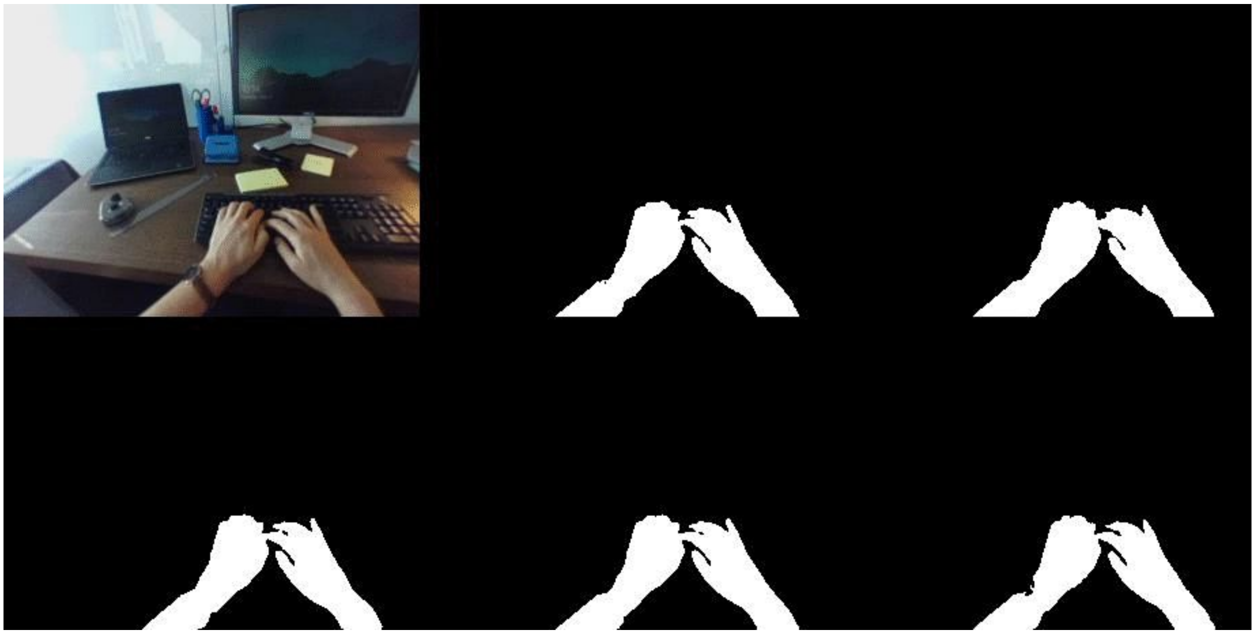
COMPUTER VISION

Hand Segmentation using RefineNet

Team Name : fun@images



Introduction

In this project, we try to analyze the problem of hand segmentation over both the indoor as well as videos taken from the surrounding. For the purpose of this several different datasets are used which include **Ego-Youtube hands** which contain egocentric videos with hands in the wild and **HandsOverFace** to analyze the performance of the Network while similar colors are present. The major motivation behind this is learning the configuration of the hands tells us a lot about we are planning to do or what we pay attention to. Also, hand segmentation will help us in the understanding of hand-object manipulation and hand-eye coordination.

Previous works which are done on the hand segmentation will generally be done on third party videos in which along with the hand the whole body is present, however in the egocentric dataset the body is not present therefore the previous algorithms/work are insufficient for this.

Different Datasets Used

Person-Parts Dataset

1. This is a subset of the Parts-dataset that is based on VOC 2010.
2. Around 24 different body parts are annotated.
3. Mostly third person view.
4. Corresponding GitHub link: http://www.stat.ucla.edu/~xianjie.chen/pascal_part_dataset/pascal_part.html

Ego-Hands Dataset

Requirement

1. 48 videos recorded with google glass.
2. 4 activities: playing puzzles, cards, jenga and chess.
3. Environment: Living room, courtyard, office.
4. 15000 hand instances.
5. 4800 ground truth in total.
6. Corresponding GitHub Link: http://vision.soic.indiana.edu/projects/egohands/

Limitations: This dataset only contains only indoor scenarios.

Ego_YouTubeHands Dataset

1. Overcoming the shortcoming of the previous dataset this is generated from the random videos taken from Youtube about 3-6 minutes long.
2. Every 5th frame is annotated, 1290 frames in total.
3. Total of 2600 hand instances. (1800 first person, 800-second person)
4. Corresponding GitHub Link: https://github.com/aurooj/Hand-Segmentation-in-the-Wild

Georgia Tech Egocentric Activity Dataset

1. Containing images from 7 daily activities performed by 4 subjects.
2. Total of 663 images with pixel level hand annotations.
3. 1231 1st person hand instances.
4. Corresponding GitHub Link: http://www.cbi.gatech.edu/fpv/

Limitations: Doesn't contain images with social interactions.

HandOverFace Dataset

1. 300 images from the web.
2. Faces are occluded by hands to show the effect of skin similarity on hand segmentation.
3. Contain images of different age, gender and ethnicites.
4. Corresponding GitHub Link: https://github.com/aurooj/Hand-Segmentation-in-the-Wild

Steps taken to get the corresponding dataset into the required format.

Input format: In the input, Matlab files are given corresponding to each image of the dataset stating the class value corresponding to each pixel.

Required format: We want the images to be stored in one folder and the corresponding ground truth in another folder with the same image name.

Steps: This is done by the use of scipy in which the mat files are firstly loaded and then they are converted into the corresponding segmentation Map which can be given as an input for the training of the RefineNet.

Working

For the purpose of the hand segmentation we use RefineNet for our purpose and the segmentation which we are targeting is we want to detect every pixel that belong to a hand. For this purpose we trained the RefineNet-Res101 with two classes either as hand and no-hand and later train it on EgoHands, EYTH, GTEA, and HOF datasets.

Refine-Net

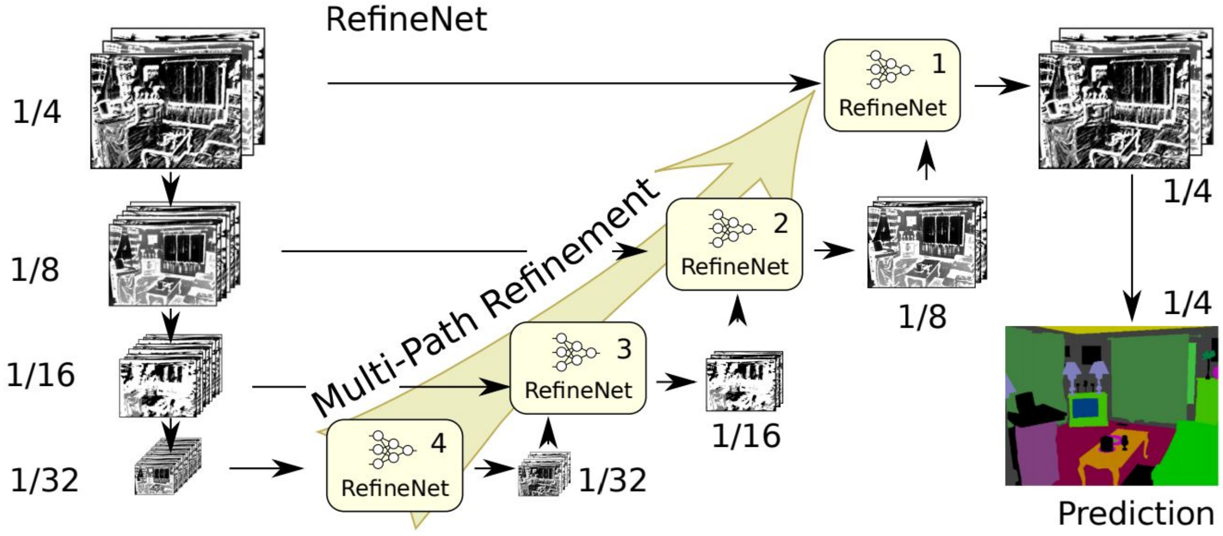
Refine-Net is the multipath refinement network that exploits all the information available along the down-sample process to enable high-resolution prediction along long-range residual connections.

The idea is since in general within the previously convolutional neural network the data is propagated from high resolution image map to low resolution map while passing through pooling layers around 32 times smaller than the original image. This lead to a loss of a lot of information, earlier ideas includes taking results from the intermediate layers, learning deconvolution or not decreasing down the size. However each of them comes with a number of problems which include increasing computations, need of more GPU memory or coarse sub-sampling of features which lead to a loss of information.

However in the Refine it goes on the principle that each of the output of the layer is essential and therefore tries to combine the results of all these layers.

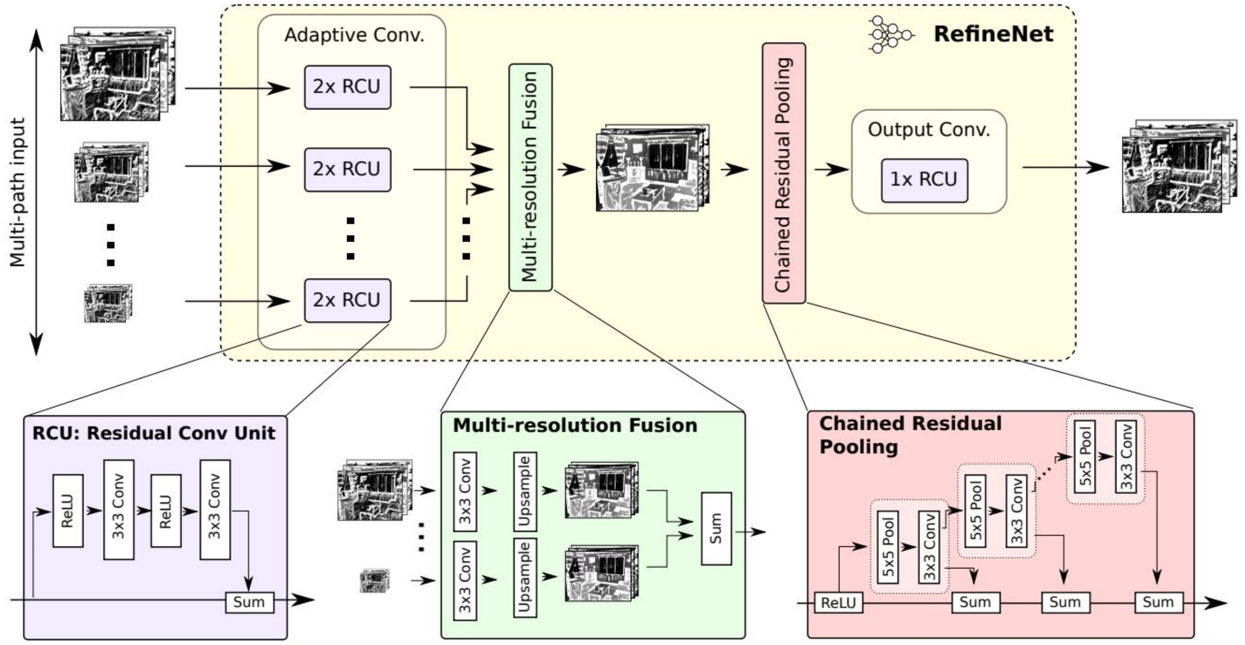
The major contributions include:

1. **Multi-level refinement network**, exploits features at multiple level for high-resolution image segmentation.



2. Effective end-to-end training through **Residual connections** with identity mapping by directly propagating through short-range and long-range residual connections.

3. **Chained Residual Pooling** able to capture background context by effectively pooling features with multiple window size and pooling them together with residual connections and learnable weights.



Modifications in ResNet

ResNet basically has a single label prediction layer, so in order to use it in the RefineNet that is been replaced by dense prediction layers, outputting the confidence of each class at each pixel. However the ResNet module which is their can be easily replaced by any other convolution module, which is one of the benefit of the RefineNet.

Exploring the architecture of RefineNet

Multi-Path RefineNet:

There are two main function of the multi-path refineNet, this includes:

1. Subsampling the image and pass each of the subsampled image into individual block of the Refine-Net to generate the corresponding output, these output is combined with the next subsample RCU output in order to

tune the results and the resulting is then given as an input to the next RCU module forming a cascade structure and this process keeps on repeating.

2. Training gradients are able to effortlessly propagate backward through network to all the way to early stages of the network by the use of the RefineNet blocks and even during the forward pass where these are used.

RCU : Each of the input is passed through 2*RCU which is the simplified version of the original ResNet done with removal of the batch-Normalization layer.

Multi-Resolution Fusion: This includes the convolution and upscaling of the input which is of lower size equivalent to that of the upper and then both are then attached using summation.

Chained Pooling: The proposed chained residual pooling aims to capture background context from a large image region. this component is built as a chain of multiple pooling blocks, each consisting of one max-pooling layer and one convolution layer.

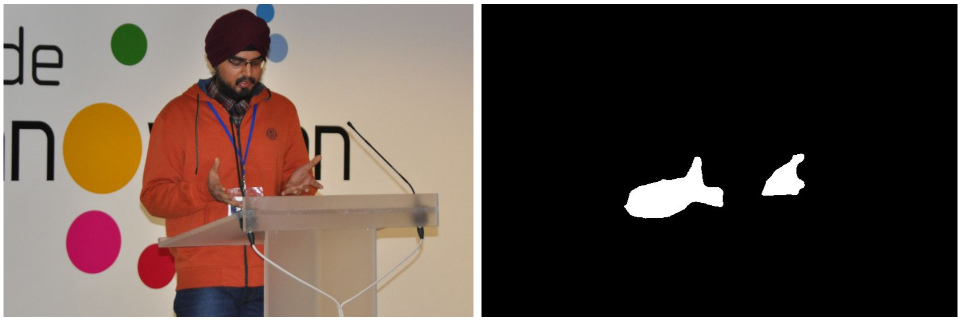
Output Convolution: The final step of each RefineNet block is another residual convolution unit (RCU). This results in a sequence of three RCUs between each block with 2 RCU before final softmax prediction step.

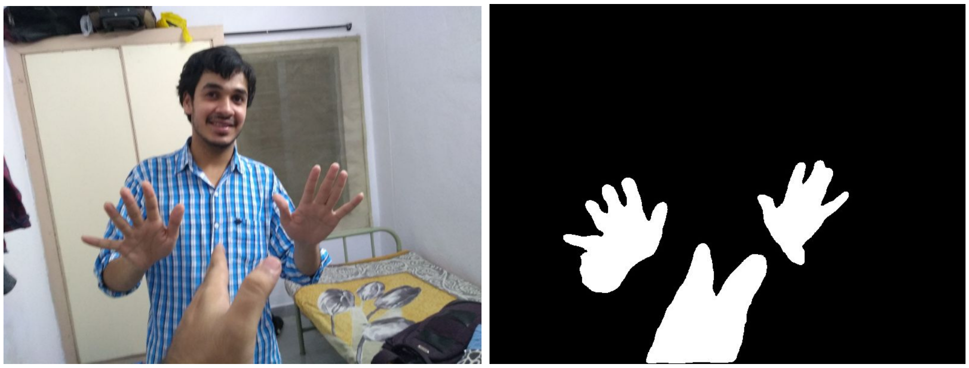
Short Term and Long Term Connections:

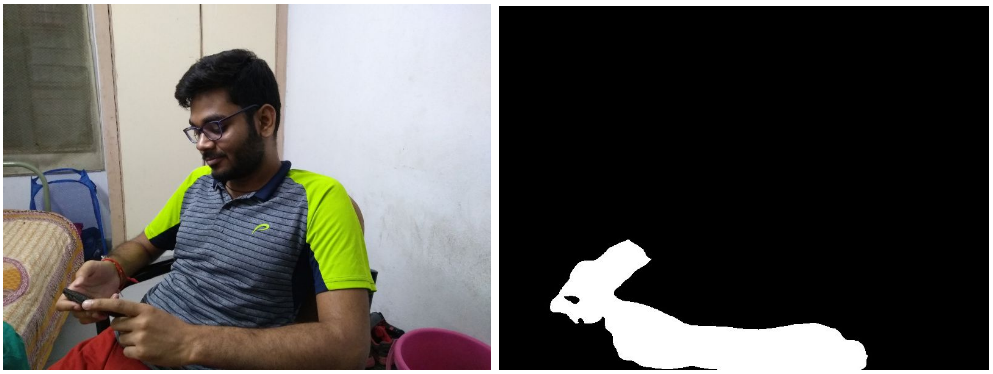
Short Term Connections are present within the Residual Convolution while the Long connections are present within the RefineNet Blocks.

Experiments

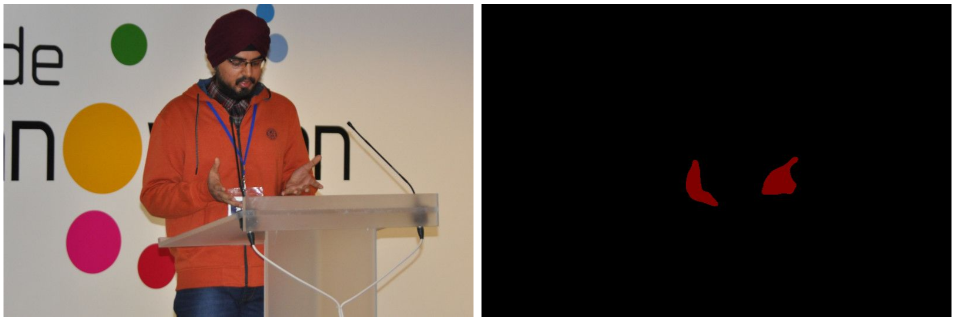
(Real Images) Trained over the Person Part Dataset

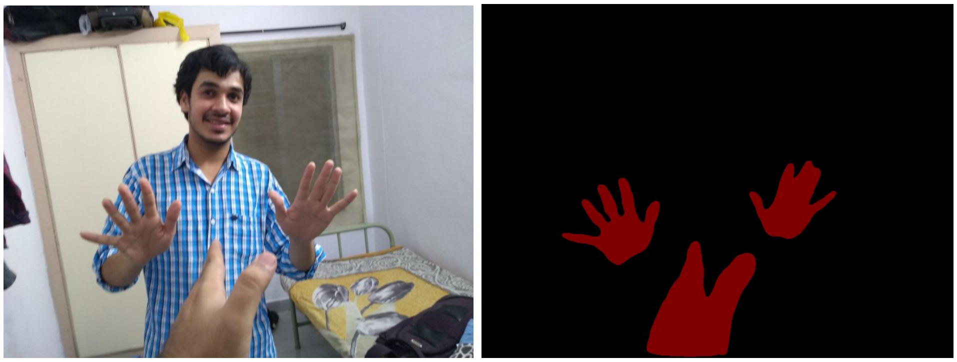


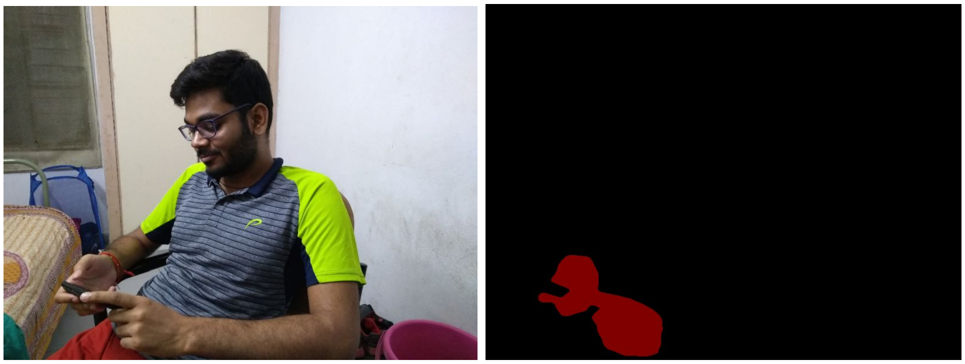




Results Tuned over EgoHands Dataset

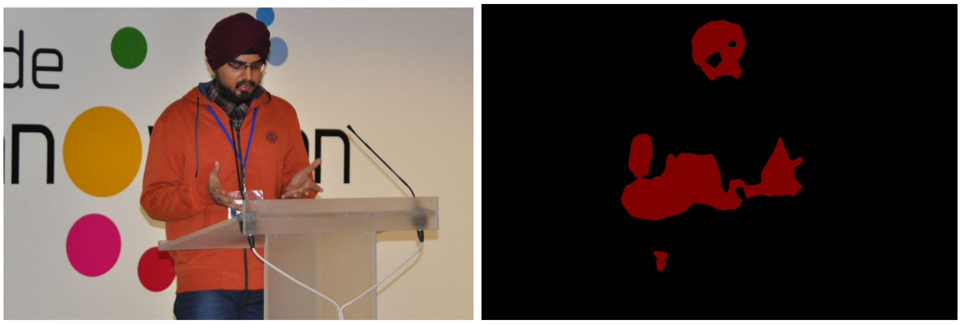


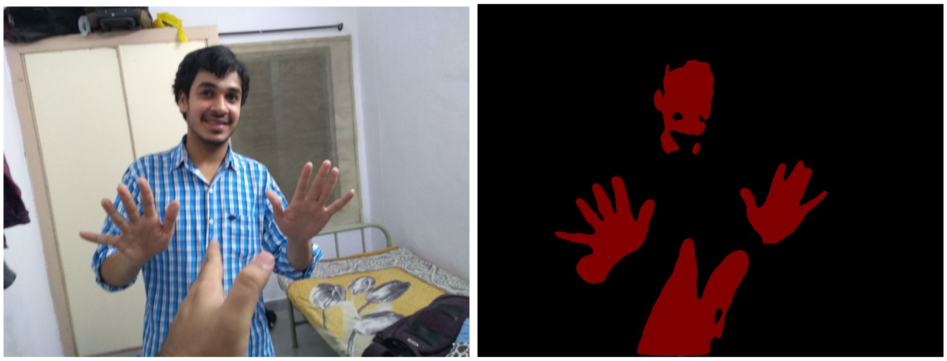


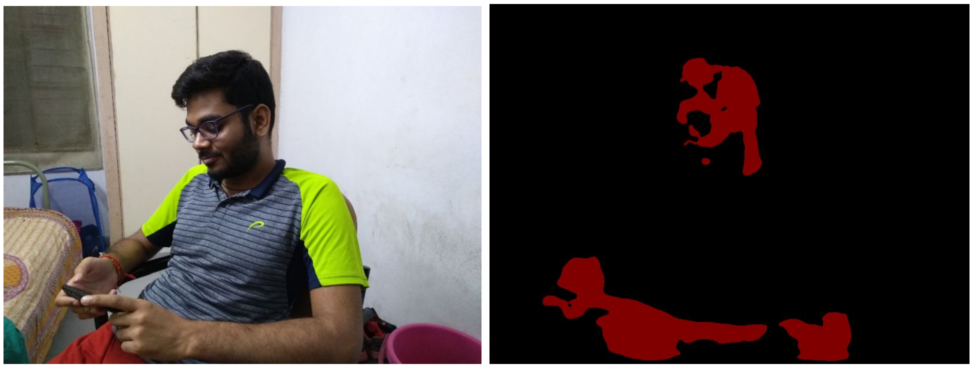


EgoHands: 0.662

Results Fine-Tuned over the GTEA

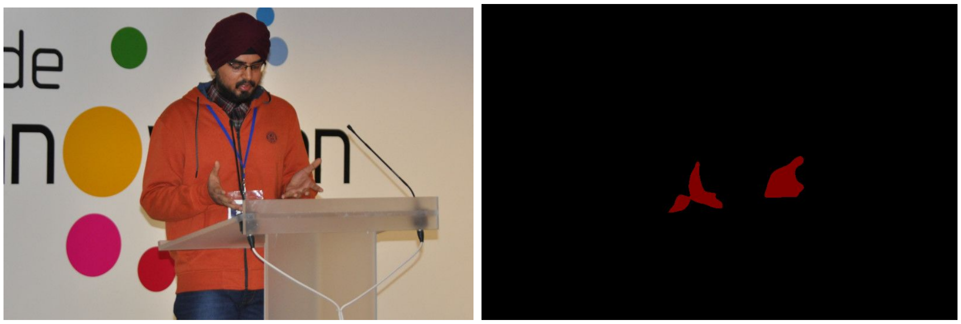


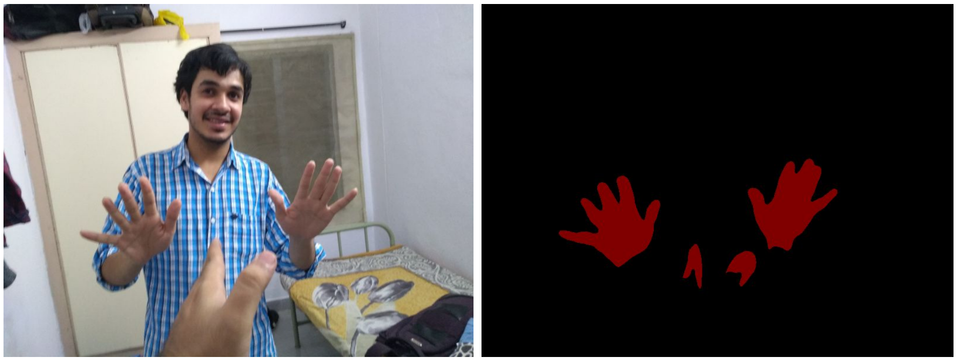


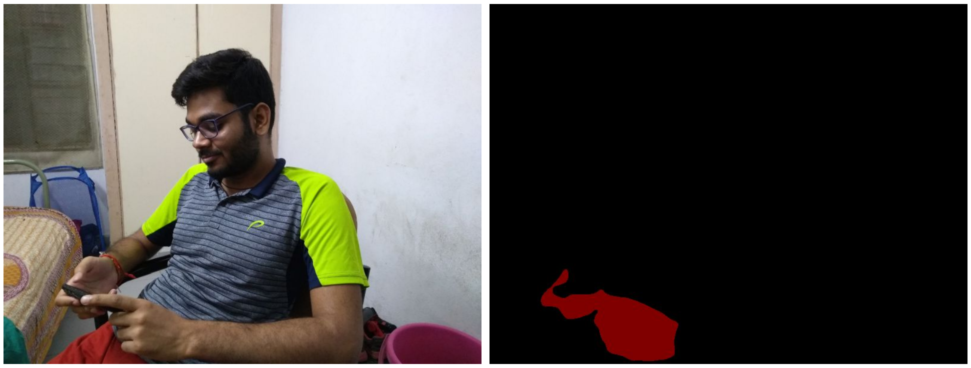


GTEA: 0.637

Results Fine-Tuned over the HOF

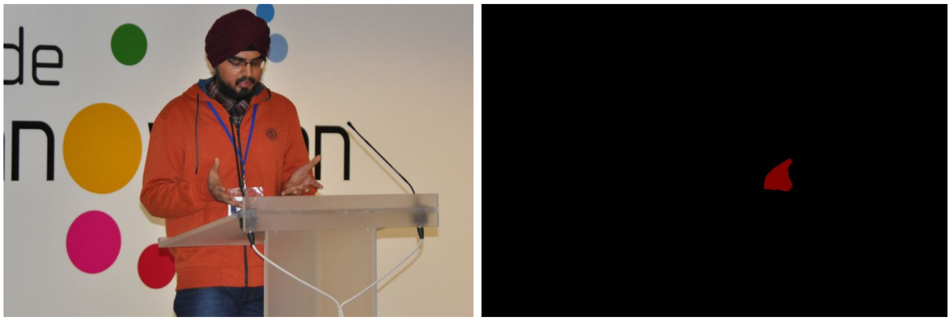


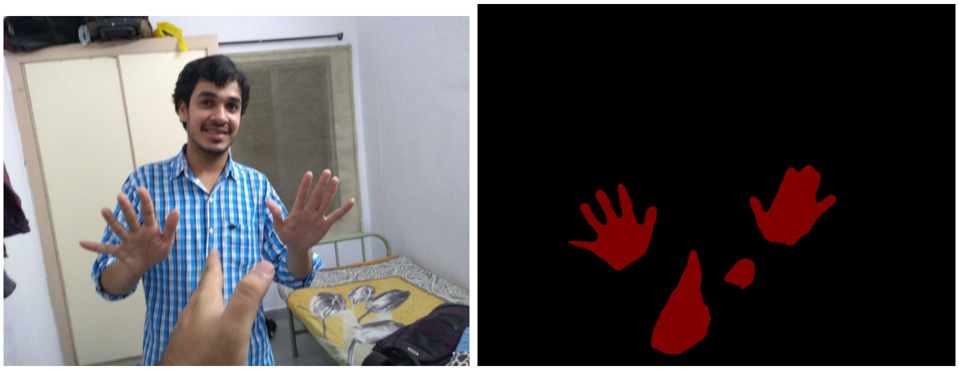


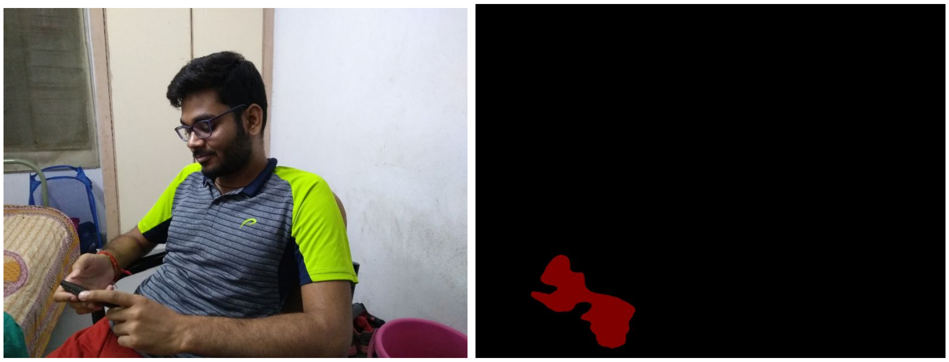


HOF: 0.623

Results Fine-Tuned over the EgoYouTube Datasets





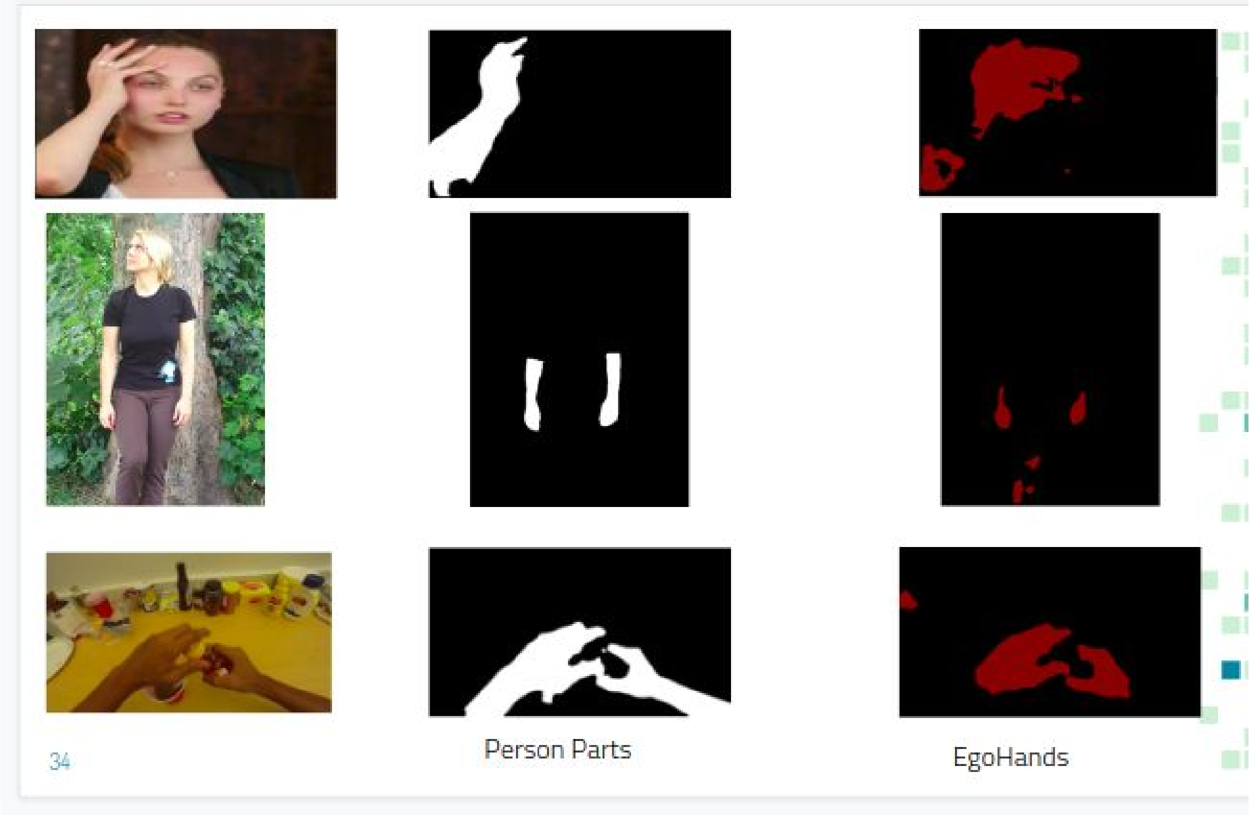


EYTH: 0.468

Experiments over Random Images

Observation

As it is seen in the case of the 1st image the **women is 3rd person**, therefore on fine tuning the above **model over the EgoHands will give us bad results as compared to the one trained on the person parts**, which is same in the case of the 2nd image. However in the third image since we are getting the **frame over the 1st person** in the image, therefore model fine tuned over the EgoHands is able to **give good results**.



In the first image which is an example of **hands over face** we see that the one which is fine tuned over the **HandonFace Dataset gives us best results** as compared to the other 4 trained model. However in general, on **random images** the model which is fine tuned over the **EgoYouTube Hands dataset works best**.