# Automated Personal Writing Pattern Replicator

**Team Members:**

**Adarsh Ghimire(THA077BCT005)**

**Kishan Adhikari(THA077BCT021)**

**Nasir Hussain(THA077BCT027)**

**Prabuddha Jung Thapa(THA077BCT032)**)

**Supervised By:**

**Er. Kiran Chandra Dahal**

Department of Electronics and Computer Engineering
Thapathali Engineering Campus

July 2024

# Presentation Outline

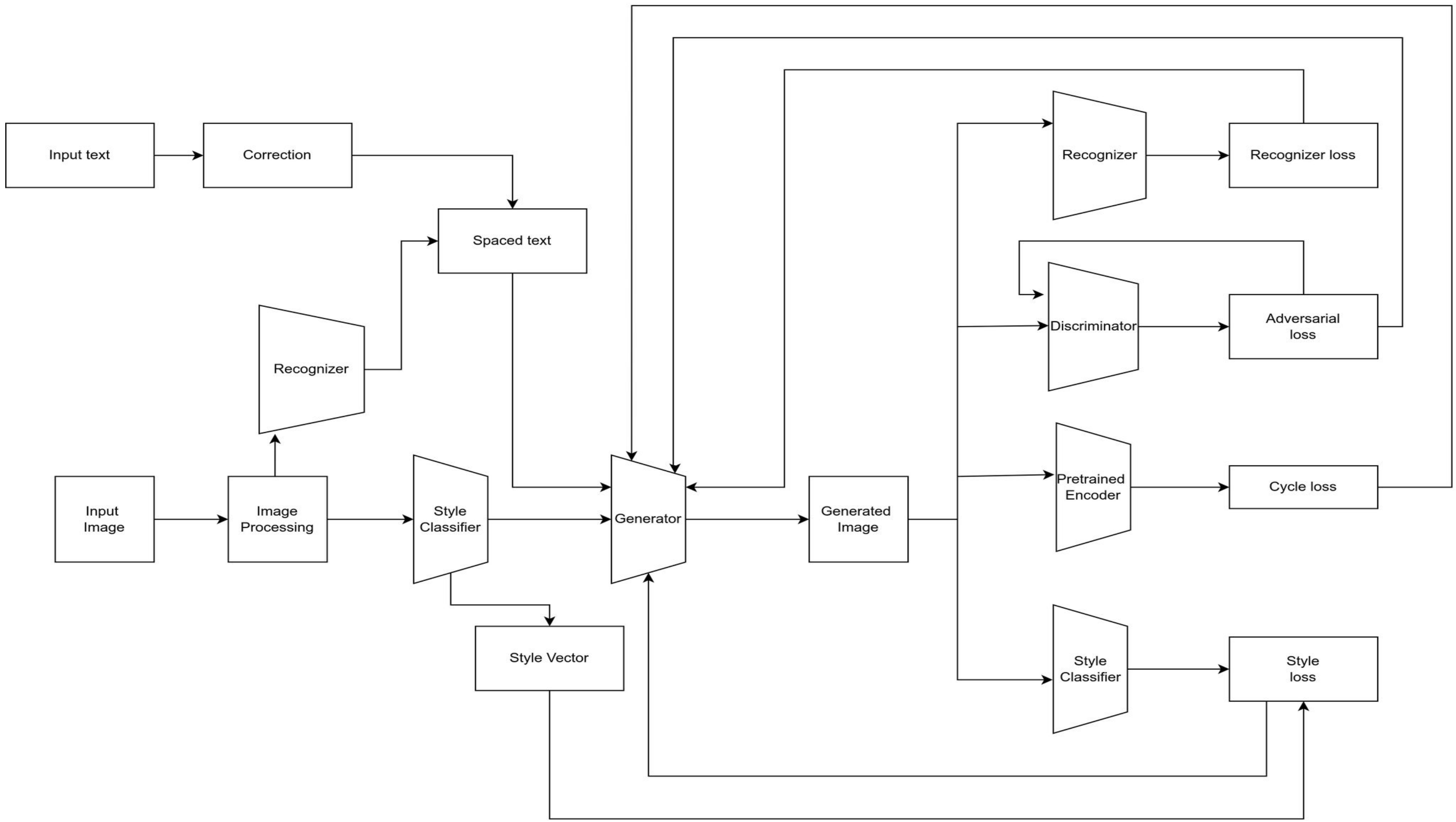
- Introduction
- Methodology
- Results & Discussion
- Expected Results
- References

# Introduction

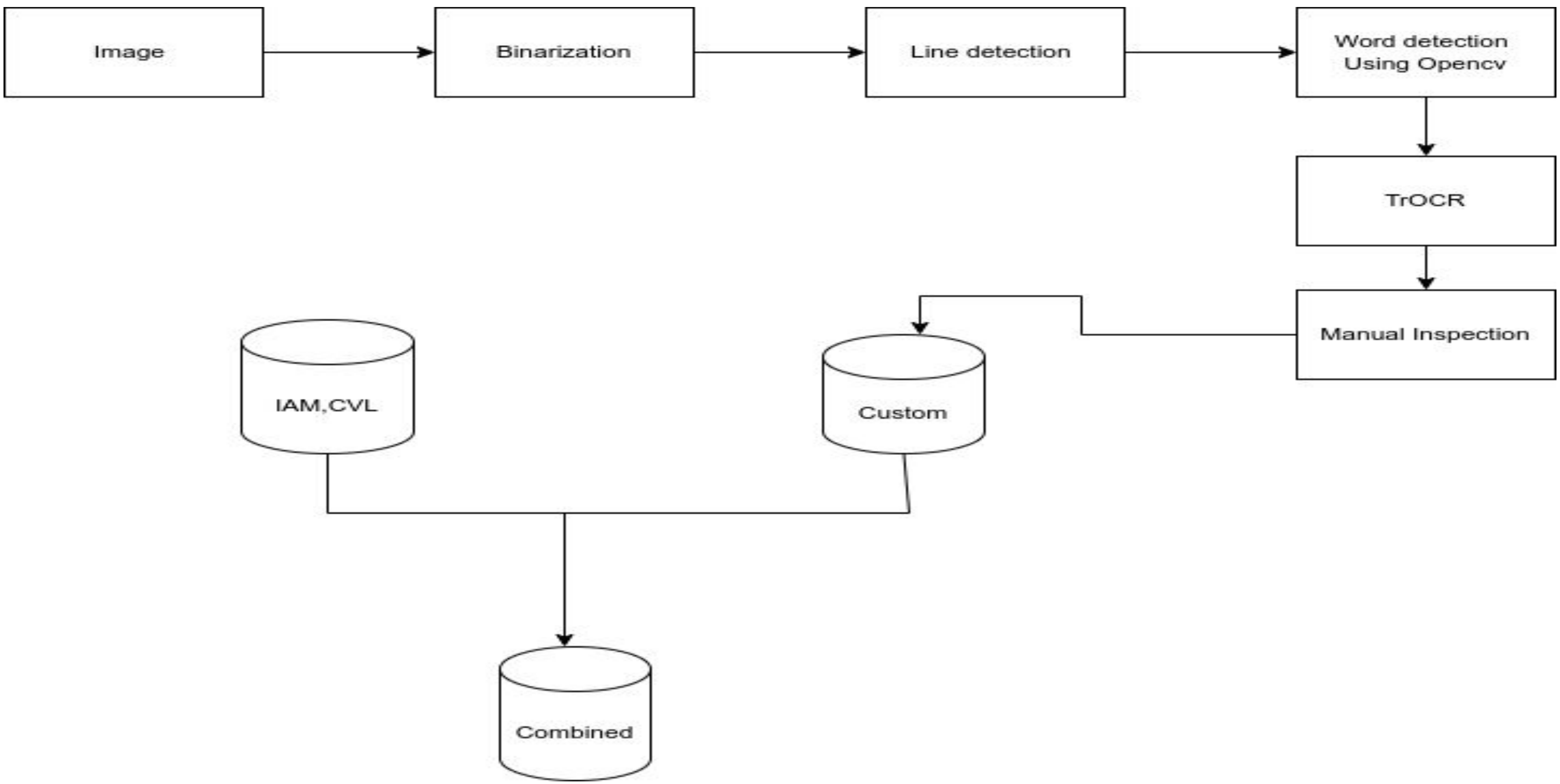
- This project focuses on accurately cloning individual handwriting styles using advanced machine learning, enabling the generation of personalized handwritten text and digital fonts.

- It aims to preserve unique handwriting characteristics, to enhance digital personalization and accessibility for various applications, including custom fonts and educational materials.

# Methodology (Project Architecture)

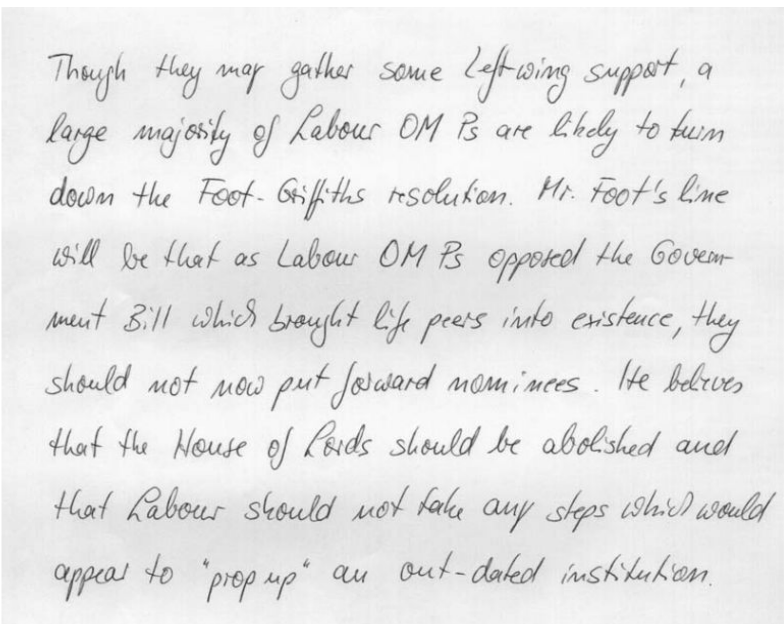


# Methodology (Dataset)

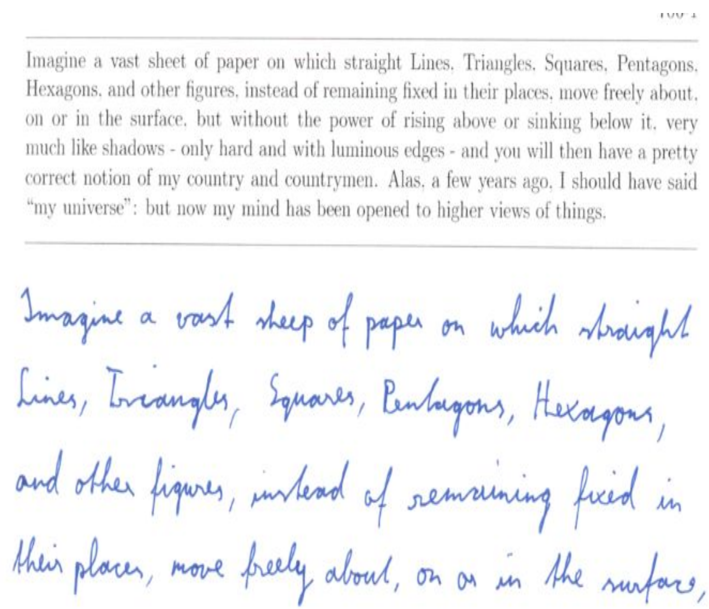


# Datasets

| S.N. | Datasets | Writers |
|---|---|---|
| 1. | IAM Handwriting Dataset | 657 |
| 2. | CVL Handwriting Dataset | 350 |
| 3. | Custom Dataset | 38 |

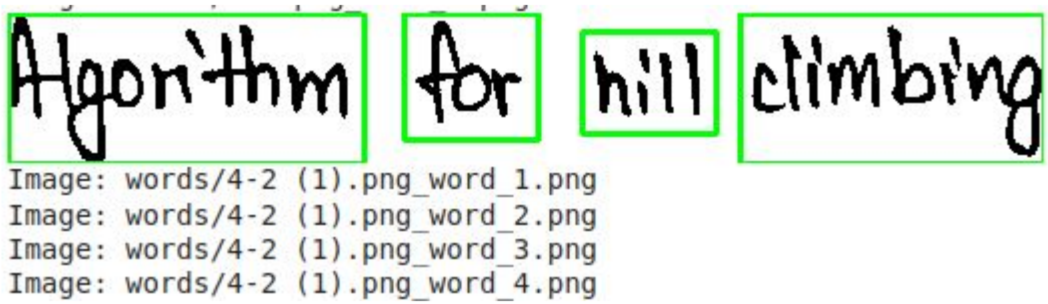


Though they may gather some Left-wing support, a large majority of Labour OM Ps are likely to turn down the Foot-Griffiths resolution. Mr. Foot's line will be that as Labour OM Ps opposed the Govern-ment Bill which brought life peers into existence, they should not now put forward nominees. He believes that the House of Lords should be abolished and that Labour should not take any steps which would appear to "prop up" an out-dated institution.



Imagine a vast sheet of paper on which straight Lines, Triangles, Squares, Pentagons, Hexagons, and other figures, instead of remaining fixed in their places, move freely about, on or in the surface, but without the power of rising above or sinking below it, very much like shadows - only hard and with luminous edges - and you will then have a pretty correct notion of my country and countrymen. Alas, a few years ago, I should have said "my universe": but now my mind has been opened to higher views of things.

Imagine a vast sheep of paper on which straight Lines, Triangles, Squares, Pentagons, Hexagons, and other figures, instead of remaining fixed in their places, move freely about, on or in the surface,



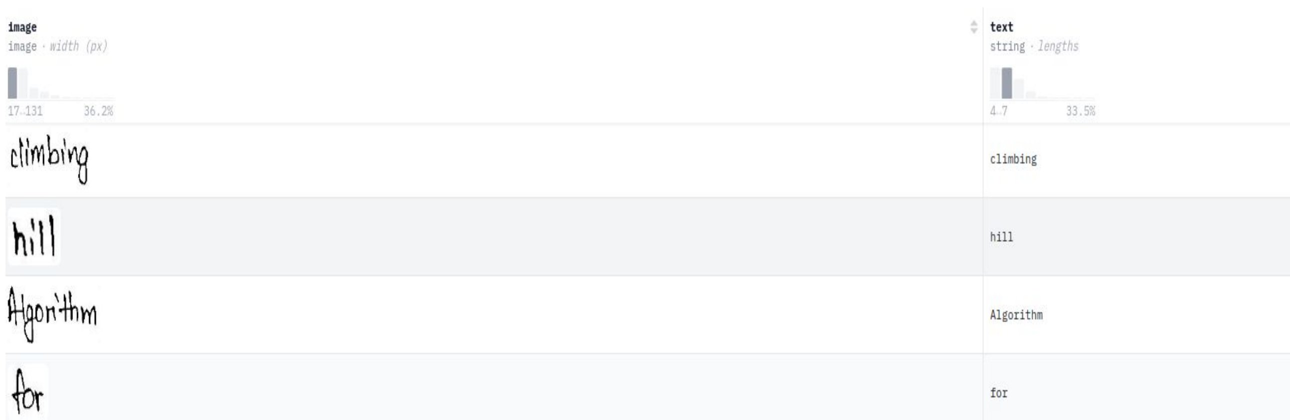




Fig: Custom Dataset Preprocessing

# Methodology (Data Processing)

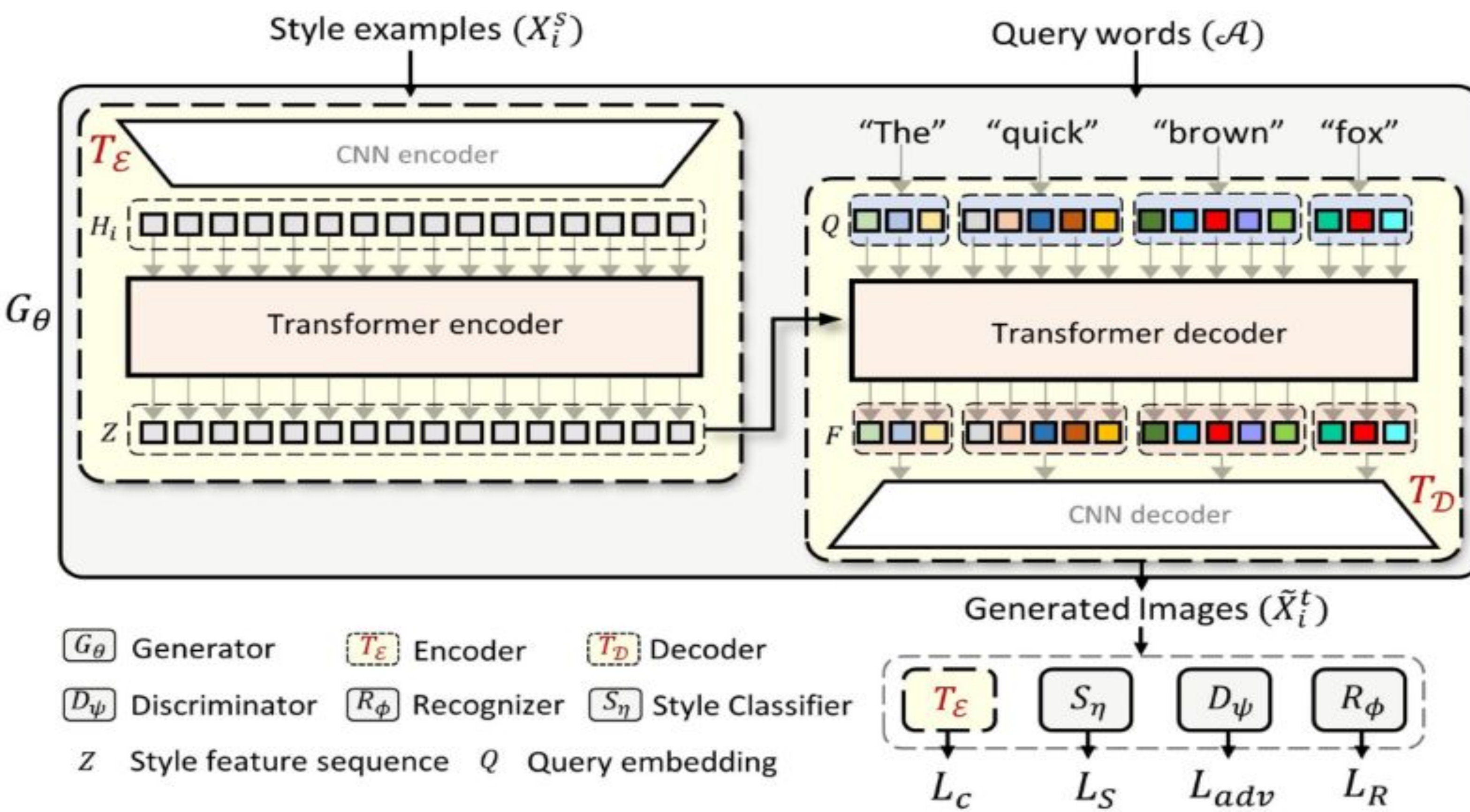
- Dataset present in IAM , CVL database consists of handwritten pages but for our model training we need to crop out the lines.
- For that we will extract the lines by defining the bounding box.
- Converting the images to grayscale to simplify processing and reduce computational load.
- Resize images to a consistent size to ensure uniformity in the training data.

# Methodology (Cont.) (Data Processing)

- For passing the image to train our model, we will use a CNN-based network to obtain the convolutional features from the images.

- The obtained features are flattened and passed to the transformer-based encoder layer.

# Methodology (Proposed Model Architecture)- Generator



# Methodology (Proposed Model Architecture)- Generator

## Style Encoding:

- **CNN Encoder**: Extracts stylistic features from input handwriting examples, converting images into feature representations.

- **Transformer Encoder**: Processes the style feature sequence to capture handwriting nuances, derived from the CNN Encoder features.

# Methodology (Proposed Model Architecture)- Generator

## Content Encoding:

- **Query Embedding**: Transforms the words from the query into embeddings, representing the content that needs to be handwritten.

## Decoding:

- **Transformer Decoder**: Merges the style features and the content embeddings to create a unified feature representation to ensures that the generated handwriting maintains the desired style and accurately reflects the content of the query words.
- **CNN Decoder**: Converts the combined feature representation into the final image of the handwritten text.

# Methodology
# (Proposed Model Architecture)

## **Recognizer**:

- Ensures the generated handwriting is legible and correctly represents the input text by recognizing and validating the characters.

## **Discriminator**:

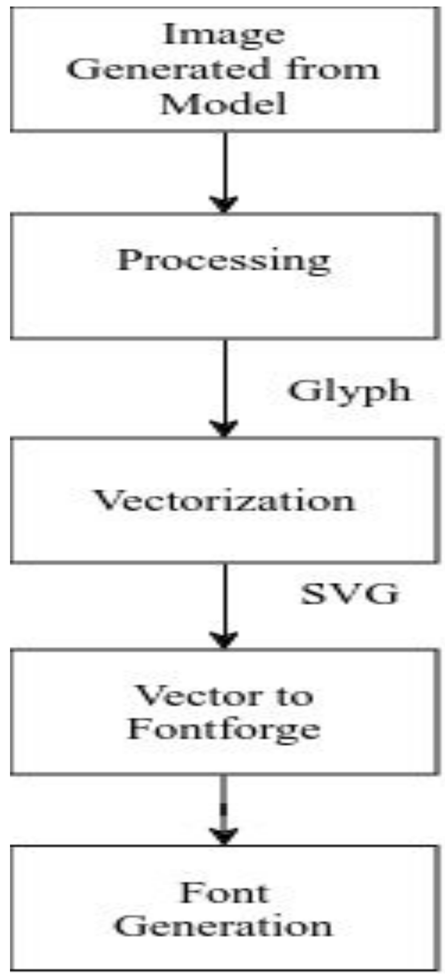
- Differentiates between real and generated handwriting samples to improve the generator's output through adversarial training.

# Methodology (Proposed Model Architecture)

## Style Classifier:

- Classifies and extracts individual handwriting styles, ensuring the generated handwriting matches the original style attributes of the writer.

# Methodology (Generated Image to Font file)



# Methodology
# (Generated Image to Font file)

## **Image Generated from Model**:

- Image of handwritten text generated by the handwriting cloning model.

## **Processing**:

- Images undergo some processing to prepare them for vectorization.

## **Glyph**:

- Processed images are converted into individual glyphs.

# Methodology (Generated Image to Font file)

## **Vectorization**:

- The glyph images are then vectorized.
- It involves tracing the outlines of the characters to create scalable vector graphics (SVG).

## **SVG**:

- The vectorized glyphs are stored in SVG (Scalable Vector Graphics) format.

# Methodology (Generated Image to Font file)

## **Vector to FontForge**:

- SVG files are imported into FontForge using its Python API.

## **Font Generation**:

- FontForge is used to generate the font file font format such as TrueType (.ttf) or OpenType (.otf).

# Methodology (Evaluation)

## FID(Fréchet Inception Distance)

- It is a metric used to measure the similarity between two sets of images

## Geometric Score

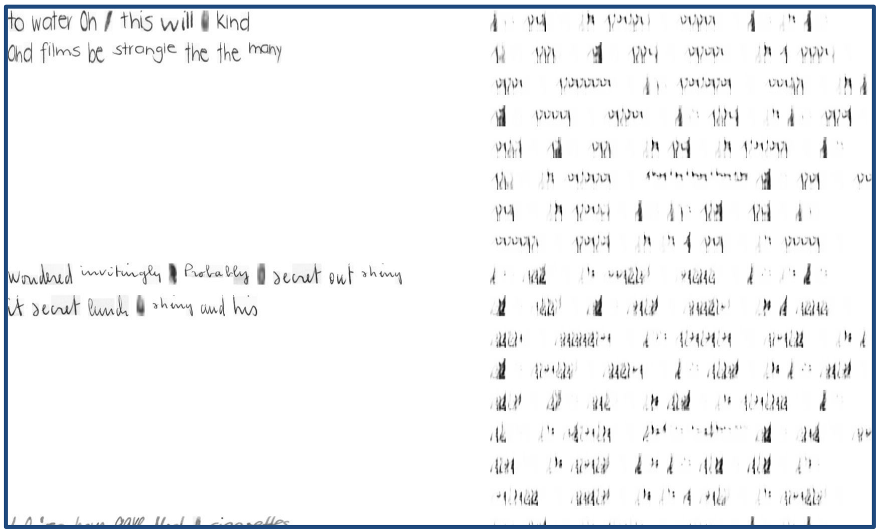
- It gives how closely the generated handwriting matches the topological properties of the original handwriting

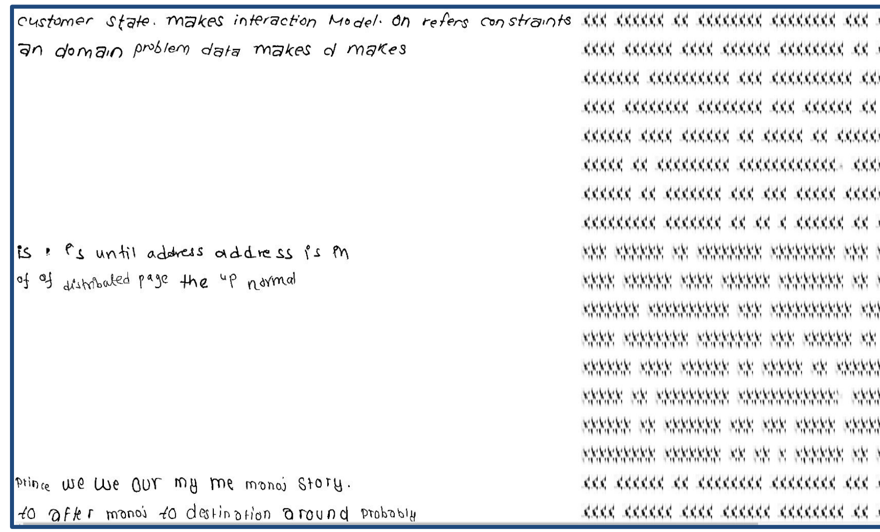
## Human Evaluation

- Finally, We perform our model performance using human experience.

# Results & Discussion

- So far we have trained our model on custom dataset that we have prepared and we ran around 130 epochs of training on that dataset and saw the result in validation set like this :

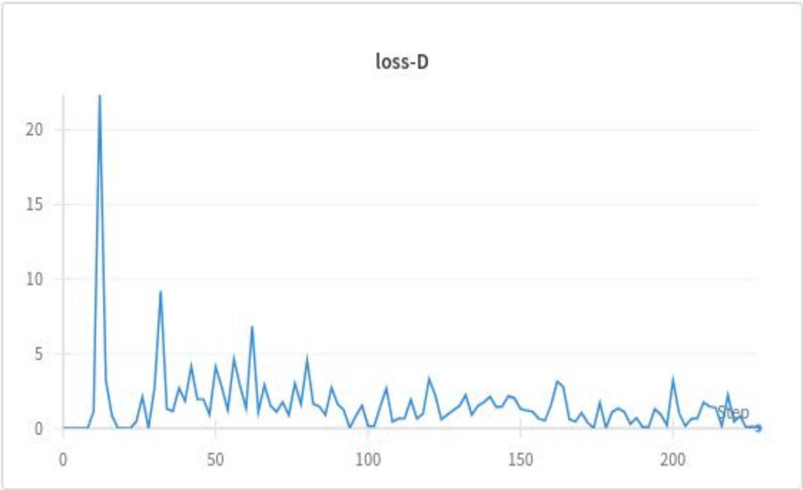


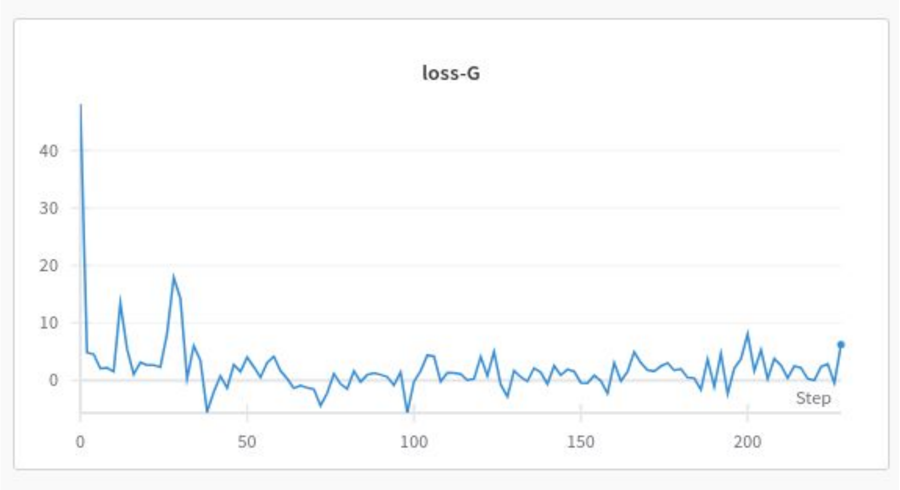



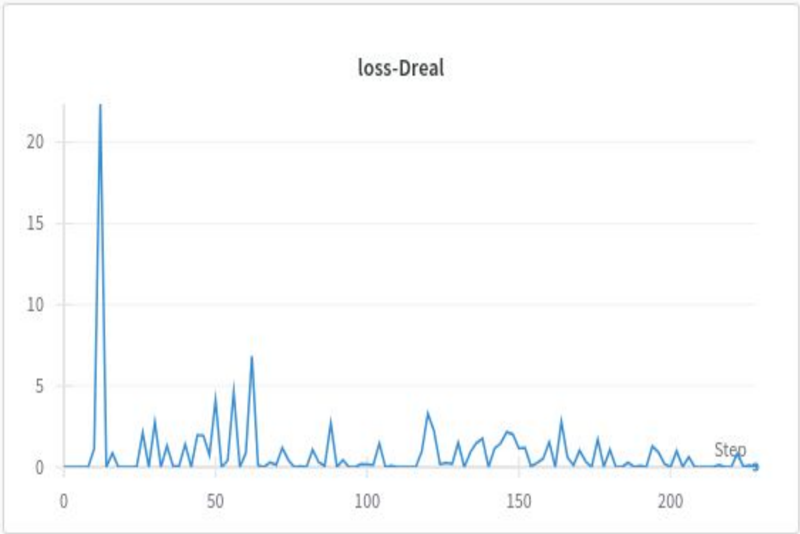

- So we can see that model is still learning the style characteristics of the writers present in the dataset.

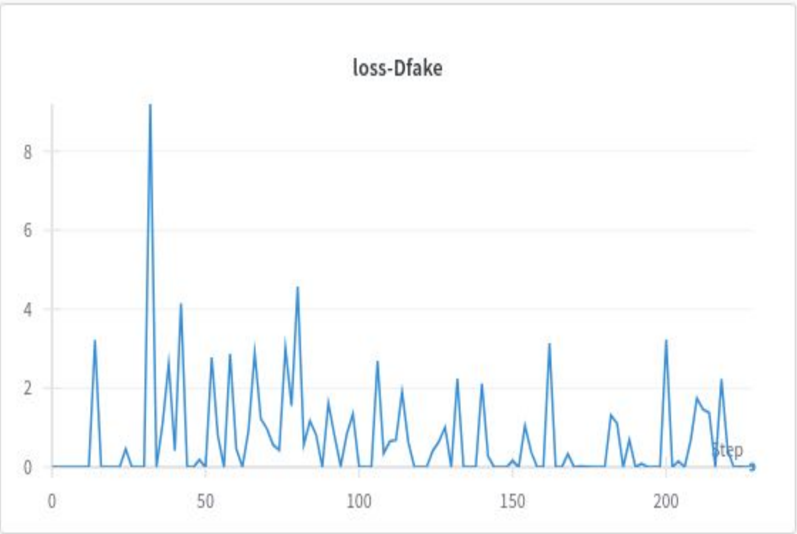
- To learn good style characteristics pattern model need to be trained to large number of epochs and on more datasets than we have trained the model on.

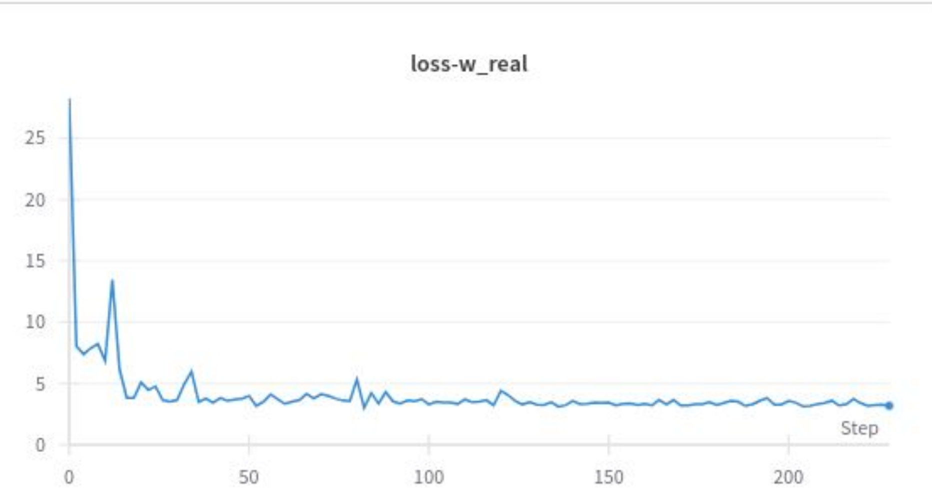
# Loss

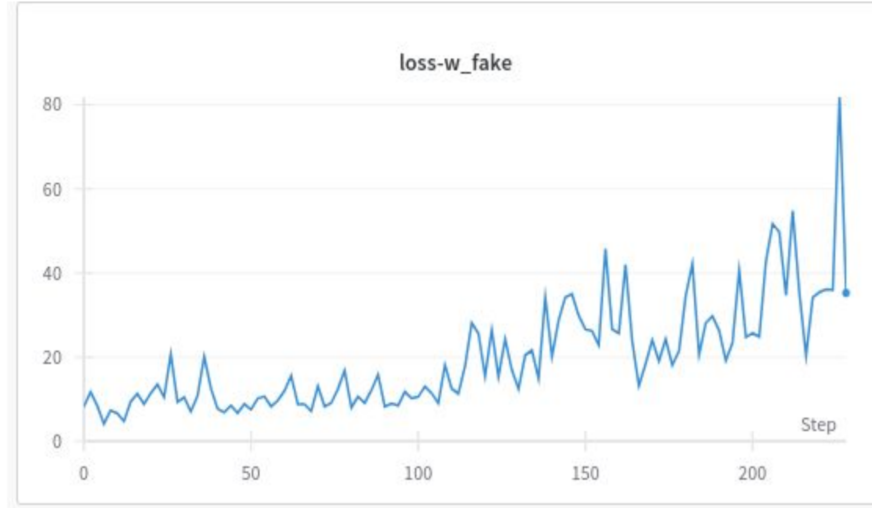

loss-D
20
15
10
5
0
0
50
100
150
200
Step

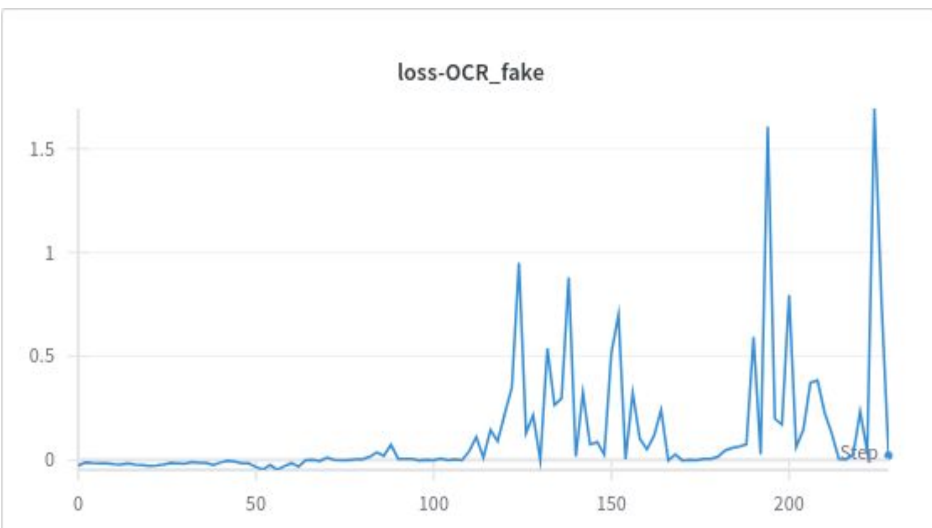

loss-G
40
30
20
10
0
0
50
100
150
200
Step

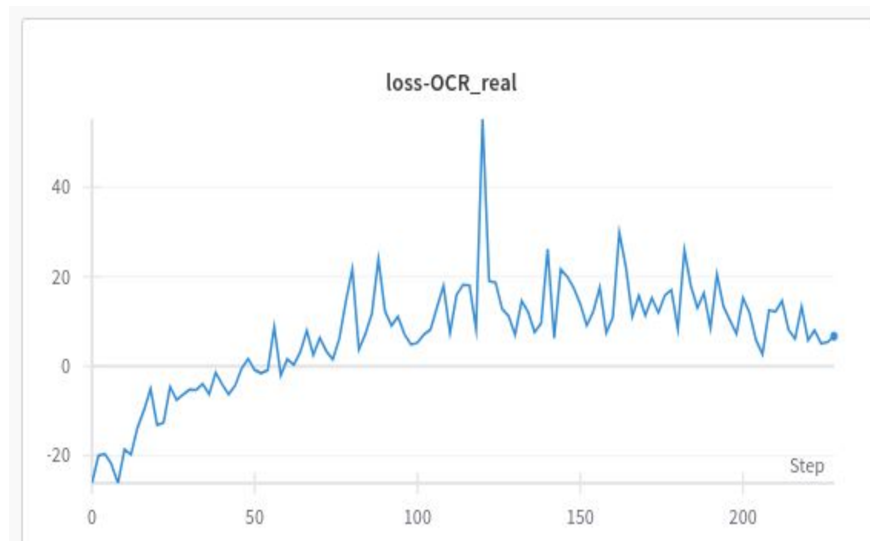

loss-Dreal
20
15
10
5
0
0
50
100
150
200
Step


loss-Dfake
8
6
4
2
0
0
50
100
150
200
Step


loss-w_real
25
20
15
10
5
0
Step
0
50
100
150
200


loss-w_fake
80
60
40
20
0
Step
0
50
100
150
200


loss-OCR_fake
1.5
1
0.5
0
Step
0
50
100
150
200


loss-OCR_real
40
20
0
-20
Step
0
50
100
150
200

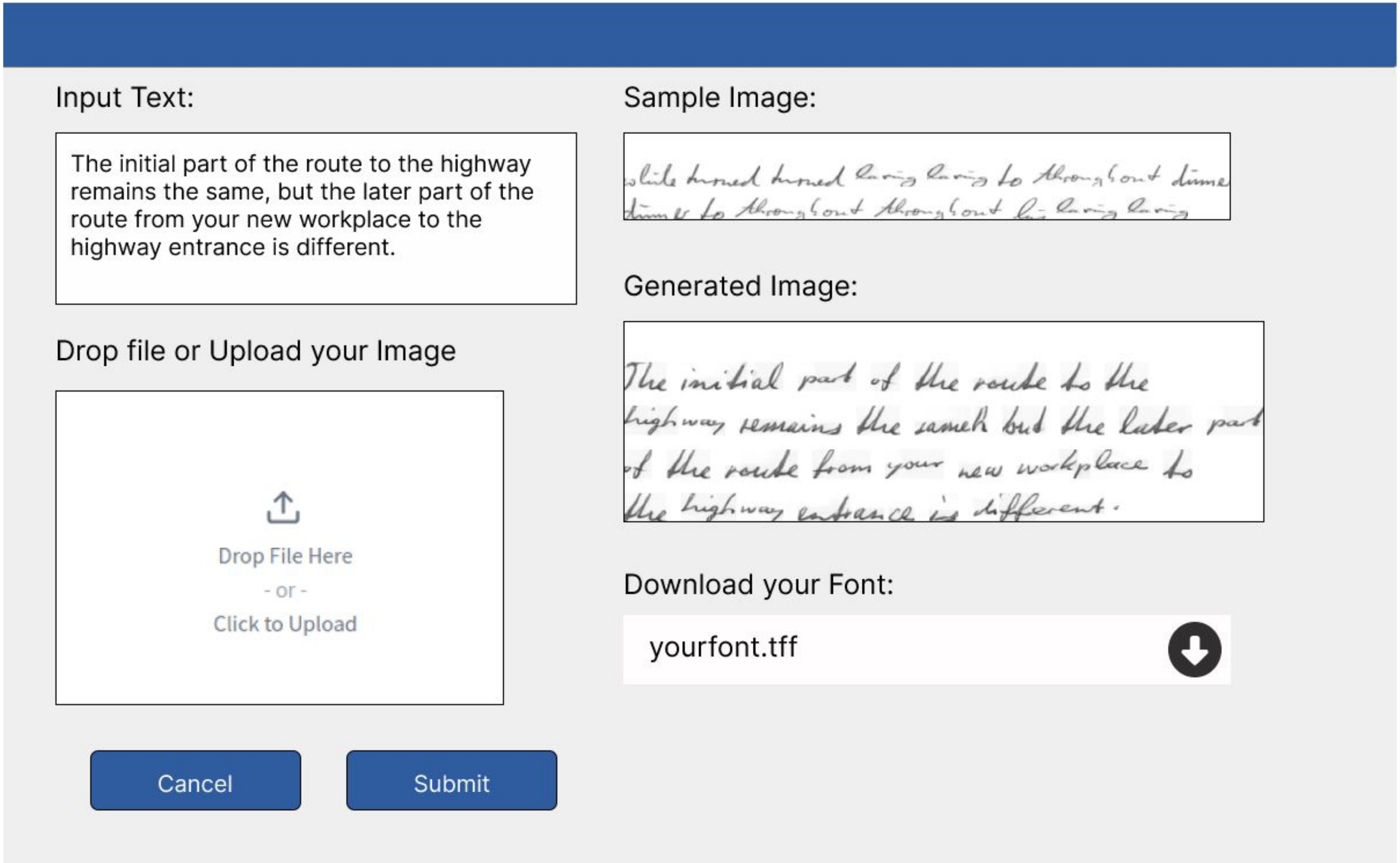
# Loss Analysis

- **loss-G**: 6.0051
  - Generator loss (G) - measures how well the generator produces realistic data.
- **loss-D**: 1.2253
  - Overall Discriminator loss (D) - measures the discriminator's ability to differentiate between real and fake data.
- **loss-Dfake**: 1.2253
  - Discriminator loss on fake data (Dfake) - indicates how well the discriminator identifies generated data as fake.
- **loss-Dreal**: 0
  - Discriminator loss on real data (Dreal) - shows how well the discriminator identifies real data as real.
- **loss-OCR_fake**: 0.0939
  - OCR network loss on fake data - measures the OCR network's ability to recognize generated handwriting.
- **loss-OCR_real**: 1.9465
  - OCR network loss on real data - measures the OCR network's ability to recognize real handwriting.
- **loss-w_fake**: 58.2379
  - Writer classification loss on fake data - indicates how well the classifier identifies the writer of generated handwriting.
- **loss-w_real**: 3.1669
  - Writer classification loss on real data - indicates how well the classifier identifies the writer of real handwriting.

# Remaining Task

- To further train the model and check model accuracy.
- To diversify our dataset including alphanumeric dataset.
- To work on creating font of generated dataset.
- To create website for project deployment.

# EXPECTED OUTPUT



# References

[1] E. Alonso, B. Moysset, and R. Messina, “Adversarial generation of handwritten text images conditioned on sequences,” in 2019 international conference on document analysis and recognition (ICDAR), IEEE, 2019, 481–486.

[2] B.Davis, C. Tensmeyer, B. Price, C. Wigington, B. Morse, and R. Jain, “Text and style conditioned gan for generation of offline handwriting lines,” arXiv preprint arXiv:2009.00678, 2020.

[3] A. K. Bhunia, S. Khan, H. Cholakkal, R. M. Anwer, F. S. Khan, and M. Shah, “Handwriting transformers,” in Proceedings of the IEEE/CVF international conference on computer vision, 2021, 1086–1094.

# References

[4] M. Zimmermann and H. Bunke, “Automatic segmentation of the iam off-line database for handwritten english text,” in 2002 International Conference on Pattern Recognition, IEEE, 4, 2002, 35–39.

[5] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative adversarial networks: An overview,” IEEE signal processing magazine, vol. 35, no. 1, 53–65, 2018. [6] S. Fogel, H. Averbuch-Elor, S. Cohen, S. Mazor, and R. Litman, “Scrabblegan: Semi-supervised varying length handwritten text generation,” in Proceedings of the IEEE/CVFconferenceoncomputervisionand pattern recognition, 2020, 4324 4333.