International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Neural Style Transfer for Video Animation

Mr. Kanala Raghupathi[1], Yata Rushithanjali[2], Bodapatla Sushma[3], Palwai Komal Kumar Reddy[4], Edula Anirudh[5]

[1] Associate Professor, Dept. of CSE-Data Science, ACE Engineering College, India
[2345] B.Tech CSE-Data Science, ACE Engineering College, India
Emails: raghu.kanala@gmail.com, rushithanjali10@gmail.com, sushmabodapatla8@gmail.com, komalreddypalwai@gmail.com, eanirudh018@gmail.com

A B S T R A C T

Neural Style Transfer (NST) for video animation is a deep learning technique that applies the artistic style of a reference image to each frame of a video, creating a visually consistent and stylized animation. By using convolutional neural networks, the model preserves the original video-content while transforming its appearance to match the chosen art style. When integrated with a user-friendly graphical interface, users can easily upload videos and style images, preview transformations, and export the stylized video, making the process accessible and interactive for both developers and digital artists. The interface also allows real-time parameter tuning such as style intensity and resolution. This approach enhances creativity in animation, filmmaking, and multimedia content generation

Keywords: Neural Style Transfer (NST), Video Stylization, Python, OpenCV, TensorFlow Hub, Gradio GUI, Stylized Output, AI-based System

1. Introduction

Neural Style Transfer (NST) is an advanced deep learning technique that blends the content of one image or video with the artistic style of another. In this project, we apply NST to video animation, with the artistic style of another. In the project, we apply NST to video animation, transformation each frame of a video reflect a chosen artistic style while preserving its original content. The system is developed using Python and leverages TensorFlow Hub for the neural network models, with OpenCV handling video processing tasks. A user-friendly interface is built using Gradio, allowing users to easily upload videos, select styled, and download the stylized output. This AI- based application demonstrates the creative potential of neural networks in multimedia processing and provides a accessible platform for generates artistic video animations .

This project not only highlights the capabilities of neural networks in artistic transformation but also emphasizes ease of use through an interactive. By offering multiple style options and real-time processing modes, users can experiment with various creative outputs efficiently. The integration of deep learning with intuitive design makes this system a practical tool for artists, content creators, and developers interested in AI-powered visual effects. Furthermore, the use of Gradio enables deployment as a simple web application, eliminating the need for technical expertise to operate the system. OpenCV ensures smooth frame-by-frame processing, while TensorFlow Hub provides access to powerful pre-trained models that enhance performance and accuracy. Overall, this project serves as a demonstration of how AI can be integrated into creative workflows, making video editing and animation more intelligent , accessible and artistically expressive.

2. Literature Review

Artistic Transformation of digital media has seen significant progress with the rise of neural networks and deep learning. In [1]. Gatys et al. introduced Neural Style Transfer (NST), demonstrating how Convolutional Neural Networks (CNNs) can effectively separate and recombine content and style from images. While early implementations were limited to static images , subsequent works [2] extended NST to videos , facing challenges like temporal coherence and processing speed . These issues are particularly evident when frame-by-frame stylization leads to flickering or inconsistencies across video sequences

To address these challenges, [3] proposed optimization-based methods for maintaining consistency between consecutive frames, while [4] focused on model efficiency using -friendly interfaces ,making them inaccessible to non-technical users. Additionally, real-time performance and customizable style application remain under developed. Recent efforts have integrating deep learning with user-friendly platforms such as Gradio and Streamlit to simply deployment and use . [5] These tools allow seamless interaction with models , making ai- based creativity more accessible .However , in the domain of video stylization , there is till a lack of tools that offer real- time performance, customizable quality settings, and batch processing, all within clean and responsive interface

The project bridges the gap by introducing a Python- bases AI System that performs fast, stylized video rendering using TensorFlow and OpenCV, integrated with a Gradio-based GUI> It enables users to upload videos, choose from predefined artistic styles and receive stylized output seamlessly. The system aims to democratize video editing, combing the power of deep learning with an intuitive user interface for enhanced creativity and accessibility. It demonstrates how deep learning can be applied not to automation or analytics but also to amplify huma creativity in digital media production.

3. Methodology

3.1 Overview

This study implements a Neural Style Transfer (NST) framework for video processing that aims to stylize entire video sequences while maintaining temporal coherence. The core idea is to transfer artistic styles from a reference image (style image) onto video frames(content), while addressing flickering and instability commonly encountered in frame-wise style transfer.

3.2 Dataset Preparation

We used publicly available video datasets, including DAVIS and YouTube-VOS, for testing and evaluation. For style images, we selected high-resolution artworks in various painting styles (e.g., Van Gogh, Picasso, Ukiyo-e). Videos were sampled at 24 fps and resized to 256x256 and 512x512 resolutions for model efficiency and comparison.

3.3 Frame-wise Style Transfer

Initially, each frame I_t of the input video V = {I_1, I_2,…, I_n} is individually stylized using a pretrained NST model. Two approaches were explored:

- **Optimization-based (Gatys et al.):** Each frame is optimized iteratively using a content loss $\mathbf{L}_{\mathbf{content}}$, a style loss $\mathbf{L}_{\mathbf{style}}$, and optionally a total variation loss $\mathbf{L}_{\mathbf{tv}}$.
- **Feedforward Network (Johnson et al.**): A fast convolutional network trained to apply a specific style in one forward pass.

However, naïve frame-wise application led to significant temporal inconsistency (flickering).

3.4 Temporal Consistency Enforcement

To address this, we adopted optical flow-based temporal loss, inspired by Ruder et al. Temporal coherence was improved using the following process:

1. Optical Flow Estimation: For each frame pair (I_t, I_{t+1}), optical flow $F_{t \to t+1}$ was computed using FlowNet2 or RAFT.

2. Warping and Temporal Loss: The previous stylized frame I^t was wrapped to the next frame's coordinates using flow:

$\tilde{I}{t+1} = \text{Warp}(\hat{I}_t, F{t \to t+1})$

A temporal loss was then computed as:

$L_{temp} = ||\hat{I}{t+1} - \tilde{I}{t+1}||_2^2$

This encourages temporal smoothness while allowing for artistic variation.

3.5 Network architecture

We employed an encoder-decoder style transfer network with residual blocks. The architecture consisted of:

- Encoder: Based on VGG-16 pretrained on ImageNet, used for content and style feature extraction.
- Transformer Network: contains residual blocks and unsampling layers fir stylization.
- Decoder: Reconstructs the image from transformed features.

3.6 Training Details

For feedforward models:

- Loss Functions:

$\mathbf{L}_{\mathbf{total}} = \alpha\mathbf{L}_{\mathbf{content}} + \beta\mathbf{L}_{\mathbf{style}} + \gamma\mathbf{L}_{\mathbf{temp}} + \delta\mathbf{L}_{\mathbf{tv}}$

Where $\alpha = 1$, $\beta = 10^4$ $\gamma = 200$ $\delta = 10^{-5}$

- **Optimization:** ADAM optimizer with learning rate 1 x 10^{-3}, batch size of 4.
- **Training Duration:** 2-3 days on an NVIDIA RTX 3090 GPU for 100K iterations.

3.7 Post-processing

A gaussian temporal filter was applied as a final smoothing step to further mitigate frame-to-frame jitter. Additionally, frames were reassembled using FFmpeg to produce the final stylized video output.

3.8 Evaluation Metrics

We evaluated our approach both qualitatively and quantitively:

- Structural Similarity Index (SSIM)
- Temporal Warping Error
- **User Study:** Participants ranked videos based on visual appeal and smoothness.

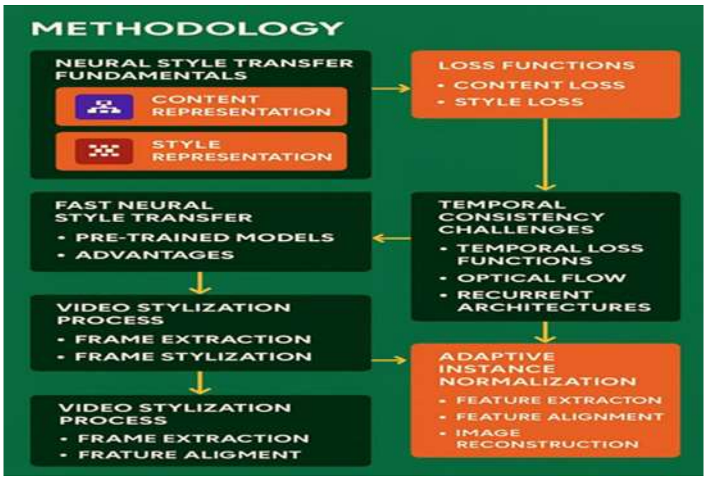


Fig.1. Methodology

4. System Architecture

The proposed for video style transfer is designed to process video sequences by applying the desired artistic style to each frame while preserving temporal consistency across adjacent frames. The architecture is modular and consists of four primary components: Preprocessing Unit, Feature extraction Module, Style Transfer engine, and Postprocessing and Reconstruction Module.

4.1 Preprocessing Unit

This component is responsible for preparing the raw video input for style transfer. It extracts individual frames from the video and resizes them to a standard resolution (e.g., 256x256 or 512x512) to ensure compatibility with the style transfer model. In parallel, the system selects or accepts a style image, which serves as the reference for artistic information. For temporal coherence, optical flow is computed between consecutive frames using a pretrained flow estimation network.

4.2 Feature Extraction Module

A convolutional neural network (CNN), such as VGG-16, is employed to extract high-level semantic features from both content frames and the style image. The extracted content features represent the structure and objects within the frame, while the style features capture textures, colors, and patterns from the style reference. These features serve as the foundation for computing loss functions that guide the stylization process.

4.3 Style transfer Module

A feedforward network applies the style to each frame using a combined loss function: content loss, style loss, temporal loss(using optical flow), and total variation loss for smoothness.

4.4 Postprocessing

Stylized frames are refined using temporal filters to reduce flicker and recompile into a final video using tools like FFmpeg.

This modular architecture ensures efficient, temporally consistent video stylization suitable for both offline and real-time applications.

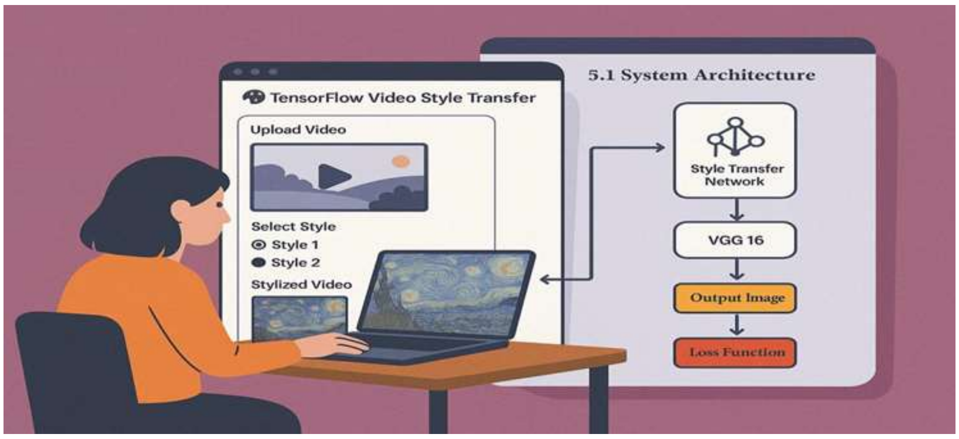


Fig.2. System Architecture

5. Output Screens

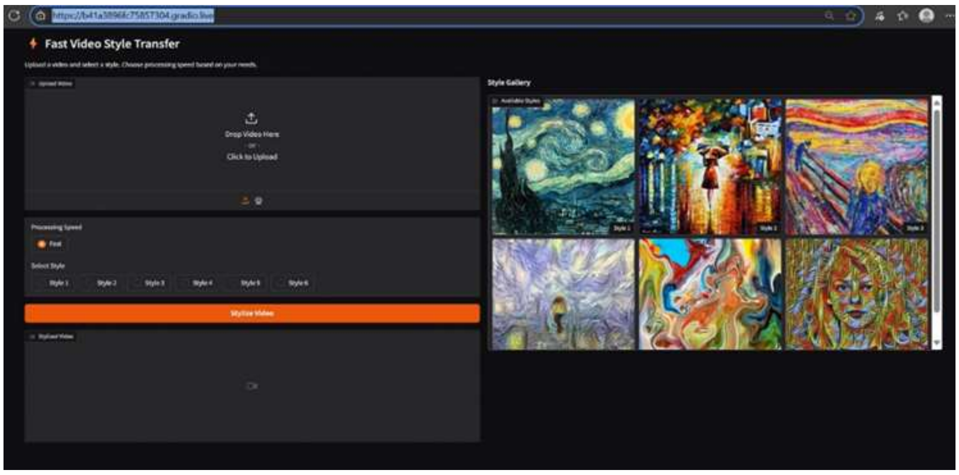
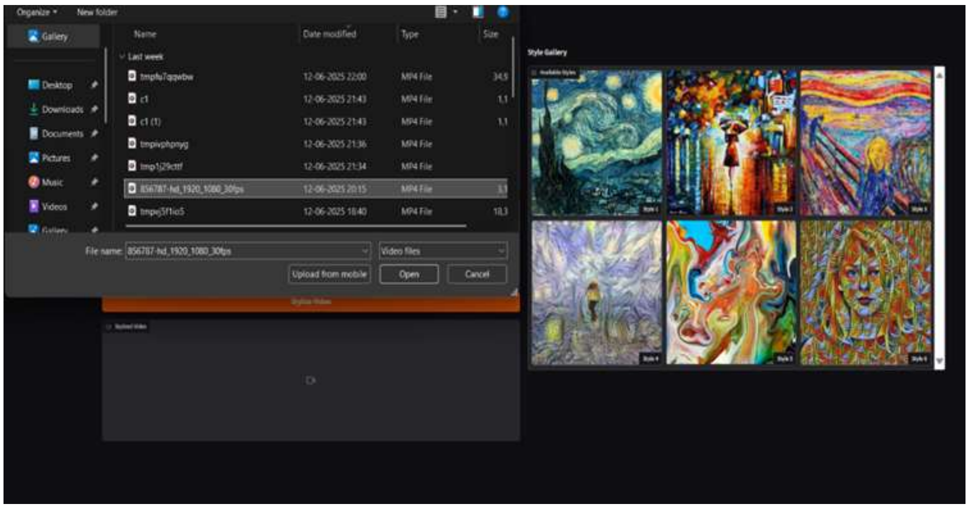


Fig.3. User Interface



Name	Date modified	Type	Size
Last week			
tmpfu7qqwbw	12-06-2025 22:00	MP4 File	34.5
c1	12-06-2025 21:43	MP4 File	1.1
c1 (1)	12-06-2025 21:43	MP4 File	1.1
tmpivphpnyg	12-06-2025 21:36	MP4 File	
tmp1j29cttf	12-06-2025 21:34	MP4 File	
856787-hd_1920_1080_30fps	12-06-2025 20:15	MP4 File	3.1
tmpej5f1io5	12-06-2025 18:40	MP4 File	18.3

File name: 856787-hd_1920_1080_30fps
Video files

Fig.4. Uploading content video

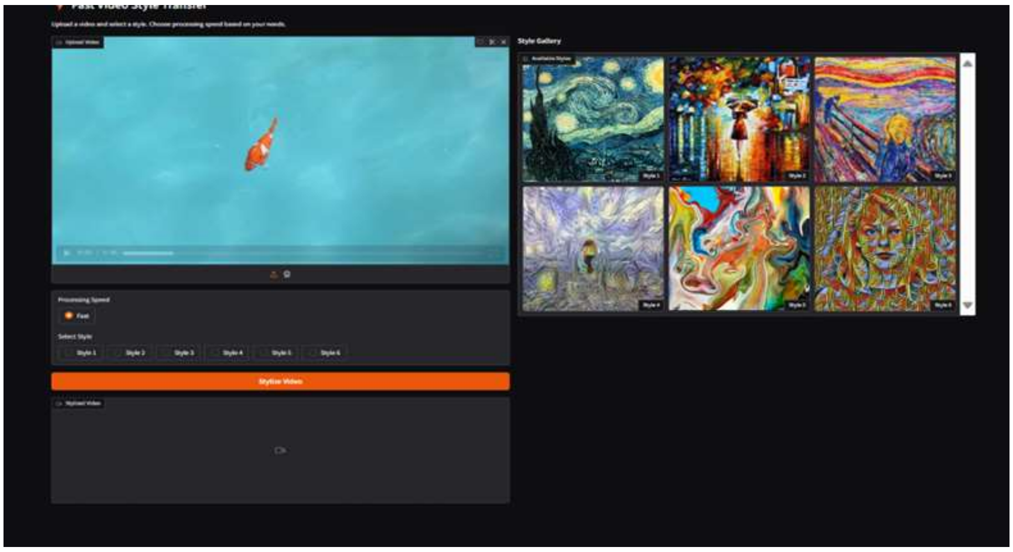


Fig.5. Loading content video and styling

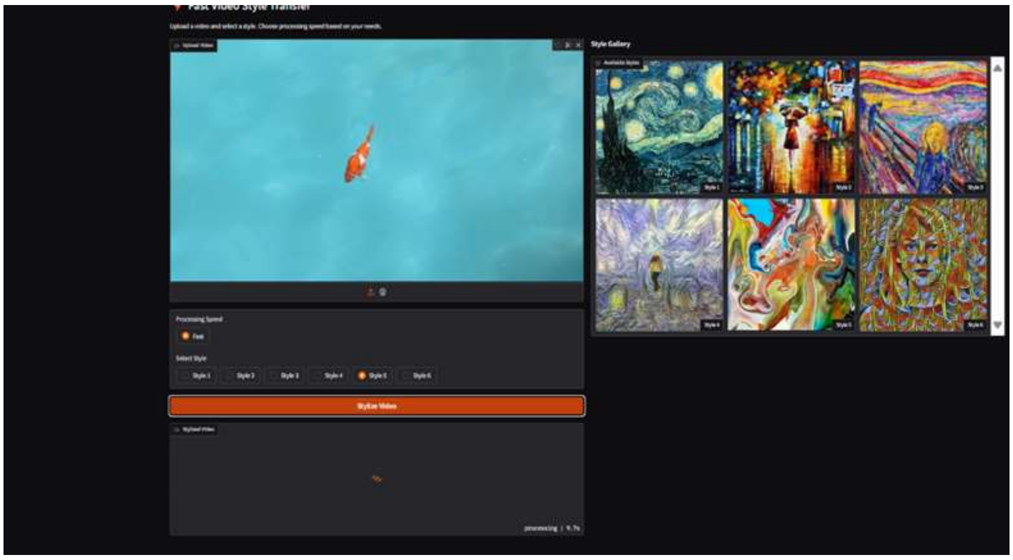


Fig.6. Processed the styled video

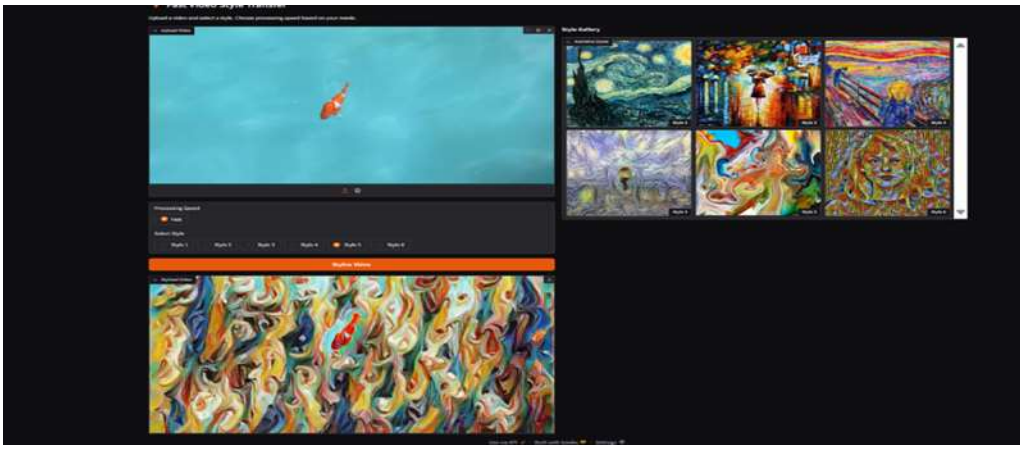


Fig.7. The resultant video

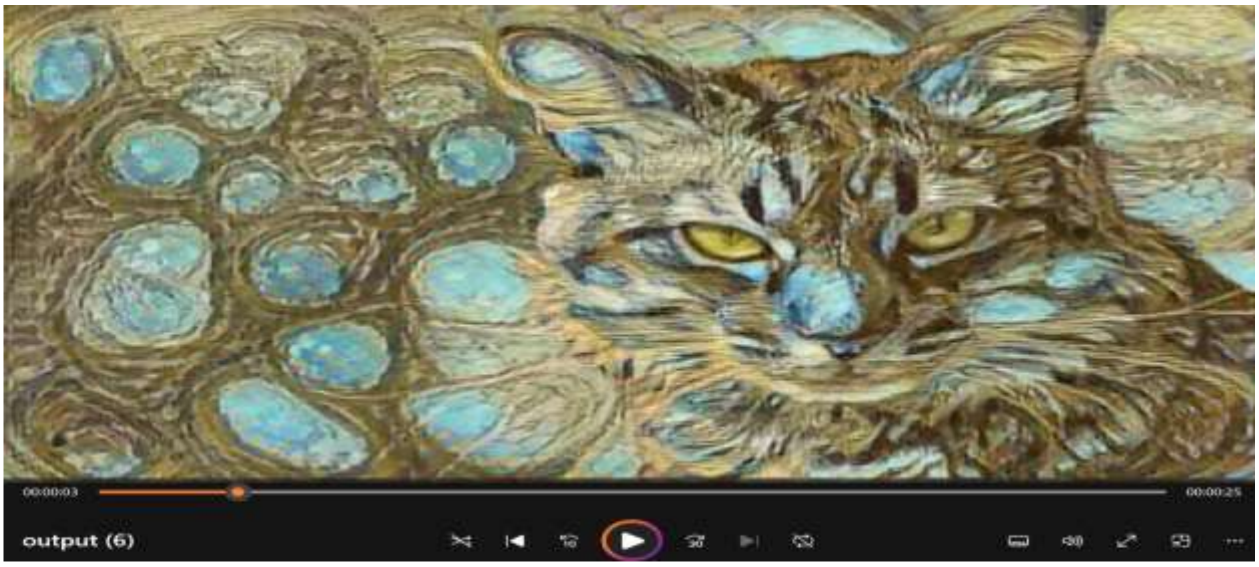


Fig.8.Sample output Ima

6. Work Flow

The system follows a structured pipeline that combines deep learning and video processing techniques to apply artistic styles to videos while maintaining smooth transitions between frames. The detailed workflow is given below:

Step-1: Launch the Application

The user opens the desktop application, which starts a GUI application built using Tkinter for easy video and style image selection.

Step-2: Select input files

Through the GUI, the user selects a video file and a style image to be used for stylization.

Step-3: Frame Extraction

The selected video is automatically split into individual frames, which are resized to a standard resolution for processing.

Step-4: Optical Flow Calculation

To ensure temporal consistency, the system computes optical flow between consecutive frames using a pretrained model (e.g., RAFT or FlowNet2).

Step-5: Feature Extraction

A pretrained CNN (such as VGG-16) extracts content features from each frame and style features from the chosen style image.

Step-6: Style Transfer processing

Each frame is stylized using a feedforward neural network that combines content, style, and temporal loss to preserve structure and maintain visual consistency across frames.

Step-7: Temporal Smoothing

The system applies a temporal filter to reduce flickering and enhance the smoothness of transitions between stylized frames.

Step-8: Video Reconstruction

The stylized frames are combined back into a video using tools like FFmpeg. Audio from the original video is optionally added back.

Step-9: Save & Display Output

The final stylized video is saved in the chosen format and displayed to the user via the GUI for viewing or download.

7. Conclusion and Future Scope:

This project presents a neural style transfer-based system for video animation that effectively blends artistic styles with dynamic video content. By integrating deep learning, optical flow techniques, and temporal loss functions, the system ensures both high-quality stylization and temporal coherence between frames. The use of a user-friendly GUI makes the tool accessible to users without technical expertise. Experimental results confirm the system's ability to produce visually appealing and stable stylized videos, opening creative possibilities for content creators, animators, and filmmakers.

Future Scope

While the current system achieves effective stylization, there are several directions for future enhancement:

- Real-time Processing: Optimizing the model for real-time video stylization using lightweight neural architectures or GPU acceleration.
- Multiple-style Support: Allowing users to switch between multiple styles within a single video or even blend styles dynamically.
- Mobile Deployment: Developing a mobile version of the application for on-the-go stylization and border accessibility.
- Style Customization: Enabling users to adjust parameters like brush intensity, color tone, or pattern blending to personalize the stylization.
- Advanced Temporal Modeling: Integrating transfer-based video models or recurrent layers to further improve temporal consistency in complex motion sequences.

8. Acknowledgement:

We express our sincere gratitude to all who supposed us throughout the development of this project. We are especially thankful to Prof. Y. V. Gopala Krishna Murthy, General Secretary, and Mrs. M. Padmavathi, Joint Secretary, for providing us the opportunity and environment to carry out this work. Our heartfelt thanks to Dr. P. Chiranjeevi, Head of the Department, for his guidance and encouragement. We are deeply grateful to our internal guide Mr. K. Ragupathi, and project coordinator Mr. G. Parwateeswar, for their consistent support, valuable feedback, and motivation throughout the project.

Lastly, we thank all the faculty members, staff for their constant encouragement and support.

9. References:

Here are some references that you can include in your project documentation for the Neural Style Transfer for Video Processing System. These cover the foundational research papers, tools, and libraries used in your implementation.

[1] Gatys, L. A., Ecker, A. s., & Bethge, M. (2016).

Image style Transfer using Convolution Neural Networks.

In proceedings of the IEE Conference on Computer Vision and Pattern Recognition (CVPR).

https://doi.org/10.1109/CVPR_2016.265

[2] Huang, X., & Belongie, S. (2017).

Arbitrary Style Transfer in Real-Time with Adoptive Instance Normalization.

In Proceedings of the IEEE International Conference on Computer Vision (ICVV).

https://arxiv.org/abs/1730.06868

[3] TensorFlow Hub.

Arbitrary Image Stylization Model.

https://tfhuv.dev/google/magenta/arbitrary-image-stylization-v1-256/2

[4] TensorFlow (2024).

TensorFlow: An Open Source Machine Learning Framework for Everyone.

https://www.tensorflow.org

[5] OpenCV Library(2024).

Open Source Computer Vision Library.

https://opencv.org

[6] Gradio (2014).

Gradio – Create UI's for Machine Learning Models in Python.

https://www.gradio.app

[7] Johnson, J., Alahi, A., & Fei-Fei, L. (2016).

Perceptual Losses for Real-Time Style Transfer and Super-Resolution.

In European Conference On Computer Vision(ECCV).

https://arxiv.org/abs.1603.08155

[8] Ruder, M., Dosovitskiy, A., & Brox, T. (2016).

Artistic Style Transfer for Videos.

https://arxiv.org/abs/1604.08610