Neural Style Transfer with Object Detection - Project Report

Prepared by:

Sujal Singh (E22CSEU0770)

Sahil Gupta (E22CSEU0756)

Saatvika Singh (E22CSEU0805)

Introduction

Neural Style Transfer (NST) is a deep learning technique that merges the content of one image with the style of another, resulting in a new, stylized image. Originally introduced by Gatys et al., NST leverages convolutional neural networks (CNNs) to extract and recombine image content and artistic style. In this project, we extend the concept by incorporating object detection to selectively apply style transfer to specific regions or objects within the image.

Objectives

- To implement Neural Style Transfer using a pre-trained VGG19 model.
- To integrate object detection to localize and isolate regions for selective style transfer.
- To understand and manipulate content and style representations using feature maps.
- To visualize how an input image is transformed artistically over training iterations.

Tools & Technologies Used

- **Programming Language:** Python
- **Libraries:** PyTorch, torchvision, PIL, matplotlib, OpenCV
- **Platform:** Google Colab
- **Models:** Pre-trained VGG19 for NST, and YOLOv5 for object detection

---

Methodology

Step 1: Image Loading, Object Detection and Preprocessing

- Load the content and style images.
- Use YOLOv5 to detect objects in the content image.
- Extract bounding boxes and mask detected objects.
- Apply transformations such as resizing and normalization.
- Convert images into tensors for PyTorch compatibility.

Step 2: Model Setup

- Load the pre-trained VGG19 model for feature extraction.
- Freeze model weights to prevent training.
- Extract features from selected layers relevant for content and style.

Step 3: Defining Loss Functions

- **Content Loss:** Measures difference between high-level features of content and generated image.
- **Style Loss:** Uses Gram matrices to capture texture and style representation.
- **Object Masking:** Apply losses selectively on detected object regions.
- Total loss is a weighted sum of content and style losses within and outside detected regions.

Step 4: Optimization

- Initialize the generated image as a clone of the content image.
- Use L-BFGS or Adam optimizer to minimize the total loss.
- Run multiple iterations to iteratively refine the generated image, focusing on objects if specified.

Results and Discussion

- NST successfully stylized the content image using the artistic pattern of the style image.
- When combined with object detection, the style transfer could be limited to key objects, enhancing creative control.
- Visual inspection shows the effectiveness of combining deep learning with creativity and semantic understanding.

Challenges Faced

- Object detection accuracy directly impacts the quality of localized style transfer.
- Convergence time is high for large images.
- Hyperparameter tuning for loss weights significantly affects output.
- Real-time processing is limited due to computational constraints.

Future Enhancements

- Apply NST on videos frame by frame with real-time object tracking.
- Use instance segmentation for more accurate style application.
- Optimize models using TensorRT or ONNX for deployment.
- Integrate into mobile applications using TensorFlow Lite.

Conclusion

This project demonstrates the power of combining Neural Style Transfer with object detection to produce intelligent and creative visual effects. The ability to apply styles selectively to detected objects opens new opportunities in design, AR/VR, and visual storytelling. With further optimization and real-time capabilities, this hybrid technique has the potential to be widely adopted in creative tech applications.

References

1. Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image Style Transfer Using Convolutional Neural Networks.
2. PyTorch Documentation - https://pytorch.org
3. Google Colab - https://colab.research.google.com
4. YOLOv5 by Ultralytics - https://github.com/ultralytics/yolov5