

Sematic Segmentation using Resource Efficient Deep Learning

Naresh Kumar Gurulingan
naresh.gurulingan@smail.inf.h-brs.de

Hochschule Bonn-Rhein-Sieg

November 9, 2018



**Hochschule
Bonn-Rhein-Sieg**
University of Applied Sciences

Table of Contents

Introduction

Applications

Dataset

Annotation process

Artificial image generation

Dataset variants

Dataset analysis

DeepLabv3+

Results

Contributions and future work

Semantic segmentation

Divide an input image into different regions which contain a desired object or background.



Left: Input image; Right: Segmentation result.

Applications

- a Autonomous cars
- b Robotics
- c Augmented reality



(a) Street scene



(b) Indoor scene



(c) Augmented guide

Dataset

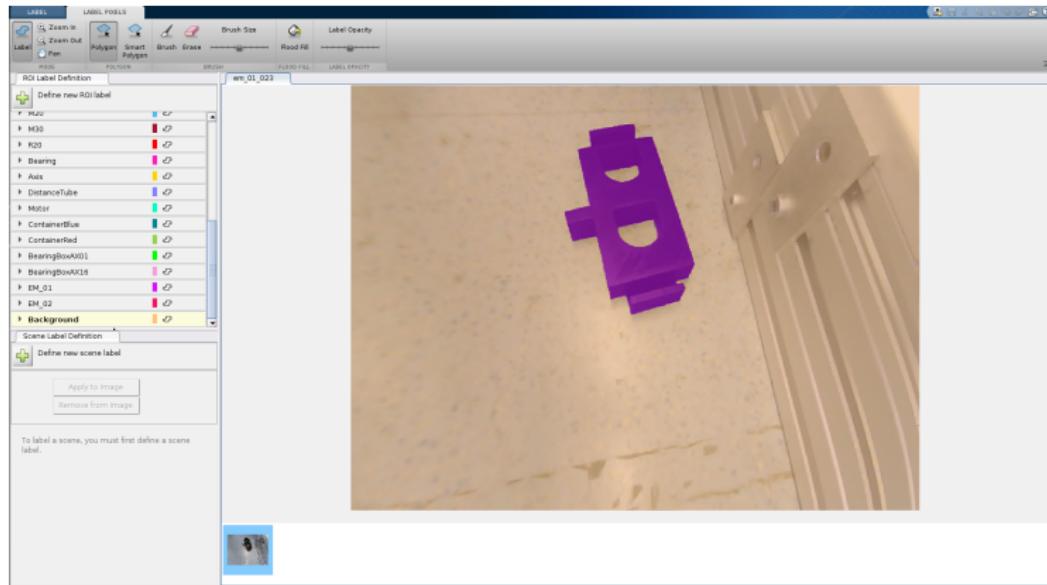
Objects in the dataset



This figure shows all the 18 objects in the dataset. First row from left: "distance_{_}tube", "m20", "bearing", "axis", "r20", "m30", "m20_100", "motor", "bearing_{_}box_ax16", "bearing_{_}box_ax01", "f20_20_B", "f20_20_G". Second row from left: "em_01", "s40_40_B", "s40_40_G", "em_02", "container_{_}box_red", "container_{_}box_blue".

Annotation process

MATLAB ImageLabeler



A sample object being labeled in ImageLabeler.

Motivation

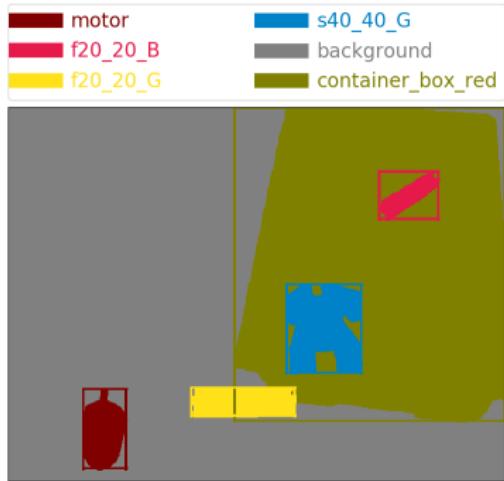
- ▶ For an image containing 1 desired object, roughly 4 minutes was spent for manual annotation.
- ▶ Capturing diverse real-world variations is time consuming.

Process

- ▶ Collect RGB intensity values of objects using manual annotation.
- ▶ Create a list of all the collected objects.
- ▶ For an artificial image, select a background image and random objects from the list of objects.
- ▶ Place the selected objects at random locations and at random scales.
- ▶ Correspondingly generate semantic labels and object detection labels.

Artificial image generation

Sample result



Sample results produced by the artificial image generation algorithm. In each row, the image on the left shows the generated artificial image and the image on the right shows a visualization of the semantic segmentation label and object detection label. At the top of every label visualization image, the objects in the image and their corresponding colors in the visualization are indicated.

Dataset variants

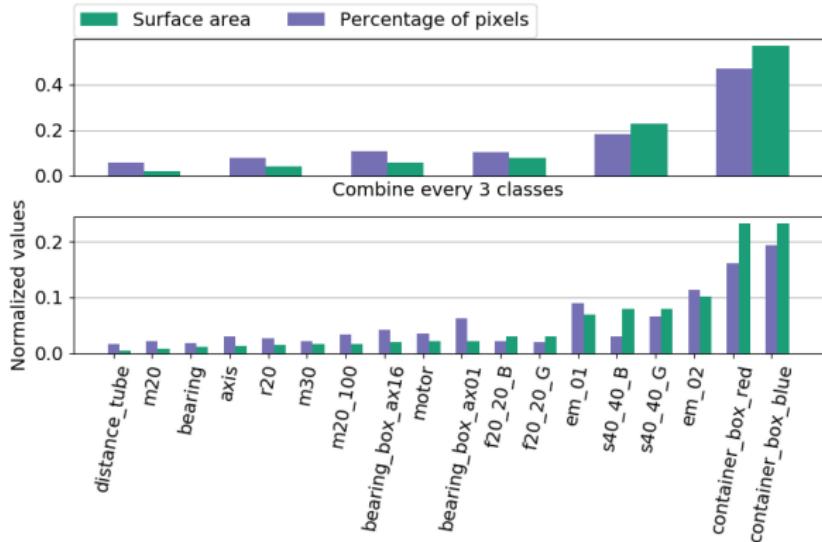
Motivation

- ▶ Inability to distinguish size.
- ▶ Inability to distinguish shape.

	Training	Validation	Test
Real Images	396	72	69
Artificial Images	7104	870	870
Total Images	7500	942	939

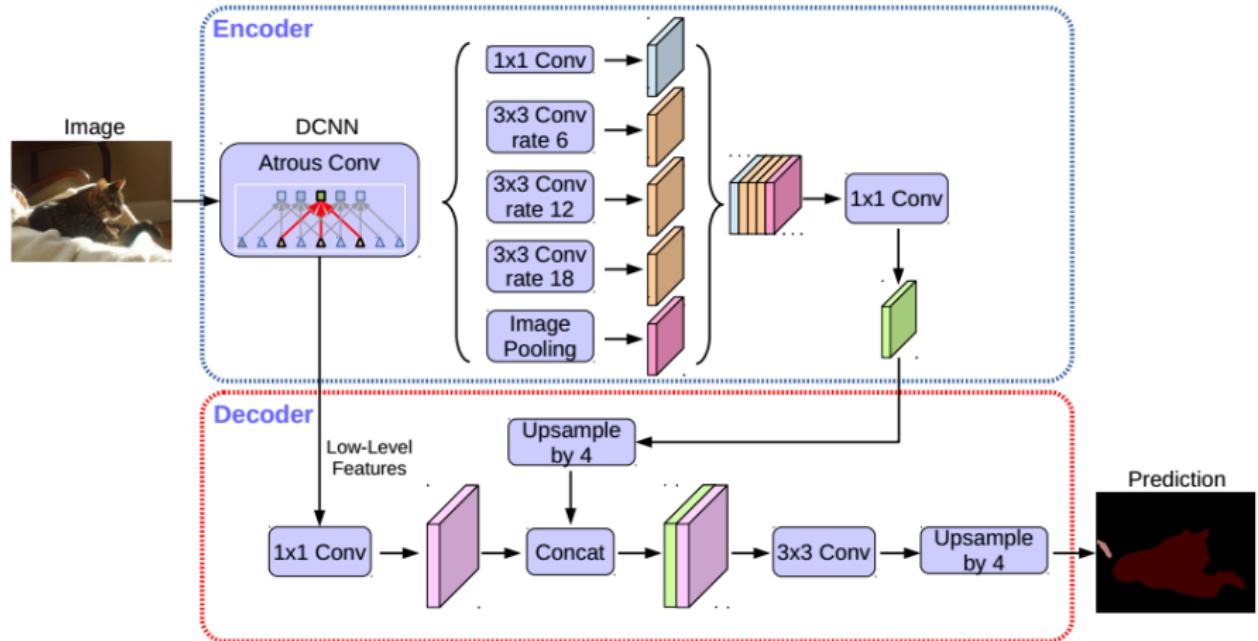
Meta-data of the dataset.

Dataset analysis



Relation between percentage of pixels occupied by different objects and their corresponding real-world surface area. (Percentage of pixels of an object = $\frac{NP_o}{NP_s}$.
NP_o = Number of pixels occupied by the object in the training set and NP_s = Total number of pixels in the training set)

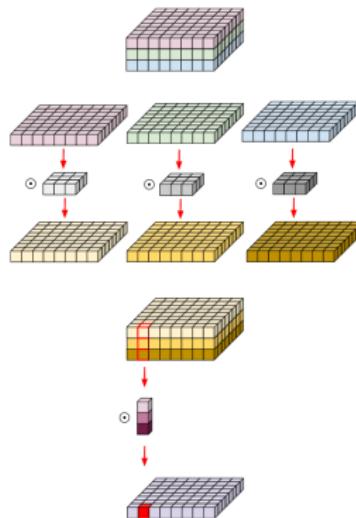
Architecture of DeepLabv3+



An illustration of DeepLabv3+ architecture. The encoder extracts features at different scales and the decoder refines object boundary delineation.

Depthwise seperable convolutions

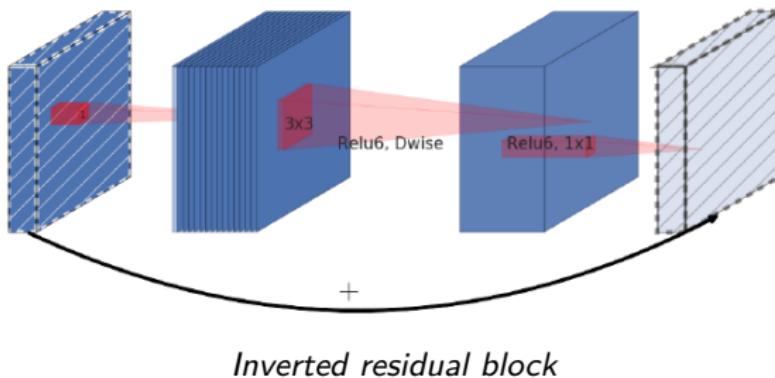
- ▶ First depthwise convolution, then pointwise convolution.
- ▶ Row 2 to row 5: depthwise convolution, 6th row: pointwise convolution.



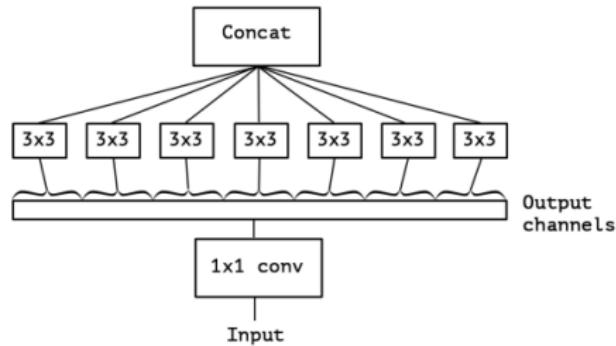
Depthwise seperable convolution.

MobileNetv2 encoder of DeepLabv3+

- ▶ Depthwise seperable convolutions.
- ▶ Inverted residual with linear bottleneck.



Xception encoder of DeepLabv3+



Xception module

Atrous separable convolutions

Comparing encoders

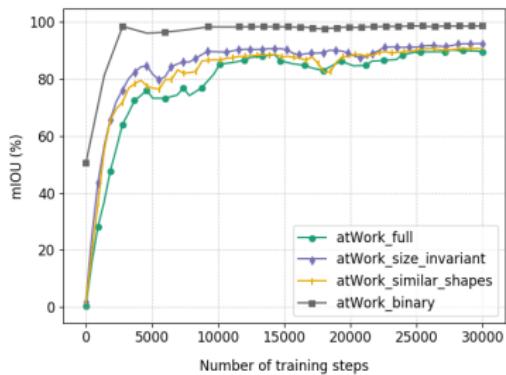
- ▶ Per class IOU = $\frac{\text{ground_truth} \cap \text{prediction}}{\text{ground_truth} \cup \text{prediction}}$
- ▶ mIOU = mean of all class IOUs.
- ▶ DeepLabv3+ with Xception encoder achieves higher mIOU on all four dataset variants.

Dataset variant	mIOU in %	
	MobileNetv2	Xception
atWork_full	77.47	89.63
atWork_size_invariant	83.10	92.47
atWork_similar_shapes	82.10	90.71
atWork_binary	96.06	98.68

This table lists the mIOU obtained by DeepLabv3+ with MobileNetv2 and Xception encoders on 4 dataset variants.

Comparing dataset variants

- ▶ Background/foreground segmentation leads to the highest mIOU.
- ▶ Treating all objects as different classes leads to the lowest mIOU.
- ▶ Combining objects similar in shape, size or color improves mIOU.

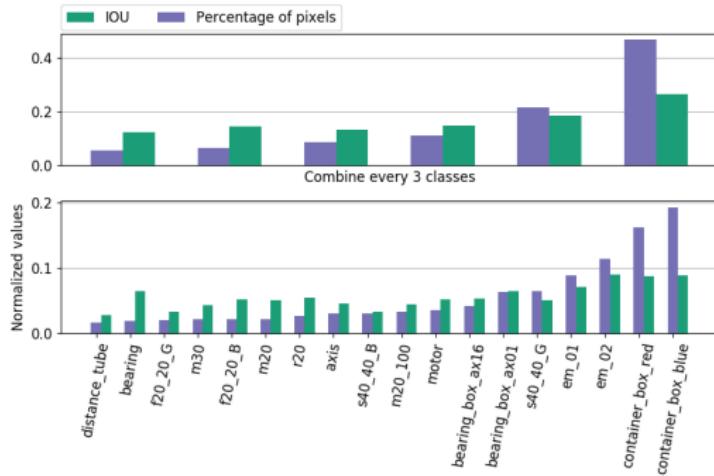


*mIOU (%) vs Number of training steps
DeepLabv3+ with Xception encoder on all dataset variants*

Comparing individual classes

Comparing individual classes

- ▶ Class IOU shows an increasing trend with increase in Percentage of pixels.
- ▶ Percentage of pixels is shown to increase with surface area [Slide 10].
- ▶ DeepLabv3+ tends to learn larger objects first.



Individual class IOUs achieved by DeepLabv3+ with MobileNetv2 encoder is plotted with the percentage of pixels.

Quantizing the inference graph

- ▶ The inference graph is quantized.
- ▶ With MobileNetv2 encoder, 67 % drop in occupied disk memory is achieved. Drop in mIOU is around 9 %.
- ▶ With Xception encoder, 73 % drop in occupied disk memory is achieved. Drop in mIOU is around 2 %.

Quantizing the inference graph

Encoder	mIOU (%)	Number of parameters	FLOPS	Disk memory (MB)
MobileNetv2	84.66	2.11M	6.41B	8.7
MobileNetv2-8	75.17	2.11M	328.87M	2.8
Xception	92.42	41.05M	126.27B	165.6
Xception-8	90.4	41.05M	1.94B	44.7

This table summarizes the average mIOU across all four dataset variants, number of parameters, and floating point operations (FLOPS) of both the quantized and full precision encoders of DeepLabv3+. "M" denotes million and "B" denotes billion.

Conclusion and future work

Contributions

- ▶ Artificial image generation algorithm.
- ▶ Segmentation dataset with 18 atWork objects.
- ▶ Evaluation of DeepLabv3+ with resource efficient encoders MobileNetv2 and Xception.

Future work

- ▶ Model interpretability.
- ▶ Architecture search.
- ▶ Fusion of 2D image data with point cloud information.

Thank you very much!

Are there any questions?

References