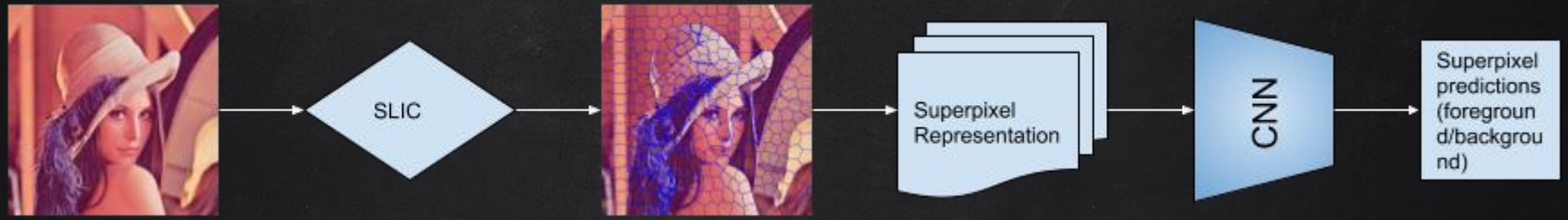
Learning Salient Objects in a Scene using Superpixel-augmented Convolutional Neural Network

Problem Overview

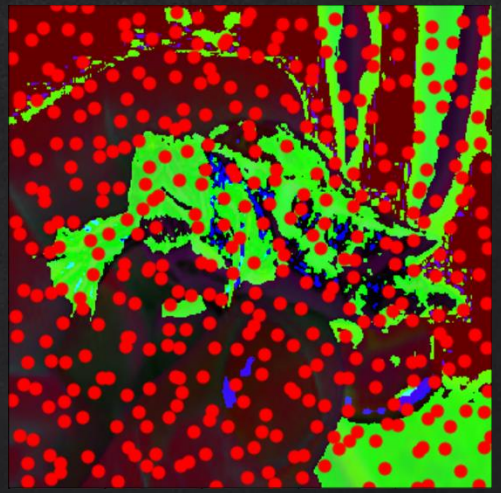
Given a natural scene image, objective is to detect "salient" regions.

Method overview



1. Superpixels are generated using SLIC segmentation algorithm
2. Saliency detection specific Hand-crafted representations are obtained
3. Per superpixel binary classification is done using shallow CNN

SLIC superpixel generation



$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

$$D' = \sqrt{(\frac{d_c}{N_c})^2 + (\frac{d_s}{N_s}^2)}$$

Superpixel representations

$$Q = \begin{bmatrix} q^c_{11} & \cdots & q^c_{1j} & \cdots & q^c_{1M} \\ \vdots & \ddots & & \ddots & \\ q^c_{x1} & \cdots & q^c_{xj} & \cdots & q^c_{xM} \\ \vdots & \ddots & & \ddots & \\ q^c_{N1} & \cdots & q^c_{Nj} & \cdots & q^c_{NM} \end{bmatrix}$$

$$q^c_{xj} = t(r_j).|C(r_x) - C(r_j)|.w(P(r_x), P(rj))$$

$$Q' = \begin{bmatrix} q^d_{11} & \cdots & q^d_{1j} & \cdots & q^d_{1M} \\ \vdots & \ddots & & \ddots & \\ q^d_{x1} & \cdots & q^d_{xj} & \cdots & q^d_{xM} \\ \vdots & \ddots & & \ddots & \\ q^d_{N1} & \cdots & q^d_{Nj} & \cdots & q^d_{NM} \end{bmatrix}$$

$$q^d_{xj} = t(r_j).|P(r_x) - P(r_j)|.w(C(r_x), C(rj))$$

Classification using shallow CNN

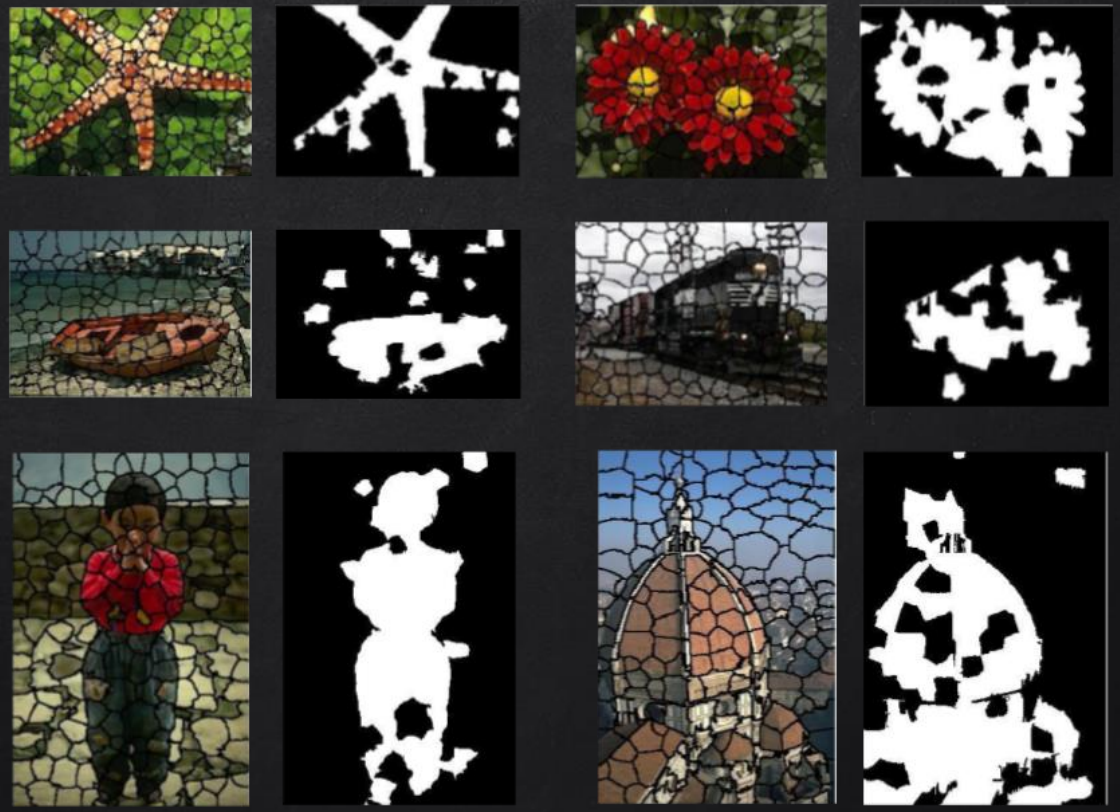
Type	Size/Stride	Number of channels
input	M	3
Conv1D, tanh	10×1	5
MaxPool	$2 \times 1/2 \times 1$	-
Conv1D, tanh	20×1	10
Maxpool	$2 \times 1/2 \times 1$	-
Conv1D, tanh	20×1	20
Maxpool	$2 \times 1/2 \times 1$	-
Global-avg-pool	-	-
Fully-con, tanh	20	-
Dropout (0.3)	-	-
Fully-con, Softmax	2	-

Table 1: CNN architecture used in this project. Note: size of Q matrix is M×M.

Total number of parameters: 5,233

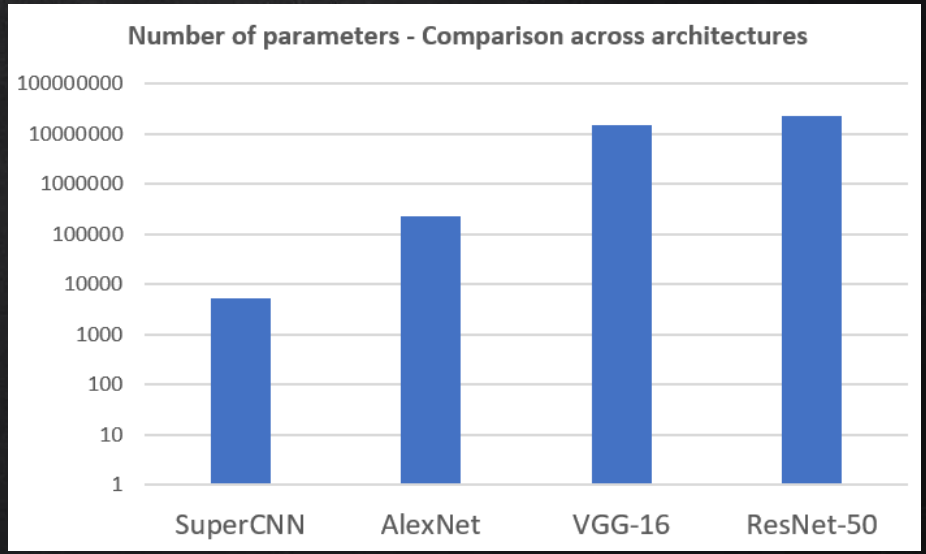
Total Number of parameters in conv layers of vgg-16: 14,714,688

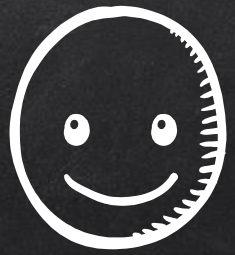
Results



Conclusions

1. Hand-crafted features capturing **global spatial context** at superpixel level along with CNN classifier shows decent performance for saliency detection in images.
2. Superpixel-CNN based approach does not require large training data and number of parameters in CNN are in **few thousands** compared to **10s of millions** of parameters in standard CNNs on raw image pixels.
3. Lower number of parameters reduces the computational cost **significantly** at both training and **test time**.





THANKS!

Shashank Tripathi
Yash Patel

shashanktripathi123@github
yash0307@github

Code is available here: **https://github.com/yash0307/SuperCNN**