appropriate in a multi-class classification scenario, but in the multi-label case it is highly ineffective. In the defined problem, most common choices include the multi-label alternative of the categorical cross-entropy, namely the binary cross entropy (BCE), and the Poisson loss. Concerning the latter, it is actually a measure of the divergence of each predicted label vector's distribution from a Poisson ground-truth distribution, which in our case is true given its binary form and the sparsity of ones. The corresponding formulae of the two adopted loss functions are defined as:

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n} [\, y^{(i)} \log{(\hat{y}^{(i)})} + (1 - y^{(i)}) \log{(1 - \hat{y}^{(i)})}] \quad \text{(BCE)}, \tag{5.2}$$

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}^{(i)} - y^{(i)} \log{(\hat{y}^{(i)})}) \quad \text{(Poisson)}. \tag{5.3}$$

The scalar value n represents the number of training samples associated with each training batch, $y^{(i)}$ corresponds to the ground-truth label vector of the i-th sample of the batch and $\hat{y}^{(i)}$ corresponds to the predicted score vector for the same sample.

5.2.2 Online image data augmentation

Most state-of-the-art deep learning architectures contain a massive amount of trainable parameters that often exceed the order of magnitude of a million, or even tens of millions. Having to deal with such complex models can easily result in overfitting the training dataset, if its size is not sufficiently large. For an image dataset, data augmentation can be a surprisingly effective solution and can significantly increase the total number of available, annotated training examples through a variety of simple transformations. These transformations include, among others, the rotation of the image by different degrees, image rescaling, horizontal and vertical flips, translations to the x and y-axis, and the addition of noise. By slightly changing the content of each image without tampering with its semantic information, thus preserving the same label associations, the CNN can be "tricked" to perceive the existence of a significantly larger training dataset than the one initially available. The intuition behind image data augmentation originates from the fact that a CNN must be able to discriminate between significant and irrelevant features. Feeding the network with different variations of the same image, paired with the same corresponding labels, can greatly improve its ability to distinguish between those features and, accordingly, to preserve only the relevant information. An airplane will still be an airplane regardless of its position in the given image or whether it is facing to a certain direction. However, the same allegation cannot be made if you remove its wings or its tail, crucial characteristics to its identification.

An online methodology has been deployed for data augmentation, meaning that each training batch is dynamically augmented at every training iteration, altering all the images of the batch on the fly. Compared to the offline alternative, dynamic augmentation negates the memory requirements of a bigger, statically defined dataset and reinforces, at the same time, the generalization capabilities of the network, considering that the CNN will rarely or never process twice the exact same sample.

5.2.3 Hyperparameters and Cross-Validation

Our initial experiments were conducted on a deep CNN with 3 convolutional layers, 1 dense-ReLU layer and 1 dense-sigmoid layer, with a global confidence threshold of 0.45 for all labels. Shallower networks were also examined, but as presented in Table 5.3 their performance was poorer compared to the deeper alternative. An increasing number of 128, 256 and 512 trainable filters has been deployed, per convolutional layer, with a kernel size of 3x3 and a stride of 2x2 each. For the max pooling operator, a non-overlapping window size of 2x2 has been used on all applicable convolutional layers. For the majority of our experiments the binary cross entropy loss has been utilized, along with the adagrad optimizer [96], but, in general, both Poisson and BCE have been found to perform relatively close (Table 5.4).

Different experimental variations have been considered. In each variation the

Table 5.3: Performance evaluation based on the depth of the CNN. Data augmentation is utilized.

Depth	**Accuracy**	**Precision**	**Recall**	**F-score**
1 Conv. Layer	19.32 (4.24)	55.23 (2.50)	21.68 (7.58)	31.14
2 Conv. Layers	77.72 (0.72)	85.17 (0.31)	84.96 (0.99)	85.06
3 Conv. Layers	**81.32 (0.82)**	**87.95 (0.84)**	**89.32 (0.67)**	**88.63**

Table 5.4: Performance evaluation for the defined loss functions.

Loss	**Accuracy**	**Precision**	**Recall**	**F-score**
BCE	81.32 (0.82)	87.95 (0.84)	89.32 (0.67)	88.63
Poisson	**81.74 (1.12)**	**88.00 (1.42)**	**90.48 (2.12)**	**89.22**

same setting of the previous paragraph has been applied, altering, each time, only one of the available hyperparameters. Each experimental variation has been trained and tested 5 times and for each adopted performance metric, an average value has been computed. Specifically, for each of these 5 experiments, the UC-Merced dataset is randomly split into a training set of 1600 samples and a test set of 500 samples for evaluating the performance. In all the experiments, the network has been trained for 300 epochs, with a training batch size of 10. Furthermore, batch normalization has been examined, leading to a faster convergence of the training process, as well as a minor increase in the final performance.

The transformations that were used for dynamic data augmentation include image rotation, translation and horizontal and vertical flips. Specifically, in the case of image rotation we used a degree range for random rotations of $[-45, 45]$°and for image translation we performed random shifts in a maximum range of the 20% of the total height or width of the image. Considering that the augmentation of the training set is dynamic, the size of augmented data can be calculated by multiplying the initial size of the training set by the number of epochs the network was trained. In our case, we end up with $1600 \times 300 = 480,000$ training samples. In order to minimize any potential cross-contamination between the training and the test sets, data augmentation is not performed on the test set.

5.2.4 Metrics

The conventional performance-evaluation metric of accuracy, adopted in the single-label scenario, is not suited for the multi-label case, examined in this chapter. To that end, the metrics of precision and recall have been utilized for a reliable evaluation of the performance of the proposed methodology and for a consistent comparison with the current state-of-the-art in [27]. Precision is a measurement of the percentage of the positively predicted labels that are active in ground-truth. On the other hand, recall is defined as the fraction of all active labels that are successfully predicted as such. Moreover, the F-score measure is adopted, as the harmonic mean of both precision and recall, along with a multi-label variation of the metric of accuracy. The corresponding formulae of the employed metrics are defined as:

$$\text{Precision} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Z_i|}, \tag{5.4}$$

$$\text{Recall} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i|}, \tag{5.5}$$

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}, \tag{5.6}$$

$$\text{F-score} = 2\,\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{5.7}$$

where, n corresponds to the number of samples in the evaluated dataset (the test set), Y_i corresponds to the real labelset of the i-th sample and Z_i corresponds to the predicted labelset. The union ($\cup$) and the intersection ($\cap$) operators return a new set with the bit-wise OR and respectively the bit-wise AND of the elements of the two operand labelsets. Finally the $|\cdot|$ operator counts the number of active labels (number of 1s) of the given set.

5.3 Experimental analysis and discussion

The experimental setup described in sub-section 5.2.3 was used as a baseline for the various experiments conducted. As a first comparison, in Tables 5.5 and 5.6, we can observe that the use of data augmentation leads to a generous performance improvement with or without the use of regularization (i.e. dropout). Another important observation is that in the case where data augmentation is omitted, dropout can meaningfully improve the final outcome, given that initially the trained CNN overfits the small training dataset. On the other hand, as seen

Table 5.5: The performance and standard deviation (parentheses) of the proposed architecture, with different dropout options. Data augmentation has been utilized.

Dropout	**Accuracy**	**Precision**	**Recall**	**F-score**
No	81.2 (0.7)	87.3 (0.6)	88.5 (0.7)	87.9
0.25	81.3 (0.8)	88.0 (0.8)	89.3 (0.7)	88.6
0.50	**81.4 (0.4)**	**88.2 (0.5)**	**89.5 (0.6)**	**88.8**
0.75	79.7 (0.0)	86.4 (0.2)	89.1 (0.3)	87.7

Table 5.6: The performance and standard deviation (parentheses) of the proposed architecture, with different dropout options. Data augmentation has been omitted.

Dropout	**Accuracy**	**Precision**	**Recall**	**F-score**
No	68.0 (0.5)	80.4 (0.5)	77.6 (1.4)	79.0
0.25	73.7 (0.8)	85.1 (1.0)	81.1 (0.6)	83.1
0.50	75.7 (0.5)	85.4 (0.6)	83.3 (0.7)	84.3
0.75	**77.7 (0.2)**	**85.5 (0.1)**	**85.8 (0.4)**	**85.7**

in Table 5.5, the impact of dropout greatly diminishes as the augmentation of the training set leads to a stronger mitigation of the effects of overfitting.

In Figure 5.3, we explore the impact of the different sigmoid thresholds and how they are translated to the network's increased requirements for more confident predictions. The results demonstrate that low threshold values lead to over-optimistic (high recall - low precision) predictions while high threshold values result to conservative (low recall - high precision) predictions. Nevertheless, all thresholds seem to result in reasonable F-score evaluations, with values between 0.3 and 0.4 qualifying as the optimal selections.

In Figure 5.4, we perform a data-driven analysis on how the initial size of the given training set can affect the final performance of the trained CNN. In the case

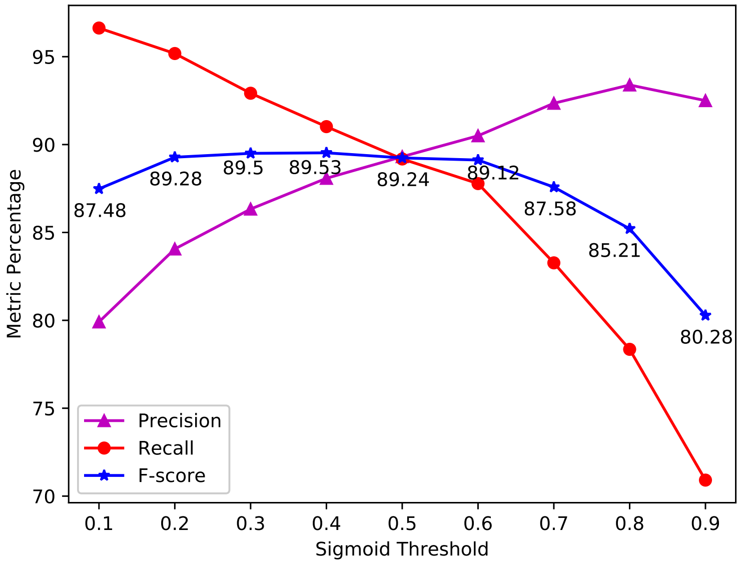


Figure 5.3: Plot of the precision, recall and F-score percentages for different sigmoid probability thresholds. The remaining hyperparameters stay unchanged. The extreme threshold values of 0 and 1 have been excluded.

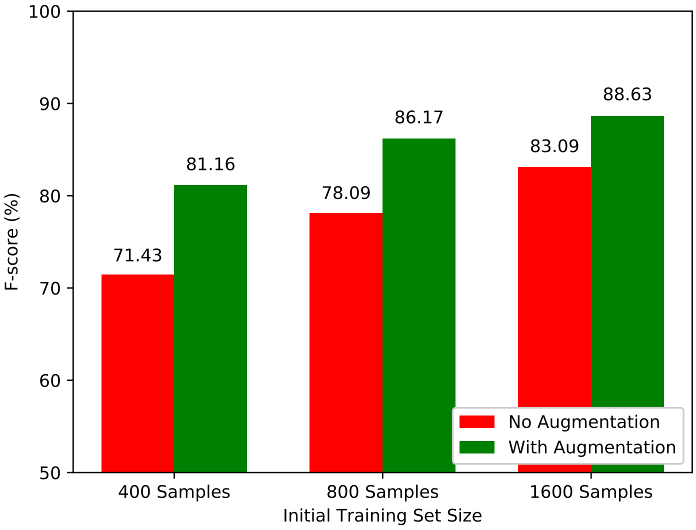
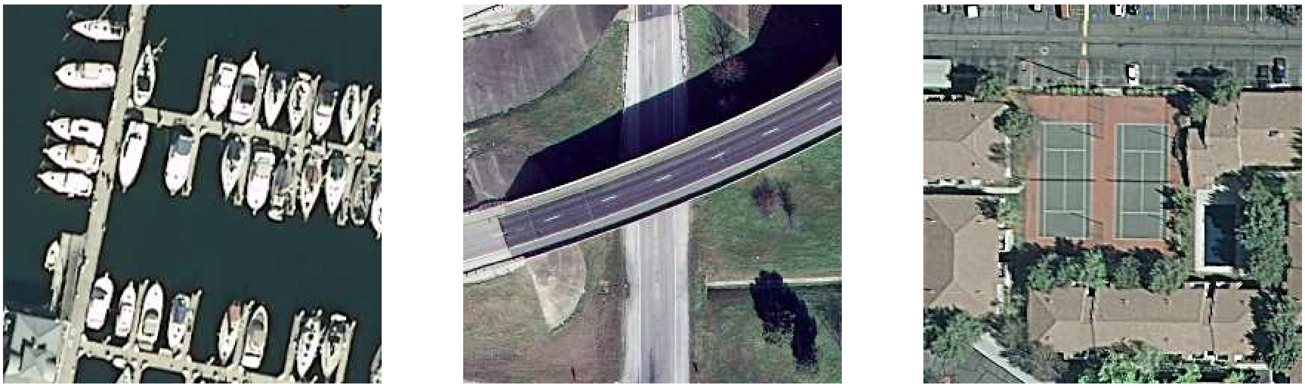


Figure 5.4: Demonstration of the impact of the initial size of the training set, with and without the utilization of data augmentation. In the case of data augmentation (green bars) the initial size is projected into much larger numbers. Given that in all experiments, the network was trained for 300 epochs, the final size of the training dataset is increased to 120,000, 240,000 and 480,000, indicated by the first, second and third green bar, respectively.

where data augmentation is employed, even though there is an obvious benefit with each increase, this benefit is not as pronounced as in the no-augmentation scenario. This result is inline with our intuition, since with data augmentation the transformation of the initial training examples leads to a fairly large dataset, regardless of its original limited size, whereas without data augmentation the initial size remains unaltered, rendering each increase far more impactful.

Figure 5.5, presents some indicative annotations inferred by the proposed model. We observe that the trained network manages to correctly predict the majority of the ground-truth labels of the tested images. Certainly, there are some cases where it fails to perceive the existence of certain objects. For example, in image (a) it misses the building in the lower left corner, presumably because it has inferred that buildings are usually found in groups and rarely in maritime environments. In other cases, and for equivalent reasons, it attributes specific labels to the image that in reality are false positives. For example, in image (b) the network is confident that it detects cars, given that in most freeway images in the dataset cars are present. Last, in image (c), we can observe that the network might fail to distinguish between different objects that might share some common attributes. For example, the green color of the existing courts seems to confuse the trained CNN, which falsely decides the existence of the grass label, instead of that of the court.



(a) **dock**, **ship**, **water**, buildings

(b) **bare soil**, **grass**, **pavement**, trees, *cars*

(c) **buildings**, **cars**, **pavement**, **trees**, court *grass*

Figure 5.5: Examples of inferred annotations where **bold** indicates correctly identified labels, *italics* denotes labels detected by the proposed method but not identified as active in the ground-truth, and underlined are ground-truth labels not found by the proposed method.

Table 5.7: Comparison between the multi-label image retrieval model (MLIR-CF) [28], the graph-based approach (GB) [27] and our proposed CNN with data augmentation (CNN DA).

Metric	**MLIR-CF** [28]	**GB** [27]	**CNN DA**
Accuracy	61.88	74.29	**82.29**
Precision	68.13	85.68	**88.08**
Recall	81.77	80.25	**91.02**
F-score	74.33	82.88	**89.53**

Finally, in Table 5.7 we present a comparison between our best trained model and the aforementioned works on the same topic. The increase of 6.65% on the F-score and 8% on the multi-label accuracy, compared to the current state-of-the-art, clearly show the capabilities of the proposed approach.

Chapter 6

Conclusions

In this thesis, we have addressed different classification problems in remote sensing, by utilizing the cutting-edge deep learning methodology of Convolutional Neural Networks. We have done so, in a case where the availability of data falls under the category of the Big Data paradigm, but also in a different scenario where data deficiency calls for alternative solutions.

Regarding the first application field, we proposed an alternative solution for the problem of spectroscopic redshift estimation in astronomy, through its transformation from a regression to a multi-class classification problem. We deployed a one-dimensional variation of a Convolutional Neural Network and we thoroughly examined its estimating capabilities for the issue at hand, using big volumes of training observations in various settings. Experimental results unveiled the great potential of this radically new approach in the field of spectroscopic redshift analysis and triggered the need for a deeper study, concerning Euclid and other spectroscopic surveys.

In the second case study, we demonstrated the benefits of using deep CNN architectures along with data augmentation to efficiently address the problem of multi-label, land cover scene categorization on a limited in size dataset. The performed experiments demonstrated the impressive capabilities of the proposed methodology that managed to outperform the current state-of-the-art by more than 6% F-score, in a multi-label modified version of the UC-Merced Land Use Dataset. Both case studies serve to further confirm the potential of deep learning for simultaneous feature extraction and classification.

6.1 Future directions

Future work in spectroscopic redshift analysis, concerning the Euclid space telescope, includes the introduction of new noise patterns that will complement the existing noise-scenario to an outright realistic simulation. Using these data, a robust predictive model can be built, capable of pioneering in the area of our study, and a form of transfer learning can be applied [9], exploiting future, real Euclid observations. Another avenue of applications involves other spectroscopic surveys. The Dark Energy Spectroscopic Instrument (DESI) [103] is one of the major upcoming cosmological surveys currently under construction and installation in Kitt Peak, Arizona. It will operate in different wavelengths and under different observational and instrumental conditions compared to Euclid, and consequently will be able to detect galaxies with different redshift properties.

In the case of multi-label classification in remote sensing, our focus can be concentrated on the design of alternative techniques for managing the problem of low data availability and on the application of the proposed methodology on hyperspectral imaging modalities.

Finally, given that all current deep learning libraries provide frameworks of CNN architectures that are limited up to the three-dimensional case, we can establish the development of a generalized scheme for n-dimensional Convolutional Neural Networks.

Bibliography

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[2] A. G. Ivakhnenko and V. G. Lapa. Cybernetic predicting devices. Technical report, Purdue Univ. Lafayette ind. school of Electrical Engineering, 1966.

[3] Andrew Beam. Deep learning 101 - part 1: History and background. `https://beamandrew.github.io/deeplearning/2017/02/23/deep_learning_101_part1.html`.

[4] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[5] H Keffer Hartline. The response of single optic nerve fibers of the vertebrate eye to illumination of the retina. *American Journal of Physiology–Legacy Content*, 121(2):400–415, 1938.

[6] Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279. ACM, 2010.

[7] Randal Bryant, Randy H Katz, and Edward D Lazowska. Big-data computing: creating revolutionary breakthroughs in commerce, science and society. A white paper prepared for the Computing Community Consortium committee of the Computing Research Association, 2008.

[8] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

[9] Lorien Y Pratt. Discriminability-based transfer between neural networks. In *Advances in neural information processing systems*, pages 204–211, 1993.

[10] Patrice Y Simard, Dave Steinkraus, and John C Platt. Best practices for convolutional neural networks applied to visual document analysis. page 958. IEEE, 2003.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[12] Gianfranco Bertone. *Particle dark matter: Observations, models and searches.* Cambridge University Press, 2010.

[13] E. J. Copeland, M. Sami, and S. Tsujikawa. Dynamics of Dark Energy. *International Journal of Modern Physics D*, 15:1753–1935, 2006.

[14] Geoffrey Marcy, R Paul Butler, Debra Fischer, Steven Vogt, Jason T Wright, Chris G Tinney, and Hugh RA Jones. Observed properties of exoplanets: masses, orbits, and metallicities. *Progress of Theoretical Physics Supplement*, 158:24–42, 2005.

[15] Planck Collaboration, P. A. R. Ade, N. Aghanim, M. Arnaud, M. Ashdown, J. Aumont, C. Baccigalupi, A. J. Banday, R. B. Barreiro, J. G. Bartlett, and et al. Planck 2015 results. XIII. Cosmological parameters. *A&A*, 594:A13, September 2016.

[16] W. J. Borucki, D. Koch, G. Basri, N. Batalha, T. Brown, D. Caldwell, J. Caldwell, J. Christensen-Dalsgaard, W. D. Cochran, E. DeVore, E. W. Dunham, A. K. Dupree, T. N. Gautier, J. C. Geary, R. Gilliland, A. Gould, S. B. Howell, J. M. Jenkins, Y. Kondo, D. W. Latham, G. W. Marcy, S. Meibom, H. Kjeldsen, J. J. Lissauer, D. G. Monet, D. Morrison, D. Sasselov, J. Tarter, A. Boss, D. Brownlee, T. Owen, D. Buzasi, D. Charbonneau, L. Doyle, J. Fortney, E. B. Ford, M. J. Holman, S. Seager, J. H. Steffen, W. F. Welsh, J. Rowe, H. Anderson, L. Buchhave, D. Ciardi, L. Walkowicz, W. Sherry, E. Horch, H. Isaacson, M. E. Everett, D. Fischer, G. Torres, J. A. Johnson, M. Endl, P. MacQueen, S. T. Bryson, J. Dotson, M. Haas, J. Kolodziejczak, J. Van Cleve, H. Chandrasekaran, J. D. Twicken, E. V. Quintana, B. D. Clarke, C. Allen, J. Li, H. Wu, P. Tenenbaum, E. Verner, F. Bruhweiler, J. Barnes, and A. Prsa. Kepler Planet-Detection Mission: Introduction and First Results. *Science*, 327:977, February 2010.

[17] R Laureijs, J Amiaux, S Arduini, J-L Augueres, J Brinchmann, R Cole, M Cropper, C Dabin, L Duvet, A Ealet, et al. Euclid definition study report. *arXiv preprint arXiv:1110.3193*, 2011.

[18] P. A. Abell, J. Allison, S. F. Anderson, J. R. Andrew, J. R. P. Angel, L. Armus, D. Arnett, S. J. Asztalos, T. S. Axelrod, S Bailey, et al. Lsst science book, version 2.0. 2009.

[19] Zachary D Stephens, Skylar Y Lee, Faraz Faghri, Roy H Campbell, Chengxiang Zhai, Miles J Efron, Ravishankar Iyer, Michael C Schatz, Saurabh Sinha, and Gene E Robinson. Big data: astronomical or genomical? *PLoS biology*, 13(7):e1002195, 2015.

[20] G Efstathiou, Wo J Sutherland, and SJ Maddox. The cosmological constant and cold dark matter. *Nature*, 348(6303):705–707, 1990.

[21] Richard Massey, Thomas Kitching, and Johan Richard. The dark matter of gravitational lensing. *Reports on Progress in Physics*, 73(8):086901, 2010.

[22] Jun Ding, Bo Chen, Hongwei Liu, and Mengyuan Huang. Convolutional neural network with data augmentation for sar target recognition. *IEEE Geoscience and remote sensing letters*, 13(3):364–368, 2016.

[23] Christopher D Lippitt, Douglas A Stow, and Lloyd L Coulter. *Time-sensitive remote sensing*. Springer, 2015.

[24] Qin Zou, Lihao Ni, Tong Zhang, and Qian Wang. Deep learning based feature selection for remote sensing scene classification. *IEEE Geosci. Remote Sensing Lett.*, 12(11):2321–2325, 2015.

[25] Grant J Scott, Matthew R England, William A Starms, Richard A Marcum, and Curt H Davis. Training deep convolutional neural networks for land–cover classification of high-resolution imagery. *IEEE Geoscience and Remote Sensing Letters*, 14(4):549–553, 2017.

[26] Weiwei Sun and Ruisheng Wang. Fully convolutional networks for semantic segmentation of very high resolution remotely sensed images combined with dsm. *IEEE Geoscience and Remote Sensing Letters*, 15(3):474–478, 2018.

[27] Bindita Chaudhuri, Begüm Demir, Subhasis Chaudhuri, and Lorenzo Bruzzone. Multilabel remote sensing image retrieval using a semisupervised graph-theoretic method. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2):1144–1158, 2018.

[28] Zhenfeng Shao, Ke Yang, and Weixun Zhou. Performance evaluation of single-label and multi-label remote sensing image retrieval using a dense labeling dataset. *Remote Sensing*, 10(6):964, 2018.

[29] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[30] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[31] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. In *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*, pages 411–415. World Scientific, 1987.

[32] Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130, 1988.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[36] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361, 2015.

[37] Grigorios Tsagkatakis, Mustafa Jaber, and Panagiotis Tsakalides. Goal!! event detection in sports video. *Electronic Imaging*, 2017(16):15–20, 2017.

[38] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[39] Konstantina Fotiadou, Grigorios Tsagkatakis, and Panagiotis Tsakalides. Deep convolutional neural networks for the classification of snapshot mosaic hyperspectral imagery. *Electronic Imaging*, 2017(17):185–190, 2017.

[40] Fan Hu, Gui-Song Xia, Jingwen Hu, and Liangpei Zhang. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, 7(11):14680–14707, 2015.

[41] Wei Hu, Yangyu Huang, Li Wei, Fan Zhang, and Hengchao Li. Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, 2015, 2015.

[42] Guang Xu, Xuan Zhu, Dongjie Fu, Jinwei Dong, and Xiangming Xiao. Automatic land cover classification of geo-tagged field photos by deep learning. *Environmental Modelling & Software*, 91:127–134, 2017.

[43] D Tuccillo, Etienne Decencìère, Santiago Velasco-Forero, et al. Deep learning for studies of galaxy morphology. *Proceedings of the International Astronomical Union*, 12(S325):191–196, 2016.

[44] D Tuccillo, E Decenciére, S Velasco-Forero, H Domínguez Sánchez, P Dimauro, et al. Deep learning for galaxy surface brightness profile fitting. *Monthly Notices of the Royal Astronomical Society*, 2017.

[45] AK Aniyan and Kshitij Thorat. Classifying radio galaxies with the convolutional neural network. *The Astrophysical Journal Supplement Series*, 230(2):20, 2017.

[46] Fabian Gieseke, Steven Bloemen, Cas van den Bogaard, Tom Heskes, Jonas Kindler, Richard A Scalzo, Valério ARM Ribeiro, Jan van Roestel, Paul J Groot, Fang Yuan, et al. Convolutional neural networks for transient candidate vetting in large-scale surveys. *Monthly Notices of the Royal Astronomical Society*, 472(3):3101–3114, 2017.

[47] Edward J Kim and Robert J Brunner. Star-galaxy classification using deep convolutional neural networks. *Monthly Notices of the Royal Astronomical Society*, page stw2672, 2016.

[48] CE Petrillo, C Tortora, S Chatterjee, G Vernardos, LVE Koopmans, G Verdoes Kleijn, NR Napolitano, G Covone, P Schneider, A Grado, et al. Finding strong gravitational lenses in the kilo degree survey with convolutional neural networks. *Monthly Notices of the Royal Astronomical Society*, 472(1):1129–1150, 2017.

[49] Francois Lanusse, Quanbin Ma, Nan Li, Thomas E Collett, Chun-Liang Li, Siamak Ravanbakhsh, Rachel Mandelbaum, and Barnabas Poczos. Cmu deeplens: Deep learning for automatic image-based galaxy-galaxy strong lens finding. *arXiv preprint arXiv:1703.02642*, 2017.

[50] Narciso Benitez. Bayesian photometric redshift estimation. *The Astrophysical Journal*, 536(2):571, 2000.

[51] Christopher Bonnett. Using neural networks to estimate redshift distributions. an application to cfhtlens. *Monthly Notices of the Royal Astronomical Society*, 449(1):1043–1056, 2015.

[52] Iftach Sadeh, Filipe B Abdalla, and Ofer Lahav. Annz2: Photometric redshift and probability distribution function estimation using machine learning. *Publications of the Astronomical Society of the Pacific*, 128(968):104502, 2016.

[53] David W Gerdes, Adam J Sypniewski, Timothy A McKay, Jiangang Hao, Matthew R Weis, Risa H Wechsler, and Michael T Busha. Arborz: photometric redshifts using boosted decision trees. *The Astrophysical Journal*, 715(2):823, 2010.

[54] Karl Glazebrook, Alison R. Offer, and Kathryn Deeley. Automatic redshift determination by use of principal component analysis i: Fundamentals. *The Astrophysical Journal*, 492:98–109, 1998.

[55] Daniel Guidici and Matthew L Clark. One-dimensional convolutional neural network land-cover classification of multi-seasonal hyperspectral imagery in the san francisco bay area, california. *Remote Sensing*, 9(6):629, 2017.

[56] Haoning Lin, Zhenwei Shi, and Zhengxia Zou. Maritime semantic labeling of optical remote sensing images with multi-scale fully convolutional network. *Remote Sensing*, 9(5):480, 2017.

[57] Yi Zhu and Shawn Newsam. Land use classification using convolutional neural networks applied to ground-level images. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 61. ACM, 2015.

[58] Jingxiang Yang, Yongqiang Zhao, Jonathan Cheung-Wai Chan, and Chen Yi. Hyperspectral image classification using two-channel deep convolutional neural network. In *Geoscience and Remote Sensing Symposium (IGARSS), 2016 IEEE International*, pages 5079–5082. IEEE, 2016.

[59] Konstantinos Karalas, Grigorios Tsagkatakis, Michalis Zervakis, and Panagiotis Tsakalides. Feature learning for multi-label land cover classification.

[60] Jie Geng, Jianchao Fan, Hongyu Wang, Xiaorui Ma, Baoming Li, and Fuliang Chen. High-resolution sar image classification via deep convolutional autoencoders. *IEEE Geoscience and Remote Sensing Letters*, 12(11):2351–2355, 2015.

[61] Nataliia Kussul, Mykola Lavreniuk, Sergii Skakun, and Andrii Shelestov. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782, 2017.

[62] Grigorios Tsagkatakis and Panagiotis Tsakalides. Deep feature learning for hyperspectral image classification and land cover estimation. *ESA Symbosium*, 2016.

[63] Dimitrios Marmanis, Mihai Datcu, Thomas Esch, and Uwe Stilla. Deep learning earth observation classification using imagenet pretrained networks. *IEEE Geoscience and Remote Sensing Letters*, 13(1):105–109, 2016.

[64] Marco Castelluccio, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. Land use classification in remote sensing images by convolutional neural networks. *arXiv preprint arXiv:1508.00092*, 2015.

[65] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[66] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[67] Steven W Running. Estimating terrestrial primary productivity by combining remote sensing and ecosystem simulation. In *Remote sensing of biosphere functioning*. Springer, 1990.

[68] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[69] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[70] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[71] Matthew R Boutell et al. Learning multi-label scene classification. *Pattern recognition*, 37(9), 2004.

[72] Zhi-Hua Zhou et al. Multi-instance multi-label learning with application to scene classification. In *Advances in neural information processing systems*, 2007.

[73] Ricardo S Cabral et al. Matrix completion for multi-label image classification. In *Advances in Neural Information Processing Systems*, 2011.

[74] Image adopted from:
Francisco Herrera, Francisco Charte, Antonio J Rivera, and María J Del Jesus. Multilabel classification. In *Multilabel Classification*, pages 17–31. Springer, 2016.

[75] Leo Breiman. Arcing the edge. Technical report, Technical Report 486, Statistics Department, University of California, 1997.

[76] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[77] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.

[78] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *BMVC*, page 6, 2015.

[79] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[80] Shu Huang, Wei Peng, Jingxuan Li, and Dongwon Lee. Sentiment and topic analysis on social media: a multi-task multi-label classification approach. In *Proceedings of the 5th annual ACM web science conference*, pages 172–181. ACM, 2013.

[81] Shuhua Monica Liu and Jiun-Hung Chen. A multi-label classification based approach for sentiment classification. *Expert Systems with Applications*, 42(3):1083–1093, 2015.

[82] Dolly Carrillo, Vivian F López, and María N Moreno. Multi-label classification for recommender systems. In *Trends in Practical Applications of Agents and Multiagent Systems*, pages 181–188. Springer, 2013.

[83] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD*, volume 18, 2008.

[84] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2009.

[85] Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab, 1961.

[86] Image adopted from:
Mathworks. Introduction to deep learning: What are convolutional neural networks? https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks–1489512765771.html.

[87] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012.

[88] Yann LeCun et al. Generalization and network design strategies. *Connectionism in perspective*, pages 143–155, 1989.

[89] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[90] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[91] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[92] Image adopted from:
CS231n Convolutional Neural Networks for Visual Recognition. Convolutional neural networks. http://cs231n.github.io/convolutional-networks/.

[93] S Jouvel, J-P Kneib, O Ilbert, G Bernstein, S Arnouts, T Dahlen, A Ealet, B Milliard, H Aussel, P Capak, et al. Designing future dark energy space missions-i. building realistic galaxy spectro-photometric catalogs and their first applications. *Astronomy & Astrophysics*, 504(2):359–371, 2009.

[94] Pm Capak, H Aussel, M Ajiki, HJ McCracken, B Mobasher, N Scoville, P Shopbell, Y Taniguchi, D Thompson, S Tribiano, et al. The first release cosmos optical and near-ir data and catalog. *The Astrophysical Journal Supplement Series*, 172(1):99, 2007.

[95] O Ilbert, P Capak, M Salvato, H Aussel, HJ McCracken, DB Sanders, N Scoville, J Kartaltepe, S Arnouts, E Le Floc'h, et al. Cosmos photometric redshifts with 30-bands for 2-deg2. *The Astrophysical Journal*, 690(2):1236, 2008.

[96] John Duchi et al. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12(Jul), 2011.

[97] Richard Bellman. *Dynamic programming.* Princeton University Press, 1957.

[98] DP Machado, A Leonard, J-L Starck, FB Abdalla, and S Jouvel. Darth fader: Using wavelets to obtain accurate redshifts of spectra at very low signal-to-noise. *Astronomy & Astrophysics*, 560:A83, 2013.

[99] Konstantina Fotiadou, Grigorios Tsagkatakis, Bruno Moraes, Filipe B Abdalla, and Panagiotis Tsakalides. Denoising galaxy spectra with coupled dictionary learning. In *Signal Processing Conference (EUSIPCO), 2017 25th European*, pages 498–502. IEEE, 2017.

[100] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[101] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[102] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[103] Michael Levi, Chris Bebek, Timothy Beers, Robert Blum, Robert Cahn, Daniel Eisenstein, Brenna Flaugher, Klaus Honscheid, Richard Kron, Ofer Lahav, et al. The desi experiment, a whitepaper for snowmass 2013. *arXiv preprint arXiv:1308.0847*, 2013.