Focusing attention of Fully convolutional neural networks on Region of interest (ROI) input map. For recognition of materials within glass vessels, in chemistry laboratory setting.

This project contain code for fully convolutional neural network (FCN) for semantic segmentation with region of interest mask as additional input (figure 1). The net receive image and ROI as binary map with pixels corresponding to ROI marked 1, and produce pixel wise annotation of the ROI region of the image according to several level of categories. This code was specifically designed for semantic segmentation task of materials in transparent vessels were the vessel region of the image is already known. It also supply dataset for materials inside vessel in chemistry laboratory setting. But can be used in any case of semantic segmentation with an arbitrary shaped region of interest (ROI) is given.

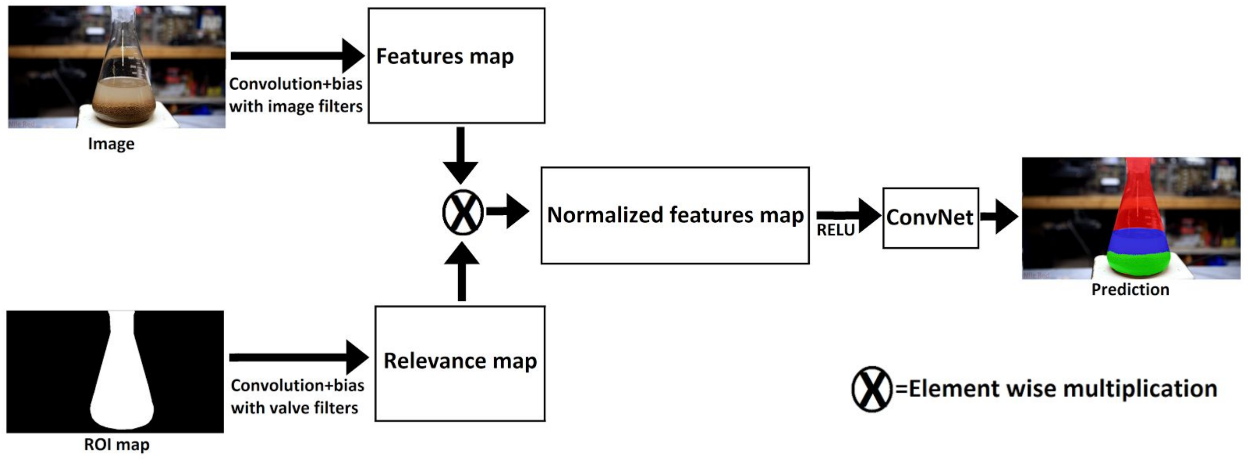
Image
Convolution+bias with image filters
Features map
Normalized features map
RELU
ConvNet
Prediction
ROI map
Convolution+bias with valve filters
Relevance map
=Element wise multiplication

Figure 1. The valve filter approach for introduction of ROI map as input to ConvNets. The image and the ROI input are each passed through a separate convolution layer to give feature map and Relevance map, respectively. Each element in the features map is multiplied by the corresponding element in the feature map to give a normalized features map that passed (after RELU) as input for the next layer of the net.

General approach (valve filter method)

Convolutional neural networks as emerged as the leading methods for detection classification and segmentation of images. Many problems in image recognition demand the recognition to be performed only on specific predetermined Region Of Interest (ROI) in the image. Example for such case is recognition of the contents of glassware vessels such as a bottle or jar. Where the glassware region in the image is known and given as the ROI input (Figure 1). Directing the attention of a convolutional neural net (CNN) to a given ROI region without losing background information is a major challenge in this case. This project use valve filters approach to focus the attention of a fully convolutional neural net (FCN) on a given ROI in the image. The ROI mask is inserted to CNN along with the image as a binary map with pixels belonging to the ROI set to one and the rest set to zero. Processing the ROI in the net is done using the valve filters approach presented in in Figure 1. In general for each filter that acts on the image a corresponding valve filter exist that act (convolve) with the ROI map (Figure 1). The output of the valve filter convolution is multiplied element wise with the output of the image filter convolution, to give normalized feature map (Figure 1). This map is used as input for the next layers of the net. In this case, the net is a standard fully convolutional net (FCN) for semantic segmentation (pixel-wise classification). Valve filters can be seen as kind of valves that regularize the activation of image filters in different regions of the image.

Requirements

This network was run and trained with Python 3.6 Anaconda package and Tensorflow 1.1. The training was done using Nvidia GTX 1080, on Linux Ubuntu 16.04.

Setup

1. Download the code from the repository.
2. Download pretrained vgg16 net and put in the /Model_Zoo subfolder in the main code folder. A pretrained vgg16 net can be download from here[https://drive.google.com/file/d/0B6njwynsu2hXZWcwX0FKTGJKRWs/view?usp=sharing] or from here [**ftp://mi.eng.cam.ac.uk/pub/mttt2/models/vgg16.npy]**
3. Download dataset of images materials in transparent vessel and extract in /Data_Zoo folder in the main code dir. The dataset can be download from here https://drive.google.com/file/d/0B6njwynsu2hXRFpmY1pOV1A4SFE/view?usp=sharing
4. If you are not interested in training the net then download a pre trained model and extract to /log folder in the main code dir. The pretrained model could be download from: https://drive.google.com/file/d/0B6njwynsu2hXRDMxWlBUTWFZM2c/view?usp=sharing

Tutorial

Prediction using trained network

(pixelwise classification and segmentation of images)

Run: Inference.py

Training network:

Run: Train.py

Evaluating net performance using intersection over union (IOU):

Run: Evaluate_Net_IOU.py

Notes and issues

1) Make sure the /Data_Zoo folder contain the downloaded dataset for for materials in vessels. Also the Label_Dir folder point to this folder and the Image_Dir should point to test images subfolder of this dataset (should the default code setting)

1) If running inference or evaluation make sure that the /log folder contain a trained net you can either download trained model or trained it yourself.

Detail valve filters implementation.

The detail implementation of the valve filters given in Figures 5 and described below:

1. The ROI map is inserted to the net along with the image. The ROI map is represented as a binary image with pixels corresponding to ROI marked 1 and the rest marked 0.
2. A set of image filters is convolved (with bias addition) with the image to give a feature map.
3. A set of valve filters convolved with the ROI map to give a relevance map with the same size and dimension as the feature map (again with bias addition).

4. The feature map is multiplied element wise by the relevance map. Hence, Each element in the relevance map is multiplied by the corresponding element in the feature map to give normalized feature map.
5. The normalized feature map is then passed through a Rectified Linear Unit (ReLU) which zero out any negative map element. The output is used as input for the next layer of the net.
6. The net in this case is standard fully convolutional neural net for semantic segmentation.

In this way each valve filter act as kind of a valve that regulates the activation the corresponding image filter in different regions of the image. Hence, the valve filter will inhibit some filters in the background zone and others in the ROI zone.

The valve filters weights are learned by the net in the same way the image filters are learned. Therefore the net learns both the features and the region for which they are relevant.

In the current implementation, the valve filter act only on the first layer of the convolutional neural net and the rest of the net remained unchanged.

Details input/output

The fully convolutional neural network receive an image with material in a glassware vessel and perform semantic segmentation of the image, such that that each pixel the in the image is assigned several labels. The network performed the pixelwise labeling in several level of class granularity and return for each level an image where the value of each pixel is the phase/material/object of this pixel in the image. All the predictions are generated simultaneously in one iteration of the net. The output segmentation maps/images are as following (See image):

a. Vessel/Background: For each pixel assign value of 1 if it in the vessel and 0 otherwise.

b. Filled/Empty: similar to above but also distinguish between filled and empty region of the vessel. For each pixel assign one of the 3 values: 0) Background, 1) Empty vessel. 2) Filled vessel

c. Phase type: Similar to above but distinguish between liquid and solid regions of the filled vessel. For each pixel assign one of the 4 values: 0) Background, 1) Empty vessel. 2) Liquid. 3) Solid.

d. Fine grain phase type: Similar to above but distinguish between fine grain specific phases regions of the filled vessel. For each pixel assign one of 15 values: 1) BackGround. 2) Vessel. 3) Liquid. 4) Liquid Phase two. 5) Suspension. 6) Emulsion. 7)

Foam. 8) Solid. 9) Gel. 10) Powder. 11) Granular. 12) Bulk. 13) Bulk Liquid. 14) Solid Phase two. 15) Vapor.

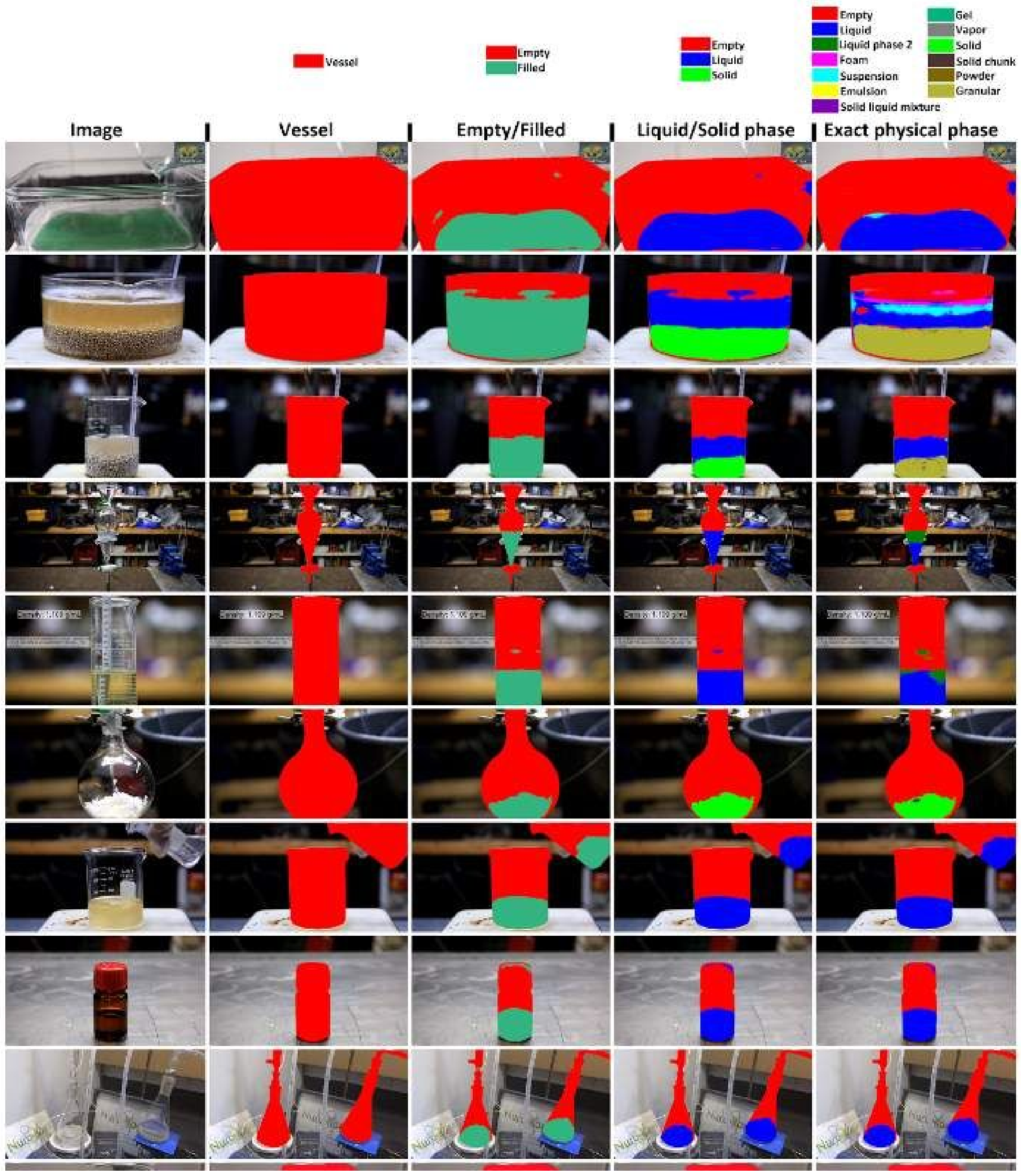

Figure 2) Prediction of the net for the dataset test are given below (the roi input is the vessel region)

Background

The net is based on fully convolutional neural net described in the paper Fully Convolutional Networks for Semantic Segmentation The main modification, other than the ROI input and valve filters, is that the last prediction layer is split to give prediction in several level of granularity for each pixel. Similarly training of the network was done with several loss function simultaneously one for each set of classes. See BuildNetVgg16.py for the network structure. The code is based on https://github.com/shekkizh/FCN.tensorflow by Sarath Shekkizhar with MIT licence. The net is based on the pretrained VGG16 model by Marvin Teichmann

Thanks

I would like to thank Alexandra Emanuel and Mor Bismuth for their work on labeling of the dataset. I also like to thank the creators of the Youtube channels NileRed, NurdeRage and ChemPlayer for allowing the use of frames from their videos for creating this dataset and this paper.