SEGNET ON DRONE IMAGES: IMAGE SEGMENTATION FOR SMART AGRICULTURE

Anders Henriksen, Asger Schultz, Oskar Wiese, Mads Andersen, Søren Winkel Holm

s183904, s183912, s183917, s173934, s183911

ABSTRACT

The abstract should appear at the top of the left-hand column of text, about 0.5 inch (12 mm) below the title area and no more than 3.125 inches (80 mm) in length. Leave a 0.5 inch (12 mm) space between the end of the abstract and the beginning of the main text. The abstract should contain about 100 to 150 words, and should be identical to the abstract text submitted electronically along with the paper cover sheet. All manuscripts must be in English, printed in black ink.

***Index Terms*—** One, two, three, four, five

1. INTRODUCTION

1.1. Dataset and preprocessing

Our dataset consists of two large, high resolution orthomosaic RGB images of a sugar cane field[1]. The first image consists of several drone images stitched together, with the second image of the same size being the by an expert biologist manually labelled human ground truth. The three classes each have a corresponding colour – crop rows are green, weeds are yellow, and soil is red. Void pixels are black.

2. METHODS

2.1. Preprocessing and data augmentation

In order to get the most out of the data, preprocessing is needed. First, the RGB values of the non-void pixels of the aerial image are standardized, and a matrix is used to represent the ground truth. Each entry corresponds to a pixel and contains a number 0-2 for the different classes or 3 for void. The images are then padded with black pixels and cropped into smaller 512×512 pixel images. Any of these images containing only black pixels are discarded, leaving a total of 108 pairs of aerial photo/ground truth images. These are then split into 69 training, 18 validation, and 25 test images.

In order to increase the effective size of the dataset, we perform aggressive data augmentation. When training the network, each pair of aerial photo/ground truth images is randomly cropped into 256×256 pixel images. Furthermore, we applied a 50 $\%$ chance of performing a top/down flip as well as a 50 $\%$ chance of left/right flip to each image pair. Even though data augmentation is not as a good as more independent data, it still allows the network to generalize better and overfit less.

2.2. Regularization

Because deep neural networks are so flexible models, regularization is necessary on top of the data augmentation to further reduce overfitting. This is done in two ways.

Dropout at 10 $\%$ is used after each blue block in the network (see Fig. **??**). This randomly shuts off nodes during training leading to node redundancy and variability, as the same input will vary somewhat in its output, which learns the network to generalize better. We experimented with higher dropout, but found that too much would significantly reduce learning.

Batch normalization is applied after each dropout. This standardizes the activations, keeping them close to zero. As a result, the weights and biases also stay close to zero, which reduces the flexibility of the model, leading to less overfitting. Batch normalization also has several other benefits, such as reducing the vanishing gradient problem and allowing for a higher learning rate and thus faster convergence time. [1]

2.3.

The encoder part of the network creates a rich feature map representing the image content. The more layers of maxpooling there are the more translation invariance for robust classification can be achieved. The boundary detail is very important when dealing with image segmentation. Hence, capturing boundary information in the feature maps of the encoder before upsampling is important. This can simply be done by storing the whole feature map, but due to memory constrains only the maxpooling indices are saved, which is a good approximation of the feature maps.

[1]Aerial image: http://www.lapix.ufsc.br/wp-content/uploads/2019/05/sugarcane2.png
Ground truth: http://www.lapix.ufsc.br/wp-content/uploads/2019/05/crop6GT.png

2.4. Loss function: Quality over quantity

Multi-class cross entropy because:

- Softmax Network: Minus log likelihood
- Can be seen as a classic multiclass classifier – just on a pixel-by-pixel basis.

Weighted cross entropy because:

- Unbalanced class distribution: Network has to learn to focus on important pixels: Don't classify everything as dirt.
- Initial tests made the network behave as the baseline: Simple features in early layers got were not penalized enough and learning was not stable.
- Resampling expensive

2.5. Metrics

Had to use different metrics because

- Not agreement in Image Segmentation papers.
- Want to get accuracy on a global scale and on a class scale.
- Different metrics important in different fields.

The metrics [2][3]

- Global accuracy: Trivial and not very important because of class imbalance but is good for smoothness
- Mean class-wise accuracy: Takes class imbalance into account. Is what is being optimized for in the model.
- Mean Intersect over Union: "Jaccard Index". Found to be better correlated with human classification though still only ≈ 0.5. Favours region smoothness highly and not boundary accuracy.
- Harmonic mean of precision and recall. To compare to others with same project. Penalizes false positives and gives less credit to true negatives thus being better for unbalanced classes.

2.6. Regularization and Hyperparameters

Regularization

- NN's are prone to overfitting, because they are so flexible
- Prevent overfitting $\rightarrow$ better results on test data
- Three methods
- Dropout: Randomly remove nodes to increase variability. $p = 10\%$
- Data augmentation: Increase size of dataset
  - Crop each 512×512 to random 256×256
  - 50% chance of flip T/D and 50% chance of flip L/R
- Batch normalization normalizes activations
  - Faster convergence
  - Prevents ReLU from not learning
  - Introduces noise
  - Reduces vanishing/exploding gradient problem, as values stay close to 0

Hyperparameters

- Adaptive learning rate from ADAM optimizer, initialized at $2 \cdot 10^{-4}$
- Total: 26 conv + batchnorm + ReLU with dropout, 5 pool/upsample, 1 softmax
- 14.7 M parameters in encoder – significantly lower than 134 M in VGG16 because of no fully-connected layers
- Kernel size: 3×3, stride 1, maxpool: 2×2, stride 2
- Corresponding padding of 1 to prevent reduction of image size

2.7. Unification of cropped image predictions

In a real-world application of the field classification a farmer would want a complete and precise segmentation of his whole field at once, such that fertilizer and pesticides can be distributed accordingly. To accomplish this, a reconstruction of the smaller image inferences is neccesary. The most straight forward method of combining the smaller images, by simply lining them up next to each other results in a very rough transition between the smaller inferences. This can be seen in the left side of 1. The blocky nature of the field prediction is caused by a lack of information from neighbouring pixel when inference is performed near the borders of an image. To solve this problem, we have chosen to increase the size of the cropped images and add some overlap, and then infer on these enlarged pictured. In the procedure of joining the enlarged cropped pictures they are cropped again, to avoid the border areas. At the cost of computational efficiency more information is available near the borders and a smooth field prediction can be archieved. This can be seen in the right side of 1. A visualization of the reconstruction technique can be seen in appendix 2.

[2] https://hal.inria.fr/hal-01581525/document
[3] http://www.bmva.org/bmvc/2013/Papers/paper0032/paper0032.pdf

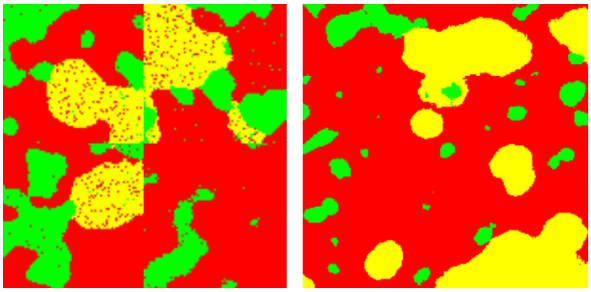

Fig. 1. **Left:** Smaller inferences put next to eachother without use of reconstruction techniques. **Right:** Reconstruction with padding and overlap between smaller inferences.

3. RESULTS

4. DISCUSSION

4.1. Comparison of different image segmentation neural networks

- Several competing network structures with high performance in image segmentation. U-net, FCN, DeepLabv1, DeconvNet
- Purpose of SegNet, efficient
- 3 out of the 4 mentioned uses the encoder from the famous VGG16 paper, but differ in decoder.
- FCN, No decoder -¿ Blocky segmentation, but very efficient in inference time.
- DeconvNet, Deconvolution and fully connected layers.
- U-Net, (different purpose), skip connections.
- Main takeaway
- (Deeplabv-LargeFOV & FCN)

[2]

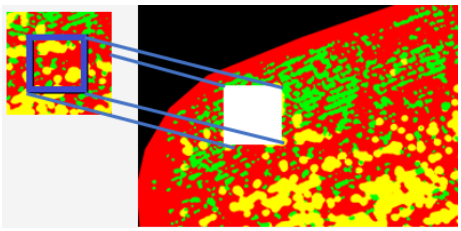

Fig. 2. Reconstruction of the smaller inferences into a unified field prediction.

4.2. Extension of network

4.3. Conclusion

5. REFERENCES

[1] Collis Jaron, "Glossary of deep learning: Batch normalization," .

[2] Alex et al. Kendall, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," .

5.1. Appendix