UKRAINIAN CATHOLIC UNIVERSITY

Application of 3D graphic synthetic dataset generation for the means of image alpha matting using deep neural networks

Machine Learning Class Project Final Report

Maksym Gontar

June 10, 2018

1 Motivation

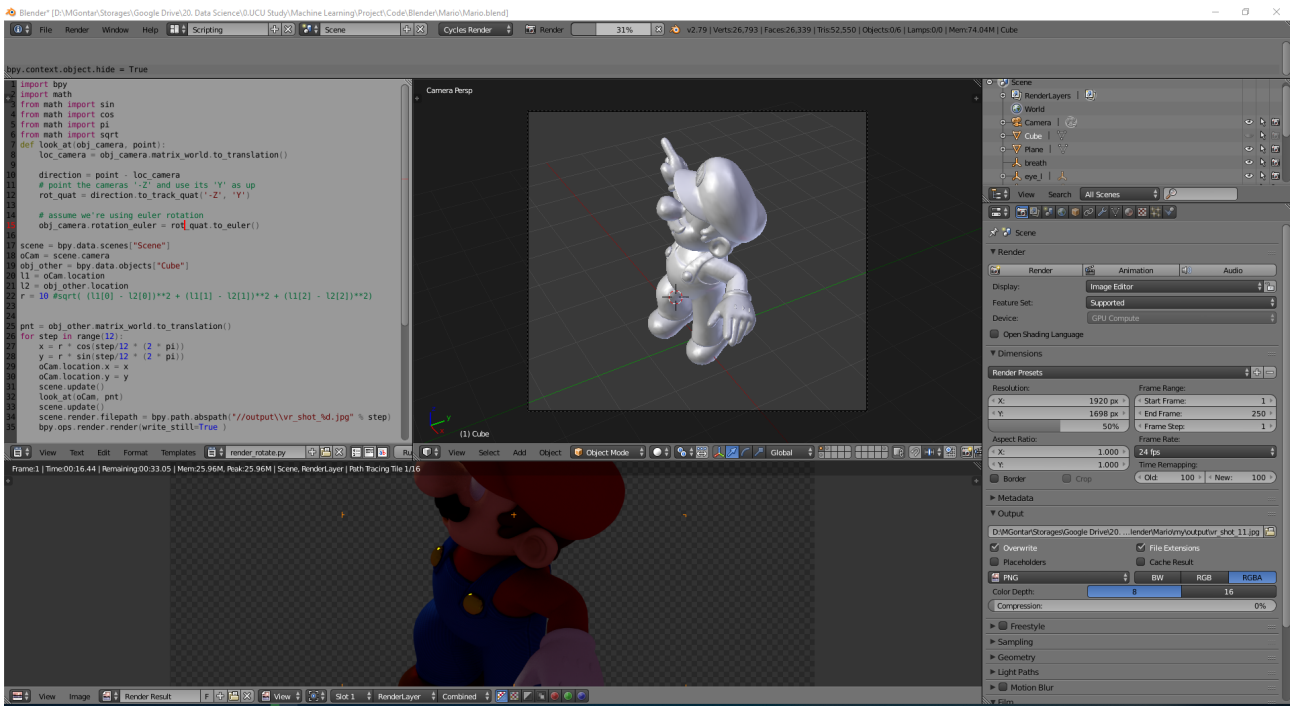
Purpose of the project is applying synthetic dataset for image alpha matting. Image matting is a process of extraction a foreground object from an image. Task of image matting has many practical applications in image and video editing (special effects, background substitution, etc). The problem of image matting has many different possible solutions, and each of them has it's own limitations. Most of them requites user interaction: a trimap - partition of the image into three regions: a definite foreground, a definite background, and a blended region where pixels are considered as a mixture of foreground and background colors. And even methods, that don't require user action [1], still need a lot of prepared data for training [4].
For the task of image matting using deep neural networks, we need a large set of images with objects on foreground, and a mask of object for each image. To produce a proper object mask, one would have to edit each image manually, using an graphical editor like GIMP or Photoshop, selecting the margins of the object on image, cutting the background, filling the object margins with a proper mask color (conventionally, white), filling the outside area with unmask color (black), etc. Which would mean a lot of boring manual work, time and money in terms of resources.
Here, synthetically generated data may come in hand. Synthetic data is any production data applicable to a given situation that are not obtained by direct measurement [2]. The idea of using in deep learning images, rendered from artificially designed 3D objects, is discussed for many years [5]. For the task of image matting, using of such process is very fitting: there is no complication in getting object mask automatically - all what is needed is to render object without a background texture or other objects with alpha channel (transparency), and then simply take transparent area as an unmasked area, and all the other - as a mask.

All of this may be done automatically (given the 3D graphics editor has a scripting interface), and replicated as many times as it's needed for different parameters of the 3D scene: camera view point, background environment, lightning parameters (number and position of light sources, their color, size, intensity, scattering), object parameters (its parts, their shape, color, texture), and other parameters, all of which may be changed automatically, which gives a large set of possible combinations, that could be generated, using minimum of design itself.
More general task, than image matting, is visual object masking, when object not necessarily is a foreground, and there maybe multiple of objects. And even more general task is image segmentation, when not necessarily only objects are masked, but also some important areas of an objects. The latter has also many practical applications, objects detection, classification, recognition in medical, transport, surveillance, and other areas. Deep learning is also used for this more general task [4], which suggests, that the problem of dataset here also may be resolved with synthetic data generation.

2 Dataset

To generate the dataset for this project I decided to use Blender - open source 3D graphic software. I've picked a ready to use 3D model Mario with creative commons license, set up render with transparent background to PNG and wrote a small python script to render the scene while rotating camera around the object - a sequence of 12 images with step of 30 degrees rotation in horizontal plane.

Figure 2.1: Mario 3D model in Blender editor

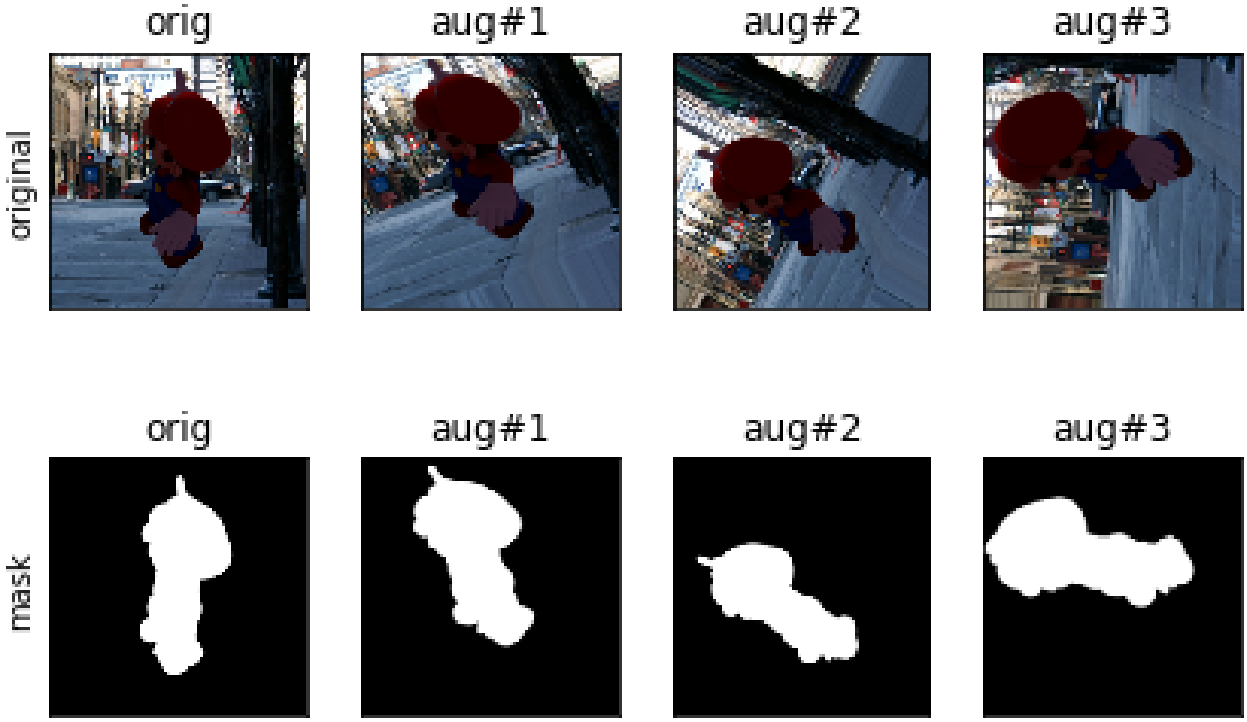


After that, I took 10 CC images from flickr and merged them with rendered Blender output in all combinations using imagemagick batch script. Also, I've created a trimap (basically a mask which marks fully/partly the background and foreground objects on image) manually in a way that it would fit all dataset images. As a result of such approach, from one 3D model and environment 10 photos, in output 120 images were generated.

Figure 2.2: Dataset item example

Since large images requires more resources and processing time, all image data was downsized to 64x64 px. For easier usage of the dataset brine package was used, and appropriate script for converting image data into brine dataset was developed. Finally, image data augmentation was used (rotation, pan, zoom, hue, saturation).

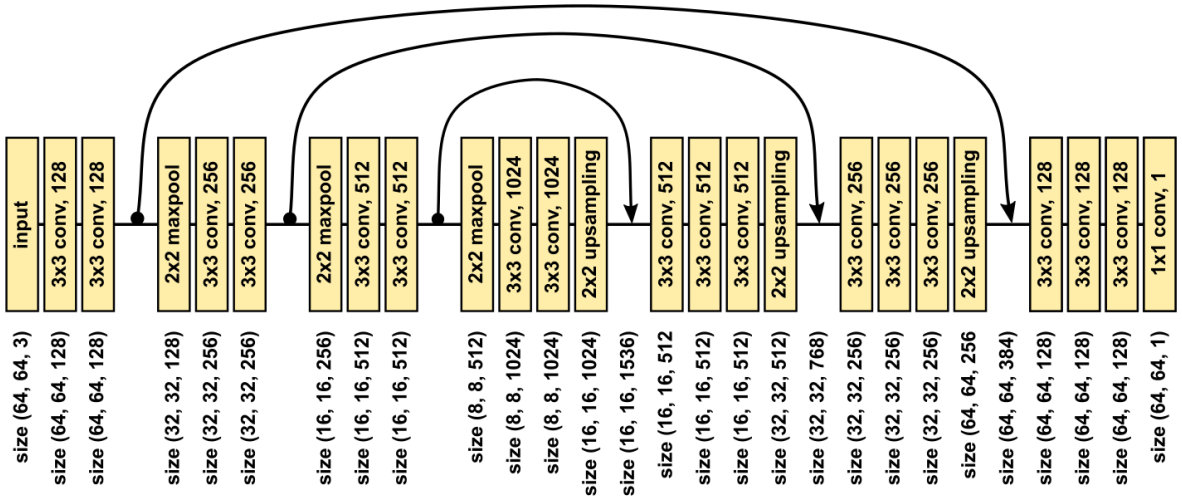
Figure 2.3: Rotate augmentation



3 Model

First I tried to implement VGG16-based convolution autoencoder model from Deep Matting paper [4], but results was very inefficient: training yielded very low accuracy which discourage from continue the experiment.
Next I tried to reuse residual convolution autoencoder UNet-based model [7], implemented by one of Kaggle competitors of Carvana Challenge. Effectiveness of the UNet based architecture for object masking task can be explained by the fact that low level feature extraction layers are contributing the most to output of the model - and here low level features like the edges are important. My modification of the model was adapted for 64x64x3 input.

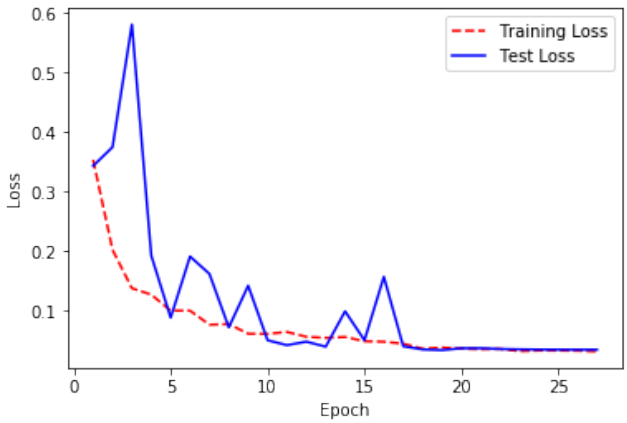
Figure 3.1: UNet-based model architecture



4 Evaluation

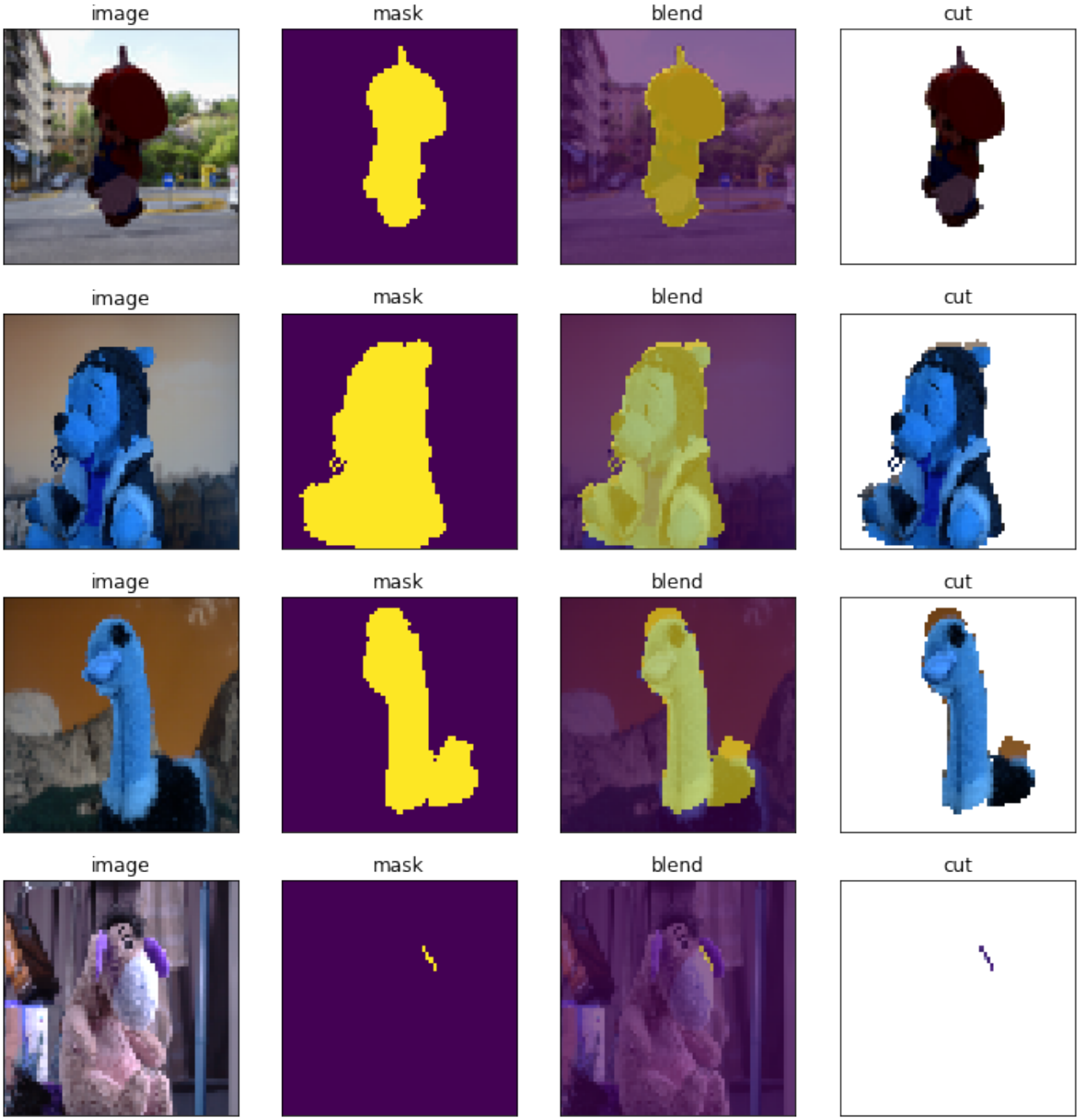
As a loss function, I have used a score metric which is a combination of binary cross-entropy and Dice loss. As the training set up, I have used up to 100 epochs, with early stopping condition on minimum loss delta. It converged with 0.9797 accuracy after 27 epochs in about 1.3 hours on machine with Intel Core i5-3210M @ 2.5 GHz x64 CPU, 12 Gb RAM and Win 10 x64 OS.

Figure 4.1: Loss plot of the model training



Prediction object mask with the same dataset shows high precision. Also, the model was evaluated with several samples from Alpha Matting Evaluation Website. Some samples shows high precision as well, some shows poor precision, some fails to predict object mask whatsoever.

Figure 4.2: Predictions



5 Final thoughts

This work shows that effective use of augmented data based on 3D graphics for the means of image matting, object masking and segmentation is possible, which may be explained by the fact that 3D graphics resample real world properties like space transformations, lighting, material physics, thus may substitute real world image/video data. The current model also may be improved a lot by implementing more variety of 3D objects, with different properties and more background photos. You can find all code, scripts and data at the project Github repository [8]

References

[1] Olivier Juan and Renaud Keriven. Trimap Segmentation for Fast and User-Friendly Alpha Matting, 2005

[2] Synthetic data. McGraw-Hill Dictionary of Scientific and Technical Terms. Retrieved November 29, 2009

[3] Anat Levin, Dani Lischinski, Yair Weiss. A Closed Form Solution to Natural Image Matting, 2007

[4] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep Image Matting, 2017

[5] Xingchao Peng, Baochen Sun, Karim Ali, Kate Saenko. Learning Deep Object Detectors from 3D Models, 2015

[6] Mihai-Sorin Badea, Iulian-IonuÈŻ Felea, Laura Maria Florea, Constantin Vertan. The Use of Deep Learning in Image Segmentation, Classification and Detection, 2016

[7] Olaf Ronneberger, Philipp Fischer, Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015

[8] https://github.com/mgontar/ucu-ml-project-alpha-matting