StarryNight:
Artist classification and Image Style Transfer with Neural Networks

Xiaozhou Feng, Xiaoyu Liu, Estefany Nunez
Bryan Reynolds, Kai Wei
Mentor: Francisco Martinez

The Erdős Institute, 2022 Bootcamp
June 4, 2022

Overview

- Facts
  - We are not artists or art experts, but enjoy art and have favorite styles
    - We wanted to make a tool to help recognize and learn about different artists and styles
- Targeted Users
  - People interested in identifying the artist of an unsigned/untitled painting
  - Aspiring art aficionados interested in learning to recognize different painting styles
  - Social media internet users interested in sharing fun custom images
- Business Opportunity
  - We created a classifier using machine learning to predict the artists of paintings, and
  - Transfer famous artist's painting styles to any image using the same neural network concepts
- Techniques:
  - Software: PyTorch, Numpy, Seaborn, Pandas, Scikit learn, Gradio, TensorFlow
  - Platforms: Jupyter Notebooks, Google Colab, Kaggle, Hugging Face

Data Gathering

Kaggle datasets:

- Image**:** *Claude Monet and Vincent van Gogh*
- Meta data**:** *wikiart*

WikiArt database:

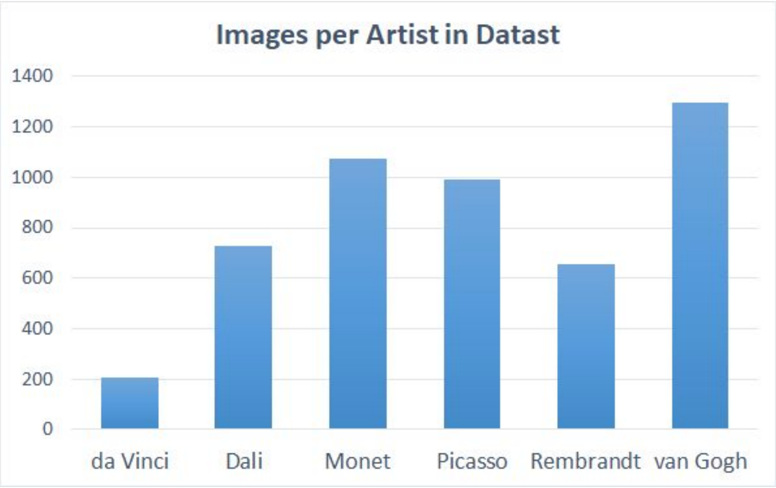
- Data scraping from wikiArt using above meta data for *Leonardo da Vinci, Rembrandt, Pablo Picasso* and *Salvador Dali* paintings
- Dataset released as public dataset in Kaggle https://www.kaggle.com/datasets/czkaiweb/subwikiarts

Exploratory Data Analysis

- Data cleaning
  - Checked for and removed duplicate images.
  - Manually identified and removed any non-painting images.
    - photos, sculptures, 3-D art installments, etc.
  - Images with 1 or 4 channels (grayscale and CMYK, respectively) were dropped.

- Image features
  - Strength in RGB channels are normalized per channel.

- Data reweighting
  - This is an imbalanced dataset- the number of images by different artists is not uniform (right).
  - Reweight applied in PyTorch to pick images from each artist with equal probability.



Modeling Approach

- Data splitting
  - Used 20%, 10% and 70% of images in the train, validation and test sets, respectively.
  - Relatively small training set chosen to learn from limited input images and maximize test set.

- Data augmentation
  - Given the small training set, training images are used 3 times each, with random cropping, rotation, and mirroring used to create functionally unique training images.

- Model fine-tuning
  - Models modified from pre-trained CNN models: EfficientNet, ConvNext, MobileNet, ResNet and VGG16.
  - Pre-trained weight loaded and fine-tuned for our feature and classifier layers.
  - *All models used the same training/validation/test set.*

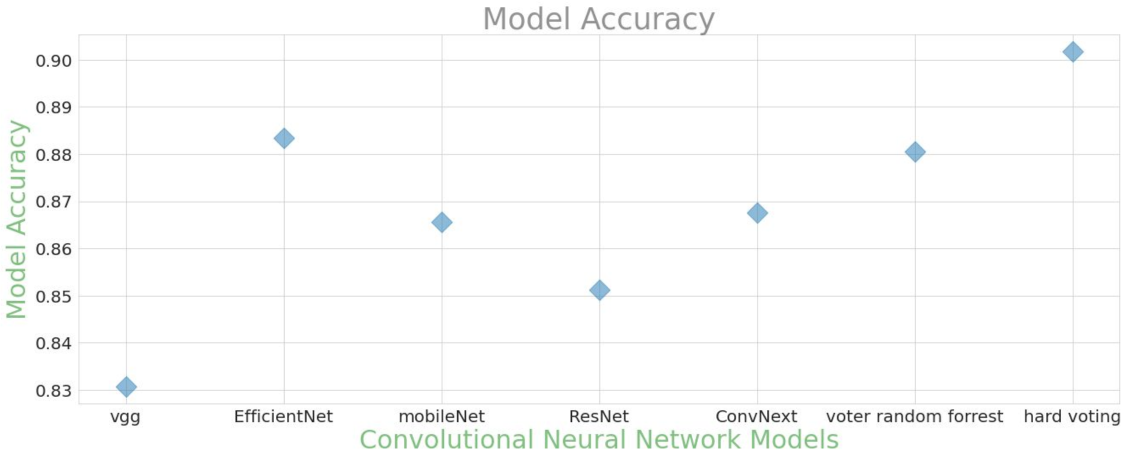
- Model bagging
  - Voting methods tried with outputs from 5 CNN model predictions.

Classifier Pipeline

Input	Image Transform	Train Model	Output
Clean data, randomized and split into three sets: training (20%) , validation (10%), and test (70%)	Sample 224 x 224 pixels of each images in each dataset	EfficientNet, ConvNext, MobileNet, ResNet, VGG **Voter**	Predictions: Dali, Monet, Picasso, Rembrandt, da Vinci or Van Gogh

Model Accuracy and Results

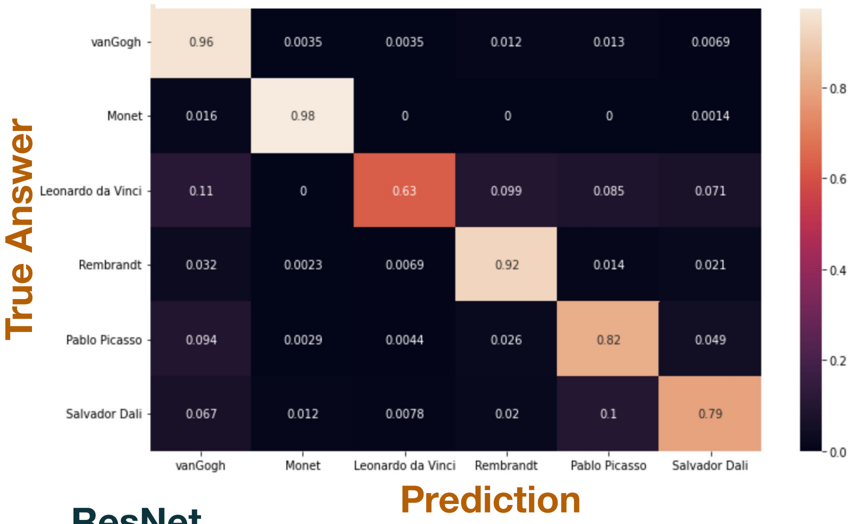
- Individual model accuracy ranges between **83% - 88%**
- Implementing a voter model doesn't increase performance by much
  - EfficientNet alone did slightly better than the random forest voter
- Hard voting performs the best at **90%**
  - Uses a majority vote from the 5 models, to pick the artist from an image



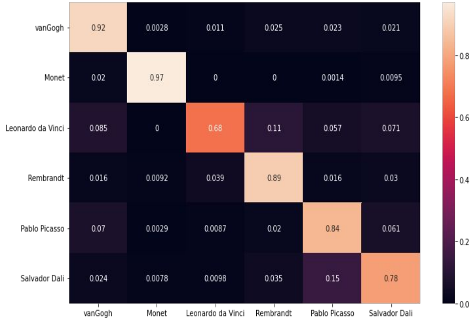
Results Continued: Confusion Matrices

- We used normalized confusion matrices to visually compare our individual model performances.
    - The y-axis represents the true answer and the x-axis the predicted
- Models perform similarly for different artists, but with some variation in accuracy
    - da Vinci seems to be the toughest to classify in all models

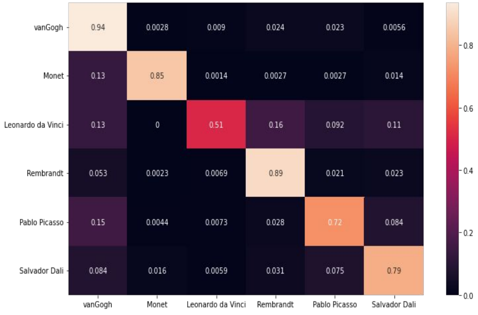
Hard Voter



EfficientNet

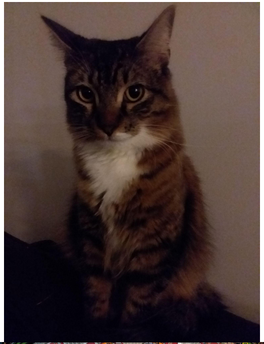


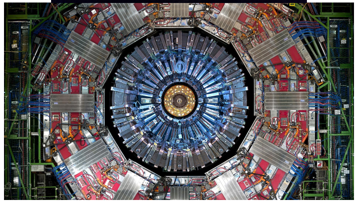
ResNet



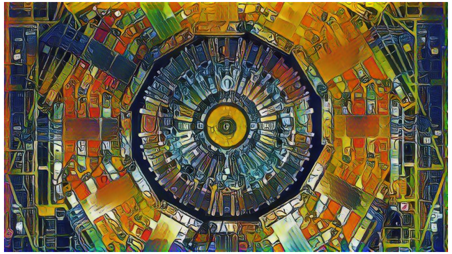
Style transfer with CNNs

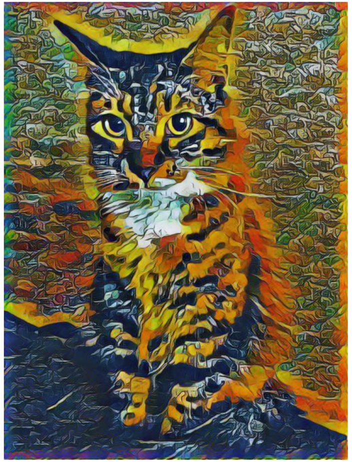
- The same techniques used by pre-trained CNNs to classify different images based on features of their artist’s style can be used to transfer those features to other images.
- A model can be fine-tuned on an image to learn its style, and then be applied to a new image.
  - When style loss (from feature layers) and content loss (from input layers) are both minimized, the result is an altered image with the style applied.
  - A popular tool for this is Magenta, which uses the MobileNet CNN to achieve style transfer.





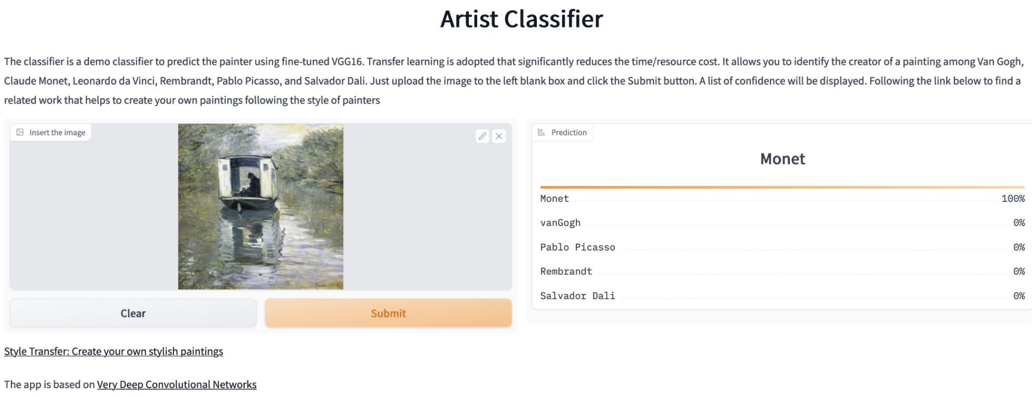
Fine-tuned CNN

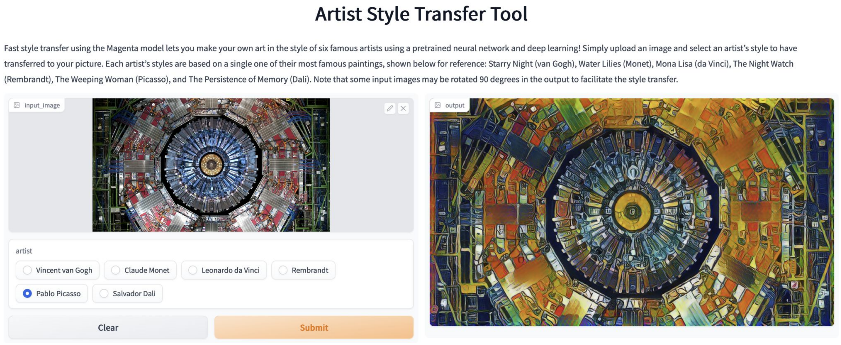




Interactive Applications

- Our models are live!
- The artist classifier takes an image as input and classifies the artist.
  - Limited to Monet, van Gogh, Picasso, Rembrandt, da Vinci, and Dali
- The artist style transfer tool takes an image and outputs a stylized version.
  - You can choose between the artists in the scope of this project





[Artist Classifier](#) [Image Style Transfer](#)

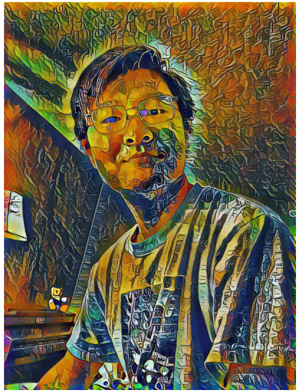
Next Steps

- Expand the artist dataset to include more famous painters.
- Branch out to classifying different types of art (not just paintings).
- Optimize web apps to run more quickly.
- Implement image classifier into a mobile app to classify images directly from the phone's camera or gallery.
- Improve app to recommend a style transfer option to users based on which artist's style the input image is most similar to.

Thank You

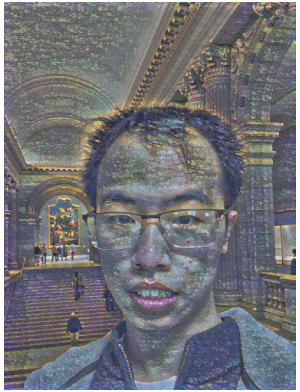
Estefany Nunez
https://github.com/marthaEstefany
estefany.nunez01@gmail.com

Bryan Reynolds
breynolds1247@gmail.com
https://github.com/breynolds1247
linkedin.com/in/bryanreynolds93/



Kai Wei
czkaiweb@gmail.com
https://github.com/czkaiweb
linkedin.com/in/kai-wei-aa840b151/

Xiaoyu Liu
https://github.com/liuxiaoyuyuyu
liu.6566@osu.edu



Xiaozhou Feng
https://github.com/jisutich
feng.933@osu.edu