oise Functions for Synthesising Abstract Deep Learning training data

Søren Winkel Holm, s183911

November 18, 2022

Contents

1 Introduction

In the field of computer vision, deep learning is often used to solve problems such as object detection and image segmentation. Two main problems are (1) the high cost of annotating the image by producing the label maps (such as this) (2) the high cost of training the deep learning models. Automatic generation of scenes with corresponding label maps might be usable even if the scenes are abstract. Such data could be used to pretrain the deep learning model and act as an inductive bias such that less training and less data is required when actually training on the real dataset. For this to work, the generated scenes should have objects with object having some assigned class which determines some visual characteristics of the object. The scenes do not have to have anything to do with the actural task but need to make the model learn som basics of object detection, classification and segmentation. An easy first solution would be having the class determine the colour of the object but it is much more realistic if the objects are textured with noise functions and the parameters of the procedurally generated noise were determined by the class.

2 Method

- Render a single sphere (using recursive subdivision) lighted by directional light using Phong shading. Texture the sphere using the noise functions from here. Allow user input controlling noise type and some of the parameters.
- Add multiple spheres to the scene, randomly selecting placement and size for each.
- Choose a number of classes (maybe 3) and give each class a distribution of shapes, of colours and a distribution of noise parameters (such as related smoothing and scale). Assign each sphere to a class and let its shape, colour and noise be sampled from the class prior. (I will be using spheres and for shape I will use a model matrix that transforms the sphere into an ellipsoid by stretching it. The amount of stretching should thus be determined by object class)
- Add the possibility of producing the label image by having a rendering mode which does not have any lighting or texturing, just colours each object according to its class. This image will then have the class in the pixel position according to the object at that pixel position. Allow for saving the current canvas to a png.

- Generate a large number (about 1000) of scenes and corresponding label maps.
- Follow a classic deep learning image segmentation benchmark such as SegNet on a subset of theCoco dataset. First test normal performance. Then pretrain the model on the generated images before transfer learning it to the coco task and compare performance. Compare performance between noise functions.

3 Results

4 Discussion