Buildings Segmentation using the U-Net Convolutional Neural Network

Sibusiso Mgidi
School of Computer Science and Applied Mathematics
University of the Witwatersrand

Thabo Rachidi
School of Computer Science and Applied Mathematics
University of the Witwatersrand

***Abstract*—**

***Index Terms*—semantic segmentation, convolutional artificial network, UNet, deep learning**

1. Introduction

2. Background and Related Work

Semantic segmentation is a process of matching each pixel in an image to a class label. It is used in cases where a thorough understanding of images is needed such as diagnosing medical conditions, navigation for self-driving cars, photo and video editing and navigation for robots.

To know how deep learning is used to tackle semantic segmentation we need to understand that its not just an isolated field of machine learning but a step in a larger process of object detection. It helps with the progression from a normal image to detecting just not what an object is but how many objects there are in the image. The natural progression is image classification, object localisation, semantic segmentation and instance segmentation.

Early work of aerial image labeling focused on ad-hoc and knowledge-based approaches [1] but this research will focus on a machine learning based approach, specifically a deep learning approach. Machine learning has led to more recent progression when interpreting aerial images and also in general computer vision problems such as labelling in images using semantic segmentation.

3. Research Methodology

3.1. Research design

The aim of this research was to improve on early methods of aerial image labelling by building a machine learning model that is able to perform semantic segmentation on aerial images of buildings. This is a qualitative research and we needed to acquire sufficient data to train a deep learning model. The dataset needed to have aerial images of buildings that would have masks as the ground truth we would use in training the model.
We considered a Unet, Fully Convolutional Network and Deeplab as architectures for the model.

3.1.1. Source Dataset.

We used the Massachusetts buildings dataset from Kaggle[1]. The dataset consists of 151 aerial images of the Boston area. The images cover an area of 2.25 square kilometres, so the entire dataset covers roughly 340 square kilometres. The images are of high quality with dimensions of 1500 x 1500 pixels. The data is already split into a training set of 137 images, a testing set of 10 images and a validation set of 4 images. Each image has a corresponding mask that are used as targets. These target maps were obtained by rasterising building footprints obtained from the OpenStreetMap project.The dataset covers urban and suburban areas with buildings of all sizes.

3.1.2. Data Pre-processing.

To reduce computation time and memory usage we downsampled the 1500x1500x3 pixel images together with the corresponding 1500x1500x1 pixel masks to 512x512x3 and 512x512x1 pixels respectively. The images and their corresponding masks pixels' where also normalised from the range [0,256] to [0,1].

3.2. Dataset split and data augmentation

We reduced the training set to 131 images and transferred them to the validation set. This gave us a total of 10 images for the validation set. This was done to help reduce the training time and the amount of memory needed to store the images.
To help increasing the training data we performed data augmentation of the images by flipping the images and their corresponding masks horizontally and vertically.

3.3. Training the Model

Due to insufficient memory on our local machines we used Google Colaboratory which gives us 16Gb of memory to train our model.
The UNet architecture was used as the model for prediction. A Unet modifies and extends a Fully Convolutional Network(FCN). It is a U-shaped architecture that uses a

[1] https://www.kaggle.com/balraj98/massachusetts-buildings-dataset

encoder-decoder scheme. It consists of three sections a downsampling, bottleneck and upsampling section.

To train we used a batch size of 6 and trained the model for 100 epochs. We used a learning rate of 0.001 and a dropout rate of 0.5. We used the Adam optimiser as the optimising algorithm. It is a extension of stochastic gradient descent and has seen a rise in popularity for deep learning applications in computer vision. It combines the advantages of two other extensions of stochastic gradient descent, specifically adaptive gradient algorithm and root mean square propagation. For the loss function we used a binary cross entropy.

4. Results and Discussion

4.1. Evaluation Metrics

To ensure that our model was performing correctly we need ways to monitor its prediction performance. The evaluation metrics we used to monitor the performance of the model were training and validation loss, accuracy and intersection over union.

Training and Validation loss.

To ensure that our model does not underfit or overfit on the training data we had to monitor the training and validation loss of the model. Underfitting occurs when the training loss is greater than the validation loss while overfitting occurs when the training loss is less than the validation loss this means the model is fitting to the training data and is not able to generalise. The aim is to have both losses equally low and to make sure they both converge over time.

Accuracy.

Accuracy is calculated as the number of true positive predictions and true negative predictions divided by the total number of data points we used for prediction. It gives us a basic indication of what fraction out of all predictions did we get right. Accuracy has its drawbacks because a model is trained using a biased dataset and evaluated using accuracy it can give a false sense of high accuracy.

Intersection over union.

Intersection over union is popular metric used in evaluation of image segmentation models. It is used to quantify the percent overlap between the mask images, which is the target, and the prediction output. It measures the number of common pixels the target and prediction masks divided by the total number of pixels present across both masks.

4.2. Experimental Results

4.3. Discussion

5. Conclusions and Future Work

To extend this research we could look at employing different architectures for comparison. We could also use data acquired from different areas so that the model is able to generalise.

References

[1] V. Mnih, “Machine learning for aerial image labeling,” Ph.D. dissertation, University of Toronto, 2013.