Project Name: Image 3D Parametrized

Table of Contents

Demo

СВЕРШИЛОСЬ!

СОВЕТСКИЙ ЧЕЛОВЕК – В КОСМОСЕ

СООБЩЕНИЕ ТАСС

О ПЕРВОМ В МИРЕ ПОЛЕТЕ ЧЕЛОВЕКА В КОСМИЧЕСКОЕ ПРОСТРАНСТВО

Об успешном возвращении человека из первого космического полета

Первый пилот-космонавт Юрий Гагарин

К КОММУНИСТИЧЕСКОЙ ПАРТИИ И НАРОДАМ СОВЕТСКОГО СОЮЗА!
К НАРОДАМ И ПРАВИТЕЛЬСТВАМ ВСЕХ СТРАН!
КО ВСЕМУ ПРОГРЕССИВНОМУ ЧЕЛОВЕЧЕСТВУ!

Обращение Центрального Комитета КПСС, Президиума Верховного Совета СССР и правительства Советского Союза

ПРОЛЕТАРИИ ВСЕХ СТРАН, СОЕДИНЯЙТЕСЬ!

ЛИТЕРАТУРНАЯ ГАЗЕТА

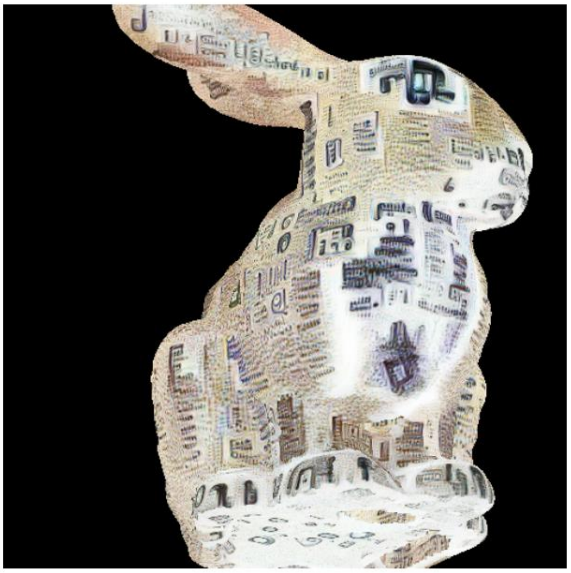
ОРГАН ПРАВЛЕНИЯ СОЮЗА ПИСАТЕЛЕЙ СССР

Год издания 32-й
№ 45 (4324)
Четверг,
13
апреля
1961 г.

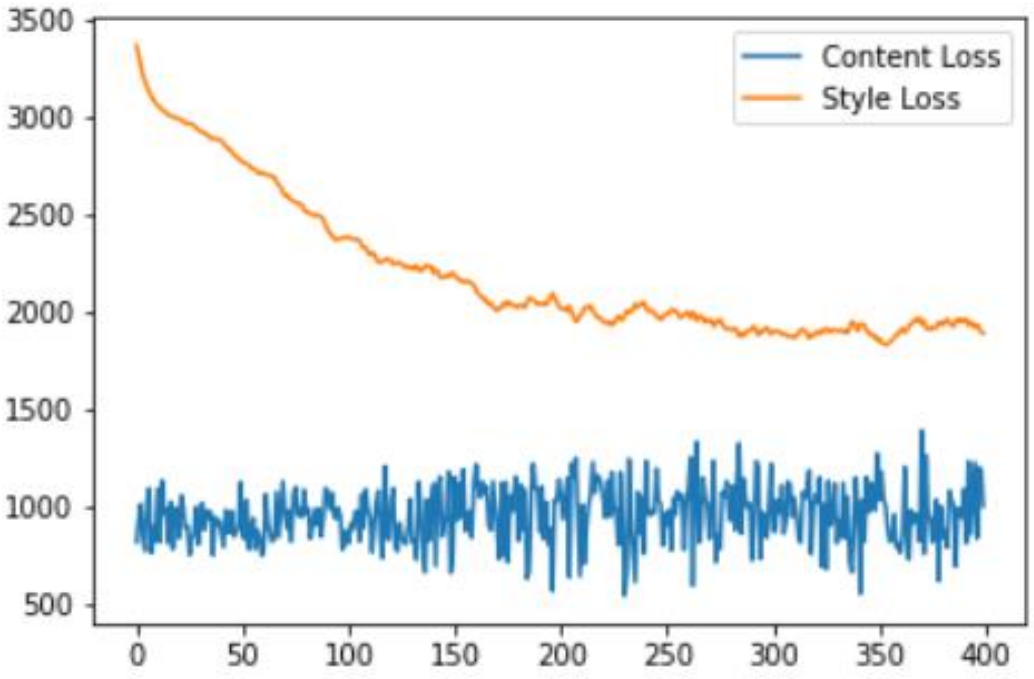
Цена 4 коп.

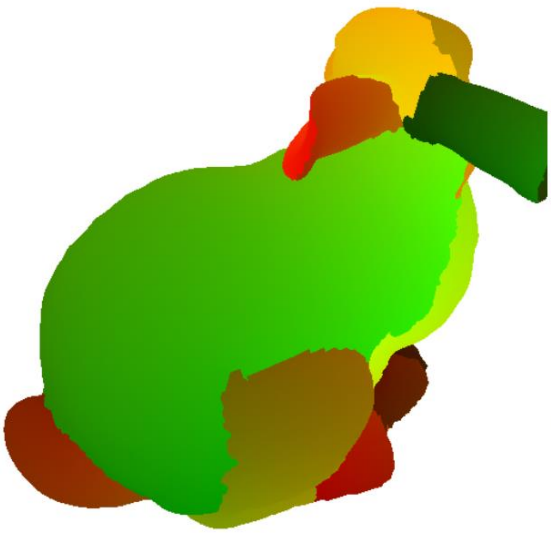
СОВЕТСКОМУ КОСМОНАВТУ, ВПЕРВЫЕ В МИРЕ СОВЕРШИВШЕМУ КОСМИЧЕСКИЙ ПОЛЕТ,

current frame



content





Overview

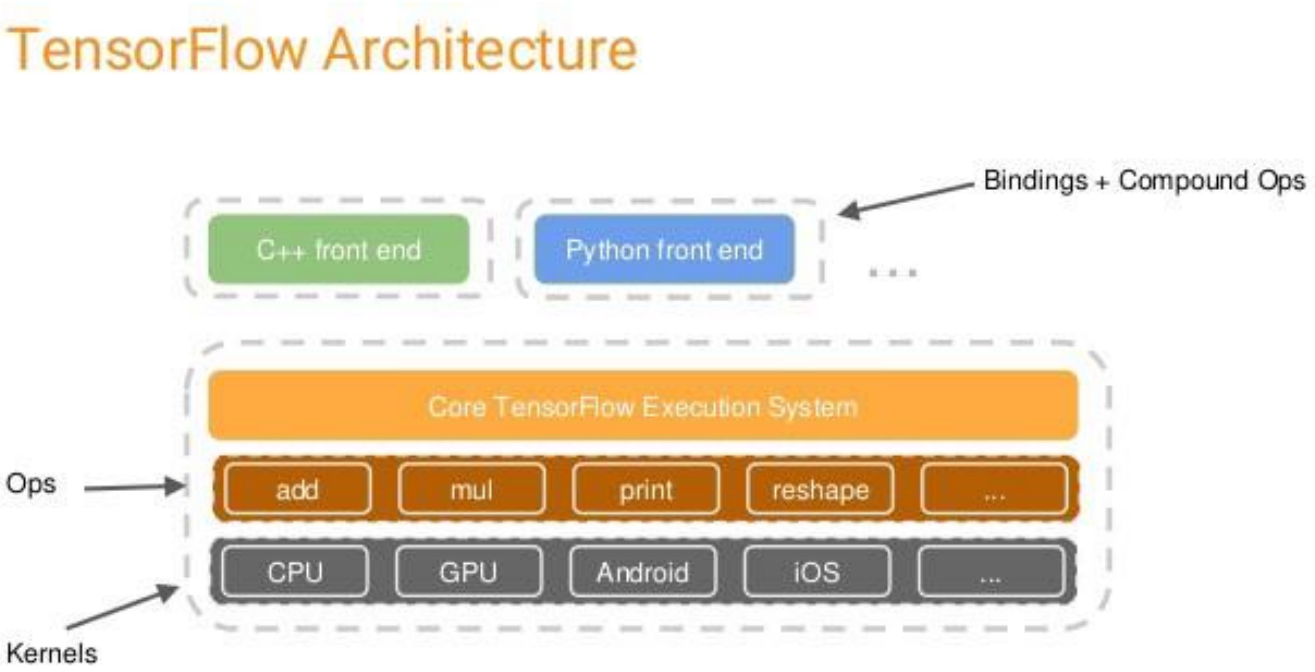
This is diving into Image 3D Parameterized Concept.
It uses Lucid to implement style transfer from a textured 3D model and a style image onto a new texture for the 3D model by using a Differentiable Image Parameterization.
This repository contains the code for Feature Engineering using python's various libraries.
It used lucid, Tensorflow, Numpy and Matplotlib libraries.
This libraries helps to perform individually one particular functionality.
Numpy is used for working with arrays. It stands for Numerical Python.
Matplotlib is a plotting library.
TensorFlow is a Python library for fast numerical computing created and released by Google.
Using Lucid is extremely simple. The framework can be installed as a simple Python package.
The purpose of creating this repository is to gain insights into working with 3D Style Transfer.
These python libraries raised knowledge in discovering these libraries with practical use of it.
It leads to growth in my AI repository.
This above few screenshots will help you to understand flow of output.

Motivation

The reason for building this is, because a month ago I have seen one travelled enthusiast video and in that they were exploring Finland. And in that place, in some town in Finland has cow's 3D Structure with Google Map of nearby location printed on it. This really caught my attention. I also like to travel and explore new places, their culture, customs and meet people. So, I came across this story which fascinated me a lot and that is when I started to research about this concept and made this one. I also think that, there is huge chances of making integration of this type is great business to do. Hence, I continue to gain knowledge while practicing the same and spread literary wings in tech-heaven.

Technical Aspect

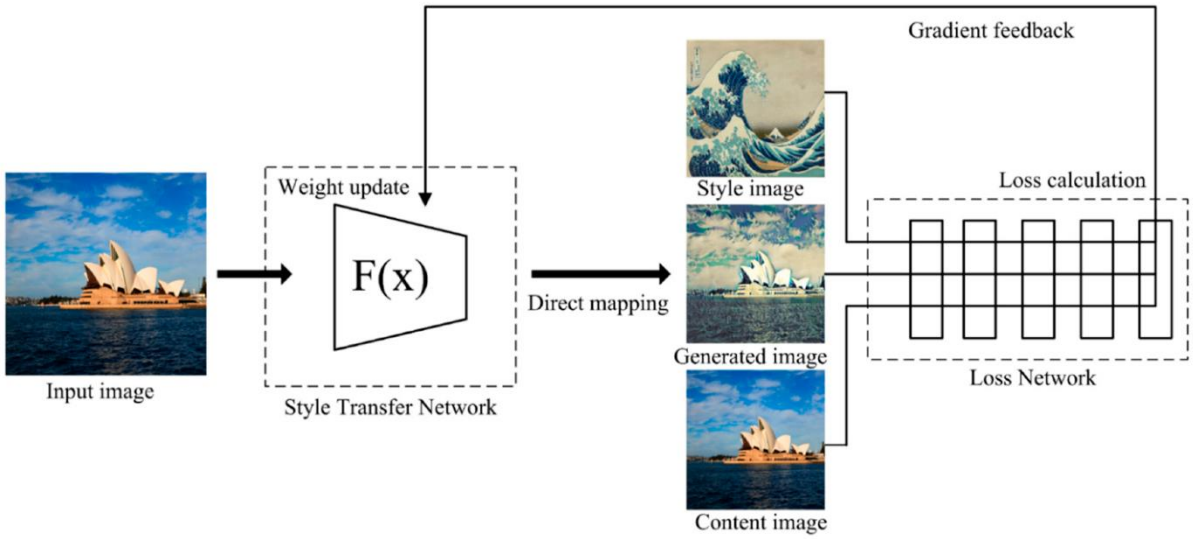
Tensorflow has excellent community support and provides scalability. TensorFlow provides a better way of visualizing data with its graphical approach. It also allows easy debugging of nodes with the help of TensorBoard. This reduces the effort of visiting the whole code and effectively resolves the neural network. Both TensorFlow and PyTorch provide useful abstractions to reduce amounts of boilerplate code and speed up model development. The main difference between them is that PyTorch may feel more "pythonic" and has an object-oriented approach while TensorFlow has several options from which you may choose.



The Lucid Library is an open source framework to improve the interpretation of deep neural networks. We use Lucid to produce feature visualizations on 3D mesh surfaces by using a Differentiable Image Parameterization.

Texture Synthesis We describe this process in the Efficient Texture Optimization through 3D Rendering. Remember that the main ingredients beside the 3D model we just loaded are: a way to sample random views of the 3D model a renderer, which turns the view, model, & texture into a flat image a model, which we feed that flat image to calculate a loss and gradients with respect to the flat image and from there we can use our knowledge of which parts of the 3D model were visible in the flat image to backpropagate that gradient through the rendering process and into the learned texture.

Renderer means we provide a way to sample random views onto a 3D model. You can specify the range of distances, and the resulting views will be centred on the object. The resulting 4x4 matrix is interpreted as a Model-View matrix.



Installation

Using intel core i5 9th generation with NVIDIA GFORECE GTX1650.
Windows 10 Environment Used.
Already Installed Anaconda Navigator for Python 3.x
The Code is written in Python 3.8.
If you don't have Python installed then please install Anaconda Navigator from its official site.
If you are using a lower version of Python you can upgrade using the pip package, ensuring you have the latest version of pip, *python* -m pip install --*upgrade pip and press Enter*.

Run/How to Use/Steps

I have created Image 3D Parameterized in Colab.
You can copy the code from my Colab to your Colab and can start GPU then run all the cells.

Directory Tree/Structure of Project

Folder: 20)Image 3D Parameterized
Image_3D_Parameterized.ipynb

To Do/Future Scope

Can try with other patterns and object.

Technologies Used/System Requirements/Tech Stack

Credits

Krantirk

https://www.kdnuggets.com/2020/04/openai-open-sources-microscope-lucid-library-neural-networks.html