Deep Learning

Rho: Mads Christian Berggrein Andersen

Raport: Prøv netværk på Eskil data, Cross Entropy without loss. Instance vs object

## Data (Kort GT, Label, Class og behandling af data)

**Brief overview of our data:**

Our data consist of one large drone photo of a sugar cane field, which has been geometrically corrected. The field has been manually labeled by an expert biologist to create a human-ground truth, these labels are represented as a GT-matrix with 3 possible classes. The possible class are crop row (green), weed (yellow), background (red). The large photo is cropped into smaller images and afterwards augmentation techniques are used to gain more data. We will return to the augmentation part later. The images with exclusively black pixels are removed from our data set, and images with some black pixels are ignored when calculating the loss function

(**Orthomosaic:** It has been geometrically corrected such that the distortion due to curving of earth itself, lens distortion and camera tilt is accounted for. This means that distance measures from the photo easily translated back to precise GPS-coordinates.)

[Peter.jensen@Cellari.io](mailto:Peter.jensen@Cellari.io)

Our data consist of one large drone image of a sugarcane field. The field has been manually labeled into three categories by expert biologist to create a human ground truth. The three classes are crops, weed and soil. And we want our network to be able to predict these classes. For practical reason the image has been divided into smaller images, and with augmented versions these smaller images represent our dataset.

As mentioned earlier, the field is divided into smaller images when inference is performed. If these smaller images are unified a lot of edges will be visible in the prediction due to a lesser quality of predictions near image borders. We have solved this problem by increasing the cropping size with some overlap and then by running inference on the larger images and we can remove border areas and get a smooth unified prediction.

**The strength of CNNs in image classification and segmentation comes from it’s ability to take spatial information into account when making predictions. This means that neighboring pixels are used during inference. But this also means that border predictions are of less quality. Our solution to this problem is divide the field into a size larger images than we actually would want and with some overlap. This means that after running inference we can cut away areas close to borders. When dealing with the borders of the field, we are using mirroring techniques to increase predictions near the borders. The result of this (these) technique are that we get a smooth unified field prediction, whereas our preliminary results showed that without these techniques the field predictions is a little blocky.**

**Before running inference on a field, the aerial image are for practical reason chopped into smaller images, and then inference on smaller images would be performed. After inference the smaller images can be stitched back together again for a unified prediction. However, this division into smaller images, means that the CNN does not take advantage of its ability to take neighboring pixels into account when making predictions near boarders, hence border-predictions are of less quality. One way to solve the problem is to pad the images with values drawn from the larger image. In practice, one would chop larger images from the original with some overlap, and then crop the borders away. The result is a whole field prediction that is smoother.**

How does the SegNet compare with the other leading architectures in the field of image segmentation. I want to emphasize that the purpose of SegNet is to be efficient in inference time and memory wise, such that it can be used in embedded systems. For example in drones or in self-driving cars. Besides SegNet the most acknowledged structures are named U-net, DeepLabv1, DeconvNet and FCN. They have very similar performance and the encoding architecture is almost identical. But they differ in decoding structure and techniques and as a common denominator the SegNet alternatives uses a lot of memory during inference. Among the heavy users of memory are the fully connected layers of the DeconvNet and FCN, and the skip connections in the U-Net. And in comparison, with the other networks, SegNet uses half the memory during inference and comes in second with regard to inference time. Which means that SegNet proves to be a promising architecture for embedded systems.

**Before we move on to a comparison of the leading network architectures in the field of image segmentation, It is important to emphasize that the purpose of SegNet is to be efficient in inference time and memory wise, such that it can be used in embedded systems. For example in drones or in self-driving cars. Besides SegNet the most acknowledged structures are named U-net, DeepLabv1, DeconvNet and FCN. They have very similar performance and the encoding architecture is almost identical. But they differ in decoding structure and techniques and as a common denominator the SegNet alternatives uses a lot of memory during inference. Among the heavy users of memory are the fully connected layers of the DeconvNet and FCN, and the skip connections in the U-Net. And in comparison, with the other networks, SegNet uses half the memory during inference and comes in second with regard to inference time. Which means that SegNet proves to be a promising architecture for embedded systems.**

**Moving on to a small comparison of different leading network architectures within the field of image segmentation. Besides SegNet the most acknowledged structures are named U-net, DeepLabv1, DeconvNet and FCN. They have very similar performance and the encoding architecture is almost identical. But they differ in decoding structure and techniques and as a common denominator the SegNet alternatives uses a lot of memory during inference. Among the heavy users of memory are the fully connected layers of the DeconvNet and FCN, and the skip connections in the U-Net. And in comparison, with the other networks, SegNet uses half the memory during inference and comes in second with regard to inference time. The fastest network is the FCN, and this is becaused it does not have a decoder structure, but makes a prediction directly from the output from the encoder, and this leads to blocky predictions.**

VGG, U-Net,

Upsampling FCN

\_ 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) content...

## Resultater, sammenligning af netværk (Andre segmentation løsninger. Reconstruction.

In a real-world application of the segmentation a farmer would want a complete and precise segmentation of his whole field at once, such that fertilization and pesticides can be distributed accordingly. However, it turns out that the prediction quality is of less quality near the boarders of the image. Therefore, a naïve stitching of the cropped images leads to a full-blown image prediction with obvious flaws near the boarders between the cropped images. To solve this problem, we have chosen to increase the size of the cropped images, and infer on these enlarged pictured. In the procedure of joining the enlarged cropped pictures the pictures are cropped again, to avoid the near border areas. This is computationally inefficient, but it works, and since the inference time is not that big, it is an alright solution. For industrial purposes, another approach might be beneficial.

An encountered issue is that when the segmentation has been performed on the cropped images, the prediction near the borders are of less quality. (WHY?) Which is why a full-scale image prediction is not optimally performed with a direct stitching of the smaller images.

VGG16, U-Net, SegNet

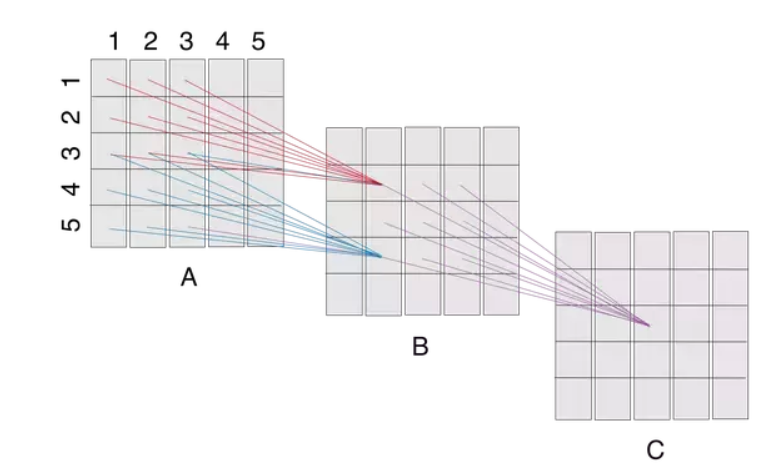
#Skip connections

#Augmentation -> Hvorfor og hvilke metoder bruges?

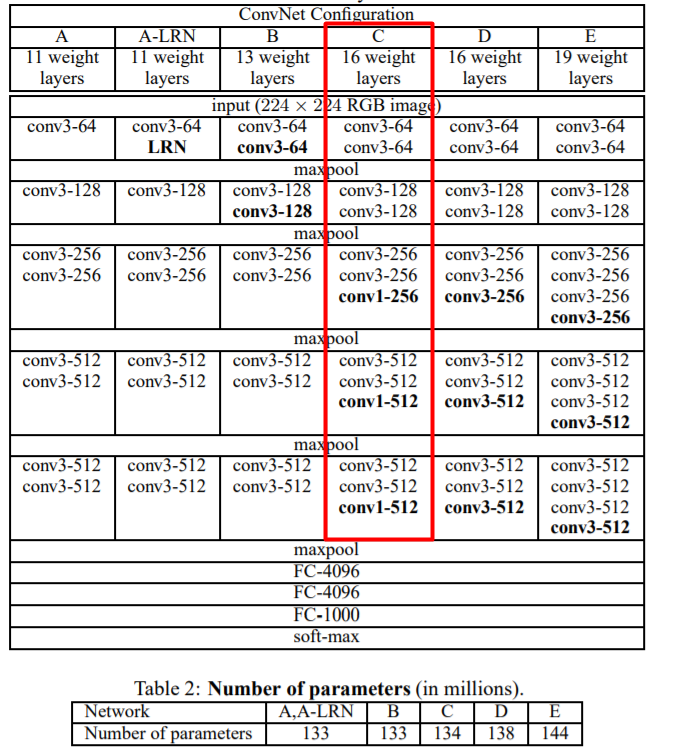
#Formler.

#Netværk arkitektur.

<https://medium.com/@abhigoku10/topic-dl03-receptive-field-in-cnn-and-the-math-behind-it-e17565212a20>



VGG16:



On a system equipped with four NVIDIA Titan Black GPUs, training a single net took 2–3 weeks depending on the architecture.

**Our data consist of one large drone image of a sugarcane field. The field has been manually labeled into three categories by expert biologist to create a human ground truth. The three classes are crops, weed and soil. And we want our network to be able to predict these classes. For practical reason the image has been divided into smaller images, and with augmented versions these smaller images represent our dataset.**

**Before running inference on a field, the aerial image are for practical reason chopped into smaller images, and then inference on smaller images would be performed. After inference the smaller images can be stitched back together again for a unified prediction. However, this division into smaller images, means that the CNN does not take advantage of its ability to take neighboring pixels into account when making predictions near boarders, hence border-predictions are of less quality. One way to solve the problem is to pad the images with values drawn from the larger image. In practice, one would chop larger images from the original with some overlap, and then crop the borders away. The result is a whole field prediction that is smoother.**

**Before we move on to giving a brief comparison of leading network architectures in the field of image segmentation, I want to emphasize that the purpose of SegNet is to be efficient in inference time and memory wise, such that it can be used in embedded systems. For example in drones or in self-driving cars. – Mere om droner --**

**Moving on to a small comparison of different leading network architectures within the field of image segmentation. Besides SegNet the most acknowledged structures are named U-net, DeepLabv1, DeconvNet and FCN. They have very similar performance and the encoding architecture is almost identical. But they differ in decoding structure and techniques and as a common denominator the SegNet alternatives uses a lot of memory during inference. Among the heavy users of memory are the fully connected layers of the DeconvNet and FCN, and the skip connections in the U-Net. And in comparison, with the other networks, SegNet uses half the memory during inference and comes in second with regard to inference time. The fastest network is the FCN, and this is becaused it does not have a decoder structure, but makes a prediction directly from the output from the encoder, and this leads to blocky predictions.**

Our data consist of one large drone image of a sugarcane field. The field has been manually labeled into three categories by expert biologist to create a human ground truth. The three classes are crops, weed and soil. And we want our network to be able to predict these classes. For practical reason the image has been divided into smaller images, and with augmented versions these smaller images represent our dataset.

As mentioned earlier, the field is divided into smaller images when inference is performed and afterwards unified to get a whole field prediction. If this unification is done naively there is going to be clearly visible edges where the smaller images are stitched together. This is due to a CNNs inability make precise predictions near the borders as a result of less information available. To get around the problem, we have increased the cropping size with some overlap and then by running inference on the larger images and we can remove border areas and get a smooth unified prediction.

Before I give a brief comparison of SegNet and its competitors within the field of image segmentation, I want to emphasize that the purpose of SegNet is to be efficient in inference time and memory wise, such that it can be used in embedded systems. For example in drones or in self-driving cars. Besides SegNet the most acknowledged structures are named U-net, DeconvNet and FCN. They have very similar performance and the encoding architecture is almost identical. But they differ in decoding structure and techniques and as a common denominator the SegNet alternatives uses a lot of memory during inference. A principle that is used in the three alternatives is that feature maps extracted from multiple layers are used to provide both local and global information. And in comparison, with the other networks, SegNet uses half the memory during inference and comes in second with regard to inference time. Which means that SegNet seems to be a promising architecture for embedded systems.

\subsubsection{Leading architectures in the image segmentation field}

In the field of image segmentation there are several different neural network architectures that excel in different types of problems. Among the most acknowledged architectures are U-net, DeepLabv1, DeconvNet and FCN. All of these networks differ in several ways and it is important to emphasize that each network is designed to excel in a specific branch of image segmentation. U-net is as an example designed for medical image segmentation, whereas SegNet is designed such that it can be used in embedded systems. Each branch of problems has its own challenges and limitations. Therefore it is necessary to apply problem-specific methods to optimally segment the images. As mentioned earlier the purpose of SegNet is to be efficient in inference time and memory-wise, such that it can be used in embedded systems. And these two aspects are the ones we are interested in comparing SegNet and the other architectures.

SegNet and its competitors have a quite similar performance in terms of accuracy on both road map scenes and the benchmark dataset known as SUN RGB-D, which is an indoor scene segmentation challenge \cite{seg}.

With respect to the architecture of the encoder of the competing networks, SegNet and three of the four competitors (FCN, DeconvNet, U-net) have an almost topologically identical encoder as the one from the well known VGG16 network. One important difference between SegNet and the three others is that SegNet is without the final fully connected layers, which decreases the amount of trainable parameters with almost 90\% \cite{seg}.

All of the networks differ in decoding structure and techniques that is applied, and as a common denominator for FCN, DeconvNet and DeepLab-LargeFov are nearly twice as much memory during inference used. Among the heavy users of memory are the fully connected layers of the DeconvNet and FCN, and the skip connections in the U-Net. In comparrison with the same 3 structures does SegNet come in second only bested by DeepLab-LargeFOV with respect to inference time.

The encoder part of the network has only 14.7 million parameters compared to partially fully connected structures that tend to have well over 100 million. \cite{seg}