KIET Group of Institutions, Ghaziabad

(An ISO – 9001: 2008 Certified & 'A+' Grade accredited Institution by NAAC)

I Project Presentation (KCS 753) (Neural Style Transfer)

Guide Name: Mr. Akash Goel

Project Members Name with Roll Number & Section

1.Palak Singh (2000290120103-B)

2.Ragini Rani (2000290120121-B)

3. Kalash Jain (2000290120080-B)

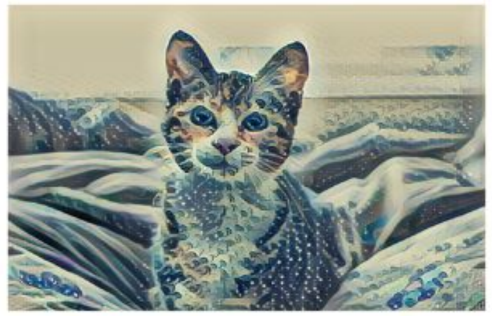
Problem Statement

The problem we wanted to solve in this project was how to improve the process of transferring the artistic style of one image to the content of another image using neural networks while maintaining the integrity of the content and achieve high quality results.
In other words, the central challenge to be addressed is how to make the neural type conversion process more efficient, productive and user-friendly, ensuring that it produces images that the tone is visually appealing while preserving the important elements of the original content.

Style transfer

content image

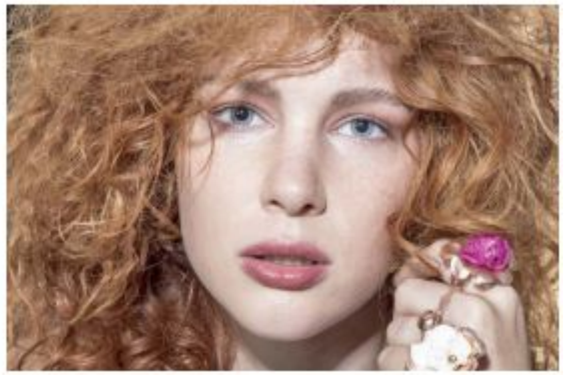
style image


target image

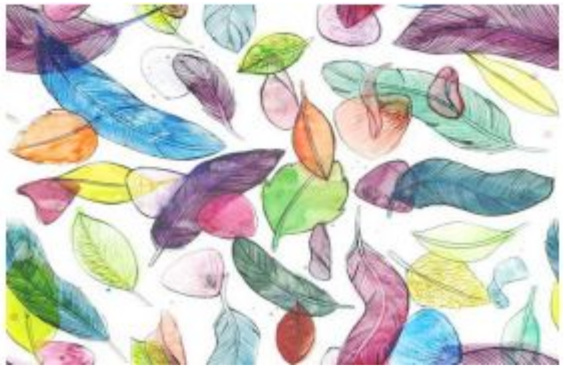
Objectives

- Content Preservation: Ensure that the generated images retain the meaningful content and recognizable objects from the original input images.
- Style Replication: Replicate the distinctive artistic styles, including textures, colors, and patterns, from reference images to create visually appealing and stylistically consistent output.

Multiple Style transfer:-

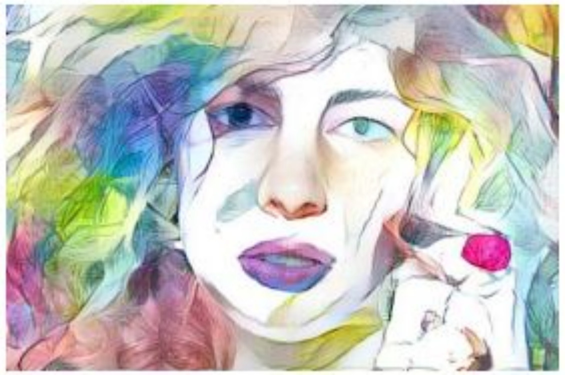


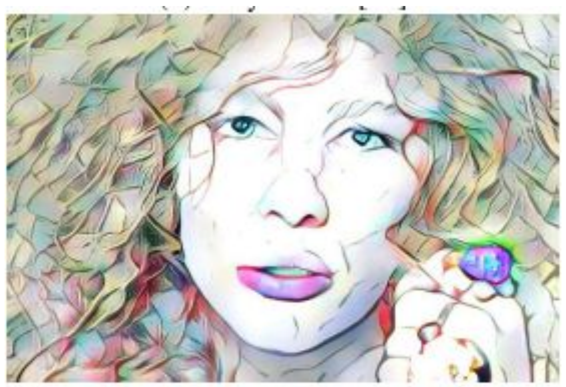
+

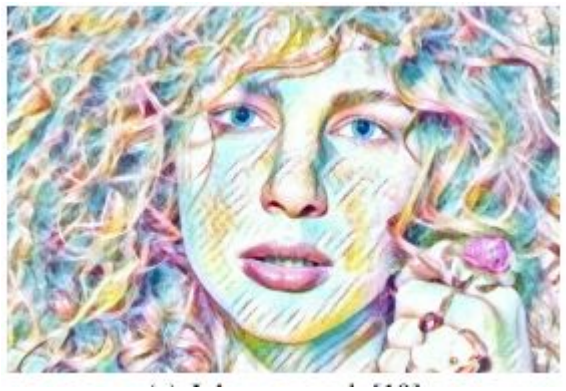


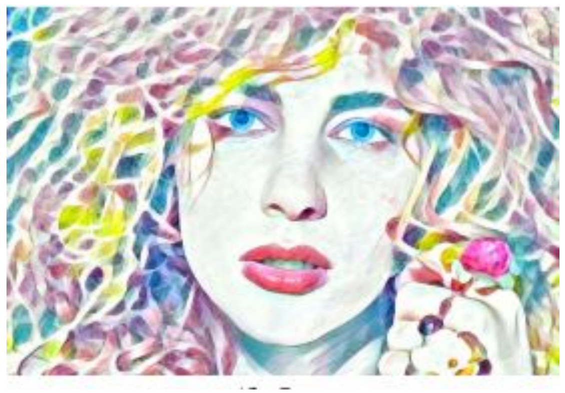
Content Image

Style Image









Output Image 1

Output Image 2

Output Image 3

Output Image 4

Technology Used

- ❖ Front end
- ❖ Backend
- ❖ Matplotlib
- ❖ Pytorch
- ❖ Tensorflow
- ❖ Numpy
- ❖ PIL(Python Imaging Library)

Literature Survey

- Paper title - Neural Style Palette: a multimodal and interactive style transfer from a single style image
- Author name - John Jethro Virtusio, Jose Jaena Marie Ople, Daniel Stanley Tan,M.Tanveer,Neeraj Kumar,and Kai-Lung Hua
- Journal Name -
- Year of publishing - 2021
- Followed by Summary of papers. -

The article discusses limitations in existing neural style transfer methods, which often produce only one stylized image from a single style input, offering limited variety and control to users. In response, the article introduces a method called Neural Type Palette (NSP). NSP is an interactive approach that generates multiple stylized images from a single style source, allowing users to influence the stylization process creatively.

NSP employs the concept of anchor styles, which act as visual guides for users. These anchor styles capture various attributes of the input image, each with a unique style. Users can blend these anchor styles to achieve their desired artistic results, offering diverse choices.

To maintain visual fidelity to the original style, NSP uses two new losses: a loss of separation to encourage distinct substructures in the stylized images and a loss of unity to ensure that the substructures cluster around the original style while introducing additional variety.

The article presents experimental results demonstrating the effectiveness of the NSP method and its potential to enhance existing neural style transfer techniques.

Literature Survey

- Paper title - Image Neural Style Transfer With Preserving the Salient Regions
- Author name - Yijun Liu , Zuoteng Xu , Wujian Ye, Ziwn Zhang , Shaowei Weng , Chin-Chen Chang ,Huajin Tang.
- Journal Name -
- Year of publishing - 2019
- Followed by Summary of papers. -

Neural style transfer has recently become one of the most popular topics in academic research and industrial applications. Existing methods can generate synthetic images by transferring different images the style of certain images compared to other images with certain content, but they are mainly focused on understanding low-level features images lose content and style, resulting in a significant change in the featured information of the content image semantic level. In this paper, an improved scheme is proposed to retain the salient regions of the transfer process the image is similar to the content image. By adding the area loss calculated from the location network, synthetic images can almost keep key highlight regions consistent with those of the original content image, facilitates saliency-based tasks such as object localization and classification. Additionally, the transfer the effect is more natural and attractive, avoiding the simple texture overlay of stylish images. Furthermore, ours the schema is also extensible to retain other semantic information (such as shape, edges, and color) of the image with the corresponding estimated networks.

Literature Survey

- Paper title - Guided neural style transfer for shape stylization
- Author name - Gantugs Atarsaikhan, Brian Kenji Iwana, Seiichi Uchida
- Journal Name -
- Year of publishing - 2020
- Followed by Summary of papers. -

Designing logos, typography, and other decorative shapes can require professional skills. In this article, we aim to create novel and unique decorative shapes by stylizing regular shapes through machine learning. Specifically, we combined parametric and non-parametric neural style transfer algorithms to transfer local and global features. Additionally, we have included distance-based instructions in the neural style transfer pipeline to decorate only the foreground shape. Finally, qualitative evaluation and ablation studies are provided to demonstrate the usefulness of the proposed method.

Literature Survey

- Paper title - Structure-Preserving Neural Style Transfer
- Author name - Ming-Ming Cheng , Xiao-Chang Liu , Jie Wang,
  Shao-Ping Lu , Yu-Kun Lai , and Paul L. Rosin
- Journal Name -
- Year of publishing - 2020
- Followed by Summary of papers. -

State-of-the-art neural convolution methods have demonstrated amazing results by training feed-forward convolutional neural networks or using iterative optimization strategies. The visual representation used in these methods, consists of two components: Style representation and content representation are often based on high-level features extracted from pre-trained classification networks. Because the classification network was originally designed for object recognition, the extracted features often focus on the central object and ignore it. Other details. As a result, style textures tend to scatter across styled outputs and break the content structure. To address this problem, we present a new image stylization method that involves complementary structure representation. Our structural representation, takes into account two factors:

i) the overall structure is represented by a depth map, and ii) the details of the local structure are represented by the edges of the image, effectively reflecting the spatial distribution of all components of a image as well as the structure of the corresponding dominant objects. Experimental results demonstrate that our method achieves impressive imaging performance, which is especially important when processing images sensitive to structural deformation, e.g. images contain multiple objects potentially at different depths or prominent objects with clear structures.

Literature Survey

- Paper title - Image Style Transfer Using Convolutional Neural Networks
- Author name - Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge
- Journal Name -
- Year of publishing - 2016
- Followed by Summary of papers. -

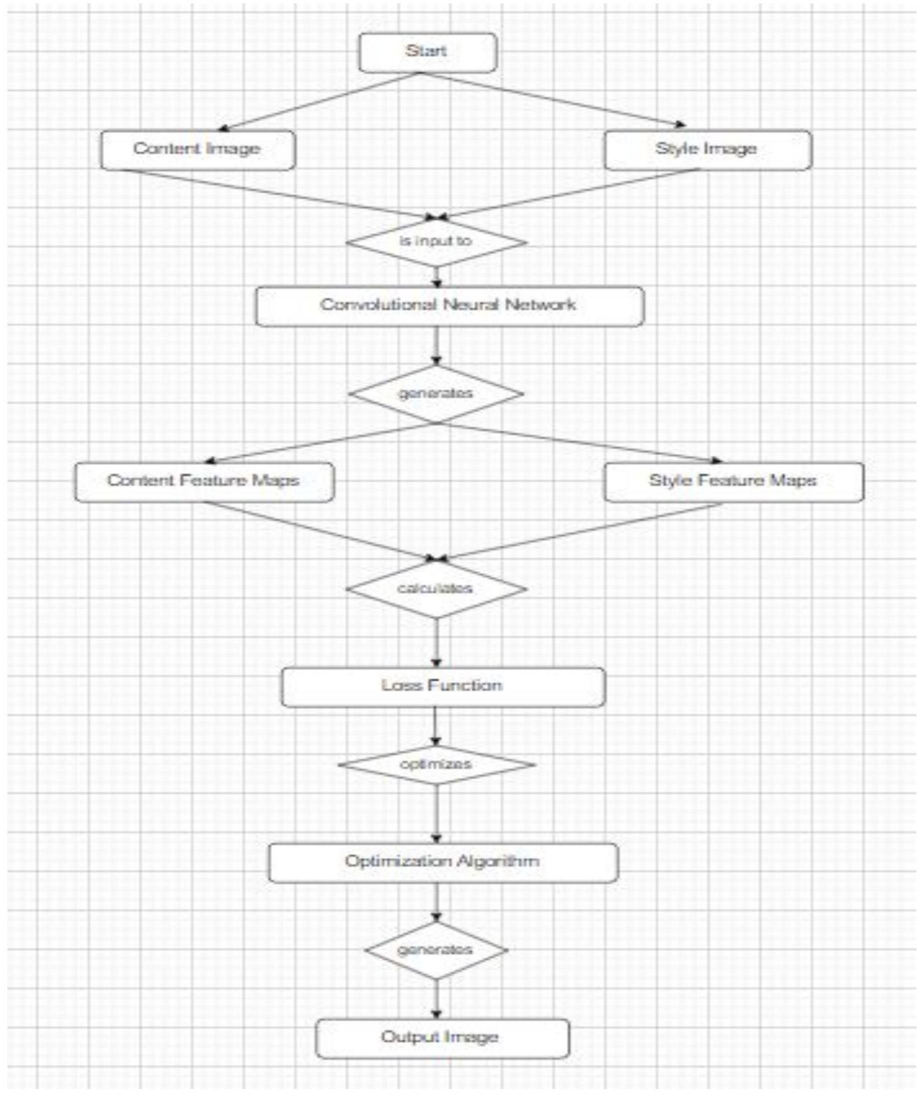
Displays the semantic content of an image in different ways style is a difficult image processing task. Without a doubt, a major one the limiting factor of previous approaches is lack visual representation explicitly represents semantic information and thus allows separation of visual content from style. Here, we use image representations derived from an optimized convolutional neural network for object recognition, which helps clarify high-level visual information. We introduces an art-style neural algorithm that can separate and recombine visual content and style from the wild pictures. The algorithm allows us to generate new images about High perceptual quality combines the content of an arbitrary photograph with the form of many famous works of art. Our results provide new information about Deep image representations are learned by convolutional neural networks and demonstrate their high-level potential synthesize and process images.

Literature Survey

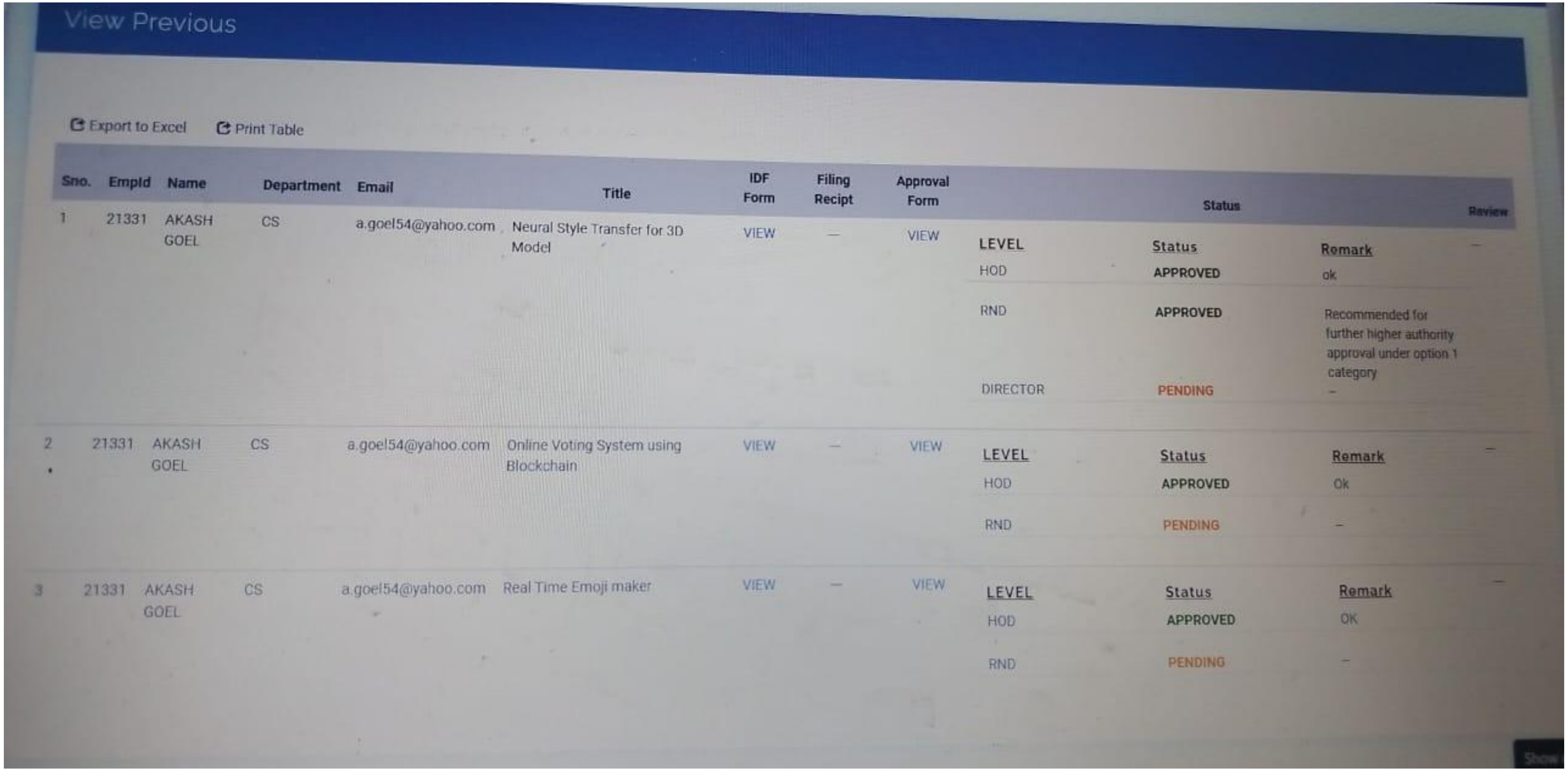
- Paper title - Deeply Supervised Salient Object Detection with Short Connections
- Author name - Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, Philip H. S. Torr
- Journal Name -
- Year of publishing - 2016
- Followed by Summary of papers. -

Recent advances in prominent object detection have been significant, largely benefiting from the explosive growth of Convolutional Neural Network (CNN). Semantic segmentation and salient object detection algorithms have been developed recently mainly based on fully convolutional neural networks (FCN). There's still a lot of room for improvement compared to the generic product FCN models do not explicitly address the problem of scale space. The comprehensive nested edge detector (HED) provides a bypass framework with deep supervision for edge and boundary detection, but the performance gain of HED when detecting saliency is not easy. In this paper, we propose a new salient object detection method by introducing short connections to the omitted layer. structures in HED architecture. Our framework takes full advantage of the multi-level and multi-scale features extracted from FCN,provides more advanced representations at each layer, an essential,property for performing segmentation detection. Our method produces state-of-the-art results based on 5 widely tested prominent object detection benchmarks, with benefits in terms of efficient (0.08 seconds per image), efficient and simple compared to existing algorithms. Additionally, we are leading one Comprehensive analysis of the role training data plays in performance. Our experimental results provide a more reasonable and robust approach training set for future research and fair comparison.

Workflow Diagram



Patent Status

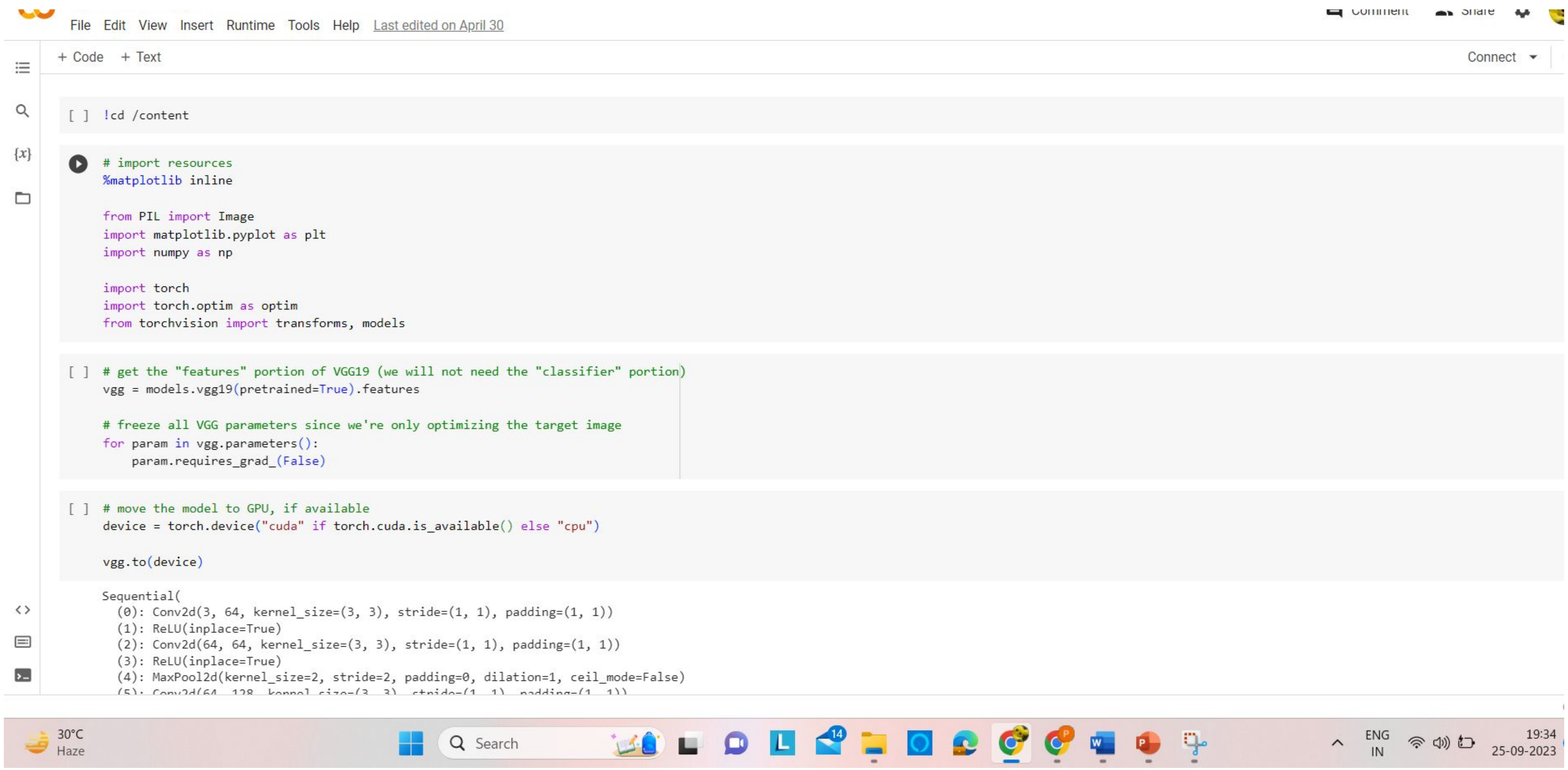


View Previous

Export to Excel Print Table

Sno.	EmpId	Name	Department	Email	Title	IDF Form	Filing Recipt	Approval Form	Status			Review
									LEVEL	Status	Remark	
1	21331	AKASH GOEL	CS	a.goel54@yahoo.com	Neural Style Transfer for 3D Model	VIEW	–	VIEW	HOD	APPROVED	ok	–
									RND	APPROVED	Recommended for further higher authority approval under option 1 category	
									DIRECTOR	PENDING	–	
2	21331	AKASH GOEL	CS	a.goel54@yahoo.com	Online Voting System using Blockchain	VIEW	–	VIEW	HOD	APPROVED	Ok	–
									RND	PENDING	–	
3	21331	AKASH GOEL	CS	a.goel54@yahoo.com	Real Time Emoji maker	VIEW	–	VIEW	HOD	APPROVED	OK	–
									RND	PENDING	–	

Project Status



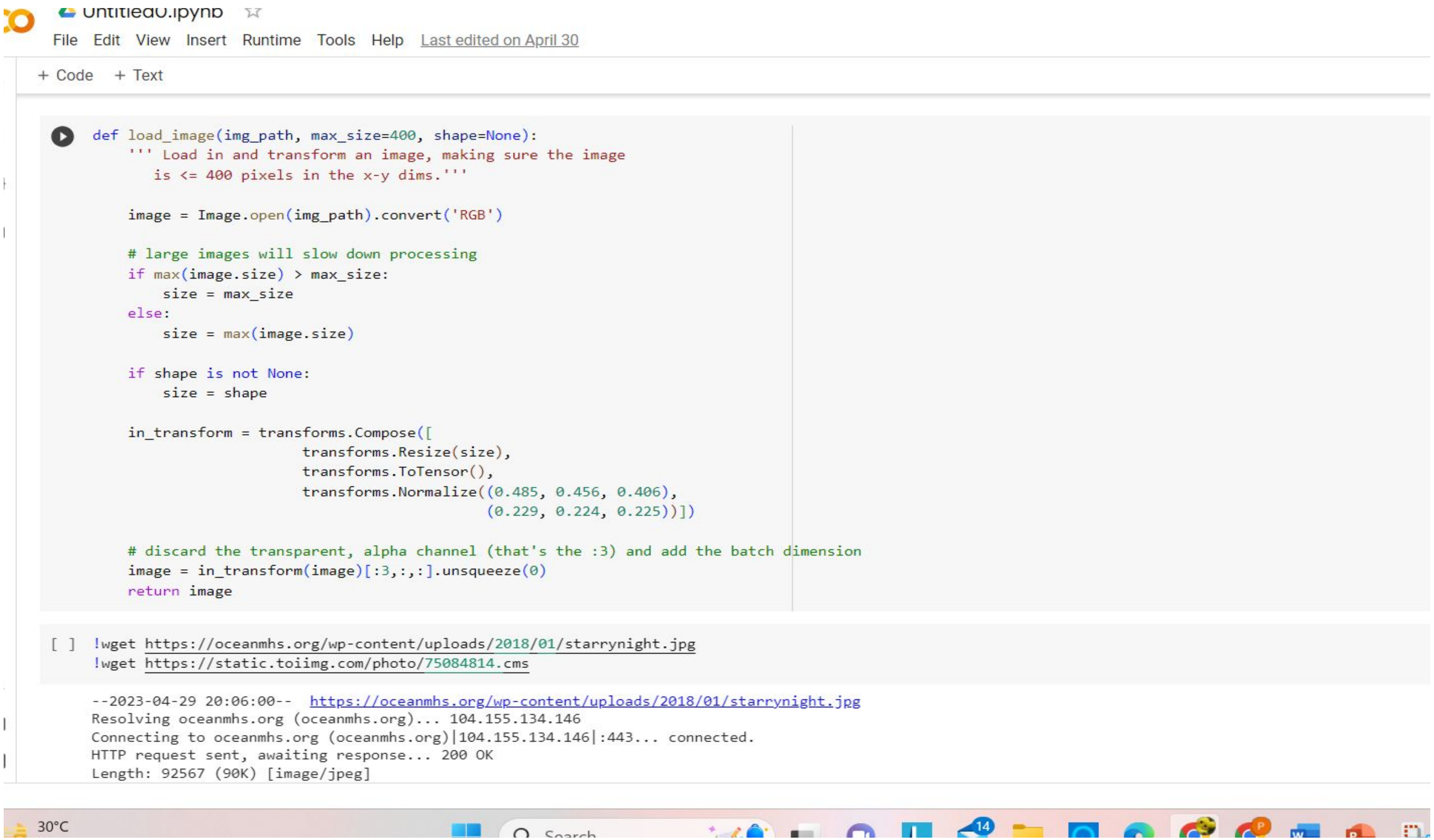
```
!cd /content
```

```
# import resources
%matplotlib inline

from PIL import Image
import matplotlib.pyplot as plt
import numpy as np

import torch
import torch.optim as optim
from torchvision import transforms, models
```

```
# get the "features" portion of VGG19 (we will not need the "classifier" portion)
vgg = models.vgg19(pretrained=True).features

# freeze all VGG parameters since we're only optimizing the target image
for param in vgg.parameters():
    param.requires_grad_(False)
```

```
# move the model to GPU, if available
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

vgg.to(device)
```

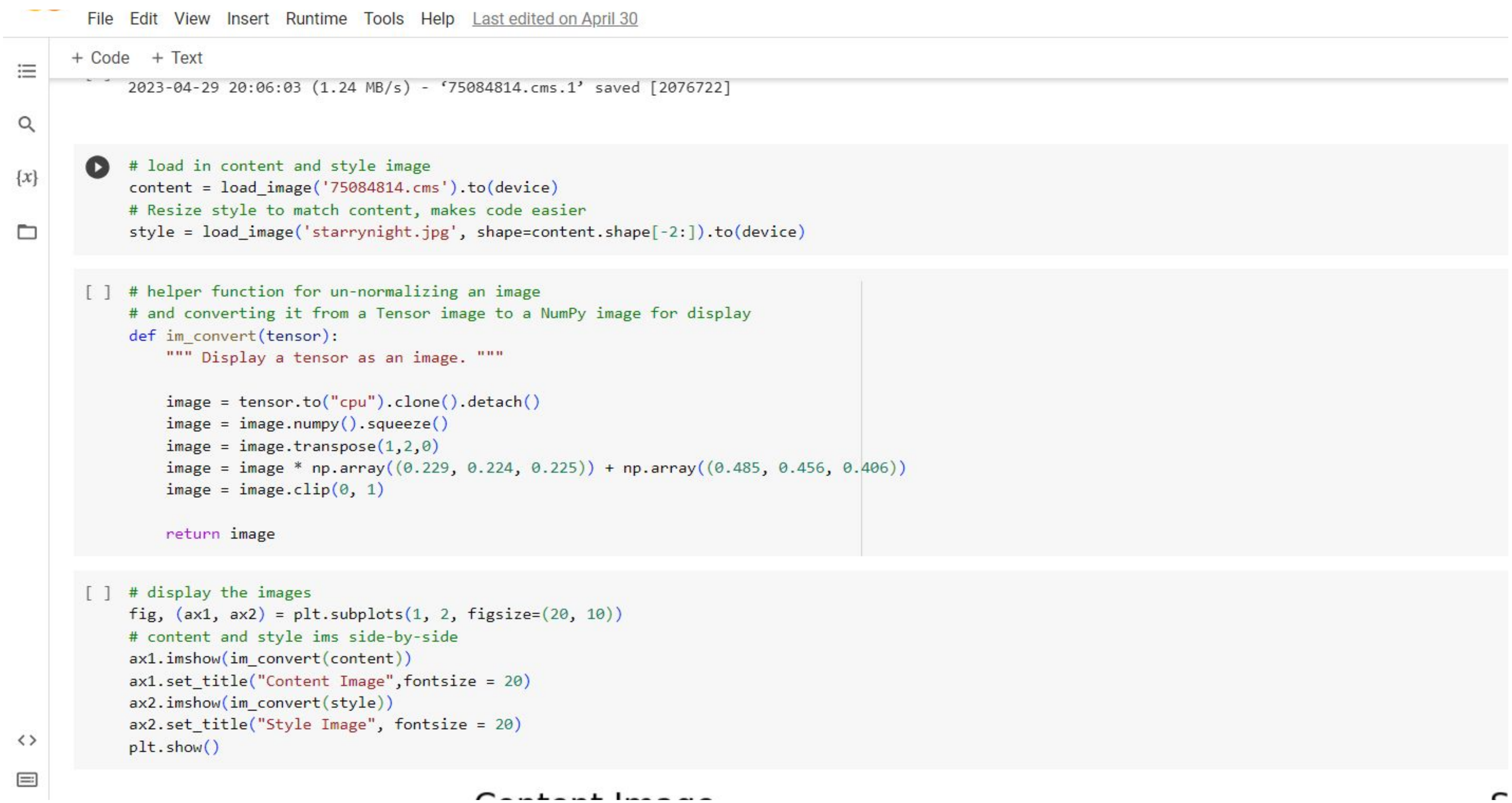
```
Sequential(
  (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (1): ReLU(inplace=True)
  (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (3): ReLU(inplace=True)
  (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
```


Untitled0.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 30
+ Code + Text

```
def load_image(img_path, max_size=400, shape=None):
    ''' Load in and transform an image, making sure the image
       is <= 400 pixels in the x-y dims.'''

    image = Image.open(img_path).convert('RGB')

    # large images will slow down processing
    if max(image.size) > max_size:
        size = max_size
    else:
        size = max(image.size)

    if shape is not None:
        size = shape

    in_transform = transforms.Compose([
                        transforms.Resize(size),
                        transforms.ToTensor(),
                        transforms.Normalize((0.485, 0.456, 0.406),
                                             (0.229, 0.224, 0.225))])

    # discard the transparent, alpha channel (that's the :3) and add the batch dimension
    image = in_transform(image)[:3,:,:].unsqueeze(0)
    return image
```

```
[ ] !wget https://oceanmhs.org/wp-content/uploads/2018/01/starrynight.jpg
    !wget https://static.toiimg.com/photo/75084814.cms
```

```
--2023-04-29 20:06:00--  https://oceanmhs.org/wp-content/uploads/2018/01/starrynight.jpg
Resolving oceanmhs.org (oceanmhs.org)... 104.155.134.146
Connecting to oceanmhs.org (oceanmhs.org)|104.155.134.146|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 92567 (90K) [image/jpeg]
```

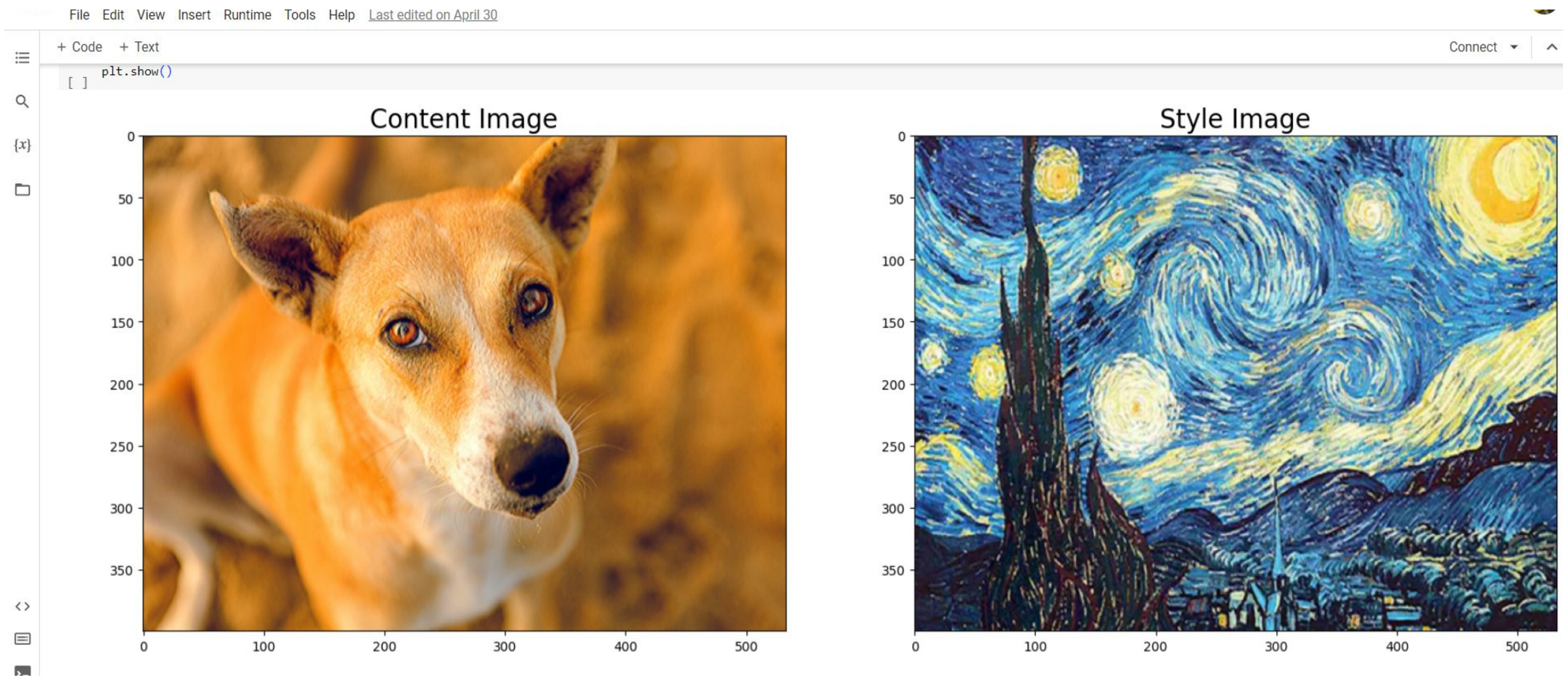


```
2023-04-29 20:06:03 (1.24 MB/s) - ‘75084814.cms.1’ saved [2076722]
```

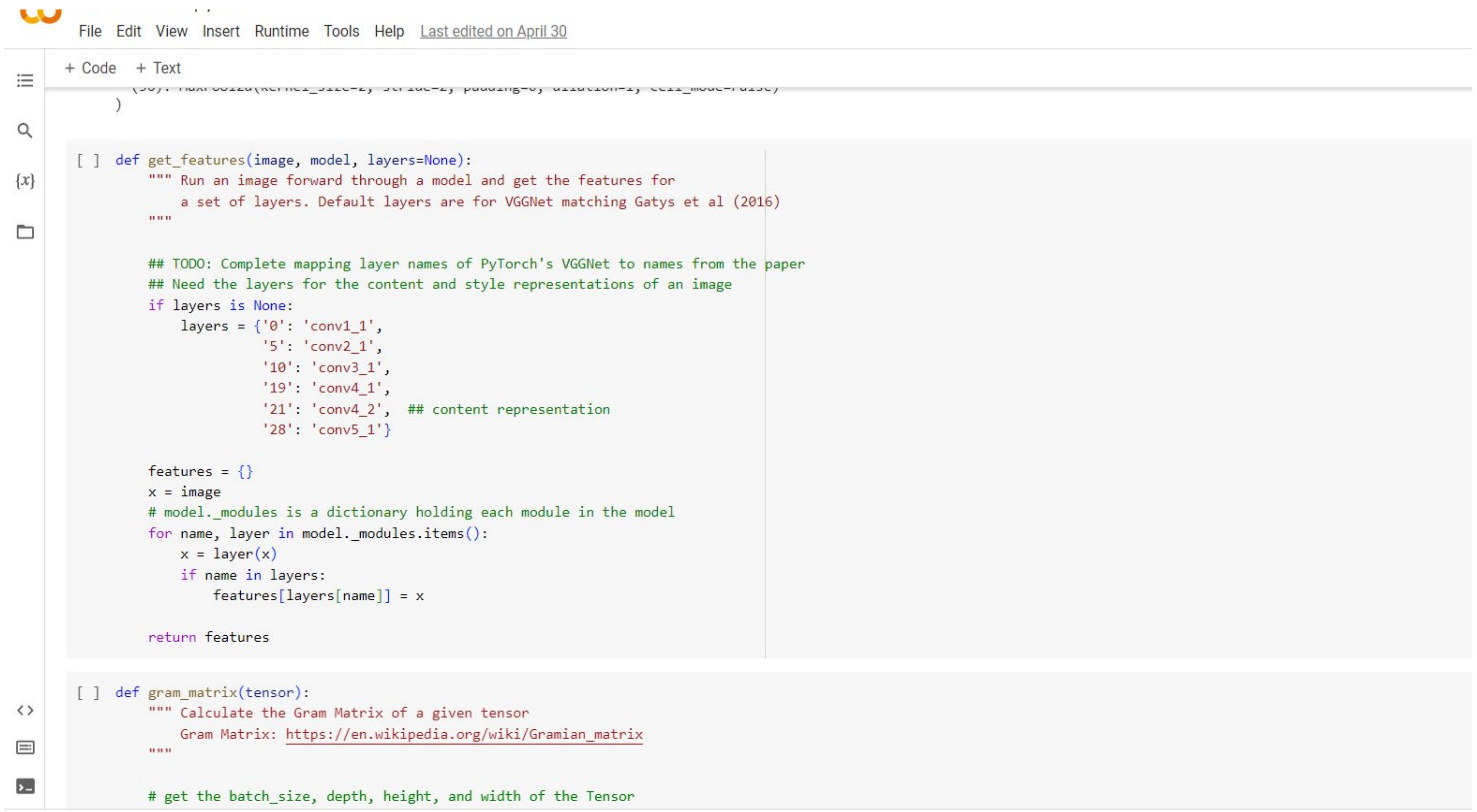
```python
# load in content and style image
content = load_image('75084814.cms').to(device)
# Resize style to match content, makes code easier
style = load_image('starrynight.jpg', shape=content.shape[-2:]).to(device)
```

```python
[ ] # helper function for un-normalizing an image
    # and converting it from a Tensor image to a NumPy image for display
    def im_convert(tensor):
        """ Display a tensor as an image. """

        image = tensor.to("cpu").clone().detach()
        image = image.numpy().squeeze()
        image = image.transpose(1,2,0)
        image = image * np.array((0.229, 0.224, 0.225)) + np.array((0.485, 0.456, 0.406))
        image = image.clip(0, 1)

        return image
```

```python
[ ] # display the images
    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(20, 10))
    # content and style ims side-by-side
    ax1.imshow(im_convert(content))
    ax1.set_title("Content Image",fontsize = 20)
    ax2.imshow(im_convert(style))
    ax2.set_title("Style Image", fontsize = 20)
    plt.show()
```


File
Edit
View
Insert
Runtime
Tools
Help
Last edited on April 30
+ Code
+ Text
Connect
Content Image
Style Image

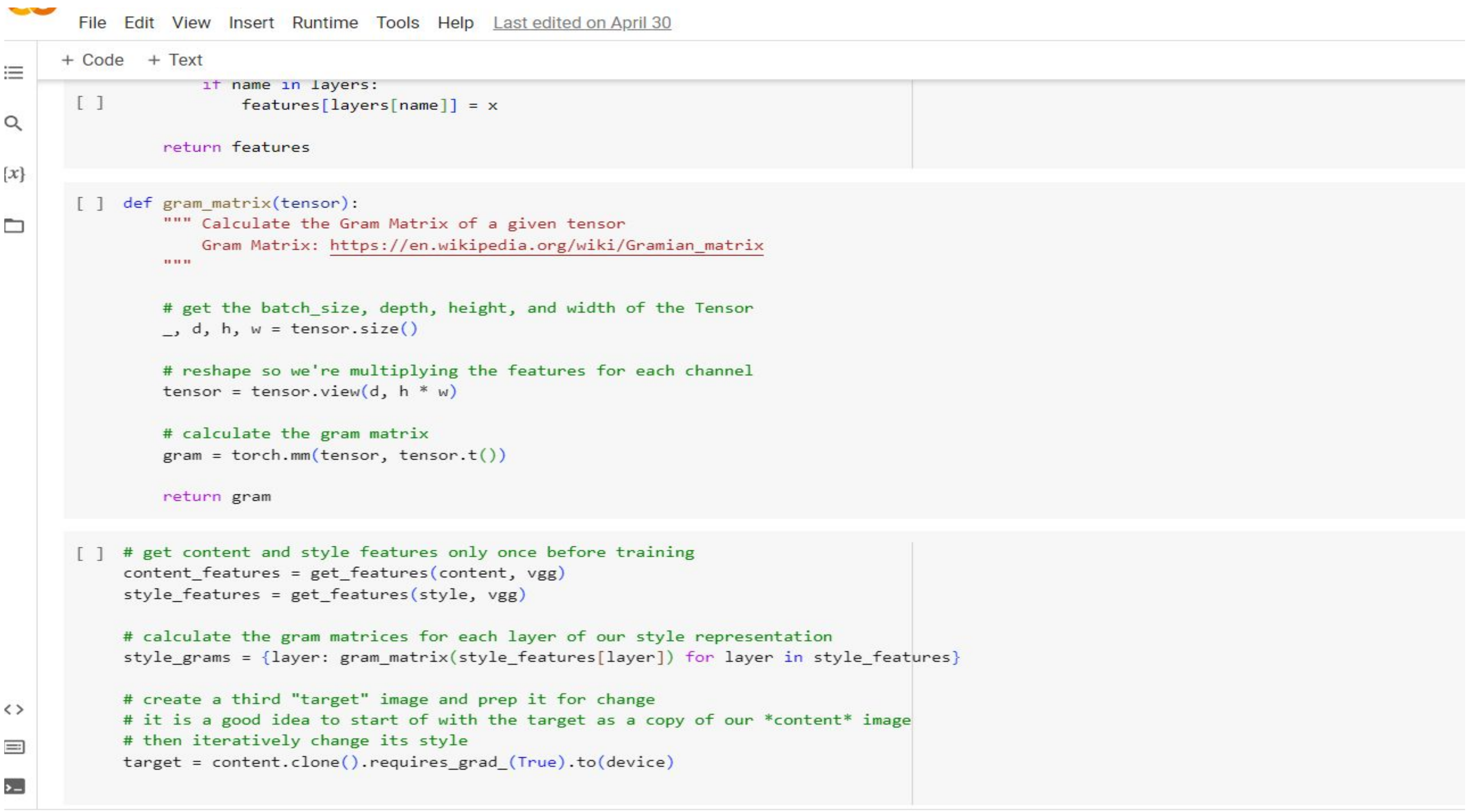
```
plt.show()
```



```
    )
```

```
def get_features(image, model, layers=None):
    """ Run an image forward through a model and get the features for
        a set of layers. Default layers are for VGGNet matching Gatys et al (2016)
    """

    ## TODO: Complete mapping layer names of PyTorch's VGGNet to names from the paper
    ## Need the layers for the content and style representations of an image
    if layers is None:
        layers = {'0': 'conv1_1',
                  '5': 'conv2_1',
                  '10': 'conv3_1',
                  '19': 'conv4_1',
                  '21': 'conv4_2',  ## content representation
                  '28': 'conv5_1'}

    features = {}
    x = image
    # model._modules is a dictionary holding each module in the model
    for name, layer in model._modules.items():
        x = layer(x)
        if name in layers:
            features[layers[name]] = x

    return features
```

```
def gram_matrix(tensor):
    """ Calculate the Gram Matrix of a given tensor
        Gram Matrix: https://en.wikipedia.org/wiki/Gramian_matrix
    """

    # get the batch_size, depth, height, and width of the Tensor
```



```
        if name in layers:
            features[layers[name]] = x

    return features
```

```
def gram_matrix(tensor):
    """ Calculate the Gram Matrix of a given tensor
        Gram Matrix: https://en.wikipedia.org/wiki/Gramian_matrix
    """

    # get the batch_size, depth, height, and width of the Tensor
    _, d, h, w = tensor.size()

    # reshape so we're multiplying the features for each channel
    tensor = tensor.view(d, h * w)

    # calculate the gram matrix
    gram = torch.mm(tensor, tensor.t())

    return gram
```

```
# get content and style features only once before training
content_features = get_features(content, vgg)
style_features = get_features(style, vgg)

# calculate the gram matrices for each layer of our style representation
style_grams = {layer: gram_matrix(style_features[layer]) for layer in style_features}

# create a third "target" image and prep it for change
# it is a good idea to start of with the target as a copy of our *content* image
# then iteratively change its style
target = content.clone().requires_grad_(True).to(device)
```

Research Paper Status

- Research paper under progress

Image Style Adaptation

Given Name Surname
dept. name of organization
(of Affiliation)
name of organization
(of Affiliation)
City, Country
email address or ORCID

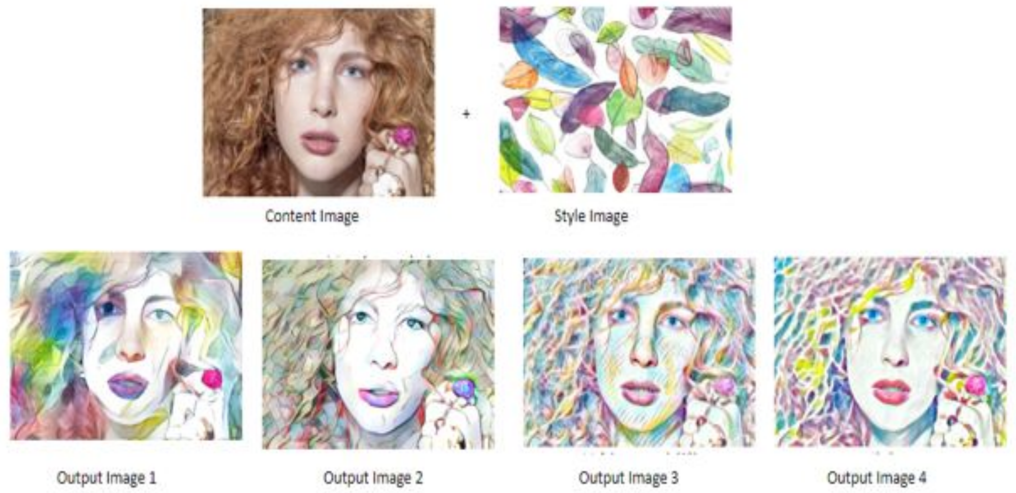
Abstract

Introduction -

Neural style transfer is a deep learning technique that involves combining the content of one image with the style of another image to create artistic images [1] [2]. This technique has attracted significant attention in recent years as it offers an interesting solution for artists to explore their creativity [1]. The effectiveness of Neural Style Transfer in generating artistic images is an area being explored by researchers and practitioners in the field of deep learning [1]. This approach separates style and content representation using a convolutional neural network (CNN), then applies visual transformation to the content and style using deep learning [1]. The reference style is faithfully transferred to the stylized image and the approach handles a variety of image content [2]. To create stylized images, a transfer network is used, which is an image translation network that takes one image as input and outputs another image [1]. This network typically has an encoder-decoder architecture and uses a pre-trained model to compare the content and style of two images [1]. The neural style transfer technique uses a pre-trained feature extractor to save the output in style and content layers for future comparison [1]. The styled image is also run through a feature extractor and the output in the content and style layers is saved [1]. Then, the content images are passed through a pre-trained feature extractor and their output at different content layers is recorded [1]. The transmission network then creates a stylized image from the content image [1]. The goal of Neural Style Transfer is to preserve the content of the original image while applying the visual style of another image [2]. The blending process requires specific inputs determined by a pre-trained convolution model and loss functions to achieve the desired visual effect [1].

Early methods in salient object detection focused on contrast-based approaches inspired by cognitive studies of visual attention. However, these methods had limitations and struggled with complex scenes. To overcome these limitations, learning-based approaches, especially convolutional neural networks (CNN), have been introduced. Fully convolutional neural networks (FCNs) have provided a comprehensive learning framework that significantly improves salient object detection. This paper presents a new approach for salient object detection by combining layer skipping structures with deep supervision. Unlike traditional methods that focus on merging features from different scales, this method takes a top-down perspective. He observed that deeper layers encode high-level semantic knowledge, while shallower layers capture rich spatial information. By connecting these layers through short connections, this method combines features at multiple levels, creating rich multi-scale feature maps at each layer. This approach improves the accuracy of the salinity map [3].

Multiple Style transfer:-



References

- John Jethro Virtusio, Jose Jaena Marie Ople, Daniel Stanley Tan,M.Tanveer,Neeraj Kumar,and Kai-Lung Hua, “ Image Neural Style Transfer With Preserving the Salient Regions” IEEE Transactions on multimedia, 2021.
- Yijun Liu , Zuoteng Xu , Wujian Ye, Ziwn Zhang , Shaowei Weng , Chin-Chen Chang , and Huajin Tang, “ Image Neural Style Transfer With Preserving the Salient Regions ” Digital Object Identifier, 2019.
- Gantugs Atarsaikhan, Brian Kenji Iwana, and Seiichi Uchida, “ Guided neural style transfer for shape stylization”, Department of Advanced Information Technology, Kyushu University, 2021

- Ming-Ming Cheng , Xiao-Chang Liu , Jie Wang, Shao-Ping Lu , Yu-Kun Lai , and Paul L. Rosin, “Structure-Preserving Neural Style Transfer, IEEE Transactions on image processing, 2020 .
- Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge, “ Image Style Transfer Using Convolutional Neural Networks”, Image Style Transfer Using Convolutional Neural Networks, 2016.
- Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, Philip H. S. Torr, “ Deeply Supervised Salient Object Detection with Short Connections”, IEEE Transactions on pattern analysis and machine intelligence, 2018.