Identification of River Defences from Digital Terrain Models using Deep Learning

David J. Wood^{1*}, Catharine R. M. Brown¹, Laura Doyle¹, Helen L. Smith¹, Simon Waller¹ and Elizabeth F. Weller¹

¹ JBA Risk Management, 1 Broughton Park, Old Lane North, Broughton, Skipton, BD23 3FD, UK

Abstract. Flood defences play a central role in the quantification of flood risk. JBA Risk Management produces undefended hazard maps that are supplemented with defence information to provide risk practitioners with the most flexible view of risk. This requires knowledge of the locations of river defences so that they can be removed from the digital terrain models prior to flood modelling. We report on work to develop a predictive model for identifying river defences. This model was created using the U-Net deep neural network for image segmentation. The model was developed over a series of iterations, where the prediction outputs were refined and used to retrain the model. We have used this model to produce national maps of defences for a range of countries.

1 Introduction

JBA Risk Management produces national-scale flood maps using the 2D hydraulic model JFlow (Crossley, 2010). The flood hazard maps are used primarily by the insurance and reinsurance industry to quantify flood risk and support decision making including for underwriting, pricing and accumulation assessment. Flood defences play a central role in the quantification of flood risk. Defences come in different forms, both natural and man-made, with the most common being raised river defences, such as levees and floodwalls. These defences typically run parallel to river channels, protecting urban areas by preventing high river flows that exceed bank-full capacity from spreading out onto the floodplain.

Raised river defences can fail during flood events. Without maintenance, defences deteriorate over time which can result in breaches of the defences in weakened areas. Even when the defences are in good condition, if river discharge exceeds the volume against which the defences are designed to withstand – their standard of protection (SoP), typically represented as a return period – the defence can be overtopped. To provide risk practitioners with the most flexibility and the worst-case view, JBA provides undefended river flood maps that are supplemented with defended areas: an additional vector dataset that outlines the areas protected by flood defences together with their standards of protection.

Modelling undefended flooding requires raised flood defences to be removed from the Digital Terrain Model (DTM) before hydraulic modelling commences, so that the hazard maps represent an artificial scenario that would occur if the river flows were unimpeded by the raised defences. This process requires the locations of the raised defences to be identified prior to modelling. Where available, third-party datasets are used as a starting point for defence information. For example, in the US, datasets such as the Federal Emergency Management Agency's National Flood Hazard Layer (FEMA, 2020) or the United States Army Corps of Engineers' (USACE) National Levee Database (USACE, 2020) provide information on the locations of thousands of river defences, although their coverage is not complete. Historically, JBA has supplemented third-party datasets by manually identifying

and annotating defences during the flood modelling process. While this is an effective way of identifying defences, relying on manual, human-led processes can be time consuming and lead to inconsistent results.

Over the past decade significant advances have been made in the field of computer vision and image analytics, in large part because the emergence of deep learning techniques. Deep learning describes a subset of machine learning that involves neural networks with many hidden layers of neurons (LeCun, 2015). These deep neural networks can learn features and abstractions directly from raw, unstructured data, such as images, textual information and sound files. The improvements in image analytics are because of a new family of neural networks known as Convolutional Neural Networks (CNN) (O'Shea & Nash, 2015). CNNs have led to improvements in the performance of image analytical techniques including image labelling, object detection within images, and image segmentation.

DTMs, and other spatial datasets, are typically stored in image-like raster formats, which makes them suitable for analysis with CNNs. In a competition hosted by Kaggle and sponsored by DSTL, competitors were asked to create maps of features such as roads, buildings, crops and waterways from an analysis of multi-spectral, high-resolution satellite imagery (DSTL, 2020). The U-Net neural network for image segmentation (Ronneberger, 2015) featured prominently in the top ranked solutions, including in the winning submission. Since then, there have been many applications of neural network-based segmentation models to earth observation and other spatial datasets (Lary, 2018; Paoletti, 2019).

This paper describes how we created a predictive model based on the U-Net neural network architecture to identify raised river defences in DTMs. We trained the initial model using a relatively small set of annotated 5km tiles of elevation data. Supplementary information describing river and transport networks was included in the input data. We describe how we refined the model over a series of iterations by applying it to new areas, then updating the training data and retraining the model. The model has been successfully used to map raised defences in Italy and Germany, where there are limited sources of public information on flood defences. These maps of flood

^a Corresponding author: david.wood@jbarisk.com

defences are used to remove raised defences from the DTMs as well as to create the ‘defended areas’ of the defences.

2 Methodology

2.1 Training Data

Our initial training dataset comprised 112 5x5km tiles located in the south of Florida, US, as shown in Figure 1. This region was chosen as it was being modelled by JBA, and because of the number of raised defences in the USACE and FEMA datasets that were also present in the DTM data. The tiles were defined from a regular, 5x5km grid with east-west bounds of 1382000 to 1542000, and north-south bounds of -1235000 to -1585000 in the North American Datum (NAD83, EPSG:102008) coordinate system. Each tile featured at least one defence present in both the DTM and in the third-party defence datasets and was represented by four input ‘X’ bands and one output ‘Y’ band (the target for the training). The input bands comprised a Local Relief Model of the DTM (described below) as well as bands for drainage lines, road and railways. The output band was the annotated river defences in a binary segmentation mask representation.

Figure 1. Florida grid used to tile training data with the tiles used for training shown in orange.

Digital Terrain Model: The DTM used was made up of the best available data and comprised of a patchwork of 5m resolution LIDAR and INSAR data.

Local Relief Model: A local relief model (LRM) is a derivative of a DTM that describes the relative local heights

of features in the terrain (Hesse, 2010). They are produced by deducting a smoothed version of the DTM from itself, which flattens out the ground and makes the local features (including raised defences) more pronounced. The DTM was smoothed with a 10x10 average rolling window, and the smoothed DTM was deducted from the original. The final LRM was normalised into a 0 – 1 range by applying a linear transformation so that 0 corresponded to local relative heights of -10m (or less), and a value of 1 corresponded to values of 10m (or more). An example of an LRM is shown in Figure 2 below. 2A shows the original DTM and 2B shows the derived LRM. The local topological features are visible in the LRM throughout the tile, and the raised embankments on either side of the rivers and drainage channels are also visible.

Figure 2. (A) The original DTM (low, blue through green, brown and white, high); (B) An LRM model produced with a moving average window of 10x10 cells (black reduced elevation through to white increase elevation); (C) Defence lines obtained from public sources; (D) A binary segmentation mask of the raised defences.

Drainage Lines: The river drainage lines were the same as those used in the flood modelling and were derived using proprietary GIS algorithms that use the DTM to calculate flow directions and accumulations and therefore enable stream delineation. They provide information on the location of rivers, which we considered to be important contextual information when locating flood defences. The drainage lines were rasterised onto a binary 5m resolution grid, with cell values of 1 representing the lines and 0s everywhere else.

Transport Networks: Vector line datasets representing roads and railways were included as inputs to the model. We felt that these layers were less important than the DTM and the drainage lines but may provide information about the nature of the features in the terrain model and help to prevent false-positive identification of infrastructure as flood defences. The transport layers were obtained from US Geological Survey (USGS, 2020) and rasterised onto separate 5m resolution binary rasters, using the same values as with the drainage lines.

Raised defences: Defences were primarily identified from known sources, including USACE’s National Levee Database (USACE, 2020) and FEMA’s National Flood Hazard Layer (FEMA, 2020). The river defences represent the target property for the machine learning. To turn the computational problem into an image segmentation problem, the raised defences were annotated as a binary grid where pixels that were part of the raised defence were assigned a value of 1 and all other pixels assigned values of 0.

The third-party defence lines were provided as line string geometries that needed to be converted to binary masks. We developed a tool called “Draw Levees” which creates the binary masks of the river defences with the following method. First, the LRM is converted to a binary raster by applying a relative height filter of 0.5m, and any small, fully enclosed holes in the resulting features are removed. Next, these rasters are vectorised and any polygons intersecting the defence lines are assigned to that defence line. Finally, the intersecting polygons are rasterised to a 5m resolution grid. An example of the output of this tool is shown in Figure 2 above. Figure 2B shows an LRM model of the area and Figure 2D shows the resulting mask for the raised defences.

As previously discussed, the third-party defence datasets were not complete in their coverage of defence locations. Furthermore, the use of incomplete defence annotations would have provided the neural network with an inconsistent definition of the raised defences, which would be expected to reduce the predictive performance of the model. We manually added in annotations for missing defences according to the following set of rules.

1. Include embankments along drainage channels
2. Include other embankments more than 2m high
3. Exclude embankments with railways
4. Exclude embankments with roads unless they are identified as a defence by a third-party source or manual research.

Defence lines that were not aligned with or not visible in the DTM were excluded. Raised defence binary masks were prepared for all training tiles.

2.1 U-net Neural Network

We based our neural network on the third-place ranking entry of the Eye in the Sky Kaggle competition (Igllovikov, 2017). The U-Net neural network is named as such because of the shape of the neural network, as can be seen in Figure 3. The left-hand side contains successive convolutional layers that summarise the contextual information in the image. The right-hand side of the network scales the tensor back up to be the shape of the input layer. ‘Skip connections’ links between the left-hand and right-hand side of the U allow spatial information to be preserved from the inputs to the outputs. This combination of contextual information and preserved spatial information makes the network highly effective at image segmentation tasks.

Training Set Batch Generation: Each training tile was represented by four input raster bands (LRM, Drainage, Road, and Rail) and one output band (the segmented raised defence binary mask). GPU memory limitations meant that we could not train the neural network on the full 1000x1000 5m-pixel tiles. As in Igllovikov (2017), we used data sampling and augmentation techniques to randomly sample 112x112 pixel tiles from the source tiles to train the neural network. This was done by choosing at random a sub-tile from a source tile and by randomly flipping the image along the x-, y- and diagonal axes. This process turns the limited number of 112 training tiles into a much greater number of unique training cases.

U-net Architecture: The implementation of U-net featured five down-sampling modules followed by five up-sampling modules. Each module comprised two convolutions, each with batch normalisation and ELU activation layers. The input layer is fed with the 112x112, 4-banded training images. The convolutional down-sampling modules reduce the images to a 7x7, 512-band tensor at the apex of the U before up-sampling it to the final 80x80 single band output image. A 16-pixel margin is removed from the output image to prevent boundary effects. The full network architecture is shown schematically in Figure 3. In total, the neural network contained 7,853,121 tuneable parameters that were initialised as random numbers between 0 and 1.

Figure 3. U-Net CNN architecture. Convolutional modules are shown as yellow boxes. Each module comprises two convolutional layers with batch normalisation and ELU activation. The max pooling steps halve the height and width of the tensor and the upsampling steps double the height and width. The number of filters for the convolutions (the number of output bands) is shown in the yellow boxes.

Loss function: A variation of the Jaccard coefficient was used as the loss function: the metric that is optimised during training. Raised defences are rare in the sense that most locations are not a part of a defence. This means that the pixel classification problem is unbalanced. The Jaccard coefficient (1), often described as the intersection over union, quantifies the level of agreement between predicted and observed positive class labels and is suitable for training models on unbalanced classes.

$$J(Y, \hat{Y}) = \frac{Y \cap \hat{Y}}{Y \cup \hat{Y}} \quad (1)$$

The intersection component of the coefficient represents the true positives, which is the set of pixels that are both predicted to be positive and annotated as positive in the training data. The union represents the set of all pixels that are either predicted or annotated as positive. A Jaccard coefficient value of 1.0 would indicate that the model has identified all the positive pixels in the training data and no others.

As with Iglovikov (2017), we used a generalisable form of the Jaccard coefficient as expressed in (2). This coefficient was combined with the binary cross entropy (3) to form the loss function that was used for optimisation of the model (4).

$$J_m = \frac{1}{n} \sum_{i=1}^n \frac{y_i \hat{y}_i}{y_i + \hat{y}_i - y_i \hat{y}_i} \quad (2)$$

$$H = -\frac{1}{n} \sum_{i=1}^n [y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \quad (3)$$

$$L = H - \log J_m \quad (4)$$

In addition to the loss function (4), we tracked the sensitivity ($TP/(TP + FN)$) and the specificity ($TP/(TP + FP)$) of the model during training, where TP is the number of true positives (positive labelled pixels predicted as positive), FP is the number of false positives (negative labelled pixels predicted as positive) and FN is the number of false negatives (positive labels predicted to be negative).

Training the model: The input dataset was divided on a 2:1 ratio into a training set of 74 tiles and a test set of 38 tiles. The test set was excluded from the training phase and used as a final test of the predictive performance of the model, allowing us to determine whether the model was generalisable to new tiles (within the same geographic area) and not overfit to the training data. We trained the model using the Nesterov Adam optimizer (Dozat, 2015) in two training phases, each comprising 50 epochs. The first phase used a learning rate of 1e-3 and the second phase used a learning rate of 1e-4. For each training epoch, 500 batches of 96 random tile samples (a total of 48,000 image samples) were fed through the neural network. The model training process took approximately eight hours on a NVidia GeForce GTX 1080 GPU.

2.2 Post-processing the model predictions

Predictions output by the model are bounded between 0 and 1, where values close to 1 are very likely to be raised

river defences and vice versa. We found that the model tended to be decisive, making predictions that were either close to 0 or to 1. To convert the raw predictions from the model into a useable spatial layer, we defined the following postprocessing routine:

1. The model's output predictions were converted to positive or negative classes by applying a threshold prediction value of 0.5.
2. The LRM raster for the same area was also converted to binary classes by applying a second threshold value. We used a threshold value of 0.5, which represents a height of 0.5m above the average local height.
3. For each feature (defined as an isolated cluster of positive pixels) in the classified LRM from step 2, we calculated the proportion of its pixels that were predicted to be defences (as defined the output of step 1). If this proportion was greater than a user specified threshold value, the LRM feature was written as a polygon geometry to the output spatial layer.

An example of the output of the post-processing is shown in Figure 4 below. The post-processing repaired gaps in the defences and produced complete defence features. These vector features can be used to remove defences from the DTM by interpolating the values of pixels surrounding the defence. We use this method to prepare our DTMs for undefended flood modelling.

Figure 4. Post-processing raw model predictions. (A) Raw model predictions with river lines shown in blue. (B) Post-processed predictions.

3 Results

Dataset	Jaccard	Sensitivity	Specificity
Training	0.73	0.88	0.85
Test	0.48	0.68	0.59

Table 1. Training and test set statistics for the initial model.

Table 1 shows a selection of training and test set statistics from the model at the end of the training phase. The Jaccard coefficient finished on 0.73 for the training set, with a sensitivity of 0.89 and a specificity of 0.85. The corresponding statistics on the test set were lower, as should be expected, but indicate that the model is generalisable to new cases.

Some examples of test set outputs are provided in Figure 5. These three examples are reasonably representative of the full test set, but also highlight some of the typical misclassification errors. The images show that the model tends to draw the raised defences cleanly. In example 1, a raised embankment in the south-east corner of the tile has been classified as a raised defence but wasn't annotated in the training data. Similarly, in example 3, an embankment on the side of a lake has been classified as a defence by the model but was also not annotated. Also, in example 1, a road passing through the tile received a weak label as a defence. This exercise highlighted to us the challenges in both defining rules and applying them consistently as to what should be labelled as a river defence.

4 Discussion

4.1 National-Scale Defence Maps

Test set statistics and training tiles provide an important insight into the behaviour of the model, but more important to us was the qualitative question of whether the model provided outputs that were beneficial to our flood modelling process when applied to full national regions. We applied the model to a larger catchment in the south of Florida. The output of this application of the model is shown in Figure 6. The model performed reasonably well, although there were some false positives; however, these could be manually cleaned up easily.

We wondered how the model would perform in a different region. The terrain of Florida is unusual and very flat. We optimistically applied the model trained with the Florida training data to Italy to see whether it could be used in completely different geographical regions. The model, perhaps unsurprisingly, didn't perform well and tended to classify alpine mountain ridges as river defences. We added 79 new 5x5km tiles from Italy to the original Florida-based training set. These included 39 tiles with raised defences and 40 tiles without any defence annotations. The 40 'negative' training tiles were focussed around the mountainous Alpine region in the north of the country. In addition, we included the DTM as a fifth input

Figure 5: Test set predictions. (A) LRM; (B) Transport networks including roads (orange), and railways (green), with drainage lines (blue); (C) Predicted defences; (D) Annotated defences.

Figure 6. Post-processed predicted defence map for a larger region of Florida (extent North=1,482,000, West=1,432,000, South=-1,535,000, East= 1,482,000.) (EPSG:102008)

band to the neural network (with a linear normalisation scheme bounded between 0 and 1,000), as we felt the terrain heights may provide further contextual information. After retraining the model, we generated a full national map of Italy for a second time. The number of false positives produced by the model was reduced significantly and the map of Italian river defences was a good starting point for defence locations in our modelling work. The post-processed Italian map is shown in Figure 7.

The map of raised defences in Italy was manually cleaned by our modellers during the flood map development projects. False positives were removed, which included anything that wasn't clearly a defence in both satellite imagery and the DTM. This map of defences in Italy was also used to derive a map of defended areas (the supplementary layers we provide that indicate the areas of the map protected by the defences).

The Final 'Italy' Model: The full, cleaned map of defences in Italy provided a large source of data for further model training. We added 1,944 5km tiles that contained defences to a new training set with 1,969 negative tiles that contained no defences. The model was retrained using

3,000 batches of 32 samples with the same batch sampling process as described above. This is equivalent to 96,000 training images per epoch. We were reassured by the test set analyses of previous experiments that our methodology produced a generalisable model. We decided to use the entire dataset as the training set and not to partition a separate test set. After the two phases of 50 epochs, the Jaccard Coefficient against the training set finished at 0.679 with a sensitivity of 0.79 and a specificity of 0.83.

Figure 7. Defence map of Italy: (A) The national layer; (B) The coastal region of the Po valley.

This predictive model is now used within our flood mapping processes for the purpose of identifying raised defences from our DTMs in areas where there are no third-party defence datasets available to us (or used in combination with third-party defence information). The model is generalisable to new regions with differing geographies, although we have encountered cases where the model fails to work as expected. Some of these cases are described below.

4.2 Pitfalls

Model Applicability Domain Issues: Application of the ‘Italy’ model to Germany highlighted further issues relating to the model’s applicability domain. The DTM models used in Germany comprised mostly INSAR 5m. However, LIDAR datasets were available and used for flood modelling in some urban locations. The LIDAR areas in the DTM were visible at a high level in this predicted defence map as the model tended to predict more features in these regions compared to the non-LIDAR regions. To resolve this issue, we created a separate LIDAR model by preparing around 100 training tiles from these areas. We have found the creation of new training datasets to be reasonably straightforward when the starting point is an unspecific model.

Creation of an Over-Sensitive Model: We attempted to refine the performance of the initial ‘Florida’ model by adding in defences that were not annotated in the original training data. This was because the initial model frequently highlighted smaller embankments, particularly in agricultural areas, that hadn’t been included in the original annotations. We felt that these smaller embankments should be removed for the undefended flood modelling. We updated the Florida training tiles with the new annotations and retrained the model. The resulting model was too sensitive for practical use and tended to label most roads as defences. We suspected that this over-sensitivity was due to the extremely flat, unvaried terrain in Florida, resulting in the LRM accentuating any and all rises in elevation. This model has not proven to be useful in our flood mapping processes.

5 Conclusions

We used modern deep learning techniques to train a CNN to detect raised river defences from digital terrain models. The approach proved successful, and over a series of training set updates, we developed a model that is effective at identifying raised defences and has become an important tool in JBA’s flood modelling processes. It is used to identify and remove defences from DTMs to allow us to create fully undefended flood maps. The defences that are identified by this approach are also turned into defended area maps that assist risk practitioners in assessing the flood risk of an area.

The model is often generalisable to new geographic areas, although when confronted with new sources of terrain data or new geographies, it can produce unexpected results. For example, when we applied our initial model to Italy for the first time, it identified alpine mount ridges as defences. Similarly, the well trained ‘Italy’ model was too sensitive and produced too many false positives when applied to a LIDAR section of Germany for the first time. Usually these issues can be resolved by adding new examples to the training data or by training a new model. Once we had a working first model and the associated software toolkit, it was straightforward to prepare new training data and create new models.

At JBA Risk Management we have continued to explore the use of deep learning and computer vision

techniques in our flood modelling pipelines. For example, for detecting and automatically fixing quality issues within our datasets.

6 References

1. Crossley, A., Lamb, R. & Waller, S. (2010). Fast solution of the shallow water equations using GPU technology. *Proceedings of the BHS Third International Conference, Newcastle, London: British Hydrological Society.*
2. Dozat, T. (2016). Incorporating Nesterov Momentum into Adam. *ICLR 2016.*
3. DSTL, K., 2020. *DSTL Satellite Imagery Feature Detection.* [Online] Available at: <https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection>
4. FEMA, 2020. *FEMA Levee Resources Library.* [Online] Available at: <https://www.fema.gov/fema-levee-resources-library>
5. Hesse, R. (2010). LiDAR-derived Local Relief Models – a new tool for archaeological prospection. *Archaeol. Prospect.*, pp. 67-72.
6. Iglovikov, V., Mushinskiy, S. & Osin, V. (2017). Satellite Imagery Feature Detection using Deep Convolutional Neural. *Computer Vision and Pattern Recognition*, Volume abs/1706.06169.
7. Lary, D. J. et al. (2018). Machine Learning Applications for Earth Observation. In: Mathieu, P. P. & Aubrecht C., ed. *Earth Observation Open Science and Innovation. ISSI Scientific Report Series, vol 15.* Springer, Cham, pp. 165-218.
8. LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *Nature*, **Vol. 521**, pp. 436–444.
9. O’Shea, K. & Nash, R. (2015). An Introduction to Convolutional Neural Networks. *arXiv*, p. 1511.08458.
10. Paoletti, M. E., Haut, J. M., Plaza, J. & Plaza, A. (2019). Deep learning classifiers for hyperspectral imaging: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, **Vol. 158**, pp. 279-317.
11. Ronneberger, O., Fischer, P. & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *Computer Vision and Pattern Recognition*, pp. 1-8.
12. USACE, 2020. *National Levee Database.* [Online] Available at: <https://levees.sec.usace.army.mil/#/>
13. USGS, 2020. *National Geospatial Programme.* [Online] Available at: <https://www.usgs.gov/core-science-systems/national-geospatial-program/national-map>