**Project 5 Script**

[**1. Introduction - Daniel 1**](#_kutkdfnfk94a)

[**2. Problem Statement - Daniel 2**](#_gus9xtx3krk5)

[**3. Methodology - Daniel 3**](#_9ijf7pwigvei)

[**4. How we got the data - Steve 4**](#_rdoby5xlbnty)

[**5. How we cleaned the data - Steve 5**](#_lvf8ak3t1qkv)

[**6. EDA (focus on specific graphs split up between eda boys) - Steve & Nick 5**](#_dq3uuqb3ab9p)

[**7. Preprocessing - Nick 6**](#_tvzab0hyyisi)

[**8. Models - Kate 6**](#_t1ozq6xq9j90)

[**9. Recommendations - Kate 6**](#_hsgfhwr6e4t1)

[**10. Next Steps - Kate 7**](#_lkfwsuhafy54)

[**11. Conclusion - Kate 7**](#_rv1ud85warhm)

[**12. Streamlit App - Nick 8**](#_vh9twy5hf0wo)

[**13. Story Epilogue - Daniel 8**](#_87r1ronox7pi)

PPT: <https://docs.google.com/presentation/d/1ZZExe9foEcnY-pDvzcdH5KFtSPEBFxBCnruPfakgGSo/>

# Introduction - Daniel

(Title slide)

Good morning everyone, and welcome to DSI Group 2’s presentation on neural networks. By Daniel, Nick, Steve, and Kate.

(Slide - Table of Contents)

A quick overview of the structure of our presentation. We’re starting with an introduction and backstory to our problem statement. After that, we’ll cover some of the terms a lay person might need to know to better understand the presentation.

Then Steve will dive into the data, talking about the dataset; and our data cleaning process.

Then Steve & Nick will go over some of the interesting things they found during EDA, before Nick talks about our modeling Process.

Then Kate will talk us through the models we tried, their performance; the final model, and some conclusions, recommendations, and next steps. And we’ll finish with some audience interaction.

(Slide - Can a painting’s art style…)

So- Nowadays, it seems like neural networks are this magic bullet tool that can do ANYTHING. …But can they tell us why kids like Cinnamon Toast Crunch, or how many licks to the center of a Tootsie Roll Pop? NO- of course not! …but maybe they CAN help us predict the style of a piece of art based on its pixel data.

Art has ALWAYS been an important medium of human expression and creativity. Through the centuries, a wide variety of art styles have emerged, each with its unique characteristics, techniques, and themes. But there really ARE a lot of styles and substyles, and if you wanna get REALLY nitpicky- they seriously CAN be rather hard to tell apart sometimes, with or without an art degree. Suffice to say- that determining the style of a painting can often be a challenging task, especially for those unfamiliar with art history and criticism.

That's where CNNs come in. CNNs, or Convolutional Neural Networks, are a type of deep learning algorithm that have shown great promise in the field of image recognition. In this presentation, we’ll discuss the basics of CNNs, the process of training a CNN model for art style recognition, and the potential applications of this technology in the real …(-ish) world- in the form of a totally plausible scenario problem statement.

\*Zoom filter on\*

# Problem Statement - Daniel

(Slide - Story Introduction / Problem Statement)

*Daniel*: So, here’s the deal… We’re a group of art thieves that struck the motherlode and pulled the heist to end all heists, scoring a haul of thousands of (hopefully) famous paintings and works of art. The only problem is that things got a bit messy during the getaway, and the boss (our resident art expert) caught a bullet in a firefight. Now, the rest of our crew has shipping containers full of artwork, with no idea what we have or how to fence and unload any of the merchandise. We’ll have no choice but to list our items on the black market eBay… ShadEbay, they call it. But who’s gonna buy any of our stuff if we can’t even properly list out what we have. We’d definitely need to at least know what style of art everything is right?

Could we just go out and try to find ourselves another art expert? Maybe… but with word of our heist all over the news, who can we trust? Any so-called “expert" showing up at our door could really be an undercover agent in a sting. No- it’s too risky. There’s gotta be another way.

*Kate*: But say… what if we were to build a machine learning model using neural networks to help us discern what style of art each piece is?

*Daniel*: BRILLIANT! You’re a genius! Yea- that’s a MUCH better idea. Now… Let’s… get down to business! (To defeat the Huns)

# Methodology - Daniel

*(Thought: i think the methodology section might actually make more sense after the how we got the data/cleaned it; right before talking about EDA)*

(Slide - Neural Net Vocabulary)

Neural nets are a computational method using nodes (or “neurons”) linked together in layers and providing input and feedback to each other to perform calculations and give predictions.

*(CNN)* - CONVOLUTIONAL neural networks (or CNNs for short) are a type of this deep learning algorithm used primarily for image and video recognition. They use a series of convolutional “layers” to detect features in the input data and learn to classify it based on those features.

Pixel arrays of RGB data - The process starts by breaking down images into pixel arrays of red, green, and blue information that can be processed separately.

Convolution filters - This is presumably where CNN’s get their name from. Convolution filters are matrices used in calculations. The “kernel” is another name for the “filter”.

During the convolution operation, the kernel moves over the input image, computing dot products. They’re especially useful for edge detection in images.

*(Kernel size)*  Typical kernel sizes in CNNs are 3x3, 5x5, and 7x7. In general, smaller kernel sizes are better at capturing fine-grained details in the input image, while larger kernel sizes are better at capturing broader features.

Batch Normalization is a technique for improving the performance and stability of CNNs during training, by normalizing the outputs of a layer for each mini-batch. It basically scales and shifts the data so that the mean of the outputs is close to zero and the standard deviation is close to one.

Regularization is usually a means to decrease variance and prevent overfitting. While regularization is USUALLY a good idea in most machine learning, it led to strange results in our testing; and is apparently somewhat discouraged for use in CNN’s like ours.

Dropout Layers. Dropout involves not using all the nodes during computation, and randomly dropping some out of the network to prevent any input from becoming too influential in our model and leading to too much variance.

A Dense Layer takes the output of the previous layer and flattens it into a one-dimensional vector; and can be used to help the network learn more complex relationships between the input and output. And so they’re super useful for tasks like image classification or object detection.

# How we got the data - Steve

(Data Dictionary slide)

After an exhaustive search we eventually decided on using a dataset that we found on kaggle.

The dataset consisted of artwork pulled from the wikiart database. Instead of containing the images themselves, it contained LINKS to the images on the wikiart servers, so we had to scrape the images ourselves manually. We then scaled the images to 250 by 250 pixels, so that we could feed them in our model.

We knew right off the bat that we were interested in the STYLE of the art pieces; but the kaggle dataset also contained other fields of data that weren’t actually used in image prediction but produced some interesting findings during EDA.

The original dataset contained approximately 120,000 pieces of art, but after a bit of cleaning and filtering, we were left with just under 90,000 works on which to train our model. Also, to note ,the variables in the yellow are from the original data set and the ones in blue were engineered.

— Next slide

This graph is just a visualization of the art styles and how many pieces of art came from each style. Impressionism having the highest count with constructivism having the lowest.

# How we cleaned the data - Steve

So for the cleaning we first got rid of the bad urls for the images. We then dropped the duplicate images, such as the initial sketches for the final works. After that we decided to drop styles that had less than 500 pieces. Next we went on to the date column..

Going through the dataset we realized that the date column had strings depicting what century the art came from and also had ranges for the dates.To solve these problems, I made a hyphen remover function that removed hyphens from ranges and kept the first number from that range. After that we had to hard code the roman numerals to the proper dates and converted all dates to numeric.

(Next slide)

After all that was done we went on to feature engineering…

Translated column = Artwork titles that were in a different language, translated to english

V-sent column = Sentiment analysis on the title of the artwork

Style\_num = Assigned number for each style of artwork

Color = Dominant color in each of the pieces

Language column = showed what the original language of the title was from

Hex = is just the color code for each dominant color

The graph on the right is a visualization of the top 10 dominant colors that were in a our dataset with black showing up the most.

# EDA (focus on specific graphs split up between eda boys) - Steve & Nick

This beautiful graph is a scatter plot of the styles by date with the coloring based on the sentiment analysis for each piece of art. As you can see, Ink and wash paintings have been used through out all the centuries

Nick—--

Tableau dashboard that shows cool stuff

1430's is when black became the most dominant color for all artworks.

From 1740 to 1778s, Giovanni Battista Piranesi released around 1200 etchings which surfaced and gained tremendous market value after his death in 1778.

Rogier van der Weyden or Roger de la Pasture was an early **Netherlandish** painter

# Preprocessing - Nick

For preprocessing, we ran into numerous problems. Since our dataset was so large, we couldn’t prepare it normally. We tried numerous approaches such as creating arrays of the images, trying filepaths to images, and even trying to create a dataframe of matrices to try and feed our data into a model.

Solution:

**Tensorflow Keras (image\_dataset\_from\_directory) function**

This function was huge and essentially saved us. It’s faster, easier, and higher quality. image\_dataset\_from\_directory is critical in our preprocessing. (Thanks, Tim Book.) This method creates a BatchDataset from a given directory and classifies them based on their folder location. We resized the images to 250x250 pixel. And since we are using a classification model, the labels are categorical(styles).

# Models - Kate

* We fiddled with Optimizers (‘adam’, rmsprop)(adam good), ran a bunch of even worse models before we realized we needed to Rescale our data, tried multiple filter layers ( 3-5)(more good), kernel sizes (2-9) (big good), varying layers of dropouts and sizes(small but numerous good), BatchNormalization(good kind of), Regularization (bad bad no good) , Extra Dense Layer (kinda good)
* Final Model: we achieved 450% over baseline accuracy, however we can see that’s still not good. We are only predicting correctly not even 1/5th of the time. Its clear that we were making progress but evident there needs to be further development of this model before it can start helping us sell these paintings on ShadyBay ~~ebay~~.

# Recommendations - Kate

story-accurate recommendations:

1) have some idea what you're stealing before you pull the heist

2) don't steal 10's of thousands of paintings at once- it really does just make everything more complicated

3) Making a CNN machine learning model to predict is too hard. Just kidnap an art expert next time

4) Get a second opinion on what style some pieces are. Even within the same genre, there are sometimes such a wide variation between works (to the untrained eye), that it’s hard to believe they really all are the same style.

5) Don’t have so many different possible prediction classes. Definitely try to combine some of the similar categories.

# Next Steps - Kate

* - More time : Given another week to experiment with the following steps I strongly feel we could've gotten better results.
* - More Processing power : this was a severe limitation on our project, as many models often had to run overnight on only 5 epochs.
* - Combine like art styles (High/Early/Late Renaissance) : 51 classes is a tall order
  + Time permitting: Let computer do unstructured learning to determine which styles are similar
* - Higher resolution images : Could've possibly improved model accuracy, but would also increase amount of time needed to run
* - Using only most populated classes : less classes and plenty of data to use, however this reduces the overall size of the data drastically
* - Using models other than CNN : we put most of our focus into making this type of neural network work, but other types such as Deep Neural Networks or utilizing Transfer Learning could have improved our prediction ability
* Use other metadata of the artworks to make a combined prediction?
* Recommender for similar artworks?

# Conclusion - Kate

* Here’s two images, can you tell me which of these two paintings is Expressionism and Romanticism?
* These are neither, they are Baroque and High Renaissance
* Now imagine for a moment you’ve never even seen or heard of art, even as a concept before, and are told to try and predict after looking at almost 100000 images. Could you do it confidently? Successfully? Be correct even 1/5th of the time?
* Our random guessing baseline, of even our majority case, was only .04. We were able to increase that score by 450%. While this would be considered a failed model in many cases, this is telling a technical machine to judge where a subjective concept like art is in one category or another. And it can! To an extent.
* Our model is a failed model. It predicts incorrectly 4/5ths of the time. That is not a case for a successful model. Our baseline was 0.0397 and although we achieved our metric of success, 0.178, it is nowhere near production ready and needs more time to develop. We've learned quite a bit from this experience and realize that predicting something as subjective as art is a difficult task, especially with 51 labels. I hope that you learn from our mistakes and find the same growth we have in this process.

# Streamlit App - Nick

Here is a streamlit application that can run and predict our model. This application will help us art thieves easily take photos on our iphones and quickly gives us a prediction on the style. I’m going to show you what OUR model says these images are.

\*shows model\*

Now, would anyone in the audience like to submit an image to our art thief style classifier? Please just search on google, and send me the link in chat. I’ll download the image and we’ll see which style our model guesses!

# ~~Story Epilogue - Daniel~~

~~asdfasfd~~