**Ministère de la digitalisation et l’Administration du cadastre et de la topographie du Grand Duché de Luxembourg**

# EXTOPIA TECHNICAL ARCHITECTURE REPORT ON DEEP LEARNING TECHNOLOGIES FOR BUILDING CHANGE DETECTION IN THE GRAND DUCHY OF LUXEMBOURG

www.gim.be
info@gim.be

Ubicenter D
Philipssite 5 bus 27
3001 Leuven
België

☎ +32 (0)16 40 30 39

RPR Leuven
BTW BE 0454.064.819
BNP Paribas Fortis
IBAN BE60 2300 3921 2470
BIC GEBABEBB

G.I.M. - Geographic Information Management nv

Doc ref.:
Doc issue.rev: 01.10
Doc date: 2021-02-25
Doc id: C201881/SS/ss/002

## Signatures

| | | |
|---|---|---|
| **Author** | Steven Smolders / Liam Moore | 2020-12-10 |
| **Reviewed by** | Steven Smolders | 2021-02-25 |
| **Approved by** | Steven Smolders | 2021-02-25 |
| **Issuing authority** | GIM | |

## Distribution list

| ORGANISATION | NAME |
|---|---|
| Administration du cadastre et de la topographie (ACT) | Mr. Jeff Konnen<br>Mr. Joe Mayer<br>Mr. Paul Mootz |
| Ministère de la digitalisation | Mr. Patrick Weber |
| | |
| | |

## Versions

| REASON FOR CHANGE | ISSUE.REV | REVIEWED BY | DATE |
|---|---|---|---|
| Initial document | 00.01 | SS | 2020-12-13 |
| Update after sprint 5 | 01.00 | SS | 2021-01-21 |
| Final Version | 01.10 | SS | 2021-02-25 |

# TABLE OF CONTENTS

# Management Summary

The subject report is composed in the context of the EXTOPIA project that was conducted by G.I.M.-Geographic Information Management NV. The EXTOPIA project is an innovation project funded by the Ministère de la digitalisation with as main stakeholder the Administration du cadastre et de la Topographie du Grand Duché de Luxembourg (ACT).
The aim of this project was to find the best possible Machine Learning approaches and set up a Proof of Concept (PoC) environment that allows to detect topographic objects like buildings on the basis of input imagery in varying resolutions as well as their changes, this ranges over a variety of visual changes, like new constructions, demolitions and adaptations by comparing yearly orthophotos. This PoC had to be operable by the ACT IT specialists in order to be able to set up a production system based on the PoC.

The work started from an existing architecture and processing pipeline that GIM developed on the basis of the DeepResUNet Deep Learning architecture for semantic segmentation.

The model was initially pretrained on Belgium data and applied on downsampled 1m resolution Luxembourgish orthophoto's and gave reasonable results.

During the project, an iterative approach was followed. Augmentation techniques were applied to improve the robustness of the segmentation. The architecture was then further improved using complex algorithmic enhancements as spatial attention gates, deep supervision, multi-scale pooling and Convolutional Block Attention Modules, which all improved the segmentation results. The final results were hence generated using a model with all of the above algorithmic enhancements applied and trained using a Luxembourg training dataset.

Experiments were conducted with 3 different loss functions. Weighted Binary Cross Entropy, Tversky and Dice. The experiments did show that both Tversky and Dice outperform the Weighted Binary Cross Entropy.  Which one to select depends on the specific use case. Tversky is the best choice if the idea is to map the maximum of uncharted buildings and obtain reasonable-quality footprints since its segmentation is less conservative than Dice. As a consequence, there will be also more false positives.  Dice on the other hand gives the best possible footprints and much less false positives but will also miss some buildings.

# 1 Introduction

## 1.1 EXTOPIA PROJECT

In the context of the continuous update and maintenance of the geographic databases managed by the ACT, there is a challenge to automatically identify changes with respect to the built-up environment on the basis of yearly acquired aerial imagery. The detection of newly constructed buildings, demolished buildings as well as buildings that have undergone significant changes in their geometry are all of high interest. Apart from changes to buildings, there are also other topographic features for which change detection needs to be performed as for instance forest walking paths and other natural topographic elements.

The ACT Department of the Grand Duchy of Luxembourg is maintaining a database of buildings covering the whole country. To keep the database up to date, the ACT is required to identify all newly constructed, demolished, or updated buildings. Orthophotos are acquired yearly, covering the entire country at a very high resolution. Currently, manual inspection and editing are the primary but very laborious approach to maintain the database. Recent advances in Earth Observation data processing technologies however demonstrate the feasibility of applying alternative techniques. In the frame of the "digital-first" program, the ACT Department is continuously looking for improving their processes by means of further digitalisation. In this regard, the ACT department is willing to start the introduction of AI-based technologies for the database maintenance on the building updates.

### 1.1.1 Project Objectives

Within the context mentioned above, the objective of the EXTOPIA project was to develop a Proof of Concept of an Artificial Intelligence (AI) based toolchain that can be used to identify building and other changes based on remotely sensed Earth Observation Imagery. This Proof of Concept had to be realised with state-of-the-art open-source tools. The project has to be seen as a highly innovative R&D project.

The ACT Department of Grand Duchy of Luxembourg is willing to set up an AI environment aiming to identify automatically the newly constructed, changed, or demolished buildings and later on for other topographical objects like walking paths in the woods. The project consists of a first step towards an automated workflow for the database update.

The main objectives of the project are the following:

- Provide an evaluation of the Deep Learning technologies that are most suited for the detection of objects on ortho-images, with a high potential for expansion to various semantic domains;
- Develop a Deep Neural Network (DNN) capable of being trained for detecting specific objects. More specifically, the model should be able to detect all kinds of buildings, in rural but especially also in urban situations and should be extensible to other objects;
- Develop an algorithm to support change detection on the build-up landscape between two years;

- Propose and implement an approach so that the ACT can improve the system by reporting detection errors.
- The Proof of Concept will be used by the IT Specialists of the customer as the basis for deploying a production ready system.

## 1.2 ACCRONYMS

| | |
|---|---|
| ACT | l'Administration du cadastre et de la topographie du Grand Duché de Luxembourg |
| AOI | Area of Interest |
| CBAM | Convolutional Block Attention Module |
| CNN | Convolutional Neural Networks |
| DCNN | Deep Convolutional Neural Networks |
| DSM | Digital Surface Model |
| FN | False Negatives |
| FP | False Positives |
| GSD | Ground sampling distance |
| IoU | intersection-over-union |
| IR | Infra-Red |
| OBIA | Object Based Image analysis |
| PCA | Principal Component Analysis |
| PoC | Proof of Concept |
| R&D | Research and Development |
| RGB | Red, Green Blue |
| TN | True Negatives |
| TP | True Positives |
| VHR | Very High Resolution |
| WBCE | Weighted Binary Cross entropy |

# 2 Architecture

## 2.1 THE BASICS: COMPUTER VISION USING DEEP CONVOLUTIONAL NEURAL NETWORKS

### 2.1.1 Introduction to DCNN

Deep Convolutional Neural Networks (DCNNs) are a type of neural network architecture designed to amongst other applications extract information efficiently from images. These DCNNs are applied heavily in computer vision for both supervised and unsupervised learning and are responsible for the explosive progress of the field in the last decade. Like other neural networks, DCNNs are typically trained by a gradient descent algorithm on a training dataset which may be labelled (containing “ground truth” values) or unlabelled depending on the objective. Within DCNNs there are a multitude of variations and architectural choices which can improve performance, speed and portability depending on several factors such as the particular kind of computer vision task at hand, the available hardware and data, and the circumstances under which the network will be deployed to generate results.

## 2.2 SEMANTIC SEGMENTATION

Semantic segmentation refers to the task of assigning every pixel in an image to one of several pre-defined categories. It is a generalisation of image classification where the algorithm outputs an abstract interpretation of an image rather than a single label which is the case in image classification. It is particularly useful in earth observation, where georeferenced rasters containing a deconstruction of raw imagery into classes like buildings, roads and trees can be generated and used for further analysis.

Due to the complexity of images, modern semantic segmentation algorithms are based on machine learning methods rather than engineered by hand. The most successful of these are deep convolutional neural networks.  Machine Learning and especially Deep Learning techniques are rapidly replacing the pixel based and Object Based Image Analysis techniques that are traditionally applied.

### 2.2.1 Methodology to be applied.

The methodology to be followed when training, evaluating and using CNNs for semantic segmentation roughly follows five main steps:

- DCNN architecture design: selecting and implementing the most appropriate architecture for the specific problem in mind, in this case building segmentation, based on an existing semantic and instance segmentation models that are documented in literature.
- Training, validation and testing dataset construction: a supervised task like semantic segmentation requires a certain amount of training and validation samples: images containing buildings with known footprints. Since the validation set is used to evaluate algorithm performance, these datasets must be fully separate. A wholly separate testing dataset should be created for final quality assessment to prevent accuracy

assessments from being biased toward models which were tuned to the idiosyncrasies of a particular validation dataset.

- Model training and hyperparameter optimization: the network is trained using the dataset described above. Hyperparameters of the model are then tuned by comparing the loss values obtained on the validation dataset using different configurations.
- Quality assessment: based on the independent testing dataset, a detailed assessment is performed where the footprint quality is quantified according to image-level metrics (such as intersection-over-union (IoU), precision and recall) and polygon-level metrics (per-building footprint IoU distributions, missed-building rates, false-detection rates). Based on these results the DCNN architecture and model parameters can be revisited in an iterative approach.
- Inference: the most performant deep neural network trained and evaluated in the previous steps is used to generate building segmentation masks on the full set of orthoimagery, where ground truth data need not be available.

## 2.3 THE STARTING POINT: DEEPRESUNET

The current architectural paradigm for semantic segmentation is the so-called “encoder-decoder” architecture, consisting of a stack of convolutional and downsampling layers intended to learn the important abstract features present at each location in an image (the encoder) and a stack of convolutional and upsampling layers intended to interpret these features and map them onto to the target classifications at each location (the decoder).

GIM has experimented since 2018 with several fully-convolutional encoder-decoder type network architectures, all recent evolutions of the well-known U-Net (Ronneberger, 2015)- itself a successor to SegNet (Badrinarayanan, 2017) which pioneered this design - which were highly-performant on public dataset benchmarks.

Our Deep Learning workflow has undergone several iterations and our current best results are with the so-called Deep Residual U-Net architecture, which we implemented based off the article Semantic Segmentation of Urban Buildings from VHR Remote Sensing Imagery Using a Deep Convolutional Neural Network, published in Remote Sensing in July 2019 (Yaning, 2019) . This is capable of producing high-quality segmentation masks (see Figure 1) on a single GPU.

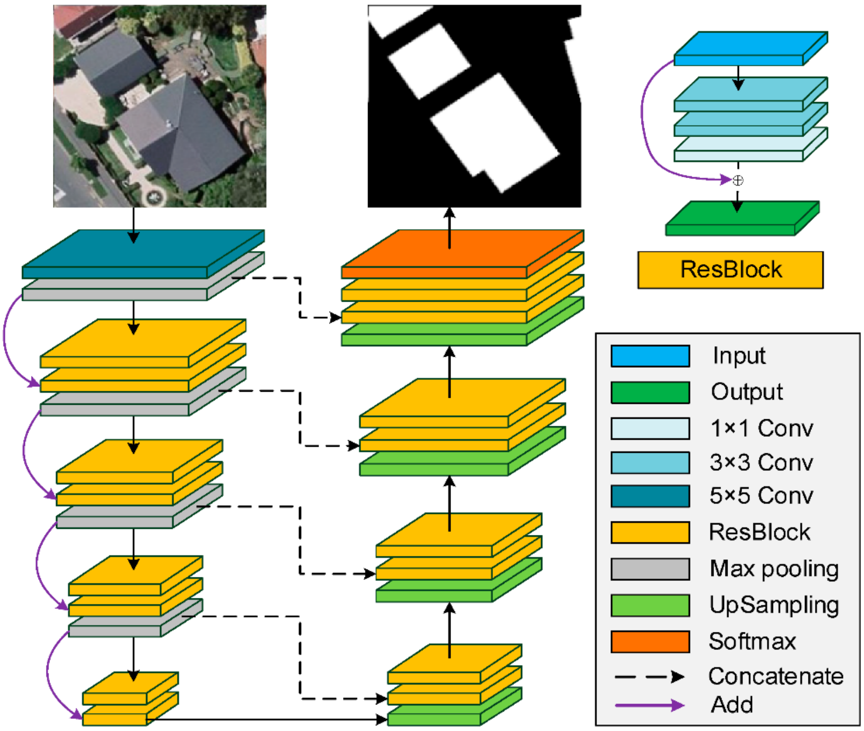


**Figure 1: "DeepResUNet" Model architecture based on the article Semantic Segmentation of Urban Buildings from VHR Remote Sensing Imagery Using a Deep Convolutional Neural Network currently implemented for building segmentation.**

Yi, Yaning et al. used this architecture for semantic segmentation of urban buildings from VHR remote sensing imagery. As shown in figure 1, it consists of a cascade down-sampling network that extracts building feature maps from the input VHR image and an up-sampling network that reconstructs the extracted building feature maps back to the same size as the input, followed by a softmax classifier. To improve accuracy considerably with increased layer depth and mitigate the issue of vanishing gradient, the architecture relies on a residual block (ResBlock) as the basic processing unit. Skip connections feed the encoder feature maps to the decoder feature maps at each intermediate spatial resolution and are intended to allow the network to progressively recover spatially finer information when constructing the output segmentation masks. The proposed architecture was evaluated with six other deep learning approaches on a dataset of aerial images covering an urban area of Christchurch City, New Zealand. With fewer false negatives and false positives, DeepResUnet outperformed the other six approaches in the semantic segmentation image of urban buildings. It also had fewer parameters than most of the models in competition but required a longer training and inference time.

## 2.4 FRAMEWORK OVERVIEW

### 2.4.1 Overview of functionality

The PoC tool developed in this project facilitates training and evaluating DeepResUNet-based segmentation models on geospatial datasets consisting of VHR RGB orthoimagery and vector or raster ground truth data (building footprints) and running inference with these models on unseen orthoimagery.

### 2.4.2 Libraries

The processing pipeline is built on open-source libraries such as:

- Tensorflow – implementation of segmentation models
- Dask – data engineering and facilitating processing of larger-than-RAM datasets
- Sci-kit learn – preprocessing and data engineering
- Rasterio – raster manipulation and data extraction
- Geopandas – polygon-level analysis of segmentation results
- GDAL – dataset preparation, rasterization and polygonization

### 2.4.3 Processing chain

- Input datasets may be declared which consist of georeferenced RGB rasters, with the option to provide ground truth polygons either in vector format with ESRI shapefiles, or directly as binary rasters with the same georeferencing and transform as the corresponding image.
- Preprocessing consists of extraction of raw image arrays from the input data files and their division into normalised patches suitable for consumption by the CNN model.
- Training facilitates feeding the processed input datasets to a segmentation model such that it learns to reproduce the ground truth masks. Here options are provided for tuning and enhancing this process, such as configurable image augmentations (operations that realistically modify, crop or rotate images at training time to enlarge the input data) and hyperparameter selection.
- Evaluation provides a suite of options for evaluating the performance of a trained model on sample Area-of-Interest (AOI) testing datasets, such as segmentation quality metrics and polygon quality metrics.
- Inference allows trained models to be run on any other declared raster datasets, producing matching GeoTIFF rasters containing predicted building probability values at every pixel.

### 2.4.4 Hardware

All results were generated, and experiments carried out on a machine with an NVIDIA Tesla V100 accelerator, an 8-core CPU and 32GB of RAM.

## 2.5 NEWLY DEVELOPED FUNCTIONALITY

Here we describe the features that were implemented in the course of the project which extended and improved the base framework existing at the conception of the project.

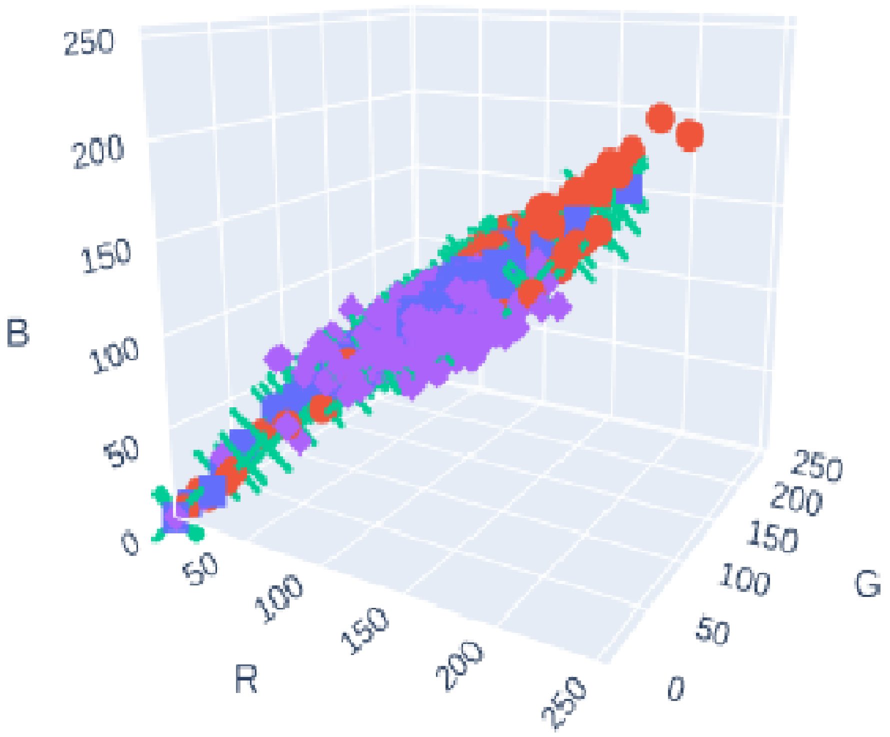
### 2.5.1 Image augmentations

Image augmentation refers to the process of transforming images in realistic ways. This procedure is used widely in machine learning applied to computer-vision to artificially enlarge training datasets and make models more robust to training datasets with different statistical properties. The most basic example might be applying a horizontal or vertical flip to an image. Another example might be changing the brightness or contrast. When such transformations are applied during model training, slightly different versions of an image will be seen during each epoch, and the model is taught to anticipate these “natural” statistical variations, which in turn improves the capability of the model to generalise.

We expanded our model training pipeline from basic augmentations (horizontal and vertical flips applied during training) to more sophisticated hand-engineered and data-driven augmentations. These were implemented by utilizing the Albumentations[1] library. The following existing augmentations were integrated:

- Random rotation by 90 degrees. This teaches a model a degree of rotation invariance.
- Random horizontal flip. This teaches a model a degree of reflection invariance.
- Random homogenous RGB offset with a maximum radius of 15. This teaches a model to be robust to small colour normalization shifts in any direction.
- Random affine transformations (a small rotation and a small translation together). These can provide both a degree of rotation invariance and small contextual shifts of objects.
- Random gaussian noise (adding random fluctuations centered about zero to the pixel values). This will teach a model a degree of robustness to noisy images.
- Random gaussian blur. This will provide a degree of robustness to images slightly out-of-focus.
- Random contrast shifts. Provides flexibility to sensors producing images with differing contrasts.
- Random brightness shifts. Provides flexibility to images of varying brightness.
- Random gamma corrections. Provides flexibility to images of varying gamma values.

[1] https://github.com/albumentations-team/albumentations



-

- **Figure 2 – Mean RGB values for images taken from Luxembourg Belair training area for four years. Note certain datasets contain more images with brighter values and the colour distributions are dependent on the conditions on the day of capture. The lone winter orthophoto (purple) in particular is dimmer and has a slightly broader colour distribution. Augmentations such as "FancyPCA" will shift the colour normalisation in other images towards these and vice versa, allowing a model to more easily learn common features across different datasets.**

In addition to these, a data-driven PCA-based colour augmentation (dubbed "Fancy PCA") based on (Krizhevsky, 2012) was implemented to provide colour shifts along the principal colour axes of the training datasets. This creates more "realistic" shifts in colour normalisation and allows the training procedure to distort the colours of images seen at training time towards those of other samples in the training dataset.

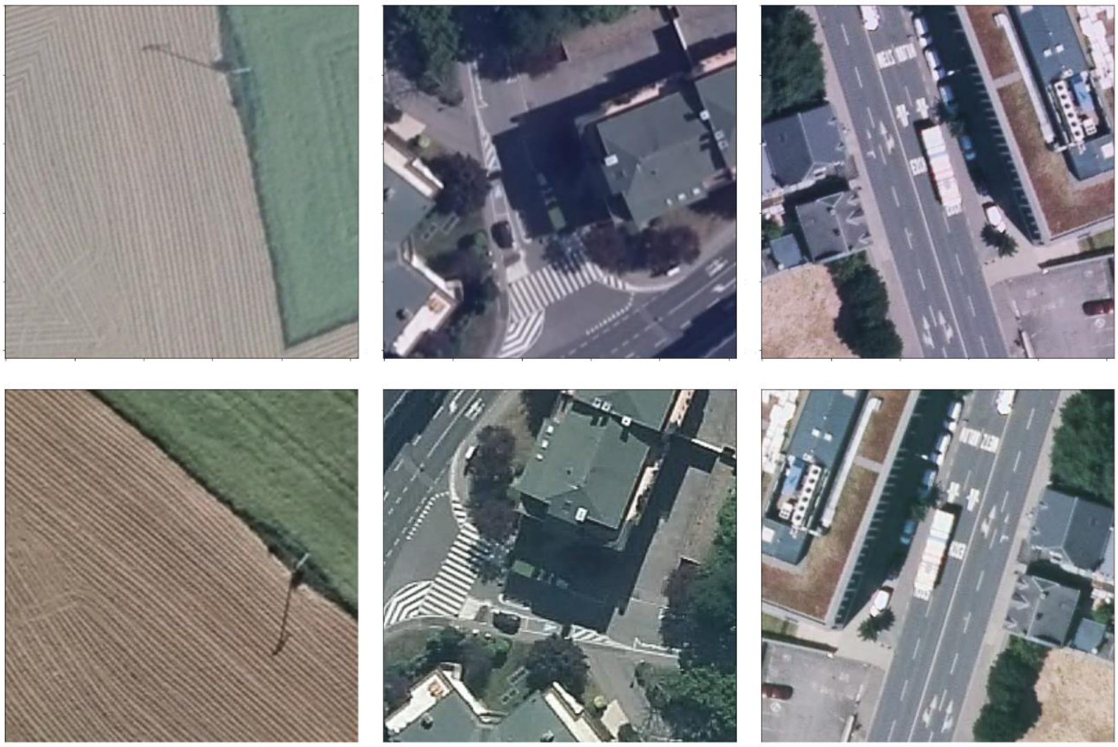


**Figure 3 – Example of augmentations on matching patches of the Luxembourg Belair training dataset as seen by the model. The colour normalisation, brightness and sharpness along with the spatial orientation of each image differs in multiple configurations during training.**

In initial experiments with 1m spatial resolution data we observed ~15% lower loss values on validation data with extensive augmentations enabled compared to the baseline.

In all subsequent experiments and in generation of results, the extensive augmentations described above were used. In Figure 4. we show an example of two models trained with and without this augmentation scheme to display the qualitative difference in segmentation mask quality.

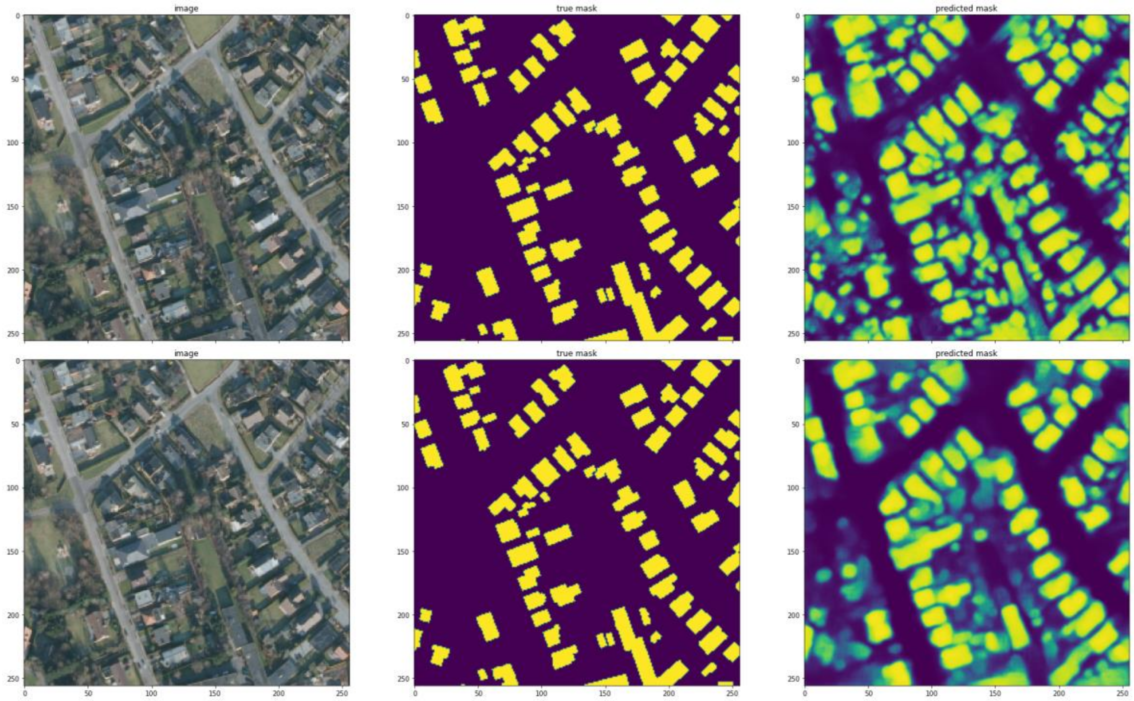


**Figure 4 - Examples of baseline DeepResUNet models trained on 1m orthoimagery without (top) and with (bottom) augmentations. Source image (left), Ground Truth (middle) and predicted (right) are shown.**

### 2.5.2 Algorithmic improvements to DeepResUNet

A number of architectural enhancements were implemented to improve the predictive power of the DeepResUNet architecture in order to derive the sharpest possible segmentation masks.

These features take the form of additional modules in the neural network architecture and were enabled independently and concurrently during an evaluation phase in which the best model was selected.

#### 2.5.2.1 Spatial Attention Gates

Spatial attention gates provide a mechanism through which the important regions of the encoder feature maps can be enhanced or suppressed depending on the more abstract semantic content of the decoder at the corresponding spatial locations.

The Spatial Attention Gate modules implemented are described in the paper Attention U-Net: Learning Where to Look for the Pancreas (Oktay, 2018).

These use kernel-size one convolutions to project each set of encoder feature maps into a new space of “key” vectors at each spatial location, and decoder feature maps into a new space of “query” vectors at each spatial location. These feature maps are then used to derive an (additive) attention map by adding these and applying a further 1D convolution with sigmoid activation. This is then multiplied with the encoder feature maps, adaptively rescaling them according to the content of the decoder feature maps.

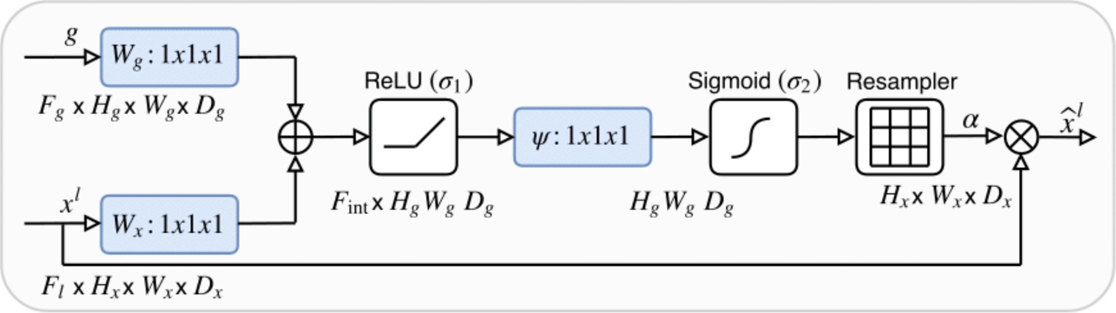


**Figure 5 - The Spatial Attention Gate module. x and g are feature maps coming through the encoder skip lines and from the deeper decoder stages respectively. These are mapped into a key, query space with kernel-size 1 convolutions, added and a ReLU nonlinearity applied to the result. A final convolution with sigmoid activation is used to generate a spatial attention map which then reweights x .**

The spatial attention gates are inserted immediately after each decoder upsampling block and intercept the encoder feature maps through the skip lines before these enter the next decoder block.

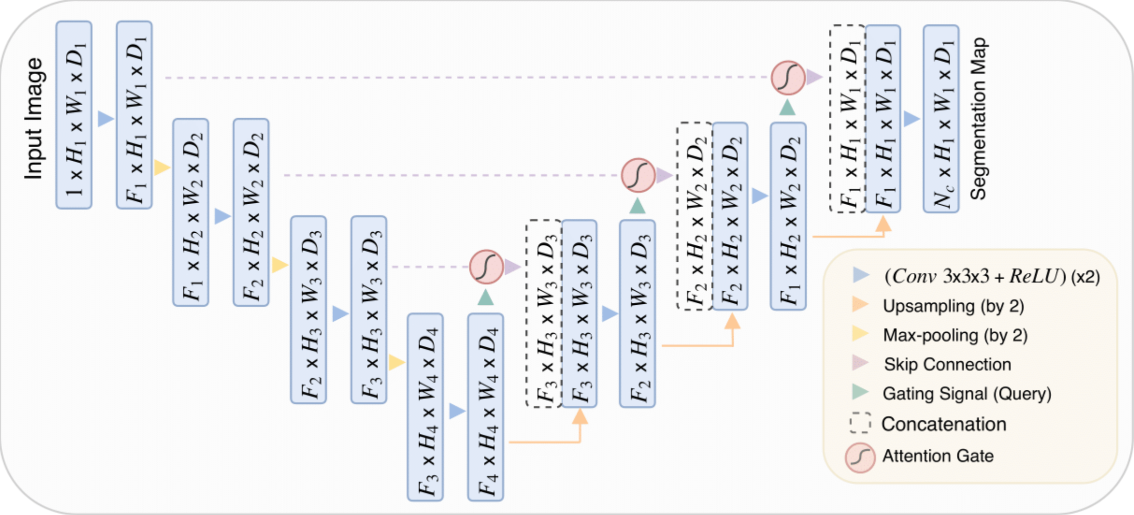

Input Image
1 x $H_1$ x $W_1$ x $D_1$
$F_1$ x $H_1$ x $W_1$ x $D_1$
$F_1$ x $H_2$ x $W_2$ x $D_2$
$F_2$ x $H_2$ x $W_2$ x $D_2$
$F_2$ x $H_3$ x $W_3$ x $D_3$
$F_3$ x $H_3$ x $W_3$ x $D_3$
$F_3$ x $H_4$ x $W_4$ x $D_4$
$F_4$ x $H_4$ x $W_4$ x $D_4$
$F_3$ x $H_3$ x $W_3$ x $D_3$
$F_3$ x $H_3$ x $W_3$ x $D_3$
$F_2$ x $H_3$ x $W_3$ x $D_3$
$F_2$ x $H_2$ x $W_2$ x $D_2$
$F_2$ x $H_2$ x $W_2$ x $D_2$
$F_1$ x $H_2$ x $W_2$ x $D_2$
$F_1$ x $H_1$ x $W_1$ x $D_1$
$F_1$ x $H_1$ x $W_1$ x $D_1$
$N_c$ x $H_1$ x $W_1$ x $D_1$
Segmentation Map
(Conv 3x3x3 + ReLU) (x2)
Upsampling (by 2)
Max-pooling (by 2)
Skip Connection
Gating Signal (Query)
Concatenation
Attention Gate

**Figure 6 - A U-Net with spatial attention gates (red circles) positioned before each intermediate decoder block. Our architecture provides the option to insert these in the corresponding position in the DeepResUNet model.**

These have been demonstrated to improve encoder-decoder type models in various segmentation tasks.

#### 2.5.2.2 Deep Supervision

Deep supervision is a mechanism to force each decoder block of the network to take on a more concrete role, namely learning to produce outputs which more directly correspond to the target segmentation map at that block's spatial resolution. It is known to improve results and speed up training in segmentation tasks and is used heavily in biomedical imaging.

The version of deep supervision implemented follows the version described in the paper Improving CT Image Tumor Segmentation Through Deep Supervision and Attentional Gates (Tureckova, 2020).

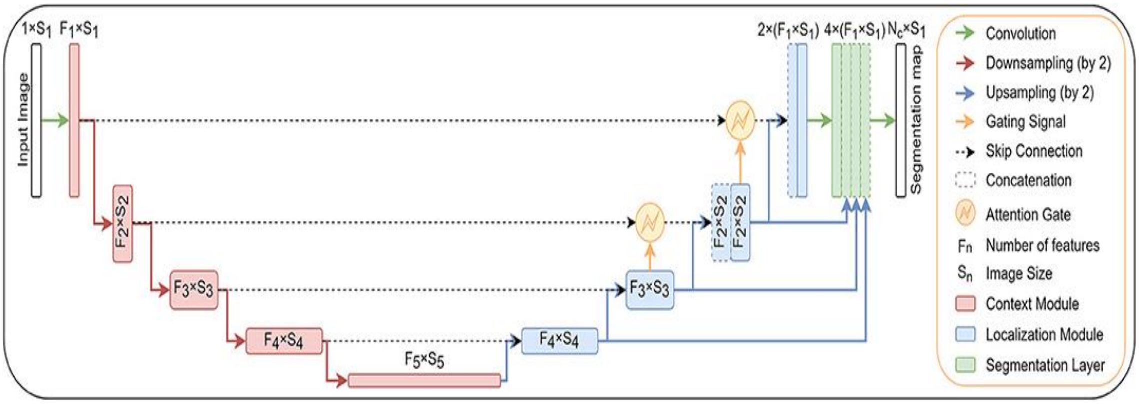


**Figure 7 - A Deep Supervision network implemented in (Tureckova, 2020) - each decoder output feature map (blue) is projected into the same space used to generate the final segmentation map (green) and these are added (with upsampling).**

The version implemented captures the output of the intermediate decoder feature maps and projects these into the same (channel) space as the final output feature maps used to generate the segmentation (before application of softmax/sigmoid). The segmentation map produced is calculated by applying the final activation to the sum of all of these intermediate decoder feature maps upsampled to the same resolution as the final decoder feature map, so that each decoder block learns to make a direct contribution to the output class probabilities.

Note that other versions of deep supervision directly generate multiple output segmentation maps (at different spatial resolutions) and train directly on the ground truth mask resampled to match these.

Deep supervision has been shown to speed up training and provide small performance benefits.

#### 2.5.2.3 Multi-scale pooling

Multi-scale pooling can have slightly different meanings depending on the context. The version implemented here makes the input image available to the network encoder at multiple spatial resolutions through downsampling. This way each stage of the network encoder has access to both the feature maps of the previous encoder layer (which access the original image indirectly) and to the image itself directly, resampled to the spatial resolution of each encoder block.

Input pyramid pooling is implemented as in the paper A Novel Focal Tversky loss function with improved Attention U-Net for lesion segmentation (Abraham, 2019).

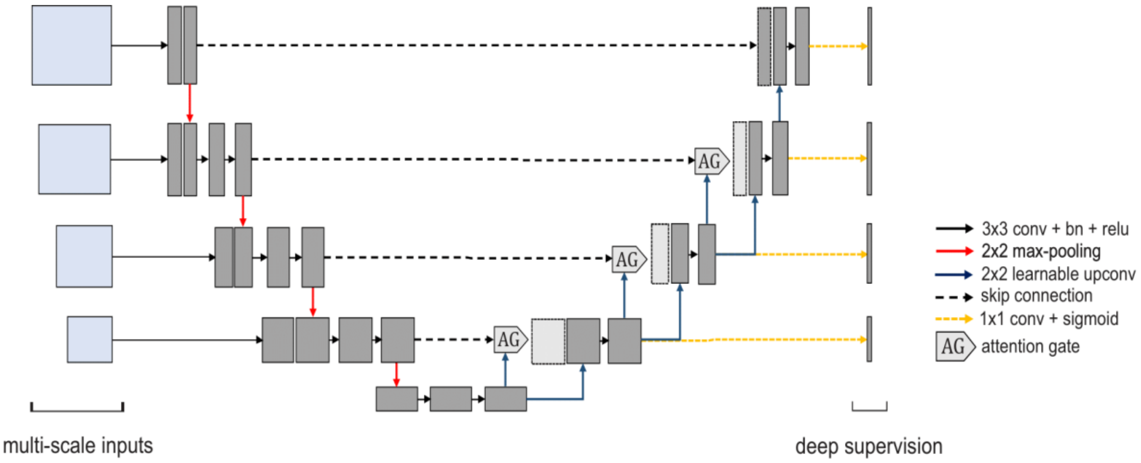


**Figure 8 - Network architecture with input pyramid pooling (blue-gray blocks on the left), spatial attention gates and (another variant of) deep supervision as implemented in (Abraham, 2019). The input image is downsampled and fed directly to each encoder block at intermediate stages of the network.**

Each intermediate encoder block is preceded by an additional set of convolutional filters which generate feature maps from the coarsened input image. These coarse feature maps are concatenated with the output of the previous encoder block and these together form the inputs of the encoder block.

In principle this addition should allow the model to learn to leverage aggregated larger-scale information in the input image. This has been shown to provide small performance benefits.

#### 2.5.2.4 Convolutional Block Attention Module (CBAM)

The CBAM module is in essence a simplified form of self-attention which enables a set of feature maps (from the output of a residual block) to calculate channel and spatial attention maps for reweighting themselves, enhancing important channels and spatial regions based on the global feature map distributions.

CBAM is implemented according to the paper CBAM: Convolutional Block Attention Module (Woo, 2018).

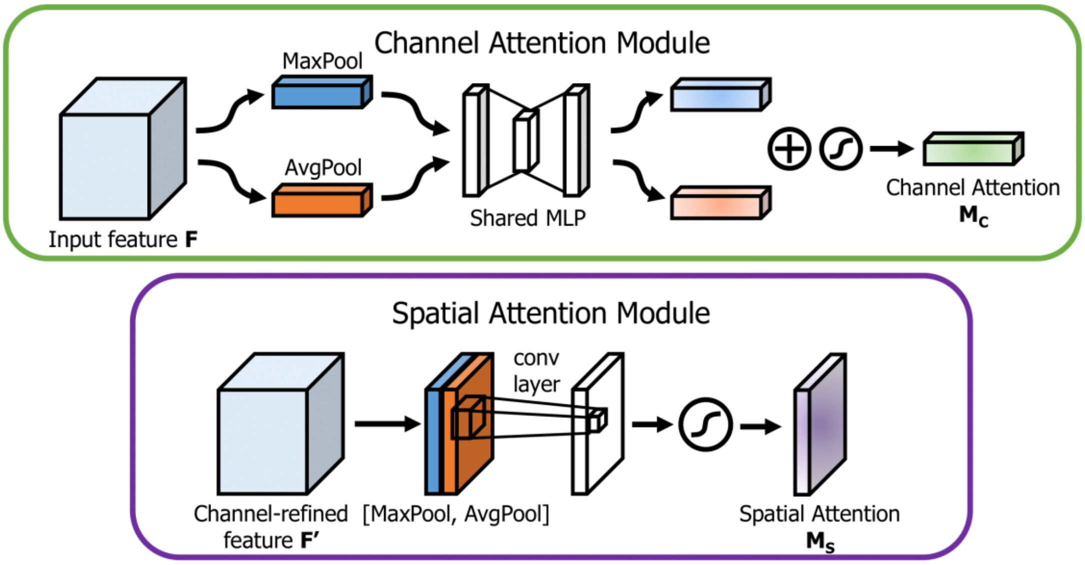


**Figure 9 - The channel and spatial attention modules making up the CBAM. The channel module passes a vector of the max and average values of each channel across the whole spatial domain through an MLP and learns a channel-wise reweighting with a sigmoid activation function. The spatial attention module performs a global max and average pooling operation to derive two channel feature descriptors across the spatial extent of the feature maps which are passed through a sigmoid convolution to derive a spatial attention map. The kernel size of the final convolution is a hyperparameter which provides a degree of context-awareness in the derivation of the attention values.**

The channel and spatial attention maps reweight the feature maps sequentially, i.e. the input feature maps are first used to calculate channel attention maps, the channels are reweighted, and these reweighted feature maps are used to calculate spatial attention which reweights the whole set (spatially) one more time.

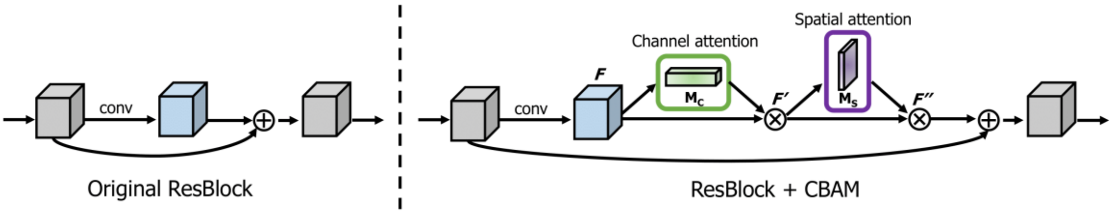


**Figure 10 - The CBAM block as positioned in a residual block; i.e. as a feature map postprocessing step.**

These blocks are quite lightweight (due to the channel and spatial pooling operations) and the dynamic reweighting capability has been shown to lead to performance gains in various computer vision tasks.

### 2.5.3 Evaluation of different loss functions

During our experiments, we trained models to produce segmentation masks guided by three different loss (or “objective”) functions. These differ in how they measure the quality of a predicted segmentation mask and prioritise slightly different objectives. The original loss function utilized in the segmentation framework, the adaptive weighted binary cross entropy, was extended by two additional loss functions for this project. Each of these is described below.

#### 2.5.3.1 Adaptive Weighted Binary Cross Entropy

This loss function (for shorthand, “WBCE”) is a generalization of binary cross entropy to unbalanced data (different numbers of foreground and background pixels, here buildings and non-buildings). The loss function is as follows:

$$L_{WBCE} = -\frac{1}{N}\sum_{\mathrm{n=1}}^{\mathrm{N}} \mathrm{wy_n} \log \mathrm{p_n} + (1-\mathrm{y_n})\log(1-\mathrm{p_n})$$

Here $N$ denotes the number of pixels in the image indexed by $n$, $y_n$ the binary building probability in the ground truth (0 or 1) and $p_n$ the foreground probability output by the network. The weight $w$ is derived on a per-image basis according to the formula:

$$\mathrm{w} = \frac{\mathrm{y}_- + \epsilon}{\mathrm{y}_+ + \epsilon}$$

Where $y_- = \sum_{n=1}^{N}(1-y_n)$ and $y_+ = \sum_{n=1}^{N} y_n$, and represents the relative fraction of the image occupied by background with respect to foreground. This rebalances the loss adaptively so that erroneously labelled foreground pixel values are punished more heavily if they occupy a smaller total portion of the image, and the degree of extra punishment is equal to the ratio of background to foreground pixels. ϵ is a small numerical factor ~ 0.00001 to prevent division by zero in case there are only background pixels in the image.

Since this loss is linear in the pixels (the summation over n), it does not explicitly address the “correctness” of global structure, only individual pixels. This results in a degree of “fuzziness” in predicted segmentation masks which must be mitigated by thresholding. It Is nonetheless a commonly used function for training image segmentation models. The upside of this situation is that the gradients of the loss function are relatively simple which makes training a model with this loss function easier.

#### 2.5.3.2 Dice Loss

The Dice loss function is based on the Dice coefficient, a segmentation quality metric, which applied to the binary case takes the form:

$$D = \frac{TP}{TP + 1/2(FP + FN)}$$

Where TP, FP and FN refer to True Positives, False Positives and False Negatives respectively. Since these strictly speaking only exist for binary variables (the neural network outputs continuous probabilities $p_n \in [0,1]$), continuous-valued proxies are used to these values:

$$TP = \sum_{n=1}^{N} y_n p_n$$

$$FP = \sum_{n=1}^{N} (1 - y_n) p_n$$

$$FN = \sum_{n=1}^{N} y_n (1 - p_n)$$

The loss function itself is simply:

$$L_{Dice} = 1 - D$$

Since this loss function is nonlinear in the pixels (its value depends on the global properties of the segmentation since it uses the total frequency of true and false positives and false negatives to derive a summary ratio) it produces qualitatively different masks to the weighted binary cross entropy, and these tend to contain more contiguous objects and very little “fuzz”. It also intrinsically weights false positives and negatives the same. The downside of its more complicated functional form is that the gradients of the loss are more complex which can make training models with this loss slower and more difficult.

#### 2.5.3.3 Tversky Loss

The Tversky loss is a generalization of the Dice loss based on the Tversky Index. This allows one to inject weighting factors which pushes a model toward performing better at reducing false positives or false negatives. The Tversky index has the form:

$$\mathrm{T} = \frac{TP}{TP + \alpha FN + \beta FP}$$

Where $\alpha, \beta \in [0,1]$ are weighting factors subject to the constraint $\alpha + \beta = 1$. It reduces to the Dice loss when the weighting factors are equal to one half.

The Tversky loss is simply:

$$L_{Tversky} = 1 - T$$

In our experiments we used $\alpha = 0.7$ and $\beta = 0.3$ to guide models to prioritise minimizing false negatives (i.e. to not miss pieces of buildings). This should result in more accurate footprints for buildings that are detected, but at the cost of increasing the rate at which spurious buildings are detected (false positives).

The Tversky loss, and gradients thereof, have a slightly more complex functional form owing to the different weighting factors than the Dice loss, which renders training models with this more difficult.

### 2.5.4 Polygonisation

In addition to exploring image-level segmentation quality metrics, binary segmentation masks were converted into polygon-level results using GDAL's polygonisation algorithm (based on pixelwise 4-connectedness). This allows a direct comparison of ground truth vector data and predicted vector data, and the identification of buildings segmented by a given model with buildings present in ACT's building database.

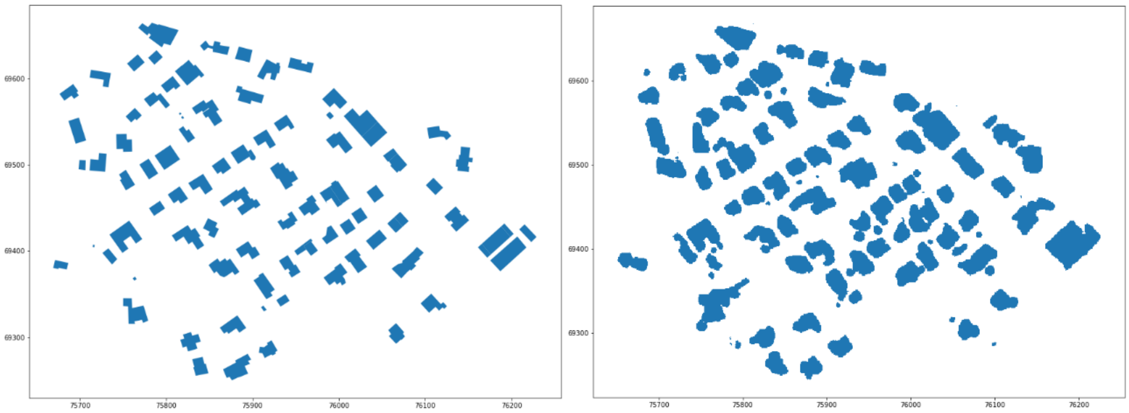


**Figure 11 - Polygon-level results (right) compared to ground truth (left) for AOI 3 generated by the baseline 1m spatial resolution model with no architectural enhancements.**

A number of quality metrics and classifications were developed for quantitative comparison of polygon-level results and integrated in the codebase for future use by ACT for evaluating model performance. These will be shown in section 3.

## 2.6 DATASETS USED

### 2.6.1 ACT orthoimagery

The main training datasets used throughout experiments consisted of a cut-out of four 20cm GSD orthophotos and corresponding ground truth building polygons provided by ACT.

These were obtained from the 2017 (summer), 2018 (summer) and 2019 (summer) regular orthophoto campaigns along with the 2019 (winter) "true" orthophoto with zenith angle of ~90 degrees. These each covered the same region of Belair, and each spanned in the region of ten square kilometres.

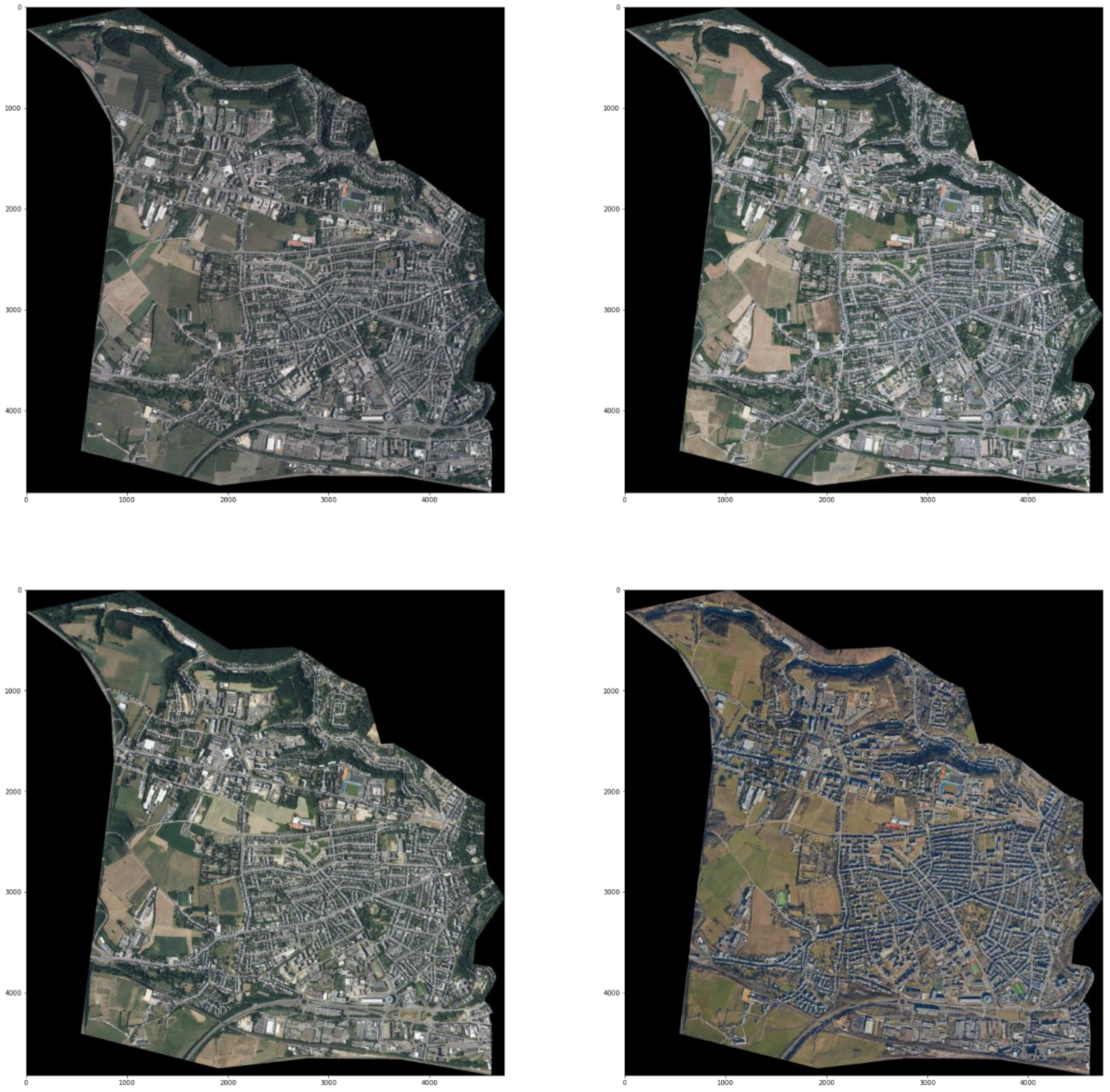


**Figure 12 - Belair training area taken from the 20cm GSD ACT orthophotos. Top: 2017 (summer), 2018 (summer). Bottom: 2019 (summer), 2019 (true, winter)**

### 2.6.2 Belgium orthoimagery

In initial experiments before the Belair training dataset was prepared, preliminary results were calculated based on a DeepResUNet model trained on 2015 and 2016 (winter) orthophotos of Belgium (25 cm GSD, downsampled to 1m for Flanders and Wallonia). Ground Truth building data was provided by GIM.

### 2.6.3 ISPRS Potsdam

The ISPRS Potsdam Dataset is a publicly available 5cm GSD IR+RGB true orthophoto + DSM segmentation dataset with annotated ground truth for buildings (alongside other objects such as vegetation, impervious surfaces and cars). A 20cm binary building segmentation dataset was extracted from this by downsampling and discarding the additional classifications to match the structure of the ACT Belair training dataset.

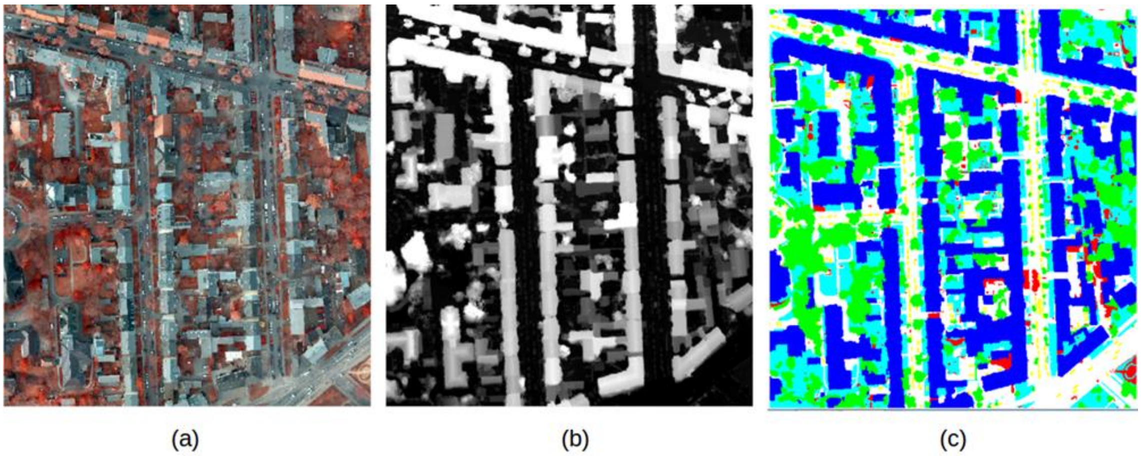


**Figure 13 - The ISPRS Potsdam segmentation dataset (left, IR-R-G bands) with ground truth building classes annotated in blue (right). A DSM is also included (middle) which was not used here.**

### 2.6.4 DSM

ACT also made available a DSM of the country of Luxembourg based on a LiDAR flight. This was in the end not used for the following reasons:

- DSM capture frequency is infrequent and out-of-sync with respect to the yearly-orthophoto captures. This would limit the applicability of the prospective data fusion model and lead to a situation where the increased accuracy could only be taken advantage of every few years.

- Based on existing literature performance gains would only be at the few percent level.

Given the limited applicability and performance gains expected from a complex data fusion orthophoto + DSM segmentation model and the major effort required to implement it, it was decided to not implement it in the course of this project.

# 3 Overview of experiments performed

## 3.1 AREAS-OF-INTEREST

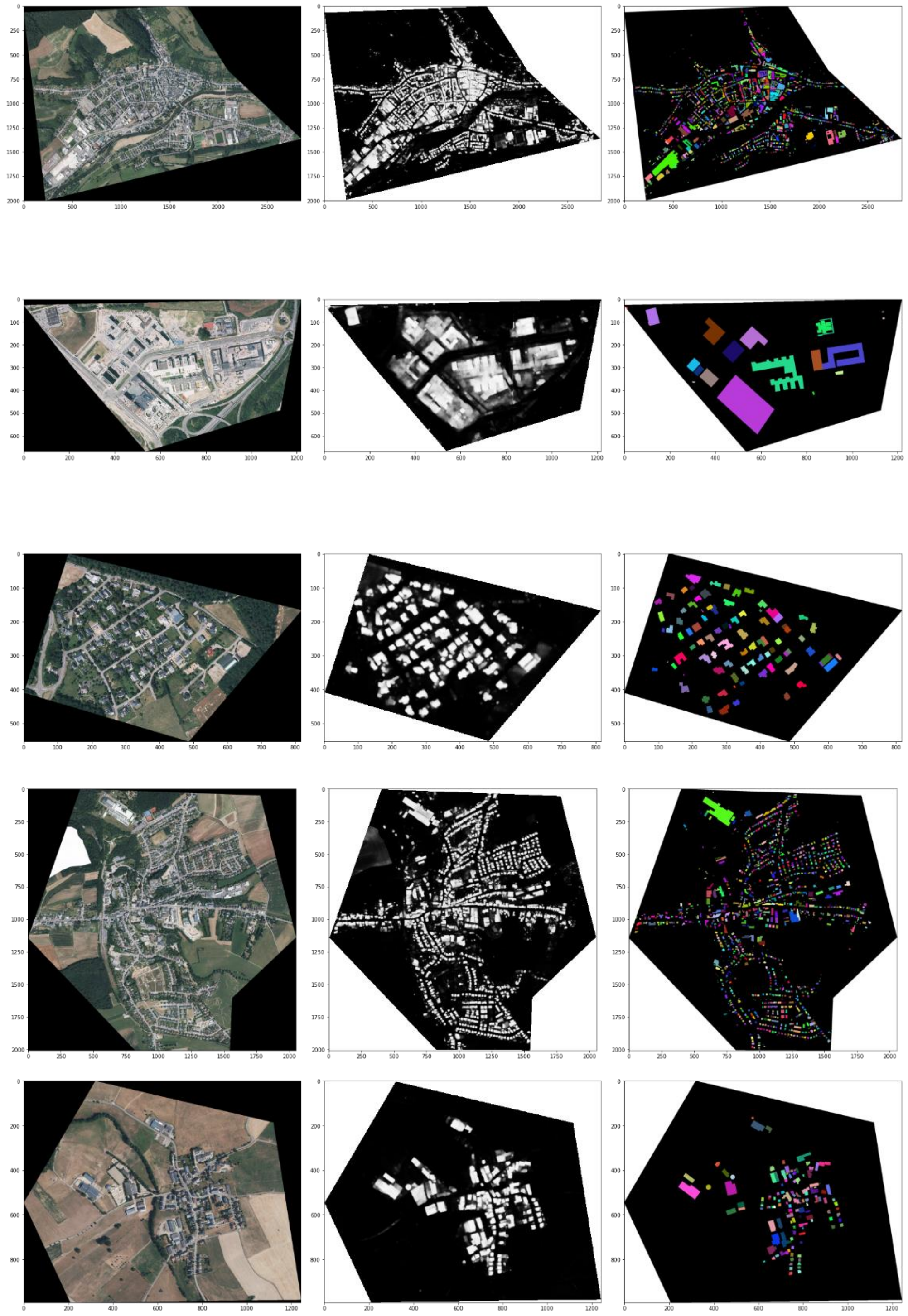


**Figure 14 – AOIs 1-5 (top to bottom, left) along with the initial 1m baseline DeepResUNet model results with weighted binary cross entropy (middle) and ground truth (right).**

All models were benchmarked on a testing dataset of five AOIs chosen by ACT to represent diverse urban and rural buildings across Luxembourg.

Ground truth data for these AOIs was accurate as of 2018, although orthoimages were available for subsequent years (where new buildings may have appeared or old ones disappeared). We show results and metrics for this year since the ground truth is most reliable here.

## 3.2 EXPERIMENTS AT 1M SPATIAL RESOLUTION

In the beginning of the project baseline results were generated using an existing unmodified DeepResUNet model trained on Belgian orthophotos from 2015 and 2016 downsampled to 1m using the weighted binary cross entropy loss function.

### 3.2.1 Architecture

The architecture used to generate the initial results was the baseline DeepResUNet described in (Yaning, 2019).

### 3.2.2 Training data

Training data consisted of Belgian publicly available orthophotos from 2015 and 2016 with ground truth building polygons provided by GIM. These were relatively low resolution, being resampled from 25cm GSD to 1m GSD before training. The model was trained using the weighted binary cross entropy loss function, using a random 10% of the patches making up these datasets as validation data. Initial learning rate was set to 0.001 with a decay of 50% after two epochs of validation loss stagnation until a minimum of 10^-5.

### 3.2.3 Inference data

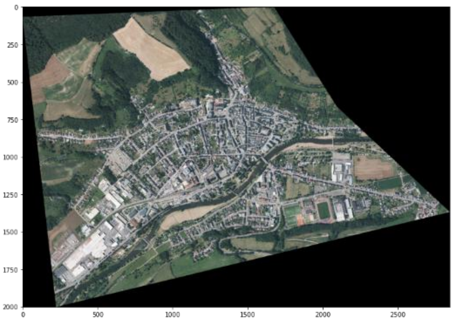
Inference was run on the five Luxembourg AOIs downsampled to 1m spatial resolution, and segmentation and polygon-level quality metrics derived and collected.

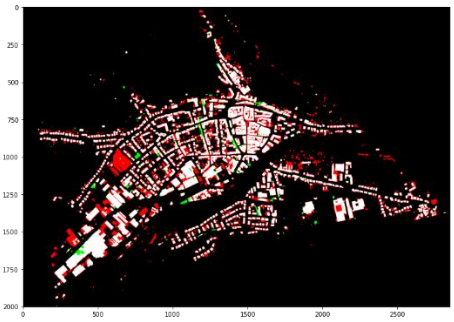
### 3.2.4 Postprocessing

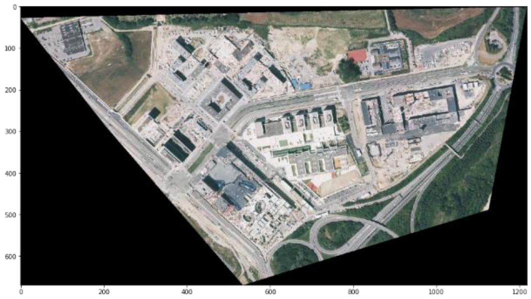
Binary segmentation masks were obtained by thresholding the segmentation model outputs (in each case, the raw results are floating point building probabilities for every pixel in the range [0,1]). For this Otsu’s method was used to derive an adaptive threshold per AOI, leading to values in the range 0.45-0.5. Polygonisation was carried out on each AOI on these thresholded results using GDAL’s 4-connectedness polygonization algorithm.

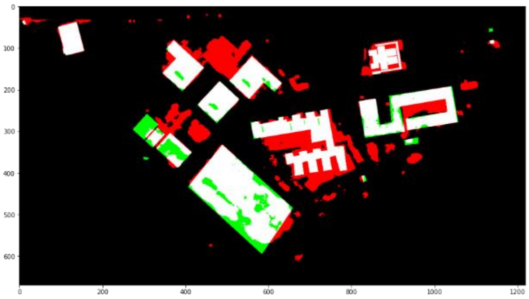
### 3.2.5 Results

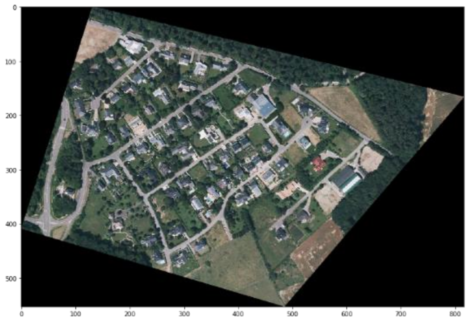
#### 3.2.5.1 Segmentation results

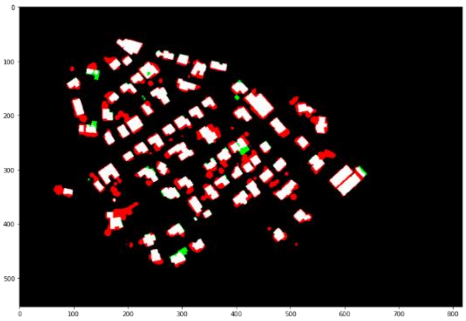


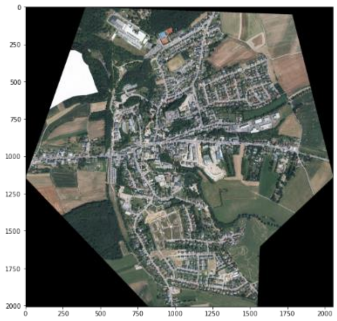


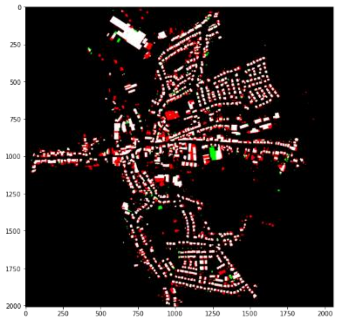


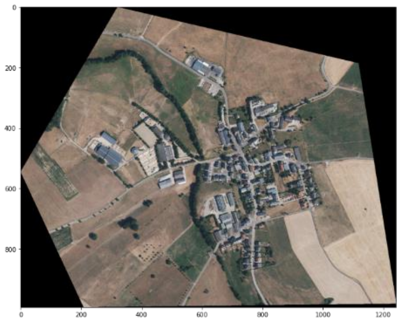












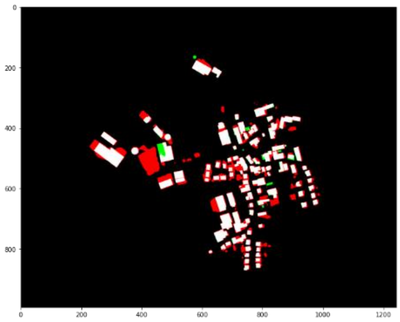


**Figure 15 - Confusion map for the five AOIs during 2018 using the 1m DeepResUNet model trained on Belgian ortho data with the weighted binary cross entropy loss. White and black pixels represent true positives and negatives respectively. Red and green pixels represent false positives and negatives respectively.**

In Figure 15 confusion maps are presented for the five areas of interest. The predominant kinds of errors are false positives (red). These can manifest as overestimated building footprints (this can be partially traced back to the effect of binarizing footprints at low spatial resolution; pixels which would be partially occupied by buildings when viewed at a higher spatial resolution are considered simply to be buildings). The low spatial resolution also leads to smaller gaps between buildings at the scale of a few pixels to be erroneously counted as buildings, effectively merging together separate buildings.

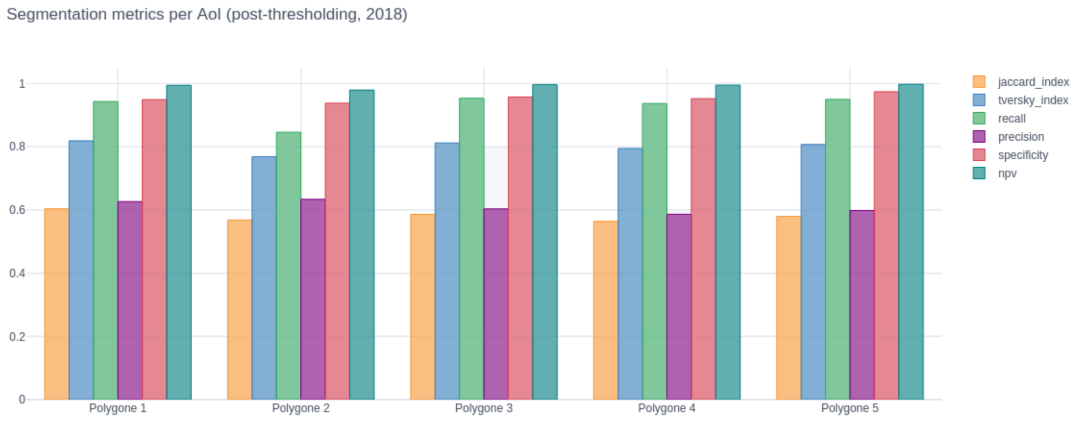


**Figure 16 - post-thresholding segmentation metrics per AOI for the 1m model. The predominance of FP-type errors can be seen in the low precision values (purple) compared to the recall values (green). The Jaccard Index (or Intersection-over-Union) in yellow provides an overall segmentation quality metric which is stable around just under 0.6 across each AOI.**

#### 3.2.5.2 Polygon- level results

The image-level binary segmentation results were polygonised using GDAL in order to investigate the quality on the level of individual buildings, and to understand the qualitative nature of the segmentation errors when translated to a group of polygons to be compared to ground truth.

An important metric in this analysis is the **Intersection-over-Union** (or IoU). For two polygons, this is defined as the ratio of the area of their intersection to the area of their union (the total area spanned by both). The Jaccard Index is a synonym for this when applied to whole-image level (i.e. where one is comparing two segmentation masks in their entirety). An IoU of 1 implies perfect alignment of polygons, while an IoU of 0 implies the two shapes do not overlap at all.

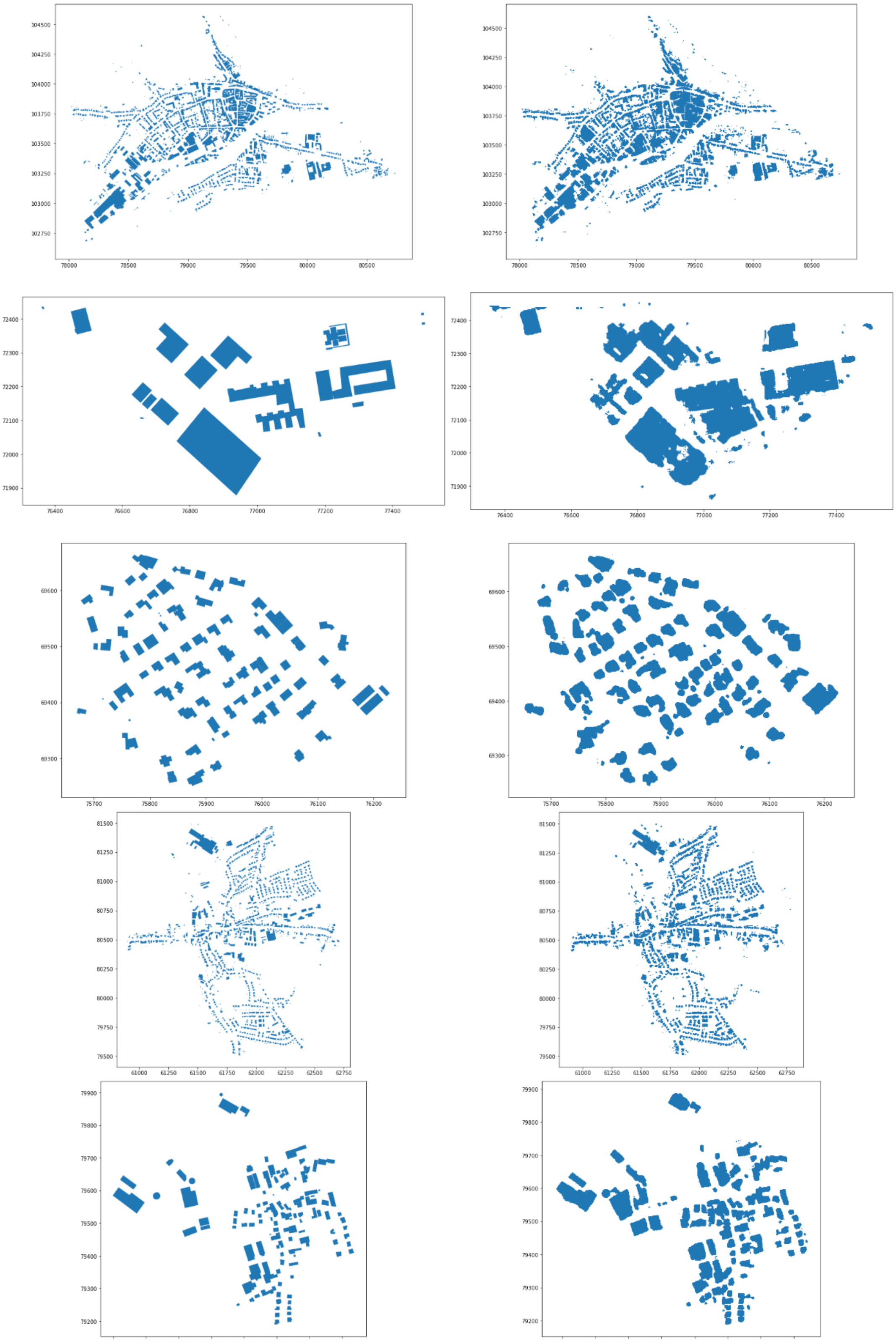


**Figure 17 – Raw polygon-level results for the 1m model on the five AOIs. In AOI 1 (top) the densely packed buildings are merged into one larger shape. Building footprints tend to be overestimated in size.**

#### 3.2.5.3 Categorisation of polygon-level results

In order to quantify and classify the polygon-level results, a number of additional metrics were calculated by comparing the ground truth to the predicted polygons. These are more complex than the pixel-level categories of TP, TN, FP and FN. It is useful to define the following cases:

1. **Single match** refers to the cases where a reference polygon (either in the set of ground truth polygons or in the set of predicted polygons) intersects exactly one polygon in the other set. In these cases, we can calculate the IoU of the pair of polygons.
    a. If the reference polygon is in the ground truth set, a single match with a predicted polygon can be either:
        i. A **unique match**, where the predicted polygon intersects no other true polygons.
        ii. **Undersegmentation**, where the single matching predicted polygon intersects other true polygons.
    b. If the reference polygon is in the predicted set, a single match with a ground truth polygon can be either:
        i. A **unique match**, where the ground truth polygon intersects no other predicted polygons.
        ii. **Oversegmentation**, where the ground truth polygon intersects other predicted polygons.
2. **No match** refers to the cases where a reference polygon (either in the set of ground truth or predicted polygons) intersects no polygons in the other set.
    a. If the reference polygon is in the ground truth set, this represents a **missed building** (or equivalently a false negative).
    b. If the reference polygon is in the predicted set, this represents a **false detection** (or equivalently a false positive).
3. **Multiple match** refers to the cases where a reference polygon (either in the set of ground truth or predicted polygons) intersects multiple polygons in the other set.
    a. If the reference polygon is in the ground truth set, this again represents **Oversegmentation** (one real building encompassing multiple predicted buildings).
    b. If the reference polygon is in the predicted set, this again represents **Undersegmentation** (one predicted building encompassing multiple real buildings).

#### 3.2.5.4 Global distributions per AOI

Here we can show some global properties of the segmented polygons for each AOI.

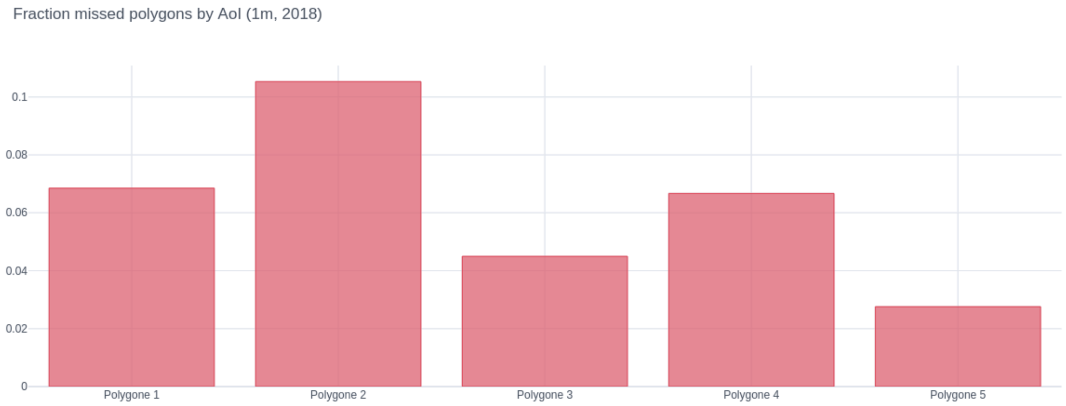


**Figure 18 - Fraction of missing buildings in each AOI for the initial 1m model trained on Belgian ortho data with WBCE loss.**

In Figure 18, the missed building rate for the 1m model is shown. As will be a recurring theme, AOI 2 (“Polygone 2”) results are notably worse than the other AOIs. This is due to the complexity of these buildings; several are multi-tiered with roofs at different heights, others have open spaces which are difficult to distinguish by eye from roofs. Nonetheless even at 1m spatial resolution, the miss rate is low for the remaining AOIs at the 3-7% range.

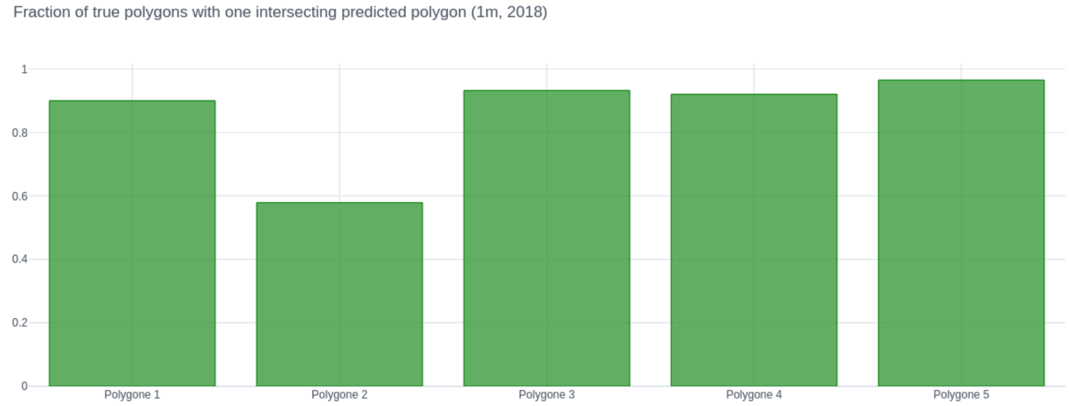


**Figure 19 – Fraction of true polygons with one matching predicted polygon (unique matches and undersegmented buildings)**

In Figure 19, the fraction of true buildings with one matching predicted building is shown. The difference between each bar and 1 represents the oversegmented or missing true buildings. In each case apart from AOI 2 we can infer that oversegmentation is a very minor issue at low spatial resolution, which we can confirm in Figure 20.

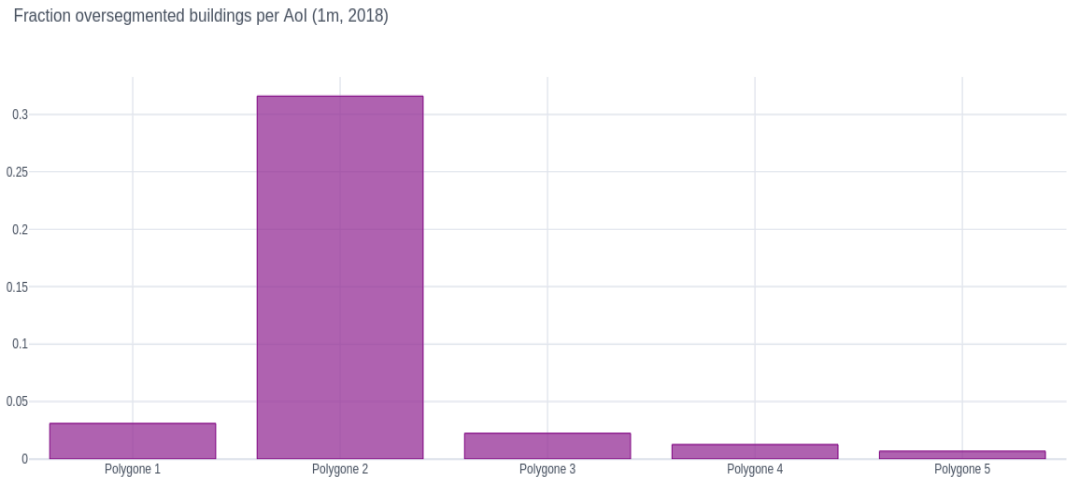


**Figure 20 - fraction of oversegmented buildings at 1m spatial resolution for each AOI using the initial model.**

In Figure 21 we plot the ground truth and predicted polygons in AOI 1 for the case where a single predicted polygon overlaps a given ground truth polygon, and colour the predicted polygons according to their average IoU with all the true buildings they intersect. Here it's clear that the most significant problem at this spatial resolution is the undersegmented buildings. These occur in situations with densely packed buildings with little space between.

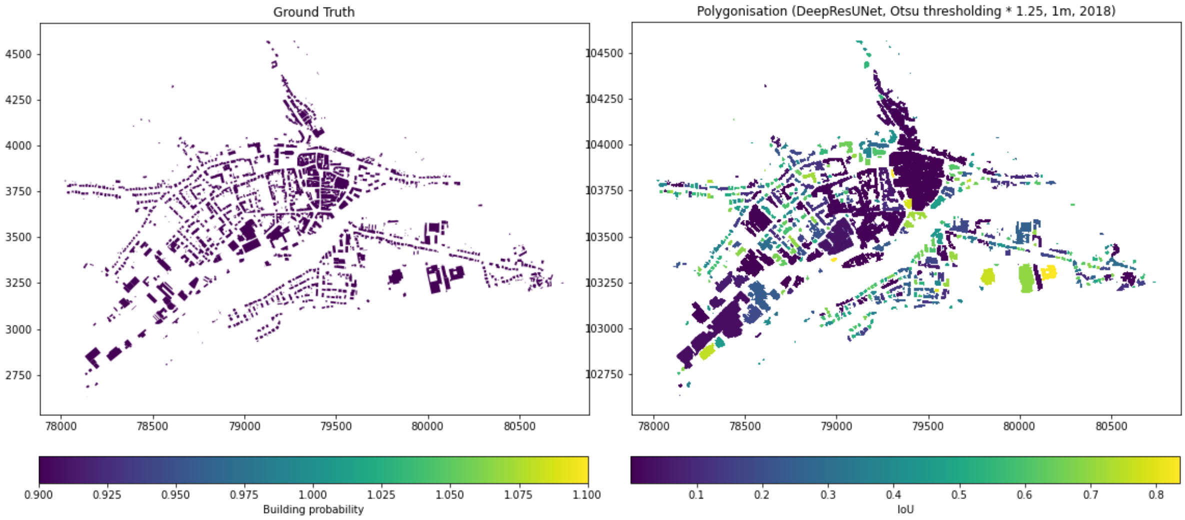


**Figure 21 - Predicted polygons which intersect true polygons coloured by average IoU with the buildings they intersect. Undersegmented buildings appear as purple-blue and occur where a single predicted polygon erroneously contains multiple true buildings.**

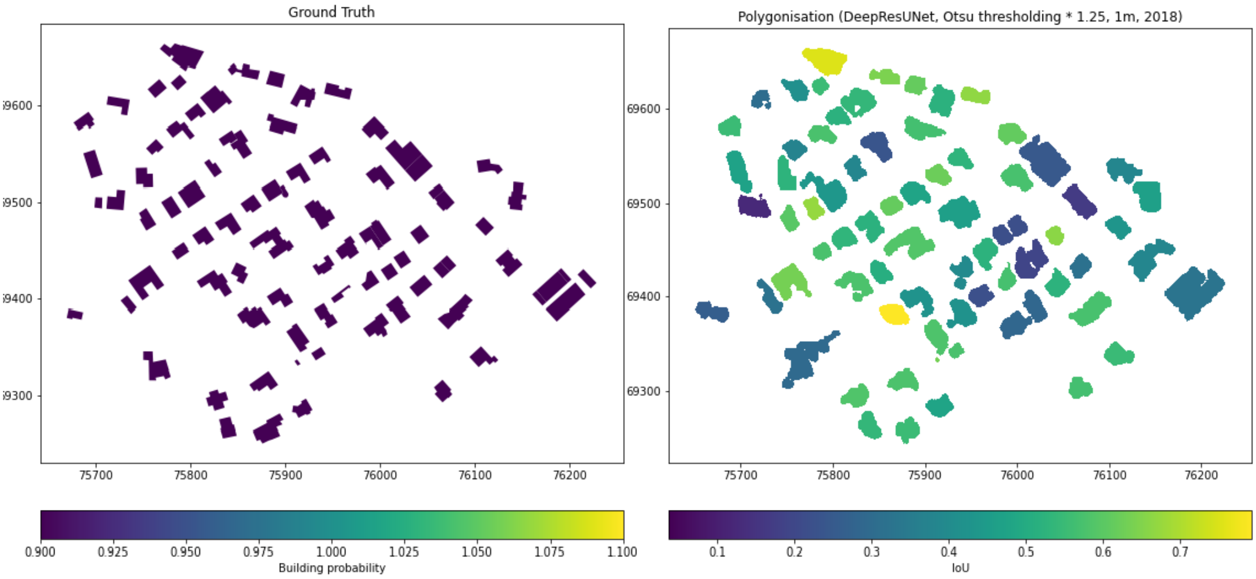


**Figure 22 - Predicted polygons of AOI 3 where these intersect one or more true polygons, coloured by average IoU.**

In Figure 22 we can observe a qualitatively different scenario. In a rural setting such as AOI 3 where buildings are more separated, undersegmentation is rare and most predicted polygons are unique matches with corresponding true polygons. In these cases, the footprint quality as measured by IoU is typically in the 50-60% percent range, with the predicted polygons in general being too large.

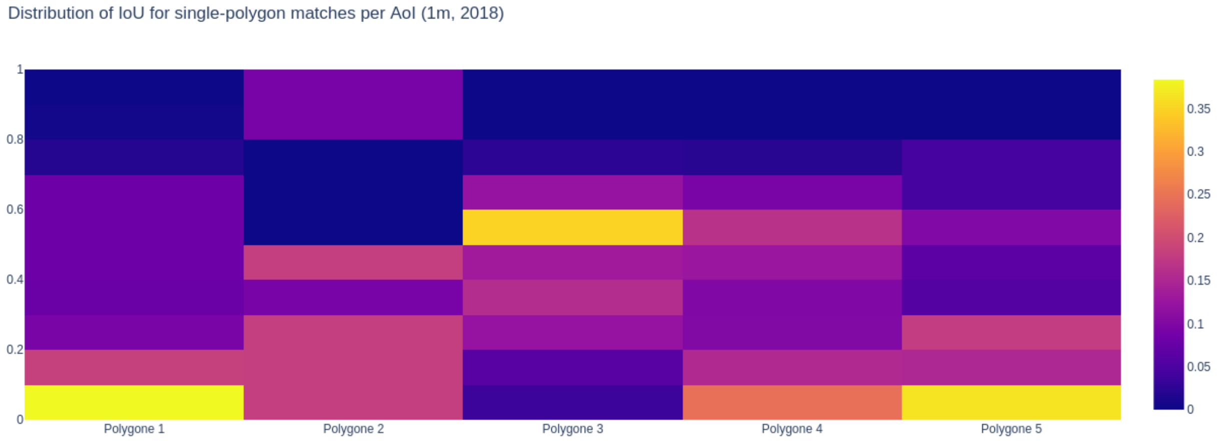


**Figure 23 - IoU distributions for true polygons which match one predicted polygon by AOI. The y-axis represents the value of the IoU in bins of 10%, while the colour represents the fraction of buildings in that IoU range.**

Figure 23 summarises the situation: for most AOIs the IoU is low due to the predominance of undersegmented buildings. In AOI 3 the majority of buildings are in the 50-60% IoU range. We should hope for distributions in each AOI peaked strongly at high IoU.

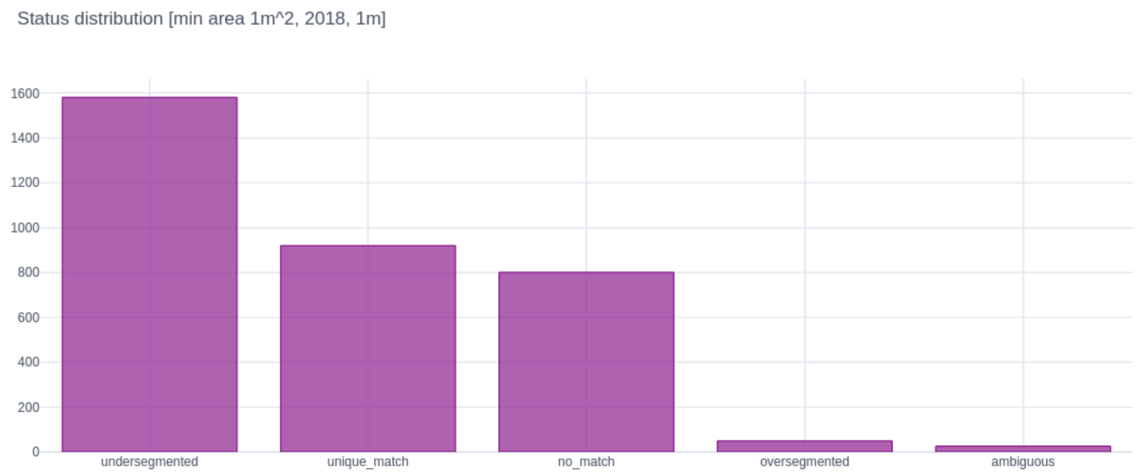


**Figure 24 - Status distribution for all AOIs for polygons predicted by 1m DeepResUNet. Around half of these are undersegmented (i.e. intersect multiple true buildings). A little over one quarter are unique matches and a little under one quarter are non-matches (i.e. false detections)**

One can conclude that 1m spatial resolution is not sufficient for distinguishing individual building polygons except for in those cases where buildings are well-separated. When this is the case the footprint quality is mediocre and tends to overexaggerate building sizes.

## 3.3 EXPERIMENTS AT 20CM SPATIAL RESOLUTION

### 3.3.1 Architecture

The procedure of generating AOI quality metrics was repeated using the baseline DeepResUNet model with WBCE loss in order to perform a like-for-like comparison with 1m results.

Following the implementation of the architectural enhancements detailed in section 2.5, a period of hyperparameter tuning and model selection was carried out whereupon the best-performing model was selected based on validation data metrics using each of the three loss functions put forward. These models were used to generate final, improved results which are also shown here.

In each case the best-performing model included all of the additional architectural elements (spatial attention gates, deep supervision, multi-scale pooling and CBAM modules on each residual block).

The optimal initial learning rate found in these experiments was 10^-4 for every loss function, with a reduction of a factor of 50% after 2 epochs of validation loss stagnation until a minimum value of 10^-6.

### 3.3.2 Training data

In all 20cm experiments training data consisted of the Luxembourg (Belair) training sample for each of the years 2017, 2018, 2019 and 2019 (winter true ortho – resampled from 10cm to 20cm), along with the ISPRS Potsdam dataset. Ten percent of the patches constituting this combined training dataset were selected at random and separated off to be used as validation data during model training.

### 3.3.3 Inference data

Inference was run on the five Luxembourg AOIs at 20cm native spatial resolution (resampled from 10cm in the case of 2019 winter), and segmentation and polygon-level quality metrics derived and collected.

In the case of the best weighted binary cross entropy model, results for the entire country were additionally generated.

### 3.3.4 Postprocessing

Binary segmentation masks were again obtained by thresholding the segmentation model. For this Otsu's method was used to derive an adaptive threshold per AOI. Polygonisation was carried out on each AOI on these thresholded results using GDAL's 4-connectedness polygonization algorithm.

### 3.3.5 Results

#### 3.3.5.1 Segmentation results – DeepResUNet + WBCE loss

In Figures 24-28 the confusion maps for each AOI are presented, which may be compared with the 1m spatial resolution results generated by the same architecture.

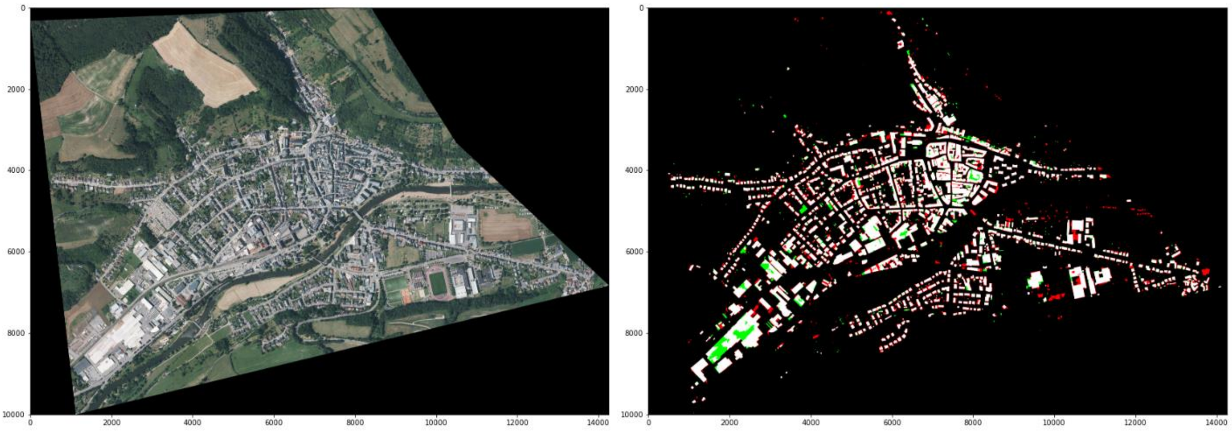


**Figure 25 - Confusion map for AOI 1, 20cm baseline DeepResUNet model. Again TP/TN white/black respectively and FP/FN red/green respectively.**

The predominant error type with weighted binary cross entropy is again the false positive kind, although the higher spatial resolution significantly reduces the “exaggerated” footprints seen at 1m.

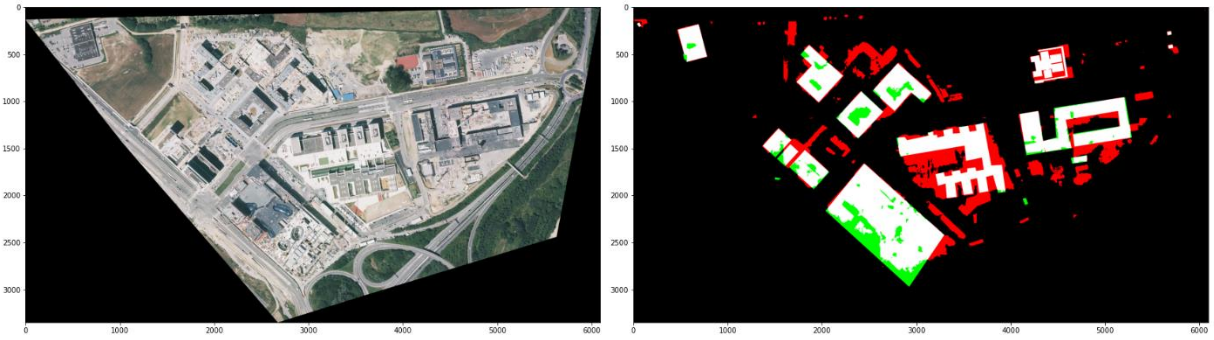


**Figure 26 - Confusion map for AOI 2, 20cm baseline DeepResUNet model. Oversegmentation is a more visibly significant problem here than at 1m and the increased spatial resolution does not significantly improve segmentation quality. This is likely a result of such atypical multi-tiered buildings lacking in training data.**

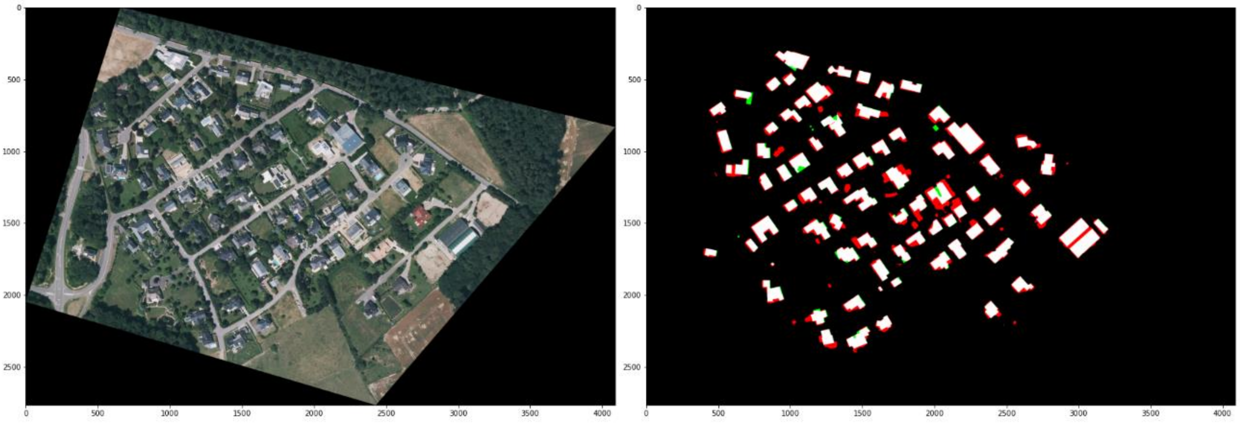


**Figure 28 - Confusion map for AOI 3, 20cm baseline DeepResUNet model. Footprint exaggeration is still an issue for this loss function although this is significantly mitigated with respect to 1m.**

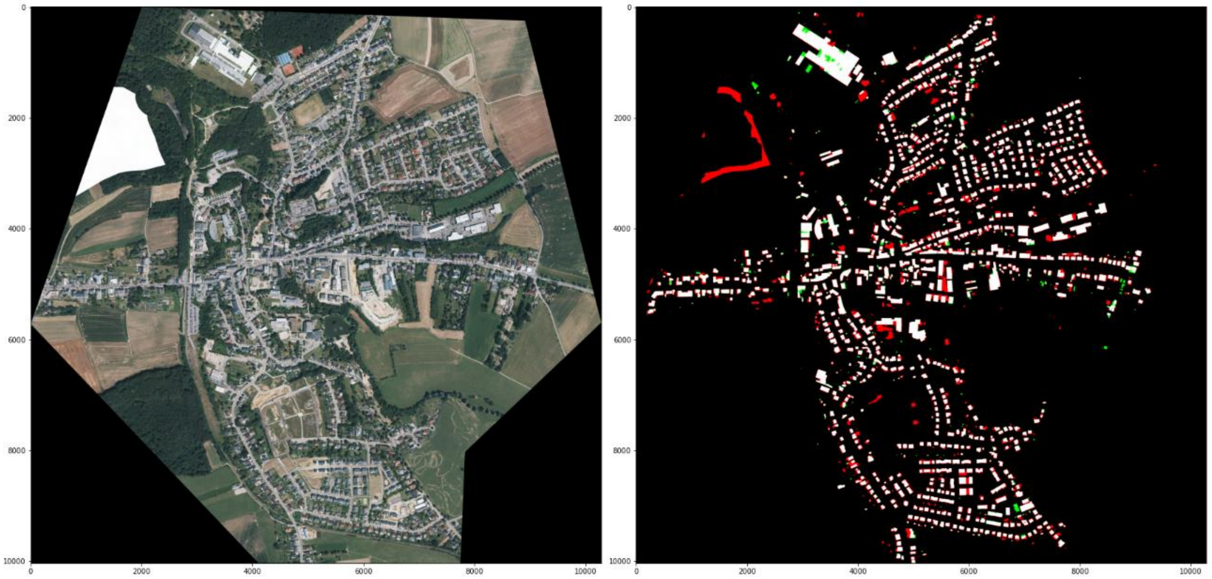


**Figure 29 - Confusion map for AOI 4, 20cm baseline DeepResUNet with the WBCE loss**

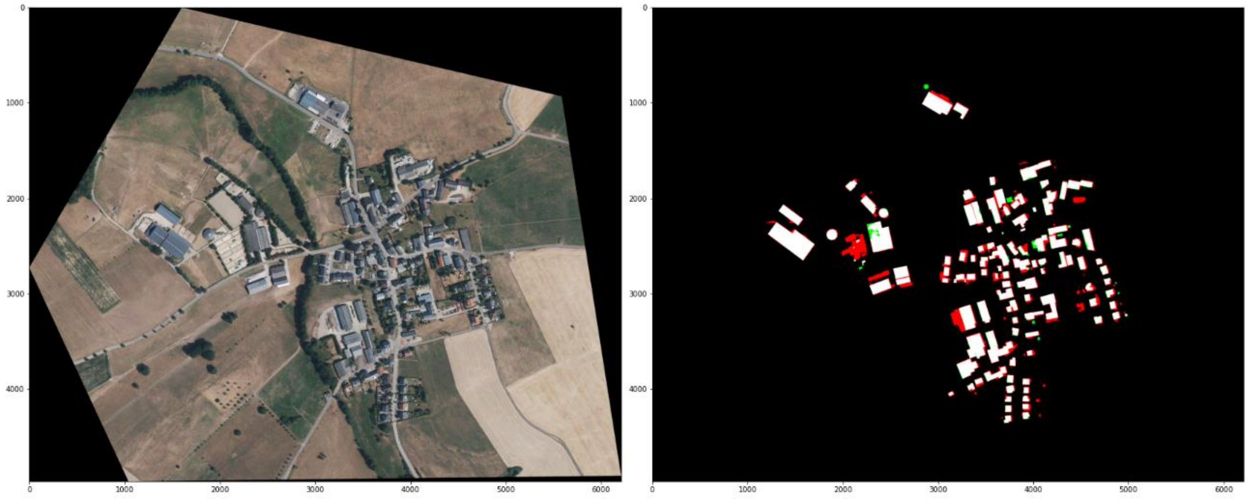


**Figure 30 - Confusion map for AOI 5, 20cm baseline DeepResUNet model trained with the WBCE loss**

With the exception of AOI 2 (Figure 26) 20cm spatial resolution represents a huge qualitative improvement to 1m, while taking a factor of ~20-25 longer to both train this model and run inference. Nonetheless 20cm inference on the country of Luxembourg was possible in approximately 12 hours on the ACT hardware provided.

In Figure 31 the global AOI-level quality metrics are shown. These reflect the significantly reduced false positive rate with respect to 1m.

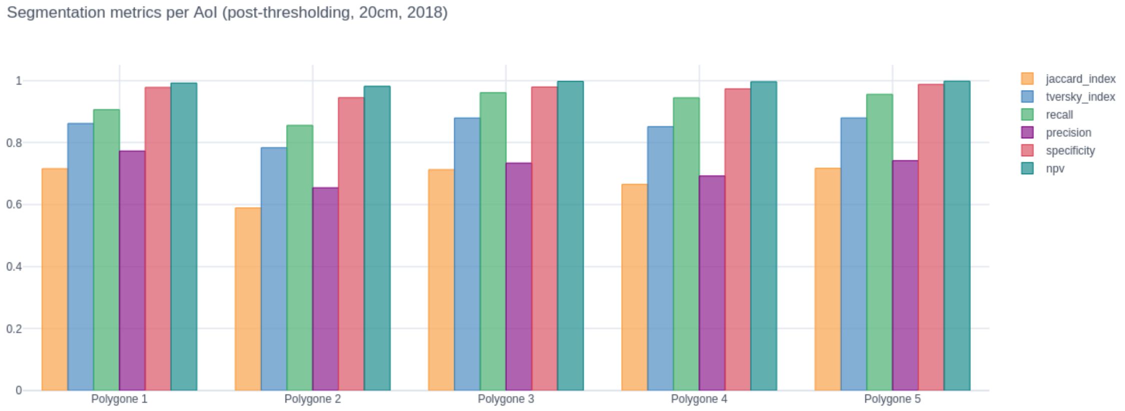


**Figure 31 - Per- AOI segmentation metrics for the 20cm baseline DeepResUNet model. With the exception of  AOI 2, Jaccard Index (global IoU) values are in the 65-70% region (up from <~60%) while precision values are in the 70-80% region (up from 60%).**

#### 3.3.5.2 Segmentation results – DeepResUNet + all enhancements + WBCE loss

Here we present the results for the best trained model with architectural enhancements in place. For the particular case of WBCE, we note that the distribution of predicted building probabilities was shifted higher than the baseline model and an appropriately higher threshold should be selected to accommodate this. In the confusion maps (Figures 32-36) we used the same adaptive thresholding scheme (Otsu's histogram method, scaled up to be more conservative by a factor of 1.25, typically resulting in a threshold in the region ~0.57) as in previous experiments which resulted in a slight decline in quality. The calculation of the segmentation metrics was repeated by increasing this scale factor to 1.5 resulting in a threshold ~0.7 which eliminated false positives produced better results (see Figure 37).

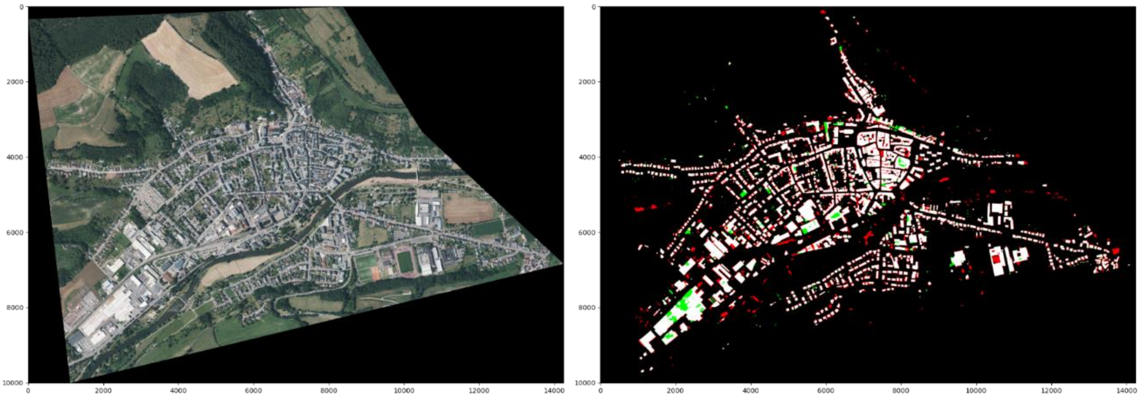


**Figure 32 - Confusion map for AOI 1 for the best-performing 20cm model trained with the WBCE loss function**

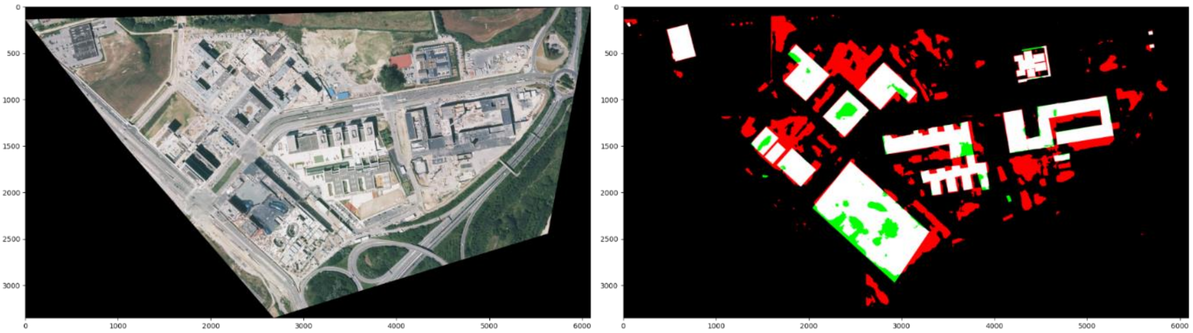


Figure 33 - Confusion map for AOI 2 for the best-performing 20cm model trained with the WBCE loss function

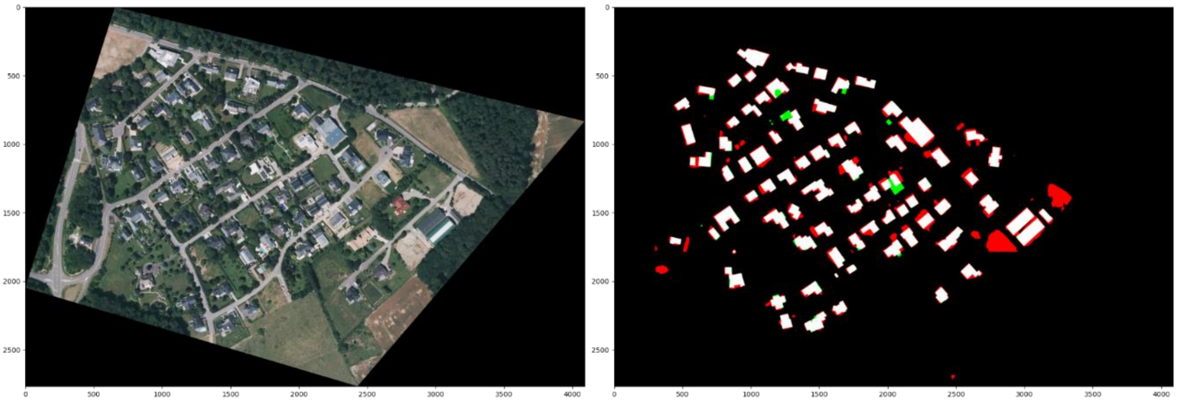


Figure 34 - Confusion map for AOI 3 for the best-performing 20cm model trained with the WBCE loss function

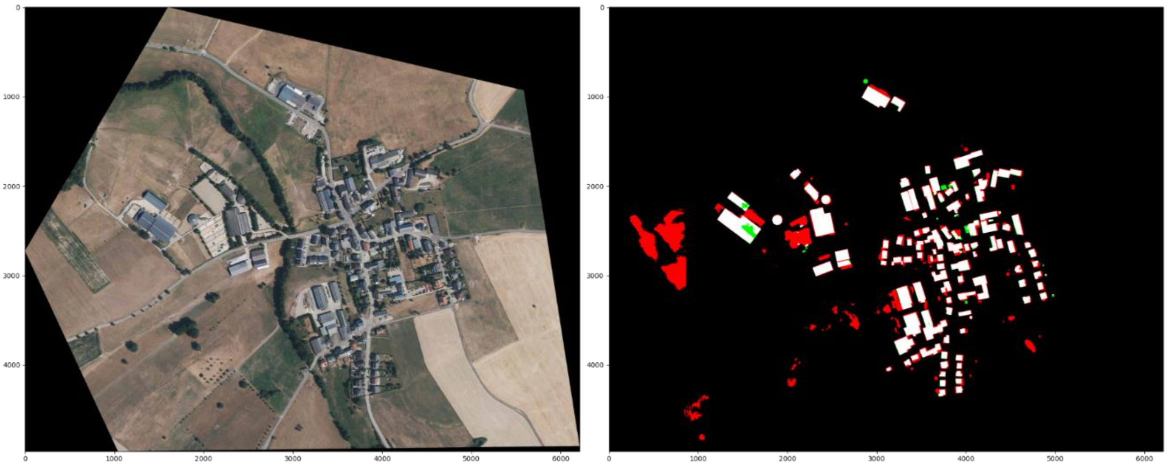


Figure 35 - Confusion map for AOI 4 with the best-performing 20cm model trained with the WBCE loss function

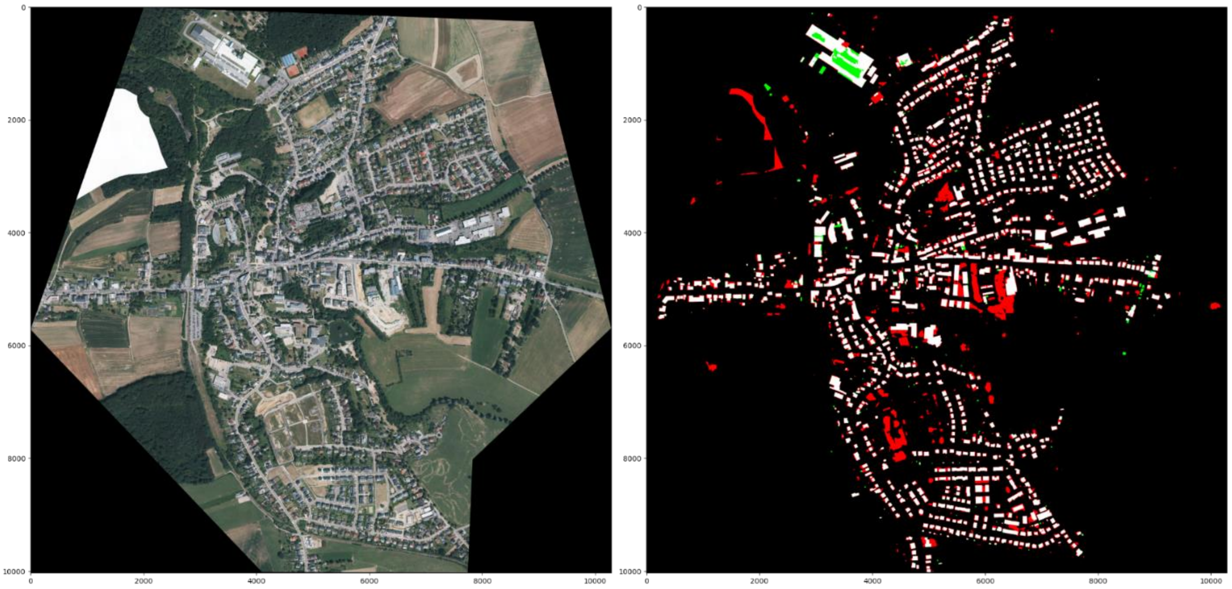


**Figure 36 - Confusion maps for AOI 5 with the best-performing 20cm model trained with WBCE loss**

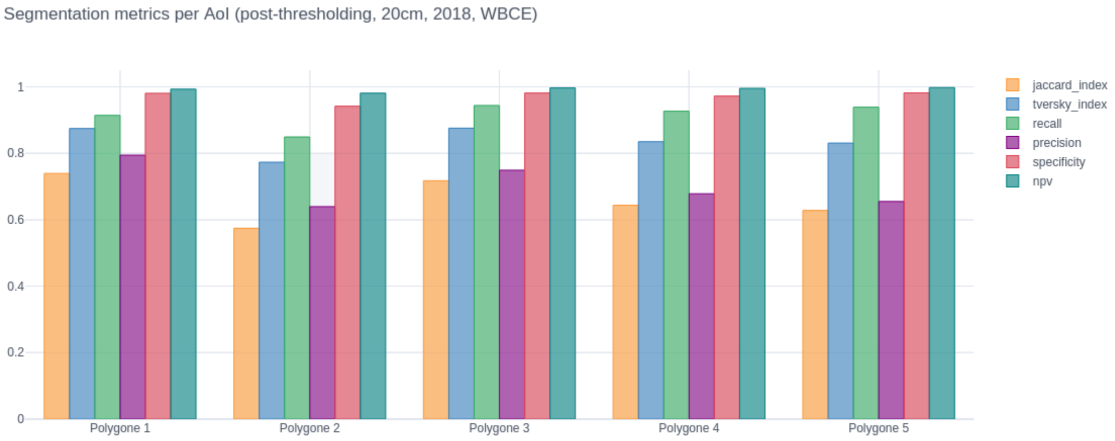


**Figure 37 - final segmentation metrics for the best-performing 20cm WBCE model.**

It's worth noting here that the disparity in validation losses observed between the baseline and best WBCE models (0.272 and 0.186 respectively) does not translate into equivalently significant gains in the AOI testing dataset metrics. This is likely due to the more complex model being able to better learn to leverage the subtleties of the Belair training dataset but failing to generalize this understanding well to the testing datasets. In general, more complex models contain more parameters and interlocking mechanisms, and thus require more data to train. It is likely that the disparity between the baseline and best model architectures would grow significantly in the presence of a larger and more diverse training set containing different samples from Luxembourg.

#### 3.3.5.3 Segmentation results – DeepResUNet + all enhancements + Tversky loss

Here we present the results for the best trained model with architectural enhancements in place. In the case of the Tversky loss, the Otsu thresholding scheme was kept, although in practice the Tversky and Dice losses produce extremely bimodal distributions, i.e. building regions are either identified with a particular shape with a very high probability approaching 1 or are entirely absent with probability approaching 0. This renders the choice of threshold nearly meaningless.

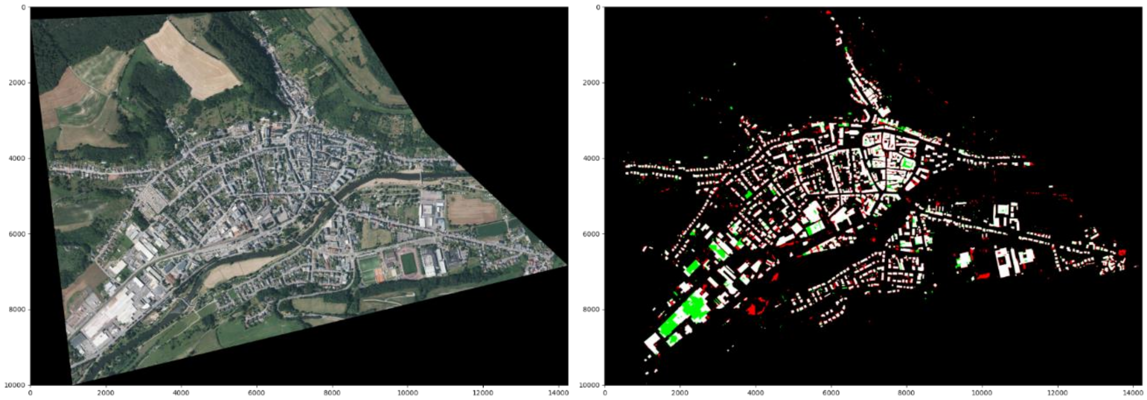


**Figure 38 - Confusion map for  AOI 1 with the best-performing model trained with the Tversky loss**

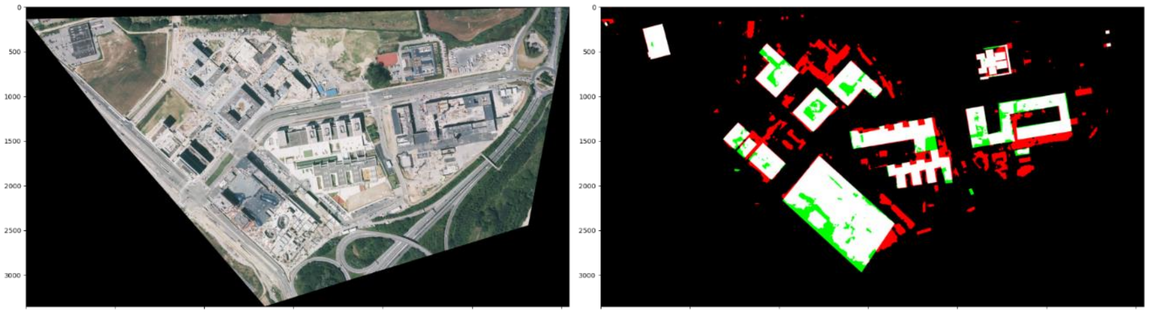


**Figure 39 - Confusion map for  AOI 2 with the best-performing model trained with the Tversky loss**

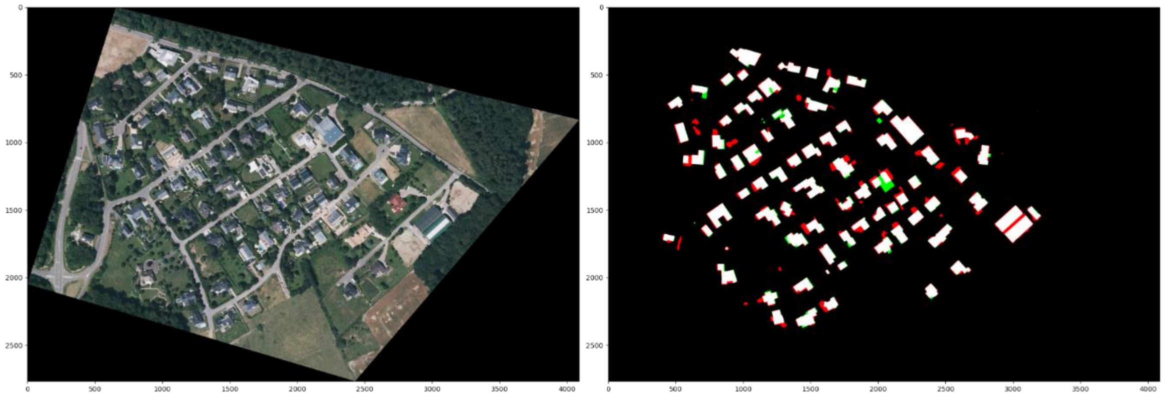


**Figure 40 - Confusion map for AOI 3 with the best-performing model trained with the Tversky loss**

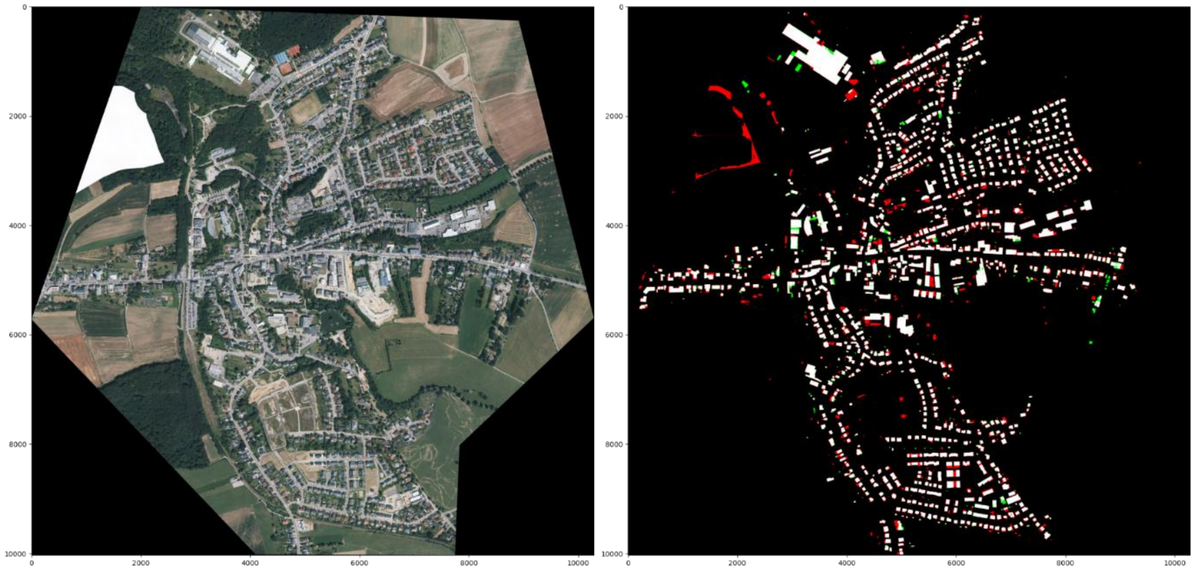


**Figure 41 - Confusion map for  AOI 4 for the best-performing model trained with the Tversky loss**

Another point worth of note is the stability of the training. As mentioned in section 2.5, this loss and gradients thereof have a considerably more complex functional form than the weighted binary cross entropy. This can cause the gradient descent algorithm to struggle to navigate a highly oscillating loss surface. We observed around three training experiments out of ten where training destabilized and the loss diverged. Results could likely be improved further by experimenting with more sophisticated learning rate schedulers, or tuning the initial learning rate more finely.

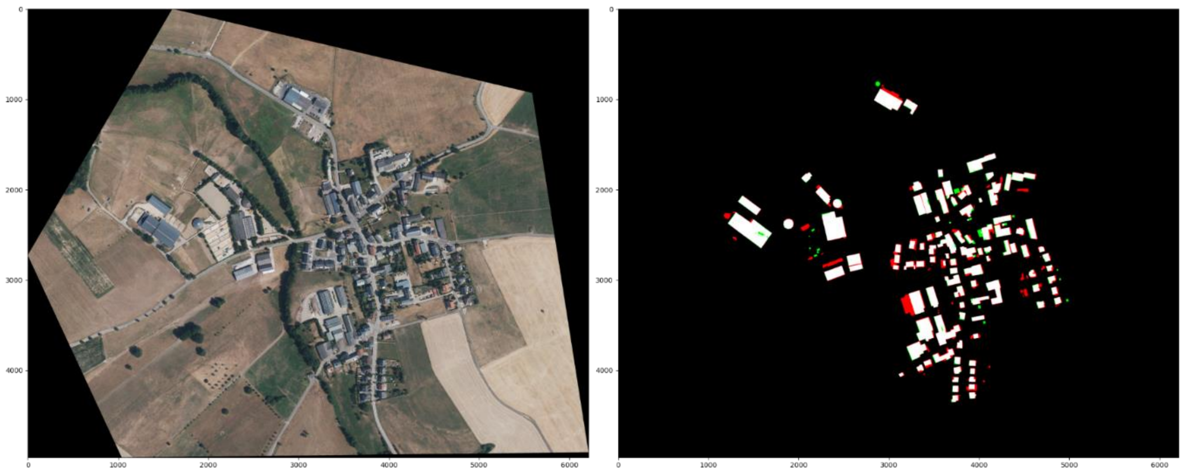


**Figure 42 - Confusion map for  AOI 5 for the best-performing model trained with the Tversky loss**

In Figures 38-42 the confusion maps for the best-performing Tversky loss model at 20cm spatial resolution are presented. In our experiments the architectural improvements represented a decline in validation loss from 0.17 to 0.13 with respect to the baseline DeepResUNet. One can see that these are a qualitative improvement on the weighted binary cross entropy model, particularly in the reduction of false positives. Nonetheless these are still the most predominant error-type which appears for this loss function.

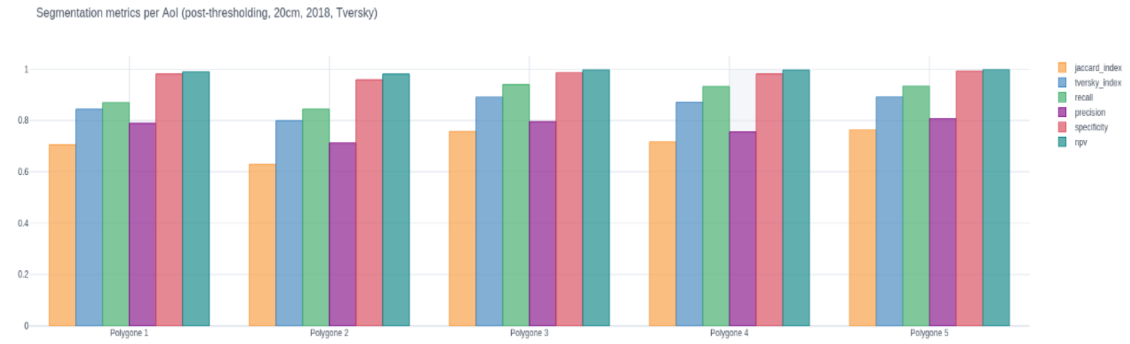


**Figure 43 - Final segmentation metrics per  AOI for the best-performing model trained with the Tversky loss.**

Nonetheless, the Tversky loss results are quantitatively a significant improvement on the weighted binary cross entropy (see Figure 43). In particular the precision values touching the 80% mark in the four more typical AOIs improves drastically in some cases on the WBCE results shown in Figure 37.

### 3.3.5.4 Segmentation results – DeepResUNet + all enhancements + Dice loss

Here we present the results for the best trained model with architectural enhancements in place using the Dice loss function. Similarly, to the Tversky loss and for the same reasons, the choice of threshold is unimportant here.

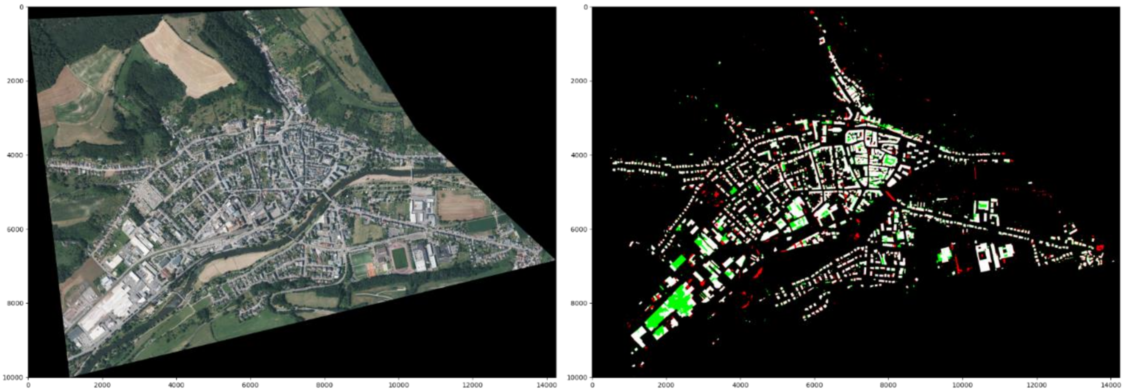


**Figure 44 - Confusion map for  AOI 1 with the best-performing model trained with the Dice loss**

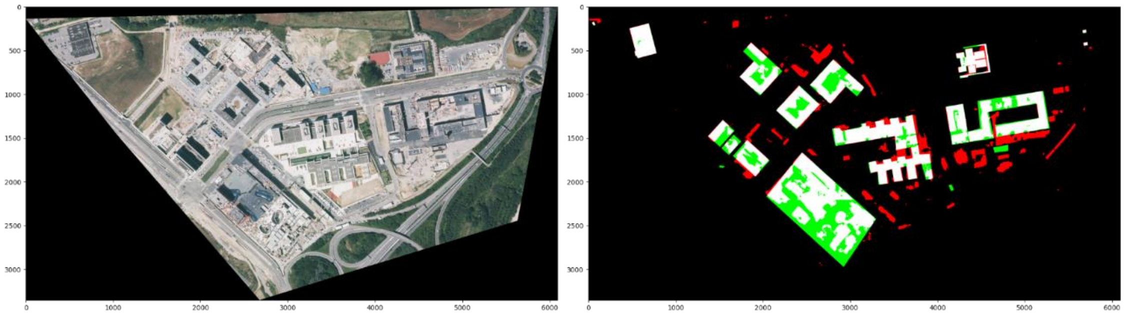


**Figure 45 - Confusion map for  AOI 2 for the best-performing model trained with the Dice loss**

In Figures 44-48 the confusion maps are presented for the five AOIs with the best model trained with the Dice loss. Due to time constraints, we did not train a baseline DeepResUNet model with this loss function. The minimum validation loss reached in our experiments was 0.15.

Immediately of note is the trading away of false positives (in red) for false negatives (in green). This can be traced back to the equal weighting factors for these types of error (in contrast to Tversky, where FNs were more severely punished) in the loss function. As a result, for the more typical AOIs (1, 3-5) most building footprints are very accurate and not exaggerated, although a larger fraction is missed entirely. Smaller, isolated false positives are also much less frequent.

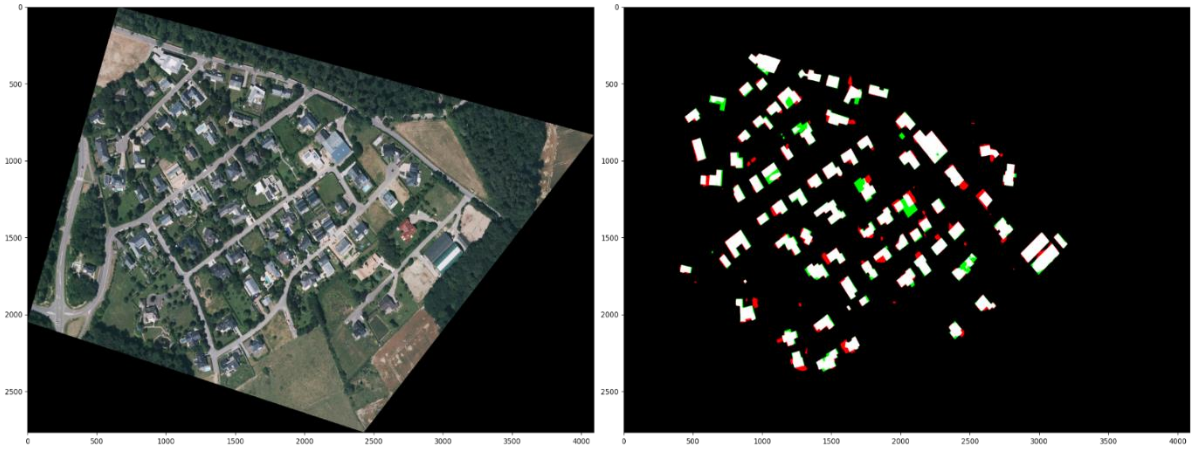


**Figure 46 - Confusion map for  AOI 3 for the best-performing model trained with the Dice loss**

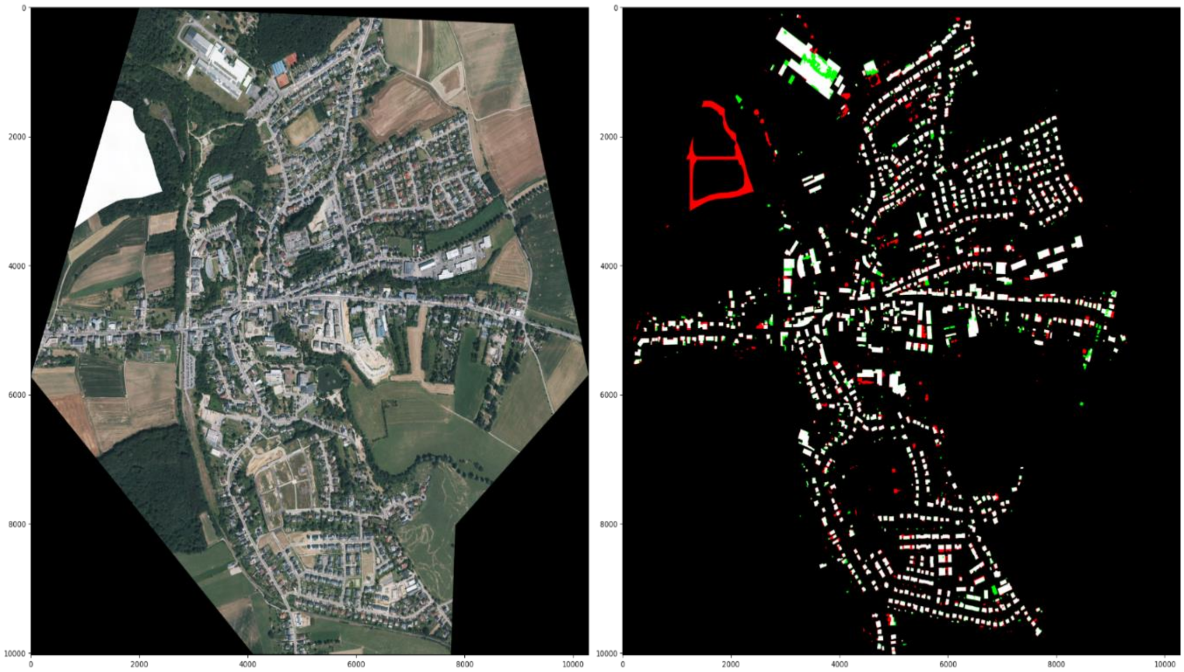


**Figure 47 - Confusion map for AOI 4 for the best-performing model trained with the Dice loss**

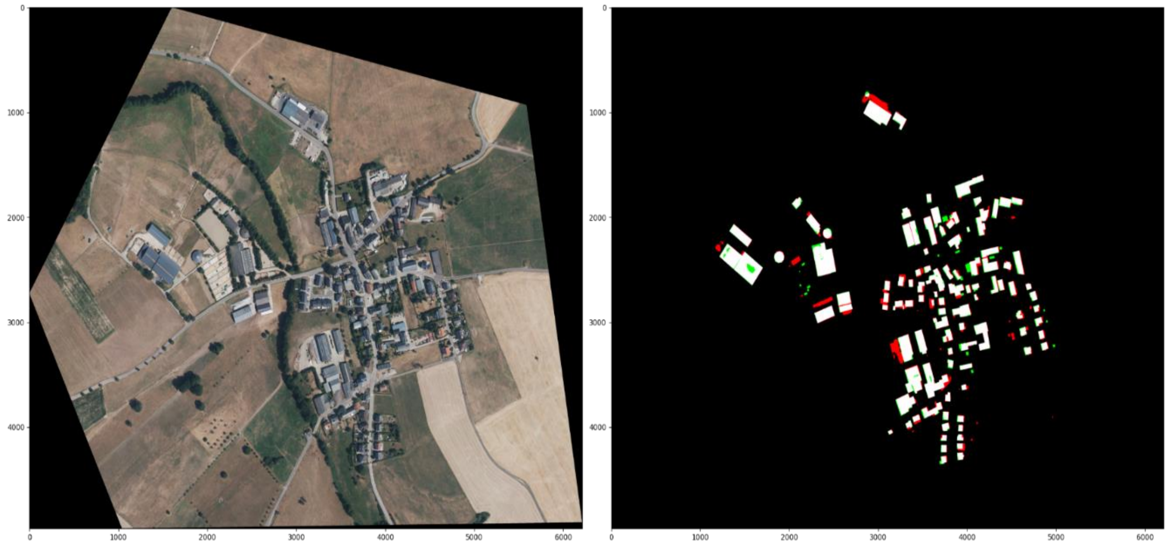


**Figure 48 - Confusion map for  AOI 5 with the best-performing model trained with the Dice loss**

In Figure 49 the overall quality metrics for each AOI are presented. The equal prioritization of FP and FN error-types manifests in the precision and recall (purple and green respectively) being more equally balanced than with the Tversky loss. The overall quality indicator (the Jaccard Index in yellow) is very similar. If one ignores the artefact caused by the presence of the border in AOI 4, the overall quality measured by the Jaccard Index is slightly better on average.

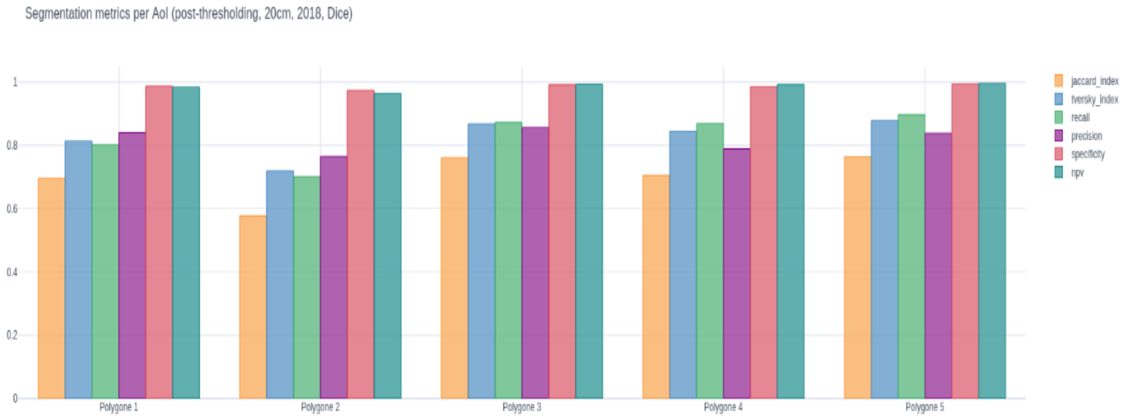


**Figure 49 - Overall segmentation quality metrics per  AOI for the best-performing model trained with the Dice loss**

While the quality metrics for the Dice and Tversky losses are similar, one would expect to see the differences emerge more prominently in the polygon-level analysis, as larger false positives rates (Tversky) are more likely to merge buildings and create undersegmentation, while higher false negative rates (Dice) are more likely to cause missed buildings and oversegmentation.

### 3.3.5.5 Polygon- level results – DeepResUNet + WBCE loss

Here we present the polygon level results for the baseline DeepResUNet model on the five testing AOIs.

In Figures 50-54, colour-coded representations of the predicted polygons are shown where the colours indicate whether that polygon was a unique match, undersegmented, oversegmented, a non-match (false detection) or ambiguously segmented. Once again undersegmentation is the predominant error-type and is prominent in dense areas with small spaces between buildings. These were not well-represented in the Belair training dataset, so it is likely that this issue could be alleviated by providing ground truth for these kinds of urban environments.

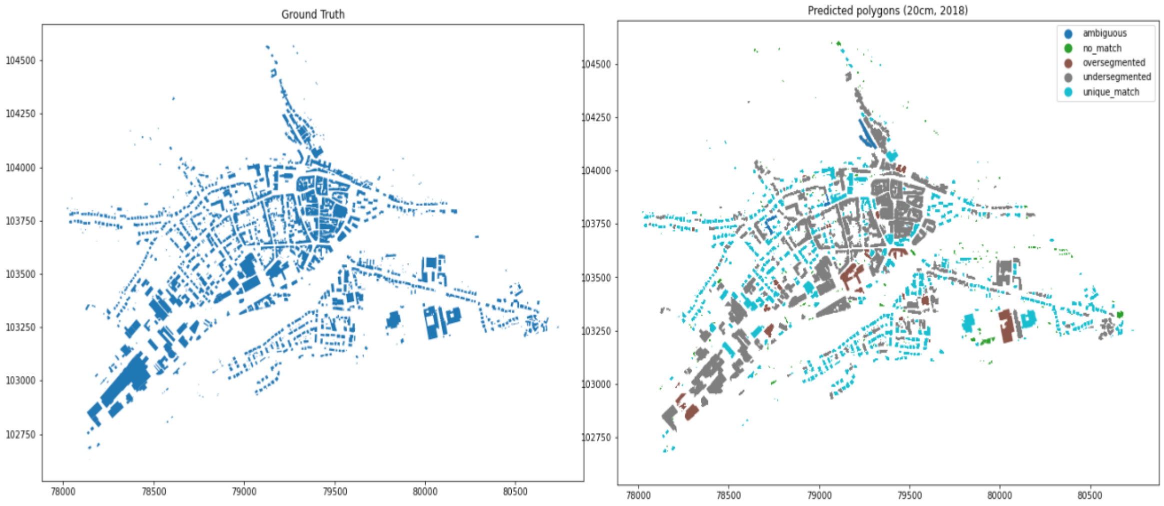


**Figure 50 - Predicted polygons (right) and ground truth (left) for AOI 1 for the baseline DeepResUNet model trained with the WBCE loss function. Predicted polygons are colour-coded by status: light blue for unique matches, gray for undersegmented, brown for oversegmented, green for no match (false detection) and blue for ambiguously segmented.**

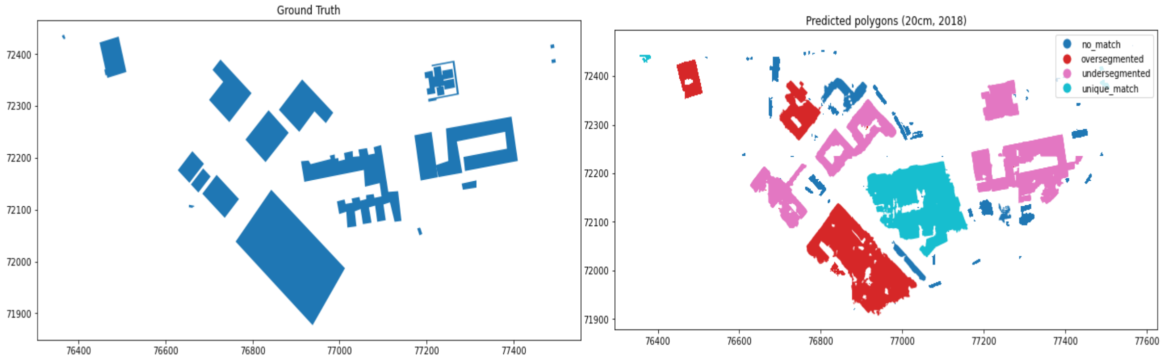


**Figure 51 -- Predicted polygons (right) and ground truth (left) for AOI 2 for the baseline DeepResUNet model trained with the WBCE loss function. Predicted polygons are colour-coded by status: light blue for unique matches, pink for undersegmented, red for oversegmented, and blue for no match (false detection).**

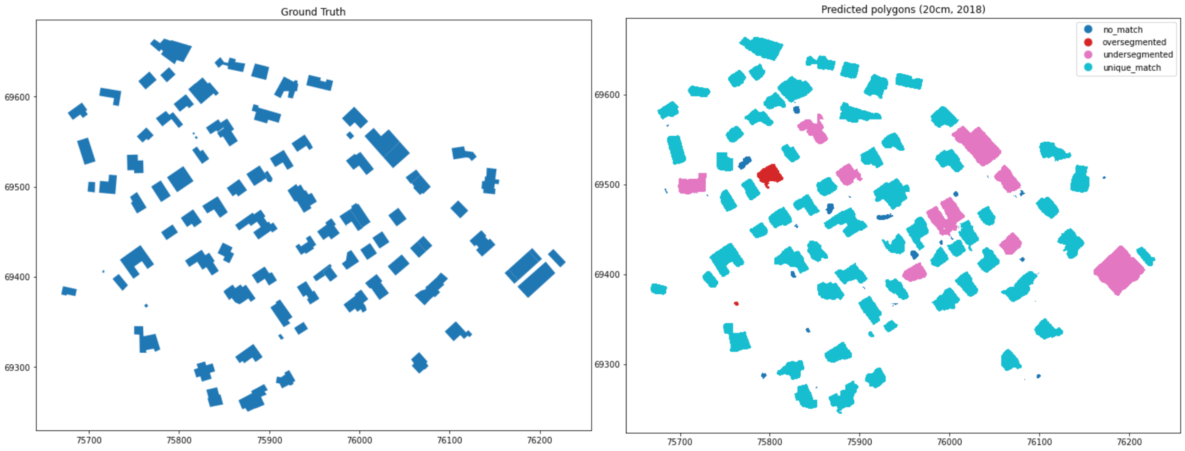


**Figure 52 - Predicted polygons (right) and ground truth (left) for  AOI 3 for the baseline DeepResUNet model trained with the WBCE loss function. Predicted polygons are colour-coded by status: light blue for unique matches, pink for undersegmented, red for oversegmented, and blue for no match (false detection).**

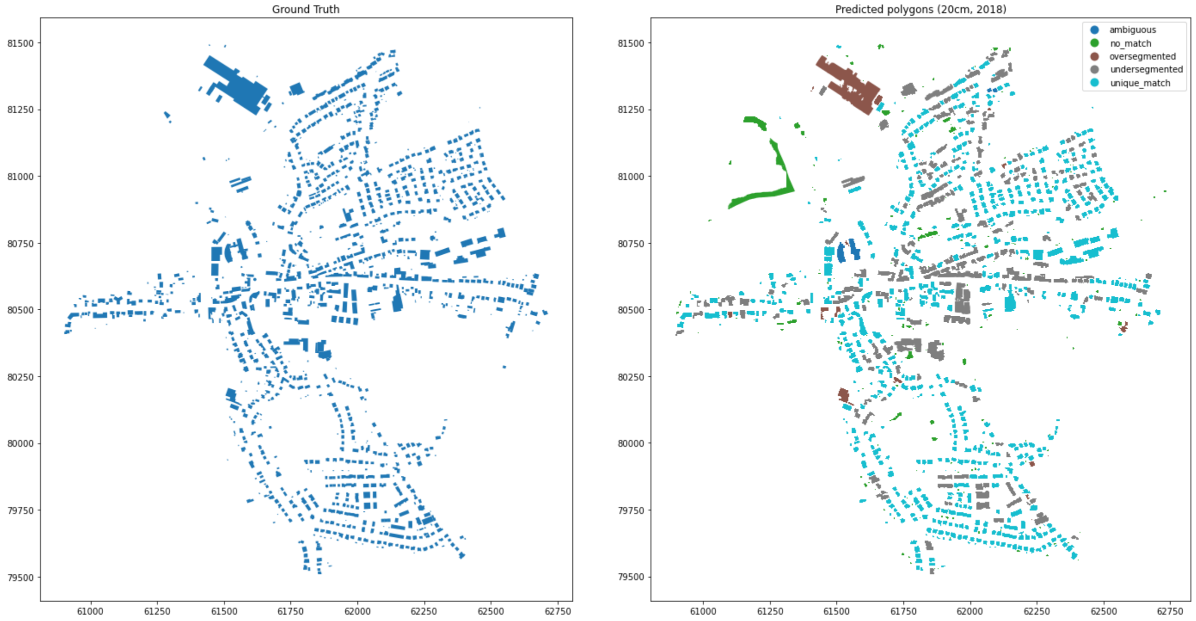


**Figure 53 - Predicted polygons (right) and ground truth (left) for  AOI 4 for the baseline DeepResUNet model trained with the WBCE loss function. Predicted polygons are colour-coded by status: light blue for unique matches, gray for undersegmented, brown for oversegmented, green for no match (false detection) and blue for ambiguously segmented.**

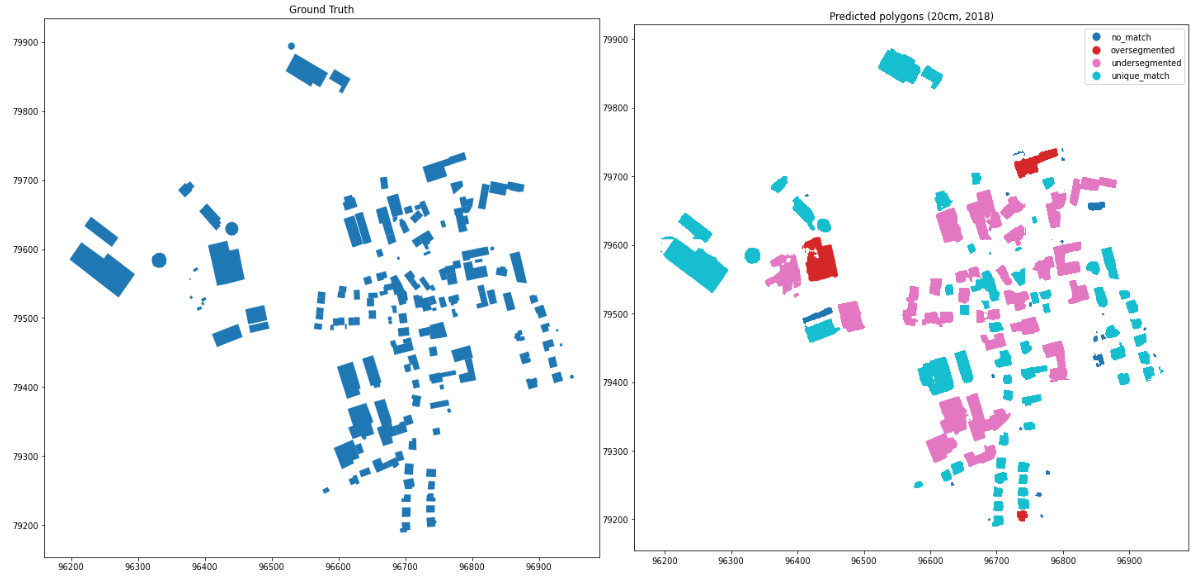


**Figure 54 - Predicted polygons (right) and ground truth (left) for AOI 5 for the baseline DeepResUNet model trained with the WBCE loss function. Predicted polygons are colour-coded by status: light blue for unique matches, pink for undersegmented, red for oversegmented, and blue for no match (false detection).**

In Figure 55 the status distribution of the predicted polygons across all AOIs is shown. Around 40% of the polygons derived are unique matches with ground truth, while undersegmentation is still a significant issue affecting around 35% of the predicted buildings. The 20cm results still represent a significant improvement over the 1m model where undersegmentation affected

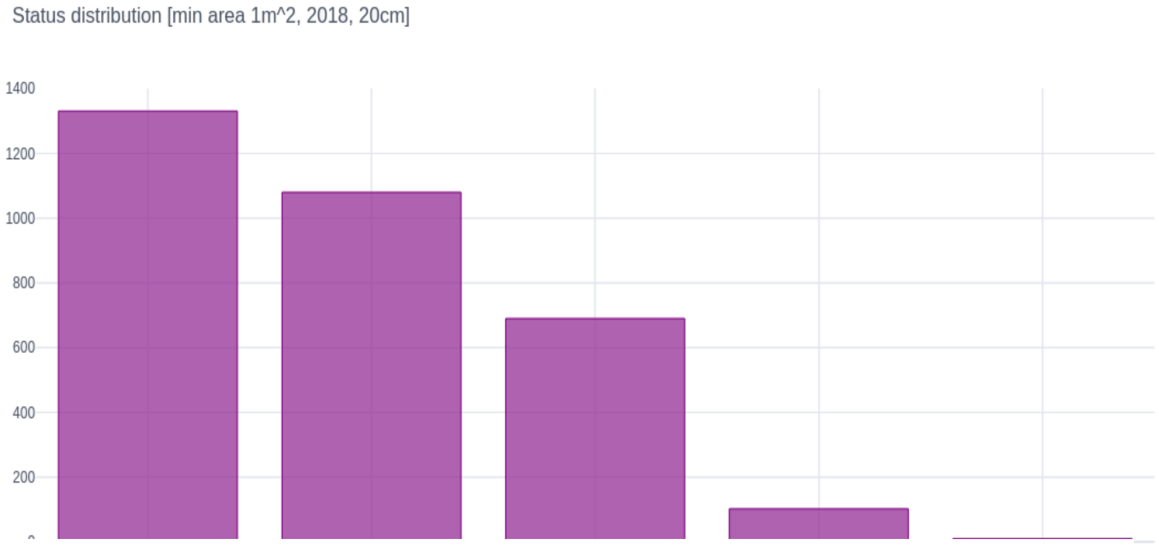


**Figure 55 - Global status distribution for the predicted polygons for the baseline DeepResUNet model trained with WBCE loss**

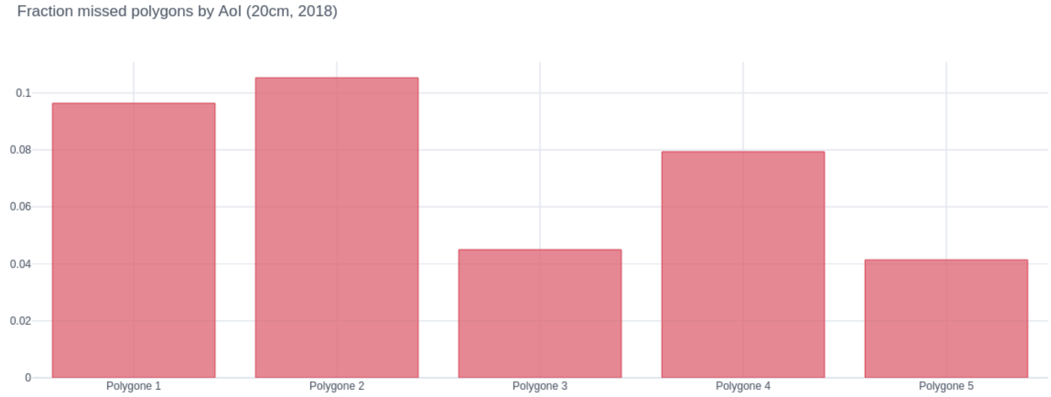


**Figure 56 - Fraction of missed buildings per AOI for the baseline DeepResUNet model trained with the WBCE loss**

In Figure 56 the fraction of missed buildings per AOI is depicted. This does not improve significantly with respect to the 1m results but is still typically in a reasonable range of 4-10%.

In Figures 57-61 the per-predicted-building segmentation quality is shown for those cases where the predicted polygon matches at least one in the ground truth polygon.

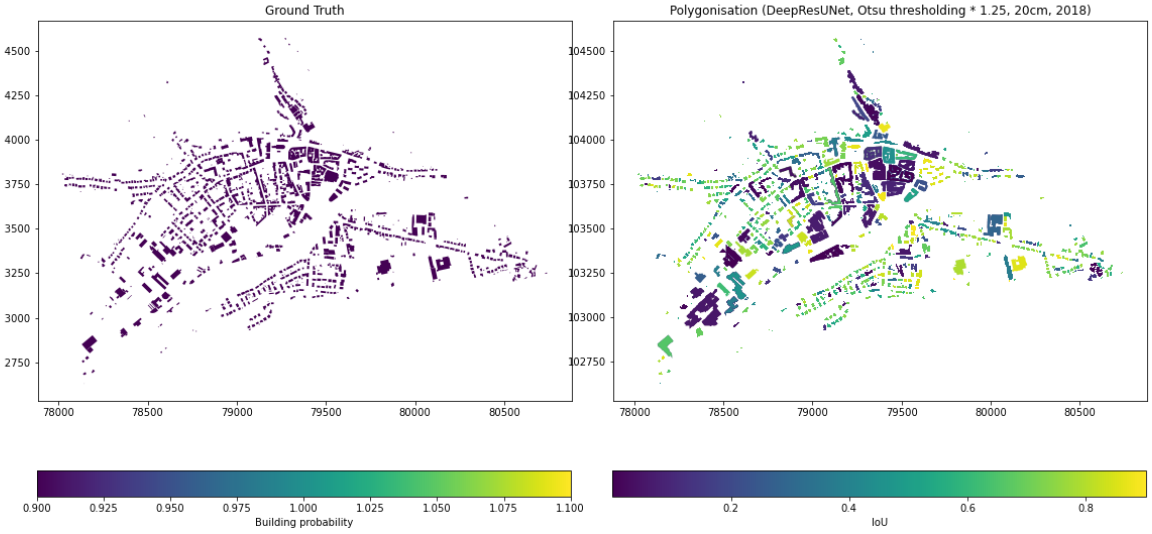


**Figure 57 - Predicted polygons which uniquely intersect true polygons in AOI 1, coloured by average IoU with the buildings they intersect. Undersegmented buildings appear as purple-blue and occur where a single predicted polygon erroneously contains multiple true buildings.**

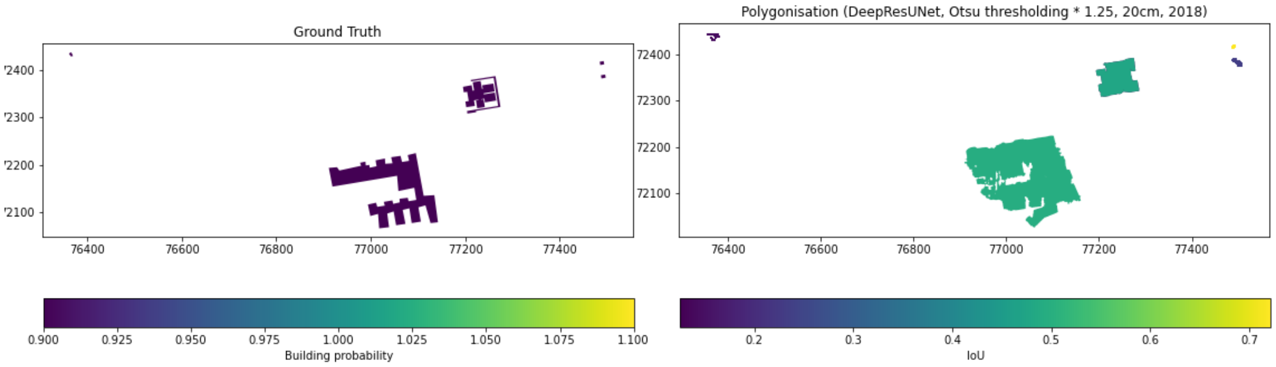


**Figure 58 -- Predicted polygons which uniquely intersect true polygons in AOI 2, coloured by average IoU with the buildings they intersect. Footprint quality is again poor in this AOI due to the complex multi-tiered nature of the buildings.**

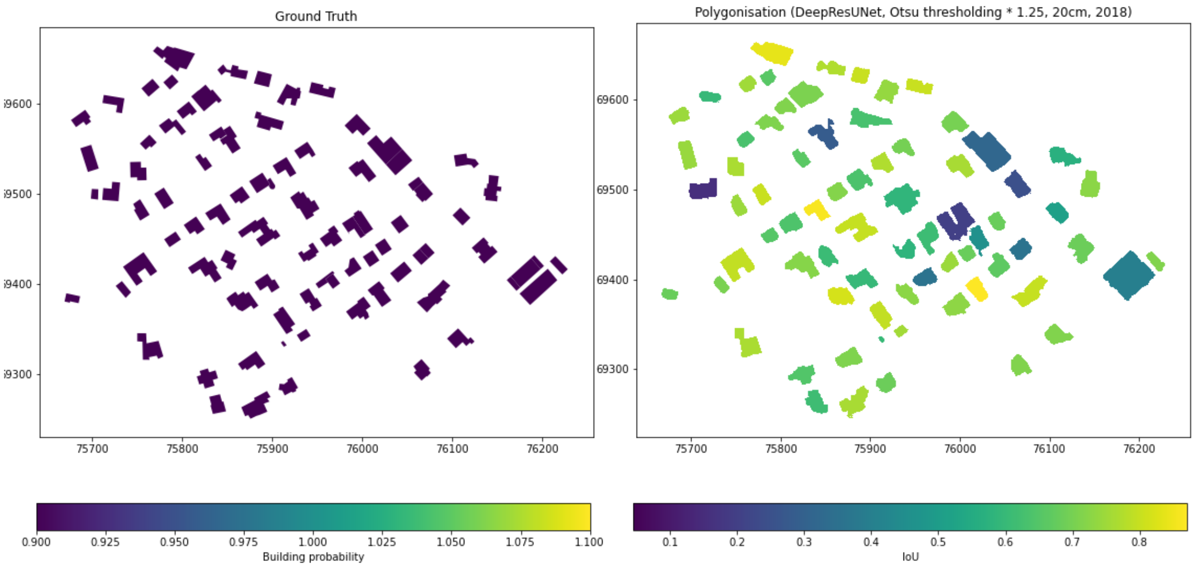


**Figure 59 - Predicted polygons which uniquely intersect true polygons in AOI 3, coloured by average IoU with the buildings they intersect. These well-separated rural buildings again have the best match quality, often around 80%.**

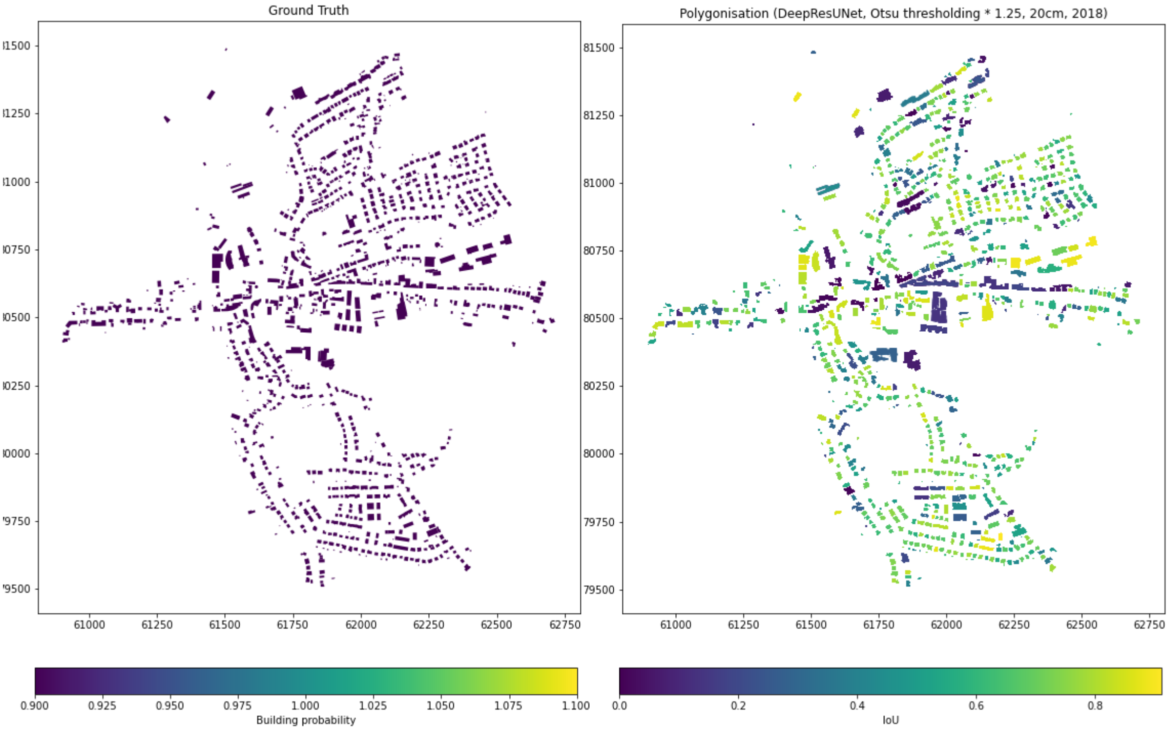

Ground Truth
Polygonisation (DeepResUNet, Otsu thresholding * 1.25, 20cm, 2018)
Building probability
IoU

**Figure 60 - Predicted polygons which uniquely intersect true polygons in AOI 4, coloured by average IoU with the buildings they intersect.**

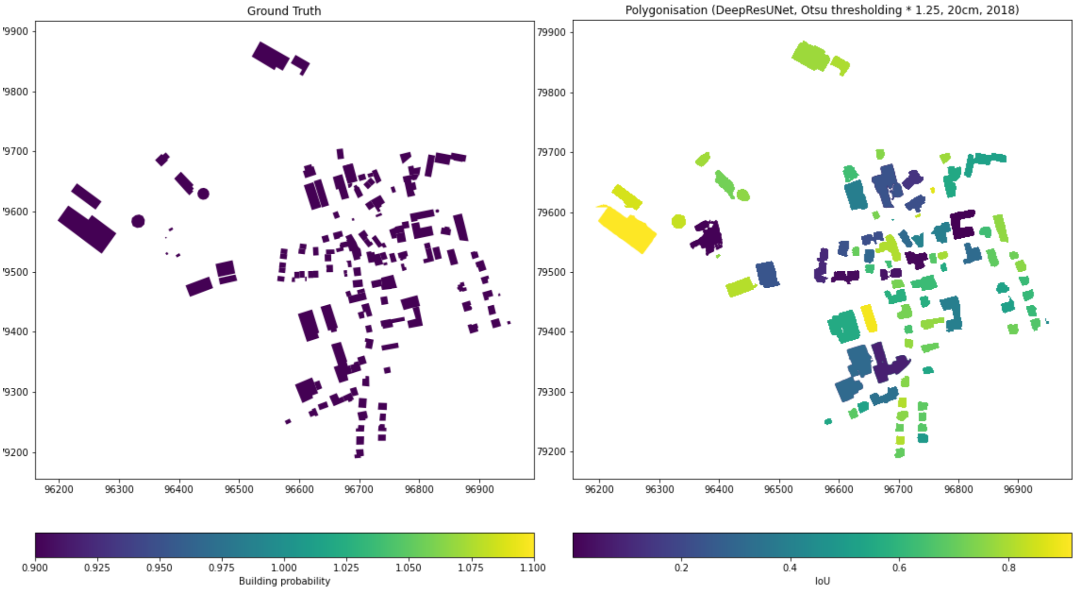

Ground Truth
Polygonisation (DeepResUNet, Otsu thresholding * 1.25, 20cm, 2018)
Building probability
IoU

**Figure 61 - Predicted polygons which uniquely intersect true polygons in AOI 5, coloured by average IoU with the buildings they intersect.**

In Figure 63 the global IoU distributions are shown for each AOI. These improve significantly on the 1m results, with the highest density of buildings in the 70-80% range with the exception of AOI 2.

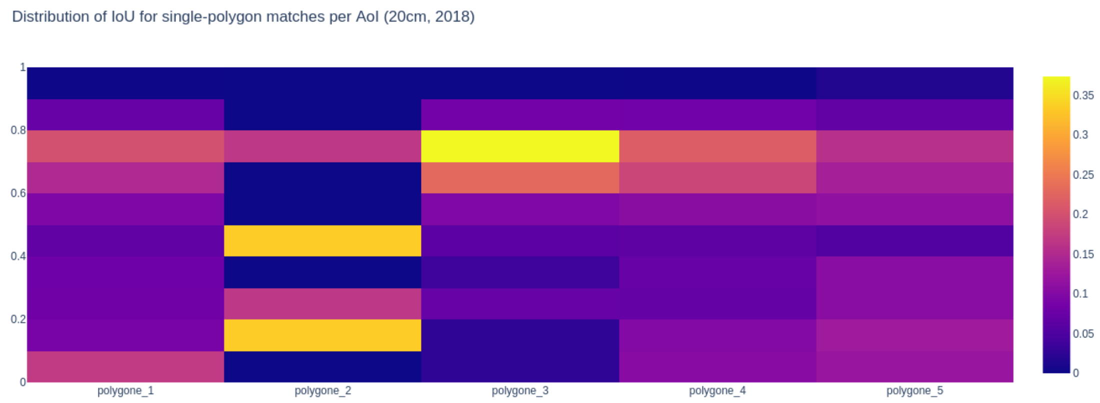


**Figure 63 - IoU distributions of predicted buildings which match at least one ground truth building per AOI.**

#### 3.3.5.6 Polygon- level results – DeepResUNet + all enhancements + WBCE loss

Here the polygon-level results for the best-performing WBCE model with all architectural enhancements are shown. For the sake of brevity only the aggregate predicted polygon distributions are shown here, as the polygon-level results are qualitatively similar to those in the previous section (the baseline DeepResUNet) and should be improved by re-thresholding which couldn't be performed again for the analysis here due to time constraints.

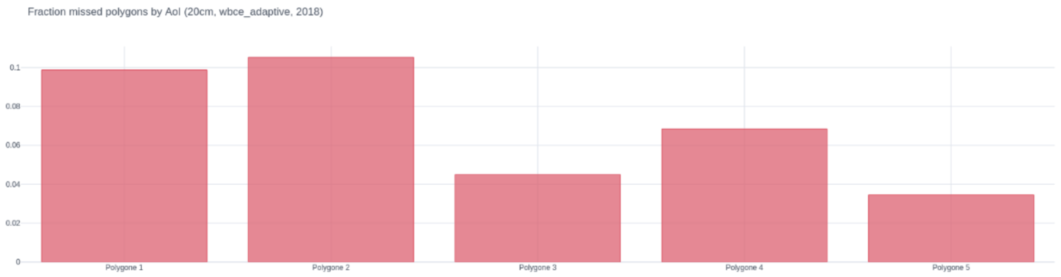


**Figure 64 - Fraction of missed buildings by AOI for the best-performing WBCE model. These are relatively unchanged with respect to the baseline 20cm result and the 1m results.**

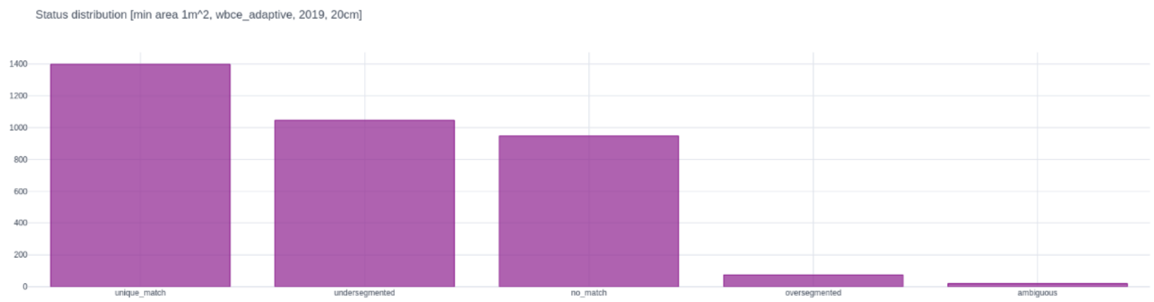


**Figure 65 - Status distributions for the best-performing model (with suboptimal threshold) trained with the WBCE loss function.**

In Figure 65 the global predicted building status distribution is shown. In this case the no-match rate is slightly higher than the baseline model (more false detections). This difference will likely be eliminated by more aggressive thresholding.

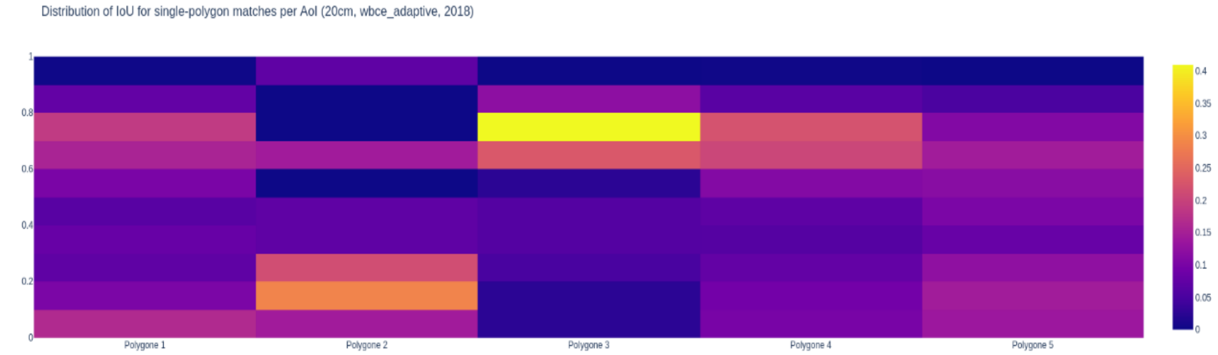


**Figure 66 - IoU distributions for predicted polygons which match at least one true polygon per AOI for the best-performing model trained with the WBCE loss function.**

In Figure 66 the global IoU distributions are shown for predicted buildings which match at least one true building. These are similar to those of the baseline DeepResUNet model, although the density of buildings in the peak IoU regions of 60-80% is slightly lower.

#### 3.3.5.7 Polygon- level results – DeepResUNet + all enhancements + Tversky loss

The results for the best-performing model trained with the Tversky loss function on polygon-level predictions are shown here.

In Figures 67-71 colour-coded representations of the predicted polygons are again shown. Undersegmentation remains the most prominent error for this loss function, but the rate at which it occurs is visibly less than for the WBCE.

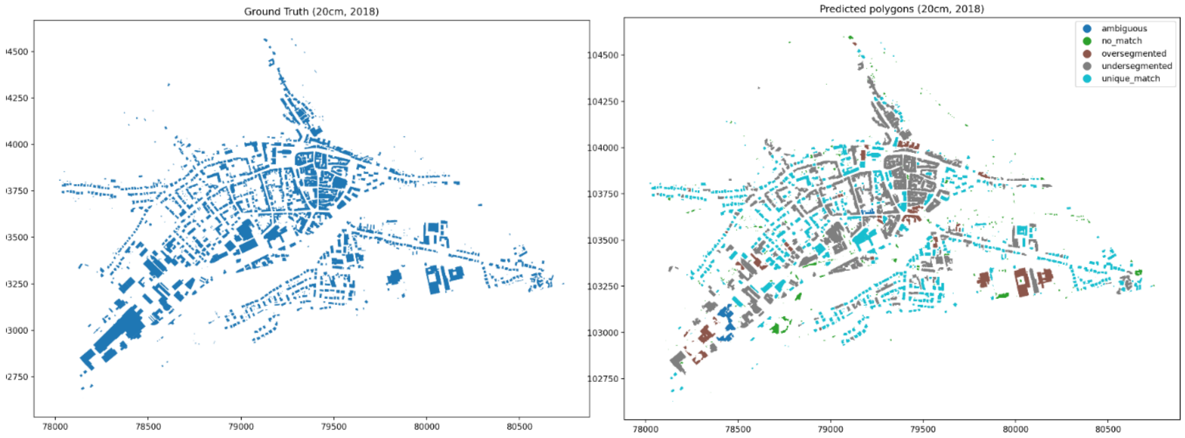


**Figure 67 - Predicted polygons (right) and ground truth (left) for AOI 1 for the best-performing model trained with the Tversky loss function. Predicted polygons are colour-coded by status: light blue for unique matches, gray for undersegmented, brown for oversegmented, green for no match (false detection) and blue for ambiguously segmented.**

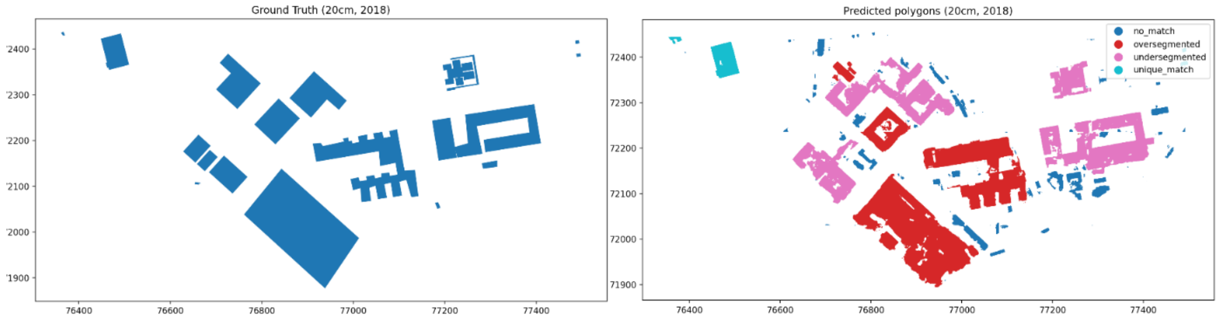


**Figure 68 - Predicted polygons (right) and ground truth (left) for AOI 2 for the best-performing model trained with the Tversky loss function. Predicted polygons are colour-coded by status: light blue for unique matches, pink for undersegmented, red for oversegmented, and blue for no match (false detection).**

Similarly, to the WBCE loss, the Tversky model struggles with very large buildings such as is visible in the southwest of AOI 1. This situation may be partly remedied by improving their representation in the training dataset (such large industrial buildings were lacking in the Belair training set). While the degree of undersegmentation is still significant, many cases are also “almost” correct, i.e. two polygons merge by merit of making contact in one small area. This may be addressed by experimenting with a postprocessing step consisting of morphological operations such as opening (a small erosion followed by a small dilation). This is recommended for a future exercise.

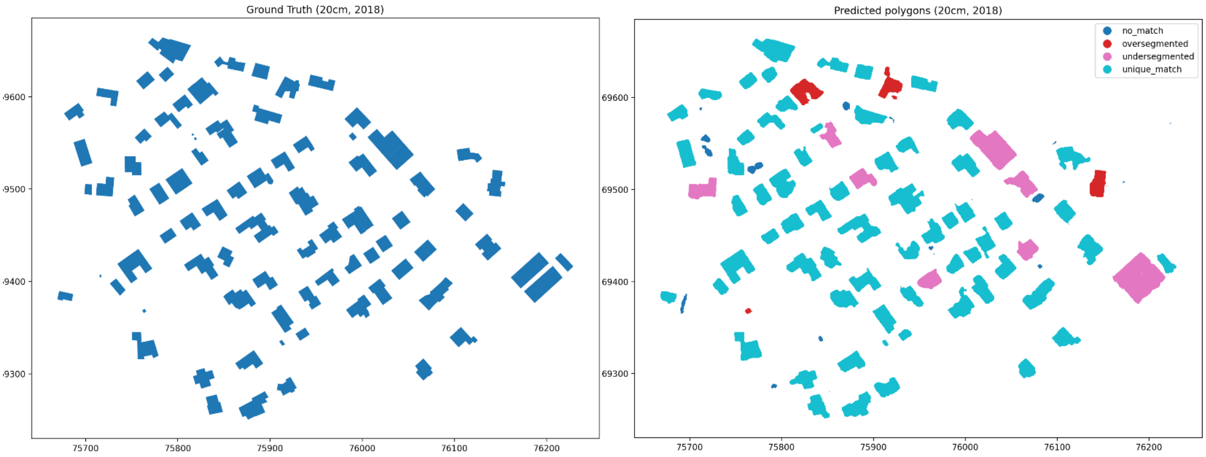


**Figure 69 - Predicted polygons (right) and ground truth (left) for AOI 3 for the best-performing model trained with the Tversky loss function. Predicted polygons are colour-coded by status: light blue for unique matches, pink for undersegmented, red for oversegmented, and blue for no match (false detection).**

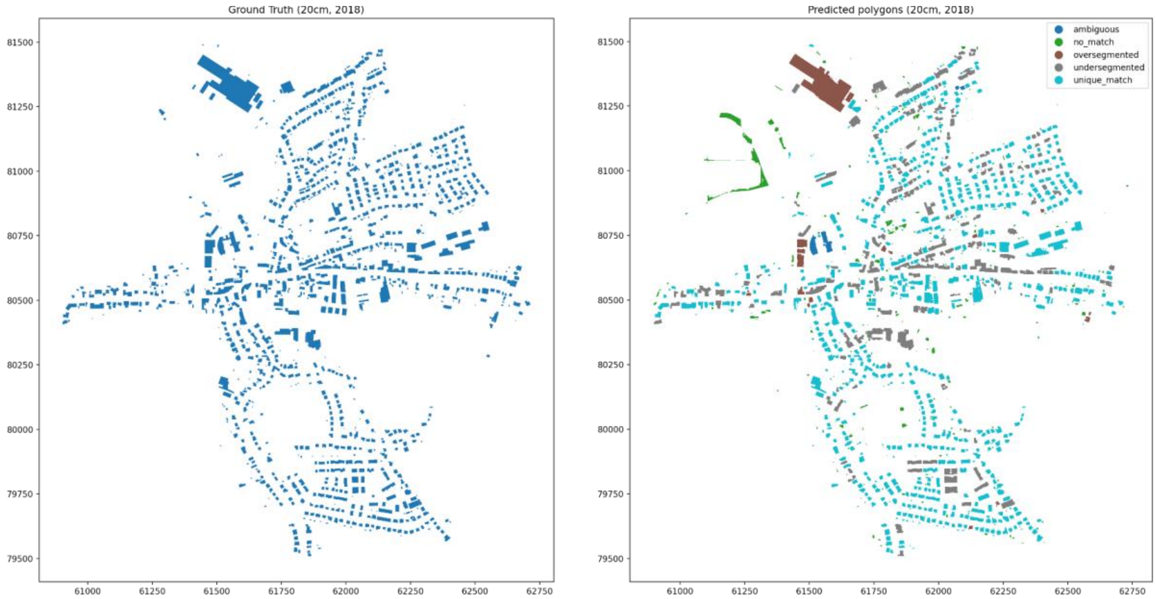


**Figure 70 - Predicted polygons (right) and ground truth (left) for  AOI 4 for the best-performing model trained with the Tversky loss function. Predicted polygons are colour-coded by status: light blue for unique matches, gray for undersegmented, brown for oversegmented, green for no match (false detection) and blue for ambiguously segmented.**

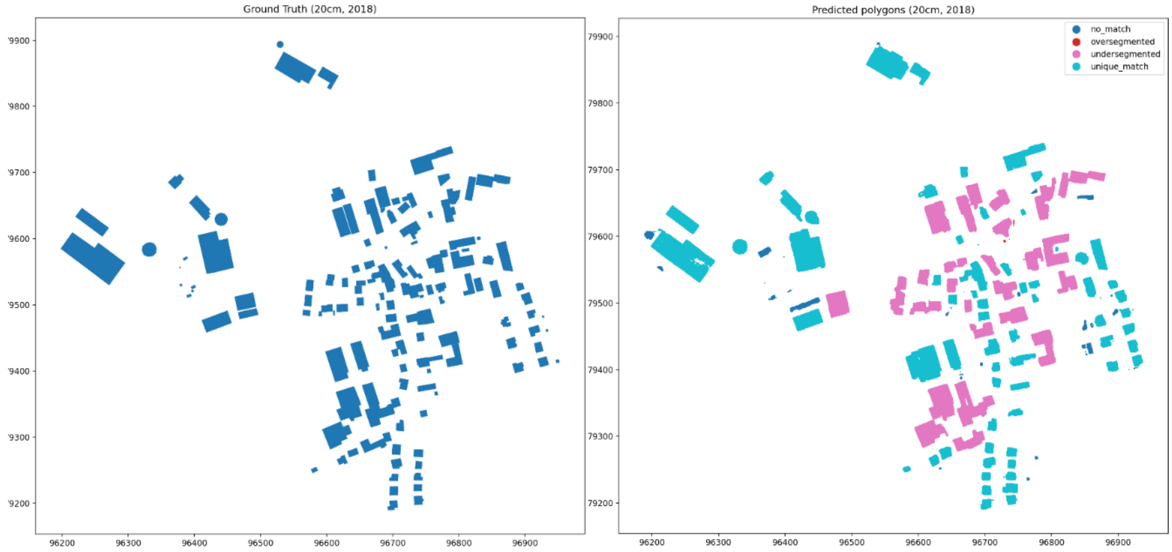


**Figure 71 - Predicted polygons (right) and ground truth (left) for  AOI 5 for the best-performing model trained with the Tversky loss function. Predicted polygons are colour-coded by status: light blue for unique matches, pink for undersegmented, red for oversegmented, and blue for no match (false detection).**

In Figure 72 the missed building rate per AOI is shown for the Tversky model. The rates are comparable to the WBCE loss (slightly worse in the case of AOI 5 from 4% to 8%). The sharper footprints from this loss function do not come at a significant cost in sensitivity to small buildings.

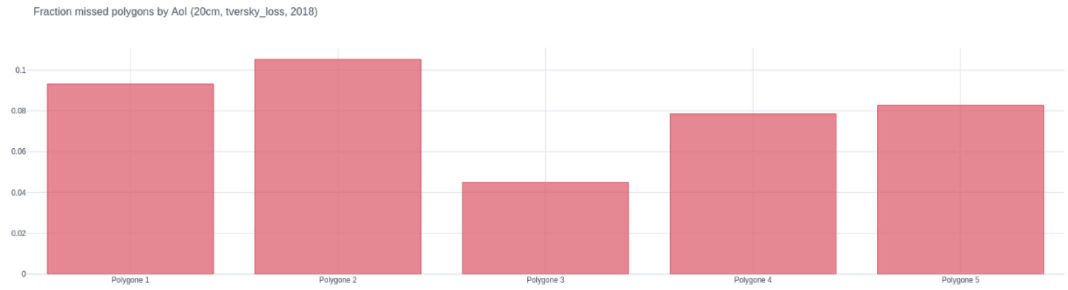


**Figure 72 - Fraction of missed buildings by  AOI for the best-performing Tversky model.**

Figure 73 displays the overall status distribution of the predicted building polygons. The proportion of unique matches improves to around 43% of all polygons, while the proportion of undersegmented buildings drops to around 27%. Just over a quarter of the predicted buildings have no match (false positives) which is a minor increase with respect to the WBCE loss, but we can see that the majority of these are very small and could also be eliminated by placing a size cut at the level of a few squared metres.

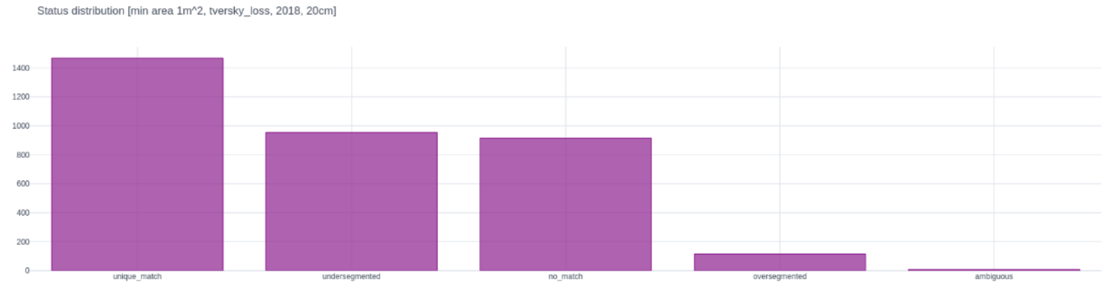


**Figure 73 – Global status distributions for the best-performing Tversky loss model.**

In Figures 74-78 the average IoU distributions are shown for the predicted polygons for the Tversky loss. There is a noticeable improvement in quality for the uniquely matching building polygons with respect to the WBCE loss.

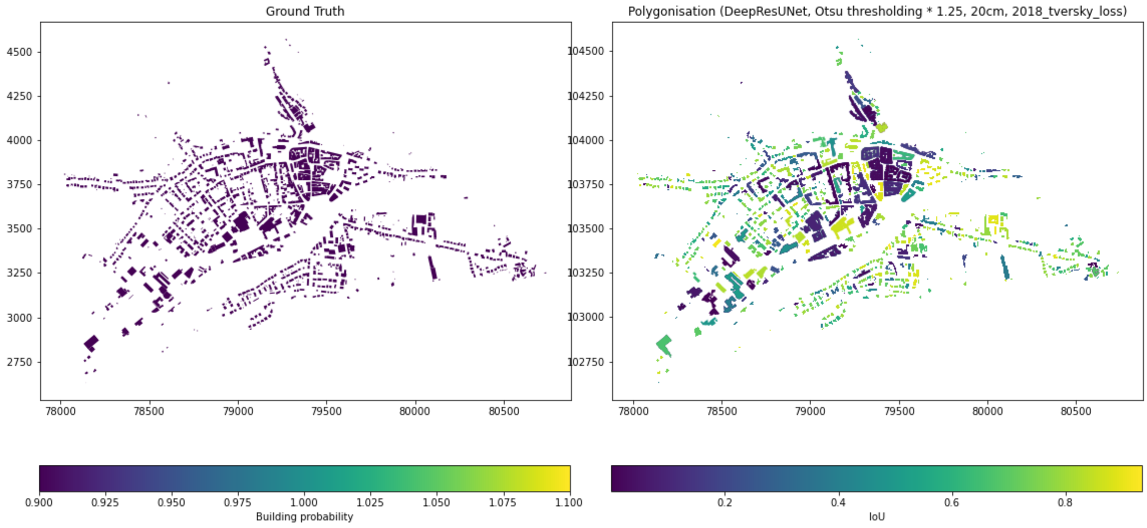


**Figure 74 - Predicted polygons which uniquely intersect true polygons in AOI 1, coloured by average IoU with the buildings they intersect. Undersegmented buildings appear as purple-blue and occur where a single predicted polygon erroneously contains multiple true buildings.**

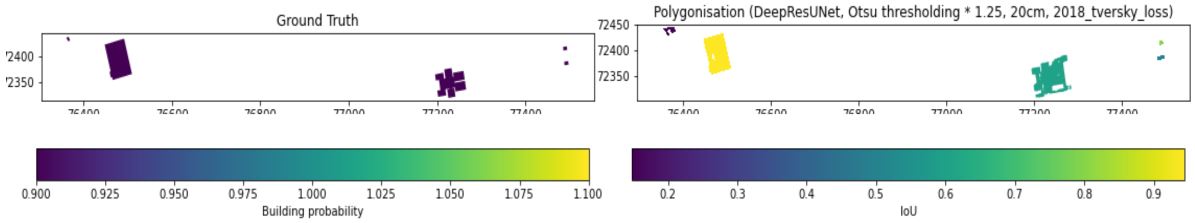


**Figure 75 - Predicted polygons which uniquely intersect true polygons in AOI 2, coloured by average IoU with the buildings they intersect.**

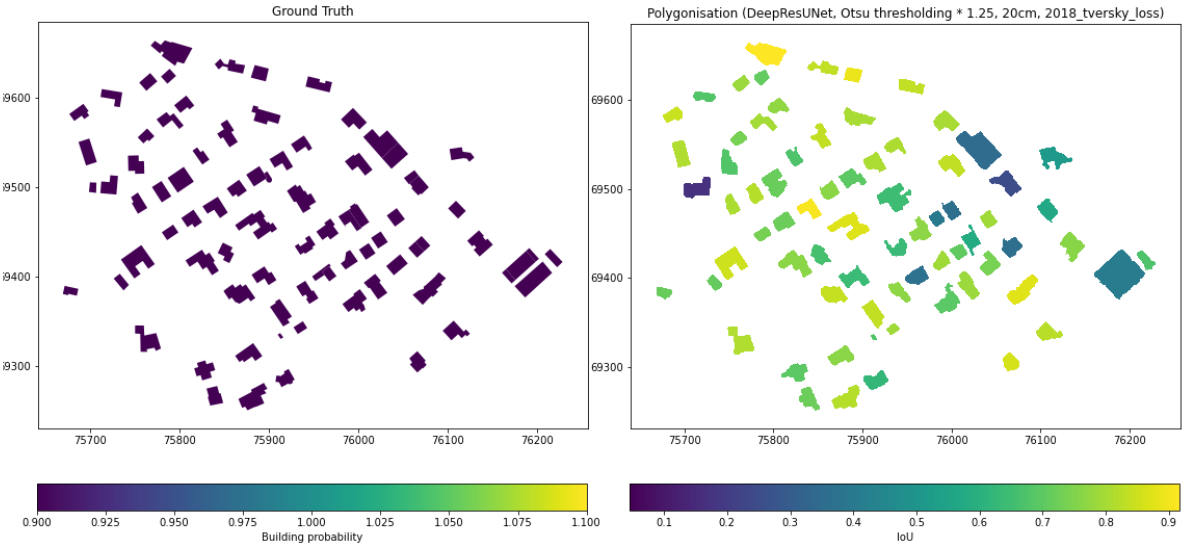


**Figure 76 - Predicted polygons which uniquely intersect true polygons in AOI 3, coloured by average IoU with the buildings they intersect.**

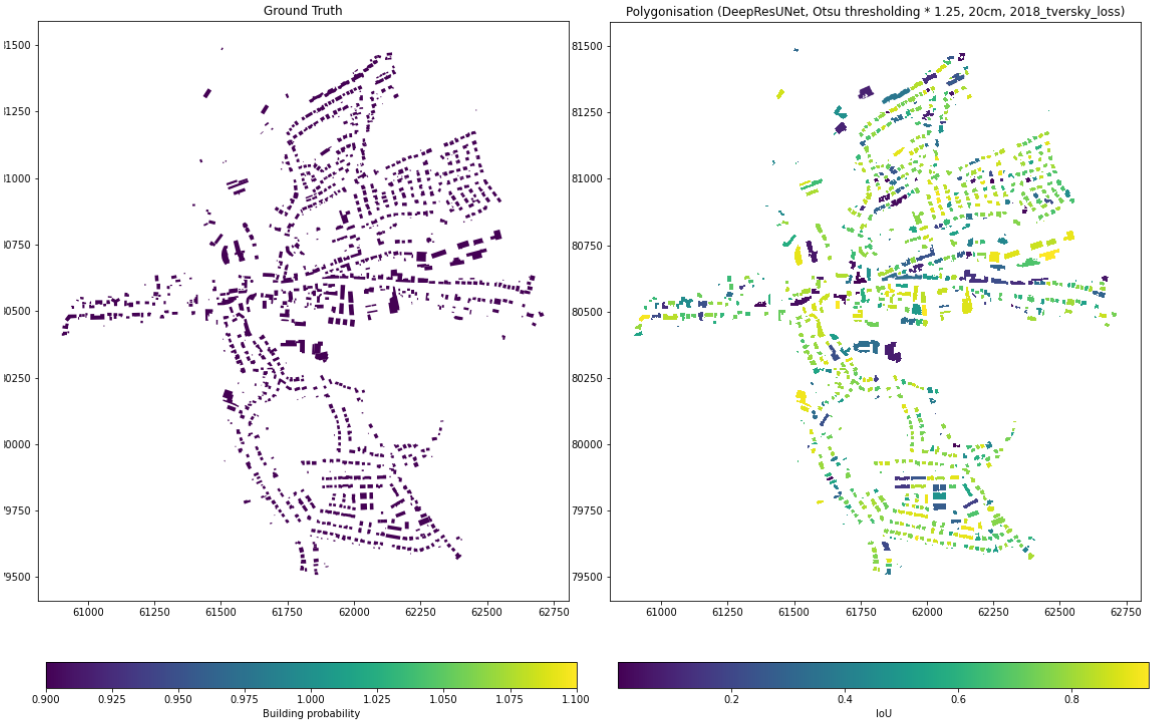


**Figure 77 - Predicted polygons which uniquely intersect true polygons in AOI 4, coloured by average IoU with the buildings they intersect.**

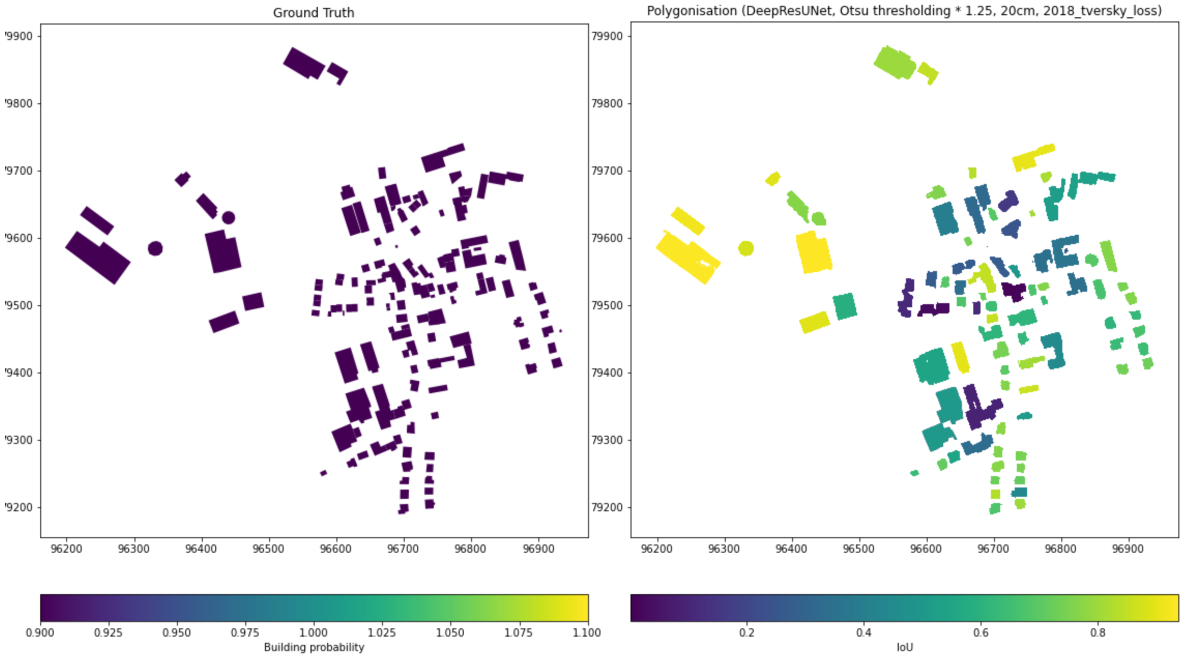


**Figure 78 - Predicted polygons which uniquely intersect true polygons in AOI 5, coloured by average IoU with the buildings they intersect.**

In Figure 79 we observe the IoU distributions for matching predicted polygons for the Tversky loss. A larger fraction of the total buildings is concentrated around the peak IoU values of 70-80% for each AOI, with the exception of AOI 2 where oversegmentation excludes most of the buildings. We can conclude from this, and the missing building rates being relatively unchanged, that the Tversky loss is superior at the polygon level and is recommended over the WBCE loss for generating future results.

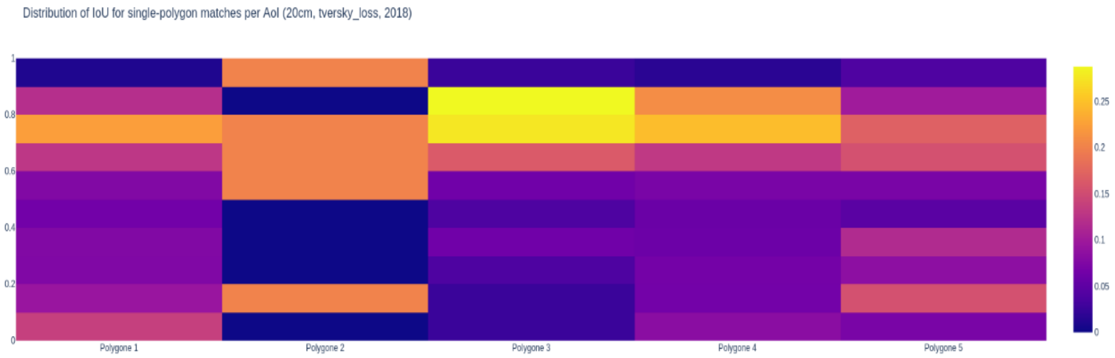


**Figure 79 - IoU distributions for predicted polygons which match at least one true polygon per  AOI for the best-performing model trained with the Tversky loss function.**

#### 3.3.5.8 Polygon- level results – DeepResUNet + all enhancements + Dice loss

The final polygon-level results to be presented are for the best-performing model (with all architectural enhancements) for the Dice loss function.

In Figures 80-84 colour-coded representations of the predicted polygons are again shown. The degree and extent of undersegmentation is reduced with respect to the Tversky loss (most undersegmented cases are mismerged polygons with a very small degree of contact). This comes at the cost of more frequent oversegmentation.

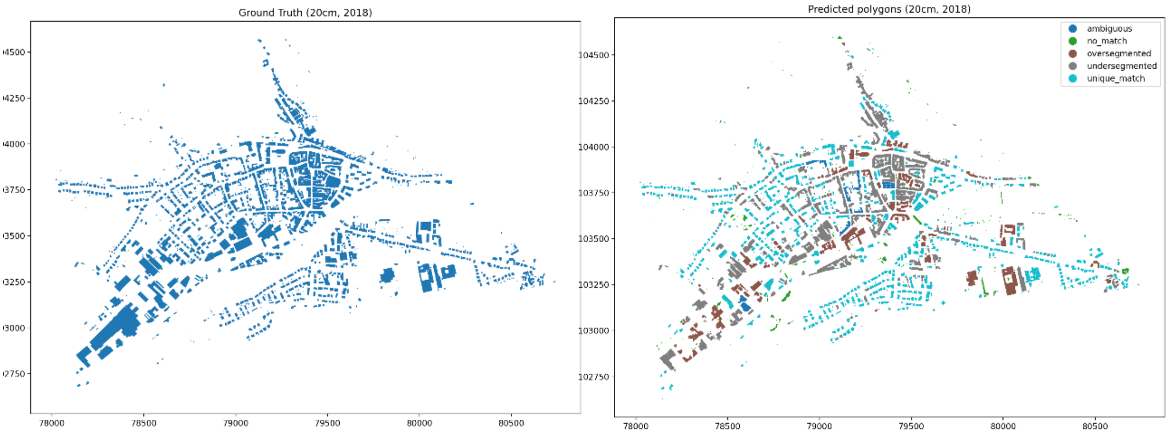


**Figure 80 - - Predicted polygons (right) and ground truth (left) for AOI 1 for the best-performing model trained with the Dice loss function. Predicted polygons are colour-coded by status: light blue for unique matches, gray for undersegmented, brown for oversegmented, green for no match (false detection) and blue for ambiguously segmented.**

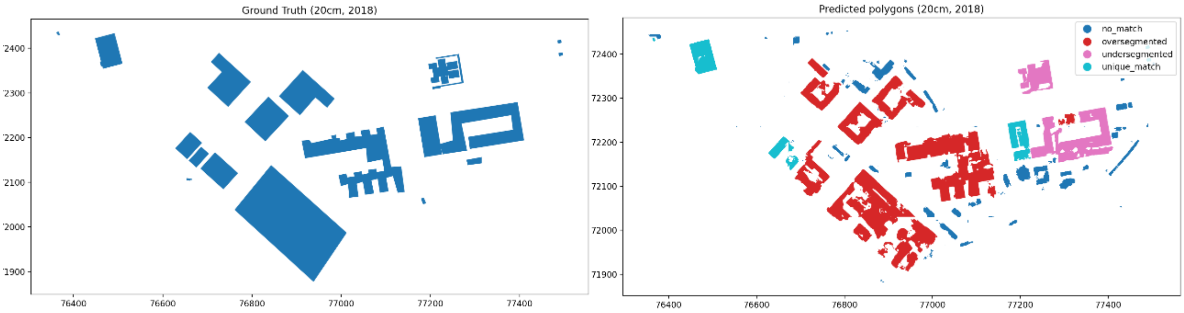


**Figure 81 - Predicted polygons (right) and ground truth (left) for AOI 2 for the best-performing model trained with the Dice loss function. Predicted polygons are colour-coded by status: light blue for unique matches, pink for undersegmented, red for oversegmented, and blue for no match (false detection).**

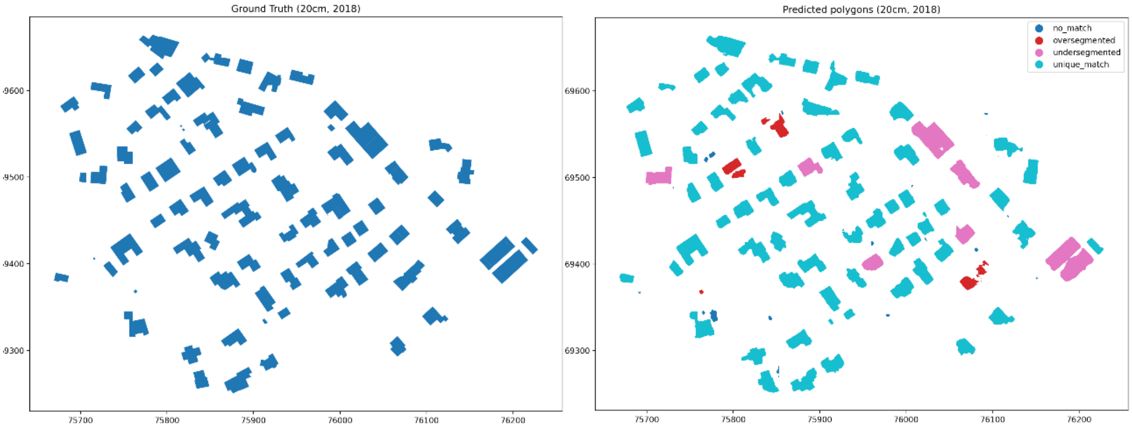


**Figure 82 - Predicted polygons (right) and ground truth (left) for AOI 3 for the best-performing model trained with the Dice loss function. Predicted polygons are colour-coded by status: light blue for unique matches, pink for undersegmented, red for oversegmented, and blue for no match (false detection).**

The building footprints tend to be more conservative with respect to the Tversky model, and in the cases of unique matches are often of higher quality. This traces back to the relative penalty of false positives being higher in the loss function. The frequency of false detections is also noticeably smaller, which may make this loss function a more pragmatic choice when searching for high-quality footprints of unknown buildings that are not particularly atypical (and thus prone to oversegmentation).

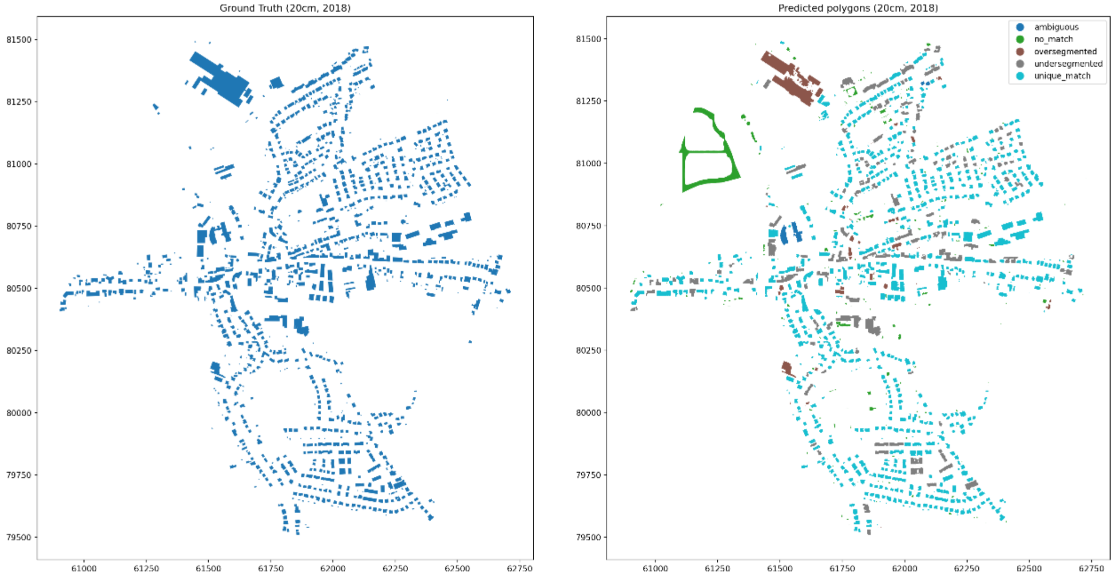


**Figure 83 - Predicted polygons (right) and ground truth (left) for AOI 4 for the best-performing model trained with the Dice loss function. Predicted polygons are colour-coded by status: light blue for unique matches, gray for undersegmented, brown for oversegmented, green for no match (false detection) and blue for ambiguously segmented.**

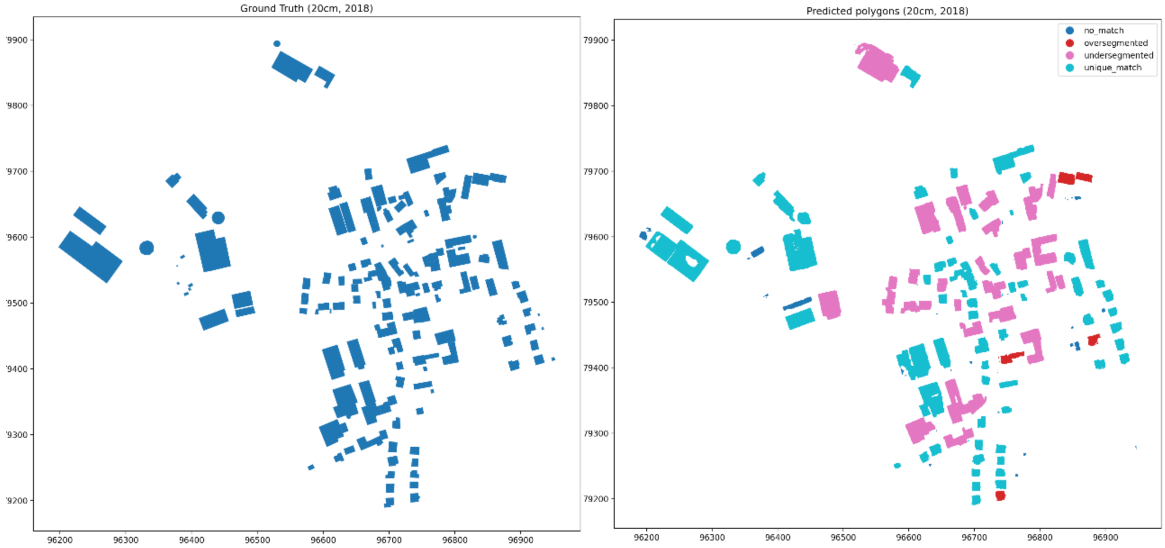


**Figure 84 - Predicted polygons (right) and ground truth (left) for AOI 5 for the best-performing model trained with the Dice loss function. Predicted polygons are colour-coded by status: light blue for unique matches, pink for undersegmented, red for oversegmented, and blue for no match (false detection).**

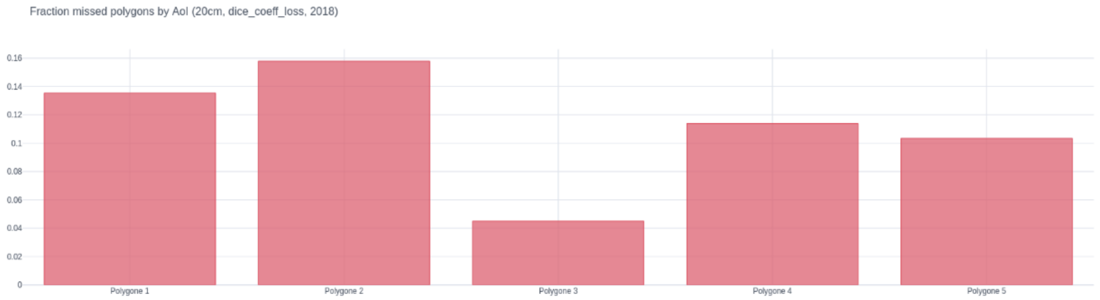


**Figure 85 - Fraction of missed buildings by AOI for the best-performing Dice model.**

In Figure 85 we can observe the cost of the more conservative nature of the Dice loss in flagging buildings as an increase in the missed building rate. For the typical AOIs this is in the 10-14% range, with the majority of these being smaller buildings. For identifying smaller buildings, it may be then more appropriate to use the model trained with the Tversky loss, although this will also result in a higher frequency of false detections.

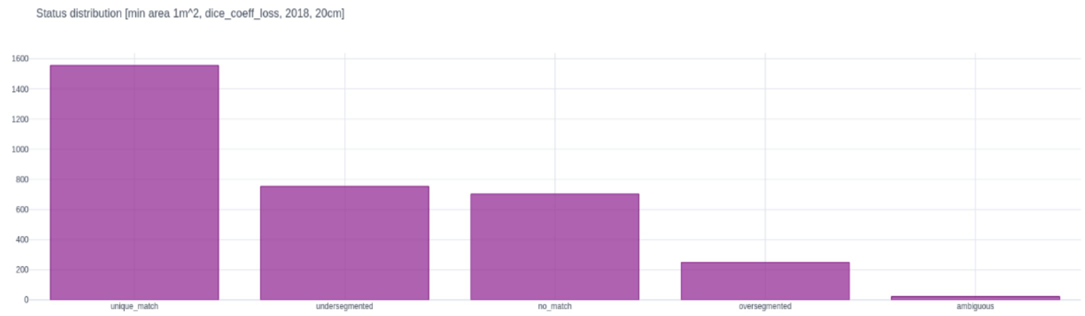


**Figure 86 - Global status distributions for the best-performing Dice loss model.**

Figure 86 displays the global status distribution for predicted polygons. The frequency of unique matches with ground truth is around 48%, a 5% improvement on the Tversky loss. The fraction of buildings undersegmented falls to around 23%, while the fraction oversegmented increases to around 6% where the Tversky model produced around 3%. The fraction of false detections falls to around 21% from around 28% with the Tversky loss. This situation would for most use-cases render the Dice loss superior, provided the higher rate of missed detections is tolerable.

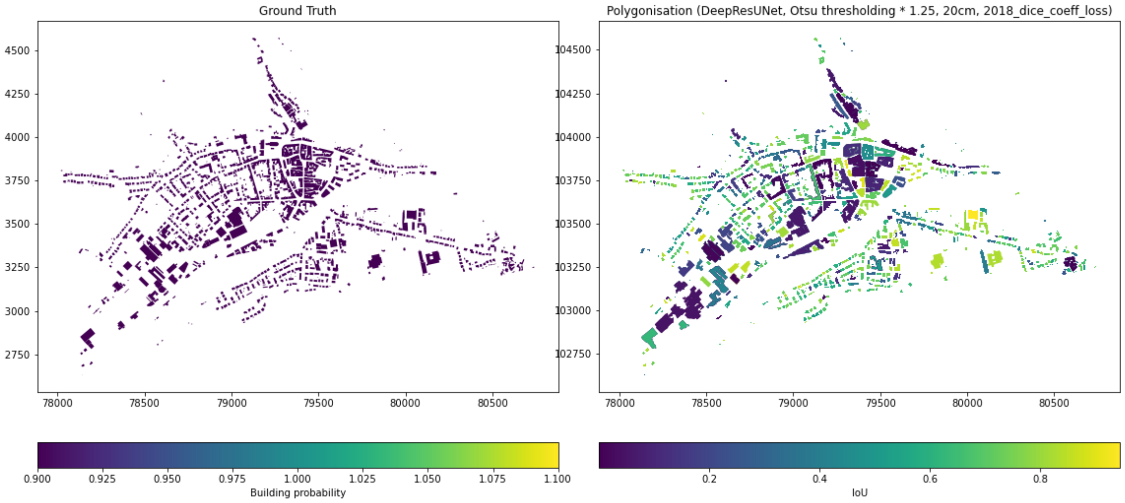


**Figure 87 - Predicted polygons which uniquely intersect true polygons in AOI 1, coloured by average IoU with the buildings they intersect.**

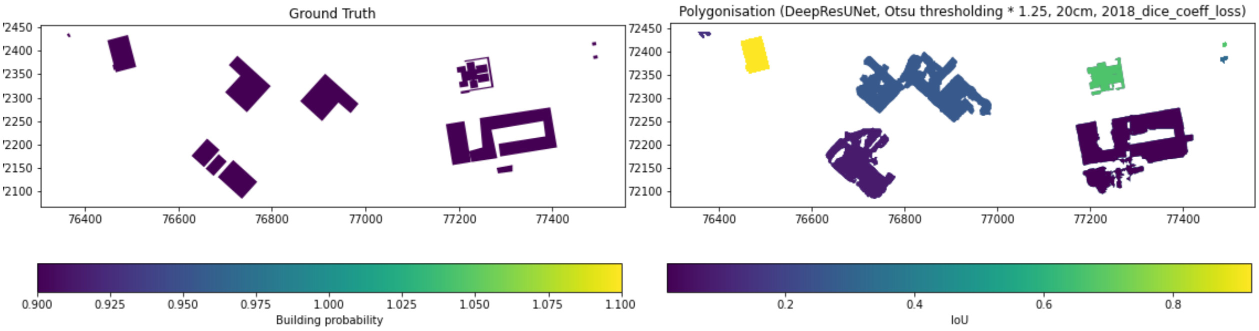


**Figure 88 - Predicted polygons which uniquely intersect true polygons in AOI 2, coloured by average IoU with the buildings they intersect.**

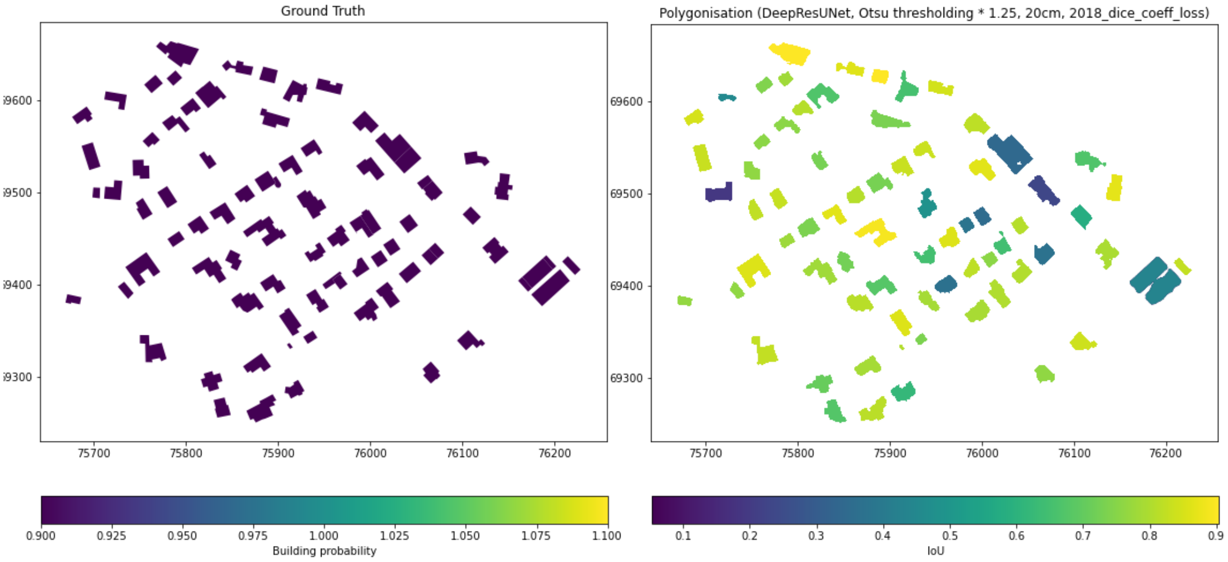


**Figure 89 - Predicted polygons which uniquely intersect true polygons in AOI 3, coloured by average IoU with the buildings they intersect.**

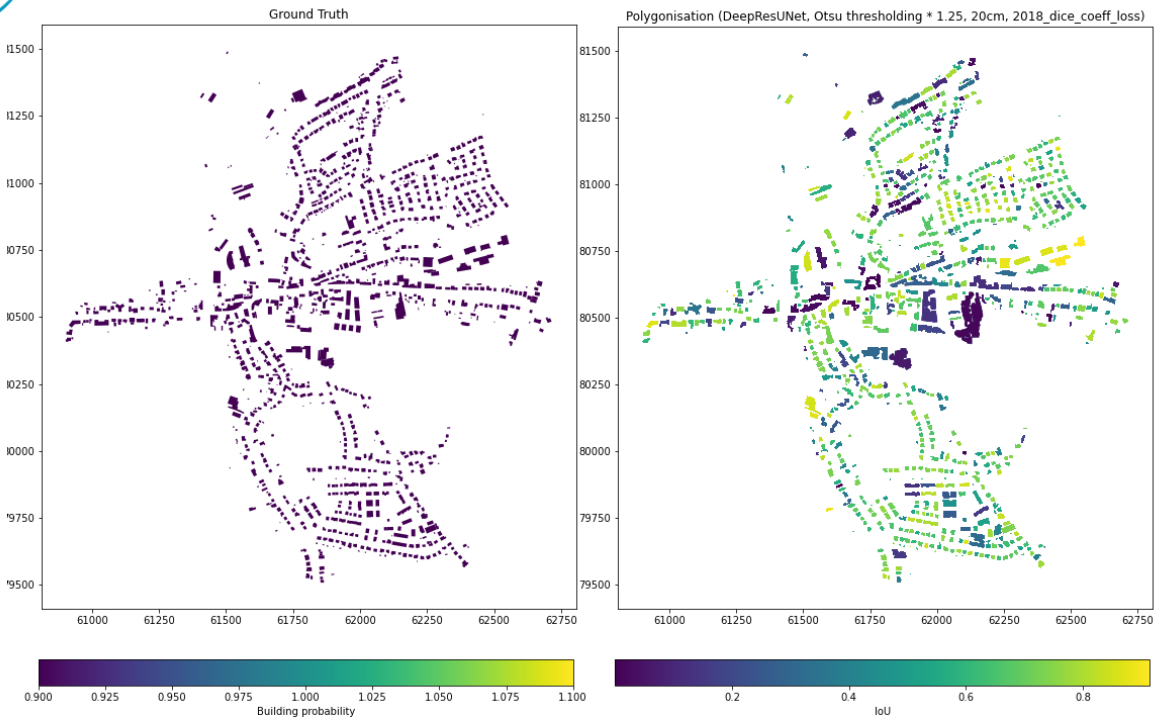


**Figure 90 - Predicted polygons which uniquely intersect true polygons in AOI 4, coloured by average IoU with the buildings they intersect.**

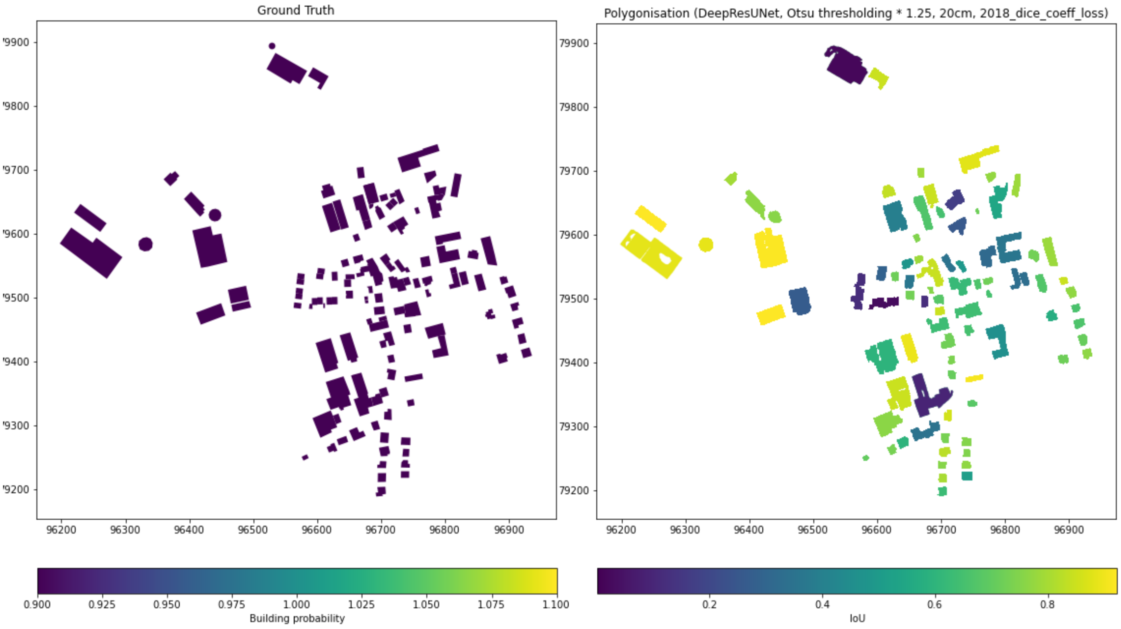


**Figure 91 - Predicted polygons which uniquely intersect true polygons in AOI 5, coloured by average IoU with the buildings they intersect.**

Figure 92 provides a final summary of the footprint quality for the Dice loss per AOI. The majority of detected buildings are concentrated in the 70-90% IoU region for each AOI to a more pronounced extent than with the Tversky loss. Together with the status distribution depicted in 85, this might be taken to mean that both the qualitative nature of detected polygons and the quantitative accuracy of their footprints are better, with the caveat that the probability to miss buildings is moderately higher for this loss function.

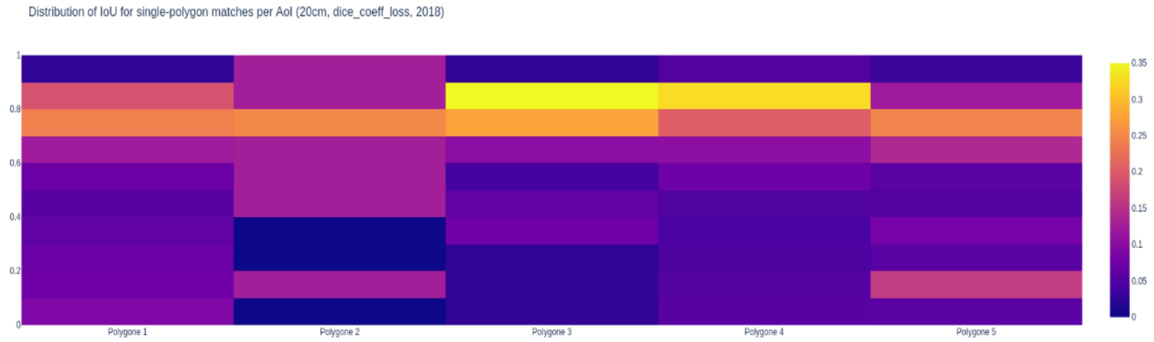


**Figure 92 - IoU distribution per AOI for the best-performing model trained with the Dice loss function.**

## 3.4 EXPERIMENTS WITH TRUE ORTOPHOTO IMAGERY

In the experiments conducted, segmentation and polygon level metrics were calculated for the five AOIs also using the 2019 (winter) true orthophoto resampled from 10cm spatial resolution to 20cm.

Here we observed very minor, percent-level performance degradations. The reasons for this are likely a combination of three factors:

- The RGB distributions in the training images used from this orthophoto are considerably different due to the winter conditions. Since the majority of the training data is from the summer captures (there was only one winter true orthophoto), it’s expected that the model performance will degrade in winter.
- Small deviations from the perpendicular zenith angle may actually be beneficial for a segmentation model for certain buildings in that it may learn to pick up on cues from partially visible facades such as windows.
- The ground truth used to evaluate the testing AoIs was only available for 2018, and there are a handful of visible changes to buildings that occur between 2018 and 2019. As such these will not be correctly evaluated this will introduce spurious performance losses.

It may be worthwhile for a future experiment to balance the proportion of true and regular orthophotos and compare captures from similar seasonal conditions to make a fair and definitive evaluation of which provides the best performance for segmentation. Minimal occlusion by trees and vegetation in winter conditions will in both cases likely lead to an improvement in footprint delineation in some cases.

To a first approximation based on the experiments carried out above it’s reasonable to assume that any performance gains from true orthophotos (evaluated fairly against their regular cousins) will be minor.

# 4 Generating and improving results

All of the models and machinery used to carry out the analysis in section 3, along with the capability of training and running inference with new models and datasets is included in the tool provided to ACT.

Building detection and segmentation can be accomplished by running inference on GeoTIFF raster data from different years or regions. Change detection is possible by comparing these results, particularly at the polygon level, in GeoICT tools such as QGIS.

## 4.1 TRAINING

The possibility to train new models is included in the Extopia segmentation framework as a convenient script which allows one to enable and disable each of the architectural enhancements implemented, and tune model hyperparameters to obtain potentially even better results.

## 4.2 INFERENCE

One may define new datasets for inference by providing RGB GeoTIFF raster data. The provided framework includes a script which performs inference with the best currently trained model with a given loss function.

## 4.3 POLYGONISATION

Polygonisation is achieved using GDAL’s 4-connectedness polygonization algorithm. The tool includes a script which runs this on large segmentation raster datasets with a given binarisation threshold.

## 4.4 EVALUATION OF RESULTS

The generation of image- and polygon-level segmentation quality metrics, including all of the plots and images shown above, is included in the framework as an evaluation notebook which may be used to benchmark new models when new data becomes available.

## 4.5 CORRECTING ERRONEOUS RESULTS

One may improve erroneous results by providing accurate ground truth footprints for the regions which were poorly segmented and retraining a model with these corrections included. These must take the form of an ESRI shapefile with the ground truth data and the appropriate RGB orthophotos in GeoTIFF format.

# 5 Conclusion

As a general conclusion to summarize the above analysis we can state the following:

As can be expected the result of the segmentation for 20cm **resolution** are considerably better than the results obtained with the 1m resolution data. The 1m spatial resolution is not sufficient for distinguishing individual building polygons except for those cases where buildings are well-separated. When this is the case the footprint quality is mediocre and tends to overexaggerate building sizes. The additional processing time for running the models at 20 cm resolution is not prohibitive seen the fact that the 20cm inference on the country of Luxembourg was possible in approximately 12 hours on the ACT hardware.

Standard **Image augmentations (**Random rotation, Random horizontal flip, Random homogenous RGB offset, Random affine transformations, Random gaussian noise, Random gaussian blur, Random contrast shifts, Random brightness shifts) in addition to a data-driven PCA-based colour augmentation helped to improve both the quality of the segmentation (~15% lower loss values) and the robustness of the model.

Several **algorithmic improvements** were made to the base DeepRESUNET model**.** The additional architectural elements (spatial attention gates, deep supervision, multi-scale pooling and CBAM modules on each residual block), all improved the segmentation results. The final results were hence generated using a model trained with the Luxembourg (Belair) training sample with all of the above algorithmic enhancements applied.

Experiments were conducted with **3 different loss functions**. Weighted Binary Cross Entropy (WBCE), Tversky and Dice.

- The Weighted Binary Cross Entropy is the simplest of the three and is considered as the baseline.
- The Tversky loss function brings a qualitative improvement on the weighted binary cross entropy function, particularly in the reduction of false positives with the precision values touching the 80% mark. Undersegmentation remains the most prominent error for this loss function, but the rate at which it occurs is visibly less than for the WBCE. The sharper footprints from this loss function do not come at a significant cost in sensitivity to small buildings. On the downside, some stability issues were observed in the training.
- The Dice loss functions trades away the false positives for false negatives as compared to Tversky. This can be traced back to the equal weighting factors for these types of error in this loss function. As a result, for the more typical AOIs most building footprints are more accurate and not exaggerated. Smaller, isolated false positives are also much less frequent. On the downside as compared to the Tversky results, a larger fraction is missed.

From the above it is clear that both Tversky and Dice outperform WBCE. Which one to select depends on the specific use case. Tversky is the best choice if the idea is to map the maximum of uncharted buildings and obtain reasonable-quality footprints since its segmentation is less conservative than Dice. As a consequence, there will be also more false positives.  Dice on the other hand gives the best possible footprints and much less false positives but will also miss more buildings.

# 6 Bibliography

Abraham, K. (2019). *A Novel Focal Tversky loss function with improved Attention U-Net for lesion segmentation.* Retrieved from https://arxiv.org/abs/1810.07842

Audebert, N. (2017, 11 23). Beyond RGB: Very High Resolution Urban Remote Sensing With Multimodal Deep Networks. doi:arXiv:1711.08681v1

Badrinarayanan, V. (2017, 12 1). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 2481-2495. doi:10.1109/TPAMI.2016.2644615

Cao, Z. (2019, 11). End-to-End DSM Fusion Networks for Semantic Segmentation in High-Resolution Aerial Images. *IEEE Geoscience and Remote Sensing Letters vol. 16 no. 11*, pp. pp. 1766-1770. doi: 10.1109/LGRS.2019.2907009.

Chen Wu, L. Z. (2017). Kernel Slow Feature Analysis for Scene Change Detection. *IEEE Transactions on Geoscience and Remote Sensing* , 2367-2384. doi:10.1109/TGRS.2016.2642125

Chollet, F. (2017, 4 4). Xception: Deep Learning with Depthwise Separable Convolutions. doi:arXiv:1610.02357v3

Chong, Y. (2020). HCNet: Hierarchical Context Network for Semantic Segmentation. doi:arXiv:2010.04962v2

de Jong, K. L., & Bosman, A. S. (2019, 03 21). Unsupervised Change Detection in Satellite Images. doi:arXiv:1812.05815v2

Emmanuel Maggiori, Y. T. (2017). Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark”. I. *EEE International Geoscience and Remote Sensing Symposium (IGARSS). 2017.*

Holländer, B. (2019, 9 25). *Self-Attention In Computer Vision.* Retrieved from Towards data science: https://towardsdatascience.com/self-attention-in-computer-vision-2782727021f6

Krizhevsky. (2012). *ImageNet Classification with Deep Convolutional Neural Networks.* Retrieved from https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

Lennert, M. (2019). Creating Wallonia's new very high resolution land cover maps. *FOSS4G 2019 – Academic Track*, (p. 7). Bucharest Romania. doi:10.5194/isprs-archives-XLII-4-W14-151-2019

McEver, R. A. (2020, 7 10). PCAMs: Weakly Supervised Semantic Segmentation Using Point Supervision. doi:arXiv:2007.05615v1

McInnes, L. (n.d.). Uniform Manifold Approximation and Projection for Dimension Reduction. doi:arxiv1802.03426

McKinley, R. (n.d.). Few-shot brain segmentation from weakly labeled data with deep heteroscedastic multi-task networks. doi:arXiv:1904.02436v1

Nabila Abraham, N. M. (2018). A Novel Focal Tversky loss function with improved Attention U-Net for lesion segmentation. doi:arXiv:1810.07842

Oktay, S. L. (2018). *Attention U-Net: Learning Where to Look for the Pancreas.* Retrieved from https://arxiv.org/abs/1804.03999

Ronneberger, O. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *MICCAI.* doi:arXiv:1505.04597

Syrris, V. (2019, 04 14). valuation of the Potential of Convolutional Neural Networks and Random Forests for Multi-Class Segmentation of Sentinel-2 Imagery. *Remote Sensing*. doi:https://doi.org/10.3390/rs11080907

Tao, A. (2020). Hierarchical Multi-Scale Attention for Semantic Segmentation. doi:arXiv:2005.10821

Tureckova, T. O.-S. (2020). *Improving CT Image Tumor Segmentation Through Deep Supervision and Attentional Gates.* Retrieved from https://www.frontiersin.org/articles/10.3389/frobt.2020.00106/full

Woo, P. L. (2018). *CBAM: Convolutional Block Attention Module.* Retrieved from https://arxiv.org/abs/1807.06521

Yaning, Y. (2019, 7). Semantic Segmentation of Urban Buildings from VHR Remote Sensing Imagery Using a Deep Convolutional Neural Network. *Remote Sensing*. doi:10.3390/rs11151774