A Study on Artist Attestation

Group 13: Alex Armstrong, Minh Dang, Shashank Saran

Overview

- Inspiration and Aims of this project
- Approach
    - Setup
    - Dataset and Preprocessing
    - Model Architectures & Implementation Details
    - Experiments
- Conclusion
- Questions

Task

- Identifying artist based on the paintings
- Important for cataloguing art added to new and ever-growing collections.
- Also used to identify forgeries.
- Traditionally done manually.
- Prior machine learning based approaches have tried using handcrafted features or generic features such as SIFT and Histogram of Gradients.

Aims of this project

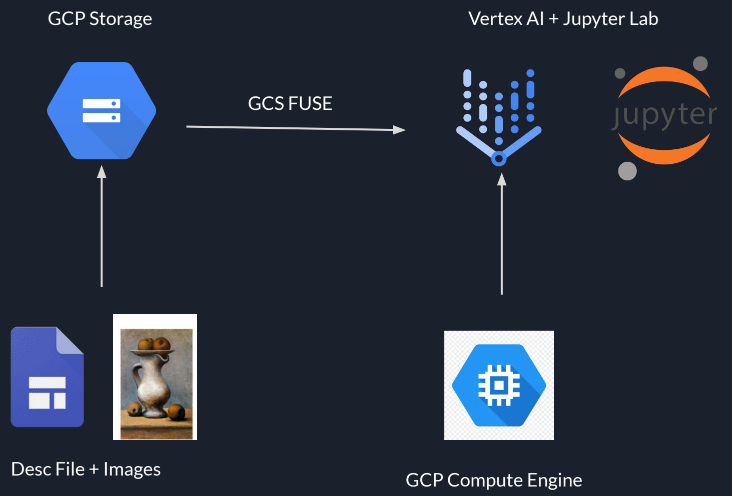
- Understand how well CNNs perform on this task of identifying art.
- Understand how well large models like Resnet trained on standard image recognition tasks adapt to the task of artist attestation.
- Experiment to see how well these models work under noisy conditions such as images captured from standard smartphone cameras.

Process

- There are 4 main steps in our projects:
  - Set up infrastructure
  - Data gathering and pre-processing
  - Building the models.
  - Testing on test set and unclean smartphone images.

Setup

- Due to large amount of image data needed to preprocess and perform modeling, running these tasks in local machine can be hard and inefficient. We decide to host and run our model on Google Cloud Platform
- Architecture:



Dataset and Preprocessing

- We use the WikiArt Dataset, which contains over 100,000 labeled images..
- Since this dataset contains many artists who have very few paintings (< 10), we pick only artists who have more than 300 paintings to prevent class imbalance and to ensure strong class representation. This prunes our dataset down to about 17,000 paintings and 57 artists.
- For preprocessing, we follow the following steps:
  - Zero-Centering and Normalization
  - Training Dataset - Random horizontal flip and random 224X224 crop of the image.
  - Testing Dataset - 224 X 224 center crop of the image.
- These preprocessing steps are applied to each batch of images.

https://www.kaggle.com/competitions/painter-by-numbers/data

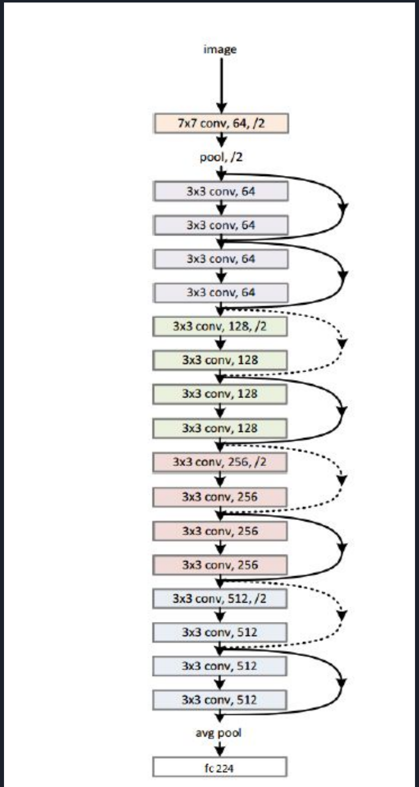
Model Architectures

- In this project, we experiment with three different models:
  - A simple baseline Convolutional Neural Network
  - Resnet-18, an 18 layer deep model, trained from scratch on our dataset.
  - Transfer learning with Resnet-18 pre-trained on ImageNet, keeping the weights frozen during training.
- Each of these models has their output layer modified to be able to predict between one of the 57 artist classes.
- We use Adam optimizer for our gradient updates and Categorical Cross Entropy loss for our loss function along with a Softmax activation on our output. We use ReLU for our activation function.
- Weight Initializations are done using sampling from a Gaussian distribution with mean = 0 and SD = √(2/layer_size).
- Hope to experiment with more pre-trained models if time permits!

Model Architectures

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1        [-1, 32, 112, 112]             896
       BatchNorm2d-2        [-1, 32, 112, 112]              64
         MaxPool2d-3          [-1, 32, 56, 56]               0
            Conv2d-4          [-1, 32, 28, 28]           9,248
       BatchNorm2d-5          [-1, 32, 28, 28]              64
         MaxPool2d-6          [-1, 32, 14, 14]               0
            Linear-7                  [-1, 228]       1,430,244
       BatchNorm1d-8                  [-1, 228]             456
            Linear-9                   [-1, 57]          13,053
================================================================
Total params: 1,454,025
Trainable params: 1,454,025
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 7.33
Params size (MB): 5.55
Estimated Total Size (MB): 13.45
----------------------------------------------------------------
```

Baseline CNN Model



ResNet-18

Training Results

Model	Top - 1 training accuracy	Top - 3 training accuracy
Baseline CNN	31.95	52.6
ResNet - 18 from scratch	34.05	56.1
ResNet-18 Pre-trained	**75.17**	**89.85**

Implementation Details

- We use PyTorch’s Dataset and DataLoader classes along with Torchvision transforms for preprocessing.
- We use Pytorch’s torch.nn module to build our baseline network and Torchvision for our pre-trained networks.

Testing on noisy images

- We know that most modern networks are capable of learning and predicting classes on pristine images.
- But are these networks able to cope with noise in the image and still make a correct prediction?
- Smartphone cameras much lower quality than professional cameras which:
    - Use several image processing tasks to enhance image
    - Big changes to original image (mainly lighting and colors)

Original Image

ASUS ROG Phone 3 Image

Oneplus 7 Pro Image

Experiment Overview

- Take 56 test images from 56 different artist from the original test dataset.
- Create 2 new datasets by taking them on smartphones (100 images)
- Preprocess the images to standardize them.
- Run the built models on these images
- Check accuracy of the results

Experiment Devices

- Original Phones
  - ASUS ROG Phone 3 - 64 Megapixels
  - Oneplus 7 Pro - 48 Megapixels
- Photos taken on default camera apps
- Both devices use software to modify the image internally

ASUS ROG Phone 3

Oneplus 7 Pro

Testing Results

Model	Top - 1 training accuracy	Top - 3 training accuracy	Precision	Recall
Baseline CNN	31.9	52.3	32.4	31.9
ResNet - 18 from scratch	33.4	55.4	35.7	33.4
ResNet-18 Pre-trained	**67.1**	**83.4**	**68.2**	**67.1**

ASUS Results

Model	Top - 1 training accuracy	Top - 3 training accuracy	Precision	Recall
Baseline CNN	10.7	19.6	3.9	10.7
ResNet - 18 from scratch	5.4	21.4	3.6	5.4
ResNet-18 Pre-trained	30.4	50.0	22.6	30.4

One Plus Results

Model	Top - 1 training accuracy	Top - 3 training accuracy	Precision	Recall
Baseline CNN	10.7	26.8	5.2	10.7
ResNet - 18 from scratch	7.1	30.4	5.4	7.1
ResNet-18 Pre-trained	42.9	58.9	32.7	42.9

Questions?