Art Forgery Detection

Tawfik Abbas

Motivation

- There is an immense cultural, and often, monetary value placed upon art, art forgery poses significant risk to the integrity of museums and other art collections
- Hence, we may posit the following question: Given a work of art, specifically a painting, and an artist, can we reliably determine if that painting had been created by that artist?

Introduction - What is Art Forgery

Art forgery - “...the creating and selling of works of art which are falsely credited to other, usually more famous artists.”

Dataset

- The training and validation datasets were sourced from Kaggle
- Composed of roughly 5200 images/paintings from 10 Impressionist artists split in the following manner:
    - Training dataset comprised of 4158 images
    - Validation dataset comprised of 990 images
- The test dataset is composed of a handful of individual images known to be art forgeries


Vincent

Dataset - cont’d

- The Impressionist artists whose paintings are within the dataset include:
  - Paul Cezanne, Edgar Degas, Paul Gauguin, Childe Hassam
  - Henri Matisse, Claude Monet, Camille Pisarro
  - Pierre-Auguste Renoir, John Singer-Sargent, Vincent van Gogh

Data Preprocessing

- The dataset required some pre-processing to make it suitable for model training
  - Images were resized and rescaled
  - As there were only 400 images per class for training, data augmentation was utilized to synthesize more images for training to mitigate overfitting

Model Construction

As this is an image classification problem, we will be making use of Convolutional Neural Networks (CNNs). There were two available approaches to model development:

- Layer-by-layer construction
- Transfer Learning

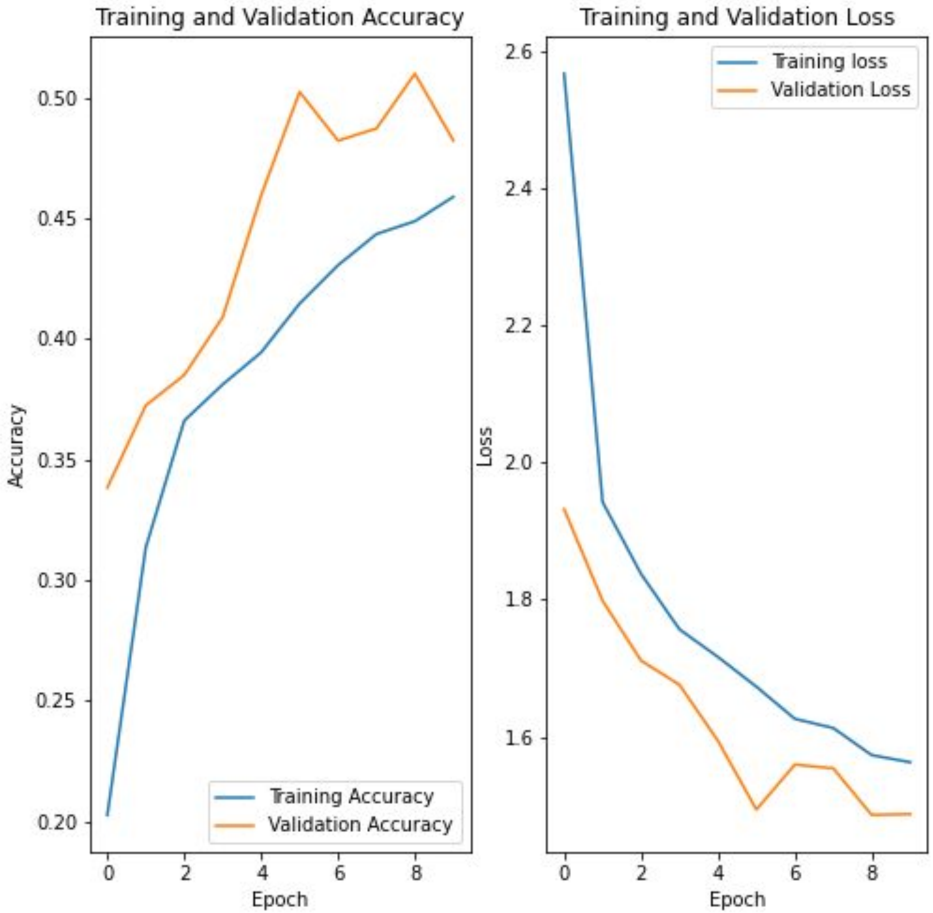
Model Construction - cont’d

- Building, and training, a CNN from scratch on this dataset would be too time consuming and resource intensive.
- As such, we will be making use of transfer learning, training 5 baseline models as a benchmark prior to model selection

Baseline 1 - VGG16

The first baseline model we used for transfer learning was VGG16, a CNN that was proposed in 2013 and submitted to the ImageNet Challenge in 2014.

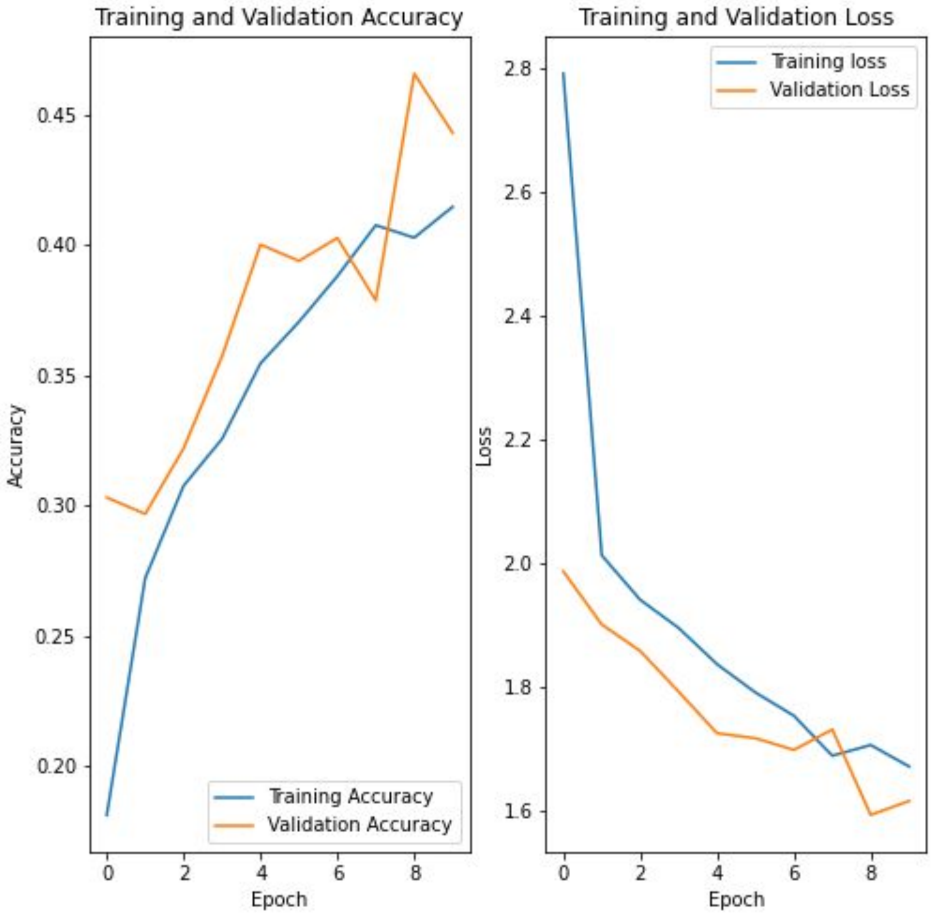
Validation Accuracy: 48.23%



Baseline 2 - VGG19

The second baseline model trained was VGG19, which is a variant of VGG16, but with nineteen layers rather than sixteen.

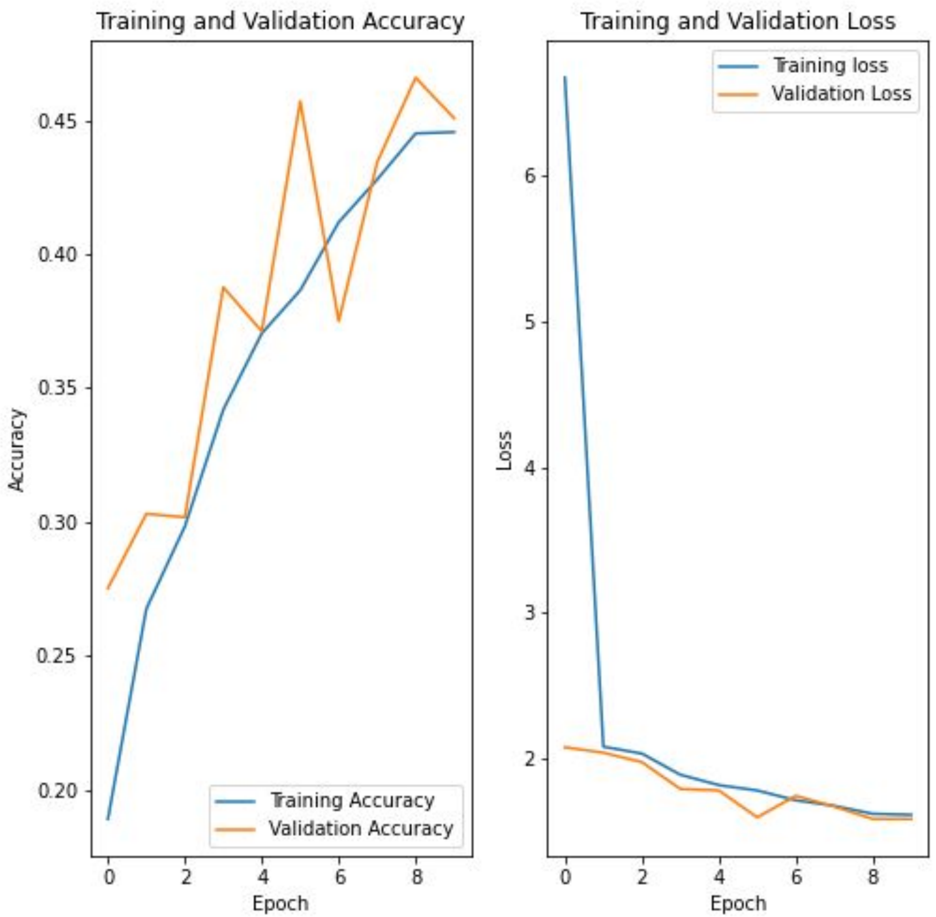
Validation accuracy: 44.32%



Baseline 3 - Inception

The third model we trained was Inception, another CNN posed to the ImageNet Challenge.

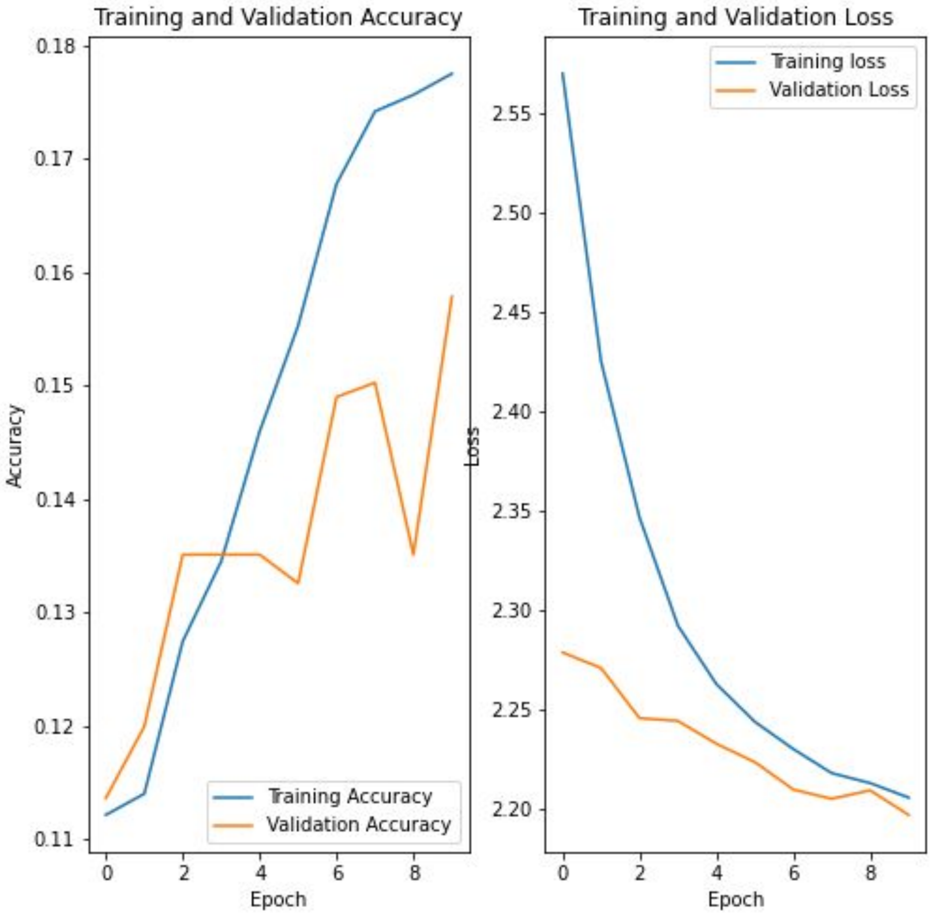
Validation Accuracy: 45.08%



Baseline 4 - ResNet50

The final baseline model that we constructed was ResNet50, a CNN that was proposed in the ImageNet Challenge in 2015.

Validation Accuracy: 15.78%



Evaluation

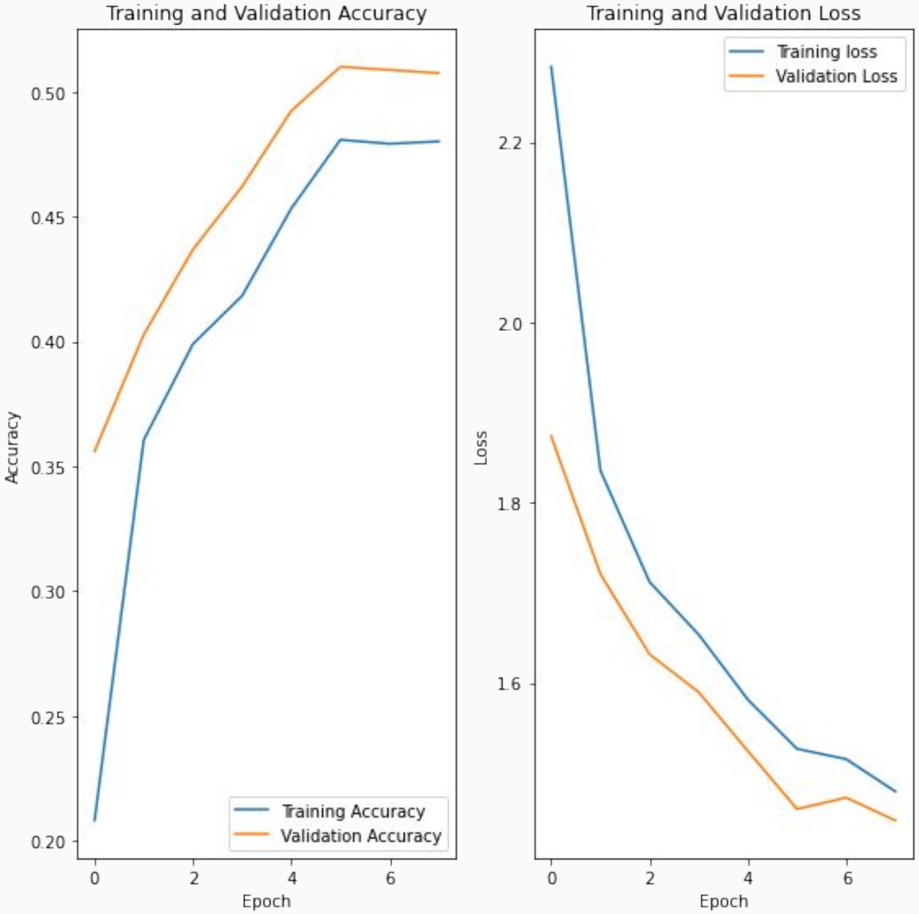
Baseline Model	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
VGG16	0.459	1.563	0.4823	1.4874
VGG19	0.4147	1.6722	0.4432	1.6166
Inception	0.4456	1.6192	0.4508	1.5908
Inception (dropout = 0.25)	0.4565	1.6271	0.423	1.6598
ResNet50	0.1775	2.2057	0.1578	2.197

Model Selection

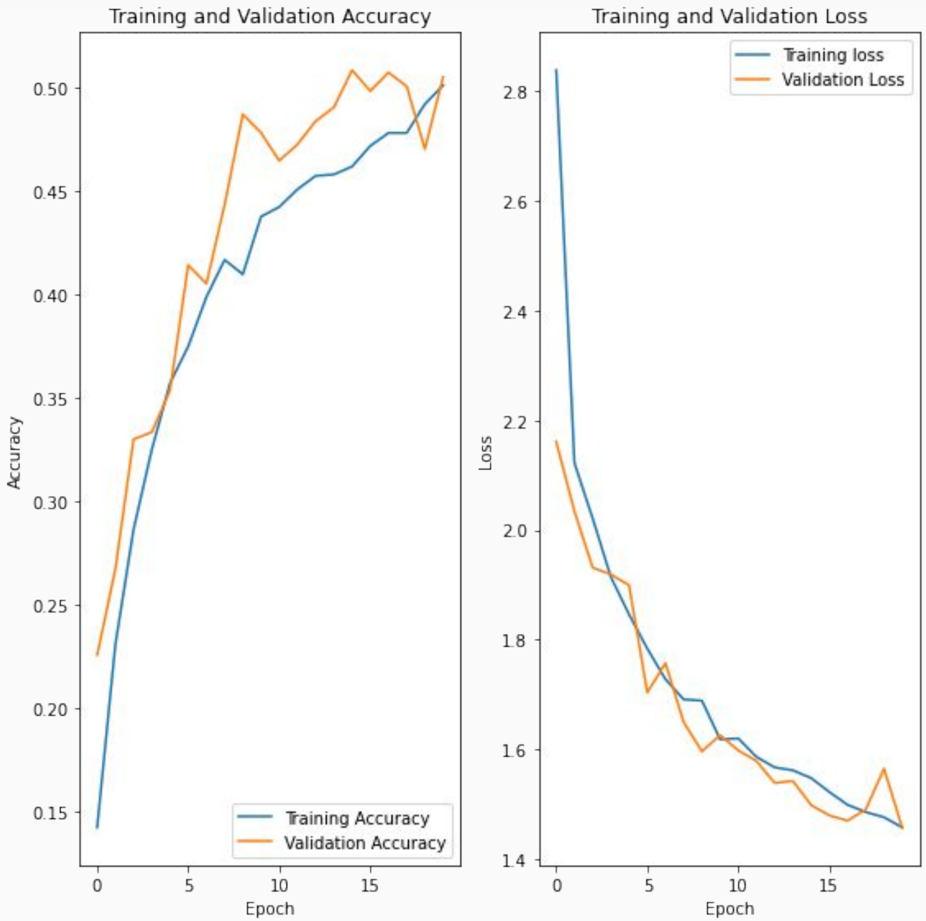
When training these baseline models, we noticed that there were still relatively low accuracy scores while error rates (loss functions) decayed very slowly in later epochs.

That being stated, the architectures selected were VGG16 and Inception, given that they had the highest accuracy scores and lowest loss values

VGG-16


Training and Validation Accuracy
Accuracy
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0
2
4
6
Epoch
Training Accuracy
Validation Accuracy
Training and Validation Loss
Loss
2.2
2.0
1.8
1.6
0
2
4
6
Epoch
Training loss
Validation Loss

Inception



Training and Validation Accuracy
Accuracy
Epoch
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0
5
10
15
Training Accuracy
Validation Accuracy
Training and Validation Loss
Loss
Epoch
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
0
5
10
15
Training loss
Validation Loss

Test Cases

Key

We defined a function that takes an image file path and an artist name as an input, and outputs the following 1x4 array:

(Prediction Index, Prediction Likelihood, Artist Index, Artist Likelihood)

<u>Legend for Outputs</u>

0: Cezanne,	1: Degas,	2: Gauguin,	3: Hassam
4: Matisse,	5: Monet,	6: Pisarro,	7: Renoir
8: Sargent,	9: van Gogh		

Test Case 1

VGG16 Output:
(0, 0.348270485, 9, 0.10976794)

Inception Output:
(7, 0.24960655, 9, 0.036583018)

Test Case 2

VGG16 Output:
(5, 0.27732363, 5, 0.27732363)

Inception Output:
(5, 0.39511997, 5, 0.39511997)

Test Case 3

VGG16 Output:
(5, 0.34927684, 9, 0.09115124)

Inception Output:
(2, 0.25939465, 9, 0.04012476)

Room for Improvement

- Develop a web scraper to scrape painter images to increase the size of the dataset without data augmentation
- Train other baseline CNN architectures such as Xception, ResNeXt, DenseNet, etc.
- Experiment with a wider range of learning rates and CNN architectures, utilizing Keras Tuner, RandomSearch, or GridSearch

Any Questions?


Thank You!