Foundations of Deep Learning Project Report

Hurricane Harvey Challenge [DSBA 2022-2023]

ESSEC & CentraleSupélec

Darius CHUA
darius.chua@student-cs.fr

Xinran YAO
xinran.yao@student-cs.fr

I. Introduction

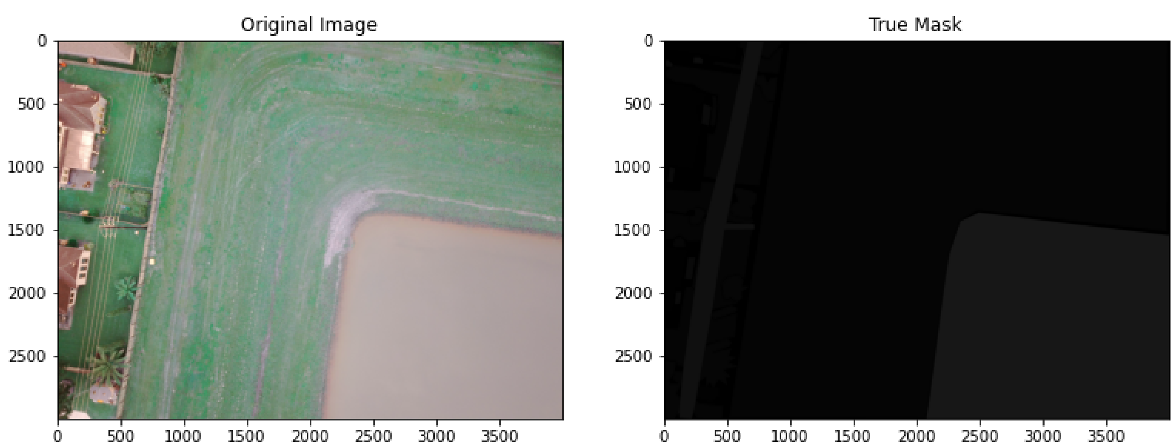
Hurricanes are one of the most common and dangerous natural disasters in the US, causing significant damage and loss of life on a frequent basis. Image segmentation can play a vital role in predicting and monitoring such events. The goal of this project is to develop image segmentation models for aerial post-flood imagery after a major flood event.

In the provided **Hurricane Harvey** dataset, 374 high resolution images were provided, of which 299 of them had dense image masks that contained data on 26 targeted segments such as property roofs, swimming pools, and street lights. In addition, a synthetic dataset with images created from the mask and image data from the original dataset was also provided. However, it was found early on that integrating this data into the model training process tended to result in worse performance. It was thus excluded from subsequent models.

We implemented several deep learning techniques to solve the task and the U-net model stands out with a public score on Geoengine of 0.7010.

II. Methodology

1. Data Transformation



Original image (left) and the mask (right)

There are 374 UVA images in Hurricane Harvey dataset, with different sizes (4000*3000, 4592*3072). We first loaded and split the data into train datasets for images and masks, as well as a test dataset for the images. The train images and masks datasets contain 299 in each, while the test dataset contained 75 images. The train data was then subdivided into an actual training set and a validation set in a 4:1 ratio. The above figure displays an example of the original image and its training mask (*for the classes mentioned below*).

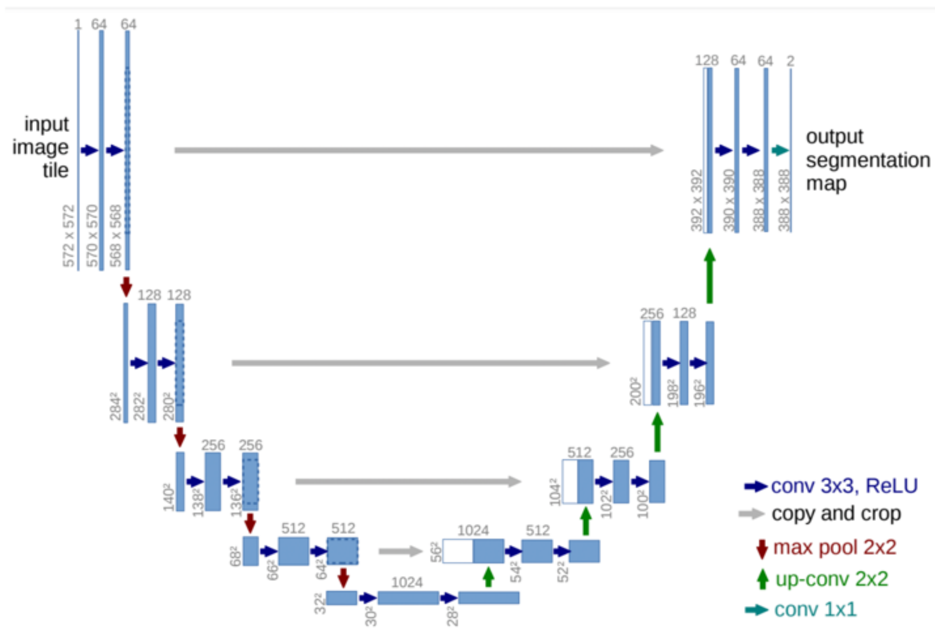
Count	374 raster masks 22,419 annotation vectors (generated from masks)
Classes	Property Roof, Secondary Structure, Swimming Pool, Vehicle, Grass, Trees / Shrubs, Solar Panels, Chimney, Street Light, Window, Satellite Antenna, Garbage Bins, Trampoline, Road / Highway, Under Construction / In Progress Status, Power Lines & Cables, Bridge, Water Tank / Oil Tank, Parking Area - Commercial, Sports Complex / Arena, Industrial Site, Dense Vegetation / Forest, Water Body, Flooded, Boat, Parking Area

In order to provide more data for the training model, we applied data augmentation techniques such as rotating and flipping the image. This would help improve the model as the model would be able to train on images of objects from different perspectives. Rotating was done for each 90 degree angle, while the flipping was done horizontally and vertically. This increased the size of the dataset by a factor of 6. However, only a portion of these images were used due to limitations in computing power. 1000 images were randomly selected, with 800 going into training and 200 into the validation set.

The images were also resized into squares. Due to the high resolution of the images provided, fairly large dimensions could have been used in theory. Unfortunately, due to limitations in computing power, images of size 512x512 were usually used. Several experiments were made with 256x256 images in order to free up resources to include more images from the augmented set, but it was found that the loss in resolution affected performance much more severely than the improvement of an increased dataset provided, resulting in a worse overall score. A small handful of experiments were done with 1024x1024 images were run, of which the best performing model is one of, but due to the long training time, there was not much room for a gridsearch of hyperparameters or optimisation of the size of the training set. 2048x2048 images were too computationally expensive to run at all.

Then the augmented dataset was converted to tensors and normalized with the mean and standard deviations that were calculated in the preparation phase. (*mean*=[0.38980951, 0.46497215, 0.43926388], *std* = [0.21997646, 0.19580692, 0.21212297])

2. Model



U-Net architecture from the following research paper

source: https://towardsdatascience.com/unet-line-by-line-explanation-9b191c76baf5

U-Net is a convolutional neural network architecture that was first introduced in 2015 by Olaf Ronneberger, Philipp Fischer, and Thomas Brox. One key feature of this architecture is its ability to handle small and irregularly shaped objects in images, making it well suited for image segmentation tasks, and proved also the best performance in our case.

The other techniques we have implemented will be summarized here. ResNet's performance was close but still not as good as Unet. ResNet50 is a Microsoft developed architecture. One is ResNet50 which is widely used for object detection and image recognition and popular for transfer learning.

DeepLabV3 by Google was also utilized in the project and was shown to be the second best performer. Running the model using the pre-trained weights was shown to have better performance than training the model from scratch, which is an expected result.

3. Training

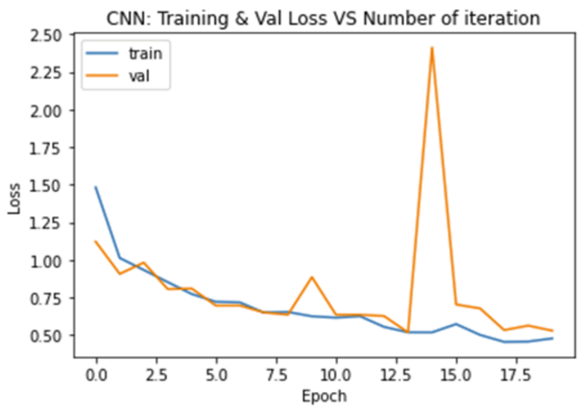
The optimiser was Adams while the loss function utilized the cross entropy loss, as it is a classification task. Through a gridsearch using low resolution images, a learning rate (lr) of 0.001 was found as ideal. Each model was run through 20-25 epochs.

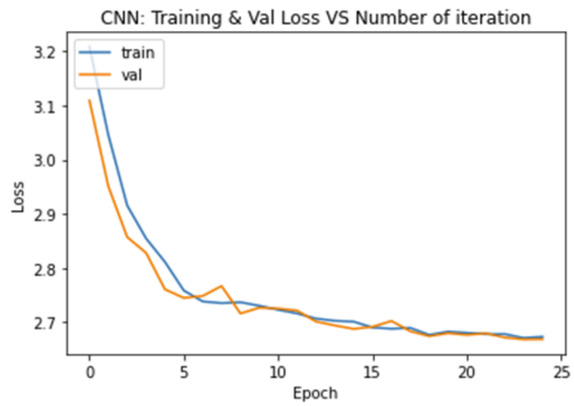
4. Prediction

The prediction masks are obtained by putting the test images which had undergone the same preprocessing as the training sets into the trained model. Using the OpenCV library, the prediction masks are then resized to their original size. The interpolation parameter for the upsizing was found to have a significant effect on the test results: cv2.INTER_NEAREST was the best performing one, by an order of magnitude as compared to some others.

III. Evaluation

The models were evaluated using the loss function. The following plots show the change in loss for each epoch the model was trained with, where the left plot is from training the DeepLabV3 model, while the right comes from training the U-Net model. In general it was shown that the U-Net model performed better than the DeepLabV3 model.





IV. Discussion

The 0.7010 result obtained from the U-Net model trained on 1024x1024 images showed that we have a model that is better than the baseline model (0.6812, code not provided) and is also fairly good at identifying objects in aerial images of areas devastated by hurricanes.

Unsurprisingly, training the model on augmented data, larger batch sizes, as well as on higher resolution images improved the performance. However, as all 3 increase the computational resources required, some optimization of these elements are required. Unfortunately, there was insufficient time to explore this avenue. In this project, it was also assumed that the optimal hyperparameters for training the model on low resolution images were the same as high resolution images. This may not necessarily be the case, and is another interesting avenue of research.

V. References

1. U-Nets: *https://towardsdatascience.com/unet-line-by-line-explanation-9b191c76baf5*
2. DeepLabV3: *https://towardsdatascience.com/review-deeplabv3-atrous-convolution-semantic-segmentation-6d818bfd1d74*