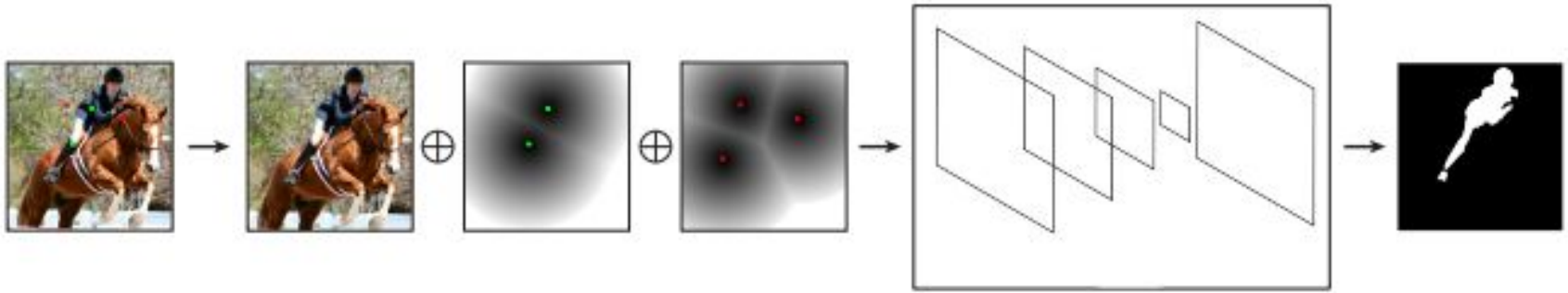
Interactive Object Segmentation

Previous Architecture



Given an input image and user interactions, our algorithm first transforms positive and negative clicks (denoted as green dots and red crosses respectively) into two separate channels, which are then concatenated (denoted as ⊕) with the image's RGB channels to compose an input pair to the DeepMask model. The corresponding output is the ground truth mask of the selected object.

Problem Statement

Leverage Semantic Segmentation to obtain better results on interactive instance segmentation which in turn will reduce the number of user clicks required.

Note:

Instance segmentation can be viewed as a more complex form of semantic segmentation, since we are not only required to label the object class of each pixel, but also its instance identity.

Semantic segmentation model

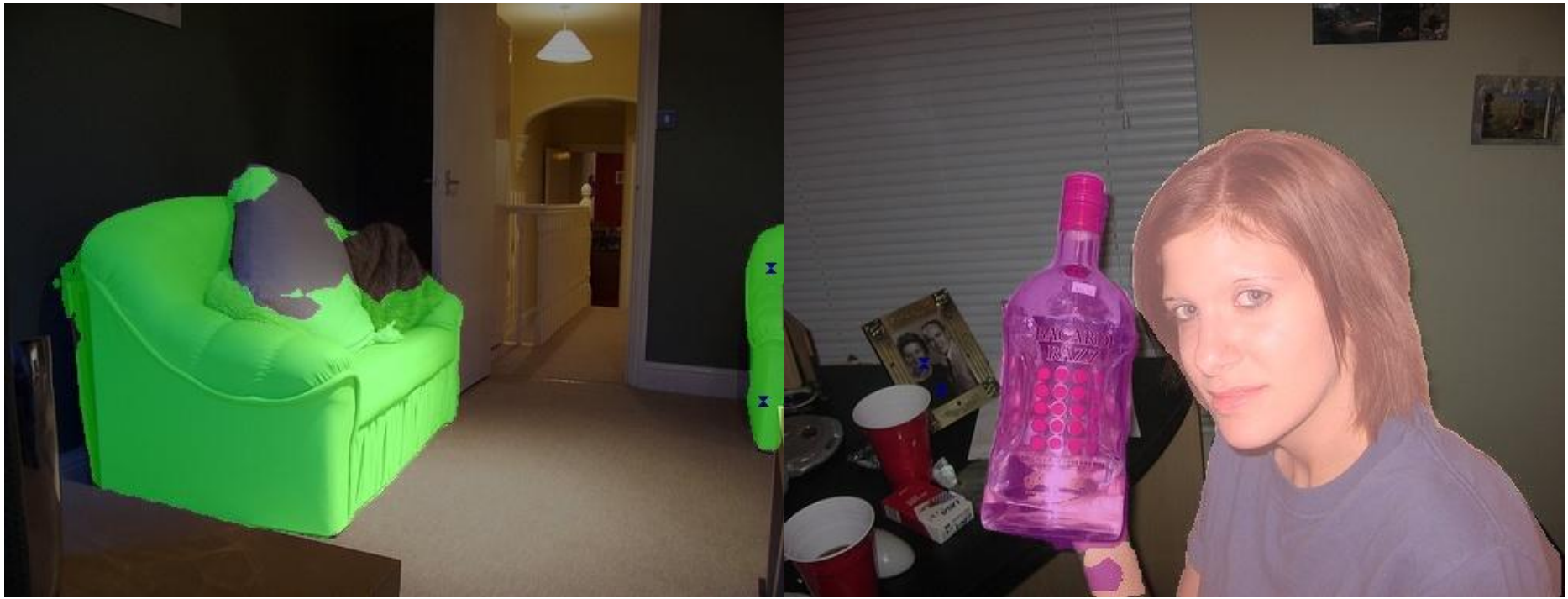
- Using DeepLabv3+ for obtaining the semantic segmentation results.
- Utilizing probability maps instead of using the final model prediction (argmax of logits).
- No of channels in the image depends on the no of classes in the dataset. For PASCAL VOC, it is 21
- Mean IoU on PASCAL VOC dataset: 84.56
- Crop size: 513 X 513
- Perform slicing of the image to the original size.

Modifications to the sample clicks script

- Modified the sample clicks script to sample points away from the object boundary. [1]
- More close to how a user will interact with the image.
- Label maps should not be blank.[2]
- Excluded the IGNORE_LABELS (255)

Segmentation and sample click overlay

Segmentation and sample click overlay



Channels selected based on user clicks

- Assuming at least one positive click on the image.
- Exclude background label if there is at least one click on some other object. (in some cases clicks are still near the object boundary)[1].

Steps:

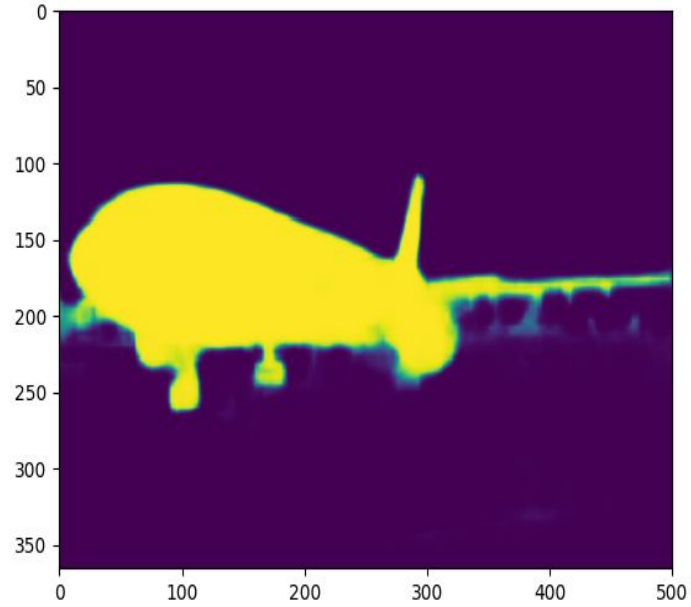
- Finds out the location of the positive clicks.
- Performs voting on these pixel locations.
- Selects the channel map with the highest probability for these pixel location.
- Sends the channel map as an additional input to the network.

Training Parameters:

- Batch size = 7
- Number of epochs = 10
- Learning rate = 1e-6 [1]
- Momentum = 0.9
- Weight decay = 2.5e-5
- Trained using Using Gradient Descent with Momentum
- Loss: Pixel-wise cross-entropy loss

Mean IoU Score on test dataset: 0.745 (Previously, 0.724 for the old model)

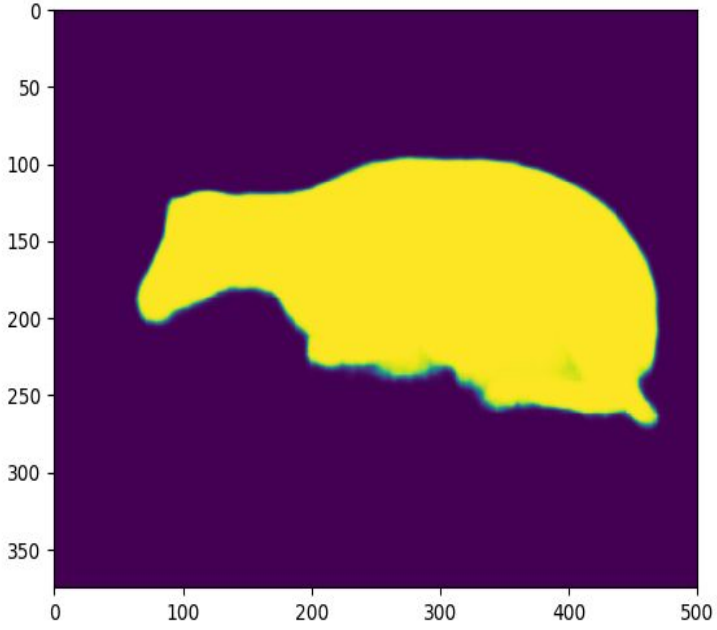
Semantic overlay result



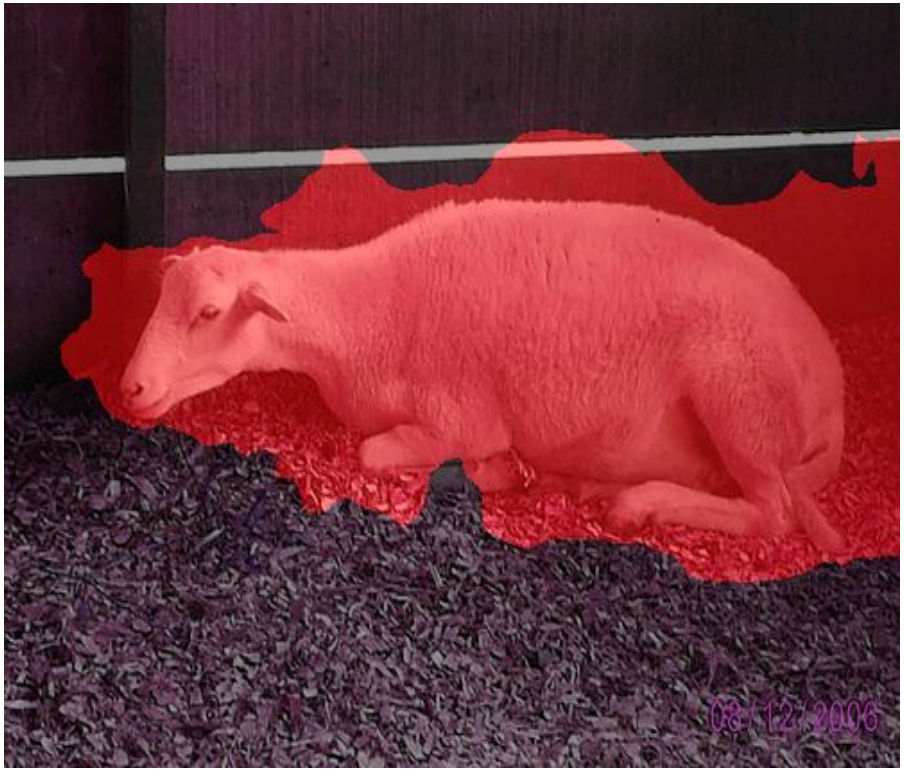
Results

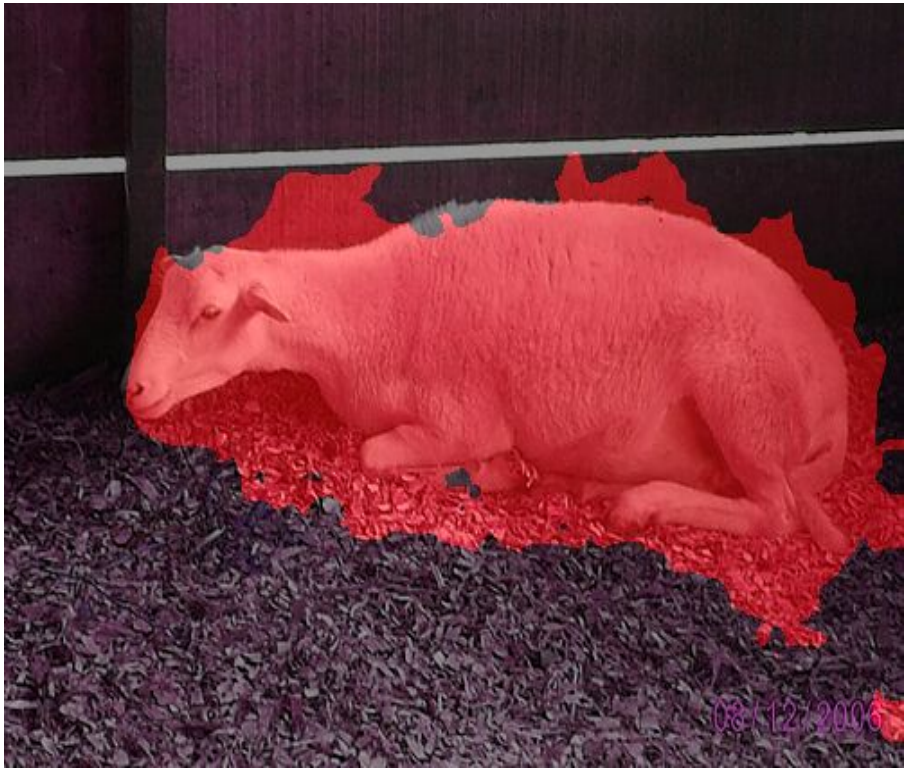
Baseline (Left) Mean Iou: 0.724 vs New Model (Right) Mean IoU: 0.801

Semantic overlay result



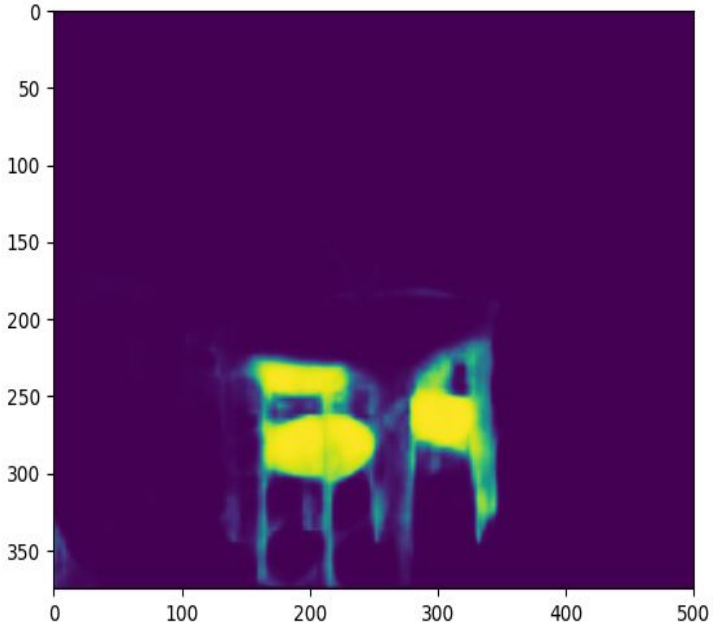
Results





Baseline (Left) Mlou: 0.689 vs New Model (Right) MIoU: 0.743

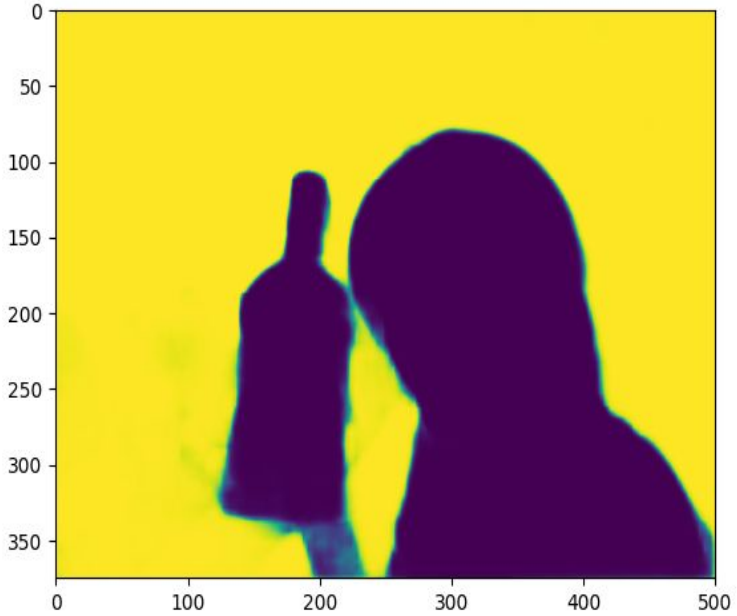
Semantic overlay result



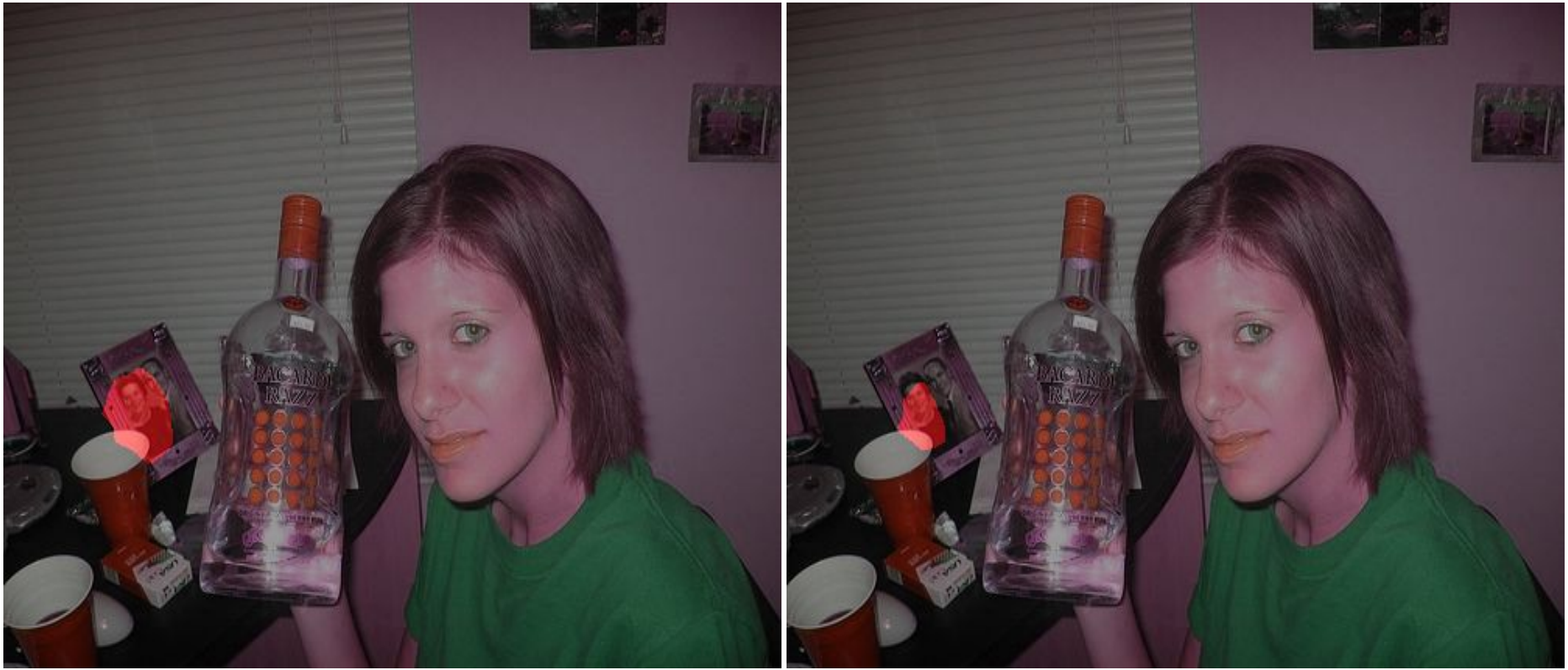
Results

Baseline (Left) Mean Iou: 0.656 vs New Model (Right) Mean IoU: 0.630

Semantic overlay result



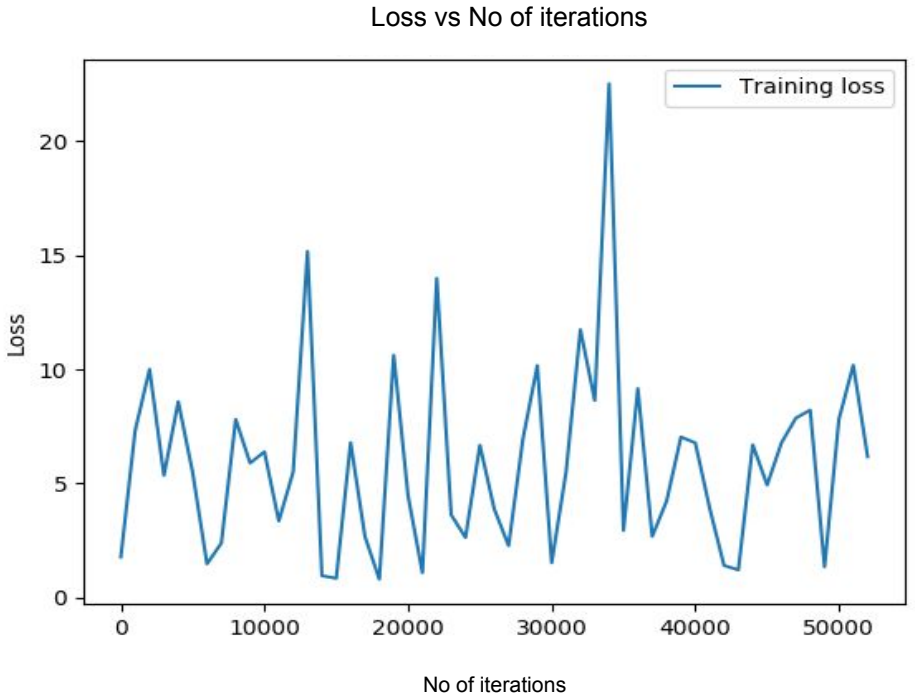
Results



Baseline (Left) Mean Iou: 0.702 vs New Model (Right) Mean IoU: 0.723

Problems

Oscillating loss while training.



Some solutions to look at:

- Reduce the learning rate
- Train for longer iterations
- Modify the channel selection process
- Reshuffle the dataset
- Augment the dataset
- Measure the improvement along the object boundaries.[1]
- Possible change in the model architecture - Do an initial prediction using the clicks and input image. Then use this output along with the semantic segmentation result to obtain better result.

Thank you!