NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA SURATHKAL

"Web Portal for Neural Style Transfer"

A PROJECT REPORT
In partial fulfilment for the award of the degree of
MASTER OF COMPUTER APPLICATIONS

Department of Mathematical and Computational Sciences
National Institute of Technology Karnataka, Surathkal

Submitted By-
Sudiksha Agrawal
194CA046

Submitted To-
Mrs. Deepthi L.

BONAFIDE CERTIFICATE

BONAFIDE CERTIFICATE Certified that this project report "**Web Portal for Neural Style Transfer**" is the bonafide work of **Sudiksha Agrawal (194CA046)** who carried out the project under my supervision. This is to further certify to the best of my knowledge, that this project has not been carried out earlier in this institute and the university.

Signature

Mrs. Deepthi L.

Certified that the above-mentioned project has been duly carried out as per the norms of the college and statutes of the university

Signature

(Prof. Shyam S. Kamath)

HEAD OF DEPARTMENT

ACKNOWLEDGEMENT

I wish to express my profound and sincere gratitude to Mrs. Deepthi L. Department of Mathematical and Computational Science NITK, Surathkal, who guided me into the intricacies of this project Web Portal for Neural Style Transfer.

I thank Prof. Shyam S. Kamath, Head of the Department of Mathematical and Computational Science, NITK, Surathkal for extending their support during the course of this investigation.

Sudiksha Agrawal
194CA046

Table of Content

Abstract

This project demonstrated the power of Chainer Neural Network framework in creating artistic imagery by separating and recombining image content and style models. This process of using Chainer to render a content image in different style models is referred to as Neural Style Transfer (NST). Neural Algorithms of Artistic Style are used that can separate and recombine the image content and style of natural images. The algorithm allows us to produce new images of high perceptual quality that combine the content of an arbitrary photograph with the appearance of 2 style models. The results provide new insights into the deep image representations learned by Chainer Neural Networks and demonstrate their potential for high level image synthesis and manipulation.

Introduction

This is a Chainer implementation of techniques like Image Style Transfer Using Chainer – a python-based framework for neural networks, Preserving Colour in Neural Artistic Style Transfer. Additionally, techniques are presented for semantic segmentation and multiple style transfer. The Chainer algorithm synthesizes artistic image by combining content images with style models.

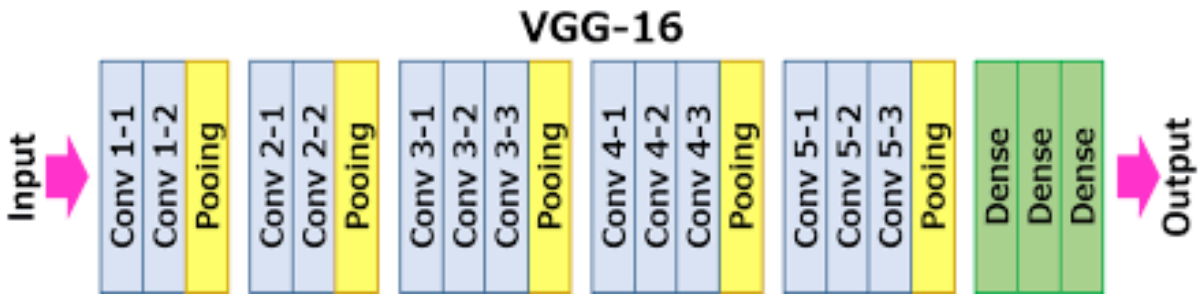
This project aims at transferring the style from one model onto another image, goal is to synthesise a texture from a source model while constraining the texture synthesis in order to preserve the semantic content of a target image. The style model and the target image can be uploaded by the user from the user interface, the user can also click her/his own picture through a webcam to generate his own target image. The synthesised image is saved in the local system of the user and is displayed onto the user interface for inspection.

What Chainer?

That is why Chainer's design is based on the principle "Define-by-Run" - network is not pre-defined at the beginning but is dynamically defined on-the-fly. It uses forward computation - written as a regular program code with special variables and operators, allowing arbitrary control flow statements. The computational graph can be modified within the loop/iteration. All of the above makes forward computation intuitive and easy to debug. Within "Define-by-Run" scheme, rather than storing programming logic, Chainer saves the history of computation and programming logic in Python can be used at its fullest potential.

Chainer is powerful, flexible and intuitive framework of neural networks that supports almost arbitrary architectures. It helps to "bridge the gap between algorithms and implementations of deep learning". To get more information you can visit Chainer website or browse its code in GitHub repository.

VGG16 model



Objective

The objective of Web Based Image Style Transfer Using Chainer Neural Network framework is to implement algorithms that take as input a content image and a content style — and outputs a new image depicting the objects of the content image in the style model. This technique is quite a hot topic and is widely used nowadays to create different "image filters" or "selfie filters" which are popular in social media platforms like Instagram and Snapchat.

System Requirements

Hardware Requirements
Operating system - Windows 7 or later
Processor - Intel Pentium 4 or later
Memory - 2 GB minimum, 4 GB recommended
GPU: NVIDIA GeForce GTX 1080/PCIe/SSE2
1 CPU Core
TF_CUDA_VERSION "64_101"

Software Used
python: 2.7.12 or later
tensorflow: 0.10.0rc
opencv: 2.4.9.1
Django web framework
VGG16 model
HTML, CSS, Bootstrap for front end

Design and Implementation

Implementation -

HTML, CSS, javascript and bootstrap are used for front end and styling

Chainer is used for implementing neural networks

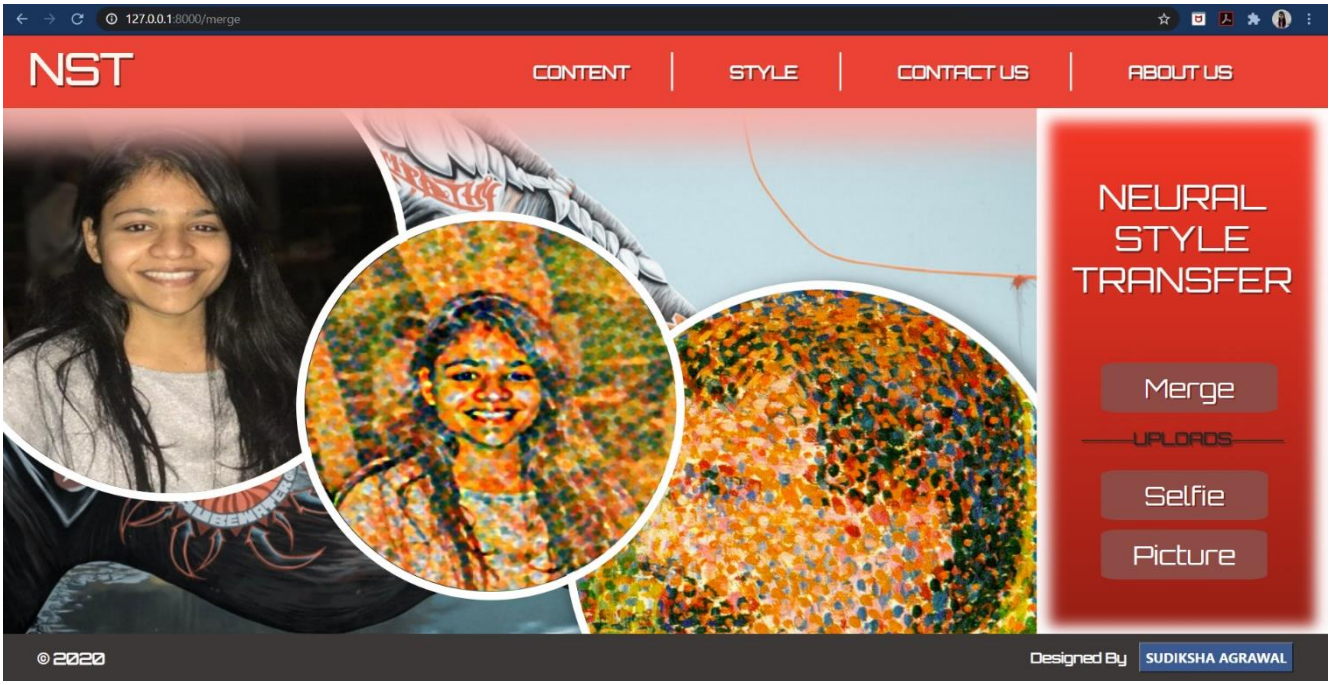
opencv is used to capture image dynamically

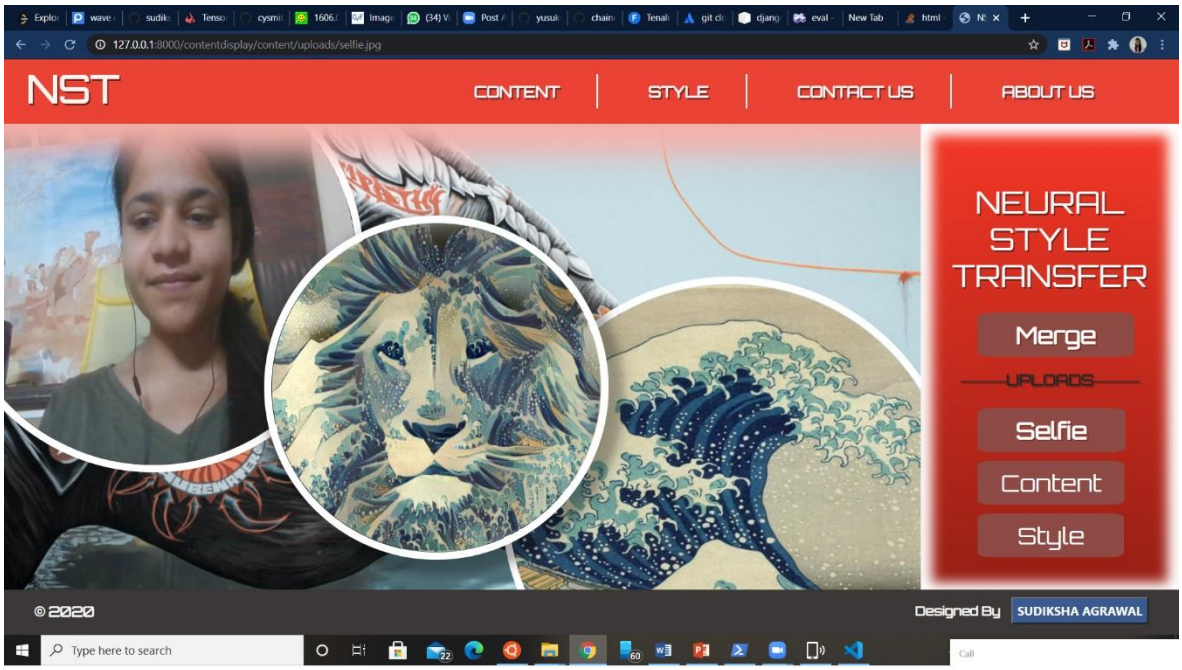
Django is a server-side web framework which is used to handle HTTP requests, database control and management, URL mapping, etc.

Design-

This page is index.html. It handles all the functionality of this project.

http://127.0.0.1:8000/index

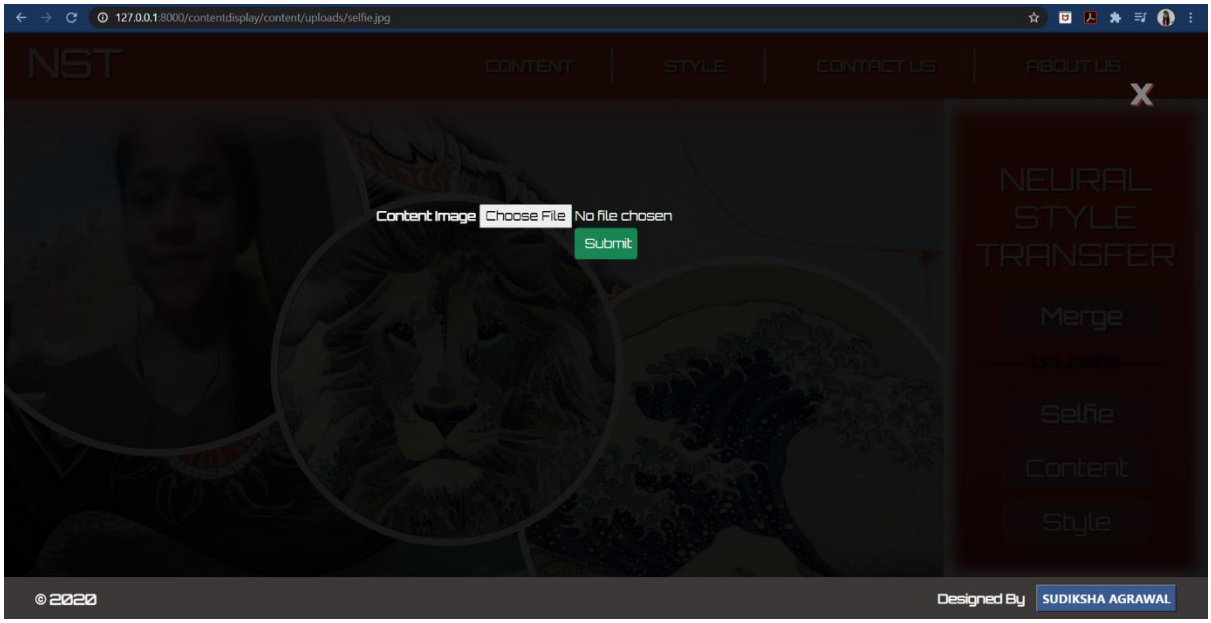




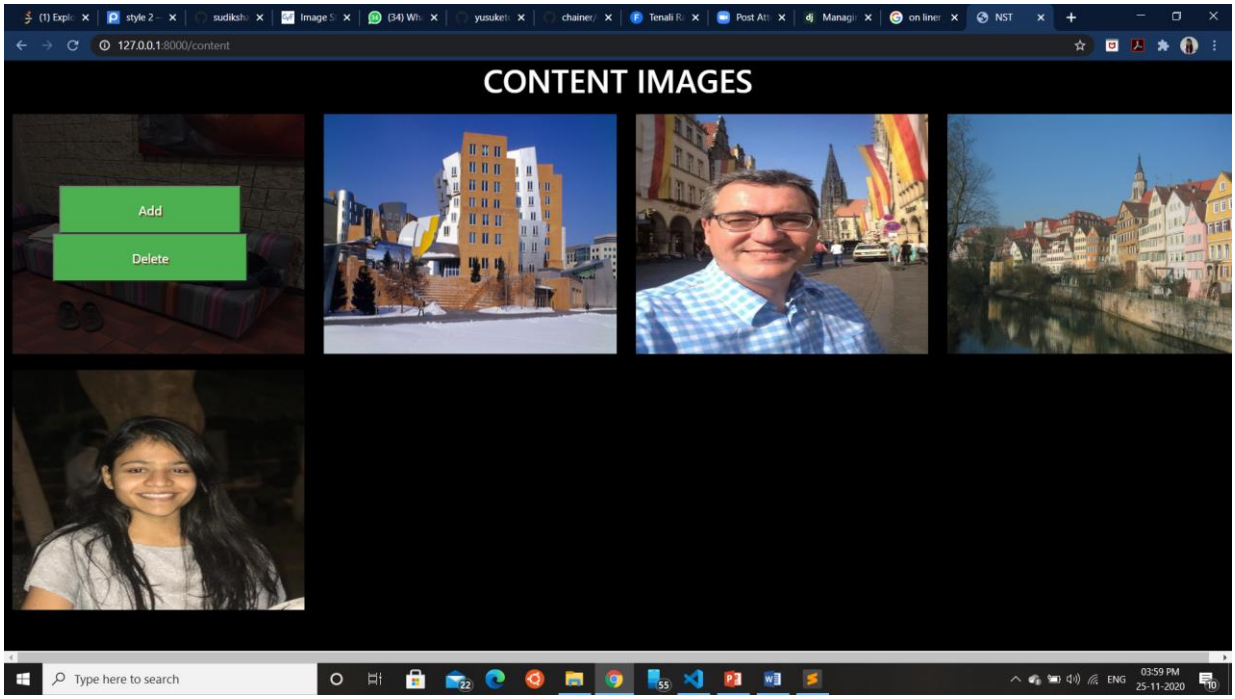
After clicking on selfie button your picture is clicked from python's opencv module and saved on your system dynamically, it is also saved in database.

http://127.0.0.1:8000/contentdisplay/content/uploads/selfie.jpg

---

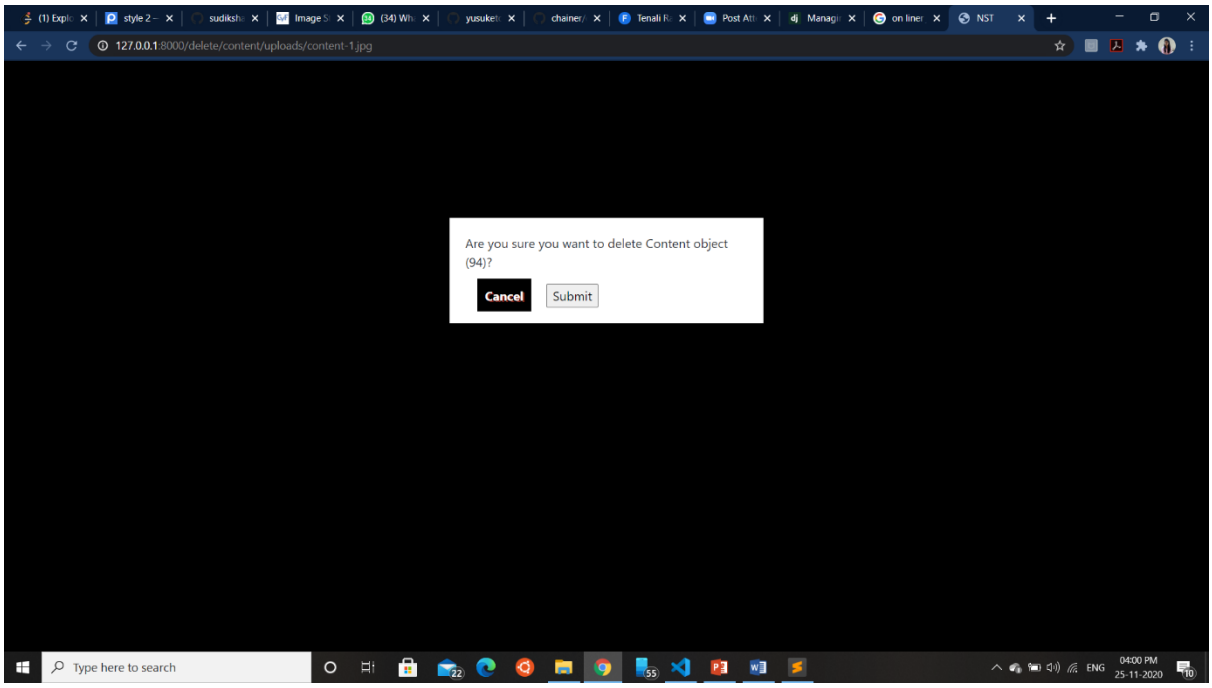


By clicking on content button on right container this overlay window is opened and you can upload any image from your local system, this image is then saved in uploads folder on your local system as well as in database. This image can be then used as content image.



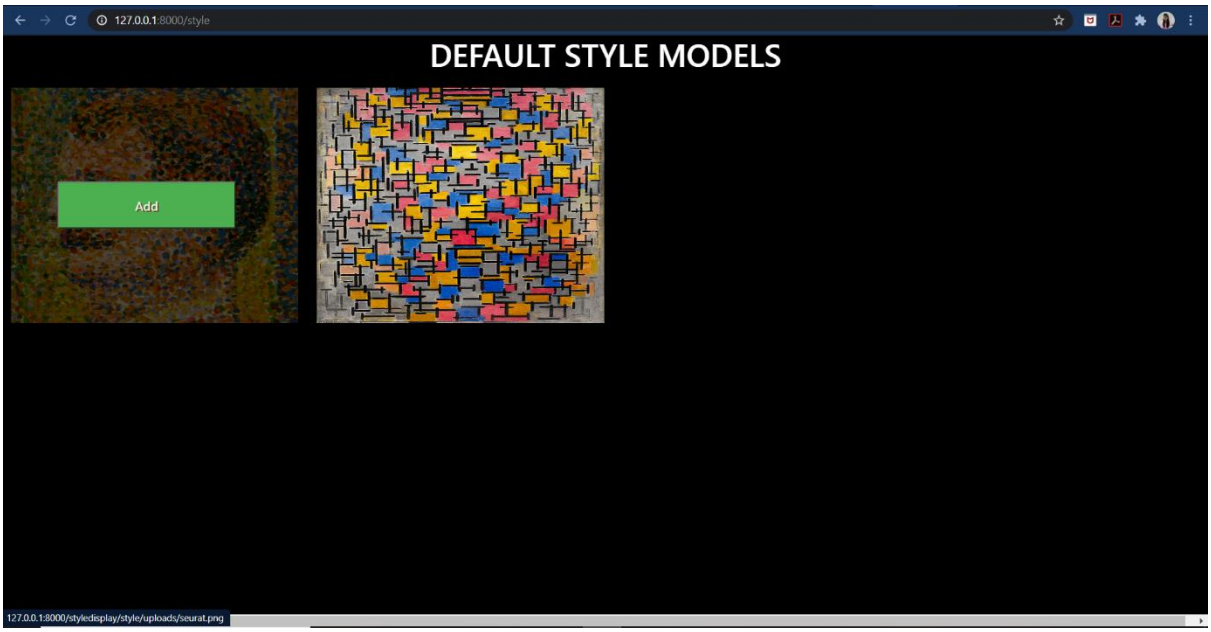
This is the content page where all the content images which you have uploaded from your local system or clicked from webcam are displayed for preview. This page provides one more functionality that you can add any image as content image for neural processing, you can also delete any content image.

http://127.0.0.1:8000/content



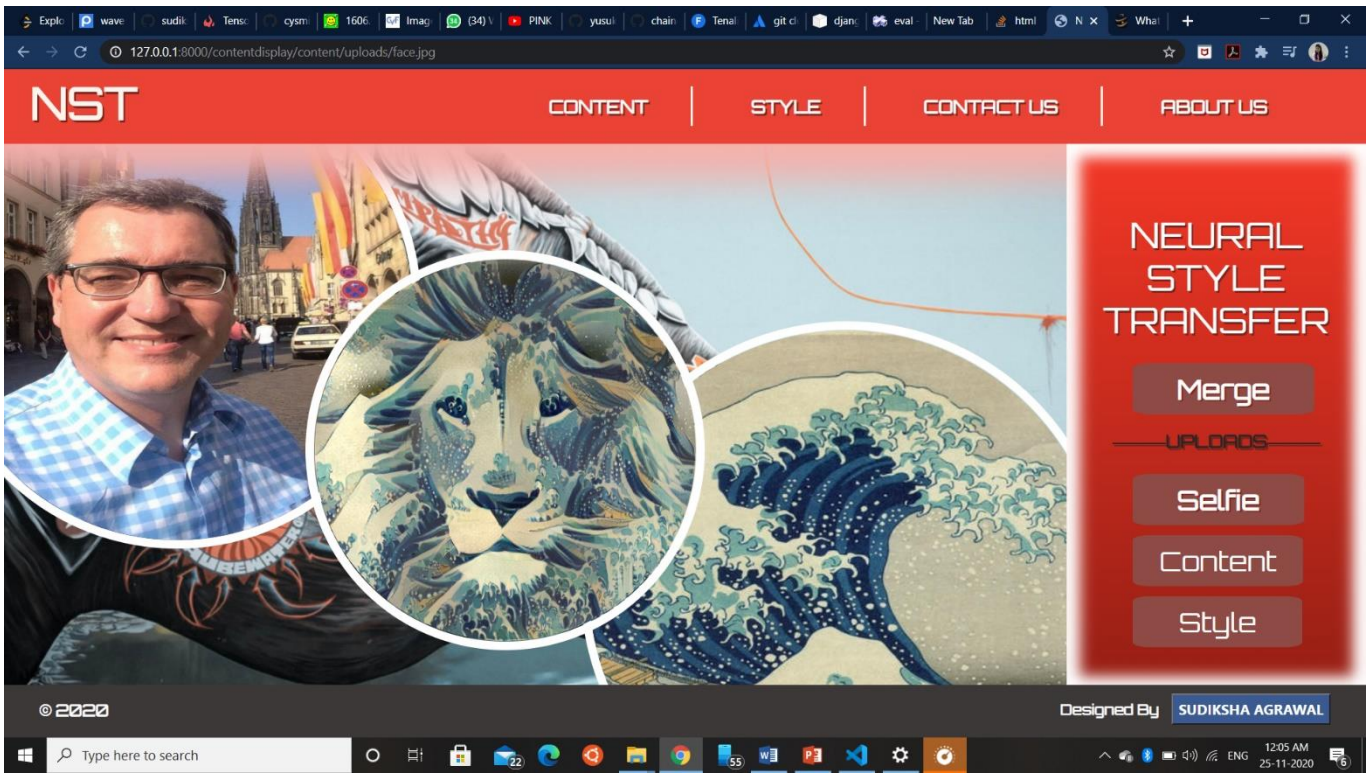
Delete page

http://127.0.0.1:8000/delete/content/uploads/content-1.jpg



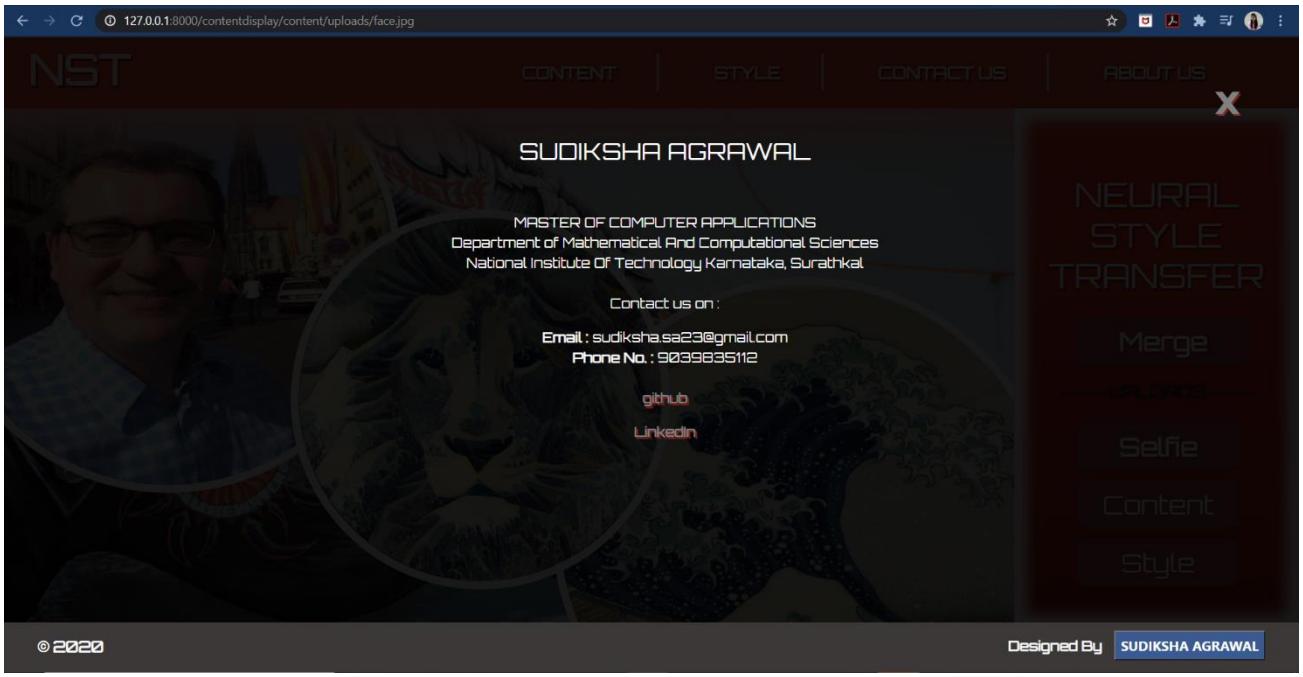
This is the style page where the style images given the preview of both the models are displayed, you can choose any style to add to you content image. This page provides one more functionality that you can add any image as content image for neural processing.

http://127.0.0.1:8000/style

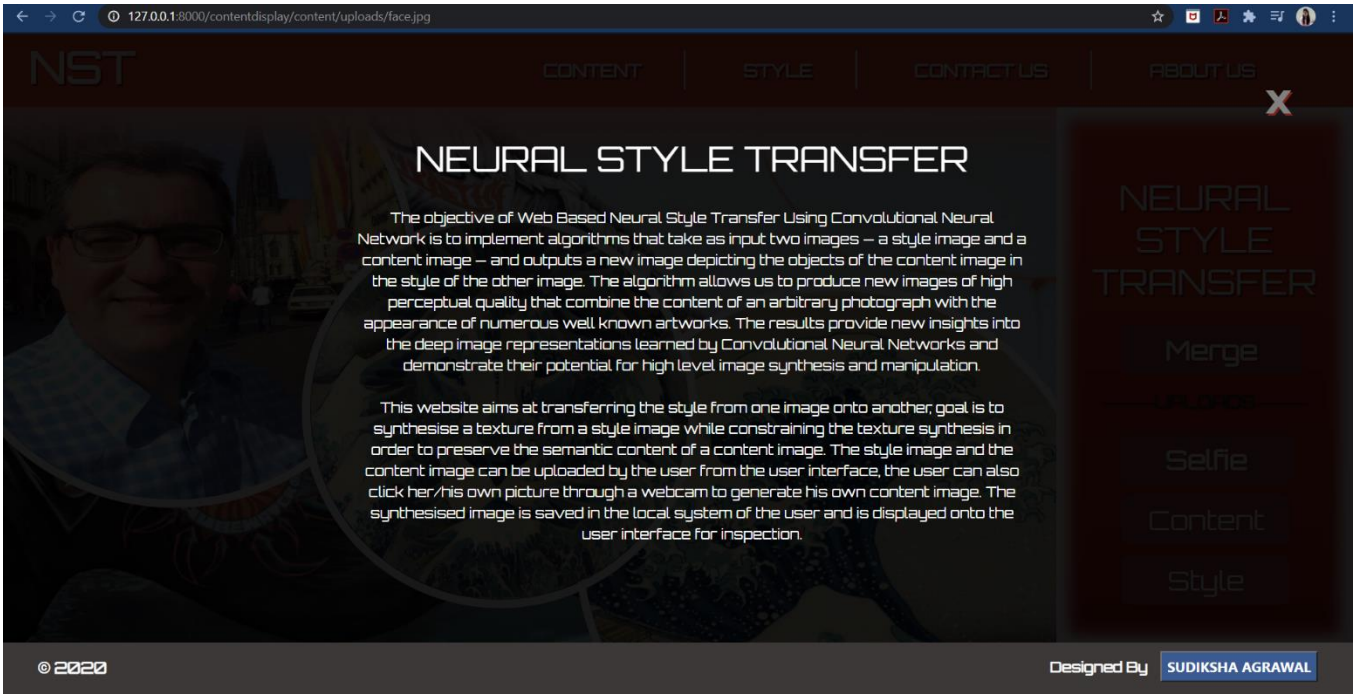


Chosen style image is displayed in the left circle and chosen content image is displayed in the right circle and the result generated is displayed in the middle circle.

http://127.0.0.1:8000/contentdisplay/content/uploads/lion.jpg



This is the contact us overlay, which displays all of my contact information.



This is the about us page which displays information about technologies used in this website and the functionality rendered by this.

Result Analysis

Seurat model used for styling

Composition model used for

Conclusion

This project is a successful implementation of fast neural style transfer using chainer framework. This implementation is made user friendly by making a web portal where users can implement chainer style models on his own picture or any picture.

Future Scopes

- We can implement mechanism such that we can choose any style image of our choice.

References

- https://getbootstrap.com/docs/4.1/getting-started/introduction/
- https://github.com/yusuketomoto/chainer-fast-neuralstyle/tree/master
- https://github.com/chainer/chainer/tree/v1
- https://docs.djangoproject.com/en/3.1/
- https://wimaxbangla.blogspot.com/2020/04/image-capture-using-opencv-in-python.html