Machine Learning Engineer Nanodegree

Capstone Proposal: Garment Segmentation

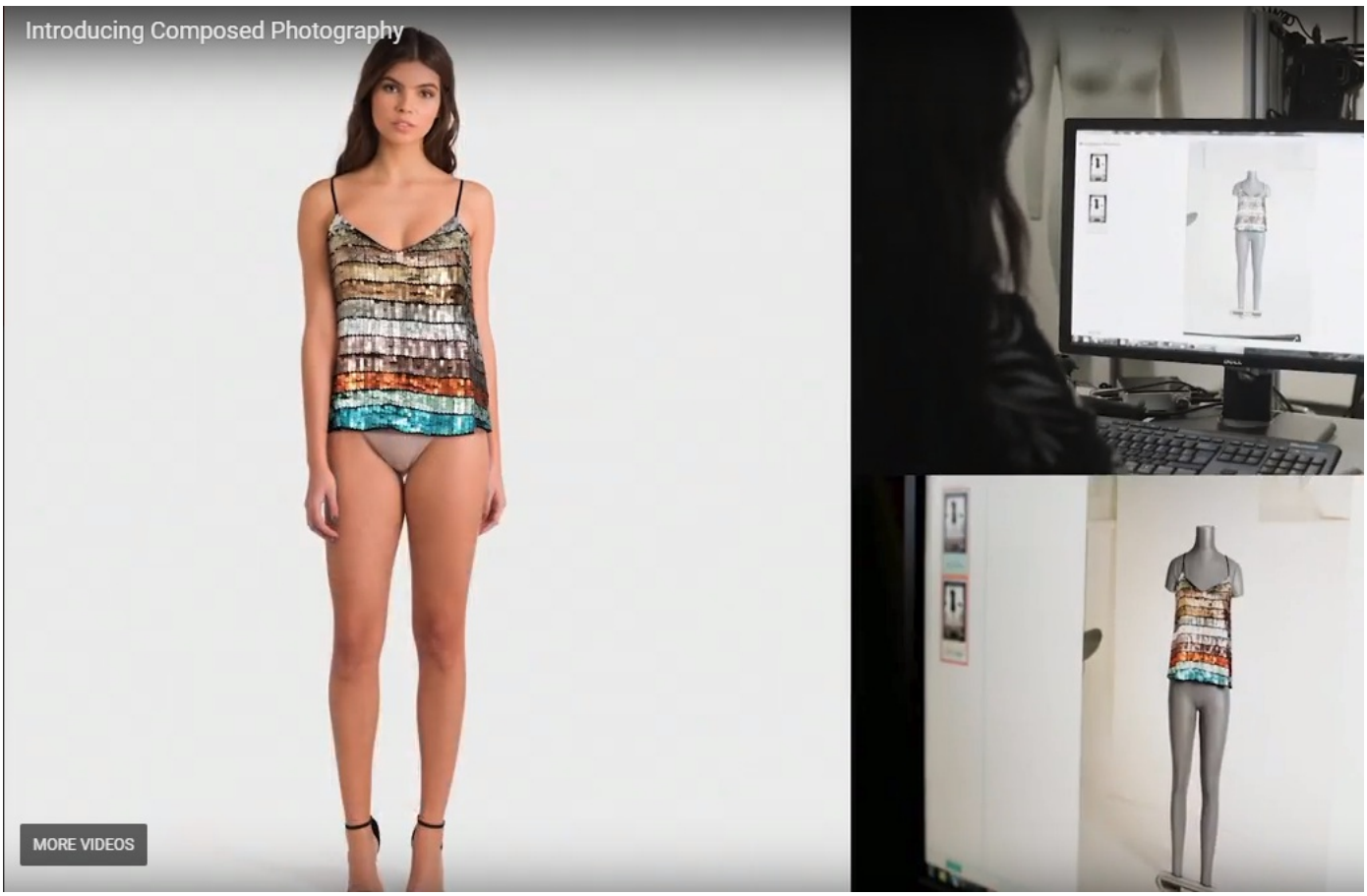
John Larsen April 4th, 2018

Proposal

Domain Background

The motivation for this project is rooted in my current work at Metail.com where we provide smart technology for fashion imagery. Meaning, shoot images without the need to hire models, photographers, or hair and make up.

A garment collection is styled and shot on mannequins, then our technology dresses the model (pre-shoot), of the customers choice, in the clothing as if he/she was photographed wearing it.



Simple. Fast. Efficient.

Well, is is actually not fast and efficient. A larger team based in India is cutting out the garment using Photoshop, then warping the cutout garment to a selected pose and finally shades are applied. Around 14% of the time in generating the final catalogue imagery is based around garment cutouts. We are determined to bring down the time it takes in order to reduce costs and lower turn around time.

What I would like to investigate is how deep learning may help us achieve those goals. In other words using CNN's to generate good quality image masks that our Indian team can use in order to speed-up the cutout stage.

The following links provide promise for a workable and production worth service:

- A 2017 Guide to Semantic Segmentation with Deep Learning
- Background removal with deep learning
- A Brief History of CNNs in Image Segmentation
- Semantic Segmentation using Fully Convolutional Networks over the years
- Image Segmentation Keras : Implementation of Segnet, FCN, UNet and other models in Keras
- Awesome Semantic Segmentation

Problem Statement

To generate a garment cutout mask that can overlay the original mannequin dressed image thereby allowing an easy way to cutout the garment, see illustration below.

How it works:

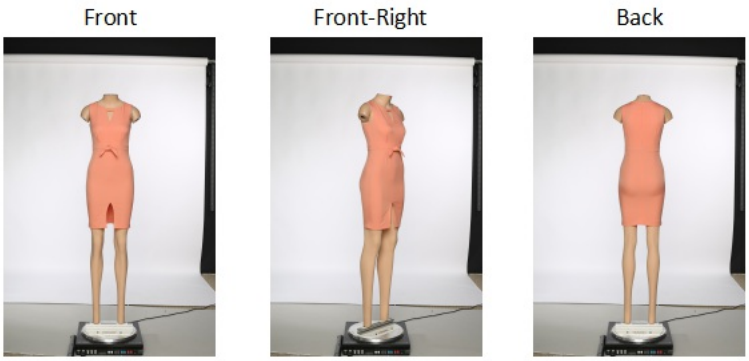
Actually our models and garments will be shoot at different viewpoints using our own custom rotating photo studio that automatically captures and processes 3 standard images of each garment. Namely Front, Front-Right and Back.

In order to measure the performance and success of the neural work I will use the Dice coefficient (also know as F1 score) as it is a well known metrics to use for measuring the performance of segmentation work. It literally measures the (Area of Overlap / Area of Union) between two datasets.

Datasets and Inputs

The dataset being used is generated from our past experience of digitizing roughly +10K garments. This is our own data generated from shooting in our own photo studios. For our product, Composed Photography, we capture the imagery in 3 viewpoints as mentioned earlier and illustrated below.

Image viewpoints:

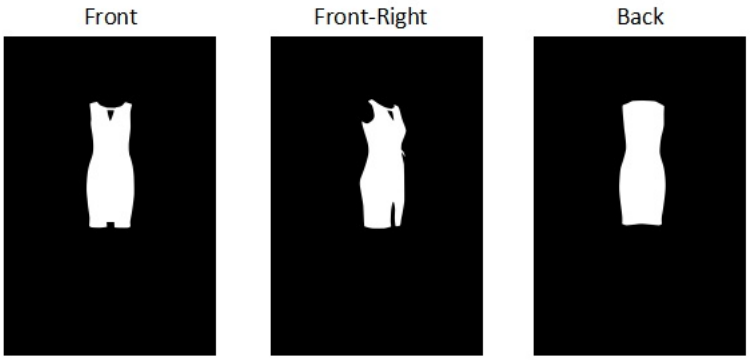


Now we need a mask to train our NN on so to learn what a garment looks like. It needs to learn if a pixel is either garment or background. For this purpose we use the manually cutouts created by our Indian team using Photoshop. These cutouts are our ground truth and illustrated here:

Image cutouts:

We will only use the cutouts indirectly as we won't train and validate against the cutout itself but rather against the alpha-mask contained in the cutout image:

Alpha cutout masks:



Note the segmentation dataset I will use consists of 1084 raw garment images and an equivalent number of cutouts (masks). The resolution of the imagery is 1152x1728. The combined size of the imagery is 1.1 GB.

In order to demonstrate the dataset I have uploaded the following to my Github account: Garment Segmentation Dataset. In the garment_segmentation folder you will find a readme file describing the dataset uploaded, in this case 358 garment images, all front view, showing original, cutout and mask images. This is roughly 1/3 of the data that will be used going forward, only uploaded 1/3 to save upload/download time

Solution Statement

As a solution to our problem we will develop a deep neural network capable of predicting if a given pixel is background or garment. In a nutshell it is a binary classification problem per pixel.

I am heavily influenced and inspired by the Kaggle competition Carvana Image Masking Challenge that illustrated some great and well performing solutions to this kind of problem. With the U-net architecture model winning and being used in 5 of the top-10 results this will be my chosen model architecture.

As already mentioned I will use the pixel accuracy but more importantly the Dice coefficient (F1 score) as the metrics for measuring the performance of the segmentation work:

Dice coefficient: $2 * |X \cap Y|/(|X|+|Y|) = 2TP/(2TP + FP + FN)$

where |X| and |Y| are the numbers of elements in the two samples. Dice coefficient is the quotient of similarity and ranges between 0 and 1. It can be viewed as a similarity measure over set (Area of Overlap / Area of Union)

Note: another common metric in segmentation work is the Intersection over Union IoU (also know as Jaccard index). These two indexes are related via:

IoU = Dice/(2 - Dice)

Note: I will compile the Keras model to use the following metrics:['dice coefficient', 'accuracy'] and likewise the loss function will also have the Dice coefficient built into it together with categorical cross-entropy.

Benchmark Model

I have two different benchmarks that I will compare the solution against:

- The first one is a trial done by our R&D team where they used similar imagery and a DeepLab-LargeFOV upsample model, exploiting atrous (dilated) convolutions to increase the field-of-view. Resulting in scores of MIoU = 98.22% and Dice = 98.32% on 2644 images.
- The second one is the earlier mentioned 'Carvana Image Masking Challenge'. Not exactly the same domain but conceptually a very similar problem. I would argue however that garments are more challenging than cars due to various fabrics, translucent materials, lace, fur etc. Nevertheless I hope to match scores as seen in the Carvana challenge where the winner demonstrated a Dice score around 99.7%

My hope and aim is to obtain a Dice score > 98.32% and close to 99.7%.

Evaluation Metrics

As already mentioned I will use the Dice coefficient (F1 score) as the metric for measuring the performance of the segmentation work. That is also directly comparable to our two benchmarks.

To recap the definition of the Dice coefficient:

Dice coefficient = $2*|X \cap Y|/(|X|+|Y|) = 2*TP/(2*TP + FP + FN)$

where |X| and |Y| are the numbers of elements in the two samples.

Project Design

The data (images) are well understood and coming from our own controlled environment. So all that needs doing is to generate the required tensors for training and validating the imagery.

Next we will build a simple model in Keras to run the training data against. Based on the lessons learned from the model we will

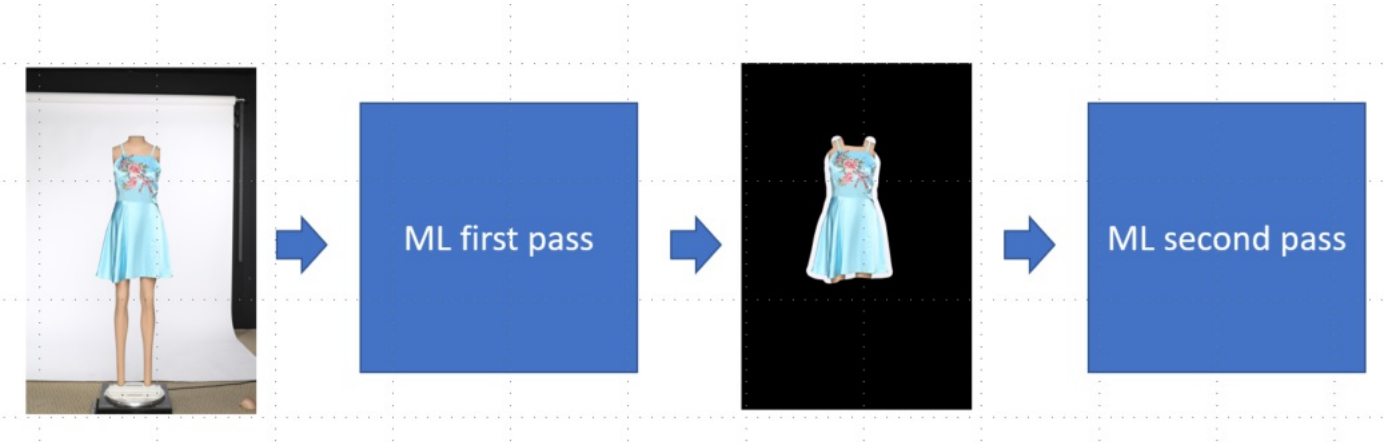
we will move onto a U-net implementation. Initially keeping it small, 128x128 model, before moving to the 1024x1024 model (the highest the GPU can handle).

With the U-net model in place the next phase is to evaluate on how:

- Learning rate affects results, while having a look at cyclical learning rates
- Optimization algorithms (Adam, RMSProp..) affects results

The dataset used is reasonably limited in size and that leads us naturally into image augmentation in form of translational invariance, zoom, coloration and intensity. Resulting in a larger dataset, more training data, but also allowing the model to generalise better (reducing over-fitting).

Lastly, looking at the raw imagery from the studios it is clear that there are some noise in the images from turntable, wires, background sheets and walls etc. What if we initially train a model to predict the garment in the original image, that is, calculate the garment mask and then use this garment mask, by dilating it, to cutout a new image with very little noisy background in it. This second stage image then feeds into a second model (same architectural model) but that has been trained on these 'dilated mask' images, illustrated below. Is that going to give us anything extra?



And we could keep going on but I will stop this project here :-)