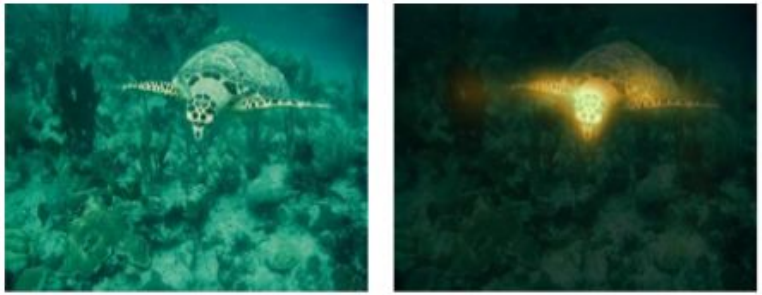
Deep Visual Attention Prediction

Akshay Kalkunte, Hardi Desai, Suraj Maniyar

Motivation

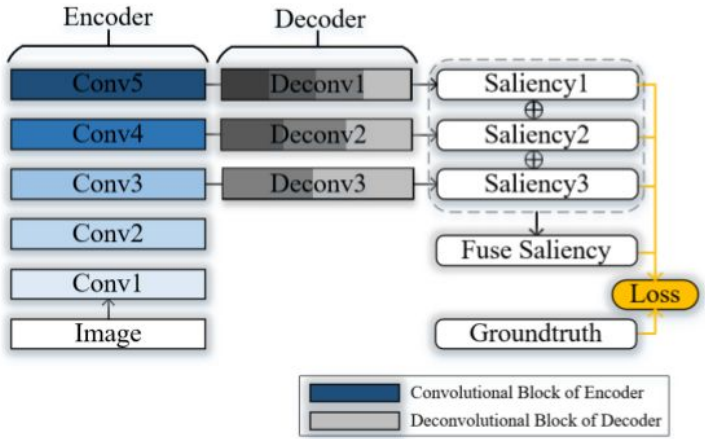
- Human Visual System has selective attention mechanism
- Predict Human Eye Fixations using Deep Learning
- Also referred as Visual Attention Prediction or Visual Saliency Detection
- Applications include Image Cropping, Object Recognition, Visual Tracking, Object Segmentation, Video Understanding, etc



(a) Image (b) Saliency Result

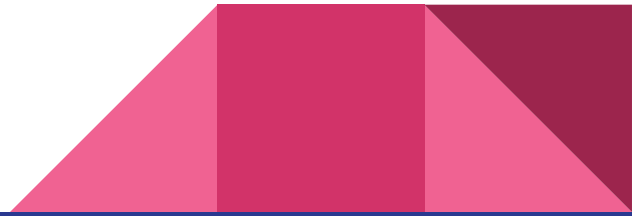
Model

- Based on skip layer architecture
- Incorporates multi-level saliency predictions
- Trainable network : works with an encoder-decoder architecture
  - Encoder: Identical to the first 5 convolutional layers in the VGG16 network
  - Decoder: Consists of deconvolutional layers
- Performs pixel wise prediction



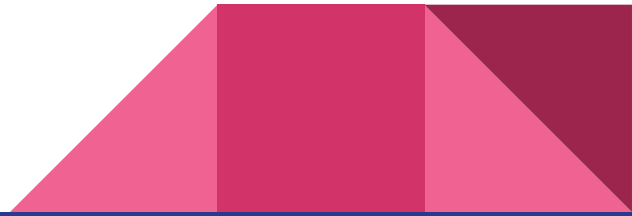
Model

- Encoder
  - Captures high-level features via convolving and downsampling the low- level feature map, which decreases the size of the feature maps from bottom to up
- Decoder
  - Upsamples feature maps, which constructs an output that maintains the original resolution of the input
  - Convolutional filters are learnable, which is preferable to the fixed interpolation kernel
  - Gradually reduces feature dimensions for higher computation efficiency

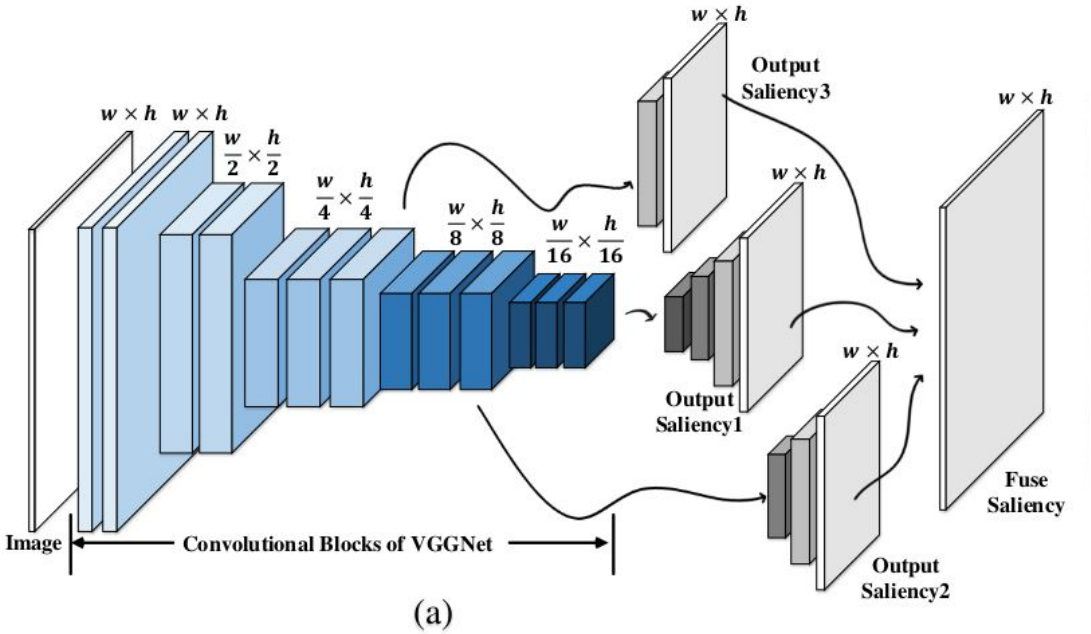


Model

- Saliency is best captured when features are considered from multiple scales.
- Hence, M layers from the encoder network are used for explicitly predicting saliency in multi-scales and multi-levels, where each selected layer is associated with a decoder network
- Stacking many convolution layers in encoder leads to gradually learning “local” to ”global” saliency information (i.e., responsive to increasingly larger region of pixel space).



Architecture

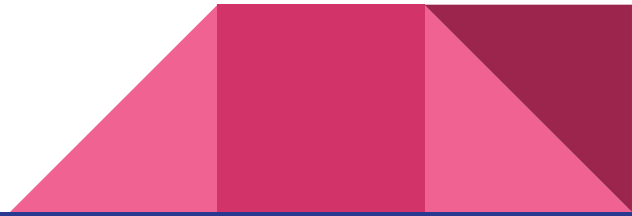


Implementation Detail

- Framework for implementation:
  - TensorFlow
- Initialization Weights
  - Five convolution blocks are initialized from the VGG16 trained on the ImageNet
  - Remaining layers are randomly initialized from a Gaussian distribution with zero mean and standard deviation of 0.01.
- Dataset:
  - SALICON consisting of 10,000 train images and 5,000 test images
  - Eye fixation annotations are simulated through mouse movements of users on blurred images.

Implementation Detail

- Hardware Used: Google Collab and GTX960
- Learning Rate: 10e-4
- Batch Size: 4
- Optimizer: Stochastic Gradient Descent



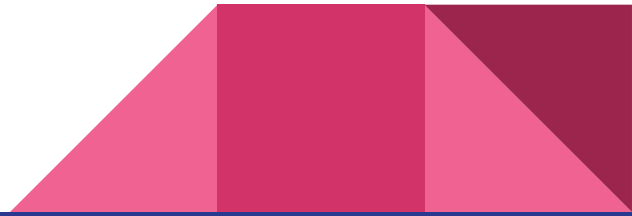
Issues

TensorFlow:

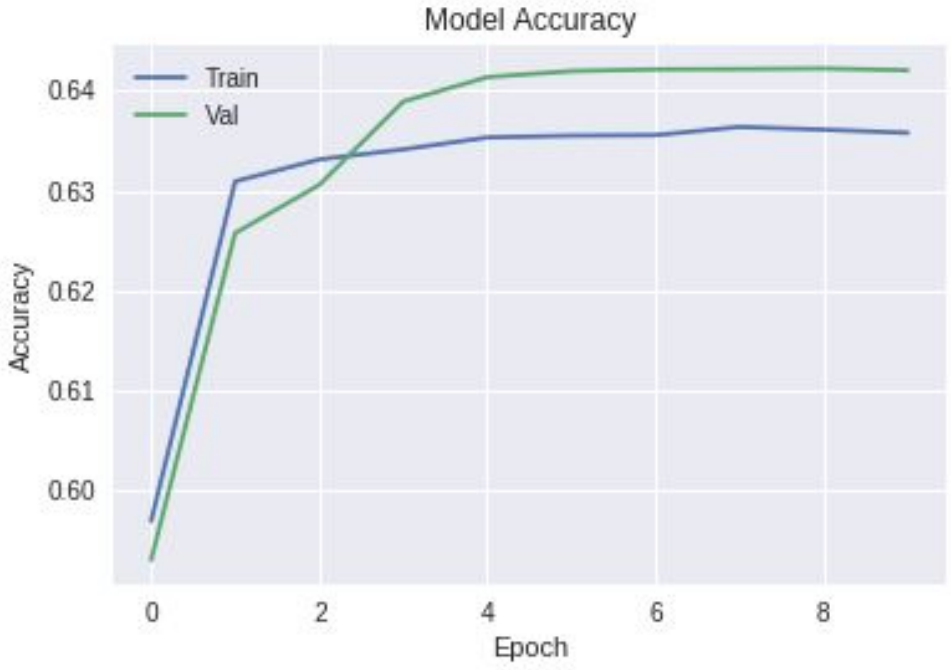
- Does not provide crop layer
- Padding is either ‘SAME’ or ‘VALID’
- Output Tensor Size in Deconvolutional layer needs to be explicitly specified else we have back propagation error

Caffe:

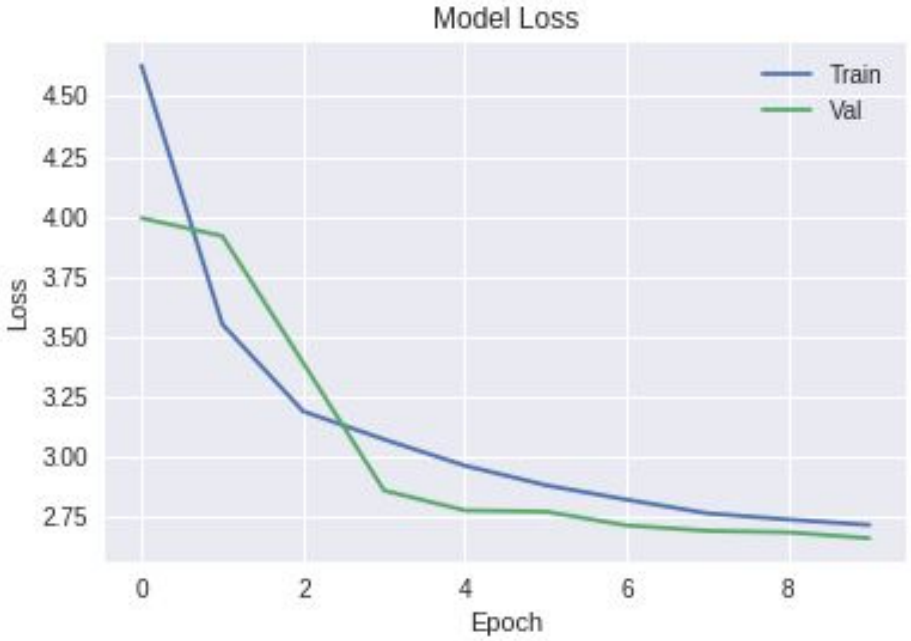
- Implementation consists of crop layer to fuse saliency
- Padding could be a definite number of pixels



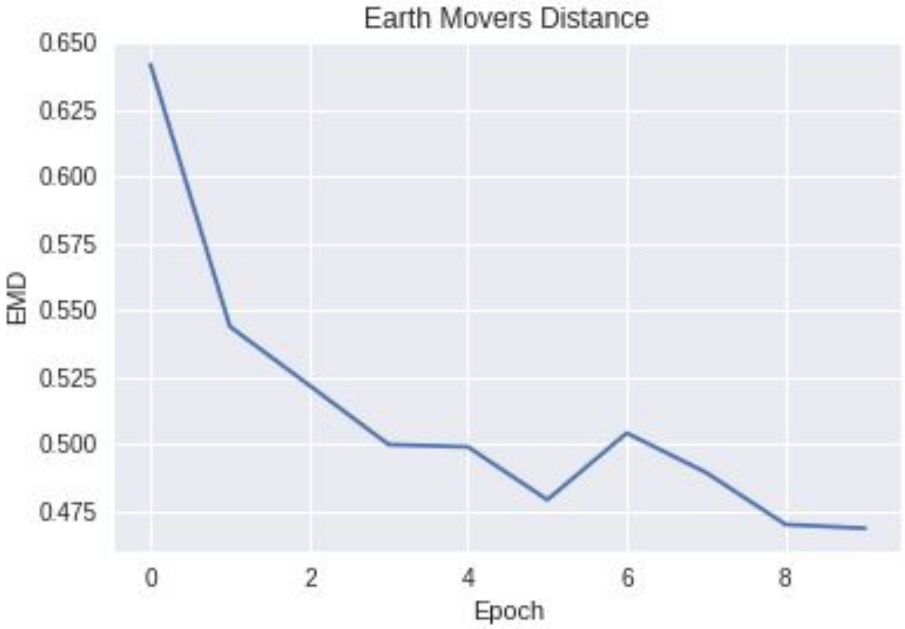
Results


Model Accuracy
Train
Val
0.64
0.63
0.62
0.61
0.60
Accuracy
0
2
4
6
8
Epoch

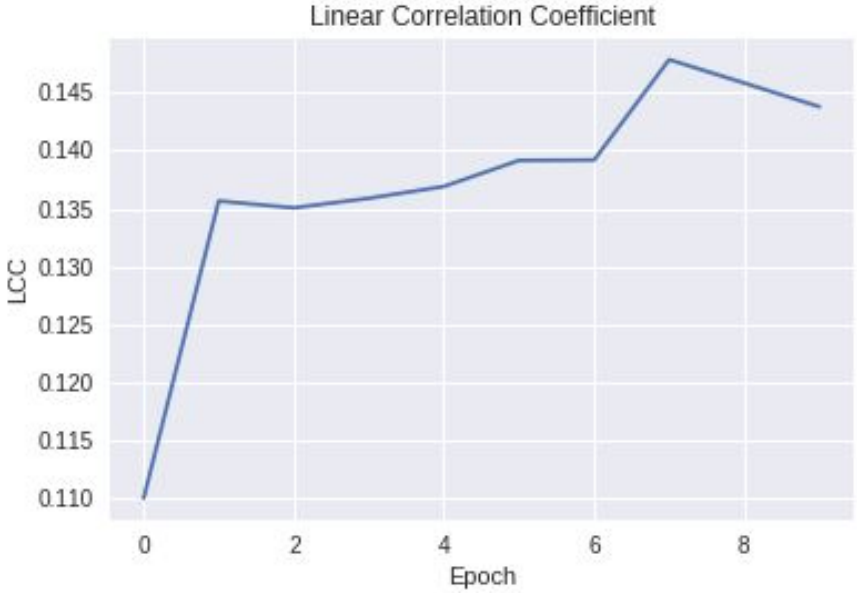
Results


Model Loss
Train
Val
Loss
4.50
4.25
4.00
3.75
3.50
3.25
3.00
2.75
0
2
4
6
8
Epoch

Results


Earth Movers Distance
EMD
0.650
0.625
0.600
0.575
0.550
0.525
0.500
0.475
0
2
4
6
8
Epoch

Results


Linear Correlation Coefficient
LCC
0.145
0.140
0.135
0.130
0.125
0.120
0.115
0.110
0
2
4
6
8
Epoch

Questions?