STATEMENT:
CREATE AN IMAGE STYLE TRANSFER PROJECT (NEURAL STYLE TRANSFER) AND CREATE IMAGES

Neural Style Transfer (NST) is a technique in artificial intelligence (AI) and deep learning that allows you to combine the artistic style of one image with the content of another image.

In other words, it merges the content of an image (like a photograph) with the style of another image (like a painting), creating a new image that has the content of the first and the artistic style of the second.

This process is typically achieved through a convolutional neural network (CNN), which is trained to separate the content and style of an image. The steps involved generally include:

- ***Content image:*** This is the image whose structure or subject you want to preserve (e.g., a photograph of a landscape).

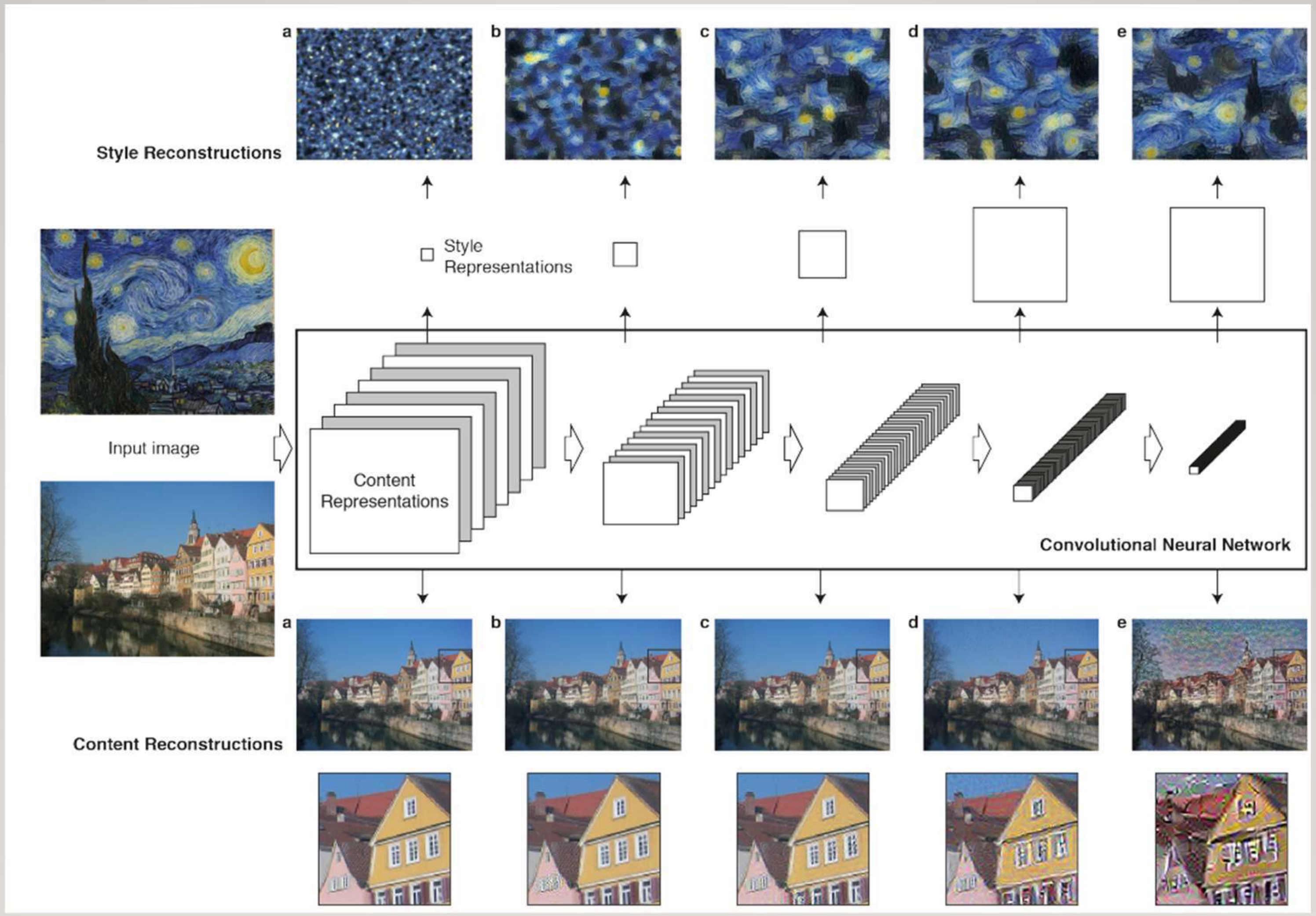
Style image: This is the image whose artistic style (e.g., brush strokes, colors, textures) you want to apply to the content image (e.g., a painting by Van Gogh).

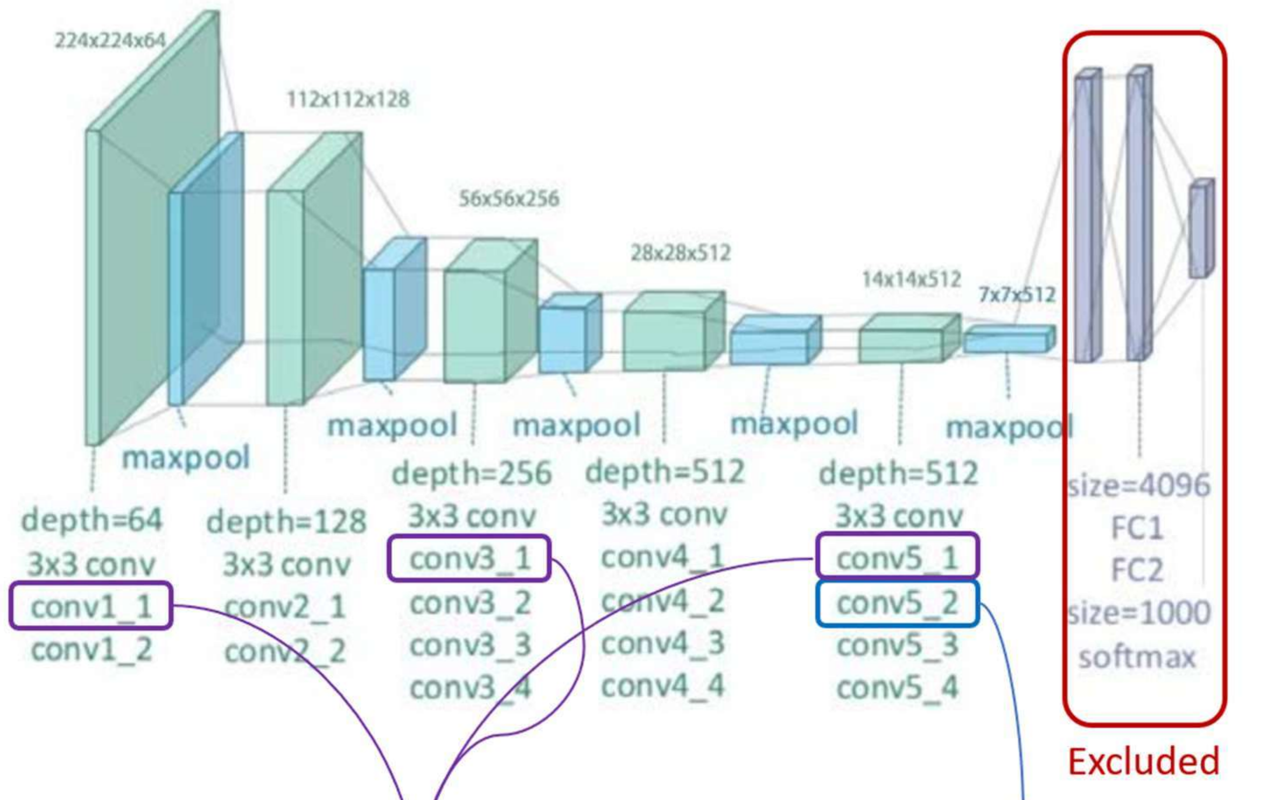
Output image: This is the final image generated by combining the content of the content image and the style of the style image.

The neural network learns the content and style representations through layers of the CNN.

It then optimizes a new image to match the content of the first image and the style of the second by minimizing a loss function that measures how different the generated image is from the original content and style.

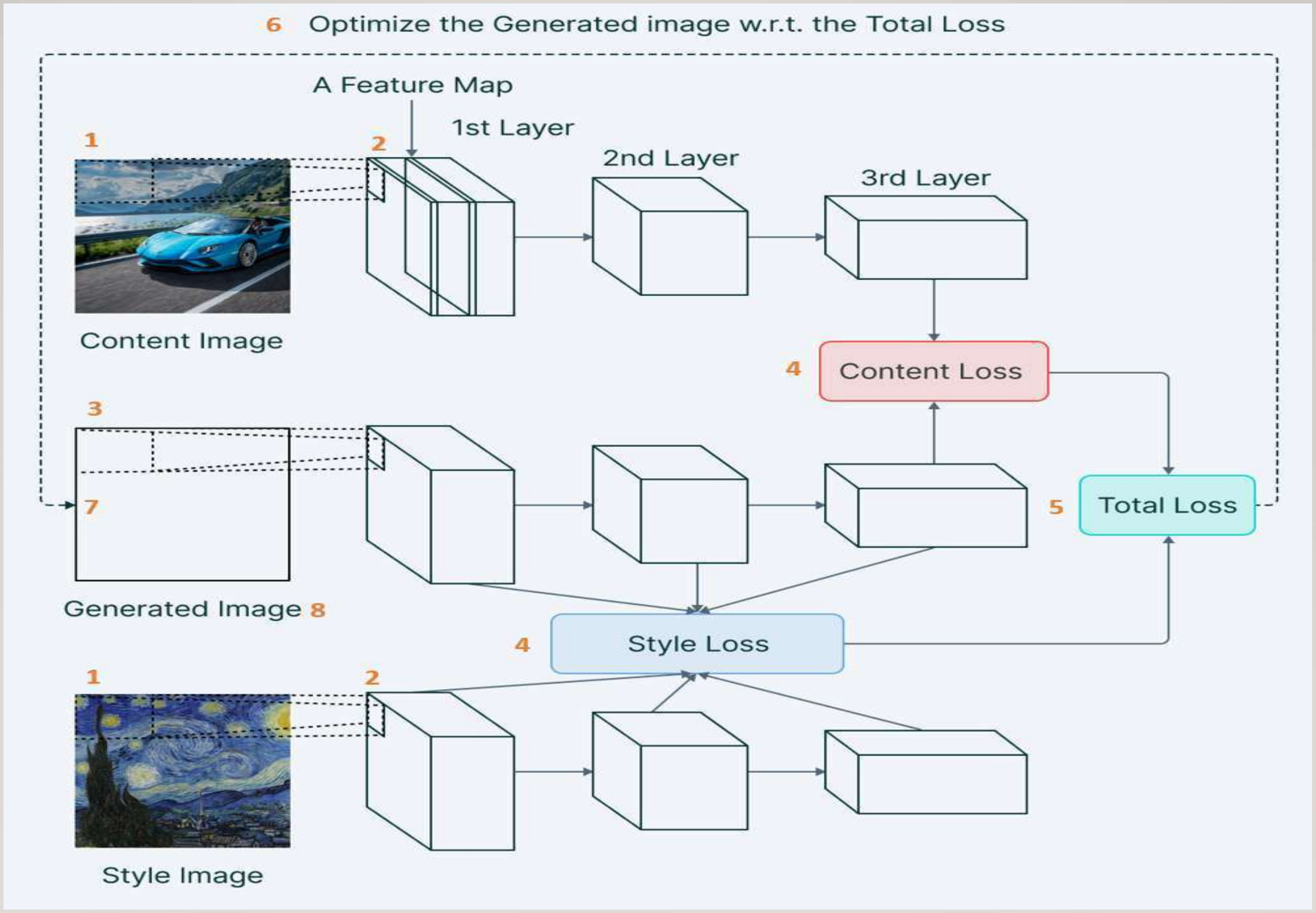
In practical terms, neural style transfer is used in apps and tools to turn photos into images that look like famous paintings or to create unique, artistic transformations of photos.


Style Reconstructions
a
b
c
d
e
Style Representations
Input image
Content Representations
Convolutional Neural Network
Content Reconstructions

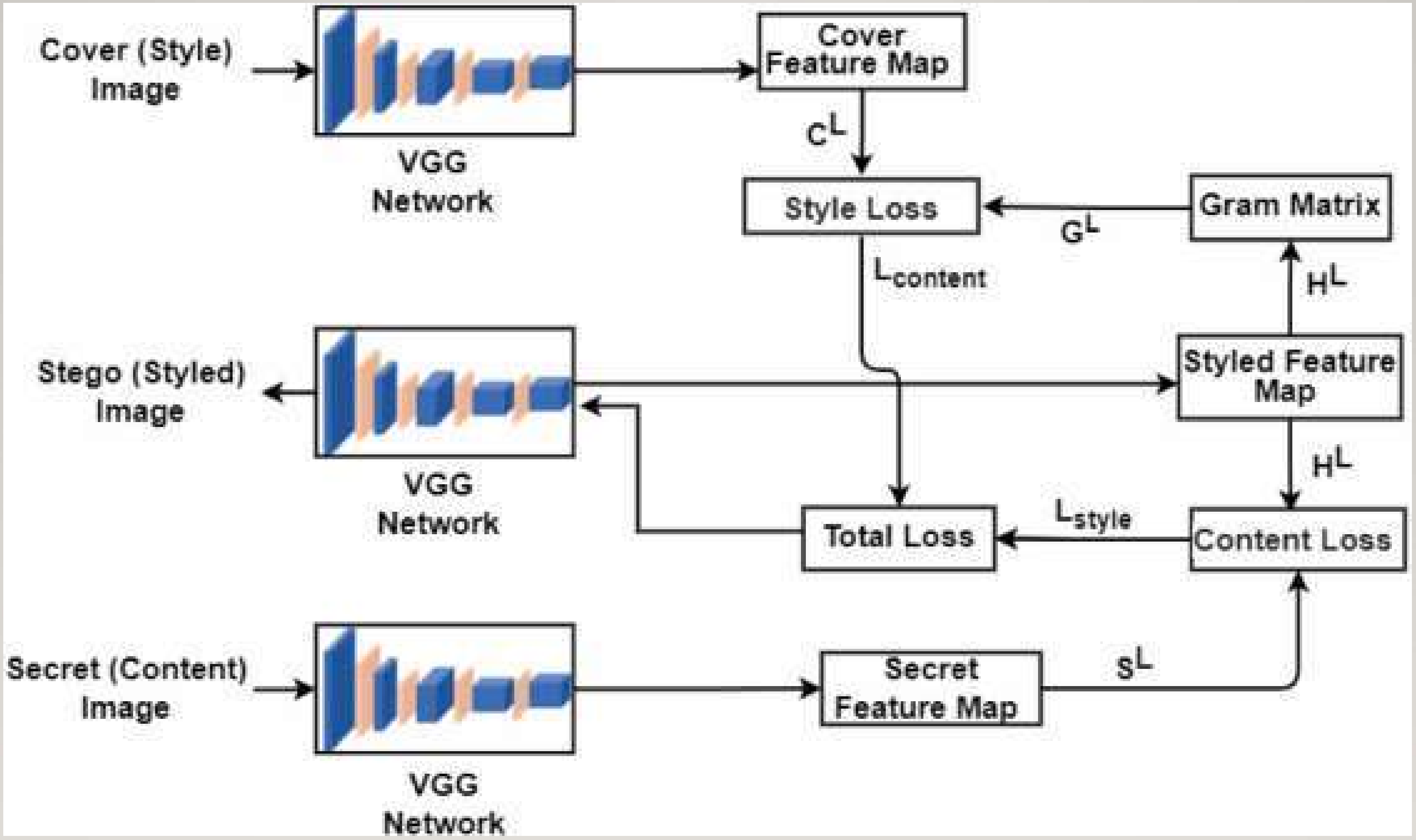


Intermediate Layer to extract from content image

Intermediate Layers to extract from style image

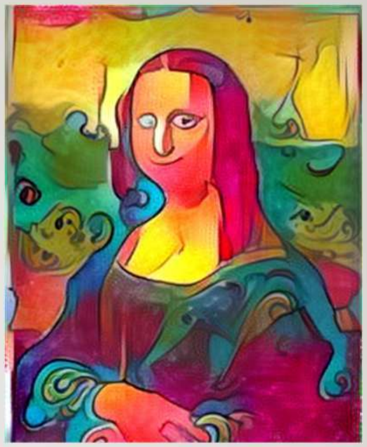


6 Optimize the Generated image w.r.t. the Total Loss
A Feature Map
1st Layer
2nd Layer
3rd Layer
1
2
Content Image
4 Content Loss
3
7
5 Total Loss
Generated Image 8
4 Style Loss
1
2
Style Image


Cover (Style) Image
VGG Network
Cover Feature Map
C^L
Style Loss
Gram Matrix
G^L
$L_{content}$
H^L
Stego (Styled) Image
VGG Network
Styled Feature Map
H^L
Total Loss
L_{style}
Content Loss
Secret (Content) Image
VGG Network
Secret Feature Map
S^L

Content Image

Style Image



Neural Style Transfer

A brief step-by-step process for the Neural Style Transfer (NST) project:

1. Set Up Environment

- Install libraries: tensorflow, numpy, matplotlib, PIL, opencv-python.
- Use Jupyter/Colab for an easy setup and GPU support.

2. Load and Preprocess Images

- Load the content image (photo) and style image (painting).
- Preprocess images to match the input size and normalize pixel values.

3. Load Pre-trained Model

- Use a pre-trained Convolutional Neural Network (CNN) like VGG19, which is available in TensorFlow.
- Load the model without the final classification layer to extract features.

4. Define the Loss Functions

- ***Content Loss:*** Measures the difference between content features in the content image and the generated image.
- ***Style Loss:*** Measures the difference in style (texture, patterns) between the style image and the generated image.
- ***Total Loss:*** A weighted sum of content loss and style loss, plus any regularization.

5. Optimization Loop

- Use gradient descent (like Adam optimizer) to minimize the total loss.
- The image is updated iteratively to transfer the style while preserving content.

6. Generate Output Image

- Initialize a random image or use the content image as the starting point.
- Apply optimization to produce the final stylized image.

7. Tune Parameters

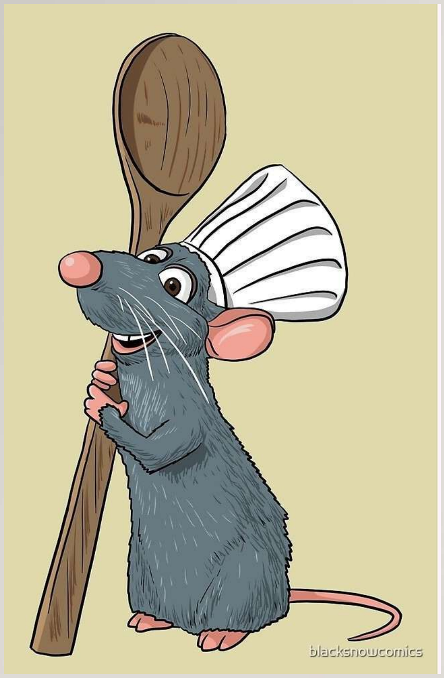
- Experiment with the weight between content loss and style loss to achieve the desired result.
- Adjust learning rate and number of iterations.

8. Display and Save Results

- Visualize the stylized image and compare it with the original content and style images.
- Save the output image.

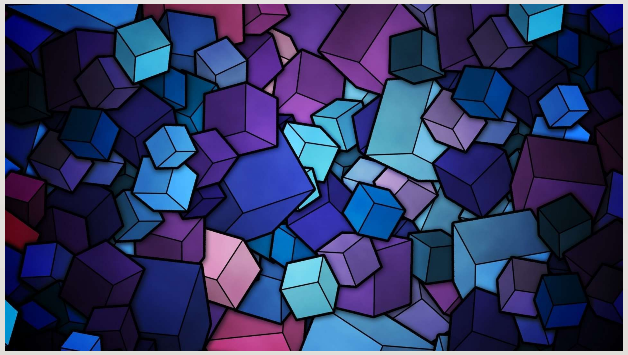
Results:

Content Image



+

Style Image



=

Generated Image