**Contributing members**

Anh Tran

Rodrigo Mouron

Kaveon Ware

**Abstract**

Our goal is given a painting, examine the painting and predict the artist that painted it.

**Background**

With paintings carrying millions of dollar price tags, identifying forgeries from authentic works has become valuable. Our model has been built in mind of identifying the artists that created the work. Although our program does not directly identify a forgery, it can be used in hopes for further research in identifying whether the painting was an original or fake

**Error Handling**

We began project by first downloading our data. This was when we ran into our first issue. The paintings by artist full dataset is 90gb in size. We realized that since we only had about a week and a half to complete this assignment we would not be able to go off of the full dataset. We decided to reach out to our professor who instructed us to use a smaller portion of the dataset, that’s how we ended up choosing a 5gb file with about 12,000 photos instead of 135,000.

Fixing data:

Once we downloaded the 5gb file we realized that our data frame containing all the names of the artists had images which were not part of the folder of images we had. To fix the data, we decided to merge our data frame with a data frame we created using the folder of images, this way we could drop all the rows containing pictures which we could not open. Our model now had the names of the images with the names of the artists and the correct images in the folder. This dropped our dataframe from 135,000 rows to the 12,000 from the folder, after dropping any NA values, we are left with around 9,000 pictures.

**Model Creation**

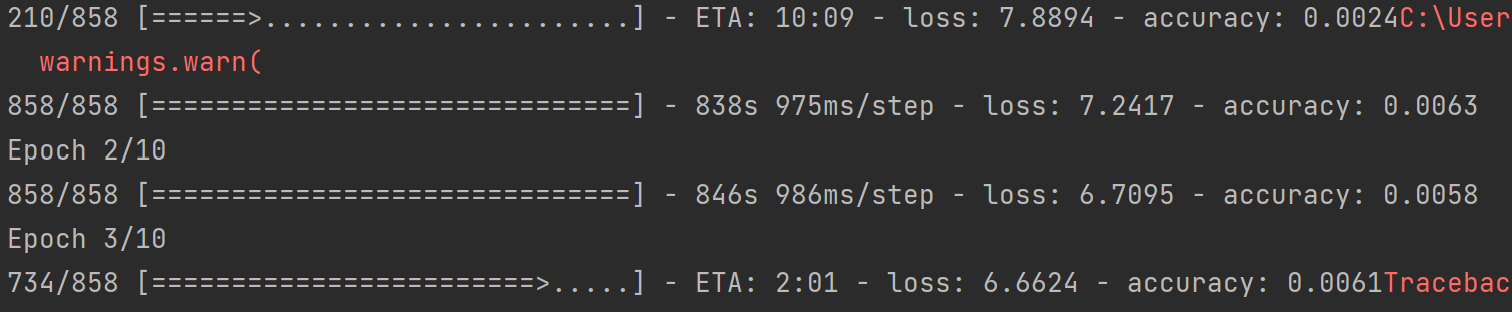
We started by testing a basic supervised CNN multi-class classification model. We had four layers that included a convolution layer with incremental filters, a pooling layer with pool size of (2,2) and a dropout layer. Afterwards we have dense layers to reduce the guessing classes. For multiclass classification we used “relu” and “softmax” for activation. We chose the standard “adam” optimizer and “catergorical\_crossentropy” loss function.

Bettering the model:

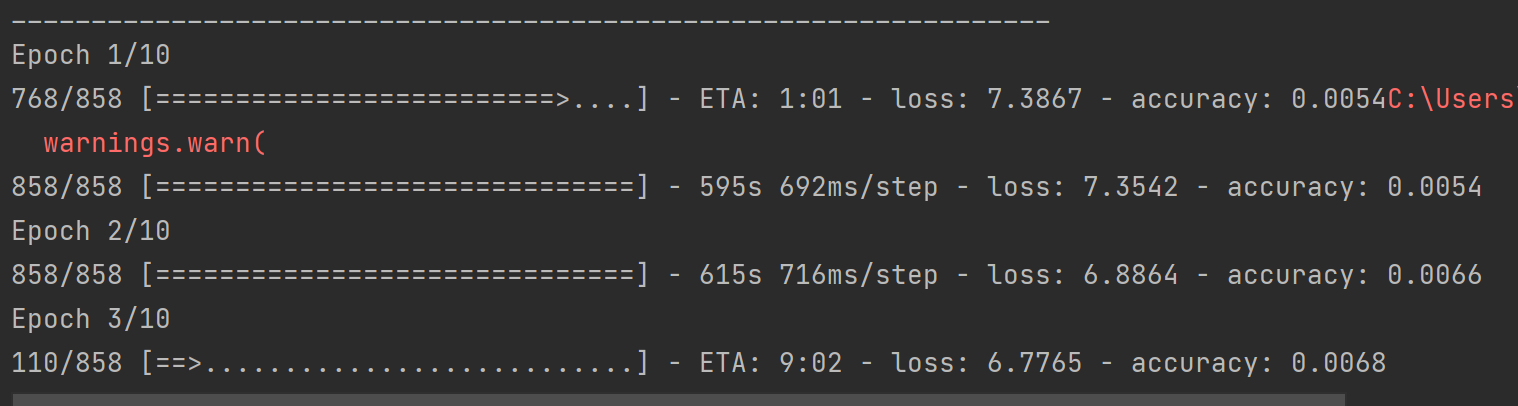
We began by dropping layers to simplify our model and cleaned our data as much as we could. Cleaning data is what really made our model improve, we spent the most amount of time cleaning data and learned many new ways to interpret information. Increasing epochs was another way we got more accuracy, doing this along with dropping layers was a good improvement. We tried using multiple different models, we ended up choosing a model that improved our accuracy by 14 fold.

**Results**

We created multiple models for this project, originally, our model was very bad, achieving around a .75% accuracy. This was obviously very bad, our model was not able to correlate the images with the names and it simply guessed wrong most of the time. This is where we realized our program might need more time to complete. Since these epochs were taking 10 minutes each we needed an hour and 40 minutes every time we wanted to change our model.

****

Here’s a picture showing some more improvements we made, we decided to simplify our layers and fix our data more. We dropped many rows of artists who didn’t have many paintings, this way our model was learning from artists with multiple paintings and would have more data to learn from.



The main way we improved our model was by fixing our data, giving the model better data is what made it better, this is the reason we believe with more resources we would be able to give our model even better data that would result in higher accuracy.

**Uses**

Our model can be used to predict the name of the artist who’s picture is imputed. Since we were rushed for time we did not have time to improve our application but we would be able to use our code in order to make a full app. If a simple front end was made we could link our code to that and make a UI for the user to check their pictures to find the artist who made it.

**Improvements**

Our major issue with this project was the time we had to complete it. We had the time to complete a model that works well for the data we were given. If we had a computer that could run our models faster and have more storage than the computers we had we could make our model much more accurate by using much more training data. Some of the artists in our dataset did not have enough pictures to train on. We had too many artists with less than 10 pictures, we need more information on each of the artists to be able to predict more accurately. With more storage, we could download the whole dataset. Faster GPU’s would also make our model fit faster, our model took around an hour and a half to run, if we could have run the whole dataset within that hour and a half we could have a much better model.

**Conclusion**

Our project was successful within the scope we were given. If given more time we would improve our model by using more powerful computers that can run our program much faster. We need more storage to have more data. The more data we have the better our model will learn. A big take-away from this project is just how important cleaning our code was, this improved our model the most by far.