Implementation of Video Salient Object Detection via Fully Convolutional Networks

Nitika Aggarwal and Karan Bhatiya

Abstract—Computer Vision is a field of Computer Science that trains computers to elucidate and understand the visual world. Using digital images from cameras and videos and deep learning models, machines can accurately identify and classify objects in the same way that human vision does and then react to what they "see" which is termed as Image Segmentation, Object detection to name a few. These tasks are interesting but difficult as we have to deal with Object Detection in Dynamic Saliency Models as well. Object detection in Static Saliency models is comparatively bit easier than Object Detection in Dynamic Saliency models because while computing dynamic saliency maps, video saliency models need to consider both the spatial and the temporal characteristics of the scene. Having to detect the salient regions in video Wenguan Wang in his paper has proposed a deep learning model to efficiently detect salient regions in the videos. In the paper author mentions 2 important issues firstly, deep video saliency model training with the absence of sufficiently large and pixel-wise annotated video data; secondly, fast video saliency training and detection. Author proposed a deep video saliency network, for capturing the spatial and temporal saliency information which successfully learns both spatial and temporal saliency cues, thus producing accurate spatiotemporal saliency estimate.

Inspired by his work in the paper, we as part of our academic project for the Computer Vision (CS512) course at Illinois Institute of Technology, worked on implementing the deep learning model for detecting salient object

regions in videos which we are presenting in this paper.

*This work was done as part of the academic project for the course CS512 - Topics in Computer Vision at Illinois Institute of Technology.

Nitika Aggarwal is a Masters Student in Computer Science department at Illinois Institute of Technology, Chicago. naggarwal@hawk.iit.edu

Karan Bhatiya is a Masters Student in Computer Science department at Illinois Institute of Technology, Chicago. kbhatiya@hawk.iit.edu.

1. Introduction

Salient object detection is an essential step in many image analysis tasks as it not only identifies relevant parts of a visual scene but may also lessen computational complexity by filtering out immaterial segments of the scene. Salient object detection aims at uniformly highlighting the salient regions, which has shown benefit to a wide range of computer vision applications. Depending on their input saliency models can be further categorised as Static and Dynamic ones. In this project, we aim at detecting salient object regions in videos. Detecting saliency in videos is a much more demanding problem due to the complication in the detection and utilization of temporal and motion information.

2. Implementation & Algorithm Details

There are 2 major steps in Algorithm: -

Firstly, for the dynamic scenes, we will create convolutional neural networks for end to-end training and pixel-wise saliency prediction.

Secondly, we propose a novel training scheme based on synthetically generated video data, both static and dynamic saliency information are encoded into a unified deep learning model.

2.1 Deep Network for Video Saliency Detection

To give an overall structure, we will feed frames of the video directly to the neural network or we can directly feed the video for which we have written the program that will convert the video in to frames and then pass those frames in to the neural networks. Our model or network will successively give the saliency maps as output where brighter pixels indicate higher saliency values.

In our Project we have used the model given by the author, which is trained on MSRA10K and DUT-OMRON dataset by using Caffe Library. The MSRA10K dataset comprising of 10K images, is widely used for saliency detection and covers a large variety of image contents – natural scenes, animals, indoor, outdoor, etc. Most of the images have a single salient object while the DUT-OMRON dataset is one of the most challenging image saliency datasets and contains 5172 images with multiple objects and complex structures. This is how our model learns spatiotemporal saliency in general dynamic scenes.

We design our model by considering both static and dynamic saliency both of which contribute to video saliency, simultaneously considering both the spatial and temporal characteristics of the scene. After creation and training of the model, final task is to do prediction on testing datasets. So, in our project we report our performance on FBMS (Freiburg-Berkeley Motion Segmentation) and DAVIS (Densely Annotated VIdeo Segmentation).

2.2 Architecture Overview of Deep Networks for Static Saliency

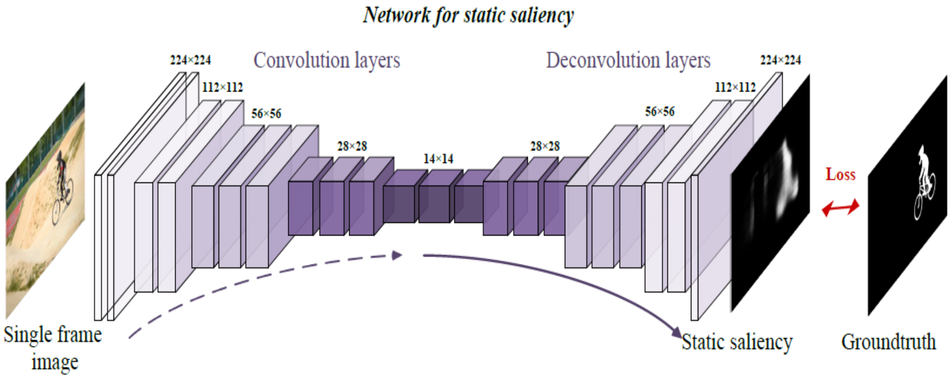


Fig1 Network for Static Saliency Detection

In the static saliency network, single frame image will be feed as a input and produce a saliency map as the same size of the input. This process is modelled with a fully convolutional network (FCN). In this we have used Convolutional layers on shared parameters (weight vector and bias) architecture and has translation invariance characteristics. The input and output of each convolutional layer are a set of arrays, called feature maps, with size h, w and c, where h, w and c are height, width and the dimensions of image respectively.

For the first convolutional layer, we have given the color image as input, with pixel size h = 224, w = 224 and three channels. Output will represent the feature map which indicates a particular feature representation extracted at all locations on the input, which is obtained via convolving the input feature map with a trainable linear filter (or kernel) and adding a trainable bias parameter.
If we denote the input feature map as X, whose convolution filters are determined by the kernel weights W and bias b, then the output feature map is obtained via:

$$f_s(X;W,b) = W *_s X + b,$$

where *s is the convolution operation with stride s. For improving feature representation capability, we apply point-wise nonlinearity (e.g., ReLU) after each convolutional layer. Moreover, convolutional layers are frequently followed by some form of non-linear down-sampling (e.g., max pooling). This results in robust feature representation which tolerates small variations in the location of input feature map. Due to the stride of convolutional and feature pooling layers, the output feature maps are coarse and reduced-resolution. However, for saliency detection, we are more interested in pixel-wise saliency prediction. For upsampling the coarse feature map, we put multi-layer deconvolution networks on the top of the convolution networks which can be represented by below given equation:

$$Y = D_S(F_S(I;\Theta_F);\Theta_D),$$

where I is our input image; FS(_) denotes the output feature map generated by the convolutional layers with total stride of S; DS(_) denotes the deconvolution layers that upsample the input by a factor of S to ensure the same spatial size of the output Y as that of the input image I.

Finally, on the top of the network, a convolutional layer with a kernel size of 1*1 is adopted for mapping the feature maps Y into a precise saliency prediction map P through a sigmoid activation unit. We use the sigmoid layer for pred so that each entry in the output has a real value in the range of 0 and 1. Due to the utilization of FCN, the network is allowed to operate on input images of arbitrary sizes, and preserves spatial information. For training, all the parameters Θs are learned via minimizing a loss function, which is computed as the errors between the probability map and the ground truth.

Given a training sample (I;G) consisting of an image I with size h*w*3, and groundtruth saliency map

$$G \in \{0,1\}^{h\times w},$$

the network produces saliency probability map

$$P \in [0,1]^{h\times w}$$

For any given training sample, the training loss on network prediction P is thus given by

$$\mathcal{L}(P,G) = -\sum_{i=1}^{h\times w} ((1-\alpha) g_i \log p_i + \alpha(1-g_i)\log(1-p_i$$

where $g_i \in G$ and $p_i \in P$; α refers to ratio of salient pixels in ground truth G. This has shown to substantially reduce training time and improve accuracy.

2.3 Architecture Overview of Deep Networks for Dynamic Saliency

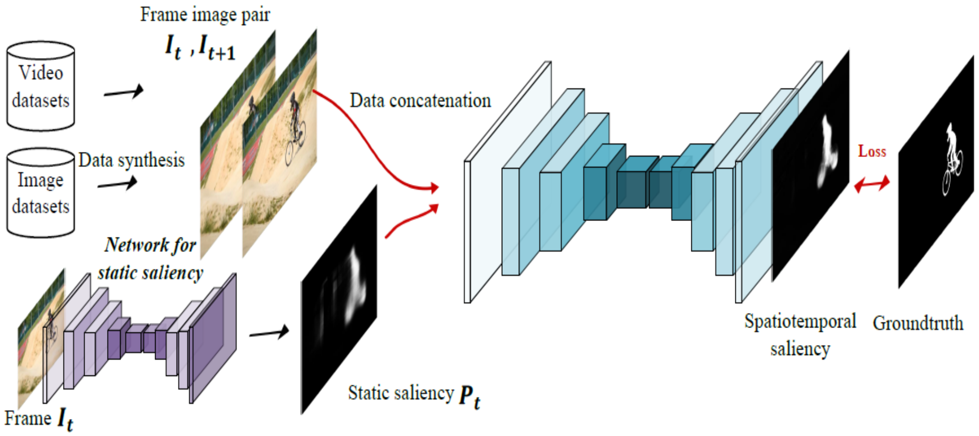


Fig 2 Network for Dynamic Saliency Detection

Network described in Fig 2 is Spatiotemporal saliency network which has a similar structure as our static saliency network, which is based on FCN and includes multi-layer convolution and deconvolution networks.

For capturing dynamic saliency, we feed successive pair of frames (It; It+1) and the groundtruth Gt of frame It in the training set into the network. Hence, we concatenate frame pair (It; It+1) and static saliency Pt in the channel direction, thus generating a tensor I with size of h *w*7. Then we feed I into our FCN based dynamic saliency network, which has similar architecture of static saliency network. Only difference is in the first convolution layer which is modified accordingly:

$$f(\mathbf{I}; W, b) = W_{I_t} * I_t + W_{I_{t+1}} * I_{t+1} + W_{P_t} * P_t + b,$$

where Ws represent corresponding convolution kernels; b is bias parameter.

During training, stochastic gradient descent (SGD) is employed to minimize the weighted cross-entropy loss. After the model is trained, given a frame image pair and static saliency prior, the deep dynamic saliency model will be able to output final spatiotemporal saliency estimate.

For testing, we first detect the static saliency map Pt for frame It via our static saliency network then frame image pair(It; It+1) and the static saliency map Pt are fed into the dynamic saliency network for generating the final spatiotemporal saliency for frame It. After obtaining the video saliency estimate for frame It, we keep iterating this process for the next frame Ik+1 until reaching the end of the video sequence. We have performed testing on DAVIS and FBMS datasets. This architecture brings two advantages. Firstly, the fusion of dynamic and static saliency is explicitly inserted into the dynamic saliency network, rather than training two-stream networks for spatial and temporal features and specially designing a fusion network for spatial and temporal feature integration. Secondly, the two adjacent frames instead of previous methods using optical flow images, thus our model gaining higher computation efficiency.

In our Project the proposed deep saliency model implemented by using two CNN network first is used for getting the static video saliency and second CNN we used for getting the result of dynamic video saliency network. We created both the CNN model and we used the weight of the model given by the author which are already trained on MSRA10K and DUT-OMRN using caffe library. We tried to train our implemented model but we are not able to run it properly. That’s why we used to the weights provided by the author. The only difference between the Static CNN model and Dynamic CNN model is that we apply multiple input and output for dynamic network. After this we normalized the output array by subtracting the mean value of RGB to get the better output. In our CNN model we used first 5 CNN network layer of vggnet model which is trained on ImagNet and then we have added our own CNN network layer for deconvolution to get the output image of size same as that of input.

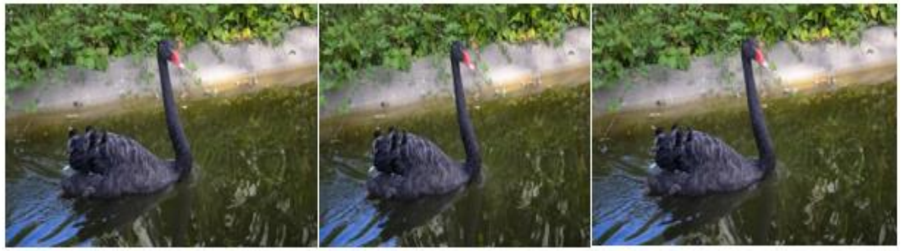
We have also implemented the feature of testing our video for dynamic salient object detection where you have to provide the path of the video. Then, the videoframe program will evaluate the video and make the frames of the video on 1fps. Then our program will apply the same procedure mentioned above to get the result. So, you can visualize the performance of our model on videos apart from dataset.

3. Experiments and Results

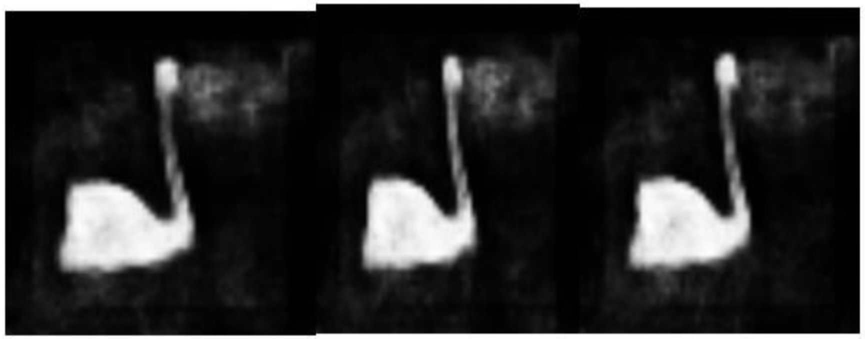
We have reported our performance on two public Datasets FBMS and DAVIS. The FBMS dataset contains 59 natural video sequences, covering various challenges such as large foreground and background appearance variation, significant shape deformation, and large camera motion. Another dataset is DAVIS dataset, which consists of 50 video sequences in total, and fully-annotated pixel-level segmentation ground-truth for each frame is available.

Results: -

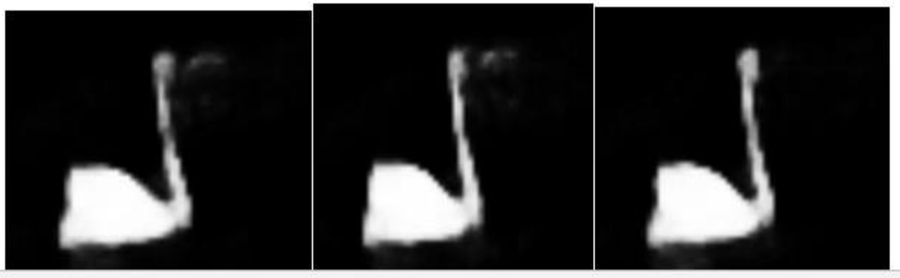
Original Image:

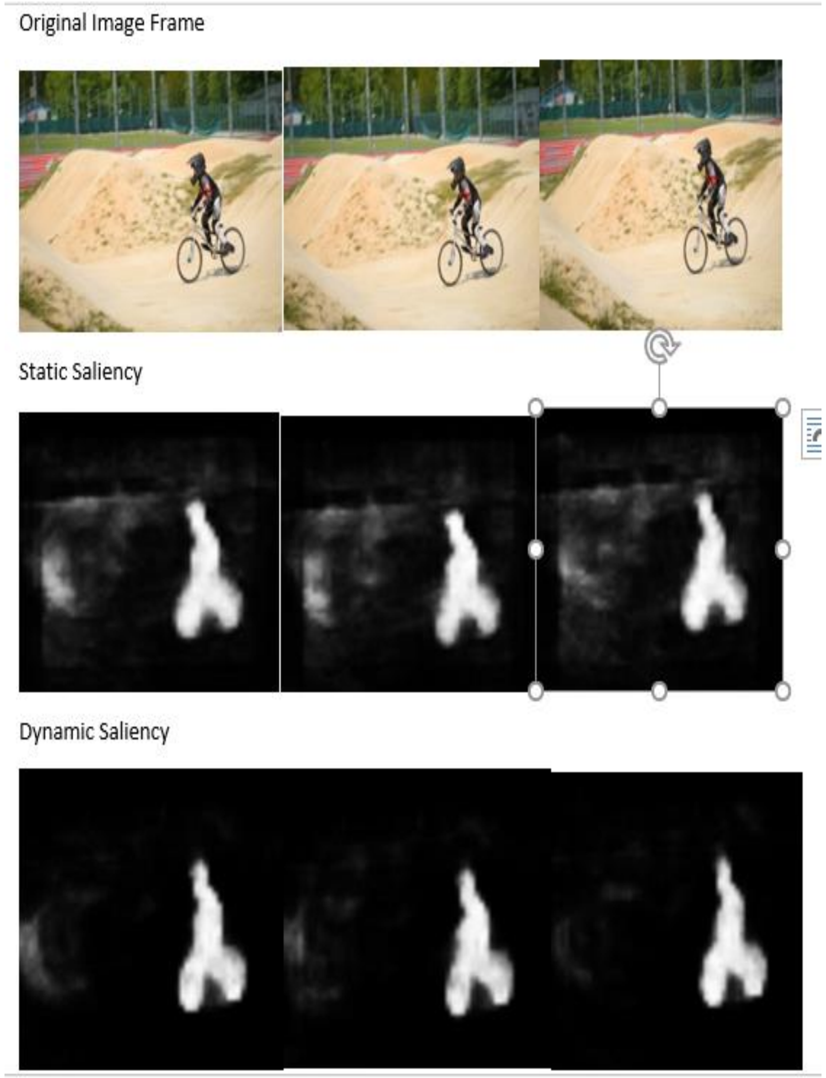


Static Salient



Dynamic Salient





4.Conclusion: -

By working on this project, we understood the Convolutional Neural Network very well and effectiveness of the CNN model in Object Detection and Recognition.

In this project, we have implemented deep learning method for salient object detection in videos using convolutional neural networks. For capturing spatial and temporal statistics of dynamic scenes our model has used 2 modules, namely static saliency network and dynamic saliency network. The groundtruth of the static saliency is assimilated in the dynamic saliency network, which instinctively fuse static saliency into dynamic saliency detection and give spatiotemporal saliency results with less computation load. Our model can generate high-quality salience maps which are confirmed by doing testing on two datasets, namely FBMS and DAVIS.

5. Refrences:-

1. https://arxiv.org/pdf/1702.00871.pdf
2. https://arxiv.org/pdf/1409.1556.pdf
3. https://davischallenge.org/

4. https://keras.io/models/model/
5. https://www.pyimagesearch.com/2018/07/16/opencv-saliency-detection/
6. https://lmb.informatik.unifreiburg.de/resources/datasets/moseg.en.html