

Exploring Avenues For Future Research: Coastline Feature Extraction

Tulsi Patel

912100

Submitted to Swansea University in partial fulfilment
of the requirements for the Degree of Master of Science



Swansea University
Prifysgol Abertawe

Department of Computer Science
Swansea University

30th September 2020


Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed  (candidate)
Date 30/09/2020


Statement 1

This work is the result of my own independent study/investigations, except where otherwise stated. Other sources are clearly acknowledged by giving explicit references. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure of this work and the degree examination as a whole.

Signed  (candidate)
Date 30/09/2020

Statement 2

I hereby give my consent for my work, if accepted, to be archived and available for reference use, and for the title and summary to be made available to outside organisations.

Signed  (candidate)
Date 30/09/2020

Abstract

Coastline mapping is safety critical problem for worldwide shipping. The complexity of global coastlines and lack of labelled images has been a challenging task for modern techniques to map accurately. This paper looks to find a human centred approach by creating a tool that allows experts to label large satellite image datasets. The initial problem space of feature extraction from satellite images via artificial intelligence networks is scoped. Then dimensionality reduction to map those latent features into a two dimensional tool. A pipeline for such a process has been implemented with a example implementation using Auto-Encoders.

In this we find that Auto-Encoders are suitable for feature extraction and can cluster the data well. However when considering coastal features the complexity of problem is clearly shown with features being so varied. Future work looks to implement new models discussed in the paper and to expand the pipeline proposed.

Acknowledgements

Firstly, I'd like to thanks my supervisor, Mark Jones for his guidance and help. Secondly friends and family for supporting me. Finally members of the CDT at Swansea for keeping me sane.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 2 |
| 2 | Literature Review | 3 |
| 2.1 | Sentinel and SEN2COR Pre-processing [13, 36] | 3 |
| 2.2 | Normalised Difference Water Index | 4 |
| 2.3 | Early Feature Extraction | 5 |
| 2.4 | Feature Reduction using Machine Learning Approaches | 5 |
| 2.5 | Enhancing Low Resolution Images Imaging | 6 |
| 2.6 | Artificial Intelligence Techniques | 7 |
| 2.6.1 | Basic Feed Forward NN [43] | 7 |
| 2.7 | CNN's | 9 |
| 2.7.1 | Convolution Layer, Pooling and Fully Connected Layers | 9 |
| 2.8 | Stacked and Sparse Autoencoders | 10 |
| 2.8.1 | Sparse Auto Encoders | 10 |
| 2.8.2 | Stacked Auto Encoders | 11 |
| 2.9 | Recurrent Convolutional Neural Networks | 11 |
| 2.10 | Long Short Term Memory | 12 |
| 2.11 | Transformers | 12 |
| 2.12 | Towards Human Centered and Responsible Innovation | 14 |
| 2.12.1 | Human Centred Design | 14 |
| 2.12.2 | Responsible Innovation | 14 |
| 3 | Pipeline Architecture | 16 |
| 3.1 | Exploring the Dataset and Preprocessing | 16 |
| 3.2 | Different Models | 18 |
| 3.2.1 | Auto-Encoders | 18 |
| 3.2.2 | LSTM | 19 |
| 3.2.3 | Transformers | 21 |
| 3.3 | Reducing Latent Features | 22 |
| 4 | Experimental Results | 24 |
| 4.1 | AutoEncoders Model | 24 |
| 4.2 | Two Dimensional Plot | 26 |
| 5 | Conclusion and Future Work | 30 |
| 5.1 | Conclusion | 30 |
| 5.2 | Future Work | 30 |

Chapter 1

Introduction

Worldwide shipping relies on up-to-date and accurate navigational charts on a global scale. Coastal shoreline are a feature that need to be mapped robustly and frequently for commercial navigational purposes. Containing a vast array of unique geographical features makes them notoriously hard to map. Coastal shorelines can vary from mangrove forests, sheer cliffs to coastal cities. Changes by man-made and geographical processes adds a temporal challenge. Satellites are the most common method for mapping, with a geocentric orbit they can provide periodic and large swath images of earth's surface. By using modern artificial intelligence techniques a geo-generalisable model could be generated to map coastlines. However challenges arise when considering the fairly small size of coastal features compared to the large pixel resolution of satellite images; creating a uncertain or "fuzzy" boundary between water and land. Combined with the challenge of no ground truth information, labelling data for training models becomes extremely difficult. Therefore a method to extract features from satellite data and label them accurately is essential. Manual labelling would be a difficult task to achieve with the sheer amount, variation and frequency of data. Likewise using purely algorithmic techniques to label data would prove unsubstantial due to such varying features. This paper looks for a method to combine both approaches by creating a tool that aids in extracting features and allows a human to select the features. An optimal solution would present a visual tool that can allow the user to select higher level extracted features of multiple images to label.

The aims of this research will look to scope and mould future research. Finding key areas of research that allow feature extraction for the creation of tool to help expert users to labels mass quantities of data. In addition it will create a simple pipeline of the necessary methods and look to implement them.

Images in this paper are taken from the Copernicus Sentinel 2 missions operated by the European Space Agency. The payload provides wide swath, high-resolution, multi-spectral images. The mission is comprised of 2 different satellites in the same orbit but phased at 180 degrees. The multi-spectral images are comprised of 13 bands; four bands at 10m, six at 20m and three at 60m spatial resolution.

Chapter 2

Literature Review

2.1 Sentinel and SEN2COR Pre-processing [13, 36]

Sentinel 2 mission is a joint adventure between European Commission(EC) and the European Space Agency(ESA) to form the Global Monitoring for Environment and Security(GMES). In a paper provided by the ESA and industrial teams from Astrium is a full breakdown of all of the Sentinel 2 mission specifications and processing of images before being distributed[13]. There are two main levels of images available 1C and a further processed image 2A. Firstly we will talk about images in level 1C which is provided by GMES and then 2A, using a framework called Sen2Cor, developed by Telespazio VEGA Deutschland GmbH on behalf of ESA[36].

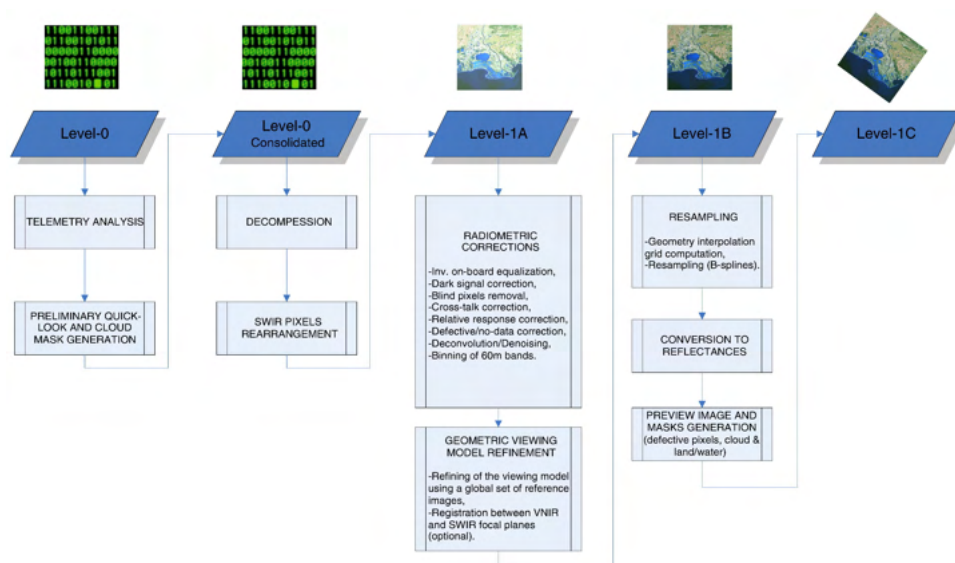


Figure 2.1: Processing of Sentinel images from level-0 to level-1C. Taken from [13].

Figure 2.1 shows the pipeline for creating 1C images from the initial Instrument Source Packets (ISP). Level 0 segments the ISP into granules and detects any initial errors by comparing the data to pre-defined ranges for the values. Granules, sometimes referred to as tiles, simply refers to the 100 by 100km segmentation of Earths' ortho-images. Furthermore level 0C are produced by creating a cloud mask based on the spectral criteria in the preliminary quicklook. Level 1A are produced by decompressing the ISP data and formatting to a JPEG2000 format. Level 1B data is radiometrically corrected radiances

and geometrically refined. Finally level 1C provides a top of atmosphere(TOA) reflectance and a cloud , land/water mask.

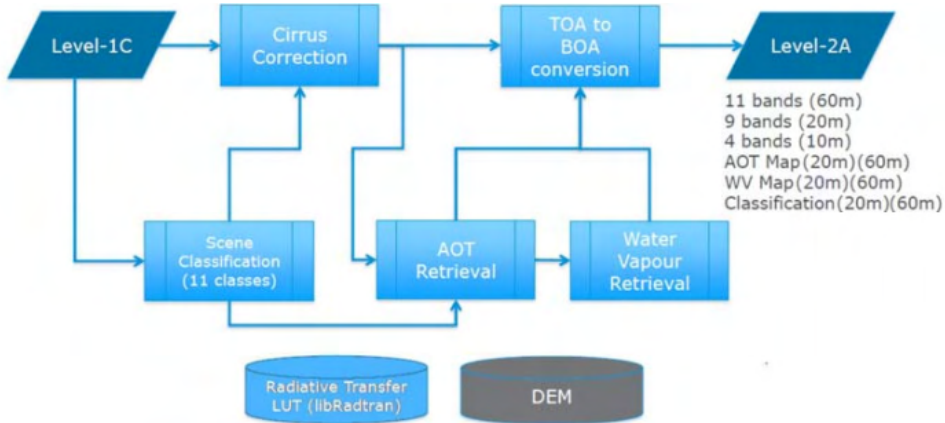


Figure 2.2: Processing of Sentinel images from level-1C to level-2A. Taken from [36].

Figure 2A similarly shows the pipeline from 1C to 2A images. The main product of the pipeline is to provide a Bottom-Of-Atmosphere (BOA) reflectance image compared to level 1C top-of-atmosphere(TOA). The key here is that the ortho-image BOA is calculated with correct reflectance. The extra outputs are as follows; Scene Classification(SC) map with cloud and snow probability, Aerosol Optical Thickness(AOT) map and a Water Vapour(WV) map. Naturally as the purpose of this project is to map coastal features the main focus will be on BOA 2A images.

2.2 Normalised Difference Water Index

The Normalised Difference Water Index(NDWI) is a widely used method, also in the Sen2Cor framework, for multi-spectral imaging when segmenting water[16, 26, 29, 32, 65, 70]. Since the early days of satellite imaging mapping bodies of water has been of particular interest, for environmental concerns or otherwise mapping has played a key role. Most of this work relies on the concept that wavelengths refract differently based on the matter composition of the material they hit. For example, NDWI concept was evolved from the Normalised Difference Vegetation Index(NDVI). The NDVI works on the simple principle that healthy vegetation reflects more near-infrared(NIR) and green light whilst absorbing more red and blue light. Similarly the NDWI shares the same basic concept to its algorithm. There are two NDWI algorithms derived from NDVI; one by Gao[6] to index water content in vegetation and secondly by McFeeters[37] to monitor water content in bodies of water. For the purposes for this research the second one is of interest, water detection relies on the phenomenon that water absorbs more visible to infrared wavelengths compared to other materials.

$$\frac{(X_{green} - x_{nir})}{(X_{green} + X_{nir})}$$

Although effective at suppressing most vegetation and land features the method would run into complication with built-up land noise. To combat this problem Xu proposed a Modified NDWI(MNDWI)[67]. Xu simply substituted the NIR for a Mid Infrared

Band(MIR), simple substitution of wavelengths without further processing seemed effective for the small regional data that was tested.

$$\frac{(X_{green} - x_{mir})}{(X_{green} + X_{mir})}$$

The solution proved to considerably suppress built-up noise however could not eliminate it, the method also provided added benefit of more subtle detection of differences in water quality. Du et al[15] produced a paper showing the results of MNDWI and NDWI on images from Sentinel 2 with different pan-sharpening techniques. The paper reinforces the evaluation that NDWI suppresses vegetation but can result in positive segments for centres of large built up areas. Whilst MNDWI suppresses the built up areas. The use of NDWI techniques can provide a fairly accurate representation of the waterline however they need adjustments and tweaking for each local region they map. One key example would be for mangrove mapping in India requires the combination of multiple indexes to achieve the desired affect[62]. The main disadvantage for spectral based approaches are that they do not contain any of the spatial features of the data; information regarding neighbouring pixels or variation of size and feature complexity. However the use of this technique would prove useful for extracting coastal locations.

2.3 Early Feature Extraction

If the spatial arrangement of the feature that needed extracting is known we can approach it from a knowledge based approach. An example of such an approach is taken by Chaudhuri and Samal [7]; bridges are surrounded by water on each side or that most roads that lead or expand over a bridge are of constant width and darker than their surroundings. Similarly for coastline extraction there is only one certainty that they occur at the edge of a water line, however the coastal land itself does not contain homogeneous characteristics as found in roads. To capture more characteristics of each individual feature could be taken by combining multiple feature extraction algorithms and applying a clustering or classifying algorithm. Huang et al [25] create a structural feature set (SFS) to tackle this issue; SFS is a combination of length width index and pixel shape index with new spatial measure of directional lines. The constructed SFS dataset is passed through a state vector machine to classify between the different features. Another paper that takes a similar is GENetic Imagery Exploitation (GENIE) [23]. GENIE takes multiple different primitive image processing operators as genes in a genetic algorithm. For different applications a different set of primitive operators can be taken. These genes are then used to build a genetic algorithm trained on labelled data for classifying other images. Although genetic algorithms are not as common today the approach starts heading towards more machine learning based structures to classification however still use non machine learning based approaches to feature extraction.

2.4 Feature Reduction using Machine Learning Approaches

The next iterative step towards machine learning techniques is using ML to extract the features directly from the multi-spectral image. Applying ML techniques results in more

non-linear features learned. The most common approach used to be principal component analysis (PCA)[48]. PCA identifies relations between features by eigen-decomposition on the covariance feature matrix. The resulting eigenvalues and vectors can be used to find the principal components of the data set, which features provide the most information about the image. Alternatively the eigenvalues and vectors can be used as their own feature regarding them as a filter[55]. This provides a way to reduce dimensionality whilst also extracting features where the most important information is only extracted. As the features are reduced the storage space required reduces and if the features is reduced to near two it becomes easier to visualise. However the algorithm has a high computational cost with the need for large amounts of memory to run[48]. In addition PCA applied globally on the dataset can remove local features present in the dataset.

2.5 Enhancing Low Resolution Images Imaging

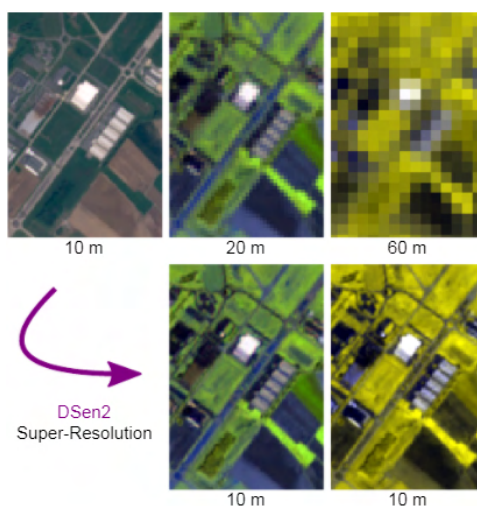


Figure 2.3: Example of enhancing low resolution using the Dsen2 framework. The 10m image is the pan-chromatic image and the first row shows the original images. The bottom row is the resulting process of pan-sharpening. Taken from [31].

This section briefly touches on the recorded spatial resolution difference in wavelengths and the process taken by many studies on rectifying the issue. Images from the Sentinel 2 mission come in three different resolutions, 10m, 20m and 60m however not all wavelengths are recorded at the highest resolution. The lower resolution, sample area per pixel, images do not contain as many wavelengths as the higher resolution images. Only four bands are recorded at 10m resolution whereas 20m and 60m have nine and eleven respectively. Having all bands at their highest spatial resolution allows for more data to train on and can be used if needed for algorithms such as MNDWI where the band required many only be recorded at low resolution. To overcome this lower resolution images, 60m or 20m, can be up-scaled to 10m creating a complete data cube at maximum resolution,figure 2.3[31]. To overcome this most commercial satellites, Landsat, SPOT, RapidEye and WorldView produce a panchromatic image band that combines the blue, green and red bands. In other words the panchromatic image contains the total light energy from the visible spectrum where each pixel is commonly represented in greyscale. As panchromatic sensors collect higher amounts of radiation, each image detects a higher

amount of intensity change per pixel resulting in a higher spatial domain. Each pixel represents a smaller area. In contrast multi-spectral images need to sample a larger region for each band as the amount of energy available is so low. This can be seen with Sentinel 2 data as only low wavelength bands are recorded in a higher spatial domain, sample area per pixel, as the amount of energy needed for information capture is lower than that of long wavelength bands. Sentinel 2 doesn't provide a panchromatic image however bands 2,3,4 (R,G,B) and 8 are provided in high spatial resolution so deriving a panchromatic image from those is possible.

The panchromatic image can be fused with the multi-spectral images to provide an image with the resolution of the panchromatic image and the spectral properties of the later image. This process is known as pan-sharpening. There are multiple methods that can be incorporated into the fusion process, with component substitution being the most common. In component substitution the multi-spectral image is projected into a new space containing spectral information and then substituted with the panchromatic image for spatial structure[59]. Then an inverse transform is applied to obtain the original space[63]. Well known component substitution methods are PCA[8, 45], Intensity Hue Saturation[47, 71], Brovey[14, 17, 22] or Gram-Schmidt(GS)[2, 30]. Another method to attack pan-sharpening is to apply a Bayesian model. By producing a statistical model to join the characteristics of the pan-sharpened resulting image and pan image[63]. Further non pan-sharpening models include the use of CNN's[31], naive interpolation and there are many others.

Thomas et al [59] highlight many problems that occur when apply pan-sharpening to an image. Firstly the acquisition time of both the Pan and multi-spectral images may not exact even if said to captured at the same time. This is evident in images with fast moving objects, such as planes. Secondly the position of the sun and other illumination factors could alter the size and orientation of shadows slightly. Most importantly spectral bandwidths reflect different depending on the matter or material and can cause extra noise in the fusion process as panchromatic image records at a different wavelength.

2.6 Artificial Intelligence Techniques

Artificial Intelligence (AI) is the most recent and promising technique for classifying images as a result of extracting high level non linear complex features. AI techniques are built on the concept of the neuron pathways and synapses found in animal brains. Neurons in the brain are connected in a vast network with each neuron linking to others. Synapses between the neurons allows for signals to be transmitted to each other. With this basic understanding an artificial Neural Network (NN) is created; multiple layers of neurons each connected fully to each other. An artificial network is comprised of an input layer, few or in most cases many hidden layers and finally an output layer.

2.6.1 Basic Feed Forward NN [43]

In a NN we expect the input values to activate certain neurons in the next layer which in turn activates a specific pattern of neurons in the next layer and so on. The final layer which normally represents each class has the highest value in the neuron for the class of the image. A simple example being if a image of the digit 5 is entered the 5th neuron representing the classification for images representing a 5 at the end of the NN should contain the highest value. Before getting into the specifics it is important to understand

that NN are layered as during training the expectation is that each layer learns more higher level features compared to the previous layer.

Each neuron in a fully connected network has edges or weights connected to it from each neuron in the previous layer. The neuron takes the weight W and value A from the previous neuron and multiplies them together, then continues to do so for each neuron in the previous layer whilst summing them together. $w_1a_1 + w_2a_2 + \dots + w_na_n$. The result of this process can be a number that varies in range. Therefore a function to map it between two values is used. For example the sigmoid function. There is an extra neuron in each layer termed the bias component. The bias is added to each summation of weights which allows for the neuron to only activate when the information is meaningful. For example if the neuron consistently gets a value of 10 even when it has no meaningful information to contribute from the previous input we might add a bias of negative 10. This is a very simple example of how information is fed forward in a NN.

The next step in a NN is to try and minimise the error of wrong predictions. To do so we train the network on images for which we know the classification. We can quantify the result error by finding the difference in the predicted and actual result, known as a cost function, C . A simple cost function may be

$$C = \sum_{j=0}^{n_{(L-1)}} (a_j^{(L)} - y_j)^2$$

where L is the layer, in this case the final layer, a is the activation and j denoting the neuron in the layer. In reality the entire combination of weights through the entire network are what give the resulting classification so the cost function incorporates all of them. To optimise the network we want to minimise the cost function by using back propagation.

$$\frac{\partial C_0}{\partial a_k^{(L-1)}} = \sum_{j=0}^{n_{L-1}} \frac{\partial z_j^{(L)}}{\partial a_k^{(L-1)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial C_0}{\partial a_j^{(L)}}$$

Each layer starting from the end of the network moves backwards through the network changing the weights. The weight changes are calculated in proportion to the corresponding weights and by how much each neuron needs to change finding the most rapid decrease to the cost. This can be seen in the equation above; the aim to find how sensitive the cost is regarding the change in weight for the previous layer k . z is the weight before activation including the bias term. Then the process is continued for each layer for all the weights and bias terms. Note the equation is somewhat amended for finding the bias term but it is simply find the ratio between the partial derivative of the cost and bias terms in the previous layer.

There are a number early papers that utilise NN to classify remote sensing data; Heerman et al 1992 [24], Bischof et al 1992 [5], Tan et al 2011 [56]. These papers similar to modern approaches focus their data on particular regions. They also focus on classifying the main types of land cover i.e. water, forest, urban, agriculture etc. hence they provide a good accuracy. Heerman found that splitting the image up into smaller sets would improve the processing time as each pixel is calculated independently from its neighbours [24]. Bischof who implemented smaller sets by using a 5 x 5 input window found that not only was it faster but provided textural information within the NN ultimately resulting in higher accuracy for land based classification[5]. This is certainly not the end all of NN in research they evolve into different architectures which are explored later. This section

explores briefly how they work so a basic understanding is established for interpreting these other architectures.

2.7 CNN's

As this project pertains to images, satellite images just contain more bands than RGB, naturally Convolution Neural Networks(CNNs) come to mind. CNNs, unlike NN, keep spatially local data together with the use of kernels. There are many papers used to classify land use and building in literature [9, 72]. Much like NN there are multiple uses for CNNs and different architectures in research. This section explains the main differences from NN to from a base understanding. CNNs are built of three main layers; convolution, pooling and fully connected layers [69].

2.7.1 Convolution Layer, Pooling and Fully Connected Layers

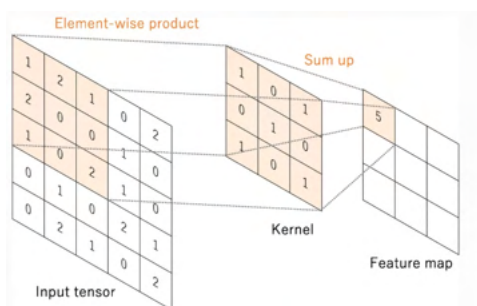


Figure 2.4: Example of convolution. Taken from [69].

The convolution layer is a linear process in which features are extracted. A kernel is applied to the input array of numbers, input tensor, to create a feature map (See Figure 2.4). The kernel simply applies an element-wise dot product at each location in the input tensor to create a new value in the feature map that relates to the original position in the tensor image. The number of computations is drastically reduced compared to a fully connected neural network layer as the new neuron is only connected to the surrounding pixels not the entire tensor. The kernel size can be changed but the most common sizes are 3×3 , 5×5 and 7×7 . In a typical layer there are multiple kernels that are applied to the same tensor to create multiple feature maps, each kernel therefore being a different feature extractor. The kernels are the underlying feature extraction force in a CNN and therefore are the parameters optimised during back propagation. As shown in Figure 2.4 the resulting feature map is only 3×3 compared to the 5×5 tensor. This is a result of the kernel being unable to center on the outer edge of the tensor. To rectify this we can create an outer edge of 0's on the tensor, this is called zero padding. Without zero padding each convolution following will create a smaller and smaller images.

That being said it is beneficial to reduce the input size, discussed in the pooling section. A way to achieve down sampling within the convolutional process is to only apply the kernel at intervals. For example skipping one pixel between each kernel application this is called the stride. Applying a stride reduces the computational time required however can result in the loss of information.

The final step to convolution is the non linear activation. The most common function is ReLU , $f(x) = \max(0, x)$. Due to ReLU's simplicity it is faster to train and has shown to achieve better performance.

If the input tensor is kept at the same size throughout the CNN there is reduced chance of learning any new features, the dimensionality stays the same. Therefore pooling is employed to reduce the dimensionality and introduce translational, rotational invariance. Pooling traverses the image in the same method as a kernel, with a filter size, padding and stride parameter. The resulting image is smaller than the previous with the new pixel summarising the information in the previous filter region. This can be achieved through multiple different pooling methods, max, min, average and global.

At the end of a CNN the result is flattened, reshaped to a vector, and like neural network fully connected to the next layer. This layer then feeds into the final output which uses a softmax activation function to get probabilities for each class. This is a very basic summary of how CNN's extract information using convolution layers, extract higher level features by pooling and subsequent convolution and finally creating a fully connected layer to reach a classification.

2.8 Stacked and Sparse Autoencoders

2.8.1 Sparse Auto Encoders

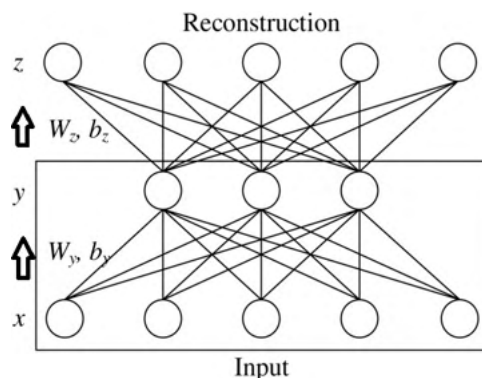


Figure 2.5: Example of an Auto Encoder for a NN. The encoder is shown by the box surrounding the last two layers, the decoder is the first layer. The input is fed from bottom to top. In this example the 5 inputs are reduced to 3 latent features. Taken from [10].

The use of a CNN on its own is not ideal for the purposes of this paper. The main flaw resides in the fact that pre labelled images are needed to effectively train and extract features. This is where auto-encoders (AE), a specific architecture for NN, plays a role. AE can be used on unlabelled data as they are simply trying to encode the image into features and decode them back into the original image. In other words the encoder creates features from the image and the decoder generates an image from the features. As such an AE is a form of generative model. The simplest form of an AE, see figure2.5, with a single hidden layer with the hopes that the hidden layer can facilitate subsequent learning[51]. As the encoder and decoder are simply the inverse of each other the following holds true for the weights , W , $W_y = W'_z = W$ [10]. That being said the decoder is trained without knowing the encoders weights. The goal of this architecture is to minimise the error

between the input and output. The result we are interested in is not the final image but the latent vectors or features trained by the encoder. If the resulting encoder features are able to represent the original data where the decoder is able to recreate the data from nothing but the features we know we have features that accurately convey the image.

2.8.2 Stacked Auto Encoders

An AE is only one layer deep and therefore is unable to provide more higher level features that we would normally find in the final layers of CNN's. To attain higher level features we can stack more hidden layers to the encoder and decoder to match, this is a Stacked Auto Encoder (SAE). The same training concepts still apply from AE to SAE, the minimisation of error between the original image and data compared to the decoders generated image is the goal. This method of training is used by multiple papers on hyper-spectral image land classification [10, 42]. Chen et al reduce the original images into patches and then apply PCA reduction to get the first 7 dimensions on the patch to feed into the SAE[10]. PCA is used as hyper-spectral data has many more channels compared to multi-spectral. Even with a reduction in spectral channels the paper notes that the SAE took a substantial time to train compared to SVM or other methods, that being said it was much quicker to test. Both papers conclude that using SAE result in useful features extracted compared to SVM, PCA, KPCA or NMF.

Ali et al published a paper that directly influences this research[3]. They looked to find a way to label time series data whilst also find repeating patterns and outliers. The main approach in the paper uses Deep Convolutional SAE (DCAE) to extract the features which are reduced in dimension for producing an interactive tool. To cope with time series data they produce a sliding window approach to produce a matrix; each row has a consecutive time step and each column has the next data entry depending on the stride. This does cause a overlap of data between data points but the paper states it is useful for avoiding lost data and for a smooth transition between time steps. If this method was amended for images the result would be a sliding window with an extra dimension stacking the relevant time steps. These matrices are then compiled for each image patch and feed through the DCAE. The result from the DCAE is fed through either PCA , Distributed Stochastic Neighbour Embedding (t-SNE)[35], Uniform Manifold Approximation and Projection (UMAP)[38] to project into a two dimensional space. The two dimensional space is required to create an interactive tool that can facilitate highlighting and exploring the features space. The paper provided multiple case studies of applications in medicine and biology which show major features can be highlighted easily via the scatter plot of the two dimensional features extracted from the DCAE. However the example cases used do not contain as many varying features as a global coastline would host, therefore is may be questionable if the DCAE could be effectively reduced to two dimensions. In addition if two dimensions are not conclusive how could two or more dimensions be mapped into a tool without overcomplicating the usability. Although the paper does not apply the method to multi spectral images it could provide interesting results.

2.9 Recurrent Convolutional Neural Networks

Recurrent neural networks are the next iteration in the progress of NN. Recurrent Networks remove the limitation of only passing through a feed forward network once. As

this network now passes images in a linear fashion there are more application based in the temporal domain than just spatial. Many of the current papers use RCNN to find and understand change detection between two time step images. Mou et al propose a CNN to find spatial features and use the recurrent architecture to analyse temporal dependence[40]. Multiple other papers focus on change detection in multispectral images[21, 34, 41]. Most uses also break the images into patches and regard them as patch-based RCNN (PB-RCNN), Sharma et al produce a PB-RNN that correctly classified land cover with 97.21% accuracy compared to NN with only 64.74% accuracy with the speculation that CNN could prove to be better in future work[52]. It is also interesting to note that the patch based method was more accurate then pixel based methods. As the next time step image used is key to training and cannot be avoided if there is too much cloud coverage, some papers choose to avoid training on the areas within the image with clouds. The main disadvantage of RNN is their long training time as each time step must be calculated consecutively therefore parallelisation can not be used. Furthermore RNNs suffer from vanishing gradients, as each time step increases the problem is amplified during back propagation.

2.10 Long Short Term Memory

Long Short Term Memory, or LSTM, are an architecture based on RNN and thus are able to handle spatial temporal data. LSTM's were developed to overcome the vanishing gradient problem produced by a RNN , however they encounter a similar problem when used on a long chain of temporal data. The vanishing gradient problem refers to a gradient becoming too small, therefore not contributing as much to learning, when back propagating. This can be understood as RNN having short term memory as information in earlier cells does not affect the later stages as effectively as a LSTM. The architecture of an LSTM consists of multiple cells linked to each other and each successive cell is given the next time step image and the hidden state. Each cell dictates what information is retained and passed onto the next cell through multiple gates; input, output and forget gate. The input gate dictates the information to retain from the previous cell, forget gate controls the information to remain within the cell and finally the output gate decides on what information to pass onto the next cell. The key here is that a hidden state passed from each cell acts as the long term memory and the cell state which is computed using the hidden state acts as short term memory. As with most research conducted with multi temporal images there are a multitude of papers conducted on land classification using LSTM's[18, 28, 49]. LSTMs unfortunately, due to their many gates and the passing of a hidden state, have remarkably more parameters then RNNs[40]. That being said LSTM's benefit in higher accuracy as well as being able to identify cloud coverage or ignore it more easily by removing it via the input gate[40]. LSTMs have also been used in pan-sharpening[58, 61].

2.11 Transformers

Transformers were introduced in 2017 by Vaswani et al which makes them more state of the art compared to other architectures[60]. The original paper focuses on translating between languages and therefore it is the example I will use to explain transformers, afterwards moving onto papers that use similar techniques on multispectral images. As

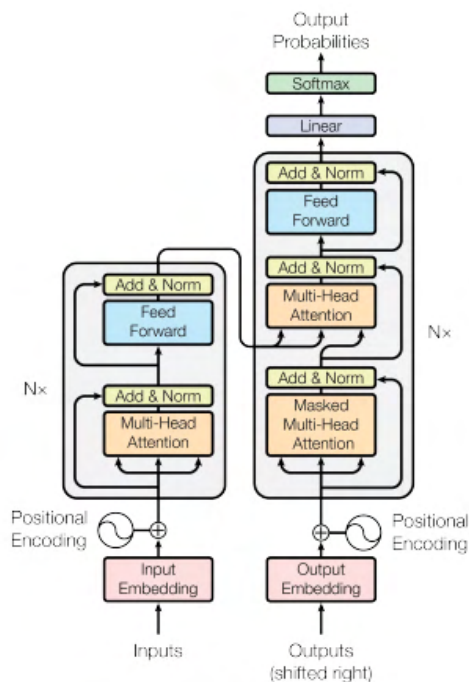


Figure 2.6: Transformer Architecture. Taken from [60].

shown in figure 2.6 we start the process by input embedding. Input embedding is the mapping of a word to a vector where similar words are closer together, the vector space that words are mapped to is called the embedding space. Afterwards we apply a position encoding, depending on where the words appear in the sentence. This ensure that there is context for each word regarding the meaning and position. Next the main and key part of transformers is the multi-head attention. Attention draws focus to how relevant each word is in the sentence compared to each other word in the sentence; capturing contextual relationship between each word in a sentence. As each word may focus on itself there are multiple attention vectors averaged as the goal here is to draw context between words, hence the name multi-head attention. These attention vectors are then passed into the feed forward network, as each vector contains the context and positioning independent of the previous we can parallelise the process. The entire block is known as the encoder and the output is a set of encoded vectors for every word. Similarly the decoder starts by embedding and positional encoding the other language words. Next is a masked multi-head attention layer. To be able to learn the next word in translation we want to use the entirety of the first language sentence but only the previous words of the new language otherwise no learning would take place, that is why there is a masked layer. The information from the encoder and masked attention is then fed to another attention layer that produces similar attention values for each word in both languages. Finally we pass the information through a NN which transitions to a linear layer with all the words in the second language and a softmax layer to give each word a probability. The biggest advantage of this method is the computation complexity is drastically reduced with parallelisation and the paper found it to be more accurate than state of the art techniques at the time[60].

As this method was developed for natural language processing the architectures used for multi-spectral classification are amended versions. There are a multitude of versions that exist but the key factor that they all aim to integrate is the attention

mechanism. Rubwurm and Korner use only the encoder from the transformer for crop classification[50]. Naturally they remove the word embedding and keep the positional encoding. They introduce L transformer blocks consisting of a multi-head self attention and multiple dense layers of a feed forward NN. In the paper, 8 such L blocks are used with the architecture ending in a max-pooling and softmax layer. When evaluating the self-attention scores in the model they found that they focus on distinct events, regardless of the input of the input time series data. They found that the each layer focus on different parts of the time series data and due to all of this the model is regarded key to suppressing "non-classification-relevant cloudy observations". Furthermore they concluded that there is an increase in separability between classes for each successive attention layer. Finally comparing the architecture to convolutional models Transformer and LSTMs produced the best results showing promise in future work. More recent papers using multi-model self attention networks provide a promising future for self-attention mechanisms[19, 64, 68].

2.12 Towards Human Centered and Responsible Innovation

This research is conducted in partnership with the Engineering and Physical Sciences Council(ESPRC). One of the motivations behind this project is to explore and use responsible innovation ideals and conduct it with a Human Centred Design(HCD) approach. The key ideas behind both standards is to find a more suitable and sustainable approach to conducting research and developing tools.

2.12.1 Human Centred Design

In a Human Centred Design (HCD), we expect the human being to handle the qualitative subjective judgements and the machine the quantitative elements forming a symbiotic relationship. Furthermore the design should support human skill and ingenuity rather than solely focusing on machines trying to objectivise that knowledge[12]. This paper in a certain regards already follows that philosophy; in the goal to create a tool that utilises professional users skill to label feature extracted by a machine.

2.12.2 Responsible Innovation

There are many responsible innovation frameworks for software development. As this project looks to implement AI methods the Artificial Intelligence and Public Standards by the committee on public standards and life will be used[44]. They have evaluated four key principle Fairness, Accountability, Sustainability and Transparency. Each principle will be explored in this section apart from fairness as this is for processing social or demographic data pertaining to features of human subjects.

Accountability

The committee discuss who is ultimately accountable for the decisions that an AI makes. They recommend that AI should be "human-centric, uphold human agency and respect human autonomy". In essence the AI should be a support tool in decision making to

achieve full human responsibility. This principle coincides very close to HCD and as discussed before this project specification and research has been closely designed to achieve that goal.

Sustainability

The end goal of this research is to geo-generalisable model that can track features of coastline around the world. With the temporal aspect this model would have to constantly update to accommodate new features and to be reliable and robust methods that can do so should be taken into consideration. In addition being a critical system for shipping routes the model should always stay to a high level of accuracy with a measure to indicate if not.

Transparency

Transparency could be considered the most important principle for this research. The end user must be able to discern why the model has labelled a specific piece of coastline. The understanding of what the model is doing would allow the end user to accommodate or look to change to model in different scenarios. This specific goal of understanding black box models is its own major section in research. Black box models are opaque with the decisions process made between the input and output, which pertains to most AI models. Algorithms like Random Input Sampling for Explanation(RISE) produce a saliency map by perturbing an input image with multiple masks to see what features affect each classification[46]. RISE falls into the category of model agnostic algorithms as only the input needed to produce an understanding. Other methods such as Layer wise Relevance Propagation(LRP) looks to achieve a similar goal however uses a form of back-propagation on the model itself to achieve an understanding[4].

Chapter 3

Pipeline Architecture

There are three main parts to the pipeline; preprocessing the data, reducing the data to latent vectors via AI methods, and finally reducing the latent vectors to two dimensions. The techniques and algorithms presented in the literature review are the main methods for dimensionality reduction that will be explored as part of the pipeline.

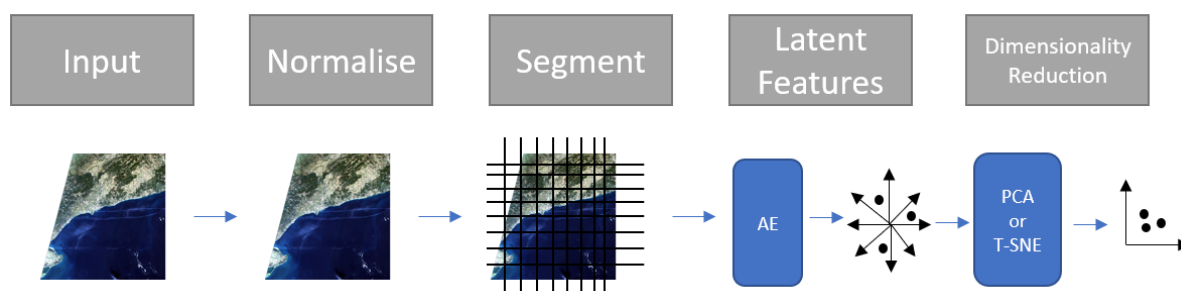


Figure 3.1: Pipeline of the process of initial input to 2D plot using an AE.

3.1 Exploring the Dataset and Preprocessing

This segment will briefly revisit the different bands and resolutions that the data is comprised off; whilst discussing the limitations that come with such large datasets. As discussed before Sentinel 2A, data comes in 3 varying spatial resolutions; 10,20 and 60m represented per pixel. As each wavelength needs a different length of time to be recorded by each sensor we find that smaller bands are recorded in a higher resolution. The smaller bands carry more information as their wavelengths are so short. In the 10m resolution contains the following bands; blue(458-523nm), Green (543-578nm), red(650-680nm) and NIR (785-899nm). In addition to the 10m resolution bands 20m contains 3 red edge bands (698-713nm, 733-748nm, 773-793nm), 2 SWIR bands (1565-1655nm, 2100-2280nm) and NIR narrow (855-875nm). The lowest resolution, 60m, contains all the bands available in both 10m and 20m resolutions.

The data ranges play a key role in deciding the methods and algorithms used when preprocessing the data or constructing the AI architectures. As shown in figure 3.2 even though 50% of the data is within a tight range, as shown by the blue bar, the maximum values can be considerably larger. This poses a challenge when looking to normalise the data into a range that an AI architecture could run optimally on. Normally a range of zero

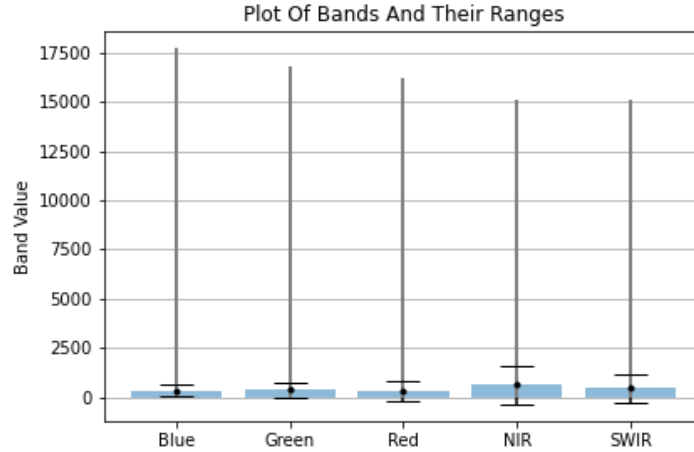


Figure 3.2: Simple bar chart showing the different Bands and their corresponding values as minimum, maximum ,mean and standard deviation.

to one or negative one to one is used. The use of normalising boasts two main benefits, firstly the calculation is performed on smaller numbers allowing for faster computation and secondly activation function commonly lay within those ranges allowing the values to propagate further into the network. Considering these factors normalising would be ideal however doing so with such large ranges present in the data creates very small values, below 0.001, once normalised. This poses a problem as such small values do not propagate efficiently through the model. Likewise the cost function has to be able to handle extreme values, for example Mean Squared Error (MSE) on two values in decimal points would produce an even smaller value. Therefore a more optimal approach would be to standardise the dataset by calculating the mean and associating each value with the standard deviation from that mean. Standardisation still produces extreme values however less so than normalising as the data is no longer trying to squeeze the dataset into the range from negative one to one. The more extreme values from standardising would hope to be captured as a feature in the first layer of the AI architecture by passing it through a convolution before any activation layer.

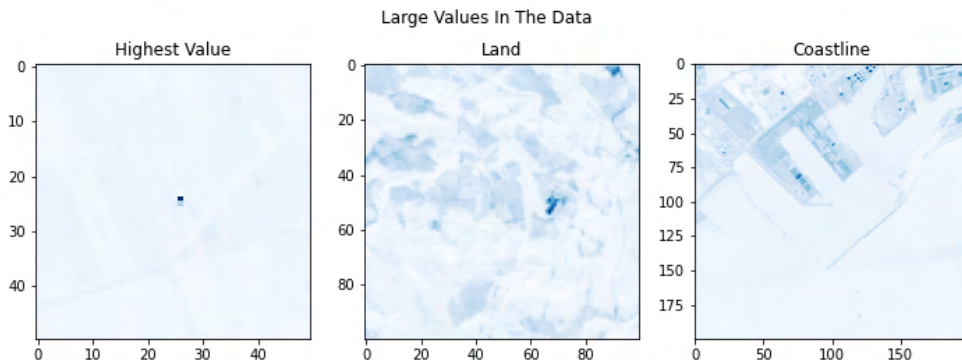


Figure 3.3: Pixels with large values from the blue spectral band with 20m resolution.

The second option would be to consider the large data values as outliers and restrict the input values to a suitable range. Most of the large values are from commercial areas where reflectance is high. As shown in figure 3.3 the highest value is not a discernible

location in a field however there are other values similar to it in both land patches and near the coastline where there are built up areas. Important to note that each band has shared pixel with large values. The information gained from these pixels could be attributed to learning features where there are large amount of buildings. By reducing their values it may affect the features learnt, however, would optimise the training as mentioned before. It would be interesting to compare both results with and without large values to see the effect for each model.

The 10m resolution would have been perfect as it contains the most information for any value. That being said, with the limitation on hardware only having 16GB only 4 images can be loaded into memory. Therefore the main resolutions used are 20m and 60m. That being said there are further limitation, such as producing patches with a small stride length. When utilising 20m resolution images having a patch length of 60 pixels and a stride of 30 pixels unfortunately easily hits the edge of memory capacity. The main datasets used in this project therefore use either one or more 60m resolution images depending on the stride and 20 or less 60m resolution images depending on the stride.

The choice of using patches is mainly stems from previous literature and hardware limitations currently present. Patches are the main method used when detecting crops, vegetation or other classification problems present in the literature as they show better performance and allow for sub-regions to be classified. Even though coastline detection for a critical system needs to be somewhat pixel by pixel classification, this project will look to see if building on the methods used could be beneficial. In addition the use of patches for this project increases the overall available data to train on and gives the basis for detecting coastline either present or not in a patch. Patch size also is key as a CNN would learn on the pixels surrounding neighbour hood based on the masks size. If the patch is too small the CNN would not be able to recognise the information. Likewise if the patch is to large it may incorporate too many features causing more complicated models.

3.2 Different Models

Different models have different interactions and limitations when considering the latent features extracted. This section will briefly explore the different issue with implementation and how each latent feature is extracted. We want AI models to extract latent features as simple processes such as PCA or early feature extraction shown in literature does not produce complex features. The papers referenced here are centred towards the techniques rather than the domain of multi-spectral imaging.

3.2.1 Auto-Encoders

AEs are fairly simple in explanation and as explored in the literature the goal is to create and encoder to extract latent features and a decoder to generate images from those features. The cost function is the difference between the two values inputed to the encoder and output of the decoder. As for the specific implementation many parameters need to be considered such as each convolutional layer; the kernel size, number of kernels or activation. The number of kernels dictates how many different filters the network is to learn and optimise in that particular layer. The kernel size dictates the neighbouring pixels considered, having a large kernel size increases the information of neighbouring

pixels recorded. The main process that ties the layers to each other is max-pooling for the encoder; taking the maximum value given a kernel size and therefore reducing the size of the input. Max-pooling introduces translation invariant features whilst reducing the image size. At each successive layer we want to reduce the kernel size and features as max-pooling starts to converge these features into higher level representations. Most importantly reducing to higher level features allows for a lower number of dense layer neurons. For example, take the final layer of a encoder to have 6 kernels each with a size of $3 * 3$ this would result in $6 * 3 * 3 = 54$ once flattened. The decoder takes a similar approach however mirrors the encoder architecture. The goal would be to tweak each of these values to try and find the least number of neurons in the dense layer whilst keeping the highest accuracy. This method ensures we have features that strongly represent the data whilst keeping the number of features low to aid in dimensionality reduction. The latent features would be the values from the dense layer. As the decoder layer is a mirror image of the encoder to output the same dimensionality as the input there are limits to the neurons in the dense layer. Taking the example from before if the last layer in the encoder is $6 * 3 * 3$ the first layer in the decoder would also have to be $6 * 3 * 3$ which means that the dense layer would have to be of a multiple of nine as $3 * 3 = 9$ with arbitrary feature maps. From there we can extract the same amount of feature maps as before which was six.

3.2.2 LSTM

Quickly reviewing the process of an LSTM, there are two main outputs the hidden state and the cell state that are passed from each LSTM cell. LSTMs are more complex and better form of RNN which are used on time series data due to their ability to hold information from more previous time states as they have a hidden state passed from each cell. The quickest implementation of such a method would be to have the same number of parameter input as the image with channels. For example if the patch was $30 * 30$ with 5 bands we would have a $30 * 30 * 5 = 4500$ input parameters as a classical LSTM expects a 1D input. As the nature of an LSTM is towards time series prediction the patches would be organised by N features in T time steps. In total if there was 3 time steps for each image the input would pass three image patches as one sample.

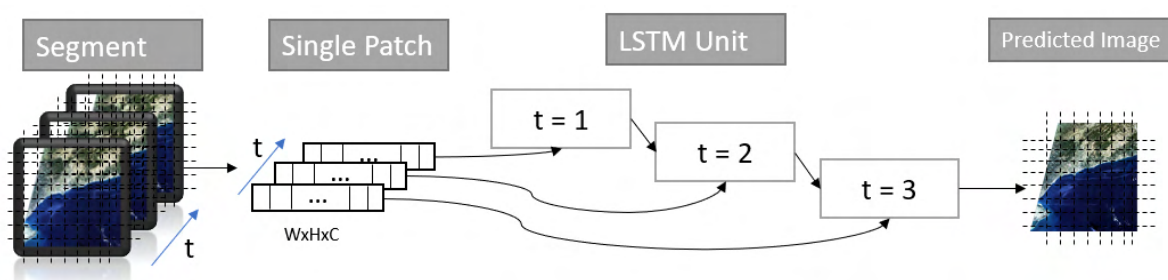


Figure 3.4: A simple LSTM that takes segmented patches at three time steps t to predict the final time step t_4 .

There are multiple different methods for extracting the latent features with LSTMs. The first being inputting a series of time step images and allowing the LSTM to predict the next image in the sequence, see figure 3.4. The idea being that if the next image predicted is accurate enough there is an assumption that the architecture has learnt

significant features. This process is extremely simple and naive however explains the methodology behind an LSTM.

The more complex image prediction methods take into account the need for a two dimensional input, as stated before LSTMs are more adept at 1D input and the input has to therefore be flattened. Convolutional LSTM or ConvLSTM is designed to overcome this proposed by Shi et al[66]. In a ConvLSTM the matrix multiplication normally present in an LSTM cell is replaced with a convolution operation. This convolution operation preserves the input dimension rather than flattening it to one dimension. This has already been used in research for predicting satellite images[39] or classification[27, 57].

Most of these approaches stack multiple ConvLSTM layers. This approach much like a NN or CNN allows for the output from one layer to allow deeper feature extraction in the next layer. However unlike taking the output of one layer the multiple time steps make the process a little more complicated. Each time step output is also an input to the corresponding time step in the next layer. With the addition of the final output of each layer passed to the first in the next, as shown in figure 3.5.

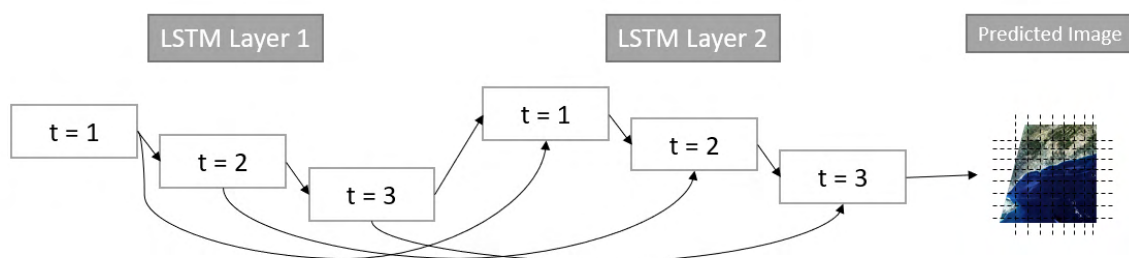


Figure 3.5: LSTM stacked with two layers. Each time step output is also an input to the corresponding time step in the next layer. With the addition of the final output of each layer passed to the first in the next.

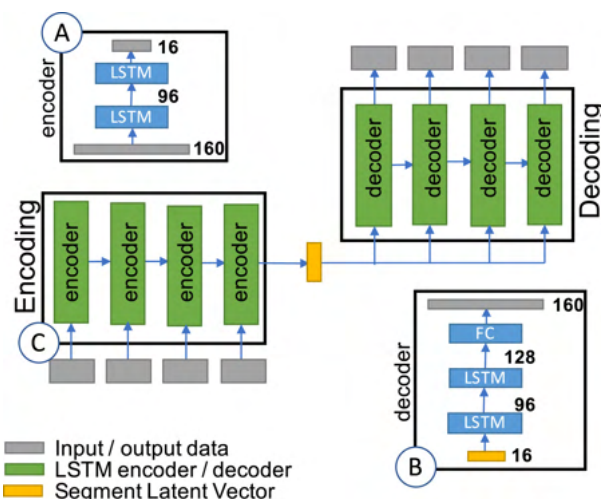


Figure 3.6: An example of an AE that utilises LSTMs to produce the latent features. Taken from [54]

The ability to stack layers provides the creation of more complex architectures such as a AE with LSTM units. Shen et al do precisely that by producing a more complex model to find latent features of traditional Chinese music, with the intent to map into

a two dimensional interface[54]. They apply a LSTM encoder and decoder model with stacked layers; the two layers reduce the features by limiting the output of the cell state at each successive layer shown in figure 3.6. The first LSTM layer takes 160 input and produces 96 and likewise the final layer takes 96 and produces 16. The final 16 outputs are the latent features; notice that the input sequence for the final layer in the encoder has not fed each time step sequence however only the final output. The decoder mirrors the encoder as seen before with AE.

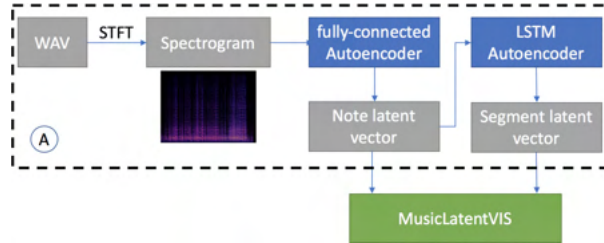


Figure 3.7: The architecture of MusicLatentVis. An AE and LSTM AE are used in unison to produce the final latent vector space. The LSTM AE further reduces the latent features produced by the AE. Taken from [54]

The full solution in the paper also utilises an fully connected AE wrapped around the LSTM AE, see figure 3.7. The purposes of the AE is to reduce the initial $501 * 1$ to the latent features that are parsed to the LSTM AE. Also interestingly both latent vector from the AE and LSTM is combined to produce the final feature space. This method not only follows the pipeline proposed by this project but also allows training without the need for pre-labelled data. Adapting this solution to fit the current data would not require a ConvLSTM as the latent features are represented by vectors in the dense layer of the AE.

3.2.3 Transformers

“Attention is all you need”![60]. The transformers main component is attention and this segment looks at the different architectures present that incorporate this feature. As attention is about embedding the data into a new feature space therefore the architecture for most models remains similar to the current approaches. The process of adding a multi-head attention layer to each layer of previous methods could be the simplest implementation including attention into the network. This approach is taken by Choi et al where the AE architecture has a multi-head attention mask before each layer in the architecture apart from the dense layer[11]. The main feature that changes is the self attention is *relative* as the domain is for melody and performance of music sequences. *Relative* self attention is described by Shaw et al however they note that for convolutional architectures positional encoding is inherent with a kernel however still benefit from positional attention encoding[53] as shown by Gehring et al [20].

Gehring proposed a encoder decoder architecture with the following position embeddings[20]. Firstly the input is embed into a distributional space, input $\mathbf{x} = (x_1, \dots, x_n)$ to distributional space $\mathbf{w} = (w_1, \dots, w_m)$. The absolute position is also embedded $\mathbf{p} = (p_1, \dots, p_n)$ with both combined being $\mathbf{e} = ((w_1 + p_1) + \dots + (w_n + p_n))$. In terms of the current papers problem space the positional embedding would allow for smaller patches as each has the relative absolute position to each other; the end goal being a pixel by pixel identification and feature extraction. Therefore we would not have to worry about losing

positional information because the convolutional kernel is too small, however the textural information would be lost as that is contained within the kernel. There are even more complicated auto-encoder models present in the literature that could warrant exploring in future work[33].

3.3 Reducing Latent Features

Even though using the AI models described above to extract features, there is still a need to reduce further to two dimensional plots for exploration. That is why this segment looks at different methods to reduce the dimensionality further.

There are two main methods to explore when reducing features, the first being Principle Component Analysis(PCA)[1] and the second being T-distributed Stochastic Neighbour Embedding(T-SNE)[35]. Both methods aim to transform and project the data into a n-dimensional space, in this case n is two. Not only does this reduce the complexity of analysing multiple features but also allows for the creation of a tool that is easily interpretable by humans.

PCA takes the covariance matrix of the entire dataset showing the positive or negative trends of each feature compared to one another. Secondly eigenvalues and eigenvectors are calculated based on the covariance, eigenvectors define the direction of the axis of a particular feature. The eigenvalue is the factor at which the eigenvector is scaled of the spread of the data along the axis. If the largest two eigenvectors are chosen then we have two axis that incorporate the largest spread among both axis. Simply from here we can transform our original data into the new two dimensional space. The produced result in simple terms is to keep dissimilar points as far apart as possible. However PCA is a linear method and has therefore disadvantages in this use case as latent features are in nature non-linear.

$$p_{i|j} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)} \quad (1)$$

$$q_{i|j} = \frac{\exp(-\|y_i - y_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2/2\sigma_i^2)} \quad (2)$$

$$q_{i|j} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq i} \exp(1 + \|y_i - y_k\|^2)^{-1}} \quad (3)$$

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1} \quad (4)$$

T-SNE on the other hand is a non linear method that takes into account the local relationships between two points using probabilities. Firstly in the original high dimensionality, latent features, we want to find the probability of each point to its neighbours. This is shown by equation (1) above. The probability is fit to a Gaussian distribution the i in σ_i denotes the perplexity, the amount of neighbouring pixels to compute. Limiting the neighbours negates some points from having disproportionate distribution. Secondly to map the points into the lower dimensionality, the two dimensional interactive plot space, the points are once again mapped to a distribution. Equation (2) is a Gaussian version of the high distribution used for the lower feature space. It is not used in by t-SNE as it squeezes the points towards the end of the distribution. This squeezing is

more apparent in the lower distribution as the space of points is squashed from the high dimensionality. Instead equation (3) is used a Student-t distribution with one degree of freedom. Finally equation (4) shows the cost function as the method uses gradient decent to settle the data points into an optimal space. The downside to this method is that if new data was introduced the whole process would have to be run again as there is no way to map new data into the same space. As there are so many computations the method would take significantly longer than PCA to run.

Chapter 4

Experimental Results



Figure 4.1: Image of the area used in testing. The area includes the city of Tarragona and the surrounding area from the east coast of Spain. Sentinel granule number 31TCF.

Only one granule, or tile, has been used for the experimental results. The processing power and hardware requirements limits the data used and due to Coronavirus access to resources has been limited. The tiles used is from the east coast of Spain. The choice was made mainly as the region has one of the lowest cloud coverages year round. Additionally the patch coastline has varying features including a coastal city and harbour.

The main method for testing has also been limited to using AE as more complicated models proved to be too resource intensive. The results look to find a preliminary view of the pipeline and what research questions arise.

4.1 AutoEncoders Model

For the autoencoder using the full range of bands in the data, 20m resolution, the following architecture,figure 4.2, produced interesting preliminary results. A single image from the 20m band was used. The encoder reduces the features down to 300 in the dense layer. The max pooling has to be chosen carefully to result in integer numbers, which in a usual CNN is not a problem, as during the up-sampling the data needs to be able to retain the shape. The use of reshaping the features after each layer in the decoder could be used to ensure the correct dimensions however that would allow for some loss of information with blank pixels introduced at each reshape. The patch size was specifically chosen to be just large enough to identify the features by eye for understanding the results when

| Layer (type) | Output Shape | Param # |
|-------------------------------|---------------------|---------|
| input_1 (InputLayer) | (None, 45, 45, 5) | 0 |
| conv2d_1 (Conv2D) | (None, 45, 45, 128) | 64128 |
| max_pooling2d_1 (MaxPooling2) | (None, 15, 15, 128) | 0 |
| conv2d_2 (Conv2D) | (None, 15, 15, 64) | 204864 |
| max_pooling2d_2 (MaxPooling2) | (None, 5, 5, 64) | 0 |
| conv2d_3 (Conv2D) | (None, 5, 5, 32) | 51232 |
| flatten_1 (Flatten) | (None, 800) | 0 |
| dense_1 (Dense) | (None, 300) | 240300 |
| reshape_1 (Reshape) | (None, 5, 5, 12) | 0 |
| conv2d_4 (Conv2D) | (None, 5, 5, 32) | 9632 |
| up_sampling2d_1 (UpSampling2) | (None, 15, 15, 32) | 0 |
| conv2d_5 (Conv2D) | (None, 15, 15, 64) | 51264 |
| up_sampling2d_2 (UpSampling2) | (None, 45, 45, 64) | 0 |
| conv2d_6 (Conv2D) | (None, 45, 45, 128) | 819328 |
| conv2d_7 (Conv2D) | (None, 45, 45, 5) | 64005 |
| Total params: 1,504,753 | | |
| Trainable params: 1,504,753 | | |
| Non-trainable params: 0 | | |

Figure 4.2: AE Architecture for patches size 45x45 with 5 channels.

comparing within the two dimension plot. The second reason for larger patches was once again due to hardware limitation. Having smaller patches with stride greatly increase the memory requirements, however smaller patches is something that the model will work towards in the future.

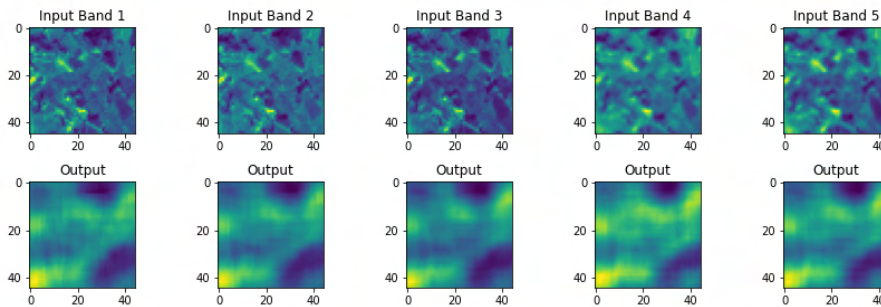


Figure 4.3: A single patch showing the input into the encoder(top row) and output(bottom row) for each band.

The AE is not fully optimised, figure 4.3, and could in the future produce much more high fidelity images given longer training times or more complex architecture. The current output however provides interesting results regarding feature reduction and is why it has been kept. The outputs resembles a blurred version of the original image with a tendency to highlight large values. The subtle differences in pixel intensity from each band is shown to somewhat be kept consistent with the output.

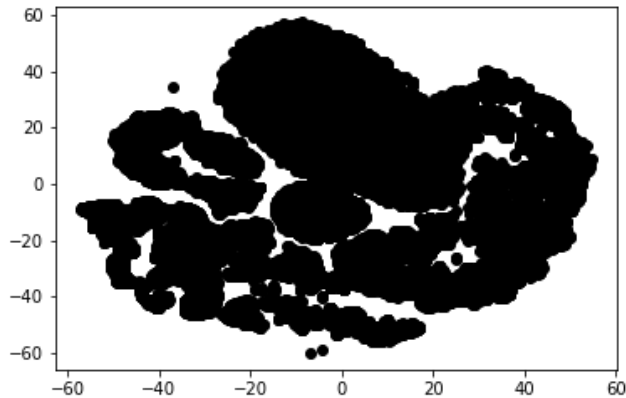


Figure 4.4: t-SNE applied on the 300 features.

4.2 Two Dimensional Plot

As for dimensionality reduction on the 300 features present in the dense layer PCA produced very poor results with a condensed plot which was hard to discern much of the information. T-NSE however provided a more human interpretable plot as it has more spread and shows signs of some early clustering.

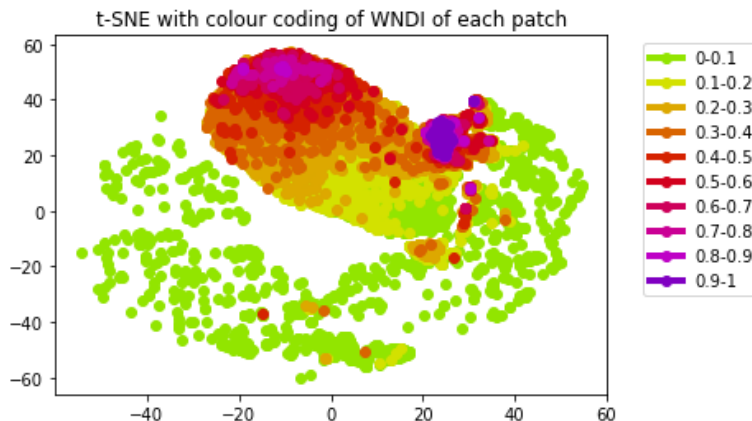


Figure 4.5: t-SNE applied on the 300 features with an colour coding to represent the water content in the patch.

As shown in figure 4.5 we can see with the help of colour coding the patches with NDWI index that the patches have clearly clustered depending on the water content. The larger water content patches have clustered and distributing outwards from the cluster is a gradient of lower water content. It is important to note that larger water content is not indicative of purely large bodies of water present in a patch; there are patches of agricultural land that are within the 90% – 100% water content, see figure 4.6. This shows that either the model is taking into account water content as a feature in the dense layer or most likely multiple features or contrarily the water content may be a symptom of different features found solely in those areas. Figure 4.6 shows patches sampled from each band of water content starting from 50% to 100%.

As there are no professionally labelled data sets available a mask had to be created

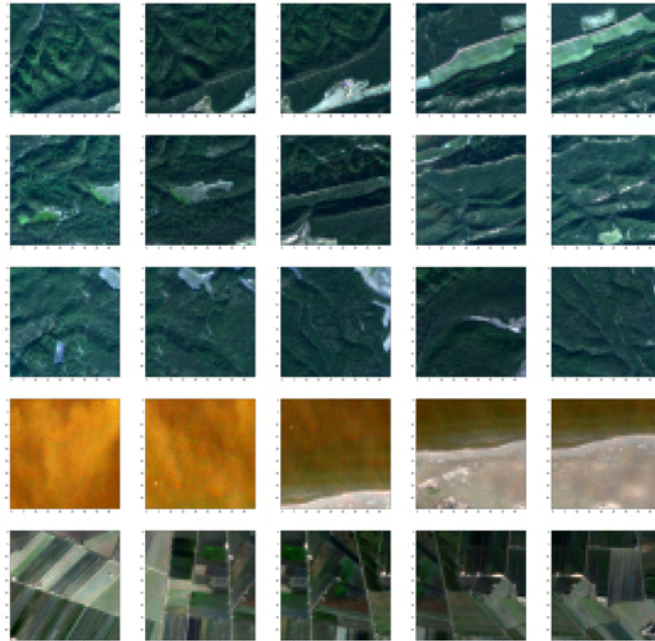


Figure 4.6: t-SNE applied on the 300 features with an colour coding to represent the water content in the patch. Top row represents patches from 50% to 60% with each subsequent row in the next 10% category.



Figure 4.7: The mask applied to extract coastal patches. Note it is not accurate and overlaps both land and water that may not be considered coastal features.

manually. The mask follows the coast however encompasses the surrounding land as well as the water to give a general area representation, see figure 4.7. This allows the results to be filtered to show just the patches that are near the coastline and to show if we can extract any trends or patterns in the two dimensional feature space.

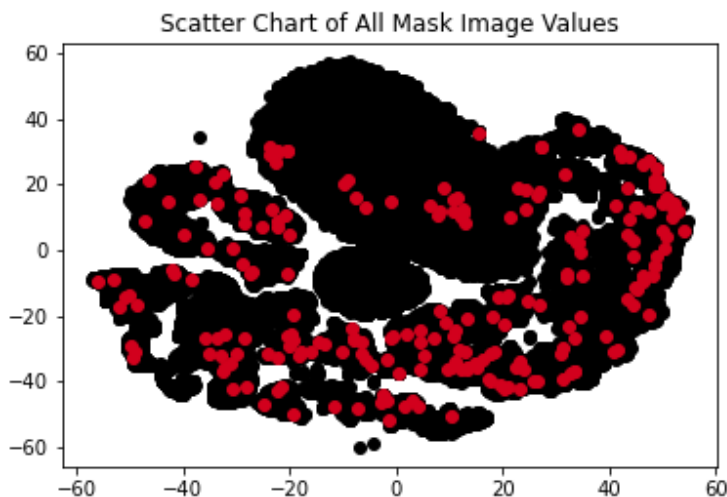


Figure 4.8: Data points in red show the coastline patches. The plot is the t-SNE embedded space of points.

Figure 4.8 shows the data points of patches that overlap with the mask. Intriguingly the features are more spread out and not within the predicted water content region. This is most likely due to the inherent complexity of the problem; there are a significant amount of coastal features that are similar to land. A method used to try and overcome this problem is to use the NDWI as a parameter when training the model. The hope would be that the model learns features that are closer to water. However as seen with figure 4.6 the water content might be misleading. Another approach is using the positional encoding introduced by transformers. Possibly adding an absolute positional encoding as well as the positional encoding for the nearest water source.

Even though the image does not show a clear clustering of the data for coastal features there may be a few sub clusters present. To find such clusters HDBSCAN was used to

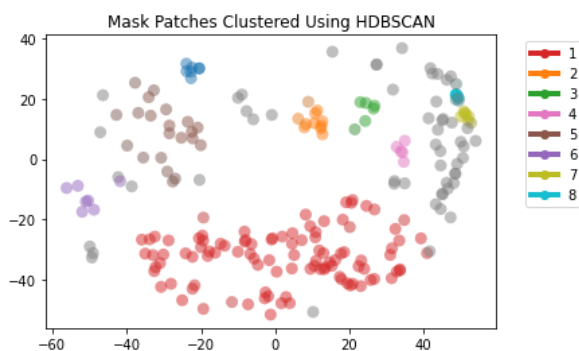


Figure 4.9: Clustering coastline data points. The legend shows the largest cluster colours in descending order.

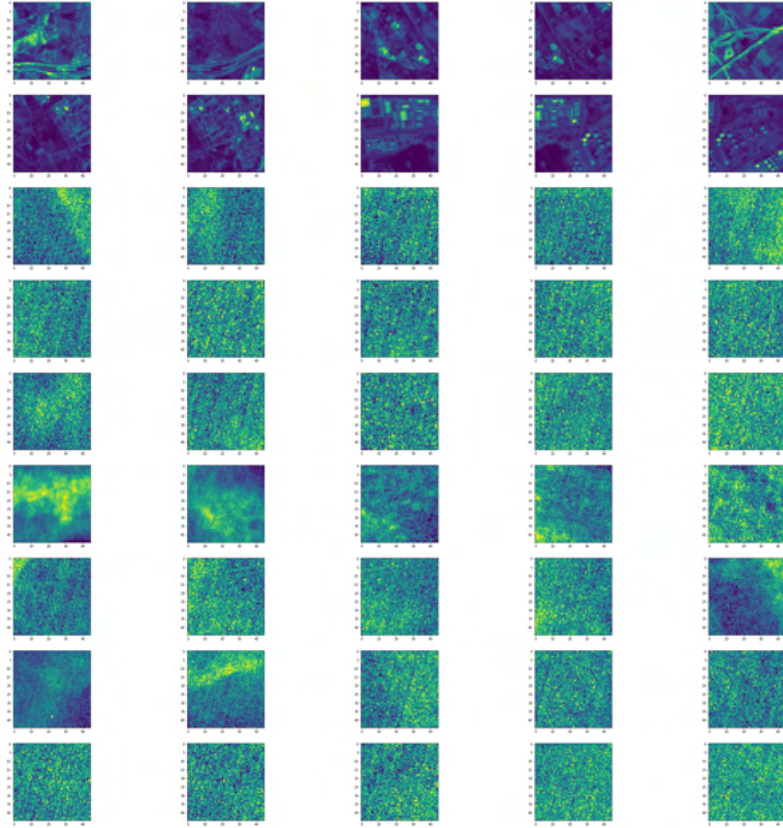


Figure 4.10: Patches representing each cluster, the highest cluster in the top row. Each row representing smaller clusters. Same order as the legend in figure 4.9.

find inherent structure within the plot. The results of clustering, see 4.9, shows eight such clusters with a minimum of five points assigned to each cluster.

The largest two clusters, refer to figure 4.10, contain mainly land with built-up areas and the second with the addition of more water. The rest of the clusters contain what looks to be grainy images but is sand features that gradually transition to water. The sand coastal clusters don't seem to present any other distinctive features even when looking at more than 5 samples.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Even though due to hardware limitations there were some initial results that were promising. The pipeline proves to be effective at extracting features and grouping the data, however when it comes to coastal features the complexity of the problem solution requires further attention and detail. Other potential additions to the pipeline have been found and would look to be considered in the future.

5.2 Future Work

The current experimental results look promising when considering applying more complex architectures such as the LSTM or Transformers self attention as discussed before. LSTMs may provide a much needed temporal feature extraction as coastal regions change periodically with time and could be captured as a feature. In addition adding a positional encoding to the data could allow the network to learn features that overlap in patches. One of the key experiments to run would be to reduce or enlarge patch size. Smaller patches may present unique features that may not be extracted by larger patches; if the patch is small enough to only contain sand or the edge of a harbour it may provide more detailed features. Smaller patches would also alter the architecture of the model as not as many layers would be needed, limitations on down-sampling. Larger patches may introduce more complex features, it is very dependent on the problem and need testing. In addition decreasing the stride would allow for more detailed feature extraction rather than capturing only a segment. The project aims to achieve explainable black box models and should therefore in the future utilise explainable AI techniques discussed to understand the output for both the architecture and experts opinion on if its a suitable model. As the final goal is to get a pixel by pixel extractions of coasts in a global context the model must be applied to much larger quantities of images, rather than the one tile used currently. The effect of increasing the observed region increases the number of features and therefore the depth of the architecture used. The many avenues to venture in future work are mostly with optimising the dataset and AI architectures with more computational resources.

Bibliography

- [1] ABDI, H., AND WILLIAMS, L. J. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics* 2, 4 (2010), 433–459.
- [2] AIAZZI, B., BARONTI, S., AND SELVA, M. Improving component substitution pansharpening through multivariate regression of ms + pan data. *IEEE Transactions on Geoscience and Remote Sensing* 45, 10 (2007), 3230–3239.
- [3] ALI, M., JONES, M. W., XIE, X., AND WILLIAMS, M. Timecluster: dimension reduction applied to temporal data for visual analytics. *The Visual Computer* 35, 6-8 (2019), 1013–1026.
- [4] BACH, S., BINDER, A., MONTAVON, G., KLAUSCHEN, F., MÜLLER, K.-R., AND SAMEK, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10, 7 (2015), e0130140.
- [5] BISCHOF, H., SCHNEIDER, W., AND PINZ, A. J. Multispectral classification of landsat-images using neural networks. *IEEE Transactions on Geoscience and Remote Sensing* 30, 3 (1992), 482–490.
- [6] CAI GAO, B. NdwI—a normalized difference water index for remote sensing of vegetation liquid water from space. *Remote Sensing of Environment* 58, 3 (1996), 257 – 266.
- [7] CHAUDHURI, D., AND SAMAL, A. An automatic bridge detection technique for multispectral images. *IEEE Transactions on Geoscience and Remote Sensing* 46, 9 (2008), 2720–2727.
- [8] CHAVEZ, P., SIDES, S. C., ANDERSON, J. A., ET AL. Comparison of three different methods to merge multiresolution and multispectral data- landsat tm and spot panchromatic. *Photogrammetric Engineering and remote sensing* 57, 3 (1991), 295–303.
- [9] CHEN, Y., JIANG, H., LI, C., JIA, X., AND GHAMISI, P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing* 54, 10 (2016), 6232–6251.
- [10] CHEN, Y., LIN, Z., ZHAO, X., WANG, G., AND GU, Y. Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7, 6 (2014), 2094–2107.
- [11] CHOI, K., HAWTHORNE, C., SIMON, I., DINCULESCU, M., AND ENGEL, J. Encoding musical style with transformer autoencoders. *arXiv preprint arXiv:1912.05537* (2019).

- [12] COOLEY, M. On human-machine symbiosis. In *Human Machine Symbiosis*. Springer, 1996, pp. 69–100.
- [13] DRUSCH, M., DEL BELLO, U., CARLIER, S., COLIN, O., FERNANDEZ, V., GASCÓN, F., HOERSCH, B., ISOLA, C., LABERINTI, P., MARTIMORT, P., MEYGRET, A., SPOTO, F., SY, O., MARCHESE, F., AND BARGELLINI, P. Sentinel-2: Esa’s optical high-resolution mission for gmes operational services. *Remote Sensing of Environment* 120 (2012), 25 – 36. The Sentinel Missions - New Opportunities for Science.
- [14] DU, Q., YOUNAN, N. H., KING, R., AND SHAH, V. P. On the performance evaluation of pan-sharpening techniques. *IEEE Geoscience and Remote Sensing Letters* 4, 4 (2007), 518–522.
- [15] DU, Y., ZHANG, Y., LING, F., WANG, Q., LI, W., AND LI, X. Water bodies’ mapping from sentinel-2 imagery with modified normalized difference water index at 10-m spatial resolution produced by sharpening the swir band. *Remote Sensing* 8, 4 (2016).
- [16] DU, Z., LI, W., ZHOU, D., TIAN, L., LING, F., WANG, H., GUI, Y., AND SUN, B. Analysis of landsat-8 oli imagery for land surface water mapping. *Remote Sensing Letters* 5, 7 (2014), 672–681.
- [17] EHLERS, M., KLONUS, S., JOHAN ÅSTRAND, P., AND ROSSO, P. Multi-sensor image fusion for pansharpening in remote sensing. *International Journal of Image and Data Fusion* 1, 1 (2010), 25–45.
- [18] ELARAB, M., TICLAVILCA, A. M., TORRES-RUA, A. F., MASLOVA, I., AND MCKEE, M. Estimating chlorophyll with thermal and broadband multispectral high resolution imagery from an unmanned aerial system using relevance vector machines for precision agriculture. *International Journal of Applied Earth Observation and Geoinformation* 43 (2015), 32 – 42.
- [19] GARNOT, V. S. F., LANDRIEU, L., GIORDANO, S., AND CHEHATA, N. Satellite image time series classification with pixel-set encoders and temporal self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020).
- [20] GEHRING, J., AULI, M., GRANGIER, D., YARATS, D., AND DAUPHIN, Y. N. Convolutional sequence to sequence learning. *CoRR abs/1705.03122* (2017).
- [21] GENG, J., FAN, J., WANG, H., AND MA, X. Change detection of marine reclamation using multispectral images via patch-based recurrent neural network. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (2017), pp. 612–615.
- [22] GILLESPIE, A. R., KAHLE, A. B., AND WALKER, R. E. Color enhancement of highly correlated images. ii. channel ratio and “chromaticity” transformation techniques. *Remote Sensing of Environment* 22, 3 (1987), 343–365.

- [23] HARVEY, N. R., THEILER, J., BRUMBY, S. P., PERKINS, S., SZYMANSKI, J. J., BLOCH, J. J., PORTER, R. B., GALASSI, M., AND YOUNG, A. C. Comparison of genie and conventional supervised classifiers for multispectral image feature extraction. *IEEE Transactions on Geoscience and Remote Sensing* 40, 2 (2002), 393–404.
- [24] HEERMANN, P. D., AND KHAZENIE, N. Classification of multispectral remote sensing data using a back-propagation neural network. *IEEE Transactions on Geoscience and Remote Sensing* 30, 1 (1992), 81–88.
- [25] HUANG, X., ZHANG, L., AND LI, P. Classification and extraction of spatial features in urban areas using high-resolution multispectral imagery. *IEEE Geoscience and Remote Sensing Letters* 4, 2 (2007), 260–264.
- [26] HUI, F., XU, B., HUANG, H., YU, Q., AND GONG, P. Modelling spatial-temporal change of poyang lake using multitemporal landsat imagery. *International Journal of Remote Sensing* 29, 20 (2008), 5767–5784.
- [27] IENCO, D., INTERDONATO, R., GAETANO, R., AND MINH, D. H. T. Combining sentinel-1 and sentinel-2 satellite image time series for land cover mapping via a multi-source deep learning architecture. *ISPRS Journal of Photogrammetry and Remote Sensing* 158 (2019), 11–22.
- [28] JIA, X., KHANDELWAL, A., NAYAK, G., GERBER, J., CARLSON, K., WEST, P., AND KUMAR, V. Incremental dual-memory lstm in land cover prediction. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2017), KDD '17, Association for Computing Machinery, p. 867–876.
- [29] JIANG, H., FENG, M., ZHU, Y., LU, N., HUANG, J., AND XIAO, T. An automated method for extracting rivers and lakes from landsat imagery. *Remote Sensing* 6, 6 (2014), 5067–5089.
- [30] LABEN, C. A., AND BROWER, B. V. Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening, Jan. 4 2000. US Patent 6,011,875.
- [31] LANARAS, C., BIOUCAS-DIAS, J., GALLIANI, S., BALTSAVIAS, E., AND SCHINDLER, K. Super-resolution of sentinel-2 images: Learning a globally applicable deep neural network. *ISPRS Journal of Photogrammetry and Remote Sensing* 146 (2018), 305 – 319.
- [32] LI, W., QIN, Y., SUN, Y., HUANG, H., LING, F., TIAN, L., AND DING, Y. Estimating the relationship between dam water level and surface water area for the danjiangkou reservoir using landsat remote sensing images. *Remote Sensing Letters* 7, 2 (2016), 121–130.
- [33] LIU, D., AND LIU, G. A transformer-based variational autoencoder for sentence generation. In *2019 International Joint Conference on Neural Networks (IJCNN)* (2019), IEEE, pp. 1–7.
- [34] LYU, H., LU, H., AND MOU, L. Learning a transferable change rule from a recurrent neural network for land cover change detection. *Remote Sensing* 8, 6 (2016).

- [35] MAATEN, L. V. D., AND HINTON, G. Visualizing data using t-sne. *Journal of machine learning research* 9, Nov (2008), 2579–2605.
- [36] MAIN-KNORN, M., PFLUG, B., LOUIS, J., DEBAECKER, V., MÜLLER-WILM, U., AND GASCON, F. Sen2cor for sentinel-2. In *Image and Signal Processing for Remote Sensing XXIII* (2017), vol. 10427, International Society for Optics and Photonics, p. 1042704.
- [37] MCFEETERS, S. K. The use of the normalized difference water index (ndwi) in the delineation of open water features. *International Journal of Remote Sensing* 17, 7 (1996), 1425–1432.
- [38] MCINNIS, L., HEALY, J., AND MELVILLE, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).
- [39] MOSKOLAÏ, W., ABDOU, W., DIPANDA, A., AND KOLYANG, D. T. Application of lstm architectures for next frame forecasting in sentinel-1 images time series. *arXiv preprint arXiv:2009.00841* (2020).
- [40] MOU, L., BRUZZONE, L., AND ZHU, X. X. Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. *IEEE Transactions on Geoscience and Remote Sensing* 57, 2 (2019), 924–935.
- [41] MOU, L., AND ZHU, X. X. A recurrent convolutional neural network for land cover change detection in multispectral images. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium* (2018), pp. 4363–4366.
- [42] NALEPA, J., MYLLER, M., IMAI, Y., HONDA, K., TAKEDA, T., AND ANTONIAK, M. Unsupervised segmentation of hyperspectral images using 3-d convolutional autoencoders. *IEEE Geoscience and Remote Sensing Letters* (2019), 1–5.
- [43] NIELSEN, M. A. *Neural networks and deep learning*, vol. 2018. Determination press San Francisco, CA, 2015.
- [44] OF WEARDALE KCB DL, L. E. Artificial intelligence and public standards: report, 2020.
- [45] PADWICK, C., DESKEVICH, M., PACIFICI, F., AND SMALLWOOD, S. Worldview-2 pan-sharpening. In *Proceedings of the ASPRS 2010 Annual Conference, San Diego, CA, USA* (2010), vol. 2630, pp. 1–14.
- [46] PETSUK, V., DAS, A., AND SAENKO, K. RISE: randomized input sampling for explanation of black-box models. *CoRR abs/1806.07421* (2018).
- [47] RAHMANI, S., STRAIT, M., MERKURJEV, D., MOELLER, M., AND WITTMAN, T. An adaptive ihs pan-sharpening method. *IEEE Geoscience and Remote Sensing Letters* 7, 4 (2010), 746–750.
- [48] REN, J., ZABALZA, J., MARSHALL, S., AND ZHENG, J. Effective feature extraction and data reduction in remote sensing using hyperspectral imaging [applications corner]. *IEEE Signal Processing Magazine* 31, 4 (2014), 149–154.

- [49] RUSSWURM, M., AND KORNER, M. Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2017), pp. 11–19.
- [50] RUSSWURM, M., AND KÖRNER, M. Self-attention for raw optical satellite time series classification, 2019.
- [51] SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117.
- [52] SHARMA, A., LIU, X., AND YANG, X. Land cover classification from multi-temporal, multi-spectral remotely sensed imagery using patch-based recurrent neural networks. *Neural Networks* 105 (2018), 346 – 355.
- [53] SHAW, P., USZKOREIT, J., AND VASWANI, A. Self-attention with relative position representations. *CoRR abs/1803.02155* (2018).
- [54] SHEN, J., WANG, R., AND SHEN, H.-W. Visual exploration of latent space for traditional chinese music. *Visual Informatics* 4, 2 (2020), 99 – 108. PacificVis 2020 Workshop on Visualization Meets AI.
- [55] SUN, Z.-L., HUANG, D.-S., AND CHEUN, Y.-M. Extracting nonlinear features for multispectral images by fcmc and kpca. *Digital Signal Processing* 15, 4 (2005), 331–346.
- [56] TAN, K. C., LIM, H. S., AND JAFRI, M. Z. M. Comparison of neural network and maximum likelihood classifiers for land cover classification using landsat multi-spectral data. In *2011 IEEE Conference on Open Systems* (2011), pp. 241–244.
- [57] TEIMOURI, N., DYRMANN, M., AND JØRGENSEN, R. N. A novel spatio-temporal fcn-lstm network for recognizing various crop types using multi-temporal radar images. *Remote Sensing* 11, 8 (2019), 990.
- [58] THERAN, C. A., ÁLVAREZ, M. A., ARZUAGA, E., AND SIERRA, H. A pixel level scaled fusion model to provide high spatial-spectral resolution for satellite images using lstm networks. In *2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)* (2019), pp. 1–5.
- [59] THOMAS, C., RANCHIN, T., WALD, L., AND CHANUSSOT, J. Synthesis of multi-spectral images to high spatial resolution: A critical review of fusion methods based on remote sensing physics. *IEEE Transactions on Geoscience and Remote Sensing* 46, 5 (2008), 1301–1312.
- [60] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A., KAISER, L., AND POLOSUKHIN, I. Attention is all you need. arxiv 2017. *arXiv preprint arXiv:1706.03762* (2017).
- [61] WANG, D., BAI, Y., AND LI, Y. Multispectral pan-sharpening via dual-channel convolutional network with convolutional lstm based hierarchical spatial-spectral feature fusion. *arXiv preprint arXiv:2007.10060* (2020).

- [62] WANG, D., WAN, B., QIU, P., SU, Y., GUO, Q., WANG, R., SUN, F., AND WU, X. Evaluating the performance of sentinel-2, landsat 8 and pléiades-1 in mapping mangrove extent and species. *Remote Sensing* 10, 9 (2018).
- [63] WANG, T., FANG, F., LI, F., AND ZHANG, G. High-quality bayesian pansharpening. *IEEE Transactions on Image Processing* 28, 1 (2019), 227–239.
- [64] WANG, Y., ALIFU, K., MA, H., LI, J., HALIK, U., AND LV, Y. Multi-modal remote sensing image description based on word embedding and self-attention mechanism. In *2019 3rd International Symposium on Autonomous Systems (ISAS)* (2019), pp. 358–363.
- [65] XIE, H., LUO, X., XU, X., TONG, X., JIN, Y., PAN, H., AND ZHOU, B. New hyperspectral difference water index for the extraction of urban water bodies by the use of airborne hyperspectral images. *Journal of Applied Remote Sensing* 8, 1 (2014), 1 – 15.
- [66] XINGJIAN, S., CHEN, Z., WANG, H., YEUNG, D.-Y., WONG, W.-K., AND WOO, W.-C. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems* (2015), pp. 802–810.
- [67] XU, H. Modification of normalised difference water index (ndwi) to enhance open water features in remotely sensed imagery. *International Journal of Remote Sensing* 27, 14 (2006), 3025–3033.
- [68] XU, J., ZHU, Y., ZHONG, R., LIN, Z., XU, J., JIANG, H., HUANG, J., LI, H., AND LIN, T. Deepcropmapping: A multi-temporal deep learning approach with improved spatial generalizability for dynamic corn and soybean mapping. *Remote Sensing of Environment* 247 (2020), 111946.
- [69] YAMASHITA, R., NISHIO, M., DO, R. K. G., AND TOGASHI, K. Convolutional neural networks: an overview and application in radiology. *Insights into imaging* 9, 4 (2018), 611–629.
- [70] YAO, F., WANG, C., DONG, D., LUO, J., SHEN, Z., AND YANG, K. High-resolution mapping of urban surface water using zy-3 multi-spectral imagery. *Remote Sensing* 7, 9 (2015), 12336–12355.
- [71] ZHANG, Y., AND HONG, G. An ihs and wavelet integrated approach to improve pansharpening visual quality of natural colour ikonos and quickbird images. *Information Fusion* 6, 3 (2005), 225–234.
- [72] ZHAO, W., JIAO, L., MA, W., ZHAO, J., ZHAO, J., LIU, H., CAO, X., AND YANG, S. Superpixel-based multiple local cnn for panchromatic and multispectral image classification. *IEEE Transactions on Geoscience and Remote Sensing* 55, 7 (2017), 4141–4156.