PHYS4037 Research Project

Slavery From Space: Deep Learning Brick Kilns from Low Resolution Images

Albert Nyarko-Agyei

Supervised by Dr Maggie Lieu

A dissertation submitted in partial fulfilment of the requirement for the degree,

Master of Machine Learning in Science

School of Physics and Astronomy

University of Nottingham

2021/22

I have read and understood the School and University guidelines on plagiarism. I confirm that this work is my own, apart from the acknowledged references.

This work would not have been possible without the support of my supervisor, Dr. Maggie Lieu through weekly meeting and discussions, so I am extremely grateful. I am also thankful for the support from Professor Doreen S. Boyd, Professor Giles M. Foody and Dr. Renoy Girindran from the Rights Lab in Nottingham. Finally, thank you to my parents Alex and Sylvette, and my siblings André and Alex Jnr for their ongoing encouragement of my every pursuit.

Abstract

In this project, semantic segmentation has been successfully applied to remote images to detect brick kilns. The segmentation was achieved using a U-Net style deep learning architecture with separable convolutions to improve inference times. Kilns were detected with a recall of 81.8% and a precision of 90% in the validation region despite the low-resolution of the input images. This suggests that the resolution of remote images need not be high in order to detect kilns to a high level of accuracy. A drawback however, was that the performance of the network can vary significantly based on the regions trained on. Therefore, the time overhead of applying data augmentation techniques is possibly warranted for future work trying to improve the accuracy of kiln detection models working with low-resolution data.

Contents

1 Introduction

The Brick Kiln industry in India employs approximately 15 million workers [1]. Having this proportion of the workforce working in the industry allows India to be the second highest exporter of bricks with an annual production of 250 billion bricks in 2020 and with the national demand expected to reach 500 billion by 2030 [1]. India's current productivity represents 81% of bricks made in the South Asia Region and this is despite brick making being an itinerant job in the country, limited to the non-rainy part of the year. Regrettably, this industry's high level of output is overshadowed by human rights violations involving bonded labour and a significant contribution to greenhouse gas (GHG) emissions. In 2017, it was reported that families of workers are hired and wages are paid exclusively to the male head of the family [2]. Of the males interviewed, 96% admitted to being given a loan before working in the kiln and all of the interviewees had their wages illegally withheld till the end of the 8-10 working season. These appear to be key parts of the systems in place to control workers. Furthermore, the operation of brick kilns often involves the burning of low-quality coal, which contributes to GHG emissions. In a step toward reducing the emissions, the Central Pollution Control Board in India mandated the conversion of Bull Trench Kilns to Zigzag Kilns in Haryana [3]. Since this mandate in June 2017 however, the traditional kilns still accounted for 70% of brick production in 2020 [1].

With these key issues in mind, the identification and policing of brick kiln activity is important to Indian Law enforcement and on an international level to many of the UN Sustainable Development Goals most saliently, number 8, decent work for all. Through the lens of remote sensing, machine learning methods offer a solution to the identification of these kilns. This approach has been successful through using deep learning networks on high-resolution images sourced from Google Earth Pro but this places a heavy burden on the platform so there is a need therefore to move towards specialised remote sensing platforms. The platform used in this project was Planet which has the advantage of providing new images on a daily level to maps that have been averaged over months or quarters. The first time period allows for close tracking of temporal information and the latter is helpful in reducing the effect of clouds blocking the areas of interest. Crucially, the major challenge to kiln detection when using this platform is that the maps provided by Planet have a significantly lower resolution than is available on Google Earth Pro. The goal of this project was to establish if kilns can still be detected at the original rate using this new data source.

2 Related Research

2.1 Mapping Kilns

This project was built on top of the results from the Slavery from Space Project that focused on estimating the number of kilns across the region spanning India, Pakistan, Nepal and Bangladesh or region deemed the 'Brick Belt'. The paper highlights that concrete statistics on the number of slaves in this region are hard to come by due to the size of the region however the estimates that 70% of kiln workers are in bonded labour is justified [4]. The project collected images of selected regions using Google Earth and the Zooniverse platform was used to host the images and ask volunteers to mark the kilns within an image. Then a script was deployed to aggregate all the regions that were marked by at least 4 volunteers and were within five pixels of each other. The density found in these images were scaled to the remaining areas in the 'Brick Belt' that were not included during surveying and the final estimate of the number of kilns was 55,387. To automate the detection of these kilns and reduce the reliance on human experts, novel methods needed to be applied to the problem.

2.2 Deep Learning

Deep Learning and Convolutional Neural Networks (CNNs) have become the solution in image recognition tasks since the introduction of multiple Graphical Processing Units (GPUs) to train them and achieve state-of-the-art performance in 2012 [5]. Image data is high dimensional compared to input data to numerical regression tasks, a small 28 by 28 by 3-pixel image involves feeding 2352 features to a model that attempts to analyse it. CNNs take advantage of the fact that neighbouring pixels will jointly represent information about an object in the image and therefore, learning the relationships between the pixels can amount to learning the objects in the image. These networks achieve this through the use of convolutional filters or matrices that are applied over the input image to extract the relevant information in the image and the whole essence of training is to find the combination of filters that optimise the detection of the target object. In using convolution filters the number of parameters in the network is also reduced because the number of parameters is mainly dependent on the size and number of filters rather than the size of the input. Furthermore, the sliding of these filters over the image, introduces an invariance of the detection ability with respect to where the object is in the input image. This is because as the filters are typically applied to all parts of the image, the object can be detected

in different sections of the input. This is in stark contrast to neural networks that use densely connected layers and result in the detection ability of the network being highly dependent on the position of the object in the image. The success of CNNs was to overcome this issue whilst reducing the number of parameters.

Deep learning was used in the detection of kilns in a 2019 study [6]. This study focused on a 120km^2 region in Rajasthan, India and used the classifications from volunteers on the Zooniverse platform as the training input for a two neural network approach to test detecting brick kilns in this area. The Faster-RCNN was the first network and it was used to propose candidate regions due to its two-stage object detection modules [7]. This network architecture introduced the Region Proposal Network (RPN) that identifies the potentially detected object's bounds and also outputs a confidence score. This was a fully convolutional approach but the real triumph was to improve real-time detection capabilities which is time efficient when a large dataset or more specifically area needs to be analysed. The second network was the GoogleLeNet, the winner of the ImageNet Large Scale Visual Image Recognition Challenge (ILSVRC) in 2014 and it did so by using less parameters than previous neural networks and being more accurate [8]. This particular network architecture, dropped the then typical approach of stacking single convolutional layers by including multiple convolutions on a single layer and with different convolutional kernel sizes. These new layers, termed 'Inception modules' have the benefit of being able to get the results of both small kernel sizes and larger kernels. In the application to finding brick kilns, the input to the GoogleLeNet was an image with the potential kiln flagged from the Faster RCNN at the centre. The justification for this approach was to reduce the over estimation of kilns by the single original network and indeed the first network identified all the 178 kilns in the test region but it introduced 188 additional false positives. The application of the second network was justified because it recognised 179 of these false positives and improved the overall accuracy of the approach. The networks used were state of the art, however, the approach raises the question if a single network can be used and in so doing, reduce complexity.

A database of classified kilns was built on the aforementioned work and extended the classification from the 120km^2 region to a 1,551,997km^2 region across India, Pakistan, Nepal and Bangladesh [9]. The paper that produced the database also highlights that machine learning methods can produce analysis of Earth Observation data in favourable time and considering that a large area has to be verified, this reduces the strain on human experts to identify kilns. A Convolutional Neural Network using the YOLOv3 architecture was used and it identified 66,455

brick kilns with a precision of 98%. The YOLO architecture is a single network approach which reduces training complexity and the first researchers that produced the network found that it is less likely to predict false positives [10]. This is highly relevant given the large area of the Brick Belt and the vast range of kiln-like structures that appear in the remote imagery, from roundabouts to buildings.

Another single network solution for this problem could use Instance Segmentation techniques that predict a mask highlighting where the kilns are in an input image. Instance Segmentation involves predicting a class label for each pixel in the input image which breaks down into identifying an object within the global scope of the image and grouping the pixels from that instance of the object together. The breakthrough paper that achieved state of the art segmentation results was U-Net [11]. Firstly, U-Net's architecture was built in the decoder encoder style that takes an input image and reduces its dimensionality through pooling (taking the maximum or average value of pixels in a given range) whilst simultaneously increasing the number of channels to preserve and extract information. In the second part of the network, the output of these down-sampling layers is up-sampled back up to the size of the required segmentation map. It is important to note that if the output map needs to be the same size as the input, then for every pooling block there needs to be a related up-sampling block. This causes the number of layers to increase dramatically and one soon runs into the problem of the vanishing gradient.

2.3 Vanishing Gradients

Vanishing gradients are characterised by the partial derivatives for the weights on each layer decreasing as we move away from the final layers. In essence, the updates to the weights of the network are not significant so training plateaus once this problem arises. The key feature of U-Net that reduces this problem was the addition of skip or residual connections. These connections link the output of the down-sampling layers and up-sampling layers that have similar spatial dimensions and take this as input to further up-sampling blocks. The motivation behind this is that the final output predicted should be very close to the input image in terms of the space of possible solutions. Therefore, by appending the learning of previous layers in the network, the network stays within a region of learning that is close to the identity function or identical mapping. This theory was introduced in the influential paper that produced ResNet, the 2015 ILSCVRC winner [12]. In the paper the authors find that adding these residual connections reduces the problem of the vanishing gradient. Adding this improvement in the quality of

updates during training allows for larger networks to be trained and therefore more modelling power to be applied to problems.

More examination of the gradients during backpropagation, the phase of training when the weights of the neural network are updated, highlights this problem. It was shown in 1986, that through using the chain rule the aforementioned partial derivatives can be derived with respect to the error in prediction [13]. This allows gradient descent to be performed or the weights of the network to be optimised to reduce the prediction error and the mechanism was inspired by the biological process of neurons that are associated with an activity being strengthened to create representations. However, the researchers admit that this is a limited version of the process. Since then, more methods have been introduced to enhance the backpropagation phase. This includes the introduction of the Adam optimizer which takes into account the first and second derivatives of the cost landscape that the network is trying to navigate and find the global minimum of [14]. Optimizers tune the learning rate of neural network during training. This is a parameter that determines how much the weights are changed during a single update. Adaptive optimizers such as Adam, allow for this parameter to change to suit the conditions of training, enabling faster training of neural networks. The creators of the Adam optimizer found that the inclusion of the second derivative which was additional feature compared to previous optimizers is particularly useful in the case of tasks where the feedback to the network about how good it's prediction was is low or sparse. In computer vision tasks where the object to be identified is small or hard to detect as in the low-resolution data, these additional methods are required to achieve results in training. In short, the initial problem of vanishing gradients is reduced by use of optimizers because they involve the gradients of many previous steps so encountering weight spaces of bad performance during training does not derail learning as easily.

An additional problem with the large number of parameters to tune associated with deep networks, is that during backpropagation there are more gradients to calculate thus increasing training times. Particularly in Convolutional Neural Networks, this problem is aggravated because increasing each dimension by one unit for a square kernel causes the number of parameters to scale quadratically. Furthermore, applying a single convolutional filter to a single channel of an image involves a large number of matrix operations, aggregating over all channels of the image and every convolutional filter makes training a computational expensive task. The introduction of separable convolutions by the Inception family of networks took advantage of the fact that some convolution filters of size N by N can be broken down into an N by 1 con-

volution followed by a 1 by N convolution [15]. This separation means that less operations are performed to achieve the same result. Although the space of separable convolutions is smaller, the speed-up in training and prediction is evident in current architectures that prioritise speed such as MobileNets [16]. These MobileNets are a type of architecture introduced to reduce the latency of networks thus making them more suitable for time sensitive real-world applications. The smaller number of parameters also means that the networks are less prone to overfitting because of the separable convolutions that are employed therefore the proposers of the original MobileNet found it important to use less regularisation techniques. This effect is because a small number of parameters enforces learning a sparse representation of the patterns in the dataset compared to when a large number of parameters are available to be tuned.

2.4 Aim

Producing the database of classifications required the use of high-resolution data. At this level of usage (covering a large area) this imagery is typically expensive or it places a strain on the open-source provider. The aim of this project was to explore the changes in predictive power when using lower resolution data whilst exploring different network architectures to perform the task of kiln detection.

3 Data

3.1 Selecting TIFFs

Figure 1: The different presentations of kilns in the database.

The input data for the neural networks were arrays created from select TIFF files downloaded from Planet basemaps interface. Due to the large spatial area being considered, it was important that the training data be equally diverse. Although the consistent structure of a brick kiln is what allows the learning of its distribution, the terrain around the brick kiln was one particular factor that changed significantly across the region of interest. As such the TIFFs were selected from areas including Andhra Pradesh in the south of India to the state Jammu and Kashmir in northern India and this range in the presentation of kilns can be seen in Figure 1 and the full list of TIFFs in the appendix (see Table 1). The kilns shown in the figure have distinct backgrounds and this increases the types of kilns captured in the training dataset and allows models training for this task to be more robust across the entire region. Furthermore, the type of kiln along with the TIFF that it was drawn from are indicated in the figure title. All these kilns are from India yet kilns next to roads, rivers and surrounded by vegetation have been recorded. The diversity of the brick kilns in India allows for the lack of representation of TIFFs from the remain countries in the Brick Belt. Although, this decision was driven more by the fact that in the database that produced the classifications, the overwhelming majority of areas with the highest density of confidently classified kilns (over 0.99%) were in India. This was an important criterion for TIFF selection as machine learning algorithms are very sensitive to the quality of the data so training on the clear-cut examples would give the model the best starting point for learning. This approach also meant that filtering the data for bad examples would take

less time as most of the examples would be accurate. These high confidence kilns were plotted on a map shown in Figure 2 and then regions with the highest density were visually selected to range across terrains and across India. The north-western part of India typically had kilns that were surrounded by a dry landscape. This makes kilns particularly difficult to pick out as kilns are built on dry terrain and the associated heat from brick firing and foot traffic makes the kiln areas even more bare. On the other hand, in the middle of the Brick Belt between Rajasthan and Bihar the kilns are difficult to pick out because of the density of farmland that is also normally rectangular and located in close proximity to active kilns. Lastly, but not definitively, the areas in India bordering Bangladesh had many kilns that were close to river deltas making it difficult to distinguish between areas of eroded land and the actual kiln structures. These different scenarios present an interesting challenge for a kiln detection network. However, concerns have to be raised about if optimising for peculiarities in local regions can scale to the entire region. This is largely a question of input data so poor nationwide performance would be improved by introducing more examples to learn from for the training networks and extended training times.

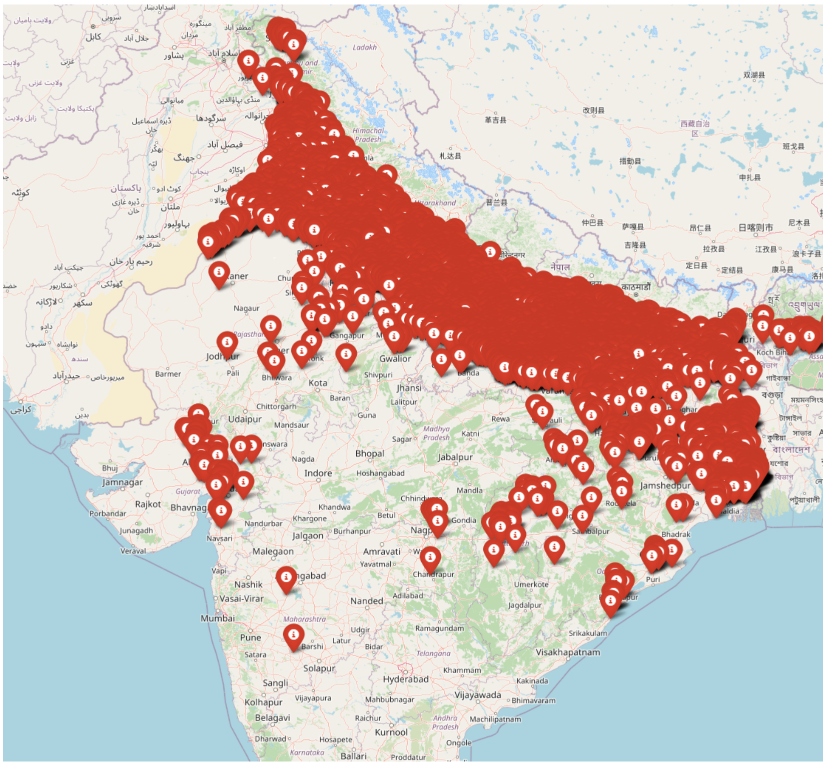

Figure 2: The locations of the most confidently classified kilns in the database.

A benefit of using the Planet data source is the level of temporal granularity that was available to download. For this project, the imagery averaged over the first quarter of 2020 was

used to train the networks. Typically, the kilns are easily identified by eye in the TIFFs that have been averaged over shorter periods of time such as a month or less however these latter TIFFs suffer from variable quality due the effect of the rainy periods reducing the visibility of kilns. The choice of cut-off date coincides with the start of the classification database so it would have the smallest proportion of kilns in the image that have not been detected. Furthermore, as the classification database is an ongoing process, any cut off date would leave some classifications out of date due to new kilns or inactive/demolished kilns.

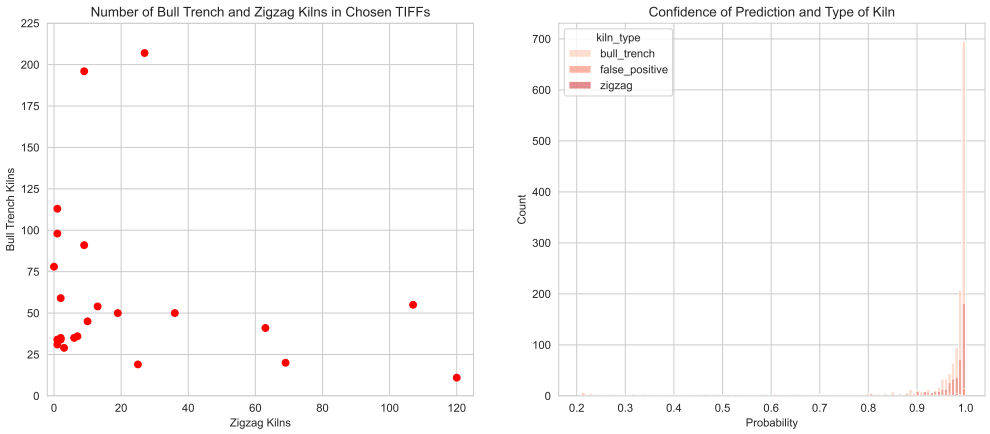

Figure 3: The distribution of kilns in the selected TIFFs.

In total, the TIFFs selected had 2087 potential kilns from the database. Out of these 1954 were actually kilns with bull trench (68.1%) or zigzag (25.5%) as the two types. The remaining cases were labelled false positives. The variation in the types of kilns present within the TIFFs can be seen in Figure 3 and the confidence of the model that identified them is also presented. In the first instance, it is visible that there are a number of TIFFs that overwhelmingly contain bull trench kilns or are dominated by zigzag kilns. This influenced training practices further along in the project to reduce the imbalance of the data that was presented to networks. The validation region was chosen with this imbalance in mind and it had 20 bull trench kilns and 25 zigzag kilns. Secondly, the kilns in the TIFFs chosen were mostly identified with a high confidence level by the model that produced the database. This gives some assurances about the quality of the data and the ease at which a new network should be able to identify these kilns.

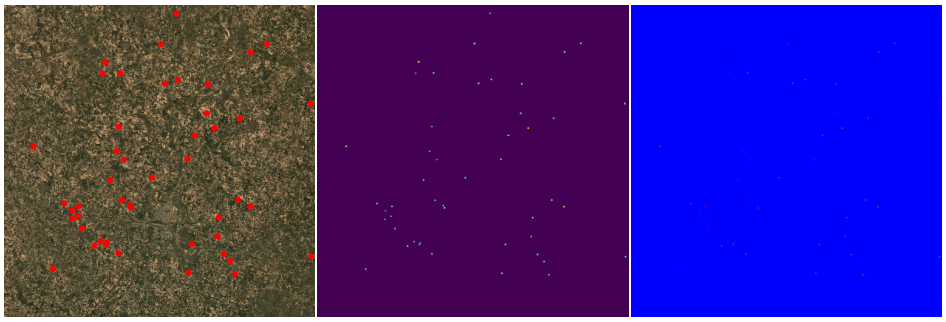

Figure 4: An example TIFF with kilns highlighted (left), mask without kiln type (middle) and with (right).

3.2 Creating Masks

Once the TIFFs had been downloaded, the coordinate systems were converted to the same reference system as in the shapefile containing the classifications. Then arrays of size 4096 by 4096 (equivalent to the spatial resolution of the TIFFs) were initialised to 0 and the respective locations where kilns were found, filled. These locations were found by extracting the information about geographic bounds encoded within a TIFF and then filtering the kiln database for all the kilns that were contained within those bounds. The initial approach was to fill the arrays with ones where the kilns were found, but for most kilns there was a small non-kiln exterior included so the filling method was changed to involve an exponential decay that increases away from the centre of the kiln using Formula 3.1, where for the pixel at index (i, j), $lat = |\text{centre latitude of kiln} - i|$ and $long = |\text{centre longitude of kiln} - j|$.

$$\text{Pixel Value} = e^{-(\frac{lat^2 + long^2)}{2*36}} \tag{3.1}$$

With the mask created for a given TIFF, 300 by 300 regions could be sampled from the TIFF and associated mask then passed as the input and target respectively to the neural networks. To ensure that the training set was balanced, half of the samples were generated by randomly selecting a kiln and then creating a 300 by 300 spatial region encompassing it. The rest of the samples were randomly selected and were not guaranteed to contain a kiln. This reduces the likelihood that the model learns a strategy that always predicts a kiln from the input image. Given the fact that Boyd et al estimated the occurrence of kilns in the Brick Belt to be 0.0357 kilns per km^2, in a 300 by 300 spatial pixel image with average 4m resolution per pixel we should

expect around 0.0514 kilns per random image from the Brick Belt [4]. This level of occurrence in training would be difficult to estimate the true gradients for a neural network so randomly cropping from a TIFF during training eases the imbalance. Notably however, the TIFFs were also selected because of their higher density of kilns to make efficient use of training time so this also plays a factor.

Figure 4 shows the kilns marked on a TIFF along with the two versions of the mask generated for a given TIFF. The first mask corresponds to a single channel array that ignores the type of kiln to be marked so bull trenches and zigzag kilns were treated in the same manner and marked according to Formula 3.1. The use of the second mask is for a network that also tries to predict the type of the kiln as well as its position in the remote image. This second mask takes advantage of the RGB display and marks the first channel with bull trenches, the second channel with zigzag kilns and the third with ones where no kilns were found or a number such that the pixel value across all three channels sums to 1 (this is just 1 minus the result of Formula 3.1). This property of the second mask allows for the creation of probabilistic classifications along with the prediction of the type of kiln. However, this mask is very similar to the first and indeed adding the first two channels of this mask recovers the original single channel mask.

4 Methodology

4.1 Model Building

4.1.1 Model Architecture

For this project, instance segmentation was chosen as the method for producing kiln classifications so the proven U-Net architecture was used to output segmentation maps. Additionally, separable convolutional filters were used instead of the classical 2D Convolutional filters to reduce training and inference times. These filters used 3 by 3 kernel sizes as a balance between traditionally large 5 by 5 kernels which significantly increase training times and the small 1 by 1 kernels which do not aggregate local information nearly as well. Being able to reduce the inference time is particularly important for this project given the large area of the Brick Belt that has to be analysed. Finally, same padding was used to prevent the loss of information along the edge of the input image where kilns could very possibly be present. A diagram of the exact final architecture has been included in the appendix for reference because of the discussed deviations from U-Net (see appendix for Figure 10).

4.1.2 Hyperparameter Search

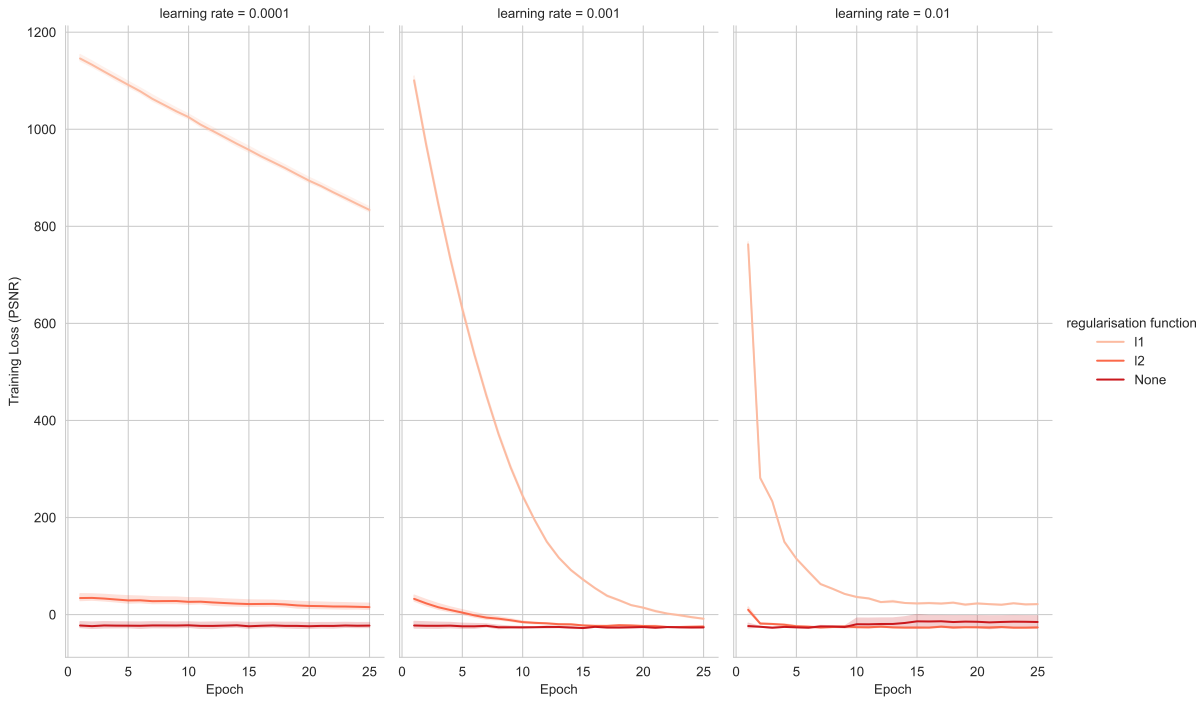

Figure 5: The results of the hyperparameter search.

After choosing the architecture, a grid search across the space of hyperparameters was performed to inform the best training practices moving forward. The parameters available to tune were the activation functions, learning rates, the weight regularisation function, up-sampling interpolation and loss function. The results of the search, visible in Figure 5, were consistent with known deep learning theories such as a large learning rate being able to rapidly move toward a lower loss but converging in local minima instead of the global minima. This confirmed the usage of a low learning rate of 0.0001 to ensure that the best weights were found for the network. The choice of regularisation function was important to how quickly the loss decreased during training. This is a method of regularisation where a function of the weights on a layer is added to the loss function. This has the effect of penalising the selection of many large weights by the model, which is associated with overfitting, in favour of a small number of weights that still perform well. If the function is the absolute value of the weight, is it deemed L1 regularisation and if it is the square of the weights, it is called L2 regularisation. When the L1 norm is used the loss function no longer has a closed form solution because the absolute value is a piecewise function that is not easily differentiable. This makes the optimisation of the L1 norm more difficult compared to optimising the L2 norm hence the slower descent in Figure 5. It is true

however that the generation of random regions from the masks during training approximates a near infinite dataset and makes this task less prone to overfitting. As a result, L1 regularisation was applied only to the two layers with the most parameters, as a result of this search. This was a trade-off between complex optimisation and preserving the learning ability.

The difficulty with doing a hyperparameter search with this task however, is that the results of segmentation are geared towards visual interpretation so although the loss may decrease with a change in parameter according to the trends in Figure 5, the actual visual output of the new model could show no progress. This was typically the case when a heavy amount of regularisation was used throughout the network because the ability to stay in a region of outputs close to the identity function, which is crucial for networks employing skip connections, was destroyed. This was another reason for reducing the weight penalisation in the network. The activation function used on the network was another problematic area. Due to the pixels in the true segmentation masks being between 0 and 1 the straightforward choice was to use the sigmoid activation function that has the equivalent range. When using this activation however, learning stalling around local minima of just predicting zeros for the entire input and did not make any visual progress for several intervals of training. This is despite the search showing that all choices of activation function resulted in identical training progress. After visually examining the segmentation outputs, the best results were achieved in networks using ELU activation. These results are a re-emergence of the vanishing gradient problem because it is known that using sigmoid activation the values outputted after activation tend to saturate around 0 and 1 and when this happens the updates during training are very slow or non-existent. It has been highlighted that it is sufficient to have just one large weight parameter for sigmodial activation functions to saturate [17]. ELU activation reduces this problem by being a function with bounds of (-1, infinity) so the range of values that can be outputted during segmentation is greater.

The choice of loss function is also an important hyperparameter to model training. The loss functions considered were the Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM). The PSNR is an extension of the MSE used for scoring a predicted image's proximity to a target image. It is calculated based on the difference between pixel values from the two images at the same index and if the target and predicted image are the same, the PSNR is infinite. The MSE is defined in Formula 4.1, where I and J are the row and column dimensions of the image array, $I_k(i, j)$ is the pixel value at index (i, j) for target mask when $k = 1$, and the predicted mask when $k = 2$. R in Formula 4.2 represents the maximum

difference between pixels in the encoding of the masks. For double-precision floats R is 1 and for 8-bit unsigned integers the value is 255.

$$MSE = \frac{\Sigma_{I,J}[I_1(i,j) - I_2(i,j)]^2}{I * J} \tag{4.1}$$

$$PSNR = 10 \log_{10}(\frac{R^2}{MSE}) \tag{4.2}$$

From Formula 4.1 and 4.2 we can deduce that it is the ratio of the maximum possible signal in the encoding of the target image and the strength of the corruption in using the predicted image as a substitute for the target. In comparison, the SSIM metric was designed to mimic the perception of humans by combining differences in luminance, contrast and structure between the two images [18]. In Formula 4.3, μ_r refers to the mean pixel value in the image or region r, σ_r is the standard deviation of those pixels and $\sigma_{r_1 r_2}$ is the covariance in the pixel values between region r_1 and r_2. The constants C_1, C_2 and C_3 ensure numerical stability and avoid division by zero if the other terms in the denominators tend to 0. The authors of the metric determined that it is best to calculate this metric over sliding windows on the image, for the same reasons convolutional filters are successsful. This justifies the regions in the formula and then the average value over the image is taken as the loss.

$$\begin{aligned} SSIM &= l(x,y) * c(x,y) * s(x,y) \\ l(x,y) &= \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \\ c(x,y) &= \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2\sigma_y^2 + C_2} \\ s(x,y) &= \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \end{aligned} \tag{4.3}$$

An objective comparison between the two metrics produced an analytical link so one can be predicted from the other and found that the only difference was in the response to Guassian noise and Image compression [19]. Nonetheless both functions were tested outside of the main hyperparameter search and the training progress when predicting kilns, visible in Figure 6, was what influenced the choice to use the PSNR moving forward. From the figure it is clear that when the network was trained on the PSNR it very quickly converged on the required strategy of identifying kilns. The SSIM trained network shows slower convergence. The focus on making the differences between the two images reduced based on the human visual system is potentially

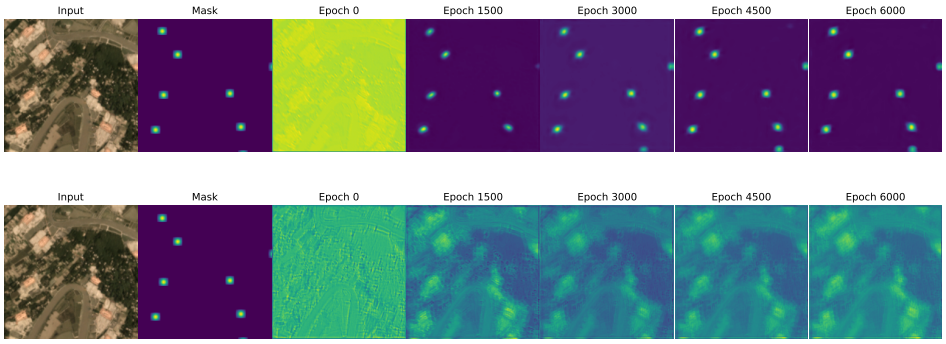

Figure 6: Progress of model prediction using PSNR (top) and SSIM (bottom) loss functions.

what reduces the effectiveness of the SSIM in this use case. Being able to match the target mask at the individual pixel level is the optimization goal of PSNR and is exactly what was needed in this task so the PSNR was used during final training. Also following convention, the losses were negated so that the aim of optimisation was to minimise.

Lastly, changes in predictions from different upsampling interpolation methods was not at all significant to the loss from the search. The preferred choice was to use bilinear interpolation on the upsampling parts of the neural network due to the smoothing effect of aggregating the pixel values from multiple neighbours compared to one neighbouring pixel value in other methods.

4.2 Model Development

4.2.1 Model Training

Once the final network was constructed with the appropriate hyperparameters, it was trained for 40,000 epochs at 4 steps per epoch. In each step a TIFF was loaded and the training set was generated from it. Every 10 epochs a sample from the validation TIFF was predicted on and saved so that training progress could be viewed. The model's weights were also saved every 20 epochs so that the model's weights at different epochs could be reloaded and scrutinised.

Training was done on three different platforms according to availability, an NVIDIA RTX 2070, an NVIDIA RTX 2080 Ti and the GPUs available on the Google Collaboratory platform taking an average of 18 seconds per epoch. Initial training was slow but through caching the specific list of kilns in each TIFF, the masks were created quicker and the training time was reduced to 4 seconds per epoch.

4.2.2 Extracting predictions from Segmentation Map

Before the results could be analysed the kiln location had to be extracted from the predicted masks and compared to the locations in true masks. Manually one can easily see if the areas of activity in a mask, i.e where kilns are predicted, match the true presence of kilns. However, due to the scale of the area considered and reproducibility objective methods have to be used to extract isolated regions in the predicted mask and these isolated regions can then be treated like kiln predictions. Another problem is that the models were trained to predict on 300 by 300 images so predicting on a 4096 by 4096 TIFF would involve some change in prediction method. In producing the classification database, the authors used a sliding window over the prediction area that included overlap to ensure that all areas were predicted on. This approach was adopted in this project, first sliding across a 3900 by 3900 by region off the TIFF in 300 by 300 steps and sliding across the edge of the TIFF that was not included thus introducing some overlap. This could have been extended to two or multiple 3900 by 3900 regions starting at different locations in order to give every kiln more opportunities to avoid being at the edge of a sliding window but this approach is computationally expensive. Decisively, the approach used managed to capture over 99.99% of the kilns confirmed in the true masks so adding further sliding windows was not necessary.

The method employed uses the region props function in the sckit-image package to extract the isolated regions from a mask after rounding the pixels to the nearer of 0 or 1 [20]. This rounding was a necessary part of the pipeline however it destroys potential information about how confident the model was about the prediction of the kiln. This is assuming that a confident prediction would have pixel values closer to 1 therefore by rounding pixel values of 0.5 and 0.9 both to 1 we lose this information. This can be recovered with the introduction of thresholds when rounding. So pixel values above the threshold value were rounded to 1 and pixel values below were given a value of 0, only then could isolated regions be extracted by the aforementioned function.

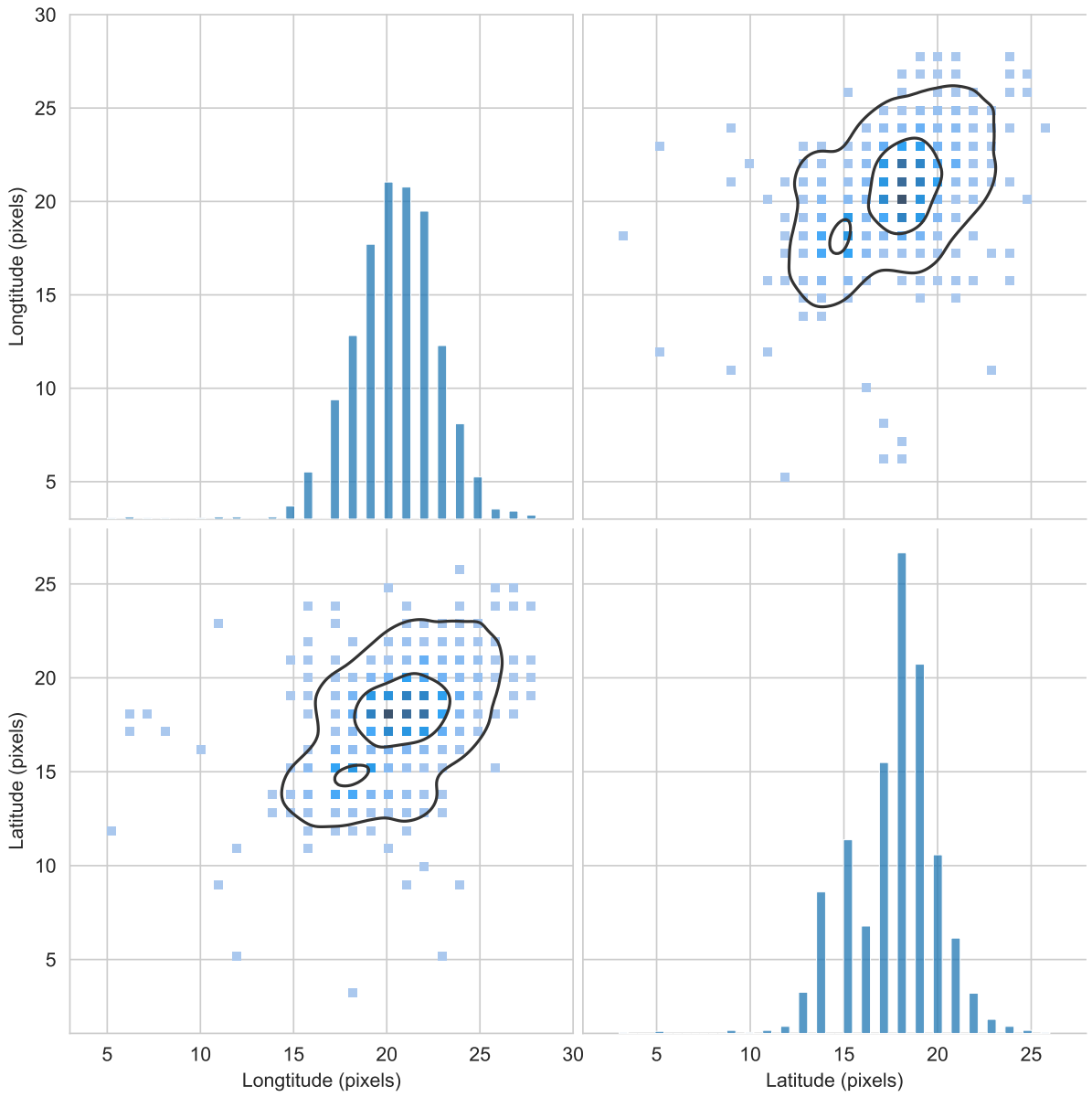

Figure 7: The pixel dimensions across latitude and longitude for kilns in database.

The further challenge was to remove all the isolated regions (kiln predictions) that were counted multiple times. This situation was almost guaranteed by the overlap introduced and also through kilns that occur on the edge of sliding windows. By analysing the distribution of the shape of the kilns in the database, it was found that all kilns were separated by a pixel distance of 20 therefore, if the centroid of a region identified was within 20 units of another identified region, then they were referencing the same kiln. The distribution is shown in Figure 7 and it highlights that the difference between the pixel index representing a kiln's lowest and highest latitude was typically under 20 and this was almost the same for the longitude. This figure was also used on the true masks and all the different kilns were correctly distinguished.

5 Results

The first result of training was that as the number of epochs increased the models predicted the kilns more confidently with pixel values closer to 1 so the threshold discussed earlier could be increased to reduce false positive rates. This also allowed better distinctions to be made between confident predictions and predictions with pixel values closer to 0. This can be seen in the ROC curve in Figure 8 plotting recall against precision for different thresholds and at different epochs during training. Recall and precision were calculated using the standard formulas in 5.1

$$\begin{aligned} Recall &= \frac{True\ Positives}{True\ Positives + False\ Negatives} \\ Precision &= \frac{True\ Positives}{True\ Positives + False\ Positives} \end{aligned} \tag{5.1}$$

Recall identifies the proportion of true kilns that were identified by a model so if there were 50 kilns in an image and 40 were identified by a model, the recall for that image is 80%. Precision on the other hand measures how many of the kilns identified by a model were actually true so if a model predicted 50 kilns and only 40 of those predictions were actually kilns, the precision of the model would be 80%. Earlier papers established that a false negative was the error that was to be avoided the most, meaning all true kilns should be correctly flagged in a kiln detection model. This is to avoid potential sites of forced labour being missed completely and as such the recall rate is the significant metric for judging the performance of a model. Both metrics are shown on the ROC curve for completeness and also because optimising for one of these metrics means performing worse on the other, a balanced performance is needed for both scores. In

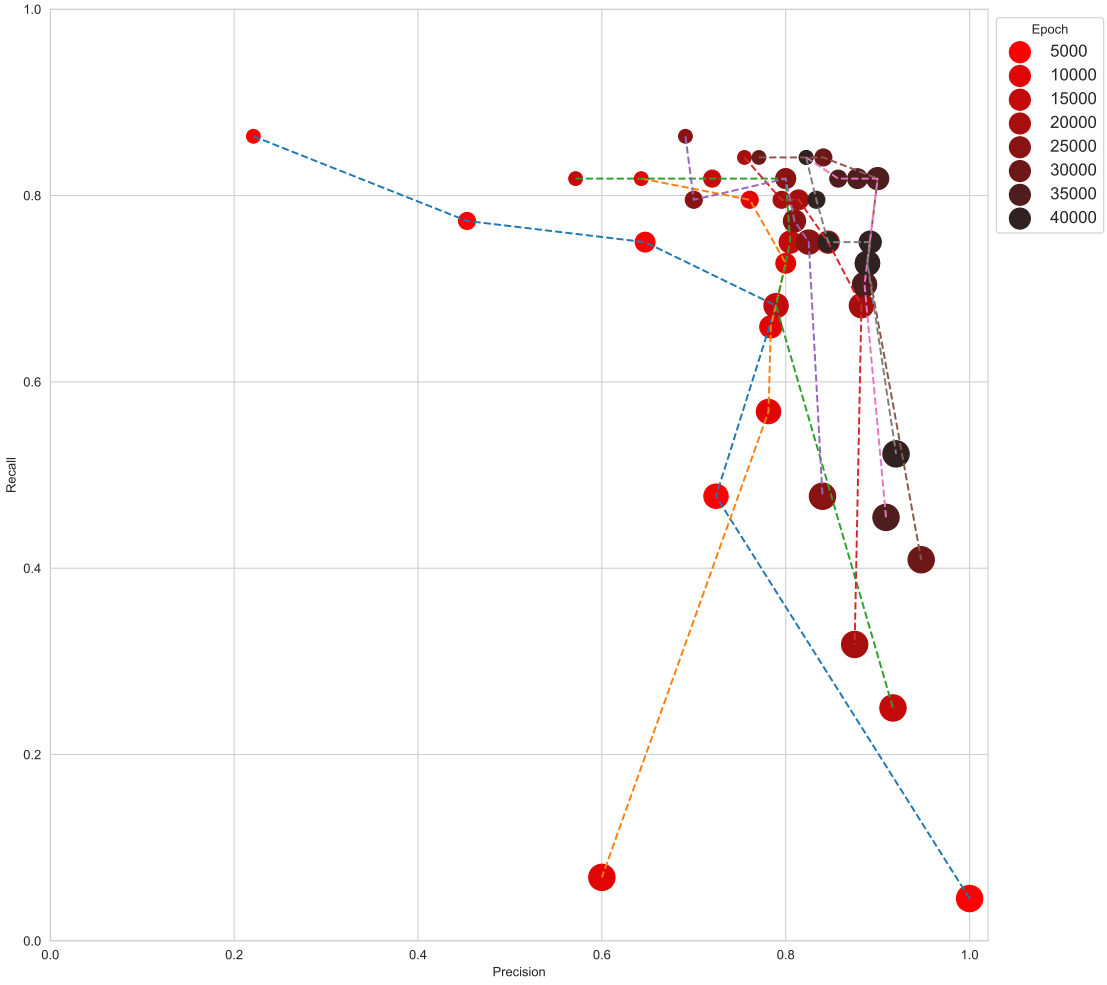

Figure 8: ROC curve showing Recall against Precision on the validation TIFF at different training points and thresholds evenly ranging from 0.15 to 0.9 denoted by small to large marker size.

theory then, the best performing model would have a recall and precision both of 1. The ROC figure shows that the best performing model was at 35000 epochs at and setting the threshold on the masks to 0.6. This model achieved a recall and precision of 81.8% and 90% respectively. This contrasts to the results of the first two-model approach described where the first network omitted no kilns so recall of 100% but precision of 50% and the second network's recall and precision were both 94.94%.

6 Discussion and Limitations

The difficulty of picking out kilns from the low-resolution data makes this task challenging for human experts. Fortunately however, neural networks have been able to outperform humans on specific tasks for a while and these networks comfortably perform classifications on the edge of human abilities such as the difficult examples on the CIFAR-100 dataset or Imagenet32 [21]. It has been suggested that this is because of the broader range of associations that neural networks can make thus allowing better extraction of features. However it was also remarked that both abilities rely on prior experience to the problem so potentially human experts given more exposure would improve performance [22]. All the same, it is the infeasibility of using human experts that drove the development of neural networks in this project and this project's aim to detect the presence of kilns in low resolution images was successfully achieved. Furthermore, the thresholding of masks that was used to eliminate weak areas of activity can also be used as a quantification of the model's certainty. For instance, kilns still detected at a threshold of above 0.9 indicate that the model is very certain that there is a kiln because the more the pixel values deviate from the most common value of 0 the greater the penalty incurred if the model is wrong. During neural network training it is typical for learning to stall around local minimas like in this case predicting 0 for every pixel every time. The divergence from this is a positive result and indicates true learning in the task.

On the subject of quantifying uncertainty, other methods that could have been used are Bayesian Neural Networks (BNNs) and the training of multiple models initialised at different regions or with different datasets to create a strong set of uncorrelated models that combined would have a high predictive power. BNNs treat the weights of a neural network as distributions so their outputs are random variables conditioned on the input. The optimization of these distributions during training is significantly longer and it has been theorised that their only benefit is to learn the variation in behaviour around the point the model has converged to

and not the actual distribution of the target variable[23]. The latter option uses the power of ensembles to improve generalisation error. Also, having a number of models predict that a given area contains a kiln is an improvement on using a prediction from just a single model and provides better certainty on the prediction because the models are uncorrelated. The main drawback of this method is the total training time required to obtain the models in the ensemble.

When creating the mask for a given TIFF, it was highlighted that the kilns contained within that TIFF's bounds were calculated and located based on the kilns in the database. This is a particularly problematic area because if there is any variance between the kilns present in the TIFF and what has been found in the database, then the training dataset loses some integrity. Typically, a network that has still learnt the target distribution will correctly identify these cases despite the training labels as long as they are the exception in the dataset and not the rule. This is because fitting to this data amounts to learning the noise in the dataset and not the structural patterns. Therefore, a model that learns these mistakes in the dataset is a clear sign of overfitting.

The TIFF used for validation was chosen for the high number of kilns present and because it was representative of the terrains encountered in the training set. This is particularly important because during the initial parts of training it was noticed that models trained on kilns from a specific TIFF did not generalise well to identifying kilns from other TIFFs. To reduce this imbalance, number of kilns in a TIFF along with the types of kilns were incorporated in a metric to guide how often a TIFF would be used in training, using the Formula 6.1, where T_K is the number of kilns in the TIFF, T_b and T_z are the number of Bull Trench and Zigzag kilns in the TIFF respectively.

$$TIFF\ Selection\ Weight = T_K - |T_b - T_z| \tag{6.1}$$

Beyond this solution, there are multiple possible reasons for this problem with generalisation. This ranges from the quality of the TIFFs from Planet changing, to kilns blending into the terrain and being harder to distinguish as in the leftmost kiln in 1. To explore this idea, models were trained and validated on one TIFF that was left out. This was done for each of the TIFF in the dataset. The results shown in Figure 9 confirm that certain TIFFs are harder to optimise predictions for and this suggests that using multiple networks to predict kilns for sections of the Brick Belt might prove more effective at capturing all presentations of kilns. That said, this is a problem of generalisation but a special case where the kilns have to be presented in

a range of different conditions. The training masks were generated from random crops around the kilns so performance would not decrease as the position of kilns in the image changed. Furthermore, because the project's approach uses Convolutional layers and these layers apply the kernels to all parts of the image, the generalisation error with respect to the position of the target object in the image is theoretically small. However, the type of error that was revealed in the leave-out-one validation is that the input data has to be augmented through contrast, brightness and image quality changes to increase the versatility of the network. Researchers have found that these changes to the input data encourage the learning of invariant features in the case of convolutional neural network training [24]. These features represent characteristics that are constant even when the kilns are viewed under different circumstances and so would cause learning to generalise better.

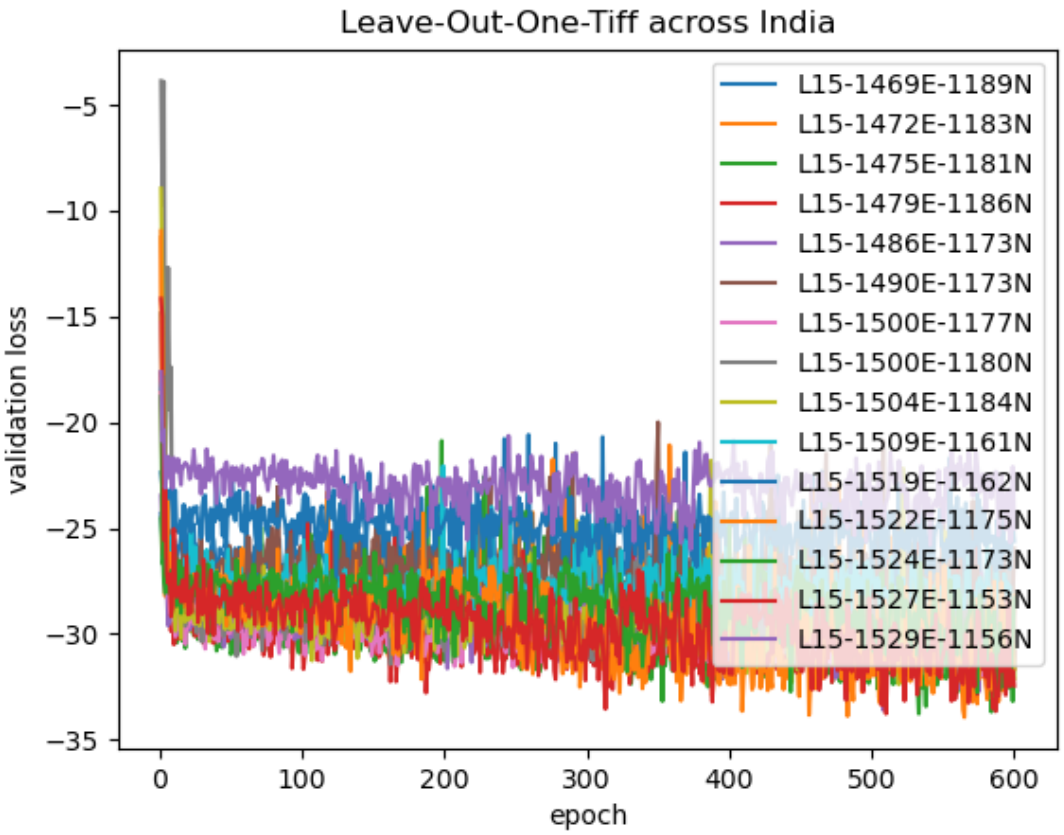

Figure 9: The results of Leave-Out-One-TIFF validation.

Augmentation techniques were explored, however, they are computational expensive so due to the limited amount of training possible it was first necessary to establish that the kilns could be detected at a high rate before raising the complexity of the task by increasing the

presentations of kilns during training. The validation strategy could also be amended to include more TIFFs as gradually increasing this number to the amount of TIFF covering India would approach the true estimate of model performance. Using just one TIFF was sufficient as shown in Figure 9 because although there were variations in the PSNR loss as discussed, performance was generally in the same region for the majority of TIFFs. As mentioned before, the loss function is misleading in terms of determining accuracy of kilns predicted because the relationship between segmentation output and loss value is many to one. With this fact in mind, it was important to visualise model progress during training by the saving frequency described earlier.

Another issue during training was setting the distance under which a prediction would be deemed identical to a specific kiln. This distance was chosen from calculating the dimensions of the kilns in the database of kilns but it also reflects the fact that the masks produced can not perfectly align with the actual position of the kiln. This is due to rounding errors introduced when producing the original classifications of the kilns and the subsequent errors produced by using the centroids of rectangular polygons around kilns when they are mainly oval in shape. This problem also means that the typical approach of producing a confusion matrix using the class of the pixel in the true versus predicted masks can not be followed strictly because although the network has successfully learned to predict the general location of a kiln, perfectly learning the exact locations in training mask would amount to learning this distance error in the dataset. In short, producing a confusion matrix from just the segmentation mask and ground truth mask would require highly accurate ground truth masks during validation.

An additional objective of the project was to predict the kiln type, this could either be a bull trench or a zigzag according to the database. The same segmentation model could be used but the final layer was adjusted to output a three channel image matching the form of the third mask in Figure 4. Also, the final activation was changed to softmax so that the output for each pixel would sum to 1, simulating a probability distribution indicating the confidence of the model in its prediction. Finally, the loss function was adjusted to use the categorical cross entropy, common for classification tasks. This is an increase in the complexity of the problem so understandably the models did not perform as well. The kilns were still detected however performance was worse than if it was assigned by chance. Potentially, this could be improved with longer training. It could also be that the detection of the kiln type is at the limit of the current architecture's abilities so adjustments need to be made to improve in this area.

With more time, the approach could be changed to use larger models to improve the detection

abilities of the network. This would result in longer training and inference times but better performance in terms of the recall and precision. The greater expressive power of larger models would truly allow the bounds of detection performance to be quantified. Potentially though, the plateau in testing performance, found at 35000 epochs as indicated by the ROC curve, could be a local minima and extending the training with the current model could improve results. The low speed of improvement however suggests that changes need to be made to reach peak performance faster such as increasing the number of training examples that have kilns in them or focusing on perfecting prediction on areas with an extremely high density of kilns and scaling to low density areas.

7 Conclusion

A U-Net inspired architecture was used to build a fully convolutional neural network and detect brick kilns. This architecture was characterised by two halves: a contracting or downsampling part and an expanding or upsampling part. This was to enable the extraction of the relevant features for the task of brick kiln detection and build towards a mask that has a pixel-wise classification of whether there is a brick kiln or not. This task was treated as an instance segmentation task so every separate instance of a kiln in an input image had to be identified. This was successfully achieved and with results close to when high resolution input data was used in other works.

Training was difficult however, due to the low level of feedback that neural networks typically get in image recognition tasks. As a result, at least one kiln was enforced to be present in half the training instances. This allowed the networks to be trained faster but also introduced some bias to the network because in reality, kilns are not present in half of all random 300 by 300 Planet images taken of the brick belt. The problems raised by this method of training would have been reflected in a low precision due to the high amount of false positives. This however, was not that case as the best performing model had a precision of 90% on testing data.

The results of the project indicate that kilns can still be accurately detected using low resolution images from Planet. The ability to pick out these kilns is impressive given the resolution of the data and the difficulty for human experts. Further work can be done to match the performance of networks trained on high-resolution data and the gap found by this work suggests that this is well within reach subject to more accurate segmentation masks and longer training periods.

References

[1] Andrew Eil, Jie Li, Prajwal Baral, and Eri Saikawa. *Dirty Stacks, High Stakes: An Overview of Brick Sector in South Asia.* 04 2020.

[2] Anti-Slavery International. Slavery in india's brick kilns the payment system: way forward in the fight for fair wages, decent work and eradication of slavery. 2017.

[3] Central Pollution Control Board. Mandate for kiln conversion or closure. 2017.

[4] Doreen S. Boyd, Bethany Jackson, Jessica Wardlaw, Giles M. Foody, Stuart Marsh, and Kevin Bales. Slavery from space: Demonstrating the role for satellite remote sensing to inform evidence-based action related to un sdg number 8. *ISPRS Journal of Photogrammetry and Remote Sensing*, 142:380–388, 2018.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[6] Giles M. Foody, Feng Ling, Doreen S. Boyd, Xiaodong Li, and Jessica Wardlaw. Earth observation and machine learning to meet sustainable development goal 8.7: Mapping sites associated with slavery from space. *Remote Sensing*, 11(3), 2019.

[7] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[8] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[9] Doreen S. Boyd, Bertrand Perrat, Xiaodong Li, Bethany Jackson, Todd Landman, Feng Ling, Kevin Bales, Austin Choi-Fitzpatrick, James Goulding, Stuart Marsh, and Giles M. Foody. Informing action for united nations sdg target 8.7 and interdependent sdgs: Examining modern slavery from space. *Humanities and Social Sciences Communications*, 8, 2021.

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2015.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[13] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning Representations by Back-Propagating Errors*, page 696–699. MIT Press, Cambridge, MA, USA, 1988.

[14] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[15] François Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.

[16] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. 2017.

[17] Anna Rakitianskaia and Andries Engelbrecht. Measuring saturation in neural networks. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 1423–1430, 2015.

[18] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[19] Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010.

[20] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in python. *PeerJ*, 2:e453, June 2014.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[22] Samuel F. Dodge and Lina J. Karam. A study and comparison of human and deep learning recognition performance under visual distortions. *CoRR*, abs/1705.02498, 2017.

[23] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. 2019.

[24] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin A. Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. *CoRR*, abs/1406.6909, 2014.

8 Appendix

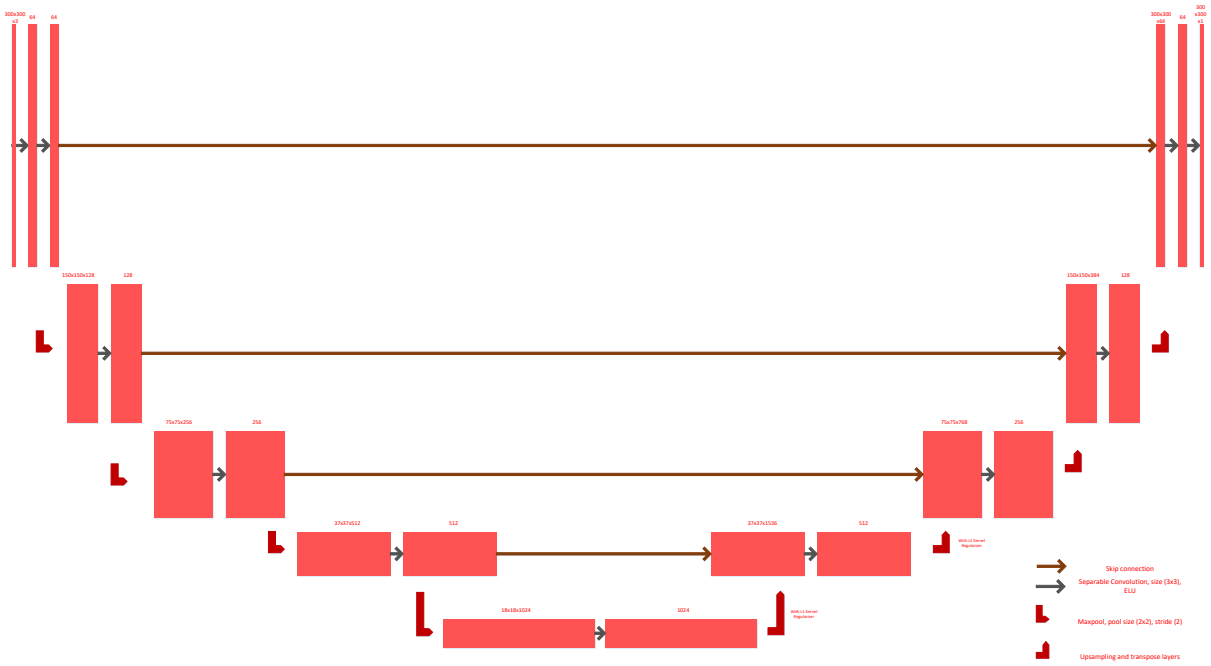

Figure 10: The U-NET inspired architecture used for segmentation.

Table 1: Names of TIFFs used from Planet.

TIFF NAME
L15-1469E-1189N
L15-1472E-1183N
L15-1479E-1186N
L15-1486E-1173N
L15-1490E-1173N
L15-1500E-1177N
L15-1500E-1180N
L15-1503E-1180N
L15-1504E-1184N
L15-1507E-1175N
L15-1509E-1161N
L15-1519E-1162N
L15-1522E-1162N
L15-1524E-1173N
L15-1527E-1165N
L15-1527E-1153N
L15-1529E-1156N
L15-1505E-1134N
L15-1460E-1193N
L15-1449E-1203N
L15-1446E-1200N
L15-1449E-1229N
L15-1495E-1173N

*The valdiation TIFF is marked in red.