BILKENT UNIVERSİTY SPRING 2023 - CS 484

Neural Style Transfer Project

Tugberk Dikmen - 21802480

What is Convolutional Neural Networks ?

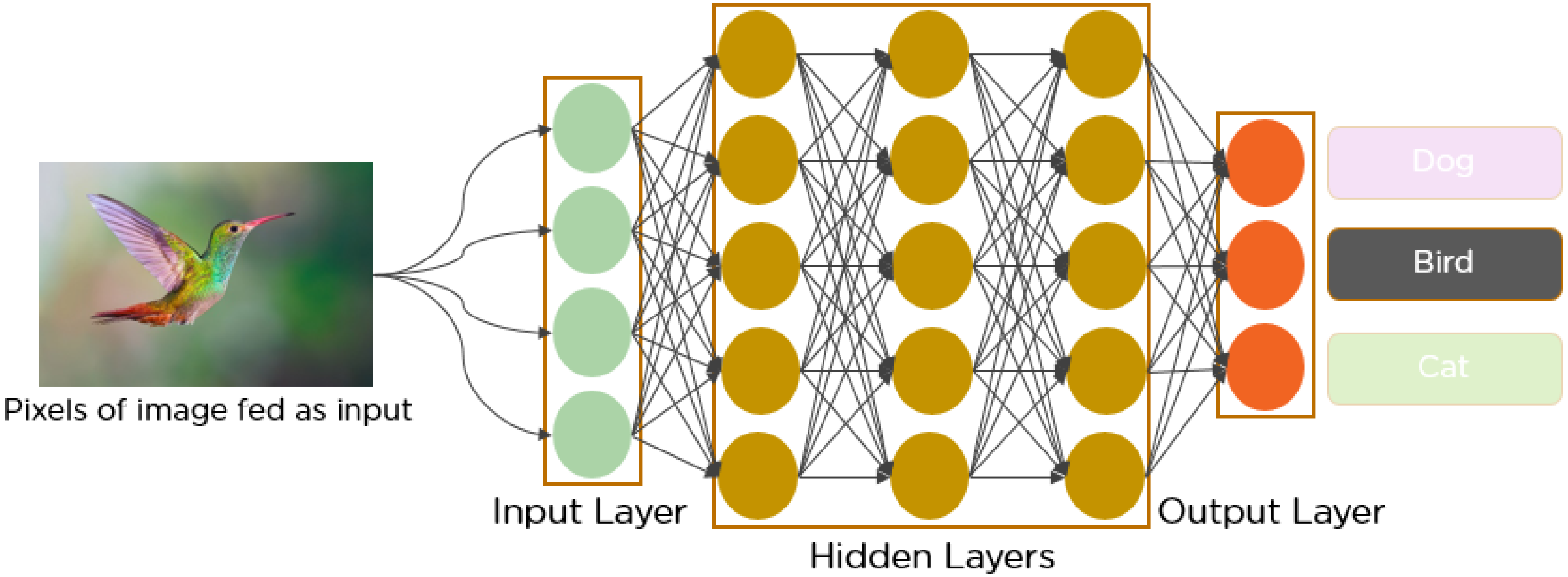


Figure 1: CNN[4]

What is Neural Style Transfer ?

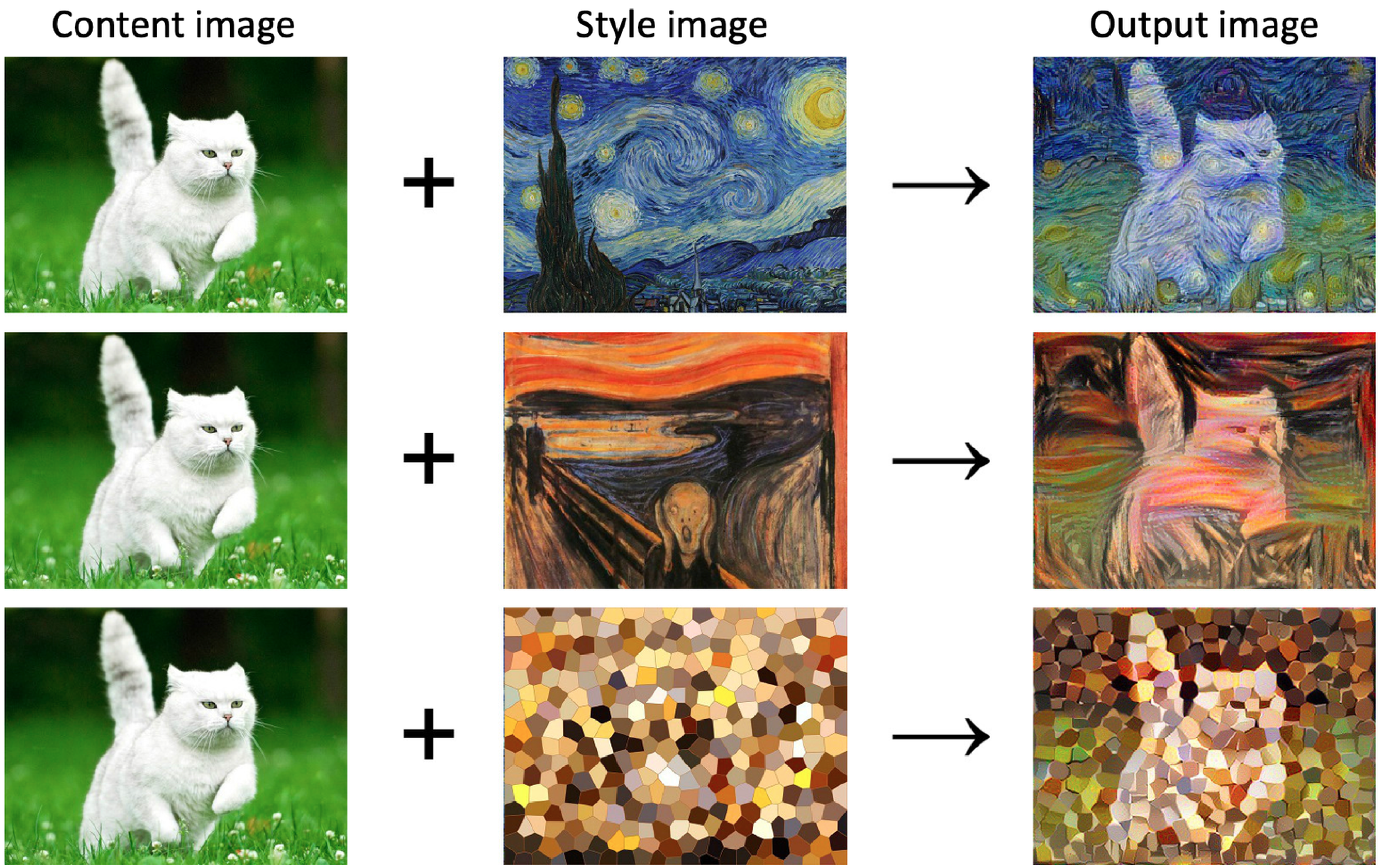
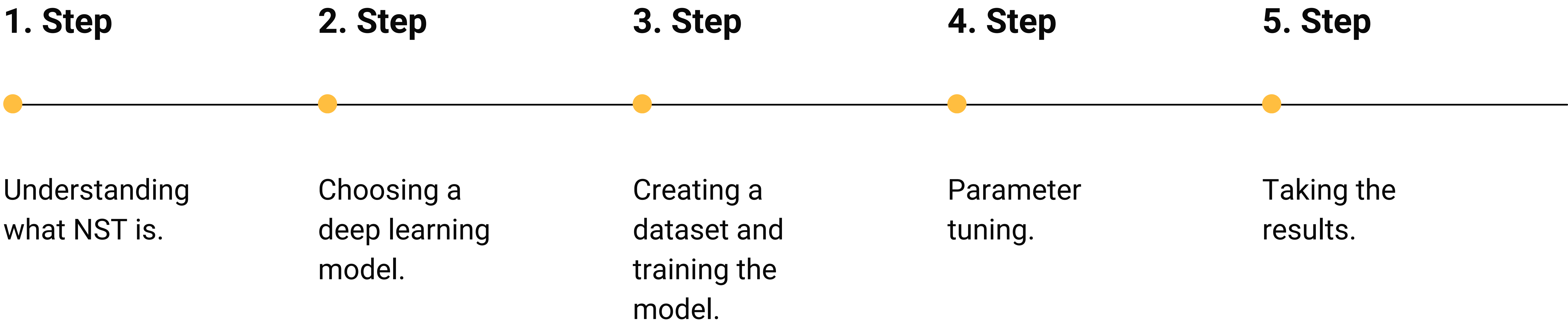


Figure 2: NST[3]

How to Implement a NST Program ?



1. Step

Understanding what NST is.

2. Step

Choosing a deep learning model.

3. Step

Creating a dataset and training the model.

4. Step

Parameter tuning.

5. Step

Taking the results.

VGG-19 Deep Learning Model

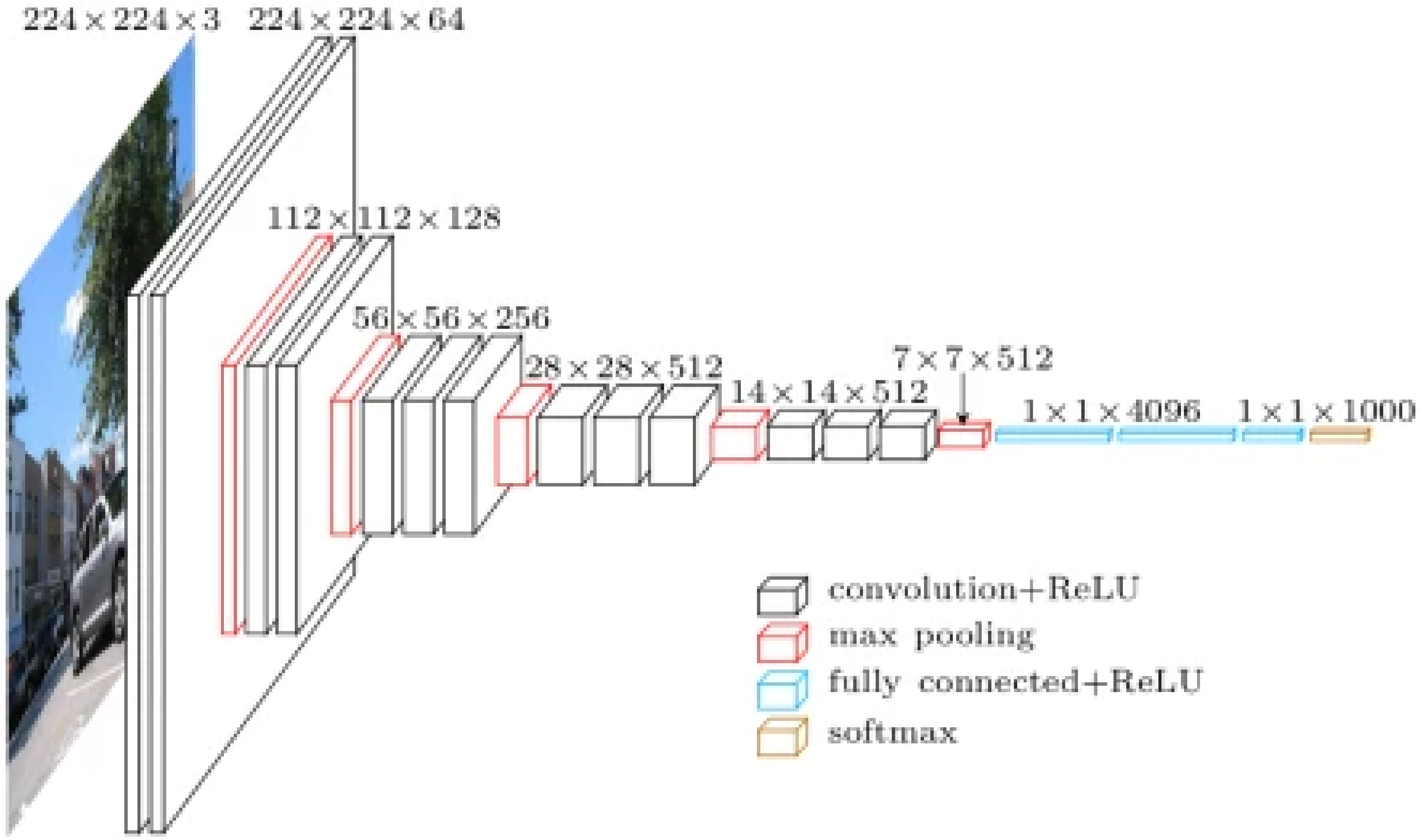


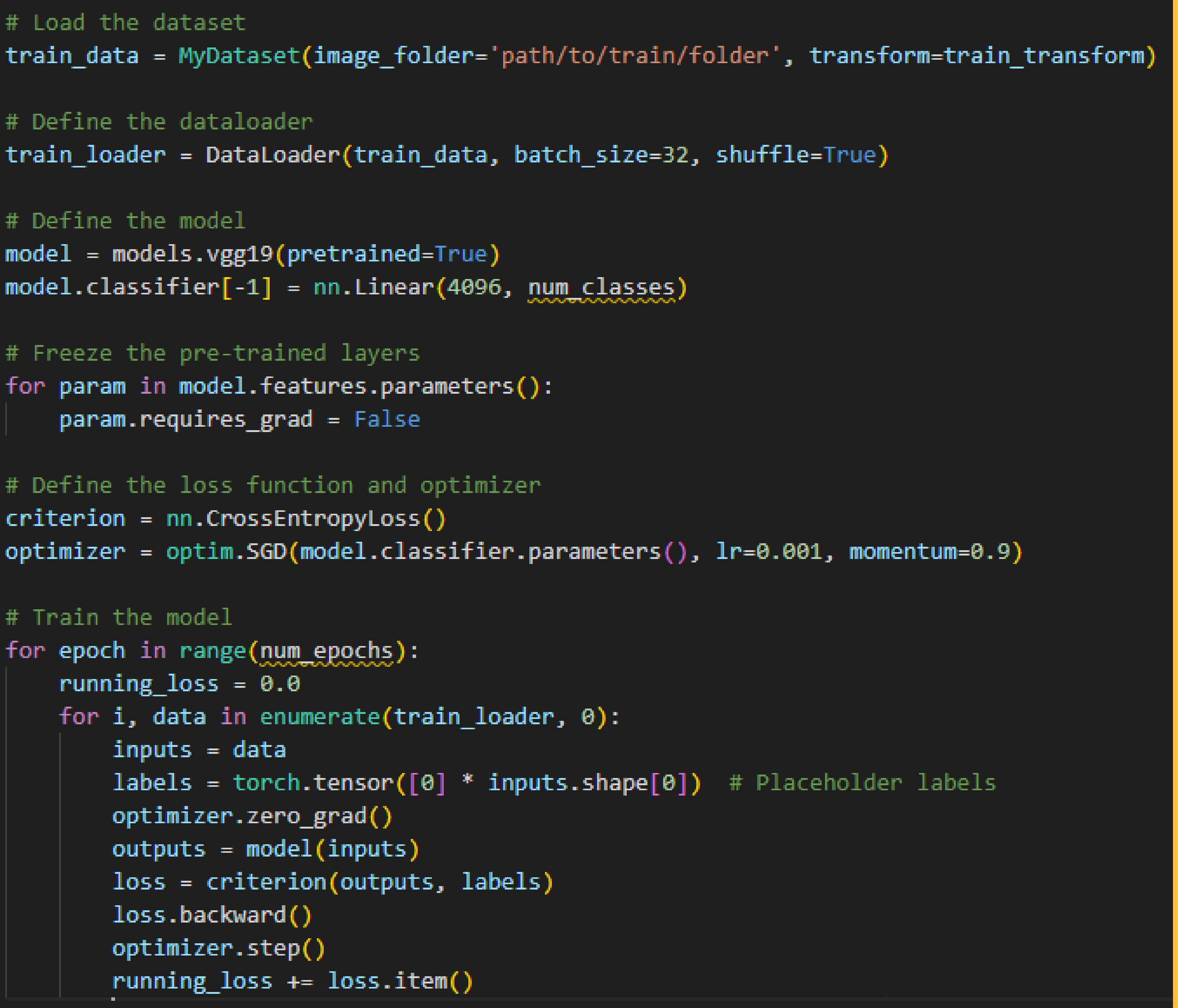
Figure 3: VGG-19[8]

Training VGG-19 Model With Specific Dataset

- Trained for content and style images, 1000 each.
- The hyperparameters and architecture should be adjusted and optimized.
- Allows to capture the specific characteristics of the content and style images

Using ImageNet Pretrained VGG-19 Model

- Trained on a large and diverse dataset.
- Already learned to extract low-level features from the images.
- Not always suitable for the specific task may require fine-tuning or re-training



```
# Load the dataset
train_data = MyDataset(image_folder='path/to/train/folder', transform=train_transform)

# Define the dataloader
train_loader = DataLoader(train_data, batch_size=32, shuffle=True)

# Define the model
model = models.vgg19(pretrained=True)
model.classifier[-1] = nn.Linear(4096, num_classes)

# Freeze the pre-trained layers
for param in model.features.parameters():
    param.requires_grad = False

# Define the loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(model.classifier.parameters(), lr=0.001, momentum=0.9)

# Train the model
for epoch in range(num_epochs):
    running_loss = 0.0
    for i, data in enumerate(train_loader, 0):
        inputs = data
        labels = torch.tensor([0] * inputs.shape[0])  # Placeholder labels
        optimizer.zero_grad()
        outputs = model(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        running_loss += loss.item()
```

Parameter Tuning for Training Deep Learning Model

Results and Conclusion

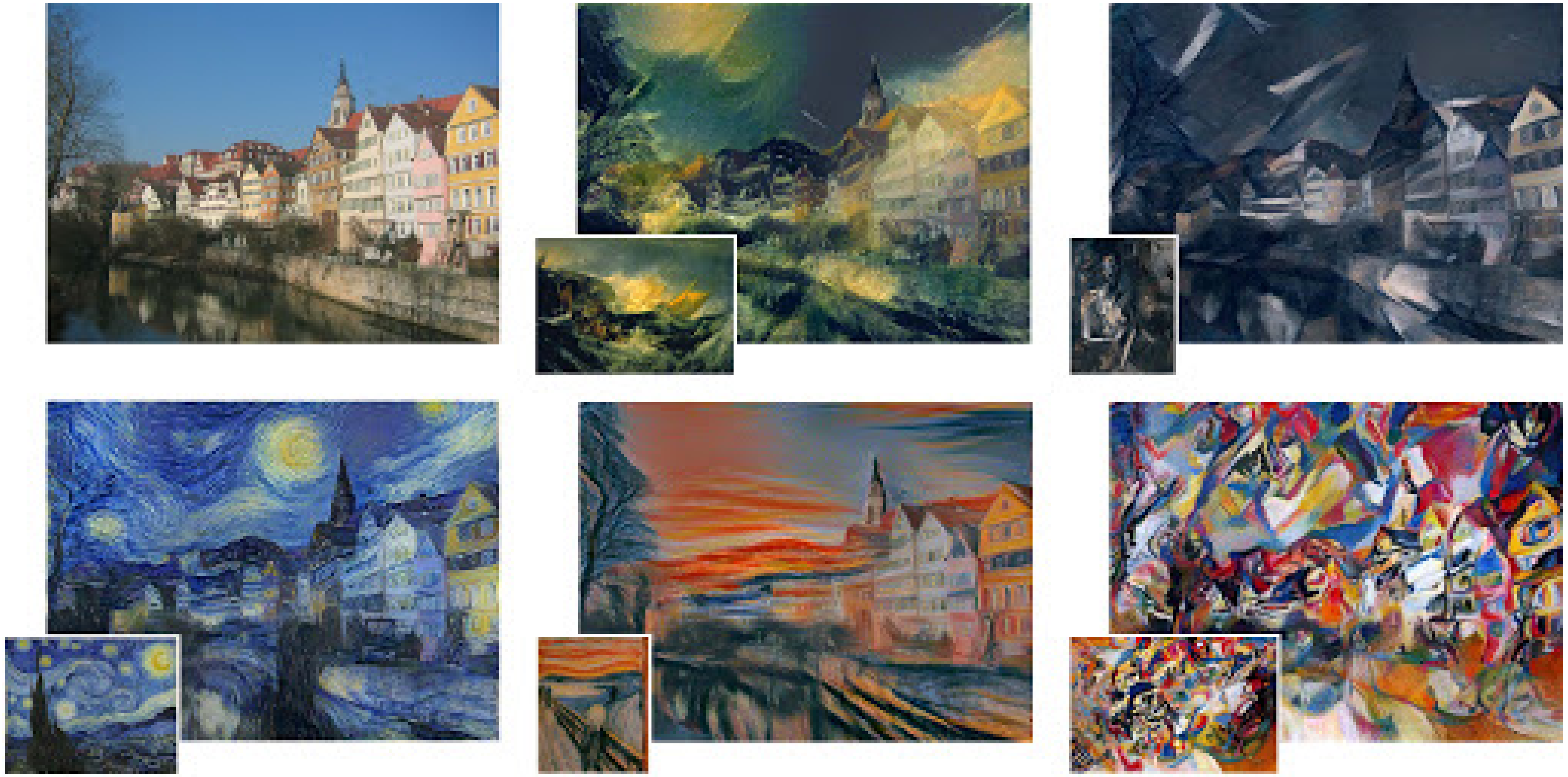


Figure 4: Example Outputs[5]

References

[1] Computer vision: Algorithms and applications, 2nd ed. [Online]. Available: http://szeliski.org/Book/. [Accessed: 07-Apr-2023].

[2] L. A. Gatys, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style,” arXiv.org, 02-Sep-2015. [Online]. Available: https://arxiv.org/abs/1508.06576. [Accessed: 07-Apr-2023].

[3] “How to style transfer your own images,” GoDataDriven, https://godatadriven.com/blog/how-to-style-transfer-your-own-images/ (accessed May 8, 2023).

[4] [1] M. Mandal, “Introduction to convolutional neural networks (CNN),” Analytics Vidhya, https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/ (accessed May 9, 2023).

[5] A. Sharma, “Introduction and implementation to neural style transfer - deep learning,” Analytics Vidhya, https://www.analyticsvidhya.com/blog/2020/10/introduction-and-implementation-to-neural-style-transfer-deep-learning/ (accessed May 8, 2023).

[6] “Neural style transfer tutorial with tensorflow and python in 10 minutes,” YouTube, 07-Jan-2021. [Online]. Available: https://www.youtube.com/watch?v=bFeltWvzZpQ. [Accessed: 07-Apr-2023].

[7] “Pytorch Neural Style Transfer tutorial,” YouTube, 02-Jul-2020. [Online]. Available: https://www.youtube.com/watch?v=imX4kSKDY7s. [Accessed: 07-Apr-2023].

[8] G. Boesch, “VGG very deep convolutional networks (vggnet) - what you need to know,” viso.ai, https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/ (accessed May 8, 2023).

[9] “Welcome to pytorch tutorials¶,” Welcome to PyTorch Tutorials - PyTorch Tutorials 2.0.0+cu117 documentation. [Online]. Available: https://pytorch.org/tutorials/. [Accessed: 07-Apr-2023].