Scribble-based Image Segmentation with Convexity Priors

Chirag Angadi, Hruday Kumar Kolla, Siddharood Biradar

Guided by Prof. Dr. Michael Moeller

UNIVERSITÄT SIEGEN

Objective

- The objective is to design an architecture that can accurately segment an image based on user-provided scribbles, while also incorporating a convexity prior. Convexity prior refers to the assumption that objects or regions in an image are often convex in shape, meaning they have a smooth and continuous boundary without concave regions.

Overview

Image Segmentation for Convex objects based on Scribbled Images:

- Approach 1: Generating the convex segmented image using fully connected network and convex network architecture.
- Approach 2: : Generating the convex segmented image using segmentation network and convex network architecture.

Input Convex Network Architecture

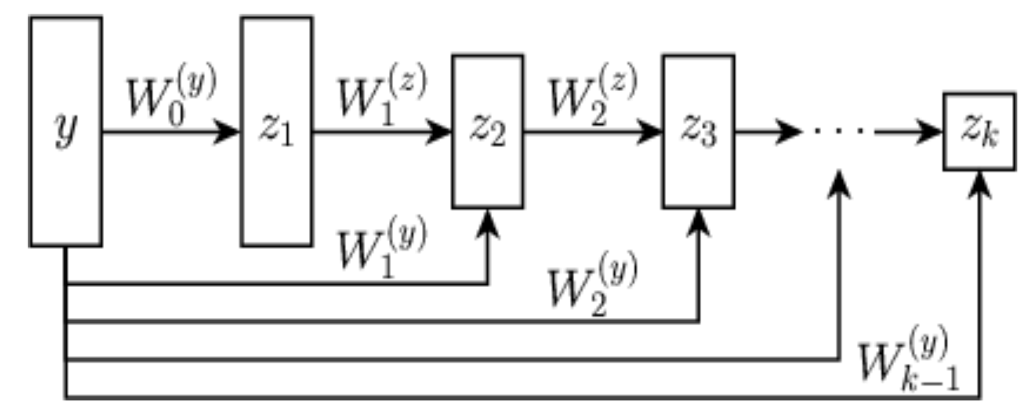


Figure: A fully input convex neural network (FICNN)

$Z_{i+1} = g_i\,(W_i^{(z)} Z_i + W_i^{(y)} y + b_i)$, $f(y;\theta) = Zk$

Where, Z_i is Layer activations (with Z_0, $W_0^{(z)}=0$),

$\theta = \{w^{(y)}_{0:k-1}, W^{(z)}_{1:k-1}, b_{0:k-1}\}$ are parameters, and g_i are non-linear activation functions.

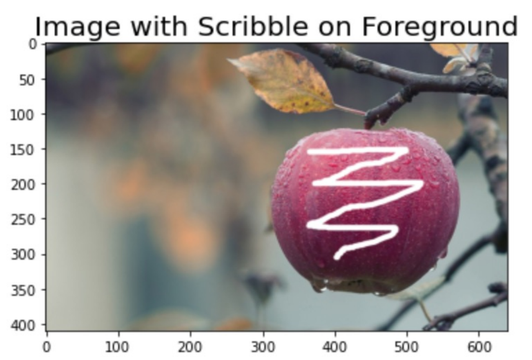
- The function f is convex in y provided that all $W^{(z)}_{1:k-1}$ are non-negative, and all functions g_i are convex and non-decreasing.

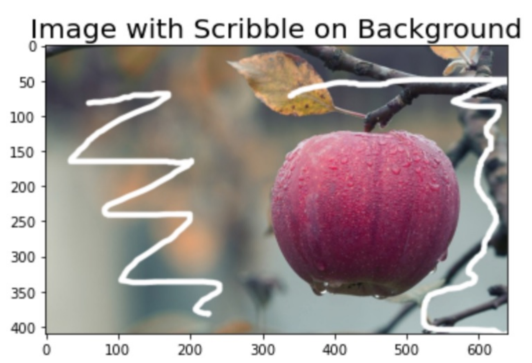
- *Source: Input Convex Neural Networks Brandon Amos[1] Lei Xu[2*] J Zico Kolter[1]*

References:

- *Input Convex Neural Networks Brandon Amos[1] Lei Xu[2*] J Zico Kolter[1]*
- *Boundary Perception Guidance: A Scribble-Supervised Semantic Segmentation Approach Bin Wang[1,3] , Guojun Qi[2] , Sheng Tang[1*] , Tianzhu Zhang[4] , Yunchao Wei[5] , Linghui Li[1,3] and Yongdong Zhang[1]*

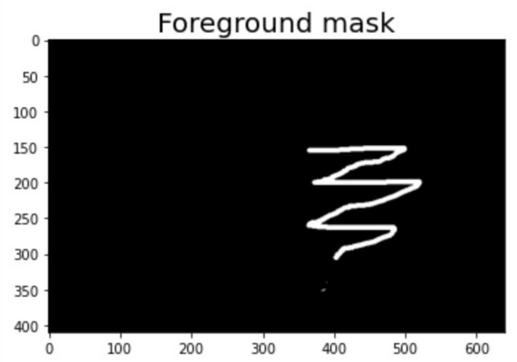
Input

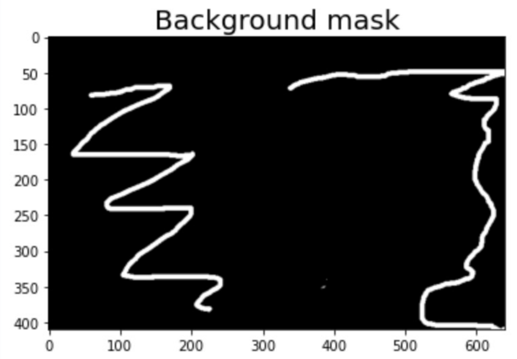




Background and foreground Scribbles are to be drawn on the image which acts as input to network.

Pre-Processing and Data extraction for Deep Neural Net





- Obtain Foreground and Background masks to extract data for the fully connected network.

- Foreground: 'n' many scribbled pixels with coordinates (x, y) and colour values (r, g, b).
- Foreground Label: '0'

- Background: 'm' many scribbled pixels with coordinates (x, y) and colour values (r, g, b).
- Background Label: '1'

- Vector of (x, y, r, g, b) with corresponding label (0/1).
- Total size of data is equal to number of scribbled pixels ('n+m').

Approach 1

Pre-Processed Data for Deep Neural Net → TRAINING → x, y, r, g, b

All pixel data of Image in the form of vector (x, y, r, g, b) → INFERENCE → x, y, r, g, b

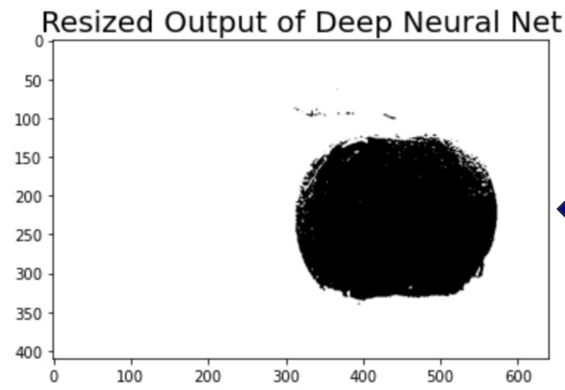
Fully Connected Neural Net:

- 5 input nodes and 1 output Node.
- 2 Hidden layers with 256 neurons each
- ReLu activation functions
- Binary Cross Entropy with Logits Loss

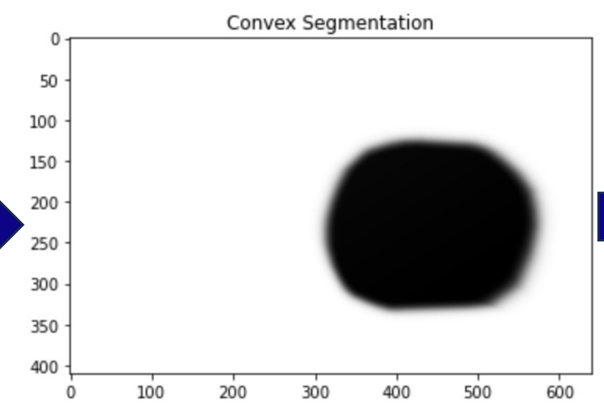
→ Out

- The Inference output has some wrongly segmented pixels as seen in figure.
- We send this output to convex Net to make it into a single convex region.

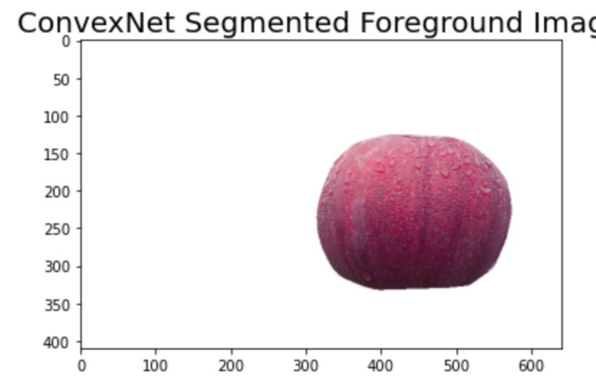
- Vector of (x,y) with corresponding label (0/1) of size (Image height x Image width) is sent to Convex Net for both training and Inference.

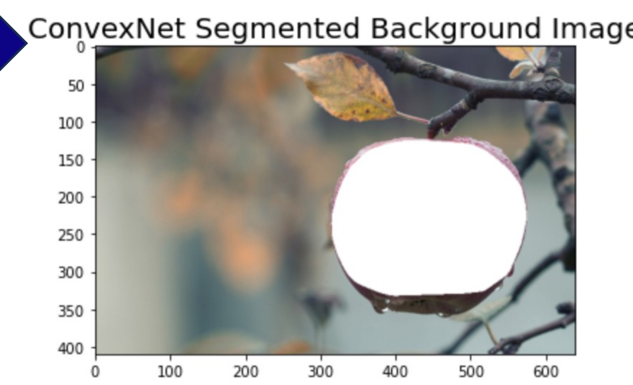


TRAINING → **Convex Network** → INFERENCE

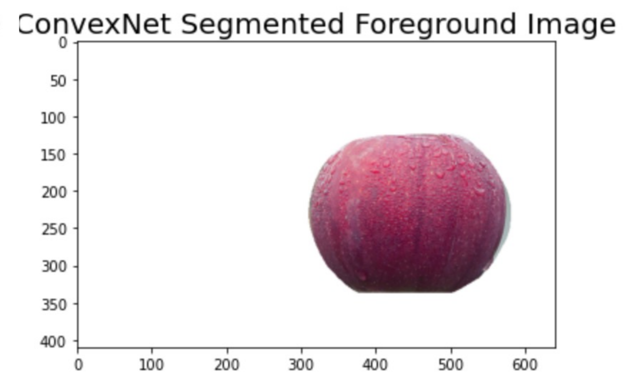


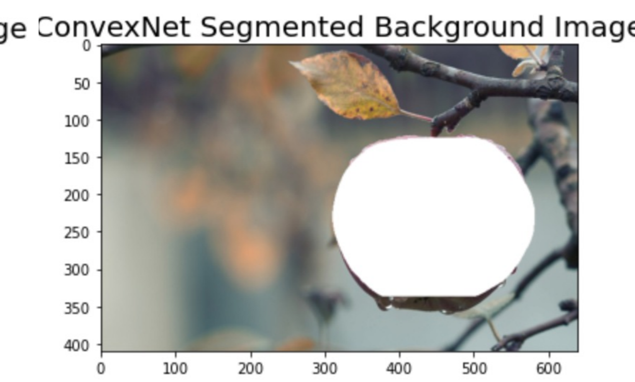
Approach 1 Result





Approach 2 Result

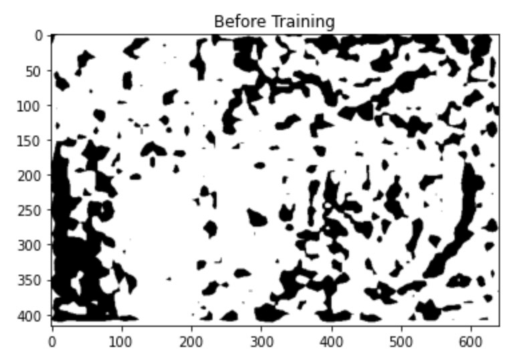




Approach 2

Input →

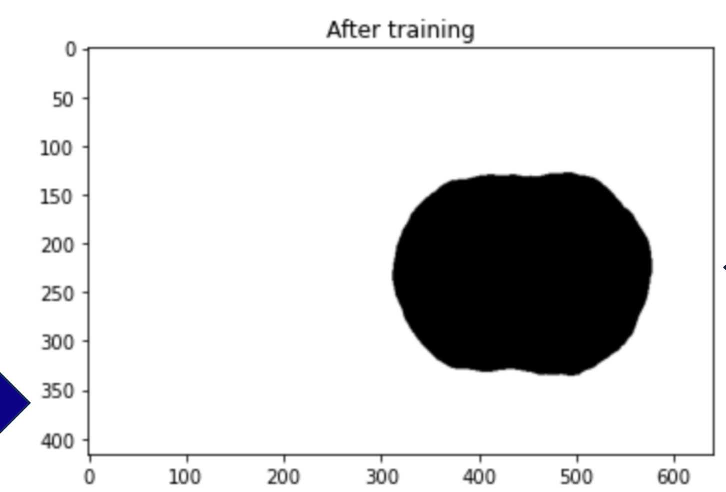
Segmentation Network:
Using the segmentation models pytorch library to create a model for segmentation. It uses the ResNet34 encoder with pre-trained weights from ImageNet for encoder initialization.



TRAINING

- Criterion = Binary Cross Entropy with Logits Loss.
- Loss = Criterion(labels of Segmentation output pixels, labels of Scribbled pixels), *for All Scribbled pixels*
- Loss back propagated through all layers

INFERENCE



TRAINING → **Convex Network** → INFERENCE