Protecting forests from the sky: scene classification of the Amazon rainforest

Sam Lubbers

Bachelor of Science in Computer Science with Business
The University of Bath
May 2019

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

This dissertation may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Signed: Sam Lubbers

Abstract

Deforestation is one of the most pressing issues that our society currently faces, threatening the well-being of humans and other species alike. There is wide evidence suggesting that accurate and up-to-date information on various aspects of our forests can help us address deforestation more effectively. A promising approach to obtain such information is through the automatic interpretation of the ever increasing quantity and quality of aerial and satellite images of forests. In this project, I conduct extensive research of existing methods of automatic interpretation of remote sensing images and conclude that the best architecture consists of a powerful feature extraction module coupled with a random forest classifier. Subsequently, I develop such a system and test it on satellite images of the Amazon rainforest using various feature configurations. In my experiments, the use of spectral features resulted in the fastest and best performing system. In addition, I proved that the Binary Robust Invariant Scalable Keypoints (BRISK) detector together with the Fast Retina Keypoint (FREAK) descriptor can be used as a more computationally efficient local feature than the more popular Scale Invariant Feature Transform (SIFT).

Contents

1 Introduction

1.1 Problem statement

1.1.1 The deforestation tragedy

Throughout history we humans have been clearing forests—a practice known as deforestation. At first, we mostly removed forests for agricultural purposes: to raise livestock or grow crops. Over time, as we found practical uses for wood, we started logging trees for fuel, furniture or paper. Due to increasing population growth and industrialization, in the past two hundred years these activities have dramatically expanded, with deforestation also increasing as a result. Thus far, we humans have destroyed more than eighteen million km^2 of forests (Williams, 2006). This is an area larger than that currently covered by China, India, Mexico, Iran, Nigeria, and France *combined*. Every single year we still lose thirty-three thousand km^2 of forests (FAO, 2016), an area equivalent to the size of Belgium. The frightening magnitude of deforestation is causing some devastating unintended consequences.

Deforestation is one of the major contributors to climate change (Ripple et al., 2017). Changes in climate patterns estimated to have a detrimental impact on humans over the course of the next century. For example, larger and more frequent droughts and floods are projected to severely impact food production, consequently affecting the health of millions of people, with poor communities being especially vulnerable due to their low adaptive capacity (Solomon et al., 2007). Climate change is largely caused by a higher concentration of greenhouse gases in the atmosphere, one of the most notably being carbon dioxide (Cook et al., 2016). Trees sequester carbon dioxide, thereby reducing the concentration in the atmosphere. When forests are cleared out, it does not only reduce the capacity of that area of land to sequester carbon dioxide but the sequestered carbon is released back into the atmosphere. The astounding scale at which we are clearing forests worldwide makes deforestation the second leading cause of climate change after fossil fuel combustion, accounting for nearly twenty percent of all greenhouse gas emissions (FAO, 2018).

But it is not just about the carbon in the trees. Deforestation is causing a dramatic loss of biodiversity (Barlow et al., 2016). Tropical rainforests support an estimated fifty percent of all living organisms that have evolved over the past thousands of years. This diverse range of animals, plants, and microorganisms support ecosystem services that are extremely valuable to us humans. Additional negative consequences of deforestation include less availability of fresh water, a rise in soil erosion and an increase in infectious diseases such as Malaria (Patz et al., 2000).

1.1.2 The need for improved forest monitoring

There is no single solution to deforestation. Instead, tackling deforestation involves a combination of sustainable forest management practices, effective policy, useful investments and informative research. All of these solutions are underpinned by reliable and up-to-date information on the extent of forest resources, their condition, management and uses (FAO, 2016). The primary source of this information is the Global Forest Resource Assessment (FRA) report produced by the Food and Agriculture Organization of the United Nations (FAO). The statistics provided in this report, defined as "sustainability indicators for forests", have influenced decision making by many international bodies (Keenan et al., 2015).

There are, however, limitations to the data published in the Forest Resource Assessment. The report is based on responses to surveys by individual governments. Because many developing countries do not have comprehensive up-to-date information on their forests, these surveys are often answered using subjective expert assessments. Furthermore, the FRA report is published only once every five years. If a country responds to its survey in 2011 and the next report is not published until 2015, the FAO will need to estimate what that countries forest data will be like by the year 2015, which invariably leads to projection errors. These issues has raised concerns about the accuracy of the data, particularly regarding changes in forest area (Grainger, 2008).

There is an urgent need for more reliable and up-to-date information on forests sustainability indicators, especially in developing countries (Baker et al., 2010). Due to increasing concerns about the impacts of the hideous state of our forests, the UN Framework Convention on Climate Change (UNFCCC) has created a mechanism for Reducing Emissions from Deforestation and Degradation (REDD+) with the aim of improving sustainable management of forests and promote the conservation and enhancement of forest carbon stocks in developing countries (UNFCCC, 2014). Many developed countries have greatly profited from consuming its forest resources, and developing countries feel entitled to do the same. The REDD+ initiative aims to avoid this disastrous scenario by having developed countries financially compensate developing countries to compensate for the emission reductions that come from protecting and properly managing their forests. Successfully implementing such a mechanism will require improved technical capabilities for more accurate and frequent forest monitoring. The importance of superior forest monitoring technologies cannot be overstated, since the aforementioned financial support could extend to *billions* of US dollars (Grainger and Obersteiner, 2011).

1.1.3 The potential of aerial imagery

The immense area of forests in developing countries makes ground monitoring too difficult and costly. Currently, the best method for such forest monitoring is the analysis of aerial and satellite images (Grainger and Obersteiner, 2011). The upcoming constellation of satellites will capture ever more forest imagery at higher spatial, spectral and temporal resolutions (Romero et al., 2016). This generates great promise and demand for more accurate and efficient automated methods of interpreting remote sensing images (Cheng et al., 2017).

1.2 Research objectives

In this project, I strive to fulfil two main objectives. The first one is to review, analyse and synthesise the existing research in automatic remote sensing image interpretation. The second one is to use these findings to develop an accurate and efficient system that could be used for effective forest monitoring. More specifically, I will develop a system that performs classification of scene images of the Amazon rainforest. A scene image is a small image patch extracted from a large-scale remote sensing image. Classifying these images consists of assigning each image with one or more semantic label that describes the form of land cover or land use present in the image, such as forest or agriculture (Cheng et al., 2017).

2 Literature and Technology survey

The manual analysis of aerial photographs for management of natural resources has been performed since the 1940s, with computerised methods coming into play in the 1970s (Kim et al., 2009). Today, automatic scene classification is an active research topic in the field of aerial image analysis (Cheng et al., 2017). In this literature and technology survey I uncover what particular architecture makes for the most accurate and efficient scene classification system.

Most scene classification systems follow a common architecture structured into two modules. The first module takes an image as input and outputs a set of features which represent informative and distinctive patterns about the image. In section 2.1 examine which features are best suited for scene classification. Once features are extracted from the image they serve as input to the second module in the system—the classifier—which outputs the given scene observed in the image, such as forest or agriculture. In section 2.2 I explain how the field of machine learning has come to dominate scene classification and which classifier produces the best results on aerial images. Lastly, in section 2.3 I analyse the potential of deep learning, an extension of the machine learning paradigm which does not require feature extraction.

2.1 Features

2.1.1 The importance of features

Most pictures are encoded as raster images, consisting of a large array of pixels, each of which contains values for the blue, green and red image channels. Thus, to a computer, pictures are just large multidimensional matrices with thousands or even millions of numbers. This creates many problems for the task of scene classification (Romero et al., 2016). Firstly, processing all these thousands of numbers for analysing every single image requires a tremendous amount of computing resources. The second and more fundamental problem is that it is incredibly hard for a classifier to identify the contents of the image based off individual pixels, as these pixels are a very high dimensional input with a large amount of noise and redundancy. Thirdly, even if it is possible to create such a program there is a risk of overfitting to the patterns of individual images on which the system is tested and generalise poorly to other images.

Feature extraction solves these problems by filtering out the signal from the noise. It reduces the large three dimensional matrix into a few features that are easier to analyse and faster to process (Lecun et al., 1998). For instance, con-

sider a system that classifies aerial images as either forest or pasture. Instead of analysing the thousands of pixels in each image, low level image processing could be used to extract one informative feature about the image: the colour. Subsequently, we could use this feature to classify our image: if the colour of the image is green then it corresponds to a forest, otherwise it is classified as pasture. It is easy to see how classifying our image using this single variable is much simpler and requires vastly less computations than by searching properties of individual pixels. It is also possible to observe from this example how feature extraction reduces overfitting. Our system will correctly classify any forest image as long as it is predominantly green, no matter what the value or arrangement of individual pixels is.

Choosing the right set of features is paramount for the correct identification of image scenes, since the classification is usually carried out using these features as input (Cheng et al., 2017). However, there are no universally best features. Instead, The right features must be chosen for the particular images and application at hand (Tuytelaars and Mikolajczyk, 2007). The features pertinent to scene classification can be grouped into two main categories: global features, which includes spectral, colour and texture features, and local features (Negrel et al., 2014). In the following subsections I will explain how each of these types of features are useful, what the most important features are within each category, and how each of them have been successfully employed in related remote sensing applications. Technical descriptions of such features will be reserved to those used in this project and can be found in section 4.1.

2.1.2 Spectral features

Our sense of sight is made possible by the ability of our eyes to perceive a certain kind of electromagnetic radiation—what we call light. Remotely sensed images—images of the Earth captured by satellite or aircraft sensors—record the electromagnetic radiation of a given scene at various spectral bands. Whereas traditional images are limited to red, green and blue spectral bands of the spectrum visible to us humans, remote sensing has evolved to capture more spectral bands, extending into the infrared region of the electromagnetic spectrum (Shaw and Burke, 2003). Because different objects in a scene reflects, scatters, absorbs and emits electromagnetic radiation in a particular way (Shaw and Burke, 2003), remote sensing images of different areas will encode different values in each of the spectral bands. Spectral features are summary statistics of these spectral bands, and have proven to be among the most discriminative features for land cover classification (Dronova, 2015).

Some of the most common spectral features are the mean and standard deviation of each spectral band in the image. A spectral feature of particular importance to this project is the Normalised Difference Vegetation Index

(NDVI), a measure computed using the red and near-infrared bands. The NDVI indicates the level of photosynthetic activity, with higher NDVI values corresponding to denser vegetation being observed in a given image (Tucker, 1979). NDVI is therefore a useful feature to distinguish between areas that are still covered by rainforests and areas where rainforests have been cleared out for human use.

A large number of researchers in this field employ the land cover mapping software eCognition (Blaschke, 2010). This software performs land cover classification utilising spectral features among others, making them widespread among a wide variety of different studies (Ma, Fu, Blaschke, Li, Tiede, Zhou, Ma and Chen, 2017; Csillik, 2017; Dronova, 2015).

2.1.3 Colour features

Colour features, as the name implies, describes the colours present in the image. They are highly relevant for scene classification because colour is often highly correlated with elements of the natural environment (Swain and Ballard, 1991). Colour features are translation and rotation invariant (Swain and Ballard, 1991), a desirable property for scene classification as remote sensing images are taken in varying orientations (Song et al., 2010). Colour features and spectral features are not commonly used in conjunction because they measure similar information in the image. Colour features could even be categorised as spectral, since they summarise information contained in the blue, green and red spectral bands.

The most common colour feature in scene classification is the colour histogram (Cheng et al., 2017), first proposed as a feature by Swain and Ballard (1991). Colour histograms stand out because of its simplicity to implement and efficiency to compute (Cheng et al., 2017). Yang and Newsam (2010) achieved 81.19% on the popular UC Merced land cover classification dataset using solely a colour histogram as a feature. Santos et al. (2010) evaluated a diverse set of colour features for the task of classifying images of pasture and coffee crops. In their experiments the colour histogram only achieved an average accuracy of 60%, and was outperformed by other colour features such as BIC (Stehling et al., 2002), ACC (Kumar et al., 1997), and JAC (Williams and Yoon, 2007) with 65%, 70%, and 80% average accuracy. Keep in mind, however, that these results do not just boil down to the choice of feature and are also largely influenced by the quality and size of the dataset, choice of classifier and even by how the evaluation metric has been computed, among many other factors (Ma, Li, Ma, Cheng, Du and Liu, 2017).

2.1.4 Texture features

Spectral and colour features provide aggregate information about the values in the spectral bands of the image, however, they do not record how those values are spatially distributed. This is where texture features come in (Haralick et al., 1973). Texture can be the rugged pattern formed by dense rainforests or the smooth pattern of agricultural lands. Even though these forms of land cover might exhibit similar spectral or colour features they have very different textures. Therefore, texture features play an important role in the classification and interpretation of remotely sensed data (He and Wang, 1990). The most common texture feature used in scene classification—as well as many other domains—is the Gray Level Co-ocurrance Matrix (Kim et al., 2009), a simple and efficient texture feature proposed by Haralick et al. (1973). Marceau et al. (1990) proved that classification of SPOT satellite multispectral images was significantly improved when including the GLCM texture feature, instead of just using spectral features. These results were later replicated by Kim et al. (2009) for very high resolution multispectral IKONOS satellite images, increasing classification accuracy from 79% to 83% when GLCM was employed on top of spectral features. Almost 50 years after its conception GLCM is still relevant, since it is included in widely used eCognition land cover mapping software (Ma, Fu, Blaschke, Li, Tiede, Zhou, Ma and Chen, 2017).

Another prominent texture feature is the Local Binary Patterns (LBP), proposed and later extended by Ojala et al. (1994, 2002). Some powerful properties of LBP are its invariance to illumination and rotation, while still being extremely efficient to compute, properties often lacking in other texture features (Ojala et al., 2002). In a study on land cover classification of IKONOS remote sensing images, LBP + spectral features obtained 83% accuracy, outperforming not only spectral features (71% accuracy), but also GLCM + spectral features (79% accuracy).

2.1.5 Local features

The features covered so far are all global features, which describe properties the image as a whole. Global features can work well in cases when a single land cover type spans the entire image. However, because global features aggregate the information of the entire image it performs poorly if there are multiple forms of land cover in the image or if we are interested in identifying more small scale phenomena like logged trees or a fracking site. In such circumstances we must resort to local features (Tuytelaars and Mikolajczyk, 2007). Local features are usually obtained by first detecting a set of keypoints—a pattern of pixels in an image which differs from its surrounding pixels—and

subsequently creating a feature descriptor for each of the identified keypoints. A very important property of local features used in scene classification is invariance to translation, rotation and scale (Yang and Newsam, 2010). This invariance criterion reduces the vast catalogue of local features to just a few promising ones.

The first scale and rotation invariant local feature developed was "Scale Invariant Feature Transform" (SIFT) (Lowe, 1999, 2004). SIFT is the most widely used local feature in the area of land cover classification (Cheng et al., 2017). Nevertheless, there are superior alternatives. Because of the vast amounts of images processed in scene classification, its is crucial for local features to be efficient to compute (Tuytelaars and Mikolajczyk, 2007). SIFT is notably slow to compute and therefore not an ideal candidate as a local feature. Bay et al. (2006) proposed "Speeded-Up Robust Features" (SURF), a variant to SIFT that makes use of approximations for faster computations and performs just as well. The downside of SURF is that, just as SIFT, it is patented and therefore cannot be widely adopted. A promising alternative local feature is the "Binary Robust Invariant Scalable Keypoints" (BRISK) (Leutenegger et al., 2011). BRISK local features achieve a comparable performance to SIFT and SURF while being dramatically more efficient to compute—more than 160 times faster than SIFT and more than 10 times faster than SURF (Leutenegger et al., 2011). The modularity of BRISK allows for the combination of its feature detector with an alternative feature descriptor and vice versa, making it possible to optimise the acquisition of local features even further for our particular application (Leutenegger et al., 2011). For this project it means that we can replace the BRISK descriptor with the "Fast Retina Keypoint" (FREAK) descriptor, which is just as robust as BRISK but is computed in almost half the time (Alahi et al., 2012).

Because a large number of local features are obtained from each image, an intermediate process must aggregate the information of all the local features of an image into a global descriptor that can be subsequently used for classification (Negrel et al., 2014). One approach to achieve this is with the Bag of Visual Words model (BoVW) (Sivic and Zisserman, 2003). The BoVW will be extensively explained in section 4.1.4, but for now it is sufficient to know that the BoVW is a histogram with frequencies of different local features occurring in an image. Yang and Newsam (2010) tested the BoVW model on the UC Merced land cover dataset, a popular dataset used in scene classification consisting 2100 aerial RGB images of 21 different scenes. Using a BoVW of SIFT features they achieved a 76.8% accuracy, a lower performance than with a colour histogram on the same dataset (81.19%). While the BoVW includes information on the frequency of local features, it completely disregards how those features are spatially distributed in the image. Yang and Newsam (2010) addressed this by developing spatial extensions to the BoVW, such as the spa-

tial co-ocurrance kernel (SCK), achieving a slight improvement in accuracy to 77.71%. There are other superior extensions to the BoVW which achieve much greater performance. These include Fisher Vectors (Perronnin et al., 2010), Vectors of Locally Aggregated Descriptors (VLAD) (Jgou et al., 2012) and Vectors of Locally Aggregated Tensors (VLAT) (Picard and Gosselin, 2013). Negrel et al. (2014) tested these three methods with a colour histogram and HOG local features on the UC Merced dataset, achieving a remarkable accuracy of 92.5% with VLAD, 93.8% with Fisher Vectors and 94.3% with VLAT. Something revealing from these results is that they were achieved using HOG, a local feature not invariant to scale and rotation. It might be that these invariant properties are not as crucial as first assumed. Kobayashi (2014) tried improving the Fisher Vector even more by extending it with Dirichlet Process Gaussian Mixture Model (DP-GMM), a non-parametric Bayesian method used in other computer vision applications by prominent researchers such as Haines and Xiang (2014). When used on top of SIFT features this configuration achieved a 92.8% accuracy on the UC Merced dataset, lower than that achieved by Negrel et al. (2014) using simple Fisher Vectors. To complete the analysis on local features, there are researchers such as Cheng et al. (2015) that argue against the use of local features because of their limited descriptive capabilities. They advocate instead for discriminative image features extracted using a set of pretrained classifiers they term "sparselets". Despite managing to achieve a respectable accuracy of 91.46% on the UC Merced dataset, they do not outperform some of the previously described local feature encoding methods.

2.1.6 Feature fusion

After covering all types of features and their respective performances it might be tempting to conclude that local features are the "best" features and we can disregard all others. But all these features are not mutually exclusive. In fact, they are complementary. In remote sensing images information is conveyed by multiple properties such as spectral response, colour, texture or local distinctive patches, and each of these properties is captured by a single type of feature. So instead of trying to answer the question "what is the best feature?" we should instead be answering the question "what is the best way to combine complimentary features?" (Cheng et al., 2017).

Answers to this question have already been proposed by various researchers. Zhong et al. (2015) suggest that the common approach of concatenating features into a single feature vector is inadequate, and instead, a multifeature fusion strategy must be adopted. They propose a semantic allocation level fusion strategy based on probabilistic topic model to combine local and global features. To test their approach they first performed classification of the

UC Merced dataset using mean and standard deviation spectral features, the GLCM texture feature and SIFT local features encoded as a BoVW. When used individually, these features achieved a respective accuracy of 77%, 75% and 73%. When combining these simple features using their proposed strategy the accuracy increased to an outstanding 88.33%. An even greater performance has been accomplished by Shao et al. (2013), who obtained a 92.38% accuracy on the UC Merced dataset by combining colour, texture, shape and local features using a hierarchical feature fusion approach.

2.2 Machine learning classifiers

Extracting informative features from aerial images constitutes just one module of a scene classification system. The other module is the classifier, which uses these features to determine which forms of land cover are present in the image. The majority of classifiers developed over the past decade are founded on a machine learning technique named supervised learning (Ma, Li, Ma, Cheng, Du and Liu, 2017). Before proceeding with the explanation of such systems it is worth going back in time to see what types of systems were developed before machine learning classifiers came along. Exploring the limitations of such systems can help us better understand why machine learning systems work the way they do and why they are superior. After the explanation of these two systems I will uncover the best classifier for scene classification.

2.2.1 The limitations of knowledge based systems

One of the first notable examples of automatic aerial image interpretation is SCHEMA, a system developed with the purpose of identifying houses and roads from remote sensing images of urban environments (Matsuyama, 1987). SCHEMA's classifier is a particular kind of knowledge based system. Knowledge based systems integrate the relevant knowledge of an expert—in this case someone specialised in aerial image interpretation—in the hope that the system performs as well as the expert does. The expert knowledge is encoded into the system with a set of rules. Let's reconsider the previous example where images are classified as either forest or pasture based solely on the value of the dominant colour in the image. To classify such images, expert systems would contain a rule stating that if the dominant colour is green the image is classified as forest, else it is classified as pasture.

This simple example makes it seem like creating a knowledge based system to interpret images is not all that difficult. But what if there is some pasture that is predominantly green? Or what if due to fog the dominant colour of a forest image is white? In all such cases the system would fail because

these examples do not conform to the knowledge that is integrated into the system. The difficulty in formalising such intraclass variability is precisely what Draper et al. (1996) identified as one of the main obstacles to developing successful knowledge based systems. This obstacle was partially overcome by the integration of fuzzy logic into knowledge based systems, allowing for probability estimations of a given image containing a certain scene. With this improvement knowledge based systems dealt better with uncertain or incomplete information, and managed to achieve a good 80% accuracy in scene classification of very high resolution multispectral IKONOS satellite images (Hofmann et al., 2011).

Unfortunately, the incorporation of fuzzy logic does not solve all of the problems inherent in knowledge based systems. Contrary to the previous example where only a single feature was evaluated, real world applications use multiple features, sometimes even combining data acquired from multiple different sensor types. Interpreting all this complex data requires very highly specialised expert knowledge, which is often lacking (Fassnacht et al., 2014). If we assume that someone does posses this knowledge, transferring it onto a knowledge based system can still be an almost insurmountable task—it took Draper et al. (1996) one and a half man months of work to design the knowledge base for two dimensional road interpretation. Finally, after the painstakingly process of defining complex sets of rules, we are left with a knowledge base that is strongly bound to a particular kind of image. Consequently, the resulting system would fail if used for a different application or even if the data of the intended application slightly changes (Wieland and Pittore, 2014). In a survey on knowledge based computer vision systems Crevier and Lepage (1997) correctly predicted that "the true power of image understanding systems will not unfold until the advent of automated learning techniques". At the time, they considered such techniques "challenging lines of research", but as we will see, research has come a long way since.

2.2.2 Superiority of machine learning classifiers

Great advancements in the field of machine learning enabled researchers to replace knowledge based systems with superior classifiers that make use of a technique called supervised learning. With supervised learning, people do not have to declare a multitude of complex rules that the system must follow to interpret an image. Instead, these classifiers learn the relationship between the input (features) and outputs (image scene) from a set of labelled images termed training data. This classifier is then able to identify the appropriate scenes based on the features present in that image (Alpaydin, 2016).

These machine learning classifiers trump knowledge based systems in many ways (Alpaydin, 2016). First of all, they have the potential to perform bet-

ter by learning relevant correlations between features and image contents, as opposed to knowledge based systems which merely employ knowledge that humans think is useful. If the classifier is powerful enough it could learn such correlations from large and complex datasets from multiple sources and deal better with the problem of intraclass variability. Secondly, instead of spending weeks defining all the rules that your knowledge based system must follow to interpret your particular set of images, an existing classifier can be used directly as long as you have sufficient high quality training data. Finally, such a classifier is much easier to maintain because if the data changes, it can adapt by simply retraining on the new data.

In a timespan of less than a decade, researchers in automatic aerial image interpretation have shifted from programmed systems to learning systems. Prior to 2010, most studies in land cover classification used knowledge based systems. Between 2010 and 2014 the number of published studies using knowledge based systems kept on increasing, but this time paralleled with an increase use of supervised classifiers. From 2015 onward, the quantity of publications with knowledge based systems began to decline and those with machine learning classifiers increased even more rapidly (Ma, Li, Ma, Cheng, Du and Liu, 2017). This trend serves as evidence that machine learning classifiers—subsequently just referred to as classifiers—are the most powerful method for scene classification. But our search does not stop here as there are numerous classifiers, and choosing the right one can make a very big difference to our application.

2.2.3 Choosing the best classifier

In an attempt to uncover the main factors potentially affecting the quality of forest biomass estimations from remote sensing data, Fassnacht et al. (2014) performed a systematic literature review to generate hypotheses and then tested their hypotheses on different types of remote sensing data. Their findings revealed that the machine learning method used was the second most substantial determinant influencing accuracy, preceded only by the type of sensor used to acquired the data. An extensive meta-analysis by Ma, Li, Ma, Cheng, Du and Liu (2017) also indicates that the choice of classifier can substantially influence the performance of land cover classification.

Individuals studies have contradictory findings in regard to what the best classifier is for the task of land cover classification. Li et al. (2016) attribute this phenomenon to the fact that the vast majority of these studies compare methods in relation to a single factor, whereas in practice classification results are affected by many factors such as the quantity and type of features, the data or differences in experimental design. This motivated Li et al. (2016) to systematically analyse the land cover classification performance of a wide array of popular classifiers under various different configurations. Their multi-

ple experiments reveal that the non-parametric random forests, Adaboost and Support Vector Machines (SVM) classifiers perform best. While more extensive than most similar studies, it is worth knowing that this study also has its limitations, since they only evaluate the different models on high resolution images of the agricultural city of Deyang in China, and these images only include three RGB visible bands. Their findings might perhaps not generalise to areas with different terrains or to remote sensing images with more spectral bands, making it necessary to consider other studies as well. Fassnacht et al. (2014) tested various models for forest biomass estimations using LiDAR and hyperspectral data from case study areas in Germany in Chile, concluding that in most cases random forest performed best. Moreover, a review of 220 studies of land cover classification by Ma, Li, Ma, Cheng, Du and Liu (2017) show that Random Forest achieves the highest classification accuracy (85.81%), closely followed by SVM (85.19%).

A large number of studies demonstrate that complex non-parametric models are able to outperform simpler parametric classifiers because they are able to better model the complexities of aerial imagery and are better suited to deal with a large number of features (Maxwell et al., 2018). However, using more complex models also introduces an increased risk of overfitting, in which the model fits to noise or trivial patterns in the training data and as a result fails to correctly classify data it has not seen before (Hawkins, 2004). For this reason, models like random forest or SVM have built-in mechanisms to deal with overfitting (Fassnacht et al., 2014). In a study comparing classifiers on different types of aerial images of urban areas, Wieland and Pittore (2014) identified great transferability of random forest and SVM models when trained on one type of data and tested on another type—a property not shared among the other tested classifiers, which performed much worse for the same tests. Another important problem to mitigate is the Hughes phenomenon or curse of dimensionality, which occurs when the number of features grows too large in relation to the size of the dataset, leading to deficient predictions Belgiu and Drăguţ (2016). When evaluating feature selection methods for land cover classification, Ma, Fu, Blaschke, Li, Tiede, Zhou, Ma and Chen (2017) found out that random forests deal best with the Hughes phenomenon, being relatively insensitive to the number features even for a small training set. In the case of other classifiers such as SVM the performance decreased with a larger number of features. Moreover, random forests have proven to outperform other models for medium-sized and large datasets in land cover classification (Li et al., 2016).

Random forests also stand out in other areas beyond classification accuracy. Random forests are really efficient, requiring less training time than other highly performing classifiers such as SVM or Adaboost (Belgiu and Drăguţ, 2016). Moreover, random forests is easier to implement than SVM as it does

not require the configuration of multiple critical parameters and it can be used directly for multi-label classification. Because of its high performance, efficiency and other desirable properties, over the past two decades the random forest classifier has been increasingly used to analyse various forms of remote sensing data—including multispectral, radar, LiDAR and thermal imagery—resulting in successful applications such as biomass mapping, land cover classification or mapping of tree canopy cover (Belgiu and Drăguţ, 2016).

2.3 Deep learning architectures

2.3.1 Feature engineering vs Feature learning

No matter how powerful a given classifier may be, it would struggle to learn and make accurate predictions if there is no correlation between the image scene and the features extracted from the image. Thus, the performance of the classifier is highly dependent on extracting good features (Domingos, 2012). As we have seen in section 2.1, a considerable amount of research is required to chose among the vast quantity and variety of features. Implementing these features poses another challenge, as the best performing set of features turn out to be the most difficult ones to implement—requiring either sophisticated feature fusion strategies of simple features, or complex encoding methods of local features. Moreover, just as the performance of knowledge-based systems is constrained by our ability to devise complex set of classification rules, the ultimate performance of machine learning systems is limited by our capacity to engineer features Lecun et al. (1998). What's more, the development of such features can be a daunting and time-consuming process, especially when comparing it with the use of a classifier, which is often general purpose and easily trainable (Lecun et al., 1998).

Historically, feature extraction was needed because classifiers were unable to learn from extremely high dimensional inputs such as pixels in an image and because operating on such large inputs was computationally unfeasible, as emphasised already in section 2.1.1. However, the confluence of improved learning techniques that can identify intricate discriminative information in high dimensional data, more potent hardware that allow for brute-force methods and larger labelled datasets which can be leveraged for training, have made it possible to replace laborious feature engineering with automatic feature learning (Lecun et al., 1998). However, simply adjusting the traditional machine learning architecture by replacing the engineered feature extraction module with a feature learning module is certainly a sub-optimal solution, as it would result in two separately trained modules, which when assembled together would still require manual parameter optimisation to improve the

overall performance (Lecun et al., 1998). Instead, feature learning performs best when integrated as part of a unified supervised learning system that can operate directly on pixels. Such a system can be trained as a whole, automatically optimising all of its internal parameters with the aim of minimising the discrepancy between its predictions and the labels of the training data (Lecun et al., 1998). The most notable unified supervised learning architectures are deep learning architectures. Whereas deep learning is a subset of the broader range of machine learning methods, in this project machine learning is just used to refer to those systems that classify features instead of pixels.

2.3.2 Convolutional Neural Networks

The deep learning architecture most suited for scene classification is Convolutional Neural Network (CNN) since it is purposefully designed for interpreting images. In contrast with traditional deep learning architectures, CNNs are faster to train, require less training data, and take into account the local spatial distribution of pixels (Lecun et al., 1998). The CNN architecture was first proposed by Fukushima (1980) and later improved by Lecun et al. (1998), but CNNs were not widely popularised until Krizhevsky et al. (2012) designed a particular CNN architecture that outperformed all other existing image classification systems on the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

In ILSVRC the performance of image classification systems is compared using a benchmark labelled dataset containing more than one million images of a thousand different object categories—which is only a subset of the entire ImageNet dataset containing 15 million labelled images of 22,000 categories (Russakovsky et al., 2014). A common metric to compare the performance of image classifiers is the top-5 error rate, which measures how frequently none of the top 5 predictions made by the classifier correspond to the object in the image. In ILSVRC-2012 the architecture proposed by Krizhevsky et al. (2012)—commonly known as AlexNet—achieved a 15.3% top-5 error rate, vastly outperforming the previous best result of 26.2%, which was obtained using SIFT features encoded as Fisher Vectors. Over subsequent years, CNNs have become ever more powerful and achieved ever greater results on ILSVRC. AlexNet was first outperformed by Overfeat (Sermanet et al., 2013), an architecture that obtained a 13.24% top-5 error rate in ILSVRC-2013. In 2014, the best generic image classification performance was again dramatically surpassed, this time by two different CNN architectures, VGG (Simonyan and Zisserman, 2014) and GoogleNet (Szegedy et al., 2015), which achieved a respective 7.32% and 6.67% top-5 error rate.

Because CNNs have vastly outperformed traditional machine learning systems in ILSVRC, a lot of research in computer vision has shifted away from the

latter and towards the former. However, as I will expose in the next section, in the particular case of scene classification the performance of some of the most powerful CNNs is still rivalled by traditional systems consisting of a feature extractor and a simple classifier. Plus, deep learning approaches face some very important challenges that limits its current adoption into scene classification.

2.3.3 Deep learning for scene classification

Risk of overfitting

Most of the best performing CNNs are designed for the task of generic image classification. Such CNNs must correctly discern hundreds of different objects at various positions, illuminations and scales, in millions of different images. Learning the complex relationship between image pixel values and each of these hundreds of objects requires CNNs to be incredibly large. Where AlexNet—the first widely succesful CNN—really stood out over previous architectures was its remarkable 60 million parameters to optimise (Krizhevsky et al., 2012). Some of the subsequent architectures that outperformed AlexNet had even more parameters, with VGG optimisng 138 million (Simonyan and Zisserman, 2014) and Overfeat optimising 145 million (Sermanet et al., 2013).

Scene classification is, however, a much simpler computer vision task than generic object recognition. Firstly, whereas the generic ImageNet dataset contains 22,000 different objects, most scene classification datasets do not contain more than a few dozen land cover types. Secondly, images of land covers have dramatically lower intra-class variability than generic objects. In simpler words, most images of forests look more similar than most images of dogs. All in all, large scale CNNs designed for generic object classification might be too complex for scene classification, leading to overfitting (Romero et al., 2016). When training and testing AlexNet on the UC Merced land cover dataset it only achieves an 87% accuracy. GoogleNet, an architecture with significantly lower parameters than AlexNet and therefore less prone to overfit, achieves an improved accuracy of 92% (Nogueira et al., 2017). The remarkable finding is that both of these "state-of-the-art" Convolutional Neural Networks are outperformed by classifiers combined with hand-crafted feature extractors covered in section 2.1, such as the hierarchical feature fusion of simple features devised by Shao et al. (2013) which achieved 92.38% on the UC Merced dataset, or local feature encoding methods of VLAD, VLAT, and Fisher Vectors, each of which obtained a respective accuracy of 92.5%, 94.3% or 93.8% (Negrel et al., 2014). Even more interesting is the fact that, except for the study conducted by Castelluccio et al. (2015), most studies that compare deep learning with machine learning approaches for scene classification do not consider these su-

perior features, and instead just compare CNNs against simple features such as colour histograms or BoVW (Penatti et al., 2015; Cheng et al., 2017; Nogueira et al., 2017).

Lack of training data

Overtraining of deep learning systems is further exacerbated by the lack of large labelled datasets for scene classification (Castelluccio et al., 2015). Correctly training a deep learning system requires a colossal amount of diverse training data, sometimes even millions of labelled images (Alom et al., 2018). Datasets of such quality and magnitude have been developed for generic image classification with ImageNet being the most notable example, however, most labelled datasets of aerial images are either too small, do not contain enough images for all types of scenes, or are not diverse enough. This lack of training data has severely limited the use of deep learning methods in scene classification (Cheng et al., 2017).

The easiest approached to increase the size of the training dataset and reduce overfitting of deep learning systems is data augmentation, which consists of artificially expanding the labelled set of images by performing label-preserving transformations such as translation, rotation, or change of pixel intensity values (Krizhevsky et al., 2012). The limitation of this approach is that it only increases the size of the dataset and not its diversity, which is another important factor of good training data (Chen et al., 2018). A better approach is to create new datasets crafted for our particular problem. A notable and recent example is the case of Cheng et al. (2017), who created a dataset of 31500 aerial images of 45 different scenes, to be used for generic scene classification. To leverage the full power of CNNs for scene classification of the Amazon rainforest a similar such dataset would have to be developed for this particular task, but doing so requires substantial resources, particularly if the labels of images are assigned using ground truth data.

Expensive and slow computations

The two properties of deep learning systems described before, namely, the vast size of the architecture and the large training data requirements, lead to a third problem: extremely high computational costs. Training large deep learning systems requires powerful Graphic Processing Units (GPUs), coupled with other expensive hardware such as large memory bandwidths and fast CPUs (Alom et al., 2018). This hardware might be too expensive for many of the researchers or small organisations working on forest monitoring. Moreover, even when using the most powerful hardware, tranining deep learning systems

can take an absurd amount of time. For instance, training a single VGG network on the ImageNet dataset required between two and three weeks using NVIDIA Titan Black GPUs (Simonyan and Zisserman, 2014). Due to these long training times using GPUs as a service from a cloud platform might also be too costly.

Most important of all is the fact that such awfully slow training times might prohibit the wide use of deep learning architectures for remote sensing applications. The large constellation of satellites being deployed will acquire a large numbers of diverse remote sensing images at different spatial, spectral and temporal resolutions–not only will there be more remote sensing images, but these images contain more information and are therefore more computationally costly too process. This flood of data calls for much more computationally efficient techniques than those currently offered by deep learning architectures (Romero et al., 2016).

Difficulties in transfer learning

A solution that is commonly proposed to address the problems of insufficient training data and high computational costs is the concept of transfer learning (Castelluccio et al., 2015; Nogueira et al., 2017). Transfer learning makes it possible to use a powerful CNN that has already been trained on a given dataset, and optimise it for our use case with just a fraction of the data and computational costs that would be required to train the model from scratch. Transfer learning of CNNs stems from the observation that many deep learning architectures trained on different natural images converge to approximately similar parameters in those layers that are closest to the input—those resembling low level feature extraction (Yosinski et al., 2014). Consequently, in many applications it is not necessary to train CNNs from scratch, instead, it is possible to just train the highest levels of the architecture as those are usually more specific to any given task.

There are two different ways to use pre-trained CNNs. The first option is to remove the last layer of the network and use the CNN as a feature extractor to obtain the so called "deep features". Next, a classifier is trained using these features and training data specific to our application (Penatti et al., 2015). Castelluccio et al. (2015) tested this method on the UC Merced dataset by extracting deep features from two CNNs pre-trained on the ImageNet dataset: GoogleNet and CaffeNet—a CNN with a very similar architecture to AlexNet Jia et al. (2014). These features then served as input to a linear SVM classifier. Both architectures obtained more than 94% accuracy, performing better than when the CNNs were trained from scratch. The second and superior approach to transfer learning is called "fine-tuning" and it consists of keeping the CNN as a unified learning system and retraining only the last layers in the architecture

Castelluccio et al. (2015). With this method, the performance on the UC Merced dataset of CaffeNet increased to 95.48% and that of GoogleNet reached an outstanding 97.1% accuracy.

You might think that transfer learning solves all of the above-mentioned problems of deep learning architectures for scene classification. Unfortunately it does not. The success of transfer learning is highly dependent on the resemblance between the original task on which the CNN is trained on and the task for which the architecture is repurposed (Yosinski et al., 2014). CNNs trained on ImageNet adapt well to the UC Merced land cover dataset because images in both training sets have similar formats—each containing the three RGB colour channels of most generic images (Castelluccio et al., 2015). Nevertheless, a lot remote sensing data is not encoded in this format. As already discussed in section 2.1.2, what makes remote sensing sensors so valuable is the ability to capture information from many more spectral bands into multispectral or hyperspectral images (Shaw and Burke, 2003). To test the transferability of deep features to such remote sensing images Penatti et al. (2015) created a labelled dataset of Brazilian coffee scenes captured by the SPOT satellite in the green, red and near-infrared band. Next, they evaluated the performance of an SVM classifier used in conjunction with various different features, inlcuding deep features extracted from the Overfeat and CaffeNet CNNs pre-trained on ImageNet. Overfeat and CaffeNet deep features still achieved a high respective accuracy of 82% and 85%. However, both of these deep features—obtained from vast and complex Convolutional Neural Networks trained on a million images—were outperformed by a single simple BIC human engineered colour feature, which achieved an 87% accuracy on this dataset. In a later study by similar researchers, the BIC colour feature also outperformed other notable CNNs, including AlexNet, VGG_{16}, and GoogleNet (Nogueira et al., 2017). Fine-tuning works slightly better, with GoogleNet obtaining 90.75% classification accuracy on the coffee scenes dataset (Castelluccio et al., 2015).

Note that when fine-tuning GoogleNet to the coffee scenes dataset it performs substantially worse than when fine-tuning it to the UC Merced dataset (90.75% vs 97.1%), serving as evidence of the transferability problem to remote sensing data with different spectral bands. And the coffee scenes dataset is not even that different from regular RGB images. These transferability issues might become even more apparent when testing ImageNet pre-trained CNNs on entirely different remote sensing data such as radar, LiDAR or hyperspectral images. This is not to say that transfer learning is not suitable for remote sensing applications. Instead, deep learning architectures must be pre-trained on remote sensing data that has greater resemblance to that of the target application.

3 Data

3.1 Description

For this project I make use of a dataset of high resolution satellite images of the Amazon basin, a large region in South America mostly covered by the Amazon rainforest. The dataset consists of 40480 labelled training images and 61192 unlabelled test images. The images have been captured by Planet, a company that designs and manufactures Earth observation satellites. Planet released these images for free in the 2017 Kaggle competition "Understanding the Amazon from Space" (Planet, 2017).

Each image has a resolution of 256x256 pixels. After orthorectifying the image to eliminate optical distortions each pixel maps to 3 meters, however, originally each pixel corresponds to 3.7 meters on the ground—a measure known as ground sample distance. Thus, the total area covered by each image is 897,187.84 m^2, almost 90 hectares.

All images are available in formats JPEG and GeoTiff—a particular kind of TIFF image format with additional georeferencing information. The image format chosen for this project is GeoTiff as it provides significant advantages over JPEG. While the JPEG images just have the standard three RGB colour channels, the GeoTiff images have an additional Near Infrared channel (NIR) which is extremely useful for detecting vegetation (Tucker, 1979). Moreover, GeoTiff images are significantly richer in information than JPEG images, which have a form of lossy compression that sacrifices image quality in favour of a smaller storage size.

Labels

The labels that describe the contents of the image can be divided into two main groups: atmospheric conditions and land cover. Each image contains one of the following four atmospheric condition labels:

Figure 3.1: Cloudy

Cloudy: 90% or more of the image contains opaque clouds that obscure the view of the ground. Any image with the cloudy label cannot have any other label.

Figure 3.2: Haze

Haze: Translucent clouds which still enable a view of the ground.

Figure 3.3: Partly cloudy

Partly cloudy: Opaque clouds that are only partially present in the image.

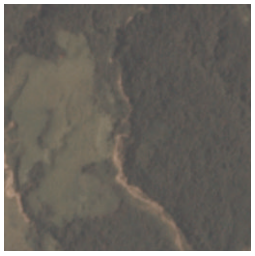


Figure 3.4: Clear

Clear: Images with no presence of clouds.

If an image does not have the cloudy label then it has one or more of the following land cover labels:

Figure 3.5: Primary

Primary: Refers to primary forest. We can distinguish between primary and secondary forests. Primary forests are pristine forests which have been unaffected for a long period of time, whereas secondary forests have regrown after a human or natural destruction of the forest. Since it is hard to distinguish primary and secondary rainforests from satellite images this label actually encompasses both types of rainforest.

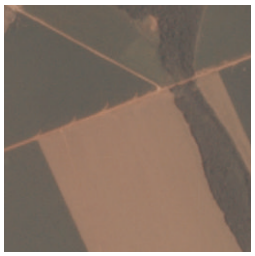


Figure 3.6: Road

Road: Presence of roads in the image. Identifying roads can be useful since the development of roads is often a precursor to deforestation.

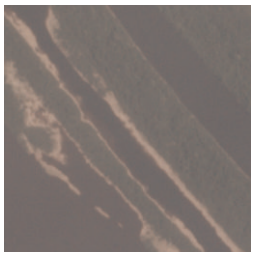


Figure 3.7: Water

Water: Rivers and lakes.

Figure 3.8: Agriculture

Agriculture: Refers explicitly to commercial agriculutre or range land, both of which are large drivers of deforestation in rainforests.

Figure 3.9: Cultivation

Cultivation: Shorthand for shifting cultivation, a relatively small scale form of agriculture practiced mostly in rural areas for subsistance.

Figure 3.10: Habitation

Habitation: Presence of buildings in the picture. The same term is used for small rural homes or large urban centres.

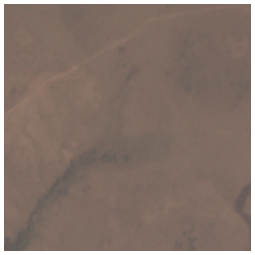


Figure 3.11: Bare ground

Bare ground: Used for areas which are naturally free of tree cover.

Figure 3.12: Selective logging

Selective logging: Logging of particular kinds of high value trees while leaving the rest intact.

Figure 3.13: Slash and burn

Slash and burn: Slash and burn is a really destructive practice by which humans deliberately cut and burn areas of forests in order to clear the land for other uses, particularly agriculture.

Figure 3.14: Conventional mining

Conventional mining: Large scale mining operations.

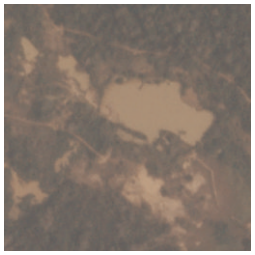


Figure 3.15: Artisinal mining

Artisinal mining: Small scale mining operations.

Figure 3.16: Blow down

Blow down: Refers to a natural phenomenon also known as windthrow, where strong winds uproot or topple large trees in the rainforest.

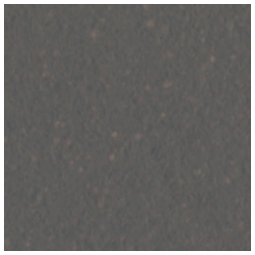
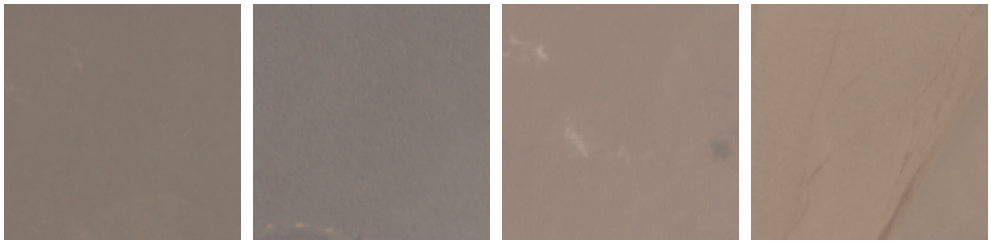


Figure 3.17: Blooming

Blooming: Planet (2017) defines blooming as a naturally ocurring phenomenon where certain trees bloom at the same same time. There does not seem to be record of this phenomenon in other online sources, and it might not be particularly relevant to the problem of deforestation.

Labelling process

The ideal method of labelling satellite images with the type land cover or land use present in the image is to use data from ground observations. Planet, the creator of the dataset used in this project, has opted instead to label the dataset via human interpretation of the satellite images. Their rationale for this method is that it enables them to generate much more labelled data at a lower cost and in a shorter time frame. However, it is not always easy to discern what is in a satellite image simply by observing it. Consequently, wrong labels may be assigned to images, leading to poor data quality. Consider for instance the images in Figure 3.18. These almost indistinguishable images supposedly all have different forms of land cover / land use.



(a) Primary (b) Cultivation (c) Water (d) Agriculture

Figure 3.18: Four images with the label haze and another different label

Planet claims that while the dataset might have defects, the proportion of correctly labelled data is large enough to outweigh the few incorrect samples. Nevertheless, because machine learning systems are highly reliant on correctly labelled training data in order to work properly, this dataset might pose an inherent limitation to the performance of the scene classification system developed in this project.

3.2 Exploration and Analysis

3.2.1 Label distribution

In Figure 3.19 we can see how frequently each label occurs in the training dataset—how many of the 40,480 images used to train our system are of rainforests, how many exhibit clear or hazy atmospheric conditions, and so on. It is immediately apparent that most images have the primary label, which designates that rainforest is present in the image. Moreover, this dataset is biased towards images with clear atmospheric conditions and away from cloudy images which obstruct the entire image. In terms of land cover and land use, we can clearly distinguish two subsets: one set of frequent labels such as agriculture, cultivation, habitation and bare ground which occur in 47% of the dataset, and a set of rare labels which includes selective logging, artisanal and conventional mines, blooming and blow down, which occurs in less than 4% of the dataset. Beware that this dataset might not be a representative sample of the set of all images of the Amazon Basin. Since we have no information on how the data has been sampled we cannot assume that the distribution observed in Figure 3.19 signifies the actual land cover and land use in the Amazon Basin.

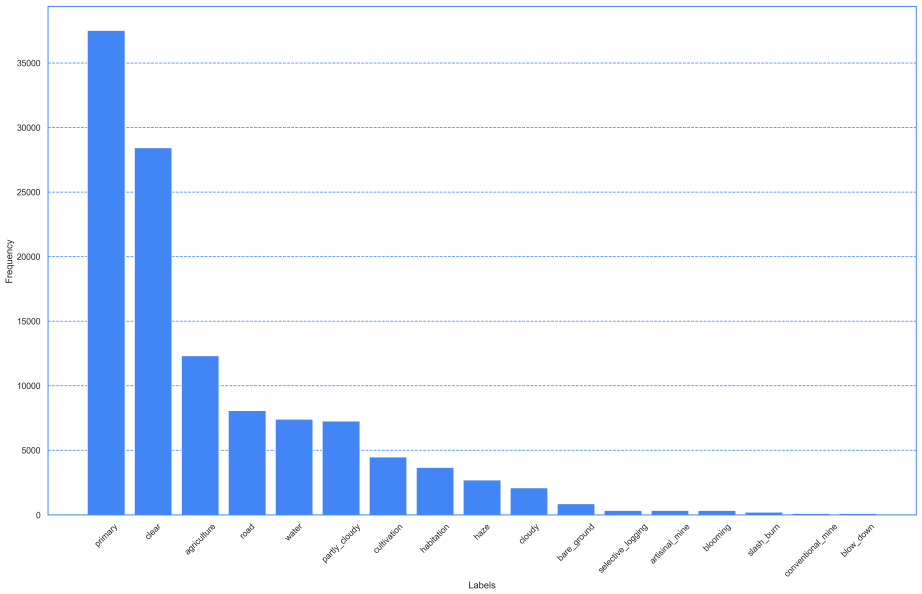


Figure 3.19: Label distribution of the training data. Total number of instances is 40,480

3.2.2 Label relations

Besides knowing how frequently each of the individual labels occur, it is also insightful to know how frequently they occur together. One method of visualising this information is to make use of a coocurrance matrix like the one in Figure 3.20, which displays how frequently the label on the X axis and the label on the Y axis occur together. It is immediately clear that the coocurrance matrix is not useful for this particular dataset because of the uneven label distribution—the primary, clear, and agriculture labels are so predominant that the relation between less frequent labels is not discernable.

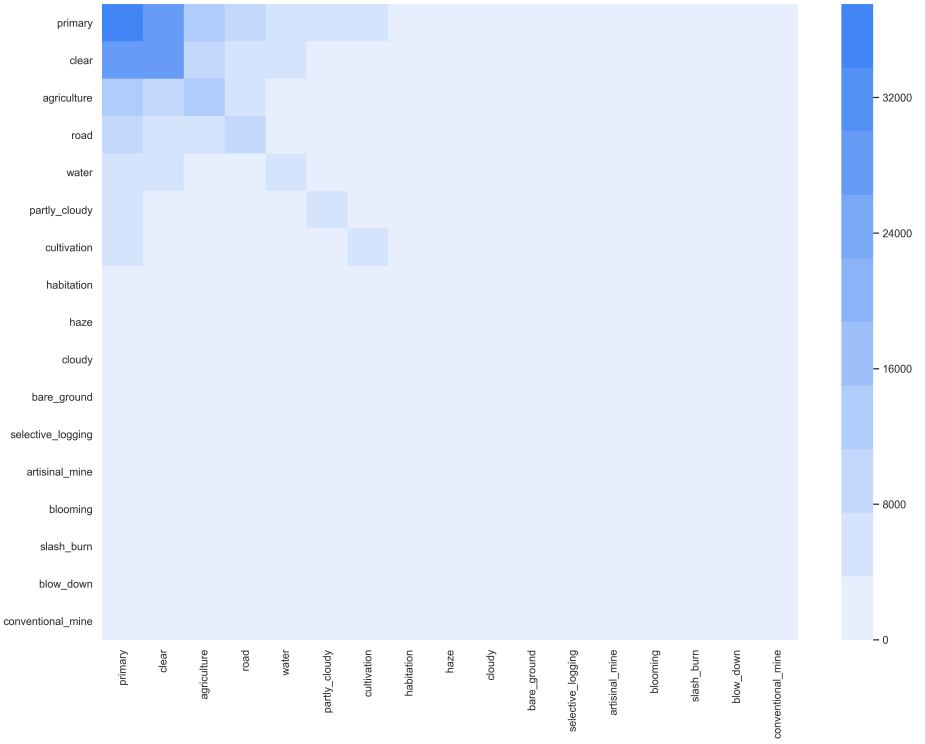


Figure 3.20: Label coocurrance matrix: how frequently the label on the X axis and the Y axis are present together in an image

An alternative approach is to employ a matrix of label relations, which shows what percentage of images with the label on the Y axis also has the label on the X axis. As we can observe in Figure 3.21 this method is much more informative. We can clearly see that all land covers often appear in concurrence with rainforests (primary label). Moreover, we can observe that the atmospheric condition labels are mutually exclusive, and that no form of land use or land cover is observed when an image is cloudy. Some other interesting insights

are that areas with habitation also have agriculture and roads, or that slash and burn often occurs near areas of agriculture or cultivation. These relations could be used as prior information in our system. For example, if the system has detected roads it could increase the probability of finding agriculture or habitation in that same image.

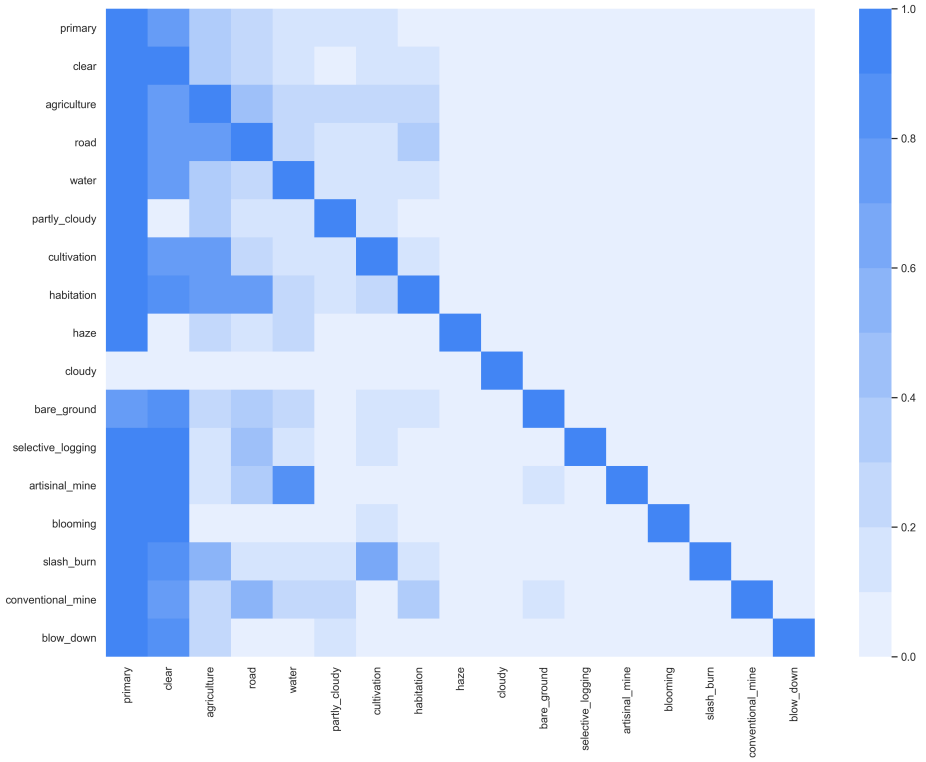


Figure 3.21: Label relation matrix: What percentage of instances with the label on the Y axis also has the label on the X axis.

3.3 Preprocessing

3.3.1 Image preprocessing

The original satellite images have already been processed by Planet to a certain extent: each image has the same squared aspect ratio, scale, resolution and colour channels (Planet, 2017). However, to extract features from these images it was necessary at times to convert the 16-bit GeoTiff image to an 8-bit format with values ranging between 0 and 255, convert the images to grayscale, or convert the data type from unsigned integer to signed float.

3.3.2 Label preprocessing

The labels assigned to each image in the training data are stored in a csv file with the format in table 3.1: one instance per row and two columns. The first column contains the image name and the second column contains the labels assigned to each image, with each label separated by a space. It was not necessary to remove any instances from the training data because each image had corresponding labels, and because each image was labelled exactly once. However, some images had duplicate labels, which were adequately removed.

image_name	tags
train_1	agriculture clear primary water

Table 3.1: Labels csv sample

While the original format of the label data can be easily interpreted by humans it is not ideal for computers. The csv file was preprocessed so that each label serves as an indicator variable: there is one column for each label, and each row has either a 1 or 0 for that column, indicating whether or not it contains that label. This format also makes it easier to analyse the data. Table 3.2 is an example with a limited number of labels, because the columns of all labels do not fit on the page.

image_name	primary	clear	agriculture	road	water	partly_cloudy
train_1	1	1	1	0	1	0

Table 3.2: Preprocessed labels csv sample (with limited number of columns)

4 Architecture

The architecture of the scene classification system developed in this project consists of a feature extractor and a classifier. From my analysis in section 2 I can confidently conclude that this architecture offers the best combination of high performance and low computational cost. Moreover, the machine learning classifier is drastically easier to implement than a knowledge based system, and requires far less training data than deep learning architectures.

The system works as follows. First, a set of features are extracted from the images in the training set. Next, the classifier is trained using the extracted features and the labels describing the scene present in each training image. With the trained classifier the system can be used to classify new aerial images of the Amazon rainforest. The way this is done is by first extracting the same set of features as is used for training and then feeding these features to the trained classifier, which predicts the image scene. In this section, I give a detailed description of the system architecture, followed by a brief explanation on how the system has been implemented.

4.1 Features

Table 4.1 contains an overview of the various features that are considered for the scene classification system developed in this project. Each of them is subsequently explained.

Type	Feature
Spectral	Statistical spectral features
	Normalized Difference Vegetation Index (NDVI).
Colour	Colour Histogram.
Texture	Gray Level Co-occurrence Matrix (GLCM): contrast, angular second moment, correlation.
	Local Binary Patterns (LBP).
Local	BRISK detector, BRISK/FREAK descriptor, Bag of Visual Words (BoVW).

Table 4.1: Overview of features

4.1.1 Spectral features

Statistical spectral features

The mean and standard deviation is computed for each of the spectral band of the images: blue, green, red and near-infrared (NIR). The mean value of each band is further aggregated by computing its mean into a value termed brightness.

Normalized Difference Vegetation Index

The Normalized Difference Vegetation Index (NDVI) is also used because of its importance in detecting vegetation (Tucker, 1979). The NDVI is computed for the entire image using equation 4.1, where NIR and Red refer to the near-infrared and red channels in the image. The mean and standard deviation from the NDVI are then extracted to serve as features for the model.

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{4.1}$$

4.1.2 Colour features

Colour histogram

The colour histogram (Swain and Ballard, 1991) is used as a feature to represent the overall colour in the image. More specifically, it expresses the probability of a pixel being of a given colour. Each bin of a colour histogram spans a range of colours, with each colour being defined as a unique set of red, green and blue values in the RGB colour space. The colour histogram has been implemented like in (Yang and Newsam, 2010), with 8 histogram bins for each of the three red, green and blue channels, resulting in a colour histogram of length $8^3 = 512$. Finally, the histogram vector is normalised to have L1 norm of 1, turning it into a colour probability distribution. The colour histogram is also computed in the CIE Lab and HSV colour space, in addition to the RGB colour space, because such colour spaces have led to improved performances in other land classification studies (Yang and Newsam, 2010).

Discarded alternatives

As described in section 2.1.3, there are other colour features that have achieved better performance for scene classification, however, these features do not

solely measure colour. For instance, the colour corrolelogram colour feature (Kumar et al., 1997) also measures the spatial distribution of colour in the image, the BIC colour feature (Stehling et al., 2002) classifies each pixel as border or interior, and the JAC colour feature (Williams and Yoon, 2007) also incorporates other measures such as density of edges. These extra properties make these colour features much harder to implement than colour histograms, and will therefore not be used in this project.

4.1.3 Texture features

The textural features used in this project are the Gray Level Co-ocurrance Matrix (GLCM) derived metrics and Local Binary Patterns (LBP).

Gray Level Co-ocurrance Matrix

The GLCM (Haralick et al., 1973) is a matrix that measures the frequency that two pairs of pixel brightness values occur together in an image. An example makes this definition more digestible. Given an image from the dataset, it is first converted to an 8-bit grayscale image, with values ranging between 0 and 255. The GLCM would then be a 256 dimensional square matrix. If a given pixel in the grayscale image has value 100 and an adjacent pixel has value 150, then the count of the cell $(100, 150)$ increases by one unit. It is advised for the GLCM to be symmetrical (Hall-Beyer, 2017), so in the previous example the count of the cell $(150, 100)$ would also increase. To complete the GLCM matrix this process is repeated for all 256 values in the image. Lastly, the GLCM is normalised to express probabilities (Hall-Beyer, 2017). The GLCM is sometimes computed from the NIR channel because it contains information that better distinguishes between land covers (Marceau et al., 1990; Kim et al., 2009).

The GLCM is a very large and sparse matrix, which makes it impractical to use as a feature. This drove Haralick et al. (1973) to invent a set of practical metrics that could be derived from the GLCM matrix. These metrics measure different properties of the GLCM, thus using a combination of them provide the greatest benefit for scene classification (Kim et al., 2009). The ones used in this project are contrast, angular second moment, and correlation. Using all 13 proposed measures is not necessary because they are highly correlated with the ones selected (Hall-Beyer, 2017). As a final remark on its implementation, GLCM is not only computed for pixels horizontally adjacent to eachother but also for vertically and diagonally adjacent pixels. This results in 4 GLCM matrices from which metrics are derived and averaged.

Local Binary Patterns

The other texture feature considered is Local Binary Patterns (LBP). Instead of examining the co-ocurrance of grey-level values in the whole image, texture can be characterised by examining the spatial distribution of grey-level values in a sub-region of the image (He and Wang, 1990). The original form of LBP (Ojala et al., 1994) examines texture at the scale of 3x3 pixels, comparing the grey-level values of the central pixel with the neighbouring pixels. If a given pixel is smaller than the central pixel it is assigned a 0, alternatively a 1. The resulting binary number formed by the 8 neighbouring pixels constitutes the local texture descriptor. The global texture feature is constructed by a histogram of all such local descriptors. Ojala et al. (2002) discovered that by just considering certain types of Local Binary Patterns termed "uniform" it was possible to create a much more compact and powerful texture feature descriptor. They further enhanced such uniform LBP by making them invariant to brightness and rotation. Finally, such LBP are not restricted to the 3x3 set of neighbouring pixels, and can be computed for any radius (R) and number of points (P) around each pixel.

4.1.4 Local features

Extracting local features is a two step process. First, a local feature detector identifies keypoints in a given image. Second, a feature descriptor is computed for each of the identified keypoints. The algorithm chosen to perform the detection is BRISK. For the feature descriptor both BRISK and FREAK are considered. Lastly, feature descriptors are encoded into a Bag of Visual Words (BoVW) feature vector that can be used for classification.

Detector

BRISK is a scale and rotation invariant feature detector and descriptor that performs as well as competing alternatives such as SIFT and SURF at a dramatically reduced computational cost (Leutenegger et al., 2011). The main steps of the BRISK detection algorithm are the following. First, a scale-space pyramid is constructed with a set of levels termed octaves and intra-octaves. The first octave in the pyramid corresponds to the original image, with every other octave being constructed by half-sampling its preceding octave. Intra-octaves are levels in the pyramid located between adjacent octaves. They are constructed similarly to octaves, by half-sampling its preceding intra-octave, with the difference of the first intra-octave being the original image downsampled by a factor of 1.5. The pyramid consists of a total of n octaves and n intra-octaves, with typically $n = 4$. This pyramid makes BRISK features scale

invariant as keypoints are identified at different levels of the scale-space. The identification of keypoints is performed with FAST (Rosten and Drummond, 2006), a robust and efficient detector used in many real-time applications. The FAST detector searches for keypoints in every octave and intra-octave that exceed a given threshold T. The last steps consist of rejecting low quality keypoints with non-maximum supression in scale space and performing sub-pixel and continuous scale refinement for each of the final keypoints. The result of the detection is a set of keypoints (x, y, σ), where x and y are pixel coordinates and σ is the scale of the detected keypoint.

Descriptor

Once the detection step is complete, the descriptor of the keypoints are computed. Both BRISK and FREAK generate binary descriptors by concatenating the results of simple brightness comparisons between pairs of pixels in a given neighbourhood of the keypoint. They differ in their approach to sampling the points in the neighbourhood of the keypoint and in how efficiently they achieve rotational invariance. BRISK samples neighbourhood points using a pattern of equally spaced points on circles concentric with the keypoint, similar to the DAISY descriptor (Tola et al., 2010). The sampled points are then divided into long-distance and short-distance pairings. The long-distance pairings are used to estimate the orientation of the descriptor, which is needed to rotate the keypoint neighbourhood and achieve rotation invariance. Lastly, the rotated short-distance pairings are used to construct the descriptor. In contrast, the FREAK descriptor samples neighbourhood points using a pattern inspired by the retina in the human visual system. Moreover, to estimate the keypoint orientation FREAK just requires 45 pairs, as opposed to a few hundreds in BRISK, resulting in a much faster computation and smaller memory load (Alahi et al., 2012). The final BRISK and FREAK descriptor corresponds to a 512-dimensional binary local feature vector.

Bag of Visual Words

After detection and description we are left with a multitude of multidimensional local feature vectors from each image. Using these local features directly for classification faces the same challenges as those faced by pixel based classification, described in section 2.1.1. To overcome these limitations, a basic approach used in this project is to reduce the local features into a Bag of Visual Words model (BovW) (Sivic and Zisserman, 2003). The BoVW is a histogram with frequencies of different local features occurring in an image. However, counting the number of *unique* local features will result in an impractically long feature vector of 2^{512} dimensions, since each local feature is

a 512-dimensional binary vector. Instead, a dimensionality reduction technique called quantization is used to compress each feature vector into a single value representing that feature, termed a "visual word". The set of all k visual words is termed a visual dictionary. With this new representation, the BoVW histogram will only have k dimensions, counting how often each of the k visual words appear in the image. Lastly, to compensate for the variability in the number of local features between images the BoVW feature vector is normalised to have L1 norm of 1.

The question still remains of how to determine which local feature maps to which visual word. For the BoVW model this is done by the clustering of all local features into k clusters, each of which corresponds to a single visual word. Note that only the training data is used to create the clusters and the local features from images in the test data are assigned to the closest cluster. The clustering algorithm used is mini-batch k-means, an efficient modification of the commonly used k-means algorithm (Sculley, 2010). A final caveat to take into account is that the k-means clustering algorithm does not work well with binary descriptors such as BRISK or FREAK. The first issue is that k-means assigns each vector to a cluster by computing the Euclidean distance (Lloyd, 1982), whereas distance between binary descriptors is computed using the Hamming distance (Alahi et al., 2012). More importantly, Trzcinski et al. (2012) has demonstrated that binary vectors are harder to assign to a given cluster because there are many binary vectors that are at an equal distance from two random binary vectors. Lastly, the high dimensionality of BRISK and FREAK descriptors further complicate the clustering process. A solution to this problem suggested by Lynen et al. (2014) is to project the high dimensional binary vector onto a lower dimensional vector of real numbers. Using this technique, each local feature vector is reduced from 512 to 64 dimensions.

4.2 Classifier

The classifier used in this project is random forests because, as detailed in section 2.2.3, it is the most suited classifiers for scene classification. Random forests is part of a group of methods known as "ensemble methods" which combine multiple classifiers to obtain a higher performance than any of the individual classifiers could achieve on its own. Specifically, random forests is an ensemble of classifiers known as decision trees. To understand random forests it is useful to start by understanding how decision tree classifiers work, and how these are combined to generate the superior random forest classifier.

4.2.1 Decision Tree

The decision tree classifier pertient to this project is trained according to the CART algorithm, introduced by Breiman (1984). It work as follows. Given a training set of image features and the labels associated to those images, a decision tree splits the training set into two subsets according to a given feature j and a threshold t_j. All instances with values of feature j below that threshold are moved to one subset and values above the threshold to another. In order to find the optimal subsets the decision tree utilizes the gini impurity measure (G). If a subset only contains instances with one particular label then $G = 0$ and the subset is deemed to be "pure". The decision tree strives to find the pair of values (j, t_j) that produce the purest subsets, weighted by the proprtion of instances in that subset. In summary, the cost function that the decision tree minimises is given by equation 4.2, where n is the total number of instances in the training set, $n_{left/right}$ is the number of instances in each subset, and $G_{left/right}$ is the gini impurity of each subset.

$$Cost_{tree}(t, t_j) = \frac{n_{left}}{n} G_{left} + \frac{n_{right}}{n} G_{right} \tag{4.2}$$

This operation is performed recursively on each of the subsets until a certain stopping condition is met and training completes. Predictions are made on a new instance by traversing through the tree using the (j, t_j) values at each node until a certain leaf node is reached. The new instance is assigned the most dominant label at the given leaf node.

For certain complex datasets, like that of image features, there is an inherent limitation to the performance of individual decision trees (Ho, 1995). Because decision trees are non-parametric models they are likely to overfit to the training data and generalise poorly to new data if they are allowed to grow too large and complex. Generalisation ability could be improved by pruning the tree and making it simpler, however, this results in poorer performance on the training data.

4.2.2 Random forests

Ho (1995) overcame the performance limitations of individual decision trees by combining multiple trees—i.e. creating a forest. He had observed that in other contexts the generalisation errors of a single classifier could be compensated by combining multiple individual classifiers with ensemble methods. A necessary condition for ensemble methods to outperform its constituent classifiers is for such classifiers to be both accurate and diverse, with diversity being defined as making different errors on unseen data (Dietterich, 2000). How can

such diversity be achieved in identical decision trees that are trained on identical data? The solution proposed by Ho (1995) was to train each decision tree on a random subset of the training data—giving rise to the concept of random forests. Such a random subset is selected via sampling with replacement, making it possible for certain instances to be sampled several times and other instances not to be sampled at all. This method is known as bootstrap aggregating or bagging.

The random forest method was further improved by Breiman (2001) who expanded the diversity of each decision tree by getting each tree to only consider a random sub-sample of all features when splitting the data at each node in the tree. The predictions of the random forest model are performed by combining the predictions of each of the individual decision trees, weighted by the probabilities associated to each prediction.

The main parameters to chose in a random forest model are the number of trees (N_{tree}) and the number of features evaluated at each node (M_{try}). After reviewing various studies using random forest for aerial image interpretation, Belgiu and Drăguţ (2016) suggest using $N_{tree} = 500$ and setting M_{try} to the square root of the number of total features.

A final positive property of random forests that is very important for this project is the fast training time of the model. Because each of the decision trees in the model are independent from one another, they can be trained in parallel. The computational complexity of training is given by 4.3, where n is the number of training samples and p is the number of processors (Belgiu and Drăguţ, 2016).

$$O(\ \frac{N_{tree}}{p}\ \sqrt{M_{try}\, n\, log(n)}\ ) \tag{4.3}$$

There are even more complex ensemble methods, known as boosting, which train various classifiers sequentially, each making up for the errors produced of the previous one. However, they are substantially slower to train than random forests due to their sequential nature and do not provide any notable performance benefits (Breiman, 2001). These results are empirically supported by Li et al. (2016) and Belgiu and Drăguţ (2016) for remote sensing data.

4.3 Implementation

I have implemented all aspects of the system with the programming language Python version 3.7. This includes the data preprocessing and analysis, extraction of various features, the development of the Random Forest model and the performance evaluation with various configurations outlined in section 5. The development of the system would not have been possible without the help of numerous open source libraries. The Python Data and Analysis library (Pandas) (McKinney et al., 2010) was useful for gaining insights into the data and to preprocess the labels. The NumPy library (Van Der Walt et al., 2011) made it possible to store images in efficient data structures and to easily compute spectral and colour features. The scikit-image library (Van der Walt et al., 2014) facilitated the processing of images, as well as the acquisition of texture features, while the OpenCV library (Bradski, 2000) was useful for obtaining local features. The scikit-learn library (Pedregosa et al., 2011) was an indispensable resource for developing a highly performant and configurable random forest classifier and for easily evaluating the performance the system. Finally, the Jupyter Notebook environment (Pérez and Granger, 2007) made it faster to develop code and visualise results.

All the code developed for this project is released as open-source software under the MIT license at `https://github.com/SamLubbers/rainforest_scene_classification`. By making my code open source I aspire to help other people around the world to freely use and expand upon my work. With this in mind, I have strived to develop the code in accordance with important software design principles such as low coupling, high cohesion, reusability, minimization of duplicate code and good documenation.

5 Design of Experiments

This section consists of two main parts. In the first part I start by explaining why and how I create a validation dataset for this project. I then describe which metrics are used to judge the performance of the system, and how such metrics are computed. I end by outlining how the computational efficiency of the system is assessed. In the second part I describe the experiments that carried out in this project.

5.1 Evaluation method

5.1.1 Validation data

The performance evaluation of a supervised machine learning system is done by comparing the predicted labels on a given set of data with the actual labels associated to that data. In this project, it consists of comparing the forms of land cover that the system identifies in a given image with the forms of land cover actually present in the image. In particular, it is necessary to evaluate the generalisation ability of the machine learning system—how well it performs on images it has never seen before. We thus need one labelled set of images for training and another labelled set for testing (Géron, 2017).

The dataset used in this project does not have labels for the test images because it has been obtained from a Kaggle competition. The only way to evaluate the performance on the test data is by uploading a submission to the Kaggle competition where the dataset originates from. This approach is very limited as we are restricted to use the performance metric used in the given competition, and we cannot gain insights into the performance on particular scenes. To solve this problem I divide the labelled training data into two subsets: a training set, containing 80% of the data (32383 images), and a validation set with the other 20% (8096 images).

When splitting data into subsets it is important to minimise sampling bias as it can negatively affect the performance of our system (Sechidis et al., 2011). As an extreme example, if all agriculture images were placed in the validation set, the model would have no training data to learn to identify such images, leading to a large number of errors. The ideal method to avoid sampling bias is stratified sampling, which guarantees that the label distribution of the training and validation sets resemble that of the original training data. But stratified sampling has the downside of being hard to implement for datasets like the one used in this project, in which each instance can have more than one label (Sechidis et al., 2011). The alternative method, random sampling,

is much easier to implement and works well enough for large datasets due to the Law of Large Numbers. Random sampling is therefore used to generate the training and validation sets in this project. You can verify in table 5.1 how there is minimal sampling bias introduced by random sampling, with a few exceptions for uncommon labels.

label	original distribution	% diff. training	% diff. validation
primary	0.93	-0.10	0.39
clear	0.70	-0.22	0.89
agriculture	0.30	0.15	-0.61
road	0.20	-0.37	1.47
water	0.18	-0.15	0.59
partly cloudy	0.18	0.18	-0.70
cultivation	0.11	-0.41	1.63
habitation	0.09	-1.33	5.33
haze	0.07	1.09	-4.34
cloudy	0.05	1.01	-4.02
bare ground	0.02	-1.10	4.41
selective logging	0.01	0.37	-1.47
artisinal mine	0.01	0.66	-2.66
slash burn	0.01	-3.11	12.44
blooming	0.01	1.66	-6.63
conventional mine	0.002	3.75	-15.00
blow down	0.002	-3.06	12.24

Table 5.1: Difference in label distributions between the training data and the training and validation subsets

5.1.2 Evaluation metrics

Evaluation metrics are used to objectively quantify the generalisation performance of a machine learning system. Two important metrics are *Precision* and *Recall*. *Precision* measures the proportion of correct predictions. If the system predicts an image to contain rainforests, *Precision* expresses the probability that the image actually has rainforests, as opposed to having incorrectly identified another form of land cover as rainforest. *Recall* measures the proportion of land cover types that are correctly identified in images. If an image contains rainforests, *Recall* expresses the probability that the system will correctly identify rainforests in the given image.

For multi-label classification problem such as this one, in which each image can have multiple forms for land cover, *Precision* and *Recall* are first cal-

culated separately for every image in the validation set, and then aggregated by computing the mean across all values (Zhang and Zhou, 2014). More concretely, let $\mathbf{x_i}$ be the vector with the set of features extracted for image i of the validation set, $\mathbf{y_i}$ the labels representing the various forms land cover present in image i of the validation set, $h()$ the classifier, $h(\mathbf{x_i})$ the predicted labels of image i, and p the number of images in the validation set. *Precision* is given by equation 5.1 and *Recall* is given by equation 5.2

$$Precision = \frac{1}{p}\sum_{i=1}^{p}\frac{|\mathbf{y_i} \cap h(\mathbf{x_i})|}{|h(\mathbf{x_i})|} \tag{5.1}$$

$$Recall = \frac{1}{p}\sum_{i=1}^{p}\frac{|\mathbf{y_i} \cap h(\mathbf{x_i})|}{|\mathbf{y_i}|} \tag{5.2}$$

It is often convenient to have a single metric which can be used to easily compare the performance of different system configurations. To that end, precision and recall can be combined into the so called F_β score, given by equation 5.3. When $\beta = 1$, the F_1 score is the harmonic mean of precision and recall. The harmonic means is preferred over the regular mean for combining *Precision* and *Recall* because both metrics need to be high for the F_1 score to be high (Géron, 2017). Nevertheless, the F_1 score weighs precision and recall equally, which is not a desirable property for this particular problem. We are much more interested in always detecting the areas inflicted by deforestation (high recall) to accidentally misidentifying certain areas as deforested when they are not (lower precision). Thus, the evaluation metric used in this project is the F_2 score, an F_β score which weighs *Recall* higher than *Precision* by having $\beta = 2$ in equation 5.3. The values of the F_2 score range from 0 to 1, with 1 being the best possible score with perfect *Precision* and *Recall*.

$$F_\beta = (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall} \tag{5.3}$$

When classifying aerial images with the goal of monitoring deforestation it is more important to identify certain forms of land use such as mining or slash and burn than others such as water or bare ground. If we just use an aggregate metric, we could obtain a high F_2 score if the system misclassifies all images with certain rare but important land cover types and correctly classifies the rest, leading us to erroneously believe that the system performs well. For this reason, in addition to measuring the mean F_2 score of all instances, I also measure the F_2 score for each of the individual labels. To obtain the label-based F_2 score for multi-label classification we first compute the number of true positive (TP), false positive (FP) and false negative (FN) for each of the j labels. Given y_{ij} to be the label j of the validation image i, and $h(\mathbf{x_i})_j$ the

predicted value of the label j of the validation image i, the TP, FP and FN of the j-th label are given respectively by equations 5.4, 5.5 and 5.6 (Zhang and Zhou, 2014).

$$TP_j = |\{\mathbf{x_i}|(y_{ij} = 1) \wedge (h(\mathbf{x_i})_j = 1), 1 \leq i \leq p\}| \tag{5.4}$$

$$FP_j = |\{\mathbf{x_i}|(y_{ij} = 0) \wedge (h(\mathbf{x_i})_j = 1), 1 \leq i \leq p\}| \tag{5.5}$$

$$FN_j = |\{\mathbf{x_i}|(y_{ij} = 1) \wedge (h(\mathbf{x_i})_j = 0), 1 \leq i \leq p\}| \tag{5.6}$$

From these values we can obtain the *Precision* and *Recall* for each j label with equations 5.7 and 5.8. Finally, the label-based F_2 score is computed using equation 5.3 with $\beta = 2$.

$$Precision_j = \frac{TP_j}{TP_j + FP_j} \tag{5.7}$$

$$Recall_j = \frac{TP_j}{TP_j + FN_j} \tag{5.8}$$

5.1.3 Computational efficiency

As previously emphasised, in scene classification systems the quality of predictions is not all that matters. The computational efficiency of the system must also be taken into account. In this project I measure the feature extraction time on the 32383 images in the training set and the model training time, as those are the most time consuming computations. Note that the feature extraction computations include the loading of images into memory as it was unfeasible to load all images into memory all at once—the images in the training set alone already occupied 16 GB of memory.

These computational efficiency measurements are strongly influenced by the hardware used and the particular implementation. All experiments are executed on an Intel Core i7 of 2.5 GHz with 8 cores. Both the training of the model and the extraction of most features is performed in parallel using all cores. The only exception is the extraction of local features, which are performed in parallel using 8 threads.

5.2 Experimental setup

Two main types of experiments are carried out in this project. The first type of experiments evaluates how different feature configurations impact the overall performance of the scene classification system. The computational efficiency of the different configurations is also tested. The configurations tested are the following:

- Statistical spectral features without the near-infrared (NIR) spectral band, with NIR and with NDVI.
- Colour histogram on three different colour spaces: RGB, CIE Lab and HSV.
- GLCM on the grayscale image vs GLCM on NIR image channel.
- LBP at various scales with parameters $(R, P) = (1, 8), (2, 16), (3, 24)$ and a multi-scale LBP which concatenates the LBP features at different scales.
- Various Local feature configurations. Unless better parameters are found, each of the following local feature experiments on local features uses by default a BRISK detector with detection threshold $T = 30$, a BRISK descriptor and a BoVW with 100 words.
  - Various pattern scales to create the descriptor: $S = 1, 3, 5$.
  - Various detection threshold values: $T = 20, 25, 30$.
  - BRISK vs FREAK detector.
  - Varying number of visual words: $10, 25, 50, 100, 250$.

The second type of experiment performed compares features with their optimal parameters side by side in terms of overall F_2 score, per class F_2 score and computational efficiency.

All experiments are performed using the data described in section 3.1 and with the evaluation methods proposed in the previous section 5.1. The architecture of the system used in these experiments is outlined in section 4.

6 Results

Before running the experiments, a baseline result has been created in order to have a reference value against which to compare the performance of the different system configurations. The baseline has been obtained using a ZeroR classifier, which "predicts" each instance of the validation set to have the most common label. Because each image must have one atmospheric condition label and one land cover label, each instance is assigned the 'clear' and 'primary' labels. This classifier produces a baseline $F_2 = 0.645$

6.1 Optimal feature configurations

6.1.1 Spectral features

Spectral features are the fastest features to extract, requiring about a minute to obtain the spectral features from all 32383 images in the training data. Moreover, see in table 6.1 how the inclusion of the near-infrared band improves the F_2 by a few decimals, whereas including NDVI on top of that does not contribute much. This might be due to the fact that the NDVI is highly correlated to the other spectral features, as it is calculated using the red and NIR bands.

	RGB	RGB & NIR	RGB & NIR & NDVI
F_2	0.801	0.824	0.826
t_x	64.617	57.786	83.234

Table 6.1: F_2 score and feature extraction time (t_x) of statistical spectral features without the near-infrared (NIR) spectral band, with NIR and with NDVI

6.1.2 Colour histogram

The findings of Yang and Newsam (2010) are replicated in this experiment, with the HSV colour space outperforming the more common RGB, although for this dataset the improvement is neglible. Moreover, changing the colour space of the entire image increases feature extraction time by almost a factor of two.

	RGB	CIE Lab	HSV
F_2	0.793	0.776	0.797
t_x	136.203	264.479	238.217

Table 6.2: F_2 score and feature extraction time (t_x) of colour histograms on three different colour spaces: RGB, CIE Lab and HSV.

6.1.3 GLCM

On this dataset, the GLCM performed on the grayscale image distinguishes better between land covers than GLCM performed on the NIR channel (table 6.3), a finding that contradicts the suggestions of some papers (Marceau et al., 1990; Kim et al., 2009).

	Grayscale	NIR
F_2	0.743	0.705
t_x	192.731	103.534

Table 6.3: F_2 score and feature extraction time (t_x) of GLCM on the grayscale image and GLCM on the NIR image channel.

In table 6.4 you can observe that the grayscale GLCM also outperforms the NIR GLCM when used together with spectral features. Testing texture features together with spectral features is necessary because they are usually used like this in scene classification systems, instead of as singular features.

	Grayscale	NIR
F_2	0.835	0.825
t_x	271.220	177.783

Table 6.4: F_2 score and feature extraction time (t_x) of spectral features used in combination with GLCM on the grayscale image and GLCM on the NIR image channel.

6.1.4 LBP

Regarding the scale of the LBP pattern, a larger radius R and considering more number of points P around each pixel improves the performance when LBP is used individually (table 6.5), but decreases the performance when combined with spectral features (table 6.6). Moreover, as suggested by Ojala et al. (2002) the best performing LBP feature is that which considers various scales, however, extracting LBP features at all all these different scales carries a much larger computational cost.

	R=1, P=8	R=2, P=16	R=3, P=24	(R,P) = (1,8)+(2,16)+(3, 24)
F_2	0.758	0.768	0.781	0.794
t_x	175.052	216.762	258.676	628.859

Table 6.5: F_2 score and feature extraction time (t_x) of LBP at various scales (varying (R, P) parameters) and a multi-scale LBP.

	R=1, P=8	R=2, P=16	R=3, P=24
F_2	0.833	0.830	0.830
t_x	242.325	283.983	325.206

Table 6.6: F_2 score and feature extraction time (t_x) of spectral features combined with LBP at various scales

6.1.5 Local features

As described in section 4.1.4, BRISK and FREAK local features are created by simple brightness comparisons of pairs of pixels sampled from the neighbourhood of each keypoint using a particular pattern. As you can observe in table 6.7, the larger this pattern is, the poorer the F_2 score, reaching a value that is even lower than the baseline. Such a low performance can be attributed to the fact that many of the detected keypoints are discarded when considering a larger pattern scale S, as the area expanded by this pattern exceeds the image boundaries—something that is made apparent by the faster feature extraction times for larger pattern scales. Note that in all of feature extraction times (t_x) for local features also include the time involved in creating the visual dictionary for the BoVW and the time needed to convert local feature descriptors to BoVW feature vectors.

	$S = 1$	$S = 3$	$S = 5$
F_2	0.721	0.709	0.630
t_x	504.628	293.916	208.529

Table 6.7: F_2 score and feature extraction time (t_x) of the BRISK descriptor generated with a pattern at varying scales (S). The complete feature extractor uses a BRISK detector with detection threshold $T = 30$, and a BoVW with 100 words.

Regarding the detection threshold T, the lower the value the higher the F_2 score as many more keypoints are considered (table 6.8). However, there is a compromise, since detecting more keypoints also requires more time to compute the descriptor for each of the detected keypoints and to create the visual dictionary for the BoVW representation. When the detection threshold is too low, such as $T = 20$, the feature extraction time balloons to over half

an hour.

	$T = 20$	$T = 25$	$T = 30$
F_2	0.731	0.725	0.721
t_x	2,172.631	862.015	504.628

Table 6.8: F_2 score and feature extraction time (t_x) of the BRISK detector with various threshold values (T). The complete feature extractor uses a BRISK descriptor created with a pattern scale of $S = 1$, and a BoVW with 100 words.

In table 6.9 you can observe that the FREAK feature detector serves as an advantageous alternative to the BRISK descriptor, not only due to its faster classification time but also because of its higher performance.

	BRISK	FREAK
F_2	0.725	0.746
t_x	862.015	828.954

Table 6.9: F_2 score and feature extraction time (t_x) of the BRISK descriptor compared with the FREAK descriptor. Both create the descriptors using a patterns scale of $S = 1$. The complete feature extractor uses a BRISK detector with detection threshold $T = 25$ and a BoVW with 100 words.

Table 6.10 shows how the number of visual words in the BoVW representation affects the performance of local features using a BRISK detector and a FREAK descriptor. Surprisingly, the best performance is achieved using only 25 visual words, contrary to the finding of Yang and Newsam (2010), who recommended using 1000 visual words for scene classification using SIFT local features. This finding might imply that FREAK local feature descriptors require drastically fewer visual words than SIFT for their successful encoding. If this is the case it adds even further evidence for the use of FREAK descriptors as a faster alternative to SIFT, as fewer visual words lead to much faster feature extraction, evidence in table 6.10.

	10	25	50	100	250
F_2	0.749	0.751	0.748	0.746	0.742
t_x	453.909	535.363	607.694	828.954	1,353.107

Table 6.10: F_2 score and feature extraction time (t_x) with varying number of visual words for the BoVW encoding method. The complete feature extractor uses a BRISK detector with detection threshold $T = 25$ and a FREAK descriptor created with a pattern scale of $S = 1$

6.2 Feature comparisons

Figure 6.1 compares the performance of all features using their optimal parameters. These results are largely consistent with that of the studies reviewed in section 2.1. Figure 6.2 makes it easy to visualise the computational efficiency of the system with different features, a property lacking in many other studies.

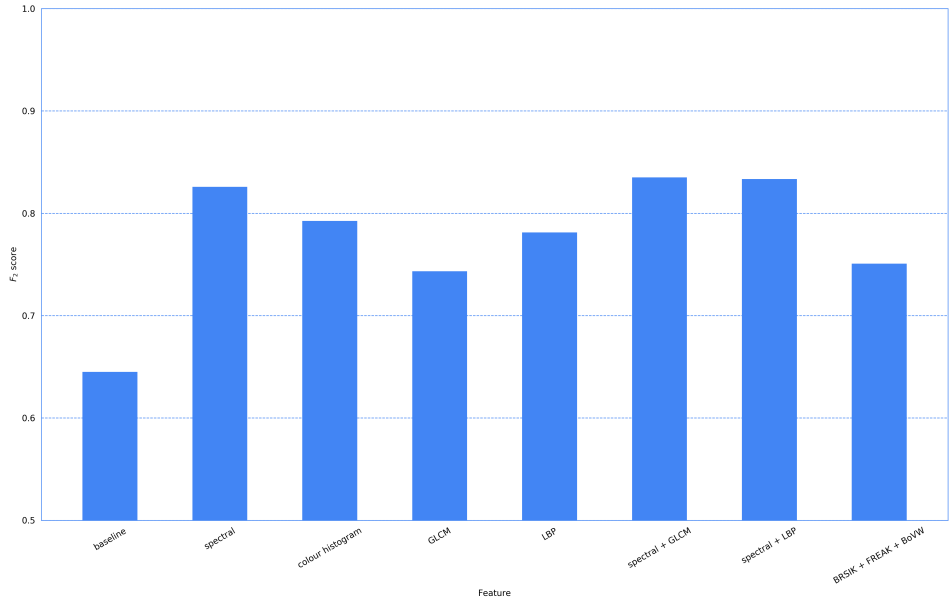


Figure 6.1: F_2 score of the tested features using their optimal parameters

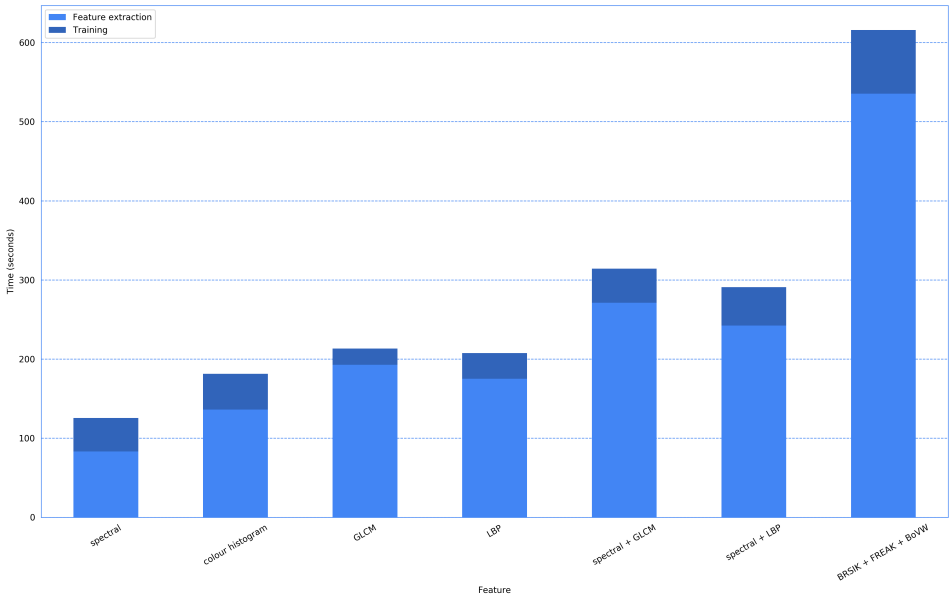


Figure 6.2: Feature extraction and training time of the tested features using their optimal parameters

While colour histograms and spectral features encode much of the same information, spectral features perform better as they also contain information from the near-infrared spectral band. As expected, spectral features outperform texture features when used individually. However, when texture features are used in conjunction with spectral features they do not achieve as much of a performance improvement as recorded by other researchers. Moreover, the slight improvement in performance that is obtained comes at the cost of almost three times as much time for feature extraction. Curiously, LBP performs better than GLCM on its own, but worse when combined with spectral features.

Regarding local features, the performance achieved is akin to that of other studies, where using a simple Bag of Visual Words is not enough to outperform spectral or texture features. Although compared to the other features in this study it might seem like local features are very slow to obtain, the feature extraction time achieved with the BRISK detector and FREAK descriptors is really efficient for local features.

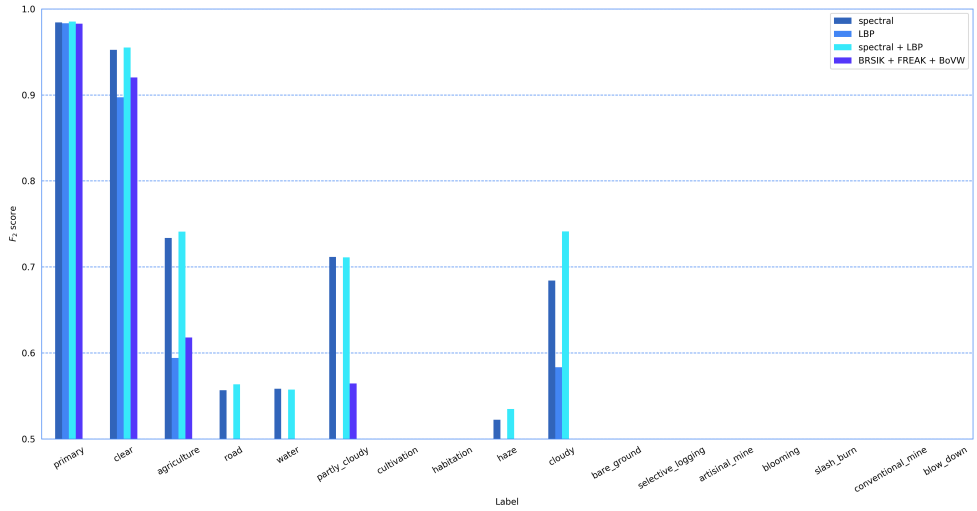


Figure 6.3: Per label F_2 score on various feature configurations.

Lastly, figure 6.3 gives us some more insight into how the model performed with different features on this dataset. The labels are organised by frequency of occurrence, similar to figure 3.19. What really stands out from the per label F_2 scores is the fact that very high results are obtained for three most common scenes, and an F_2 score of 0 is obtained for many of the rare scenes. This phenomenon is consistent for all the tested features, and might boil down to the fact that the Random Forest classifier fails to effectively deal with imbalanced training data and tends to favour classifying those labels that appear most frequently (Belgiu and Drăguţ, 2016). Improving the performance of the overall system might be possible by performing data augmentation on images of important land cover types that have a low representation in the dataset. Moreover, images of "blooming" or "blow down" scenes could be dispensed off as they are not needed for monitoring deforestation.

7 Conclusion

7.1 Contribution

I was driven to undertake this project because of its promising potential future use for helping us combat deforestation. When starting off, I had absolutely no idea how scene classification systems worked. Therefore, instead of jumping straight ahead and developing the most trendy and powerful deep learning architecture, I thought it would be much more beneficial for my project to first extensively learn about all the research and developments that had taken place up until the present day, and to summarise this information in a complete and comprehensive literature and technology survey. In my messy quest through more than a hundred research papers I encountered some very remarkable findings. Firstly, there is sufficient evidence to counter the dogma that deep learning architectures have made traditional computer vision systems obsolete. In the field of digital analysis of aerial images there are some important problems of insufficient training data and computational resources that limit the widespread adoption of deep learning architectures. Moreover, CNNs pre-trained on generic datasets such as ImageNet do not transfer well to remote sensing images that encode information in different spectral bands.

A second major finding is the remarkably high performance achieved in scene classification when using local features in combination with encoding methods such as Fisher Vectors or Vectors of Locally Aggregated Tensors. Curiously, many researchers still use these powerful encoding methods with SIFT local features, even though other local features such as BRISK or FREAK have long proven to be just as robust as SIFT while being drastically faster to compute. In this project I have demonstrated that the BRISK detector used in combination with the FREAK descriptor can be used as a local feature for scene classification. Moreover, I discovered that there might not be such thing as the best feature for scene classification. Each type of feature encodes a particular kind of information, and when used together with an appropriate fusion strategy they can produce truly outstanding results, superior to that achieved by any single feature on its own. Lastly, I have released these findings, along with the code developed in this project free of charge with the goal that other students, researchers and organisations can benefit from this work and expand upon it.

7.2 Limitations and future work

Unfortunately, I was unable to implement the promising feature fusion strategies or local feature encoding methods due to my lack of technical capabilities. In future work I would not only try to implement these methods, but I will also strive to include these into computer vision libraries to make such useful methods more accessible. Moreover, this project would have been ever more complete if I had been able to implement deep learning methods in order to compare these side by side in terms of performance and computational efficiency with machine learning methods.

Despite the problems currently faced by deep learning systems in remote sensing applications, Convolutional Neural Networks are a very promising architecture due to the impressive results they achieved in other domains. Moreover, because CNNs can be used for so many different image classification tasks, state-of-the-art architectures have been incorporated into frameworks such as Caffe (Jia et al., 2014), making them much easier to implement than complex feature extractors. Consequently, in order to make more accurate and accessible scene classification systems, the best approach might be to develop new datasets rather than new systems. If we had large and varied datasets of labelled remote sensing images it would be possible to succesfully train state-of-the-art Convolutional Neural Networks for scene classification. Once CNNs are trained on such datasets they could be fine-tuned to a myriad of scene classification applications with proportionally little training data and computational resources.

Bibliography

Alahi, A., Ortiz, R. and Vandergheynst, P. (2012), FREAK: Fast Retina Keypoint, *in* '2012 IEEE Conference on Computer Vision and Pattern Recognition', pp. 510–517.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Van Esesn, B. C., Awwal, A. A. S. and Asari, V. K. (2018), 'The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches', *arXiv:1803.01164 [cs]* . arXiv: 1803.01164.
URL: *http://arxiv.org/abs/1803.01164*

Alpaydin, E. (2016), *Machine learning: the new AI*, MIT press.

Baker, D. J., Richards, G., Grainger, A., Gonzalez, P., Brown, S., DeFries, R., Held, A., Kellndorfer, J., Ndunda, P., Ojima, D., Skrovseth, P.-E., Souza, C. and Stolle, F. (2010), 'Achieving forest carbon information with higher certainty: A five-part plan', *Environmental Science & Policy* **13**(3), 249–260.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S1462901110000225*

Barlow, J., Lennox, G. D., Ferreira, J., Berenguer, E., Lees, A. C., Nally, R. M., Thomson, J. R., Ferraz, S. F. d. B., Louzada, J., Oliveira, V. H. F., Parry, L., Ribeiro de Castro Solar, R., Vieira, I. C. G., Arago, L. E. O. C., Begotti, R. A., Braga, R. F., Cardoso, T. M., Jr, R. C. d. O., Souza Jr, C. M., Moura, N. G., Nunes, S. S., Siqueira, J. V., Pardini, R., Silveira, J. M., Vaz-de Mello, F. Z., Veiga, R. C. S., Venturieri, A. and Gardner, T. A. (2016), 'Anthropogenic disturbance in tropical forests can double biodiversity loss from deforestation', *Nature* **535**(7610), 144–147.
URL: *http://www.nature.com/articles/nature18326*

Bay, H., Tuytelaars, T. and Van Gool, L. (2006), SURF: Speeded Up Robust Features, *in* A. Leonardis, H. Bischof and A. Pinz, eds, 'Computer Vision ECCV 2006', Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 404–417.

Belgiu, M. and Drăguţ, L. (2016), 'Random forest in remote sensing: A review of applications and future directions', *ISPRS Journal of Photogrammetry and Remote Sensing* **114**, 24–31.
URL: *http://www.sciencedirect.com/science/article/pii/S0924271616000265*

Blaschke, T. (2010), 'Object based image analysis for remote sensing', *ISPRS Journal of Photogrammetry and Remote Sensing* **65**(1), 2–16.
URL: *http://www.sciencedirect.com/science/article/pii/S0924271609000884*

Bradski, G. (2000), 'The OpenCV Library', *Dr. Dobb's Journal of Software Tools* .

Breiman, L. (1984), *Classification and Regression Trees*, 1st edn, Routledge, New York.
URL: *https://www.taylorfrancis.com/books/9781315139470*

Breiman, L. (2001), 'Random Forests', *Machine Learning* **45**(1), 5–32.
URL: *https://doi.org/10.1023/A:1010933404324*

Castelluccio, M., Poggi, G., Sansone, C. and Verdoliva, L. (2015), 'Land Use Classification in Remote Sensing Images by Convolutional Neural Networks', *arXiv:1508.00092 [cs]* . arXiv: 1508.00092.
URL: *http://arxiv.org/abs/1508.00092*

Chen, G., Weng, Q., Hay, G. J. and He, Y. (2018), 'Geographic object-based image analysis (GEOBIA): emerging trends and future opportunities', *GIScience & Remote Sensing* **55**(2), 159–182.
URL: *https://doi.org/10.1080/15481603.2018.1426092*

Cheng, G., Han, J., Guo, L., Liu, Z., Bu, S. and Ren, J. (2015), 'Effective and Efficient Midlevel Visual Elements-Oriented Land-Use Classification Using VHR Remote Sensing Images', *IEEE Transactions on Geoscience and Remote Sensing* **53**(8), 4238–4249.

Cheng, G., Han, J. and Lu, X. (2017), 'Remote Sensing Image Scene Classification: Benchmark and State of the Art', *Proceedings of the IEEE* **105**(10), 1865–1883.

Cook, J., Oreskes, N., Doran, P. T., Anderegg, W. R. L., Verheggen, B., Maibach, E. W., Carlton, J. S., Lewandowsky, S., Skuce, A. G., Green, S. A., Nuccitelli, D., Jacobs, P., Richardson, M., Winkler, B., Painting, R. and Rice, K. (2016), 'Consensus on consensus: a synthesis of consensus estimates on human-caused global warming', *Environmental Research Letters* **11**(4), 048002.
URL: *https://iopscience.iop.org/article/10.1088/1748-9326/11/4/048002*

Crevier, D. and Lepage, R. (1997), 'Knowledge-Based Image Understanding Systems: A Survey', *Computer Vision and Image Understanding* **67**(2), 161–185.
URL: *http://www.sciencedirect.com/science/article/pii/S1077314296905202*

Csillik, O. (2017), 'Fast Segmentation and Classification of Very High Resolution Remote Sensing Data Using SLIC Superpixels', *Remote Sensing* **9**(3), 243.
URL: *https://www.mdpi.com/2072-4292/9/3/243*

Dietterich, T. G. (2000), Ensemble Methods in Machine Learning, *in* 'Multiple Classifier Systems', Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 1–15.

Domingos, P. (2012), 'A few useful things to know about machine learning', *Communications of the ACM* **55**(10), 78.
URL: *http://dl.acm.org/citation.cfm?doid=2347736.2347755*

Draper, B. A., Hanson, A. R. and Riseman, E. M. (1996), 'Knowledge-directed vision: control, learning, and integration', *Proceedings of the IEEE* **84**(11), 1625–1637.

Dronova, I. (2015), 'Object-Based Image Analysis in Wetland Research: A Review', *Remote Sensing* **7**(5), 6380–6413.
URL: *https://www.mdpi.com/2072-4292/7/5/6380*

FAO (2016), *Global Forest Resources Assessment*, FAO, Rome. OCLC: 964542220.
URL: *http://www.fao.org/forest-resources-assessment/en/*

FAO, ed. (2018), *Forests pathways to sustainable development*, number 2018 *in* 'State of the world's forests', FAO, Rome.

Fassnacht, F. E., Hartig, F., Latifi, H., Berger, C., Hernndez, J., Corvaln, P. and Koch, B. (2014), 'Importance of sample size, data type and prediction method for remote sensing-based estimations of aboveground forest biomass', *Remote Sensing of Environment* **154**, 102–114.
URL: *http://www.sciencedirect.com/science/article/pii/S0034425714003022*

Fukushima, K. (1980), 'Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position', *Biological Cybernetics* **36**(4), 193–202.
URL: *https://doi.org/10.1007/BF00344251*

Géron, A. (2017), *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 1st edn, O'Reilly Media, Inc.

Grainger, A. (2008), 'Difficulties in tracking the long-term global trend in tropical forest area', *Proceedings of the National Academy of Sciences* **105**(2), 818–823.
URL: *http://www.pnas.org/cgi/doi/10.1073/pnas.0703015105*

Grainger, A. and Obersteiner, M. (2011), 'A framework for structuring the global forest monitoring landscape in the REDD+ era', *Environmental Science & Policy* **14**(2), 127–139.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S146290111000136X*

Haines, T. S. F. and Xiang, T. (2014), 'Background Subtraction with Dirichlet Process Mixture Models', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(4), 670–683.

Hall-Beyer, M. (2017), 'Practical guidelines for choosing GLCM textures to use in landscape classification tasks over a range of moderate spatial scales', *International Journal of Remote Sensing* **38**(5), 1312–1338.
URL: *https://doi.org/10.1080/01431161.2016.1278314*

Haralick, R. M., Shanmugam, K. and Dinstein, I. (1973), 'Textural Features for Image Classification', *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-3**(6), 610–621.

Hawkins, D. M. (2004), 'The Problem of Overfitting', *Journal of Chemical Information and Computer Sciences* **44**(1), 1–12.
URL: *https://doi.org/10.1021/ci0342472*

He, D.-c. and Wang, L. (1990), 'Texture Unit, Texture Spectrum, And Texture Analysis', *IEEE Transactions on Geoscience and Remote Sensing* **28**(4), 509–512.

Ho, T. K. (1995), Random decision forests, *in* 'Proceedings of 3rd International Conference on Document Analysis and Recognition', Vol. 1, pp. 278–282 vol.1.

Hofmann, P., Blaschke, T. and Strobl, J. (2011), 'Quantifying the robustness of fuzzy rule sets in object-based image analysis', *International Journal of Remote Sensing* **32**(22), 7359–7381.
URL: *https://doi.org/10.1080/01431161.2010.523727*

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T. (2014), Caffe: Convolutional Architecture for Fast Feature Embedding, *in* 'Proceedings of the 22Nd ACM International Conference on Multimedia', MM '14, ACM, New York, NY, USA, pp. 675–678. event-place: Orlando, Florida, USA.
URL: *http://doi.acm.org/10.1145/2647868.2654889*

Jgou, H., Perronnin, F., Douze, M., Snchez, J., Prez, P. and Schmid, C. (2012), 'Aggregating Local Image Descriptors into Compact Codes', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(9), 1704–1716.

Keenan, R. J., Reams, G. A., Achard, F., de Freitas, J. V., Grainger, A. and Lindquist, E. (2015), 'Dynamics of global forest area: Results from the FAO Global Forest Resources Assessment 2015', *Forest Ecology and Management* **352**, 9–20.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S0378112715003400*

Kim, M., Madden, M. and Warner, T. A. (2009), 'Forest Type Mapping using Object-specific Texture Measures from Multispectral Ikonos Imagery'.

Kobayashi, T. (2014), Dirichlet-Based Histogram Feature Transform for Image Classification, *in* '2014 IEEE Conference on Computer Vision and Pattern Recognition', pp. 3278–3285.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012), 'ImageNet classification with deep convolutional neural networks', *Communications of the ACM* **60**(6), 84–90.
URL: *http://dl.acm.org/citation.cfm?doid=3098997.3065386*

Kumar, a. S. R., Mitra, M. and Zabih, a. R. (1997), Image indexing using color correlograms, *in* 'Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition', pp. 762–768.

Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998), 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE* **86**(11), 2278–2324.

Leutenegger, S., Chli, M. and Siegwart, R. Y. (2011), BRISK: Binary Robust invariant scalable keypoints, *in* '2011 International Conference on Computer Vision', pp. 2548–2555.

Li, M., Ma, L., Blaschke, T., Cheng, L. and Tiede, D. (2016), 'A systematic comparison of different object-based classification techniques using high spatial resolution imagery in agricultural environments', *International Journal of Applied Earth Observation and Geoinformation* **49**, 87–98.
URL: *http://www.sciencedirect.com/science/article/pii/S0303243416300125*

Lloyd, S. (1982), 'Least squares quantization in PCM', *IEEE Transactions on Information Theory* **28**(2), 129–137.

Lowe, D. G. (1999), Object recognition from local scale-invariant features, *in* 'Proceedings of the Seventh IEEE International Conference on Computer Vision', Vol. 2, pp. 1150–1157 vol.2.

Lowe, D. G. (2004), 'Distinctive Image Features from Scale-Invariant Keypoints', *International Journal of Computer Vision* **60**(2), 91–110.
URL: *https://doi.org/10.1023/B:VISI.0000029664.99615.94*

Lynen, S., Bosse, M., Furgale, P. and Siegwart, R. (2014), Placeless Place-Recognition, *in* '2014 2nd International Conference on 3D Vision', Vol. 1, pp. 303–310.

Ma, L., Fu, T., Blaschke, T., Li, M., Tiede, D., Zhou, Z., Ma, X. and Chen, D. (2017), 'Evaluation of Feature Selection Methods for Object-Based Land

Cover Mapping of Unmanned Aerial Vehicle Imagery Using Random Forest and Support Vector Machine Classifiers', *ISPRS International Journal of Geo-Information* **6**(2), 51.
URL: *https://www.mdpi.com/2220-9964/6/2/51*

Ma, L., Li, M., Ma, X., Cheng, L., Du, P. and Liu, Y. (2017), 'A review of supervised object-based land-cover image classification', *ISPRS Journal of Photogrammetry and Remote Sensing* **130**, 277–293.
URL: *http://www.sciencedirect.com/science/article/pii/S092427161630661X*

Marceau, D. J., Howarth, P. J., Dubois, J. M. and Gratton, D. J. (1990), 'Evaluation Of The Grey-level Co-occurrence Matrix Method For Land-cover Classification Using Spot Imagery', *IEEE Transactions on Geoscience and Remote Sensing* **28**(4), 513–519.

Matsuyama, T. (1987), 'Knowledge-Based Aerial Image Understanding Systems and Expert Systems for Image Processing', *IEEE Transactions on Geoscience and Remote Sensing* **GE-25**(3), 305–316.

Maxwell, A. E., Warner, T. A. and Fang, F. (2018), 'Implementation of machine-learning classification in remote sensing: an applied review', *International Journal of Remote Sensing* **39**(9), 2784–2817.
URL: *https://doi.org/10.1080/01431161.2018.1433343*

McKinney, W. et al. (2010), Data structures for statistical computing in python, *in* 'Proceedings of the 9th Python in Science Conference', Vol. 445, Austin, TX, pp. 51–56.
URL: *conference.scipy.org/proceedings/scipy2010/mckinney.html*

Negrel, R., Picard, D. and Gosselin, P. (2014), Evaluation of second-order visual features for land-use classification, *in* '2014 12th International Workshop on Content-Based Multimedia Indexing (CBMI)', pp. 1–5.

Nogueira, K., Penatti, O. A. and dos Santos, J. A. (2017), 'Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification', *Pattern Recogn.* **61**(C), 539–556.
URL: *https://doi.org/10.1016/j.patcog.2016.07.001*

Ojala, T., Pietikainen, M. and Harwood, D. (1994), Performance evaluation of texture measures with classification based on Kullback discrimination of distributions, *in* 'Proceedings of 12th International Conference on Pattern Recognition', Vol. 1, pp. 582–585 vol.1.

Ojala, T., Pietikainen, M. and Maenpaa, T. (2002), 'Multiresolution gray-scale and rotation invariant texture classification with local binary patterns', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(7), 971–987.

Patz, J. A., Graczyk, T. K., Geller, N. and Vittor, A. Y. (2000), 'Effects of environmental change on emerging parasitic diseases', *International Journal for Parasitology* **30**(12-13), 1395–1405.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S0020751900001417*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011), 'Scikit-learn: Machine learning in Python', *Journal of Machine Learning Research* **12**, 2825–2830.
URL: *http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html*

Penatti, O. A., Nogueira, K. and Santos, J. A. d. (2015), Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?, *in* '2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)', pp. 44–51.

Pérez, F. and Granger, B. E. (2007), 'Ipython: a system for interactive scientific computing', *Computing in Science & Engineering* **9**(3), 21–29.
URL: *https://ieeexplore.ieee.org/abstract/document/4160251*

Perronnin, F., Snchez, J. and Mensink, T. (2010), Improving the Fisher Kernel for Large-Scale Image Classification, *in* K. Daniilidis, P. Maragos and N. Paragios, eds, 'Computer Vision ECCV 2010', Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 143–156.

Picard, D. and Gosselin, P.-H. (2013), 'Efficient image signatures and similarities using tensor products of local descriptors', *Computer Vision and Image Understanding* **117**(6), 680–687.
URL: *https://hal.archives-ouvertes.fr/hal-00799074*

Planet (2017), 'Understanding the amazon from space'. [Online; accessed 2-November-2018].
URL: *https://www.kaggle.com/c/planet-understanding-the-amazon-from-space*

Ripple, W. J., Wolf, C., Newsome, T. M., Galetti, M., Alamgir, M., Crist, E., Mahmoud, M. I., Laurance, W. F. and 15,364 scientist signatories from 184 countries (2017), 'World Scientists Warning to Humanity: A Second Notice', *BioScience* **67**(12), 1026–1028.
URL: *https://academic.oup.com/bioscience/article/67/12/1026/4605229*

Romero, A., Gatta, C. and Camps-Valls, G. (2016), 'Unsupervised Deep Feature Extraction for Remote Sensing Image Classification', *IEEE Transactions on Geoscience and Remote Sensing* **54**(3), 1349–1362.

Rosten, E. and Drummond, T. (2006), Machine Learning for High-Speed Corner Detection, *in* A. Leonardis, H. Bischof and A. Pinz, eds, 'Computer Vision ECCV 2006', Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 430–443.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. and Fei-Fei, L. (2014), 'ImageNet Large Scale Visual Recognition Challenge', *arXiv:1409.0575 [cs]* . arXiv: 1409.0575.
URL: *http://arxiv.org/abs/1409.0575*

Santos, J. A. d., Penatti, O. A. B. and Torres, R. d. S. (2010), Evaluating the Potential of Texture and Color Descriptors for Remote Sensing Image Retrieval and Classification, *in* 'VISAPP'.

Sculley, D. (2010), Web-scale K-means Clustering, *in* 'Proceedings of the 19th International Conference on World Wide Web', WWW '10, ACM, New York, NY, USA, pp. 1177–1178. event-place: Raleigh, North Carolina, USA.
URL: *http://doi.acm.org/10.1145/1772690.1772862*

Sechidis, K., Tsoumakas, G. and Vlahavas, I. (2011), On the Stratification of Multi-label Data, *in* D. Gunopulos, T. Hofmann, D. Malerba and M. Vazirgiannis, eds, 'Machine Learning and Knowledge Discovery in Databases', Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 145–158.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. and LeCun, Y. (2013), 'OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks', *arXiv:1312.6229 [cs]* . arXiv: 1312.6229.
URL: *http://arxiv.org/abs/1312.6229*

Shao, W., Yang, W., Xia, G.-S. and Liu, G. (2013), A Hierarchical Scheme of Multiple Feature Fusion for High-Resolution Satellite Scene Categorization, *in* M. Chen, B. Leibe and B. Neumann, eds, 'Computer Vision Systems', Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 324–333.

Shaw, G. A. and Burke, H.-h. K. (2003), 'Spectral Imaging for Remote Sensing', **14**(1), 26.

Simonyan, K. and Zisserman, A. (2014), 'Very Deep Convolutional Networks for Large-Scale Image Recognition', *arXiv:1409.1556 [cs]* . arXiv: 1409.1556.
URL: *http://arxiv.org/abs/1409.1556*

Sivic, J. and Zisserman, A. (2003), Video Google: a text retrieval approach to object matching in videos, *in* 'Proceedings Ninth IEEE International Conference on Computer Vision', pp. 1470–1477 vol.2.

Solomon, S., on Climate Change, I. P. and on Climate Change, I. P., eds (2007), *Climate change 2007: the physical science basis: contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*, Cambridge University Press, Cambridge ; New York. OCLC: ocn132298563.

Song, C., Yang, F. and Li, P. (2010), Rotation Invariant Texture Measured by Local Binary Pattern for Remote Sensing Image Classification, *in* '2010 Second International Workshop on Education Technology and Computer Science', Vol. 3, pp. 3–6.

Stehling, R. O., Nascimento, M. A. and Falco, A. X. (2002), A compact and efficient image retrieval approach based on border/interior pixel classification, *in* 'Proceedings of the eleventh international conference on Information and knowledge management', ACM, pp. 102–109.
URL: *http://dl.acm.org/citation.cfm?id=584792.584812*

Swain, M. J. and Ballard, D. H. (1991), 'Color indexing', *International Journal of Computer Vision* **7**(1), 11–32.
URL: *https://doi.org/10.1007/BF00130487*

Szegedy, C., and, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015), Going deeper with convolutions, *in* '2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1–9.

Tola, E., Lepetit, V. and Fua, P. (2010), 'DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**(5), 815–830.

Trzcinski, T., Lepetit, V. and Fua, P. (2012), 'Thick boundaries in binary space and their influence on nearest-neighbor search', *Pattern Recognition Letters* **33**(16), 2173–2180.
URL: *http://www.sciencedirect.com/science/article/pii/S0167865512002619*

Tucker, C. J. (1979), 'Red and photographic infrared linear combinations for monitoring vegetation', *Remote Sensing of Environment* **8**(2), 127–150.
URL: *http://www.sciencedirect.com/science/article/pii/0034425779900130*

Tuytelaars, T. and Mikolajczyk, K. (2007), 'Local Invariant Feature Detectors: A Survey', *Foundations and Trends in Computer Graphics and Vision* **3**(3), 177–280.
URL: *http://www.nowpublishers.com/article/Details/CGV-017*

UNFCCC (2014), 'Report of the Conference of the Parties on its nineteenth session, Warsaw, 1123 November 2013. Addendum.

FCCC/CP/2013/10/Add.1'.
URL: *https://unfccc.int/node/8106*

Van Der Walt, S., Colbert, S. C. and Varoquaux, G. (2011), 'The numpy array: a structure for efficient numerical computation', *Computing in Science & Engineering* **13**(2), 22.
URL: *https://ieeexplore.ieee.org/document/5725236*

Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E. and Yu, T. (2014), 'scikit-image: image processing in python', *PeerJ* **2**, e453.
URL: *https://peerj.com/articles/453/*

Wieland, M. and Pittore, M. (2014), 'Performance Evaluation of Machine Learning Algorithms for Urban Pattern Recognition from Multi-spectral Satellite Images', *Remote Sensing* **6**(4), 2912–2939.
URL: *https://www.mdpi.com/2072-4292/6/4/2912*

Williams, A. and Yoon, P. (2007), 'Content-based image retrieval using joint correlograms', *Multimedia Tools and Applications* **34**(2), 239–248.
URL: *https://doi.org/10.1007/s11042-006-0087-2*

Williams, M. (2006), *Deforesting the Earth: From Prehistory to Global Crisis, An Abridgment*, University of Chicago Press.
URL: *https://www.press.uchicago.edu/ucp/books/book/chicago/D/bo3770940.html*

Yang, Y. and Newsam, S. (2010), 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', GIS '10, ACM, New York, NY, USA, pp. 270–279. event-place: San Jose, California.
URL: *http://doi.acm.org/10.1145/1869790.1869829*

Yosinski, J., Clune, J., Bengio, Y. and Lipson, H. (2014), How Transferable Are Features in Deep Neural Networks?, *in* 'Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2', NIPS'14, MIT Press, Cambridge, MA, USA, pp. 3320–3328. event-place: Montreal, Canada.
URL: *http://dl.acm.org/citation.cfm?id=2969033.2969197*

Zhang, M. and Zhou, Z. (2014), 'A Review on Multi-Label Learning Algorithms', *IEEE Transactions on Knowledge and Data Engineering* **26**(8), 1819–1837.

Zhong, Y., Zhu, Q. and Zhang, L. (2015), 'Scene Classification Based on the Multifeature Fusion Probabilistic Topic Model for High Spatial Resolution Remote Sensing Imagery', *IEEE Transactions on Geoscience and Remote Sensing* **53**(11), 6207–6222.