Artwork Classification and Style Transfer

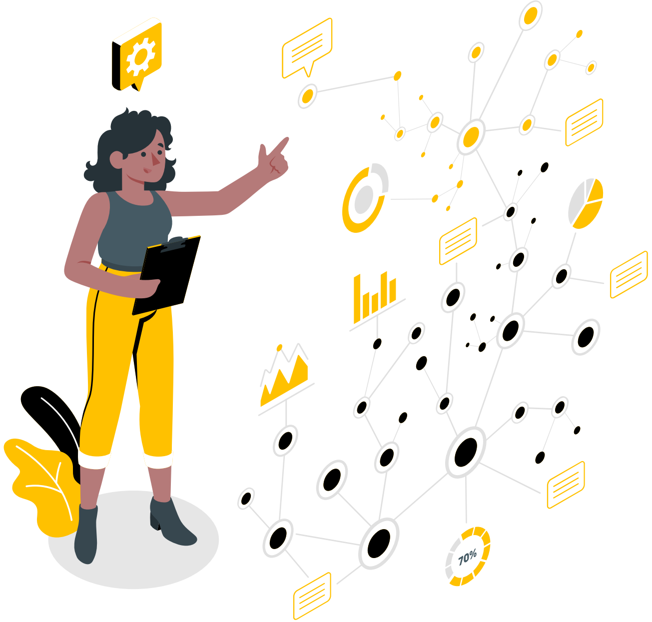
Computer Vision with Deep Learning
Yue Zhang

Agenda

1. Problem Statement
2. EDA
3. Problem 1: Classification
4. Problem 2: Style Transfer
5. Model Operation
6. Conclusion

Problem Statement

01



Problem Statement

- **Objective:** Image Classification & Style Transfer
- **Dataset:** Artworks images of 50 most influential artists
- **Image Classification:**
  - Custom CNN model
  - Transfer Learning model with ResNet50
- **Style Transfer:**
  - Style Transfer model with ResNet50 and VGG19
  - CycleGAN for Monet Style Transfer
- **Deployment:** Web Interface

Exploratory Data Analysis

02

Data Overview - CSV file

Data Source: Kaggle, “Best Artworks of All Time” by ICARO

Data Size: 50 rows and 8 columns

Missing Values: no missing values

Content:

```
First 5 rows of the artists file:
   id                 name         years                        genre  \
0   0    Amedeo Modigliani   1884 - 1920                 Expressionism
1   1  Vasiliy Kandinskiy   1866 - 1944  Expressionism,Abstractionism
2   2         Diego Rivera   1886 - 1957     Social Realism,Muralism
3   3         Claude Monet   1840 - 1926                 Impressionism
4   4        Rene Magritte   1898 - 1967    Surrealism,Impressionism

  nationality                                                bio  \
0     Italian  Amedeo Clemente Modigliani (Italian pronunciat...
1     Russian  Wassily Wassilyevich Kandinsky (Russian: Васи́...
2     Mexican  Diego María de la Concepción Juan Nepomuceno E...
3      French  Oscar-Claude Monet (; French: [klod mɔnɛ]; 14 ...
4     Belgian  René François Ghislain Magritte (French: [ʁəne...

                                          wikipedia  paintings
0  http://en.wikipedia.org/wiki/Amedeo_Modigliani       193
1  http://en.wikipedia.org/wiki/Wassily_Kandinsky        88
2       http://en.wikipedia.org/wiki/Diego_Rivera        70
3       http://en.wikipedia.org/wiki/Claude_Monet        73
4     http://en.wikipedia.org/wiki/René_Magritte       194
```

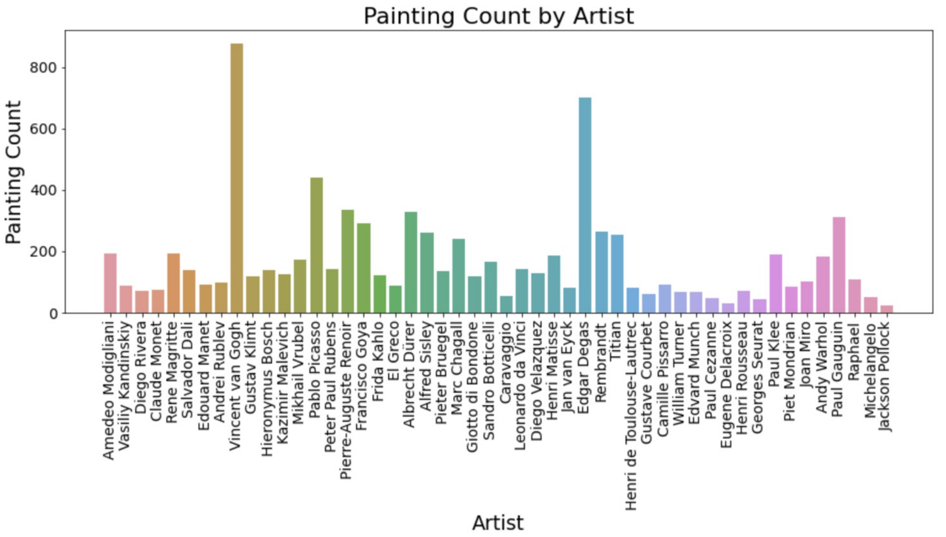
Schema:

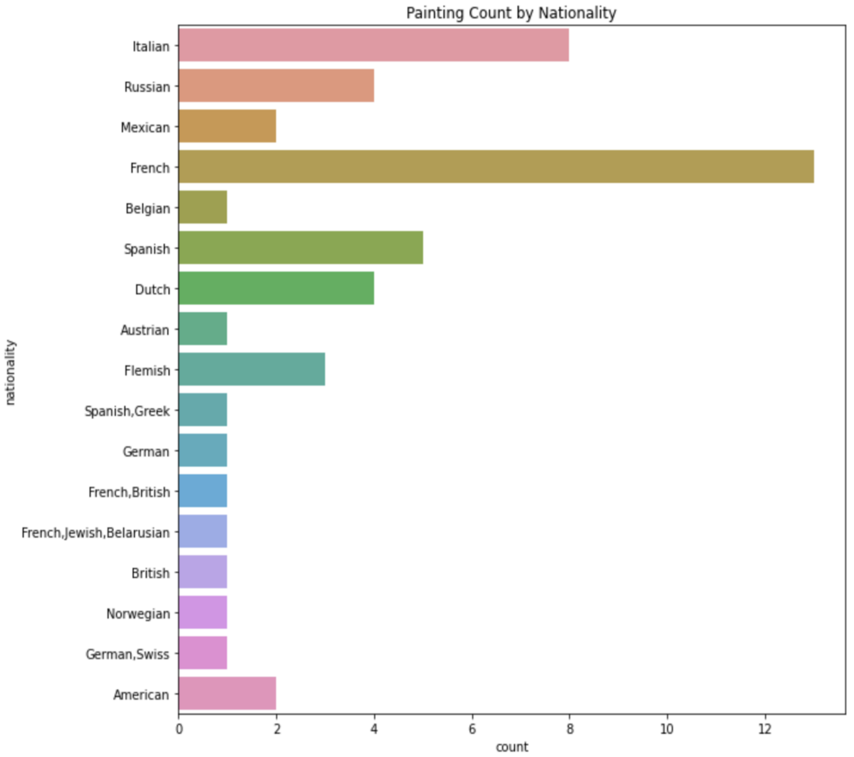
```
Schema of the artists file:
id              int64
name           object
years          object
genre          object
nationality    object
bio            object
wikipedia      object
paintings       int64
dtype: object
```

“Painting” analysis:

```
count     50.000000
mean     168.920000
std      157.451105
min       24.000000
25%       81.000000
50%      123.000000
75%      191.750000
max      877.000000
Name: paintings, dtype: float64
```

“Paintings” Visualization

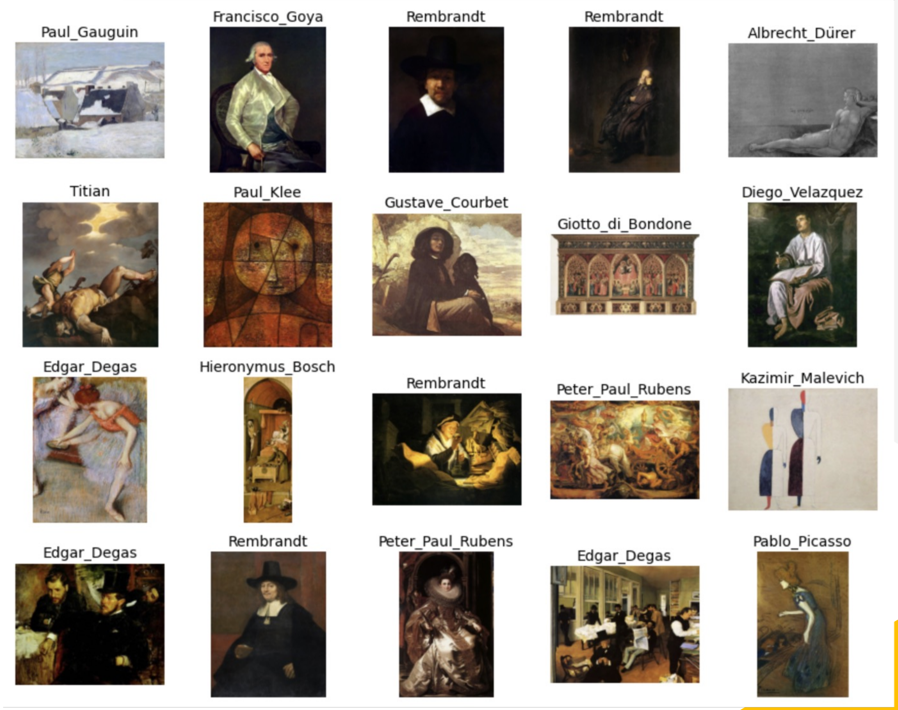




Data Overview - Image folder

Data size: 17,548 images

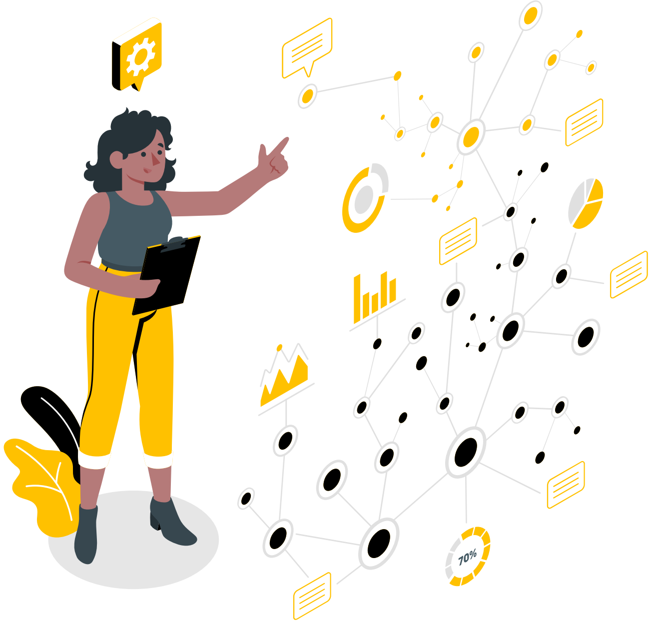
Unreadable or corrupted file: 0



Randomly selected images:

Problem 1: Classification

03



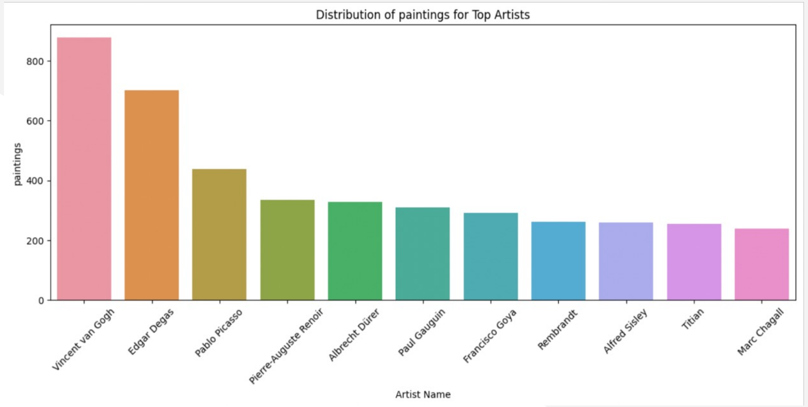
Classification

Sampling

- Choose artists with 200 or more paintings.

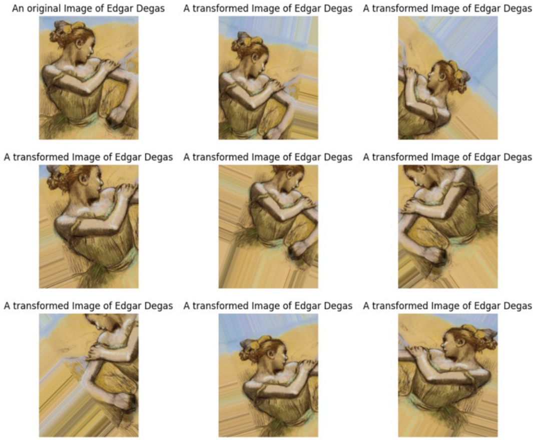
11 artists(classes) in total

Train/Validation = 80/20



Data Augmentation

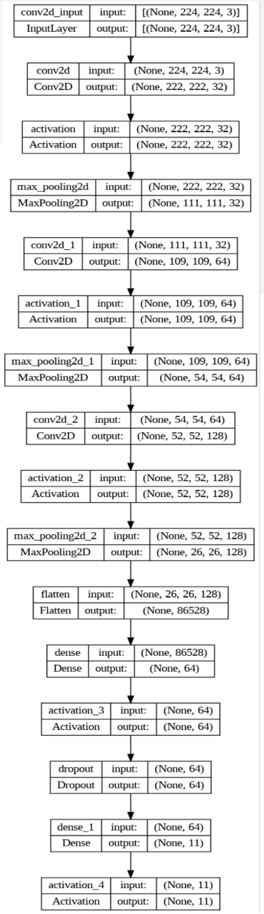
- Rescale
- Rotation
- Width Shift
- Height Shift
- Shear Transformation
- Zoom
- Horizontal Flip
- Fill Mode

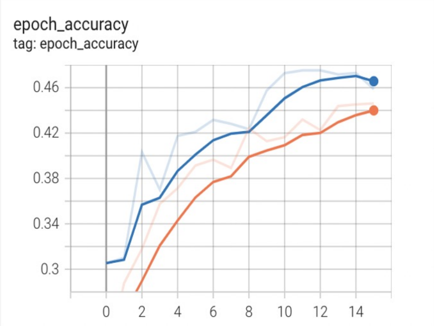


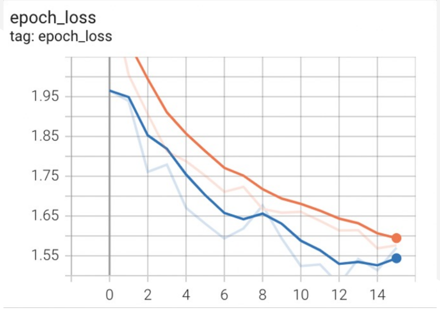
Classification

Model 1: Custom CNN

- **Input:** RGB images of size 224x224 pixels.
- **Layers:**
  - Conv2D (32 filters, 3x3 kernel) + ReLU
  - MaxPooling2D (2x2 pool size)
  - Conv2D (64 filters, 3x3 kernel) + ReLU
  - MaxPooling2D (2x2 pool size)
  - Conv2D (128 filters, 3x3 kernel) + ReLU
  - MaxPooling2D (2x2 pool size)
  - Flatten
  - Dense (64 units) + ReLU
  - Dropout (0.5)
  - Dense (11 units) + Softmax
- **Training:**
  - Loss: Categorical Crossentropy
  - Optimizer: Adam
  - Metrics: Accuracy







Train Accuracy	Validation Accuracy
0.512	0.451

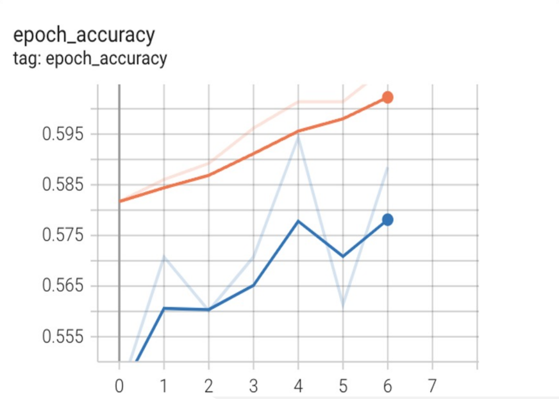
Classification

Custom CNN - After Hyper-Parameter Tuning

- **Tuned the number of neurons in the first dense layer and the learning rate of the optimizer:**
  - The best number of units in the dense layer is 384
  - The best learning rate in optimizer is 0.0001

- **The model's performance is slightly improved after parameter tuning**

	Train Accuracy	Validation Accuracy
Before	0.512	0.451
After	0.658	0.576

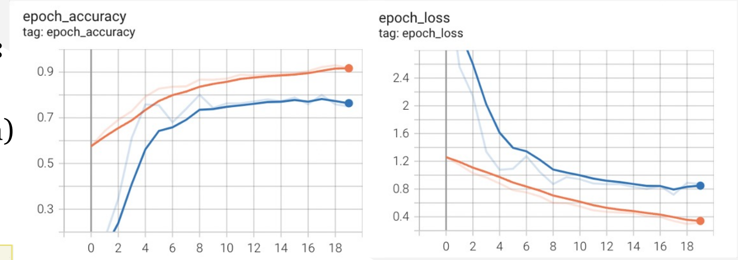


Classification

Model 2: Transfer Learning - Based on the ResNet50 architecture

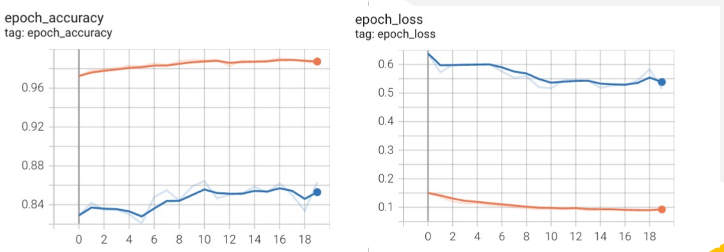
Phase 1 - Full Model training

- **Load pretrained ResNet 50 and add some additional layers:**
  - Flatten Layer
  - Dense layer with 512 units (ReLU activation and batch normalization)
  - Dense layer with 16 units (ReLU activation and batch normalization)
  - Dense layer with the number of classes and softmax activation.
- **Train the entire model**



Phase 2 - Fine-Tuning

- **Freeze the core layers** of the pre-trained ResNet50
- **Train only the additional custom layers and the top 50 layers of the base model**
- Use the ReduceLROnPlateau and EarlyStopping callbacks to control the learning rate and prevent overfitting



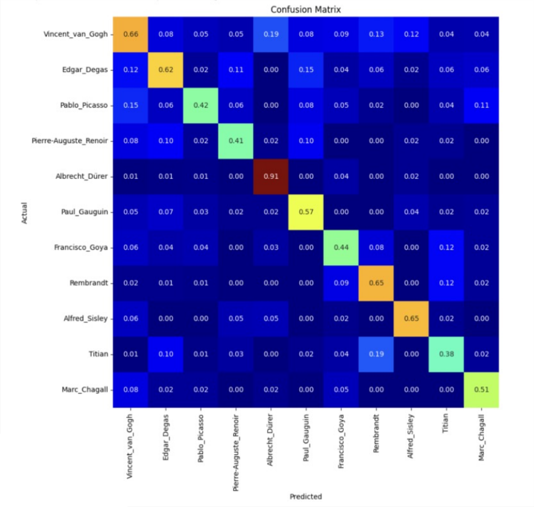
Train Accuracy	Validation Accuracy
0.9937	0.8571

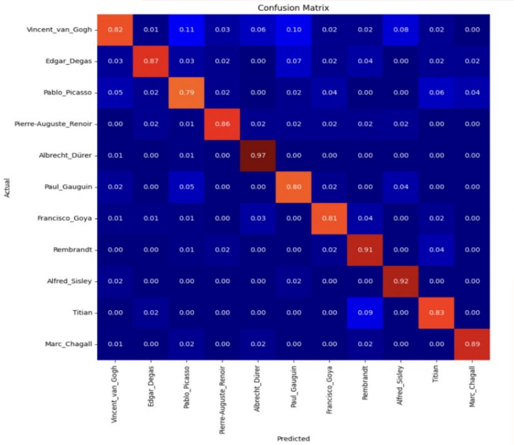
Classification

Model Evaluation and Comparison

	Train Accuracy	Validation Accuracy
Custom CNN after hyper-parameter tuning	0.6583	0.5761
Transfer Learning Based on ResNet50	0.9937	0.8571

- **From the accuracy table:** the transfer learning methods using ResNet demonstrated a significantly better performance, reaching to an accuracy of 86% on the validation set.
- **From the confusion matrix:** for the transfer learning approach, the majority of predictions fall on the correct class, resulting in fewer misclassifications compared to the custom CNN.
- **From the training efficiency:** The transfer learning approach significantly reduces the training time and computational resources, as it starts with pre-trained weights.





Classification

Classification Example

- **Predict on random images from dataset:**

Actual artist = Pierre-Auguste Renoir
Predicted artist = Pierre-Auguste Renoir
Prediction probability = 96.71 %

Actual artist = Rembrandt
Predicted artist = Rembrandt
Prediction probability = 99.08 %

Actual artist = Paul Gauguin
Predicted artist = Paul Gauguin
Prediction probability = 94.05 %

Actual artist = Titian
Predicted artist = Titian
Prediction probability = 97.71 %

Problem 2: Style Transfer

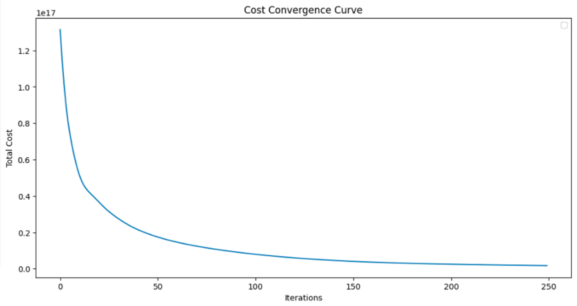
04

Style Transfer

Model 1: Transfer Learning Based on VGG19

Model Configuration:

Fully connected layer	Ignored
Model trainable	False
Content layers	'block3_conv4'
Style layers	'block1_conv1', 'block2_conv2', 'block3_conv3', 'block4_conv4', 'block5_conv2'
Random Search	'learning rate': 2.02, 'style wight': 0.03645, 'content wight': 0.0358



Cost Convergence Curve of VGG19

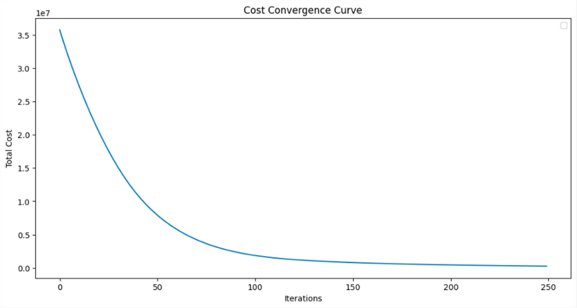
Style Transfer Result of VGG19

Style Transfer

Model 2: Transfer Learning Based on ResNet50

Model Configuration:

Fully connected layer	Ignored
Model trainable	False
Content layers	'conv3_block4_out'
Style layers	'conv1_relu', 'conv2_block3_out', 'conv3_block4_out', 'conv4_block6_out', 'conv5_block3_out'
Random Search	'learning rate': 1.5076, 'style wight': 0.06006, 'content wight': 0.0211

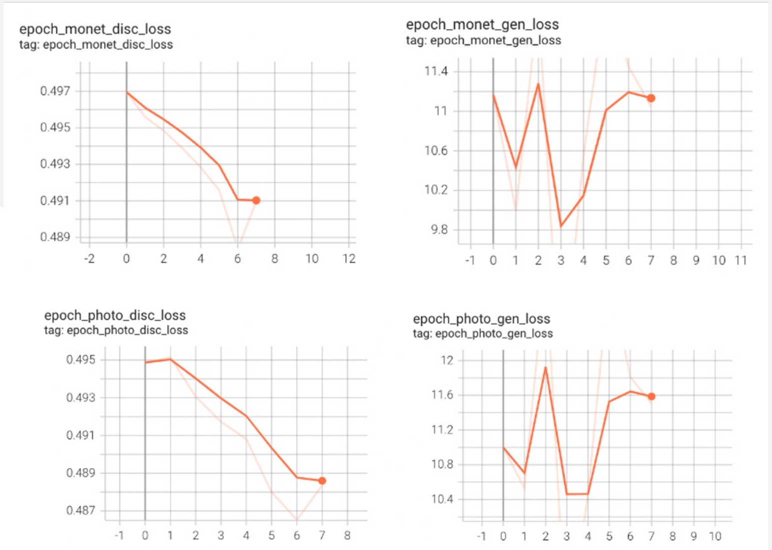


Cost Convergence Curve of ResNet50

Style Transfer Result of ResNet50

Style Transfer

Model 3: CycleGAN



loss converge curve of each discriminator and generator

Model Configuration:

- Monet Generator
- Photo Generator
- Monet Discriminator
- Photo Discriminator

Loss Function:

- Adversarial Loss
- Cycle Consistency Loss
- Identity Loss

Style Transfer Result of CycleGAN

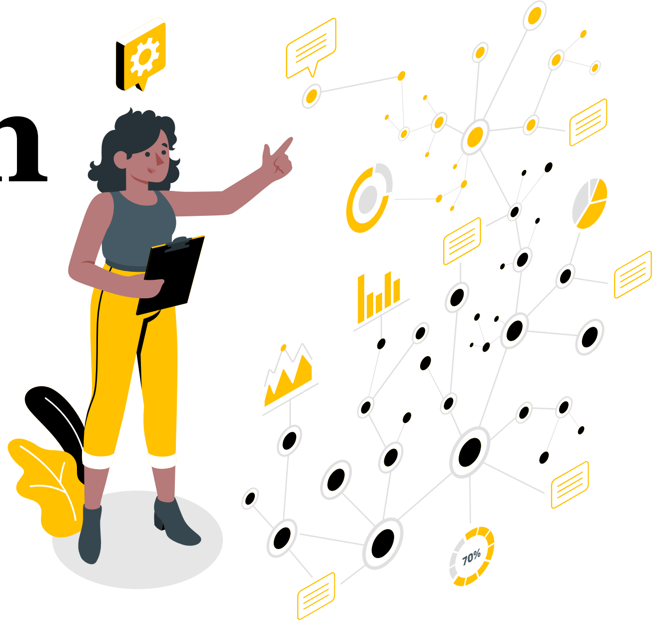
Style Transfer

Model Comparison

	Total Style	Color	Lines & Strokes	Total Performance
VGG19	Good	Good	Middle	Good
ResNet50	Best	Good	Best	Best
CycleGAN	Worst	Worst	Good	Worst

Model Operation

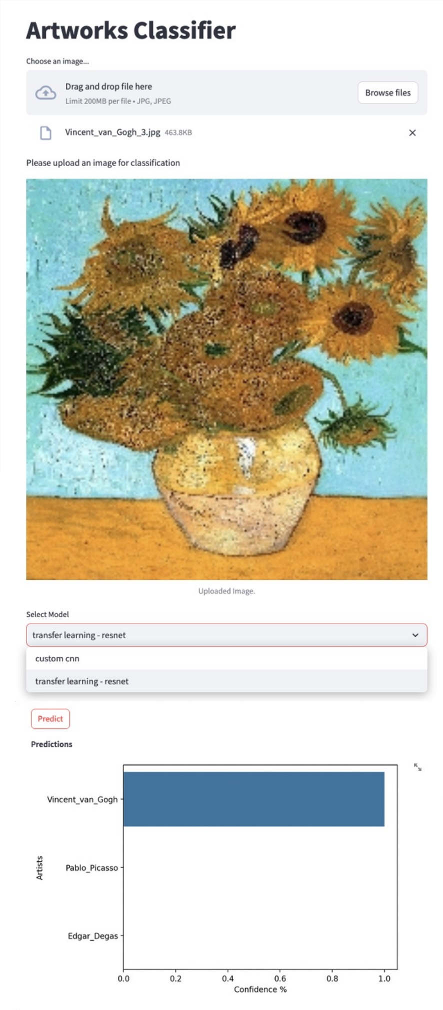
05



For Classification Models

Model Deployment

- **A web interface:**



Model Maintenance

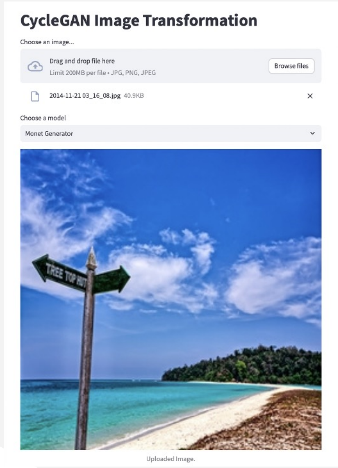
- **Performance Monitoring:** Monitor the model's performance metrics, including accuracy and confusion matrices, to detect any drop in performance.
- **Hyperparameter Tuning:** Continue hyperparameter tuning and adapt it to the changing data distribution. Use techniques like cross-validation and grid/random search
- **User Feedback and Monitoring:** Encourage user feedback to identify any potential misclassifications or inaccuracies in artist recognition. Regularly monitor user feedback to address issues promptly.

For Style Transfer Models

Model Deployment

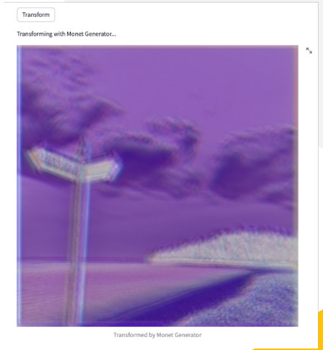
- **A web interface based on trained CycleGAN:**

  (Interface based on VGG19/ResNet50 takes too much time to generate images)



Model Maintenance

- **Performance Monitoring:** Monitor the model's performance metrics, such as visual quality, style fidelity, and content preservation, to detect any drop in performance.
- **Hyperparameter Tuning:** Continue hyperparameter tuning to optimize the style transfer algorithm and achieve better stylization results. Experiment with different combinations of hyperparameters to find the best settings.
- **User Feedback and Monitoring:** Encourage user feedback to identify any potential issues or shortcomings in the stylization process.



Maintenance in General

For both classification models and style transfer models:

- **Data Monitoring:** Continuously monitor the incoming data to identify any changes or shifts in the distribution of artworks. Regularly check for new artists or potential biases in the data.
- **Data Collection and Retraining:** Periodically collect new data and add it to the training dataset. Regularly retrain the model using the updated dataset.
- **Model Versioning:** Maintain different versions of the model to ensure backward compatibility and easy rollback if required. Implement versioning to keep track of model updates and improvements.
- **A/B Testing:** Implement A/B testing for new model versions before deploying them to the production environment. Compare the performance of the new version against the existing model to ensure it meets or exceeds the desired accuracy/stylization quality.
- **Continuous Integration and Deployment (CI/CD):** Automate the deployment process using CI/CD pipelines to ensure seamless and frequent updates to the model.
- **Error Handling and Logging**: Implement robust error handling and logging mechanisms to detect and resolve any issues that may arise during model deployment or inference.

Conclusion

06

Conclusion

Findings:

- **Classification:** The transfer learning model rooted in the ResNet architecture outperformed the custom CNN model after hyperparameter tuning.
- **Style Transfer:** VGG19 and ResNet50 both transferred the styles successfully. The convergence of both models after a span of 50 epochs indicates the stability and effectiveness of our design and training procedures.
- **CycleGAN for Monet-style Transformations:** Our CycleGAN model aimed at transforming general photos to mimic Claude Monet's painting style. However, the outcomes were not entirely satisfactory.
- **Web Interface Deployment:**

- The classification interface proved effective and efficient, delivering favorable results.
- The style transfer interface, although functional, encountered performance issues. Due to reliance on a CPU on local setup, the computation was notably protracted. Additionally, the interface exhibited a relatively higher loss.

Next Steps:

- Optimize the CycleGAN model for Monet-style Transformations, including hyper-parameter tuning, loss function redefining, or even trying a new architecture for the generator or discriminator.
- Refine style-transfer interface, possibly exploring hardware-accelerated solutions or model optimizations to ameliorate loss issues in generated images.

Questions and Feedbacks

Appendix

Classification — Models' Pros and Cons

	Pros	Cons
Custom CNN	• **Flexibility:** The custom CNN allows full control over the architecture, providing flexibility to tailor it to the specific task and dataset. • **Interpretability:** Being a manually designed model, it is easier to interpret and understand the learned representations	• **Architecture Complexity:** Designing an optimal CNN architecture requires expert knowledge and extensive experimentation, which can be time-consuming and challenging. • **Limited Generalization:** The custom CNN may struggle to generalize well to diverse artistic styles or when the training data is limited.
Transfer Learning Model Based on ResNet50	• **High Accuracy:** Leveraging knowledge from a large dataset enables the model to capture intricate features and achieve superior accuracy. • **Generalization**: ResNet50 is highly suitable for tasks with diverse and complex data, such as artist recognition. • **Faster Training**: Fine-tuning a pre-trained model significantly reduces the training time and computational resources required compared to training from scratch.	• **Limited Interpretability:** The complexity of the pre-trained ResNet50 model may make it less interpretable, as the learned representations are not explicitly defined. • **Domain Shift:** The pre-trained model may have been trained on images from different domains, potentially causing a domain shift that could affect performance.

Style Transfer — Models' Pros and Cons

	Pros	Cons
VGG19 Transfer Learning	• **Proven Architecture:** VGG19 has been validated on large image datasets and exhibits good performance. • **Style and Content Separation:** The intermediate layers of VGG19 are used to separate style and content, making style transfer more intuitive.	• **Computational Resources:** Requires relatively more computational resources, especially for high-resolution images. • **Not Real-time:** For every pair of style and content images, a separate optimization process is needed.
ResNet50 Transfer Learning	• **Depth:** Due to its depth, ResNet50 can capture complex style features. • **Residual Connections:** Improves the training process, allowing for deeper network architectures.	• **Computational Demands:** Similar to VGG19, ResNet50 also requires a significant amount of computational resources. • **Potential Overfitting:** Might overfit on small datasets.
CycleGAN	• **Unpaired Training:** No need for paired style and content images, making data collection easier. • **Bidirectional Transformation:** Can transform from one domain to another and then transform back.	• **Training Complexity:** The training process for CycleGAN is more complex and unstable • **Computational Resources:** Requires a large amount of computational resources due to its need of training separately on the entire set of style images and content images.

Reference

[1] Main dataset: https://www.kaggle.com/datasets/ikarus777/best-artworks-of-all-time

[2] Extra data for style transfer: https://www.kaggle.com/datasets/ayaderaghul/gan-getting-started-2

[3] Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A Neural Algorithm of Artistic Style. arXiv preprint.

[4] Luan, F., Paris, S., Shechtman, E., & Bala, K. (2017). Deep Photo Style Transfer.

[5] Dumoulin, V., Shlens, J., & Kudlur, M. (2017). A Learned Representation For Artistic Style. International Conference on Learning Representations (ICLR). Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[6] Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. arXiv preprint.

[7] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. Proceedings of the IEEE International Conference on Computer Vision (ICCV).

[8] Choi, Y., Choi, M., Kim, M., Ha, J. W., Kim, S., & Choo, J. (2017). Toward Multimodal Image-to-Image Translation. Advances in Neural Information Processing Systems (NIPS).