PROGRESS REPORT

Philosophy: Instead of developing a deep learning model which performs image segmentation, approach the problem from a different angle.

Instead of just finding some model or architecture and doing some hyper parameter tuning to complete the task (This can be done easily but doesn't solve the root problem**), I chose to find the root cause of the specific problem and attack on that so that the problem arise from the task is of some actual use to the fynd team.**

The problem of Image segmentation can be thought from a very different direction:

Use the present pix2pix model to generate a segmented image and then use GAN to produce the final image as we need. This way we can use some state of the artpix2pix models to work on our network and we just need to develop a GAN according to our needs.

Example: If hairs are not correct then we can make a GAN which stylize hairs getting a more profound segmented output easily.

Approach:

Find root cause -> Explore Literature (find inspiration) -> check for some pre trained model that can act as base -> Implement pretrained model or develop own model -> check performance -> repeat.

Problem in the background removal:

Although pix2pix loss-based model works well but they lack in segmenting out minute details like hairs of the model or jewels etc.

The poor performance of the model can be attributed to the following reasons:

1. Input size(resolution) of the deep learning models is quite low like (256 X 256) by reducing high resolution model image (HD images) to such low resolution intrinsically remove much details as it gets lost in with the background pixel.
2. The dataset that we use also plays a very important role as model's best performance is rarely better than the ground truth, thus ground truth images should also be perfect, more importantly they should have hairs (or the minute things that we want in the final output).

Addressing problem one i.e. deep learning model

1. We can slice the image into several images whose size is same as the ideal input size of the model so that minute details are not lost in down sampling the image to the model size.

  1.1. After removing out background from each image we can sew the image together.

2. There is roughness in the model output because the colors at the boundary of the are quite similar thus pix2pix loss fails to distinguish.

  2.1. My further exploration revealed that in majority of images there exists some difference among different channels (R, G and B). So, if we apply same model across different channels then the overall performance can be increased.

  2.2. If the colors are same across all the channels then we can apply cross masks like R channel of image and G channel of the mask.

Addressing the problem of data set

1. To have a good quality of dataset, have to automate the process of creating masks as the cost of designating a human to perform the task is very slow.

  1.1. So, to **generate** the mask, I got the inspiration from the word "generate" to use GAN networks develop masks.

  1.2. One more method that can be used is a certain combination of Auto Encoder and Decoder as in the literature also these are known to be performing well in generating some images are more consistent than GAN.

  1.3. As suggested by Gupta el. al we can use super pixel technique to generate better trimap of the image.[1]

Exploring, generating better masks for the deep learning models.

For creating the best masks, I found a very interesting dataset at http://alphamatting.com.

Process 1 Auto Encoder and Decoder

Constructed Auto encoder and decoder network to generate masks for the image provided.

Details:

1. Channels of the model of Encoder was increased in the subsequent layers to preserve the information in the propagation during down sampling.
2. RMSprop was used as optimizer as Adams and Adadelta are found to overshoot the Global minimum due their adaptive momentum and fails to converge the model. SGD was found to stuck in a local minimum.
3. Mean square error was used as loss function because it can more optimally decide the difference in the constructed image from the network

Process 2: Matting Generation

1. The main benefit of the matting is that it takes transparency into account for image segmentation.(This data is generated by the scripts in the Data generation folder)
   1.1. It is based on the fact/assumption that objects in the foreground are not 100% opaque.
   1.2. The mask generated by matting ha three regions:
      1.2.1. Black: 100% background
      1.2.2. White: 100% foreground
      1.2.3. Grey: uncertain region

Exploring Deep Learning Model:

1. **U-Nets & GAN based segmentation**
   1.1. I choose U-Net based implementation because of its success Carvana challenge (Iglovikov and Shvets 2018)
   1.2. The use of this type of convolutional network allows to identify both local features and global features in the image and use these in addition to the pixel’s color in order to estimate said pixel’s matte
   1.3. Following the literature on Image reconstruction Iizuka, Simo-Serra and Ishikawa (2017) of filling holes in the image. They adapted a GAN to add a second discriminative model, one for local information and one for global information. That way they were able to train a generative model to fill holes in image realistically based on both local and global information.
   1.4. U-Net architecture as a generative network to estimate image foreground and a deep convolutional network is used as a discriminator during training.

2. **Fast R-CNN model**
   - **2.1.** Now we have better masks of the images we have so we can use Mask R-CNN, which is based on top of Faster R-CNN. Faster R-CNN is a model that predicts both bounding boxes and class scores for potential objects in the image.
   - **2.2.** Mask R-CNN adds an extra branch into Faster R-CNN, which also predicts segmentation masks for each instance.
   - **2.3.** The model started on a base of pre trained model on COCO data set and was fine tuned for segmentation.
   - **2.4.** SGD was used as optimizer with momentum 0.0 and l5 0.005.
     - **2.4.1.** As the dataset was small thus I decreased the learning rate and was also the main motivator of using SGD.
     - **2.4.2.** Adams and Adadelta were not converging on the dataset maybe because adaptive momentum.
   - **2.5.** The implantation was implemented on Penn Fudan Dataset and it was giving good results.

Datasets:

1. **Developed dataset specially for the task:**
   - **1.1.** Scraped Images with transparent background.
   - **1.2.** Scraped patterned background
   - **1.3.** Combine them to make input image and also make mask of the image for training.
2. Penn Fudan Pedestrian dataset.

References:

[1] Gupta, Vikas & Raman, Shanmuganathan. (2017). Automatic Trimap Generation for Image Matting.