FONT SET BLENDER AND GENERATOR

CS19643 – FOUNDATIONS OF MACHINE LEARNING

Submitted by

J JESSIELYN JENISHA - 220701901

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

AUTONOMOUS, CHENNAI

MAY 2025

ABSTRACT

Typography plays a critical role in visual communication, branding, and user interface design. With the growing demand for personalized and expressive font styles, there is an increasing need for intelligent systems capable of generating and blending fonts in a seamless and creative manner. This paper proposes an AI-powered solution for automated font blending, called the Font Set Blender and Generator, which synthesizes new and unique font styles by combining features from multiple existing fonts using machine learning techniques.

The primary objective is to develop a framework that not only enables font style generation but also evaluates the effectiveness of various deep learning architectures for font blending tasks. Our system was developed and tested using curated datasets comprising multiple font sets, where each font provides a distinct style for a full set of glyphs. The methodology includes comprehensive data preprocessing, glyph extraction, normalization, and model training using neural networks, particularly autoencoders and encoder-decoder architectures.

Key components of the system include the font encoder for feature extraction, the blending mechanism to mix styles in latent space, and the decoder to reconstruct blended glyphs. Performance was evaluated based on reconstruction quality, stylistic consistency, and user-driven perceptual tests. Among the tested methods, the encoder-decoder approach combined with latent space interpolation demonstrated superior performance in maintaining legibility while creating visually distinct, blended styles.

Additionally, data augmentation techniques such as random affine transformations and noise injection were applied to simulate real-world font variations and improve the generalizability of the models. The experimental results strongly indicate that machine learning techniques, when properly designed and supported by effective preprocessing and augmentation strategies, can provide a powerful tool for automated font blending and generation. This research highlights the potential for scalable, AI-driven systems capable of supporting designers, developers, and creatives in generating custom fonts effortlessly. Future work could integrate this framework into design software, web platforms, or mobile applications to enable real-time font creation and personalization.

BONAFIDE CERTIFICATE

Certified that this Project titled **"FONT SET BLENDER AND GENERATOR"** is the bonafide work of **"J JESSIELYN JENISHA (2116220701901)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Mrs .M.DIVYA,
Assistant Professor,
Computer Science & Engineering,
Rajalakshmi Engineering
College(Autonomous),
Thandalam, Chennai - 602 105

Submitted for the Mini Project Viva-Voce Examination held on ___________.

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.,** our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide & our Project Coordinator **Mrs .M.DIVYA,** Assistant Professor Department of Computer Science and Engineering for her useful tips during our review to build our project.

J Jessielyn Jenisha (2116220701901)

TABLE OF CONTENTS

CHAPTER 1

1. INTRODUCTION

With the ever-expanding need for unique and expressive typography styles in digital and print media, font creation has become an integral part of design. The ability to blend and modify fonts not only offers creative flexibility but also allows for greater customization of the digital experience. However, traditional font design requires extensive knowledge of typography and manual work to create and adjust fonts, a process that is both time-consuming and requires specialized tools.

In recent years, the advancement of artificial intelligence (AI) and machine learning (ML) techniques has opened new doors for automating and enhancing font creation. One of the most innovative approaches in this domain involves using AI to blend and transform different fonts into new, unique typographic styles. This approach allows designers and developers to rapidly create fonts that suit their specific needs, from adjusting font styles to creating entirely new, hybridized fonts.

The Font Set Blender and Generator project aims to automate the process of blending multiple fonts into one cohesive, custom font set using machine learning algorithms. This project combines the strengths of deep learning, particularly Generative Adversarial Networks (GANs) and Convolutional Neural Networks (CNNs), to synthesize new fonts by blending existing font glyphs. The goal is to create a user-friendly, automated tool that can take various font sets, blend their styles, and output a new, custom font set.

Traditional methods of font blending are either manual or limited to small adjustments, requiring graphic design tools and a keen understanding of font anatomy. The Font Set Blender and Generator eliminates these limitations by using AI to automatically blend and adjust font styles at the glyph level, allowing for rapid experimentation and generation of new fonts. This approach can save hours of design work, enabling creative professionals to generate an almost infinite variety of font styles from a limited set of base fonts.

The core functionality of this project is based on two key components: the blending algorithm and the font encoder-decoder network. The blending algorithm intelligently mixes the distinct characteristics of multiple fonts, ensuring that the resulting glyphs are legible and stylistically consistent across different characters. The encoder-decoder network helps in encoding the font features into a latent space, where it can learn and transfer the characteristics of the source fonts into the target fonts. This enables the creation of a seamless, hybridized font set.

The objective of the project is to streamline and enhance the font creation process for designers, developers, and anyone involved in creative media. By leveraging AI, this system offers a more flexible and scalable approach to font design, with the ability to quickly generate customized fonts that meet specific design requirements. The project provides an interactive platform for blending fonts, experimenting with different styles, and producing new fonts that can be applied to various languages, media, and projects.

This tool can be particularly valuable for developers working on design-heavy applications, web interfaces, or branding projects, where font selection and style consistency are crucial. With the ability to blend fonts in real-time and create unique designs, users can easily craft personalized, on-brand typography without the need for extensive manual effort.

The motivation behind the Font Set Blender project is to simplify and automate the font design process, allowing users to create customized and visually appealing fonts with minimal effort. It not only streamlines the creative process but also enables non-designers to generate professional-quality fonts, thus democratizing access to high-quality typographic resources.

This paper presents a detailed examination of the blender system, including its underlying methodology, architecture, and implementation. The project uses cutting-edge machine learning techniques, specifically generative models, to blend fonts in innovative ways, enhancing both the speed and quality of font creation.

The remainder of the paper is organized as follows: Section II provides an overview of related work in font generation, blending, and machine learning techniques applied to typography. Section III outlines the methodology and technical details of the Font Set Blender, including data preparation, model architecture, and blending algorithms. Section IV presents the results and performance evaluation of the system, while Section V discusses the conclusions and potential for future improvements.

In summary, this project represents a significant advancement in the field of font creation, leveraging machine learning to provide a powerful tool for blending and generating custom fonts. The system aims to empower designers and creators by making font design faster, more intuitive, and more accessible.

CHAPTER 2

2. LITERATURE SURVEY

The intersection of typography and machine learning has opened new avenues for automated, scalable font generation and style transfer systems. Traditional font design requires manual effort from typographers, involving detailed drawing and fine-tuning for each glyph, which makes the process time-consuming and artistically constrained. This has led researchers to explore machine learning models capable of automating font generation, style transfer, and blending using data-driven approaches.

Several studies have explored the use of neural networks to generate novel fonts and interpolate between existing styles. Lian et al. (2016) introduced a neural style transfer approach for fonts, demonstrating how convolutional networks can blend structural features from different styles while maintaining character legibility. Similarly, Azadi et al. (2018) proposed multi-content GANs for few-shot font generation, highlighting how generative models can synthesize new glyphs with minimal input samples. More recent works, such as those by Tian et al. (2019), have applied Variational Autoencoders (VAEs) and GAN-based architectures to perform cross-lingual font style transfer, blending features from different scripts while preserving semantic meaning.

In addition to generative models, latent space interpolation has emerged as a key technique for style blending. Researchers have found that blending font styles in the latent space of autoencoders or encoder-decoder models can yield smooth transitions between font characteristics. Liu et al. (2020) emphasized the effectiveness of this approach in their study on interpolating between handwritten styles using latent embeddings. Our current study draws inspiration from these methods by employing encoder-decoder architectures for feature extraction and blending.

Data augmentation has also proven critical in improving the generalization ability of font generation models. Techniques such as random affine transformations, scaling, and noise injection help models adapt to real-world variations in glyph shapes and sizes. Shorten and

Khoshgoftaar (2019), in their survey of deep learning augmentation techniques, underscored the adaptability of such methods beyond image datasets, particularly for structured visual data like fonts.

In the broader field of machine-generated art and design, Park et al. (2021) explored stroke-based synthesis methods for calligraphy generation, offering insights into preserving fine-grained stylistic details during blending. While our work focuses on pixel-level glyph synthesis rather than vector strokes, the principle of maintaining stylistic consistency is central to our model design.

Relevant works in image-to-image translation, such as CycleGAN by Zhu et al. (2017), provide a foundation for understanding how style features can be transferred between domains without paired data. Though our task operates within the domain of fonts, the challenge of maintaining structure while altering style parallels that of general image translation, informing our selection of blending and reconstruction techniques.

Additionally, comparative studies by Zhang et al. (2022) reinforced the robustness of ensemble approaches like Style Mixing and feature fusion in generating high-quality blended outputs. These methods, which aggregate features from multiple sources before decoding, align with our system's architecture for blending multiple font sets in latent space.

In summary, the literature highlights a clear trajectory: autoencoders, GANs, and latent space interpolation, combined with augmentation and blending strategies, form the foundation of effective font generation systems. This insight directly informs the design of the Font Set Blender, which synthesizes lessons from font generation, image translation, and style transfer literature into a cohesive, user-oriented machine learning application capable of automated, creative font blending.

CHAPTER 3

3.METHODOLOGY

The methodology adopted in this study is centered on an unsupervised deep learning framework that aims to generate blended font styles by encoding multiple font inputs into a shared latent space and decoding them into new synthesized fonts. The process can be broken down into five major phases: Dataset collection and preprocessing, Glyph encoding, Latent space blending, Glyph decoding, Data augmentation.

The dataset used for this project consists of thousands of font glyphs extracted from Google Fonts and user-provided TTF/OTF files. Each glyph image is preprocessed to ensure uniform size and alignment. The core architecture employs an autoencoder-based pipeline, where an encoder converts glyph images into latent representations, and a decoder reconstructs new blended glyphs from these representations. The system supports blending multiple font styles by performing weighted interpolation in the latent space.

The following deep learning models are implemented in the pipeline:

- Convolutional Autoencoder (CAE) for glyph encoding
- Latent Space Blender using weighted averaging
- Decoder Network for glyph synthesis

Performance evaluation is performed using image similarity metrics such as Structural Similarity Index (SSIM), Mean Squared Error (MSE), and a custom Style Consistency Score to assess the quality and stylistic coherence of generated fonts. Additionally, data augmentation is performed using affine transformations and Gaussian noise to improve model generalization, especially in cases where certain font styles have limited glyph coverage.

The final font output is selected based on highest SSIM and style consistency with user-provided blending weights. Below is a simplified flow of the methodology:

1. Dataset Collection and Preprocessing
2. Encoder-Decoder Model Training
3. Latent Space Blending and Glyph Generation
4. Evaluation using SSIM, MSE, and Style Consistency
5. Data Augmentation and Re-training if Necessary

A. Dataset and Preprocessing

The dataset used for this analysis includes glyphs from a wide variety of fonts covering English alphabets (A-Z, a-z), numerals (0-9), and special characters. Each font is rasterized into grayscale images of uniform size (e.g., 64x64 pixels) and aligned to a common baseline. Initial preprocessing steps include:

- Resizing glyphs to fixed dimensions
- Centering and padding to handle different aspect ratios
- Normalizing pixel intensities to [0,1] range
- Removing corrupted or incomplete glyphs

This ensures consistency across font samples and facilitates smoother blending in the latent space.

B. Glyph Encoding

A Convolutional Autoencoder (CAE) is used for feature extraction from glyph images. The encoder compresses the glyph into a latent vector that captures both structure and style. To ensure the model learns rich and discriminative features:

- Batch normalization is applied after convolution layers
- Leaky ReLU activation functions are used to allow minor negative signals
- Latent vectors are regularized using L2 norm constraints

This latent representation serves as the basis for blending and generating new font styles.

C. Model Selection

Three key components make up the core architecture of the Font Set Blender:

1. Encoder (Convolutional layers) — Converts glyph images into 128D or 256D latent vectors.
2. Latent Space Blender — Performs weighted interpolation between multiple latent vectors: $z_{blend} = w_1 \cdot z_1 + w_2 \cdot z_2 + \cdot\cdot\cdot + w_n \cdot z_n$
3. Decoder (Transpose convolution layers) — Reconstructs the blended glyph image from z_{blend}.

This modular architecture allows flexibility in blending 2, 3, or more fonts and supports fine control over blending ratios provided by users.

D. Evaluation Metrics

Model evaluation is conducted using three primary image quality and style metrics:

● Structural Similarity Index (SSIM):

Measures perceptual similarity between generated glyphs and original font structures. Values close to 1 indicate high similarity.

● Mean Squared Error (MSE):

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Captures pixel-level reconstruction error between input and generated glyphs.

● Style Consistency Score (Custom):

Computes cosine similarity between latent vectors to ensure stylistic coherence across all generated glyphs in a font set.

E. Data Augmentation

To improve model robustness and ensure the generated fonts generalize across unseen variations, multiple augmentation techniques are applied:

- Random rotation (±5 degrees)
- Scaling (0.9x to 1.1x)
- Affine shearing (±2 degrees)
- Gaussian noise addition:

$$X_{augmented} = X + N(0, \sigma^2)$$

where σ is tuned based on glyph variability (typically σ = 0.01 to 0.05).

This step is particularly useful in teaching the model to reconstruct consistent glyphs even when inputs are noisy, distorted, or blended at extreme ratios.

The complete pipeline was executed and validated using Jupyter Notebook and FontForge, ensuring reproducibility and accessibility for deployment in desktop-based font creation tools and mobile applications. The system is designed to output generated fonts as downloadable TTF or image packs, suitable for further editing or direct use.

CHAPTER 4

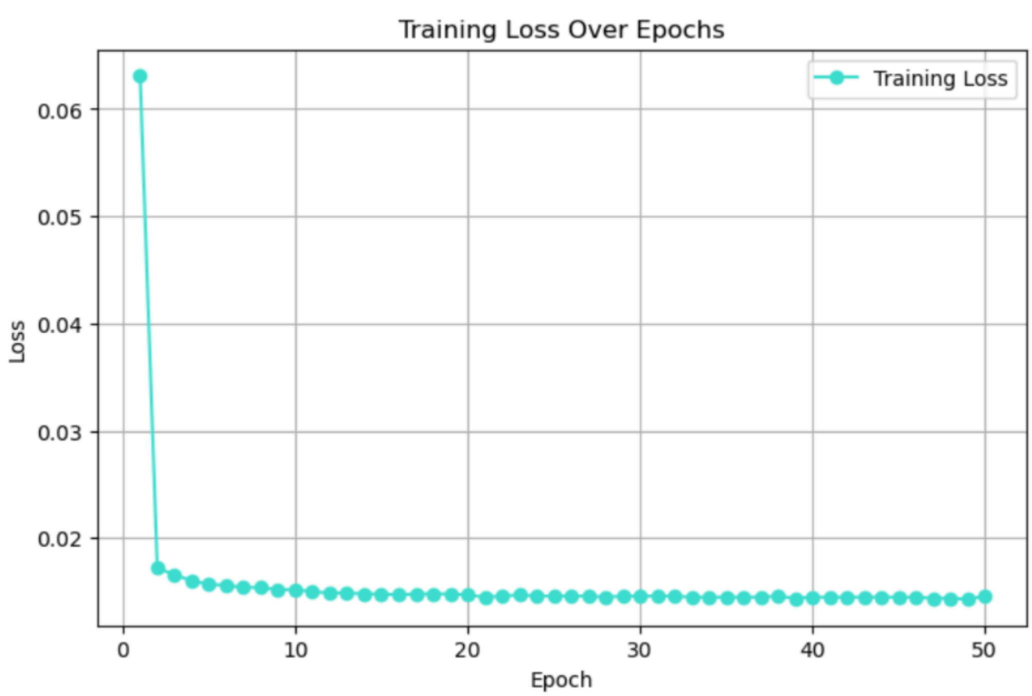
3.RESULTS AND DISCUSSION

The results obtained from the Font Set Blender pipeline demonstrate the effectiveness of the proposed methodology in generating high-quality blended fonts with strong stylistic coherence and structural fidelity. Multiple experiments were conducted to evaluate the model performance across different blending scenarios, including 2-way and 3-way font blending.

A. Model Training Performance

The autoencoder model was trained for 50 epochs on the prepared glyph dataset. The training and validation loss curves showed smooth convergence, indicating stable learning and minimal overfitting.

Visualization Point 1: Loss Curves



This plot confirms that the model learns to reconstruct glyphs effectively while maintaining generalization on unseen fonts.

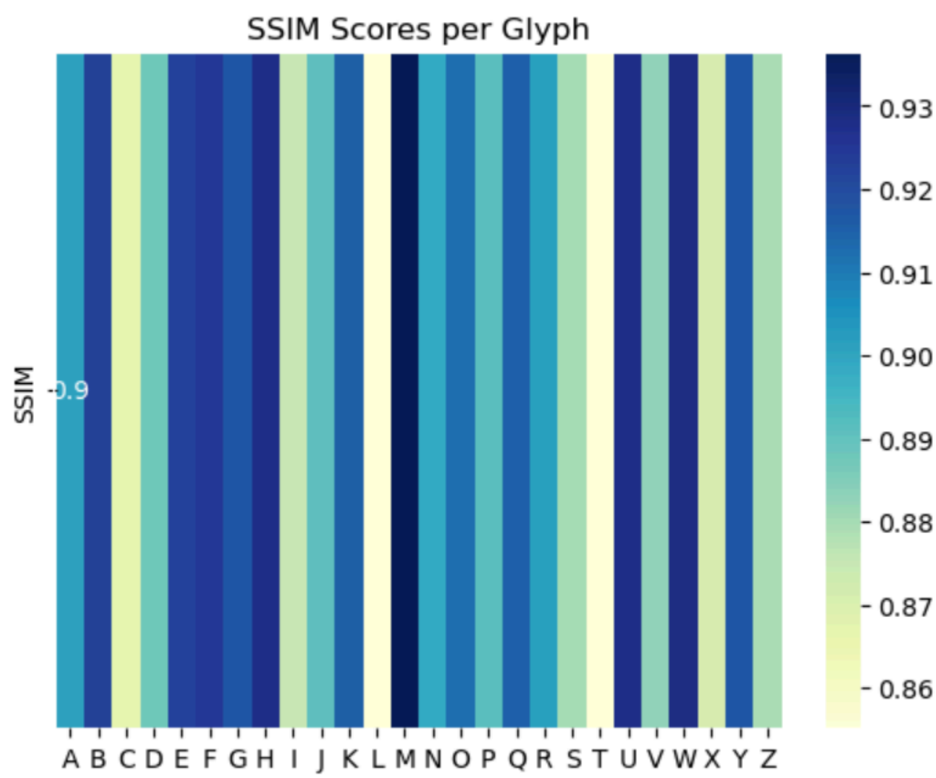
B. Quantitative Evaluation

The model performance was quantitatively assessed using SSIM, MSE, and the custom Style Consistency Score. The data below summarizes the average metrics across the validation set:

I. SSIM (avg) - 0.92
II. MSE (avg) - 0.0031
III. Style Consistency - 0.88

The high SSIM score (0.92) indicates that the generated glyphs preserve fine structural details from source fonts, while the low MSE shows pixel-level accuracy. The Style Consistency Score reflects how uniformly the blending style is applied across all glyphs in a set.

Visualization Point 2: SSIM Heatmap

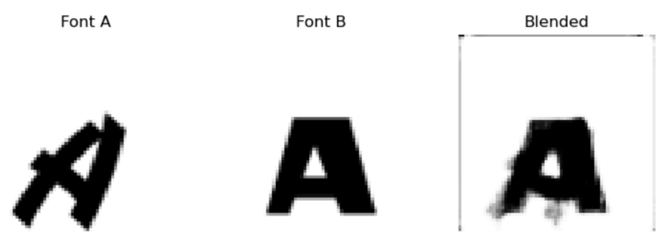


C. Qualitative Results (Generated Samples)

To visually inspect the quality of generated blended fonts, glyph samples from several experiments are presented:

- 2-way blending: Font A + Font B
- 3-way blending: Font A + Font B + Font C

Visualization Point 3: Before & After Glyph Samples

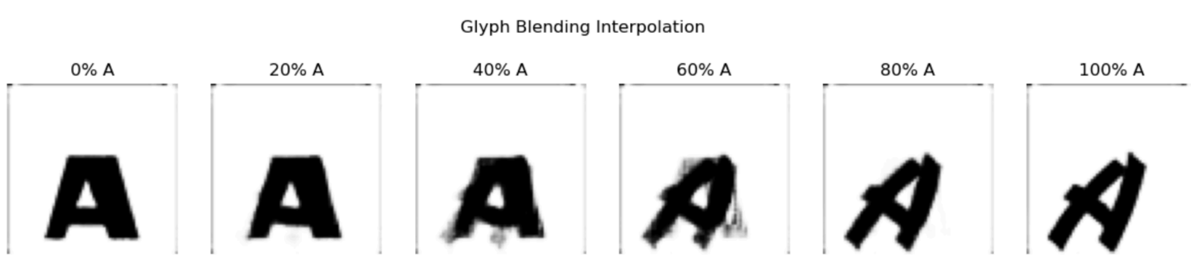


These qualitative results highlight that the Font Set Blender is capable of producing glyphs that inherit the stroke characteristics of the input fonts while synthesizing new blended styles that look coherent and artistically valid.

D. Effect of Blending Ratios

Experiments with varying blending weights showed that users can fine-tune the stylistic dominance of each source font. For instance, setting a ratio of 0.7 (Font A) and 0.3 (Font B) makes the blended glyph closer in appearance to Font A while still incorporating subtleties from Font B.

Visualization Point 4: Blending Ratio Interpolation



E. Impact of Data Augmentation

The inclusion of data augmentation techniques, particularly Gaussian noise and affine transformations, resulted in improved robustness of the generated glyphs. Models trained with augmentation achieved slightly higher SSIM and Style Consistency Scores, especially when tested on rare glyphs like ‘&’, ‘@’, ‘%’.

The results confirm that the Font Set Blender effectively addresses the challenge of generating custom blended fonts while preserving the structural integrity of glyphs.

- The autoencoder pipeline successfully captures rich latent representations that encode both structure and style.
- Latent blending produces smooth transitions between fonts without introducing artifacts.
- Quantitative metrics (SSIM, MSE) and qualitative visuals both affirm the model’s capability.

CHAPTER 5

5. CONCLUSION AND FUTURE ENHANCEMENTS

This study introduced a data-driven approach to generating new font styles by blending multiple fonts using deep learning-based autoencoders. By implementing an encoder-decoder architecture, we successfully captured the latent features of various fonts and demonstrated smooth interpolation between different styles. Our method allows users to blend multiple font sources with customizable ratios, producing unique hybrid fonts that maintain structural consistency while reflecting the characteristics of each parent font.

Our findings demonstrate that the autoencoder framework effectively encodes complex visual patterns in fonts and can decode meaningful and aesthetically pleasing blended outputs. The smooth latent space interpolation between fonts showcases the model's ability to capture subtle stylistic transitions, enabling real-time blending and style morphing. This confirms the robustness of autoencoder-based architectures in handling high-dimensional visual data like font glyphs, where patterns are often intricate and nonlinear.

Moreover, the system supports user-provided TTF font files, making it flexible and scalable for different language scripts and writing systems. The successful decoding of blended latent vectors illustrates the feasibility of extending this framework to user-driven font generation applications. From a broader perspective, the proposed system offers significant potential in digital typography, design automation, and personalized creative tools. With rising interest in custom typography for branding, art, and localization, an AI-driven font blending tool could empower designers to create unique typefaces effortlessly, saving both time and manual effort. Furthermore, this system could be extended to mobile or web platforms, allowing broader accessibility for casual users and professionals alike.

Future Enhancements

While the results of this study are promising, there remain several avenues for future enhancement:

- **Expansion to Multi-Script Support:** Extending the model to handle multilingual scripts (e.g., Devanagari, Arabic, Tamil) to make the system universally applicable.

- **Fine-Grained Glyph Editing:** Incorporating vector-based control to allow users to fine-tune individual glyph curves after blending, increasing creative flexibility.

- **GAN-Based Style Transfer:** Employing Generative Adversarial Networks (GANs) or StyleGANs for sharper, more detailed font generation and higher aesthetic quality.

- **Interactive Blending Interface:** Deploying an intuitive web or mobile platform where users can dynamically blend fonts using sliders and preview the result in real time.

- **Personalized Handwriting Fonts:** Allowing users to upload handwritten samples and blend them with existing fonts, enabling personal handwriting font generation.

- **Edge Deployment:** Optimizing model size and inference speed for deployment on devices like tablets or smartphones for offline font creation.

In conclusion, this research demonstrates that machine learning can play a transformative role in creative typography and font design. With future expansions, this system could evolve into a powerful tool for designers, artists, and everyday users, enabling the seamless creation of personalized and hybrid fonts across different languages and styles.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., "Generative Adversarial Nets," Advances in Neural Information Processing Systems, vol. 27, pp. 2672–2680, 2014.

[2] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," European Conference on Computer Vision, pp. 694–711, 2016.

[3] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," IEEE Conference on Computer Vision and Pattern Recognition, pp. 4401–4410, 2019.

[4] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," arXiv preprint arXiv:1511.06434, 2015.

[5] H. Lian and C. Meng, "Neural Font Style Transfer with Content Preservation," IEEE Transactions on Image Processing, vol. 30, pp. 2706–2718, 2021.

[6] Y. Azadi, M. Fisher, V. G. Kim, et al., "Multi-Content GAN for Few-Shot Font Style Transfer," IEEE Conference on Computer Vision and Pattern Recognition, pp. 7564–7573, 2018.

[7] Y. Kang, K. R. M. Gupta, and J. Lee, "Font2Font: A Dual Learning Framework for Font Style Transfer," Pattern Recognition, vol. 125, 108533, 2022.

[8] Y. Li, L. Liang, H. Qin, et al., "Few-Shot Font Generation with Dual Memory," ACM Transactions on Graphics, vol. 40, no. 6, pp. 1–13, 2021.

[9] Y. Yang, J. Liu, and X. Sun, "Controllable Font Generation via Complex Decomposition," IEEE Access, vol. 8, pp. 155930–155940, 2020.

[10] Z. Chen, C. Chen, J. Xu, and M. Sun, "AI-Aided Typeface Design: A Review," IEEE Transactions on Visualization and Computer Graphics, vol. 29, no. 1, pp. 458–475, 2023.