R&D Project Proposal

Semantic Segmentation using Resource Efficient Deep Learning

Naresh Kumar Gurulingan
Matrikel Nr.: 9030384

B-IT Master Studies Autonomous Systems

University of Applied Sciences Bonn-Rhein-Sieg
Advisor: Prof. Dr. Paul G. Plöger
Second Advisor: M. Sc. Deebul Nair

November 23, 2017

Abstract

Semantic Segmentation is a computer vision task in which an image is split into meaningful regions such that each region represents a different object or background. In the context of deep learning, semantic segmentation is treated as a pixelwise classification problem where a neural network is trained to assign a class to every pixel in the input image. Existing state-of-the-art methods repurpose image classification pretrained Convolutional Neural Network architectures for semantic segmentation. This enables the transfer of learned representations from CNNs which show excellent performance in image classification.

Therefore, the underlying CNN architecture is often a deep image classification model with large number of free parameters and slow inference times. This makes the semantic segmentation networks unsuitable for mobile or embedded devices because of the high demand on memory and computational power. Several approaches have been proposed in the literature to reduce the resource requirements of CNN architectures. This work aims to analyze the feasibility of using such approaches in current state-of-the-art semantic segmentation networks without considerably affecting the accuracy.

background added.

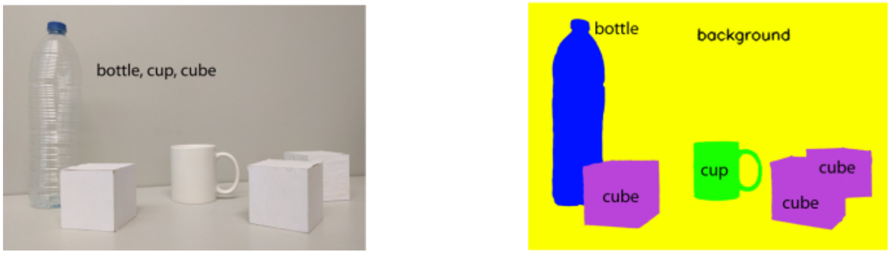
energy replaced with computational power.

Contents

1 Introduction

In recent years, deep learning has significantly impacted research in the field of computer vision. Variations of Convolutional Neural Network architectures have shown state-of-the-art performance in computer vision tasks such as image classification [1], object detection [2], action recognition [3] and semantic segmentation [4]. A considerable part of this success comes from the supervised learning paradigm through which the networks are trained with labeled samples.

State-of-the-art deep learning techniques in semantic segmentation also make use of the supervised learning paradigm. Semantic segmentation is treated as a pixelwise classification problem with the goal of assigning a class from a list of desired classes to every pixel in an image. The resultant image splits objects of interest into different regions thereby achieving the intended segmentation in a meaningful manner.



(a) Image classification

(b) Semantic Segmentation

Figure 1: Illustration of Image Classification and Semantic Segmentation[5]

background label added.

However, unlike the task of image classification[1a], where the location and boundaries of the desired object in the image is irrelevant, semantic segmentation[1b] requires the neural network to be able to spatially localize desired objects and its boundaries. Also, image classification networks are only required to produce probability scores to different desired classes based on the input image whereas a semantic segmentation network is required to produce a single channel segmentation map which is of the same size as the input image.

Despite these differences, state-of-the-art approaches adopt image classification networks and repurpose them for semantic segmentation. One reason for this adoption is to avoid much of the overhead involved in training the network from scratch. Another reason is concerned with the difficulty in obtaining datasets for semantic segmentation because of the need to annotate every pixel to obtain ground truth. This difficulty is apparent when the number of fine annotated images in the commonly used Cityscapes dataset (5000 fine annotated images) for semantic segmentation is compared with ImageNet dataset (over a million labeled images) for image classification. In light of this difficulty, it makes sense to make use of the transfer learning approach to transfer learned representations from the image classification pretrained networks to semantic segmentation networks. This enables the use of large networks though only a limited amount of training data is available.

The underlying image classification networks often contain a large number of layers and free parameters leading to high memory requirements. This makes them unsuitable for embedded or mobile devices where memory available in limited. Also, real time performance is hindered because the networks are computationally heavy. These drawbacks can be addressed by using strategies proposed in the literature such as pruning, quantization and others. The goal of this work is to make use of such strategies on state-of-the-art semantic segmentation networks to obtain a lean model suitable for application in an embedded setting.

2 Related Work

The related works can be broadly grouped into three categories:

1. Improving accuracy.
2. Resource efficient convolutional neural networks.
3. Reducing resource requirements.

Title sizes reduced.

2.1 Improving accuracy

This section addresses deep learning semantic segmentation models which focus on accuracy and not on the resource requirements.

- Fully Convolutional Network:[5]
  A popular approach which belongs to this category is the Fully Convolutional Network[6] which replaces the last few fully connected layers of image classification pretrained CNNs with convolutional layers for the

purpose of semantic segmentation. Information from the lower pooling layers are combined with the final prediction to improve the fineness of the segmentation. The authors call this as skip architecture. The major drawback however is the lack of global context awareness due to the downsampling operations which make the network spatially invariant. The importance of global context can be stated with an example. For instance in the figure 1a, it can be seen that a set white pixels, when looked at locally, could either belong to the cube or the cup. Only when looked at globally, it is possible to differentiate between the two.

- Encoder-Decoder architectures:[5]
  In this type of architectures, the encoder creates a low resolution feature map by downsampling the input image. The decoder works on the resultant low resolution feature maps and upsamples them to the original image resolution using fractionally strided convolutions. Both the encoder and decoder contain trainable kernels. A popular example of this architecture is the SegNet[7].

- Approaches to integrate context knowledge:[5]
  Since, global context awareness is significant for semantic segmentation, a number of approaches have been proposed to integrate context knowledge.
  One such approach is the use of Conditional Random Fields as a post processing step to make coarse predictions finer. The approach called DeepLab[4], for instance, makes use of fully connected pairwise CRF. Dilated convolutions is another approach in which the kernels are enlarged without an increase in the number of free parameters such that each kernel looks into a larger area of the feature map. DeepLab[4] also uses dilated convolutions. Other approaches also make use of multi-scale aggregation, Recurrent Neural Networks.

spelling fixed.

This project does not intend to look into approaches in this category in detail. Rather, a subset of approaches will be selected based on criteria such as accuracy and model size.

2.2 Resource efficient convolutional neural networks

This section addresses literature dealing with strategies to improve the resource efficiency of deep convolutional neural networks. Related approaches fall into one of the following:

- Reducing the precision of weights and feature maps:[8]

   1. Quantization: The idea behind quantization is that reducing the FLOPS (Floating Point Operations per Second) involved during inference, would lead to faster inference times. The weights could be quantized and represented in integers thereby reducing the memory requirements. Quantizing the feature maps during inference would reduce the computation requirements. However, accuracy of the resultant network would be lower. Quantization can be done to an extent to which loss of precision does not lead to considerable accuracy loss.
   2. Weight Sharing: In weight sharing, a set of weight parameters are forced to always have the same values. This could be the weights belonging to a kernel or a entire feature map. With weight sharing, one value can be stored and the weights which share this value can just store the index of the value. This could lead to reduced storage requirements at the cost of accuracy loss.

- Reducing the model size:[8]

   1. Pruning: Pruning explores the possibility to removing weights in the CNN which are redundant and do not contribute to the accuracy of the model. This leads to a reduction in the number of weights, thereby reducing memory requirements. However, this requires a strategy to determine whether a weight is redundant or not. One possible strategy is to set a feature map to zero and check whether the accuracy has changed. If the accuracy is not affected, the feature map can be removed. This process can be repeated for all feature maps. However, this is not feasible because of the sheer number of feature maps involved. Pavlo Molchanov et al. ([9]), point out that the VGG-16 net has 4224 convolutional feature maps and the discussed strategy would require 2^{4224} evaluations which is not possible.
      Therefore, a good strategy is required for pruning in order to determine the redundancy of weights.

spelling fixed.

A number of other techniques also exist such as the use of Huffman coding to reduce storage requirements[10]. These approaches will be looked into in detail by this project.

2.3 Reducing resource requirements

This section addresses deep learning semantic segmentation models which focus on reducing the resource requirements while maintaining accuracy. These approaches are closely related to this project.

- ENet [11]: In this approach, the authors exploit certain design choices which enables them to arrive at a reduction in the resource requirements namely memory and inference time. For instance, the early layers of the network heavily reduce the input size based on the notion that visual information is spatially redundant and also that the initial layers are expected to act like good feature detectors by preprocessing the input. This design change speeds up inference time because computations in the early layers are reduced. Other design choices are also made leading to further reduction in resource consumption.

ENet achieves considerable reduction in resource requirements, by exploring the use of alternate design choices. However, existing strategies described in Section 2.2 could also be used for the same purpose.

3 Problem Statement

split into problem statement and proposed approach.

The problems intended to be addressed in this project are listed below:

problem

- The underlying neural network architectures used in the state-of-the-art semantic segmentation approaches are often deep with many layers and have a large number of tunable parameters (for instance ResNet with 110 layers has 1.7 million parameters [1]).
- This hinders the use of CNNs in embedded or mobile devices, where memory and energy consumption are strict constraints.
- Strategies proposed in the literature to reduce resource requirements of CNNs, often, lead to degradation in performance. So an optimal trade-off between performance in terms of an accuracy metric such as Intersection over Union(IoU), and reduction of resource requirements needs to be identified.
- Though current state-of-the-art approaches achieve good results, efforts to visualize the representations learned by the network seems to be minimal. Often, only the final performance achieved is emphasized and the design changes made to the CNN architecture are presented as intuitions.

An outline of the approach this project intends to take is given below:

- Different strategies have been proposed in the literature to improve the efficiency of CNNs and make them suitable for embedded or mobile devices, namely pruning [9], quantization [12]. This project aims to identify such strategies and comparatively analyze their effects.
- The approach by [13], apply pruning on CNN architectures used for image classification (AlexNet, GoogleNet and SqueezeNet) and analyze the effects of such algorithms.

approach

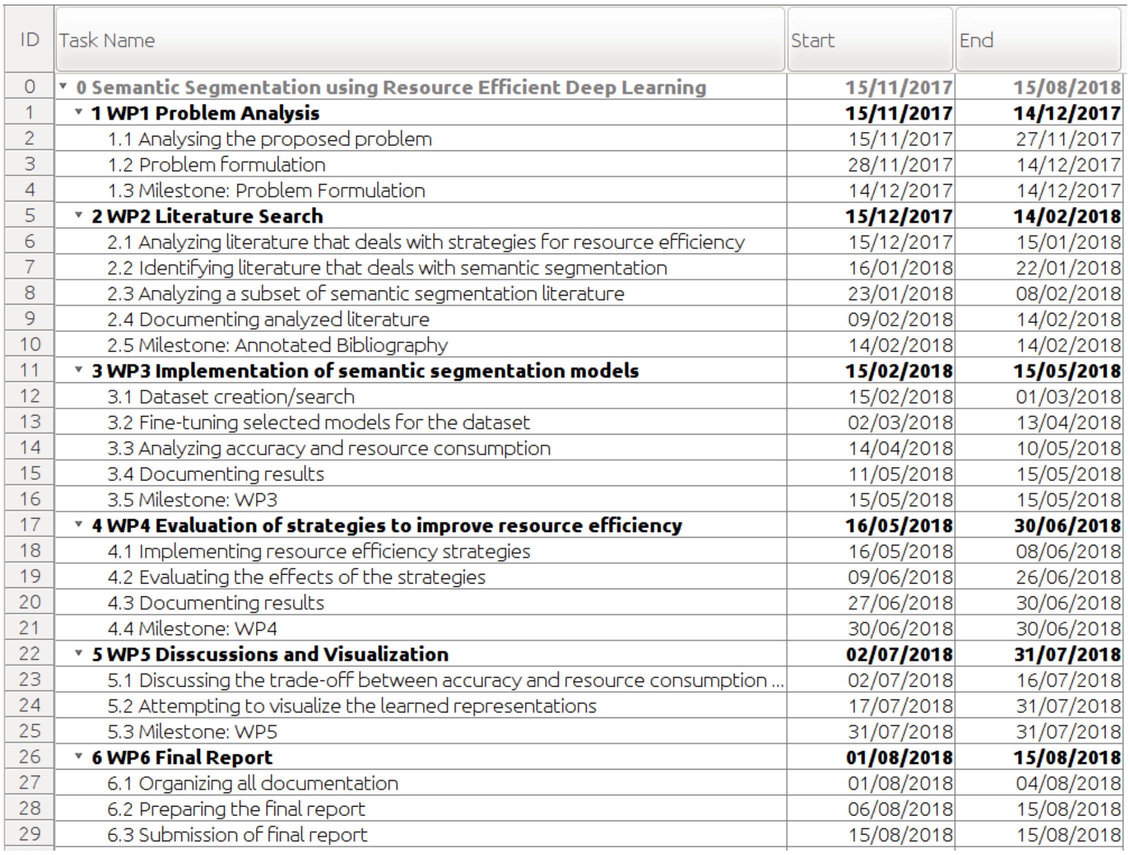
- This project intends to follow a similar approach and aims to apply the identified strategies, on a set of current state-of-the-art CNN architectures used for semantic segmentation such as FCN [6], SegNet [7] and ENet [11] and analyze the resultant effects.
- This project aims to visualize at various stages of application, the effects of identified strategies on the activations of convolution filters through existing approaches such as Zeiler and Fergus [14].

4 Project Plan

The work plan for this project begins on December 15th, 2017 and ends in August 15th, 2018. The total project duration is 8 months.

4.1 Work Packages

Figure 2: Work Packages



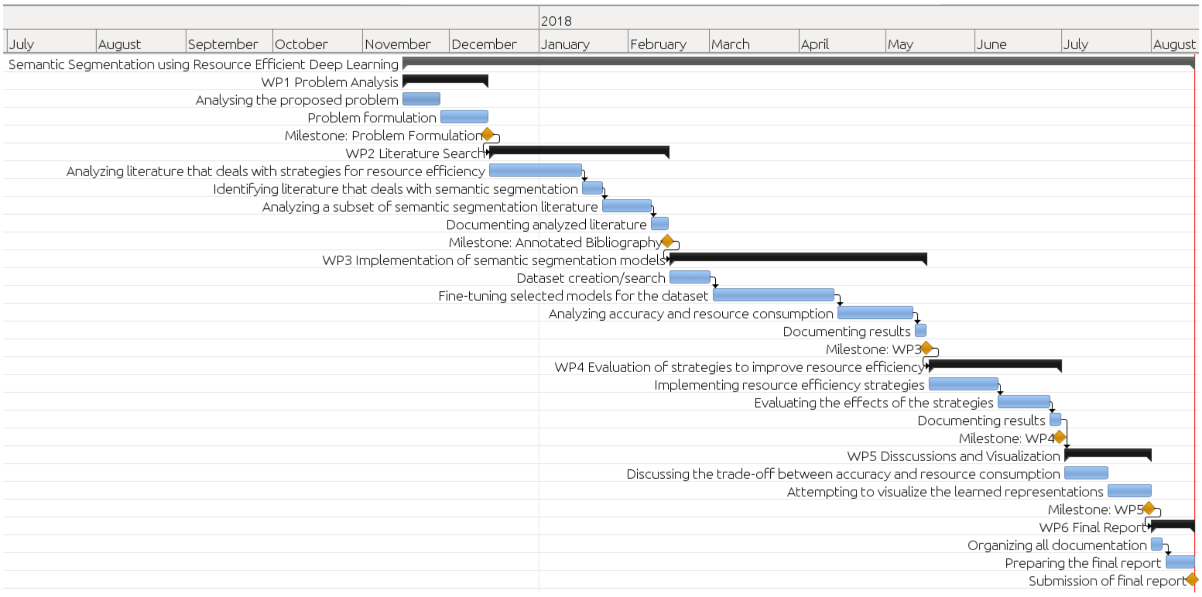
ID	Task Name	Start	End
0	▾ **0 Semantic Segmentation using Resource Efficient Deep Learning**	**15/11/2017**	**15/08/2018**
1	▾ **1 WP1 Problem Analysis**	**15/11/2017**	**14/12/2017**
2	1.1 Analysing the proposed problem	15/11/2017	27/11/2017
3	1.2 Problem formulation	28/11/2017	14/12/2017
4	1.3 Milestone: Problem Formulation	14/12/2017	14/12/2017
5	▾ **2 WP2 Literature Search**	**15/12/2017**	**14/02/2018**
6	2.1 Analyzing literature that deals with strategies for resource efficiency	15/12/2017	15/01/2018
7	2.2 Identifying literature that deals with semantic segmentation	16/01/2018	22/01/2018
8	2.3 Analyzing a subset of semantic segmentation literature	23/01/2018	08/02/2018
9	2.4 Documenting analyzed literature	09/02/2018	14/02/2018
10	2.5 Milestone: Annotated Bibliography	14/02/2018	14/02/2018
11	▾ **3 WP3 Implementation of semantic segmentation models**	**15/02/2018**	**15/05/2018**
12	3.1 Dataset creation/search	15/02/2018	01/03/2018
13	3.2 Fine-tuning selected models for the dataset	02/03/2018	13/04/2018
14	3.3 Analyzing accuracy and resource consumption	14/04/2018	10/05/2018
15	3.4 Documenting results	11/05/2018	15/05/2018
16	3.5 Milestone: WP3	15/05/2018	15/05/2018
17	▾ **4 WP4 Evaluation of strategies to improve resource efficiency**	**16/05/2018**	**30/06/2018**
18	4.1 Implementing resource efficiency strategies	16/05/2018	08/06/2018
19	4.2 Evaluating the effects of the strategies	09/06/2018	26/06/2018
20	4.3 Documenting results	27/06/2018	30/06/2018
21	4.4 Milestone: WP4	30/06/2018	30/06/2018
22	▾ **5 WP5 Disscussions and Visualization**	**02/07/2018**	**31/07/2018**
23	5.1 Discussing the trade-off between accuracy and resource consumption ...	02/07/2018	16/07/2018
24	5.2 Attempting to visualize the learned representations	17/07/2018	31/07/2018
25	5.3 Milestone: WP5	31/07/2018	31/07/2018
26	▾ **6 WP6 Final Report**	**01/08/2018**	**15/08/2018**
27	6.1 Organizing all documentation	01/08/2018	04/08/2018
28	6.2 Preparing the final report	06/08/2018	15/08/2018
29	6.3 Submission of final report	15/08/2018	15/08/2018

feedback: probably need more time for report writing

30 days for report preparation split across the packages.

4.2 Gantt Chart

Figure 3: Gantt Chart



4.3 Deliverables

4.3.1 Minimum Viable

1. Analysis of the current state-of-the-art strategies to improve resource efficiency of convolutional neural networks.

2. Selection of a subset of such strategies based on feasibility of usage for convolutional neural networks trained for semantic segmentation.

4.3.2 Expected

1. Dataset creation for objects used in the Robocup@Work or Robocuo@Home environment.

2. Selection of a set of deep architectures used for semantic segmentation, fine-tuning them on the created dataset and reporting the results based on one or more performance metrics.

3. Applying the selected resource efficiency strategies on the fine-tuned models and arriving at a trade-off between accuracy and resource consumption.

4.3.3 Maximum

Changed from Desired to Maximum

1. Visualizing the learned representations of the trained networks through existing techniques such as Zeiler and Fergus [14]. → moved to maximum.
2. Integrating the resultant network with the tasks in Robocup. ← Feedback: Integrating the resultant network with tasks in RoboCup
3. Further analyzing the resultant network to improve its efficiency to meet a tighter memory ($<$10MB) and real time ($<$1s) constraints (If this constraints are not already met by the resultant network).

Feedback: not really a robocup requirement..

changed to "tighter requirements".

References

[1] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian , "Deep residual learning for image recognition", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2016).

[2] Redmon, Joseph and Divvala, Santosh and Girshick, Ross and Farhadi, Ali, "You only look once: Unified, real-time object detection", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2016).

[3] Simonyan, Karen and Zisserman, Andrew, "Two-stream convolutional networks for action recognition in videos", in Advances in neural information processing systems, (2014).

[4] Chen, Liang-Chieh and Papandreou, George and Kokkinos, Iasonas and Murphy, Kevin and Yuille, Alan L , "Semantic image segmentation with Deep Convolutional Nets and fully connected CRFs", in International Conference on Learning Representations, (2016).

[5] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, Jose Garcia-Rodriguez, "A Review on Deep Learning Techniques Applied to Semantic Segmentation", arXiv preprint arXiv:1704.06857, 2017.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431-3440.

[7] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," arXiv preprint arXiv:1505.07293, 2015.

[8] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, Joel Emer,"Efficient Processing of Deep Neural Networks: A Tutorial and Survey", in arXiv preprint arXiv:1703.09039, 2017.

[9] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, Jan Kautz, "Pruning Convolutional Neural Networks for Resource Efficient Inference", in arXiv preprint arXiv:1611.06440, 2016.

[10] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," arXiv preprint arXiv:1510.00149, 2015.

[11] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. , "Enet: A deep neural network architecture for real-time semantic segmentation." arXiv preprint arXiv:1606.02147, 2016.

[12] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, Jian Cheng, "Quantized Convolutional Neural Networks for Mobile Devices", in Computer Vision and Pattern Recognition, 2016.

[13] T.-J. Yang, Y.-H. Chen, and V. Sze, "Designing Energy-Efficient Convolutional Neural Networks using Energy-Aware Pruning," in CVPR, 2017.

[14] Zeiler, Matthew D. and Fergus, Rob, "Visualizing and Understanding Convolutional Networks", in Springer International Publishing, (2014).