2021 Better Working World Data Challenge 1 Supporting paper

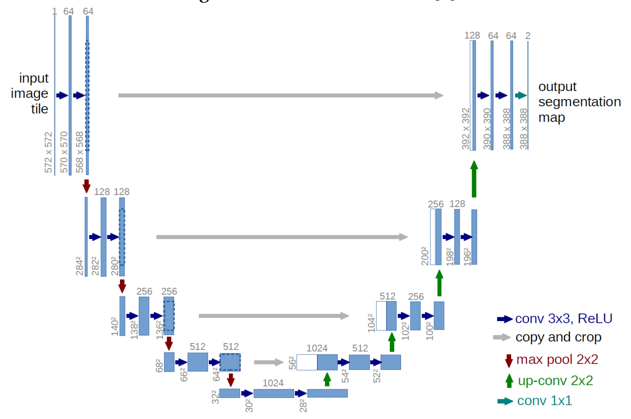
Dominik Benk
FSV IES UK
Czech Republic
19844055@fsv.cuni.cz

***Abstract*— In this paper, I will describe an approach of automatically generating features from line scan images with extensive use of augmentation. The main innovation lies in the application of well known architecture on very unique dataset, that could greatly reduce the amount of scans required to analyze by human and speed up the process of bushfire mapping.**

I. Defining the Model

First of all, I have applied a baseline techniques such as thresholding (Multi-Otsu [1]) and K-means clustering [2], but the results did not seem very promising. Therefore, I looked for alternatives from field of machine learning, but quickly realized, that there is no possibility of feeding the whole scan images to an artificial network, as memory required to generate the features would not be sufficient. Nevertheless, I found so called U-net architecture that works very well for segmentation and can be trained on small windows generated from original images. It was first described in Biomedical Image Segmentation in 2015 [3].

Fig. 1. U-net architecture [3]



On figure 1 you can see the contracting path, generating features describing the context, followed by the expansive path, that up-samples these features and localize them for segmentation. It is fully convolutional network, as you can feed it with images of different sizes, and the number of layers and filters is not fixed and directly affects the capacity of the model. This is however artificial and was subject to hyper-parameter search.

I have also tested few other U-net extensions such as Attention U-net or R2-U-net, but I did not find them superior for this task. Note that all models were implemented with a use of TensorFlow. [1]

A. Feature Engineering

Great advantage of deep learning models is, that if defined properly, they generate all the features automatically. I have also attempted to add additional image channels (sets of features), that would be generated from other data, namely Sentinel 1 and 2, but unfortunately, I was not able to connect them with the scans.

Determining the importance of the features is challenging in case of convolutional networks, as the features generated from the infrared magnitude of each pixel are quite abstract. However, if we added additional image channels, we could compare whether there is a significant effect on predictions.

II. Data Preprocessing

I started with examining the data to better understand the task and matched the polygon-linescan pairs. Apart from the direct matches, I managed to connect 38 additional polygons via composite labels and comments and decided to select only a subset consisting of the most reasonable scans, that had no corrupted polygons as well as meaningful fire edges.

The hardest part of the challenge was then to determine how to generate the train data, as the scan resources are quite limited and there is a high risk of over-fitting. To avoid this, I created so called patches, which provided me with a lot of unique data, while also being able to fit in the GPU memory. I iterated through the whole image and to make it easier for the model to learn, I kept only the

[1] https://www.tensorflow.org/

interesting ones, whose masks had to have at least 1% of fire pixels, but no more than 90, and the image patches must have had at least 75% of non zero pixels.

III. Training

A. Random Data Augmentation

To avoid over-fitting and to improve generalization of the model, the patches needed to be randomly augmented. I experimented with techniques such as random shifting, zooming, shearing, rotating, flipping or cut outs, to provide as much unique patches as possible in each epoch. It is also important to note, that the missing pixels after augmentation were always filled with scan's reflections. This procedure might be considered quite invasive, but it turned out to work very well during the training.

B. Optimization

To converge to optimal weights, I considered different type of loss functions, such as cross entropy, dice, focal, log cosh dice or their combinations. The optimization was done only on masks' centers, as I believe there is not enough information from input scan to predict the boundaries.

It turned out that the combination of cross entropy and dice converged the fastest, both in terms of intersection over union and F1 score. Determining the rest of the parameters was a little puzzling, but Adam optimization with cosine decay, batch normalization and He normal weight initialization all together significantly improved the convergence.

IV. Validation

As training a convolutional network is very time consuming, I opted to keep 20% of the generated patches for evaluation, even though cross-validation would provide more accurate estimates of performance. I also made sure, that the patches generated from the same scan do not belong to both sets.

In experiments, I altered various parameters, such as magnitude of augmentation, strides and sizes during the patches generation, or the number of filters, layers, dropouts, batch normalization in the U-net. The plot of their performance in terms of intersection of union is available in the provided code.

V. Predictions

During the predictions, I considered different setups of patch sizes as well as strides, and similarly as during the optimization, I predicted only the mask centers. To make the results more robust, each predicted patch was based on an average of four 90 degree rotations. After iterating through the whole image, I set the threshold for fire classification to 0.5, which finalized the process.

A. Run-time

The inference time is directly affected by the patch size as well as model size, where parameters such as number of layers and filters play the most important role. It took around 17 seconds for the final model to predict all 5 test images on Tesla T4, including also the four 90 degree rotations.

Nevertheless, there is still a great room for improvement, as there is no need to estimate these patches in sequence and one could also predict them in parallel, or even utilize multiple devices at the same time.

VI. Discussion and Real-world Applications

The main advantages of proposed model are, that the inference is quick and can be applied on scans of various sizes and resolutions, while requiring only a relatively small number of train scans. This allows to efficiently predict even much bigger linescans than those provided for the challenge.

Also, as a relatively small subset of scans was used during training, there is still a great potential to improve with new scans and more importantly a possibility to add additional image channels, such that would make it easier for the network to distinguish between rivers, clouds, rocks, etc.

The main limitations are the false negative cases (clouds for example), which could have horrible consequences if the process was fully automated. However, the model could be altered to focus on their minimization and then used with a human assistance, only correcting the mistakes made by model in the subset of scans, that were detected with a bushfire.

There is also a potential to apply the model for fire mapping on other continents than Australia, however this would require further investigation. Regions with similar climate conditions might be appropriate for it, but since the Australian vegetation is so unique, I am afraid that the model would have to be retrained with different set of scans.

References

[1] Huang, D., Lin, T., Hu, W. (2011). Automatic Multilevel Thresholding Based on Two-stage Otsu's Method with Cluster Determination by Valley Estimation. Available: `http://www.ijicic.org/ijicic-10-05033.pdf`.

[2] Nameirakpam D., Khumanthem M., Yambem J. (2015).Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm. Available: `https://www.sciencedirect.com/science/article/pii/S1877050915014143`.

[3] Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Available: `https://arxiv.org/abs/1505.04597`.