Painter By Numbers

Azmi Haider

Yael Nagler

Introduction

Kaggle challenge: https://www.kaggle.com/c/painter-by-numbers

Goal: building a network that learns artists' painting style.

Q:Which paintings were painted by the same artist?

A: left and middle are of the same artist, right is of different artist

Network

Given two painting, the network aims to measure distance (similarity) between them:

- Small distance = paintings share same style (same artist)
- Large distance = different styles (different artists)

- Our network aims to build an organized feature space
- Regulating of the feature space means ensuring that paintings of the same artist are clustered together. Paintings which share an "artist style" are closer to each other than to any different artist's paintings (closer = the Euclidean distance between their feature vector representation is smaller).

Why Siamese CNN

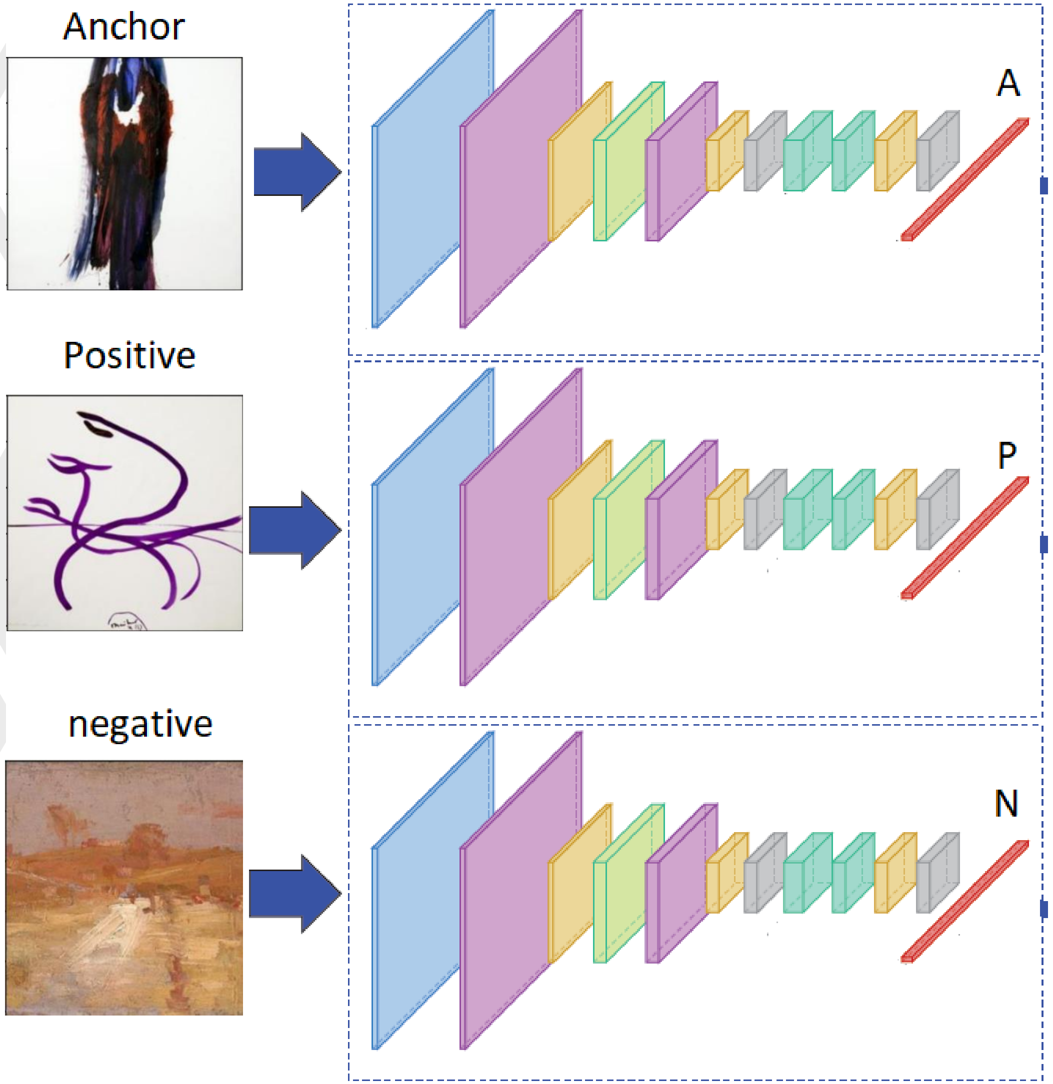
- The data used is images. Therefore, the best fit is a **CNN**.
- The network is not a classification network since the test set consists of artists not seen in the training. It should be built to produce a similarity score on a pairs/triplets of images. For this purpose, the best choice is to use a **Siamese network** which maps inputs (paintings) into a feature space and is updated to organize the feature space as describes before.

Siamese CNN with triplet loss

- **Three** Convolutional Neural Networks where **weights are shared**.

The first image -we term Anchor- is a painting which belongs to artist1.
The second image -we term Positive- is a **different** painting of the **same** artist1.
The third image -we term Negative- is a painting of a **different** artist2.

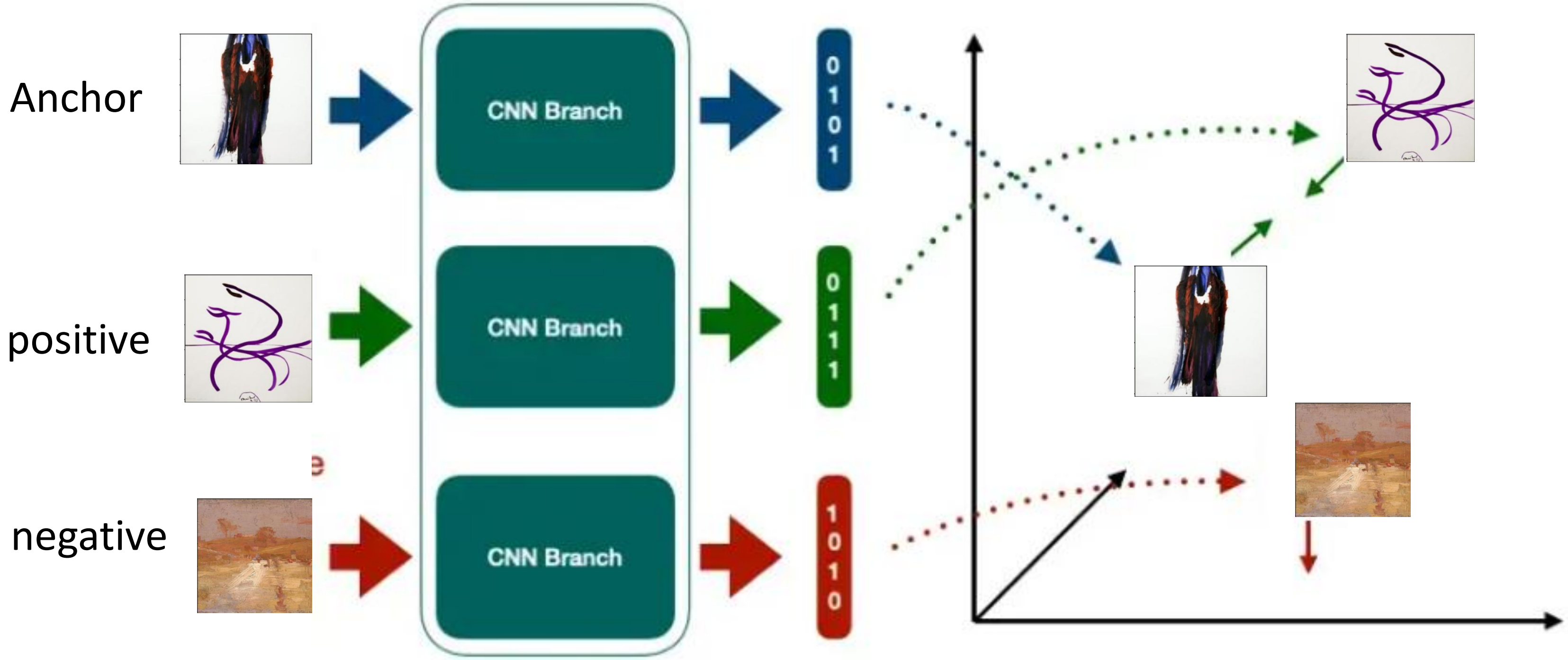
Note: The two artists and paintings in each step are all randomly chosen.

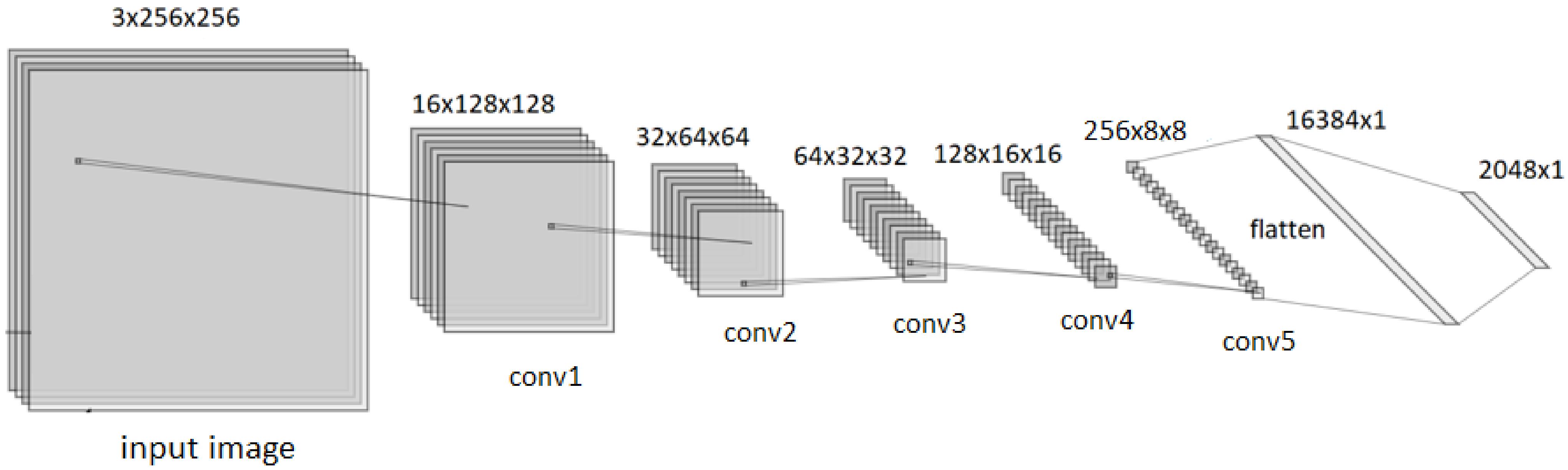


Loss functions: Triplet loss

- Network: 3 CNNs
- 3 images are processed by the network at each step: 2 of the same artist and one of an imposter.

$$L = max(0, D(A,P) - D(A,N) + margin)$$



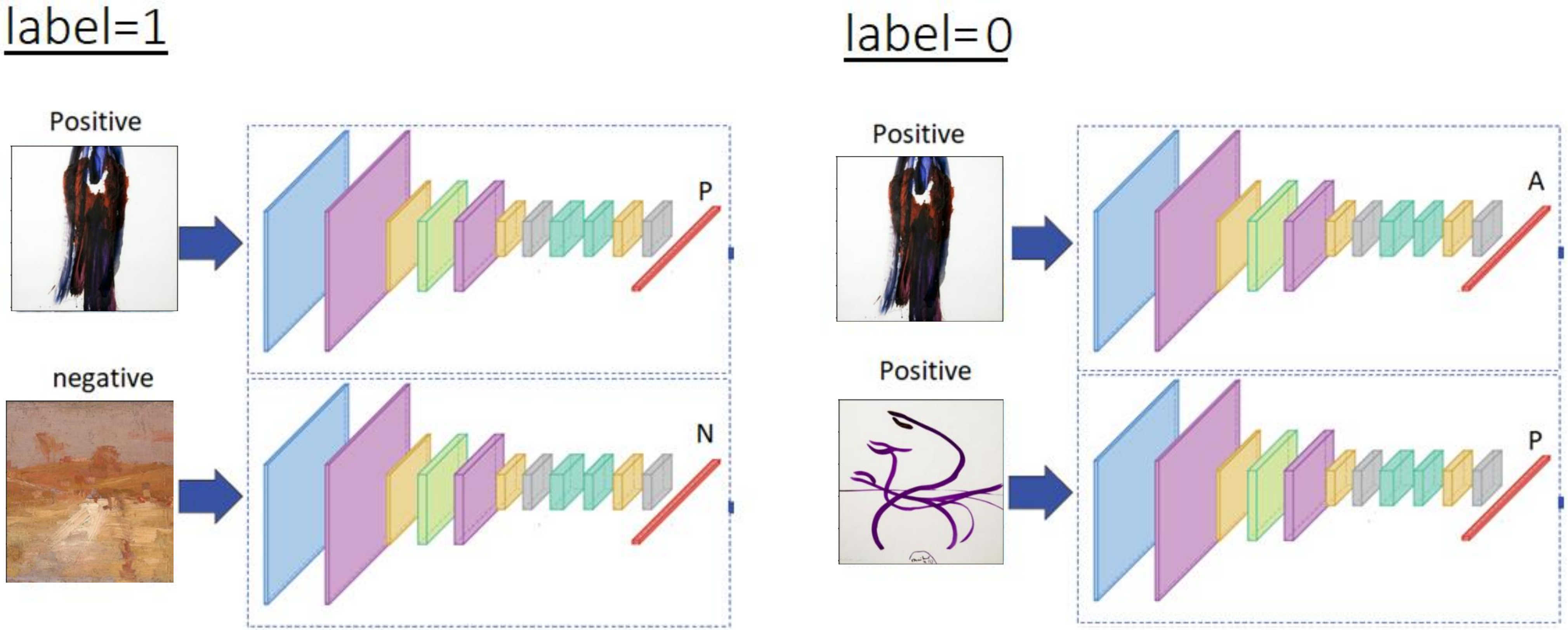


CNN architecture

- CNN consists of 5 blocks; each has 2-3 conv layers followed by a Relu activation. Each doubles the number of filters and followed by maxpool2 layer.
- The final feature vector of a given image is a 2048x1 vector.

Two CNNs with contrastive loss

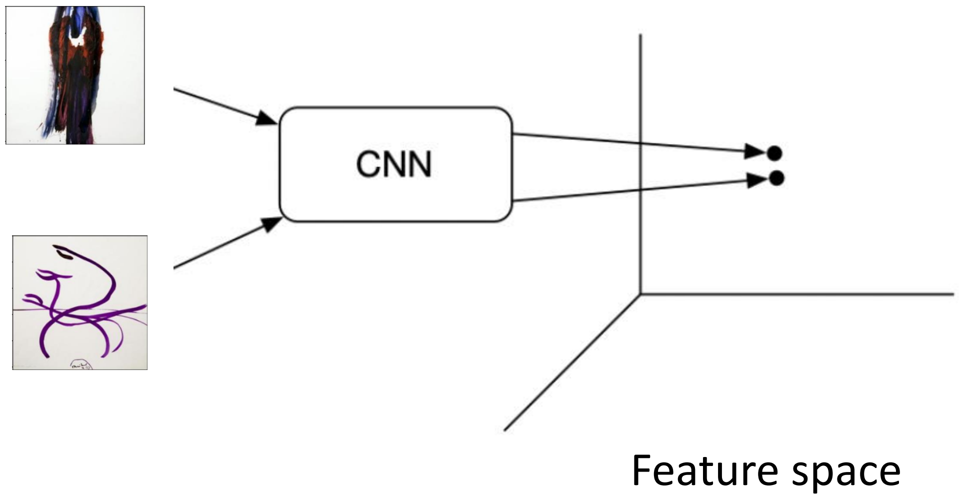
Note: we experimented with two CNNs and contrastive loss.

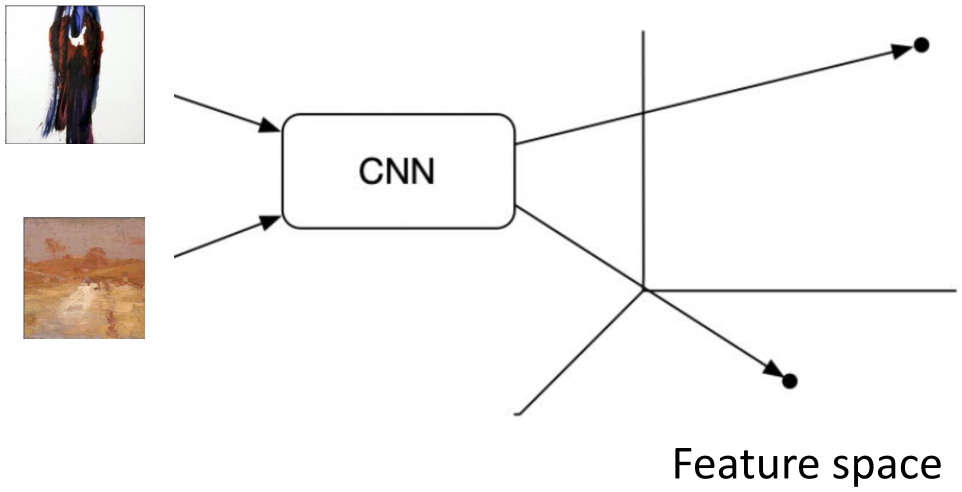


Loss functions: Contrastive loss

- Network: 2 CNNs
- Given a pair of images and a label (Y=0, same artist ; Y=1, different artist).

$$(1-Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{max(0, m - D_W)\}^2$$





Dw – distance in feature space

CNNs and loss

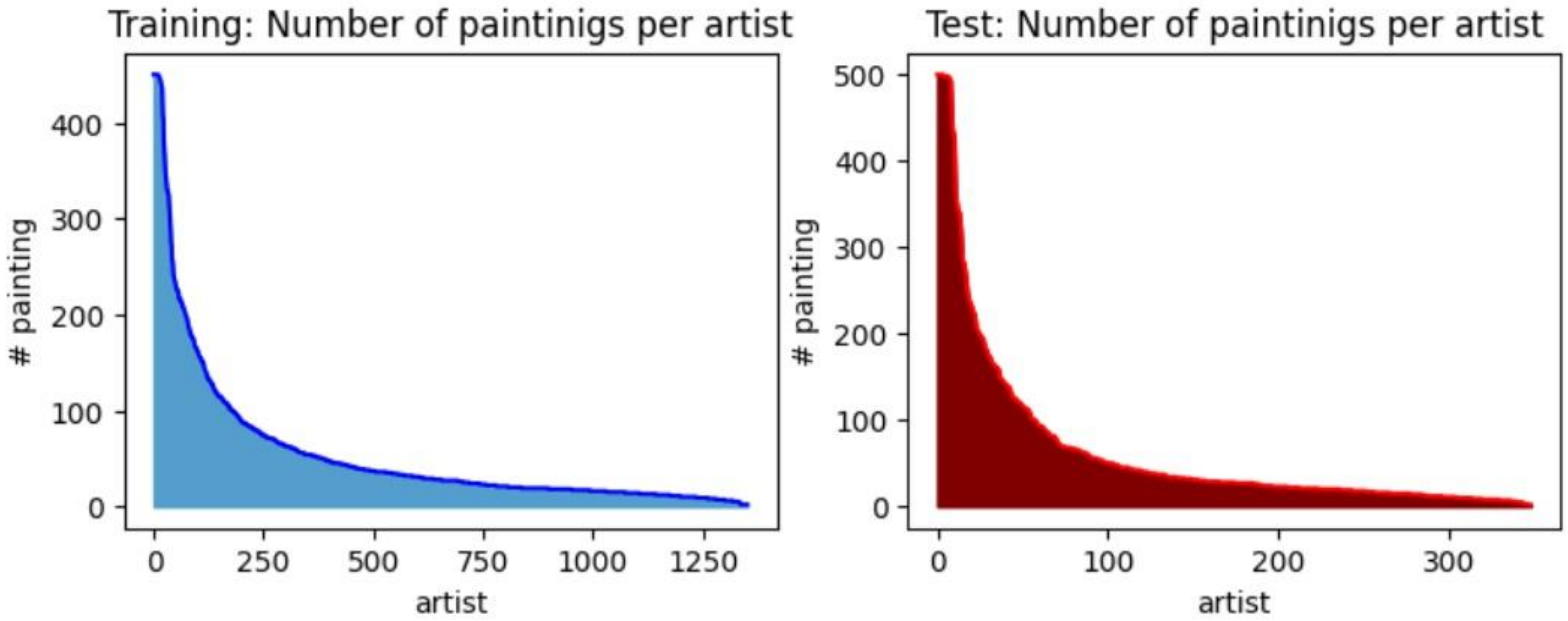
- We experimented with both approaches and losses.
- The code has a flag: ***pair_triplet*** (pair=True, triplet=False) which allows running both modes.
- We think the triplet approach is much stronger as it does what the contrastive approach does and more.
- In both approaches: artists and paintings are chosen randomly in each step.

Data preprocessing

- First, We've divided the paintings to classes based on their artists. We eliminated artists who had only one painting, this was done after noticing that some classes (artists) are not even artists (they had paintings' names) - it is clearly a mistake in data collection.
  After cleaning, the dataset has **1701 artists**, each with at least two paintings.
- Second, the dataset artists were divided to **training** (80% - 1352 painters) and **test** (20% - 349 painters).
- A fraction (10%) from training **paintings** was saved for **validation**.
- **The test set has only artists not seen in training (not one painting of theirs)!**
  The validation set has paintings not seen in training, but the artists who painted them has other paintings in the training.

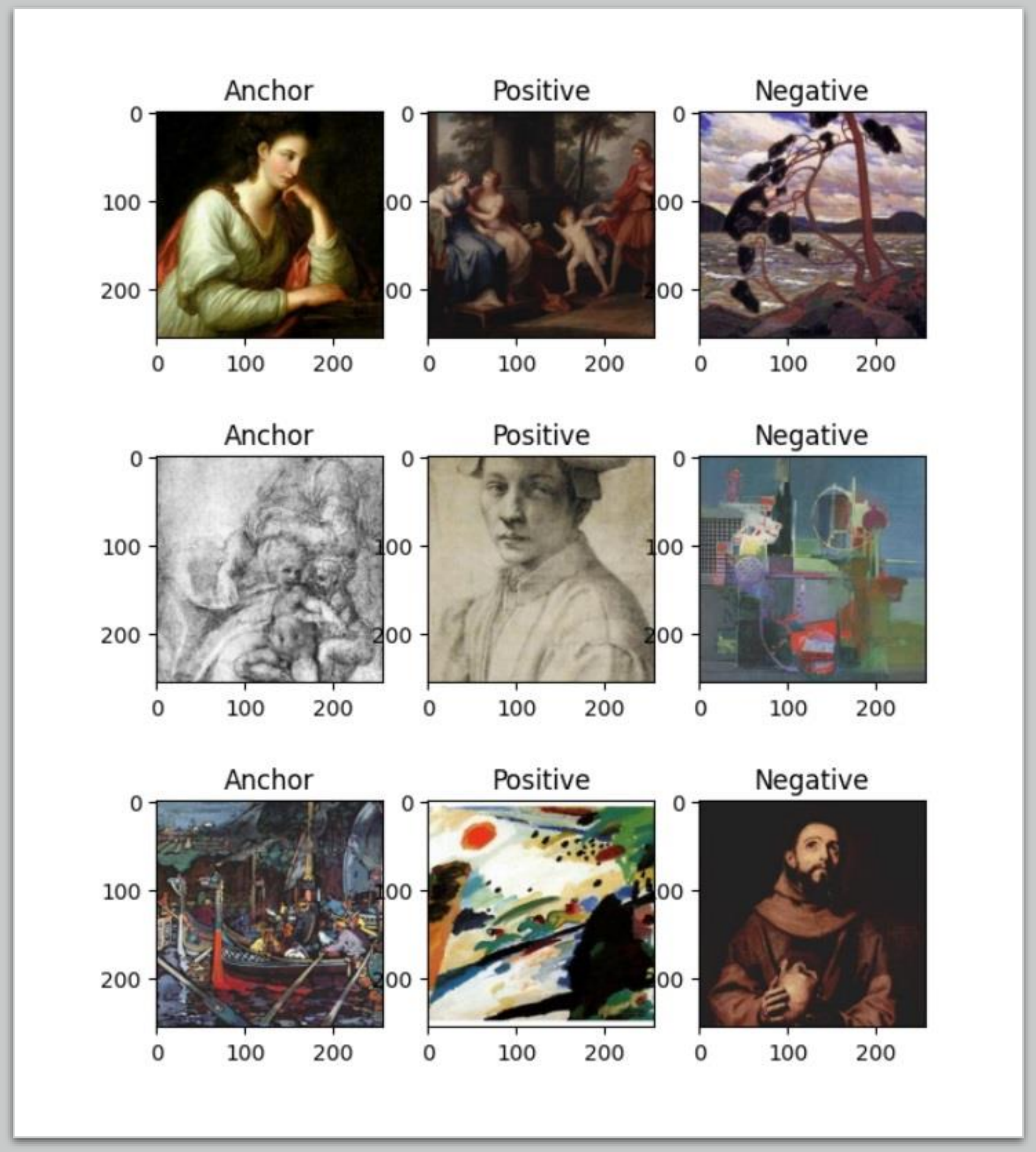
Data

- Image resizing and centering was performed on raw images to generate a 256x256x3 uniform size as an input for the network.
- Number of paintings for each artist is not uniform:



Number of images

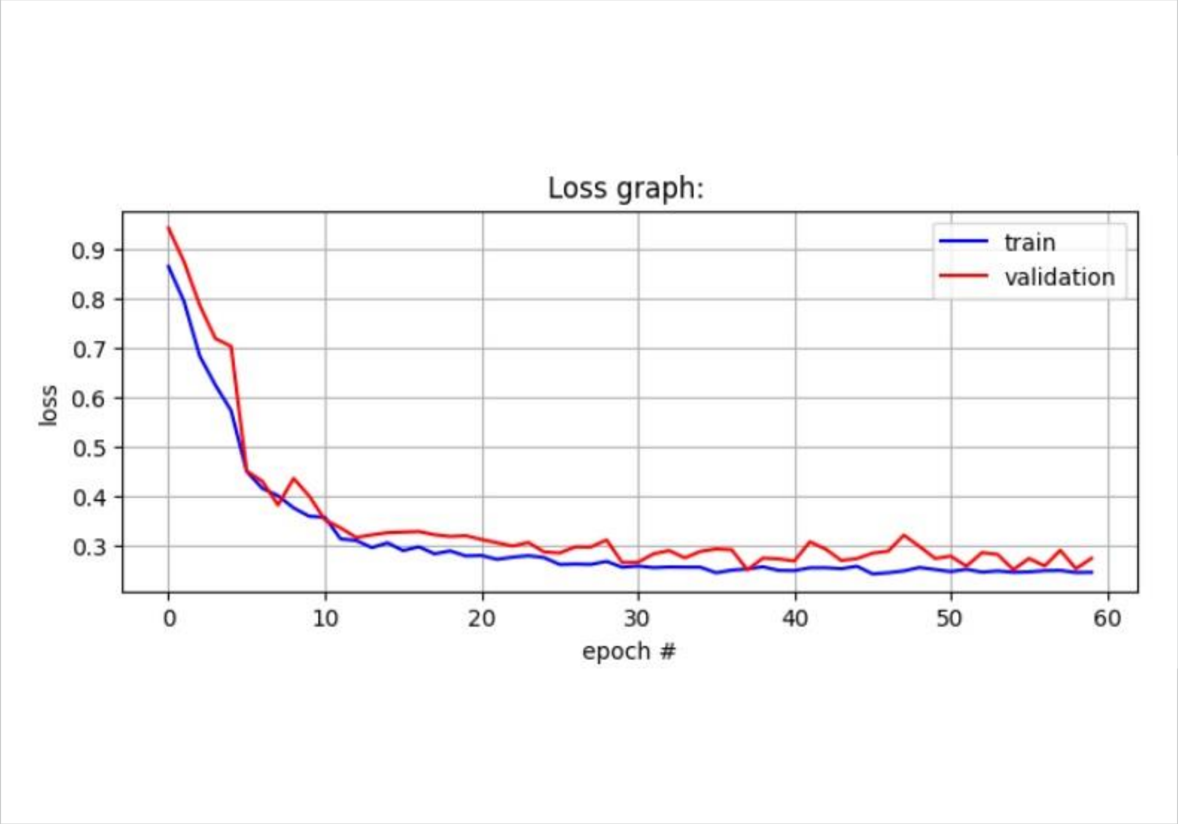
Number of paintings in training set: 73183
Number of paintings in validation set: 7083
Number of paintings in the test set: 22237

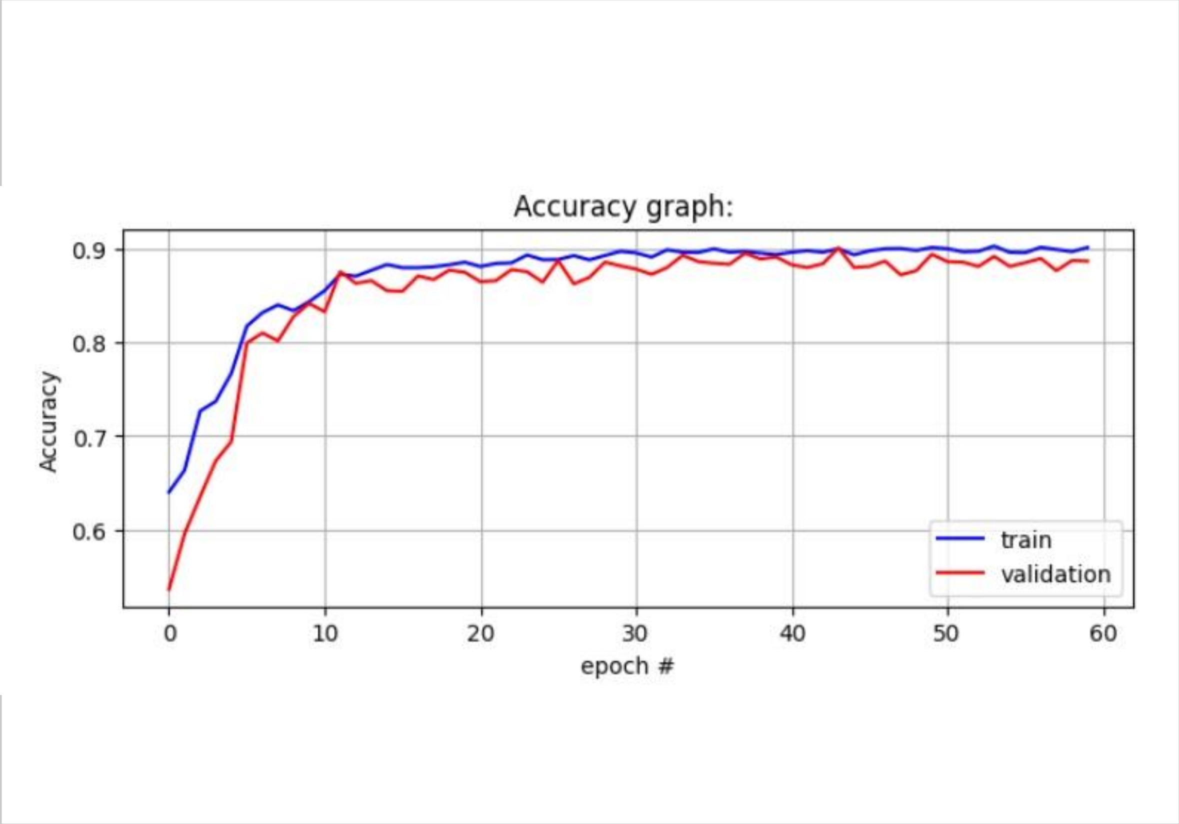


Training parameters

- Optimizer = Adam with a learning rate=0.001
  Epochs = 60
  miniBatch size = 4
  Scheduler = MultiStepLR(optimizer, milestones=[11,25,40], gamma=0.1)
- Random transformations used to augment:
  -RandomAffine (degrees=(0, 0), translate=(0.2, 0.2), scale=(1.2, 1.2))
  -RandomVerticalFlip (p=0.5)
  -RandomRotation (degrees=(0, 180))
  -RandomHorizontalFlip (p=0.5

Training results





Test

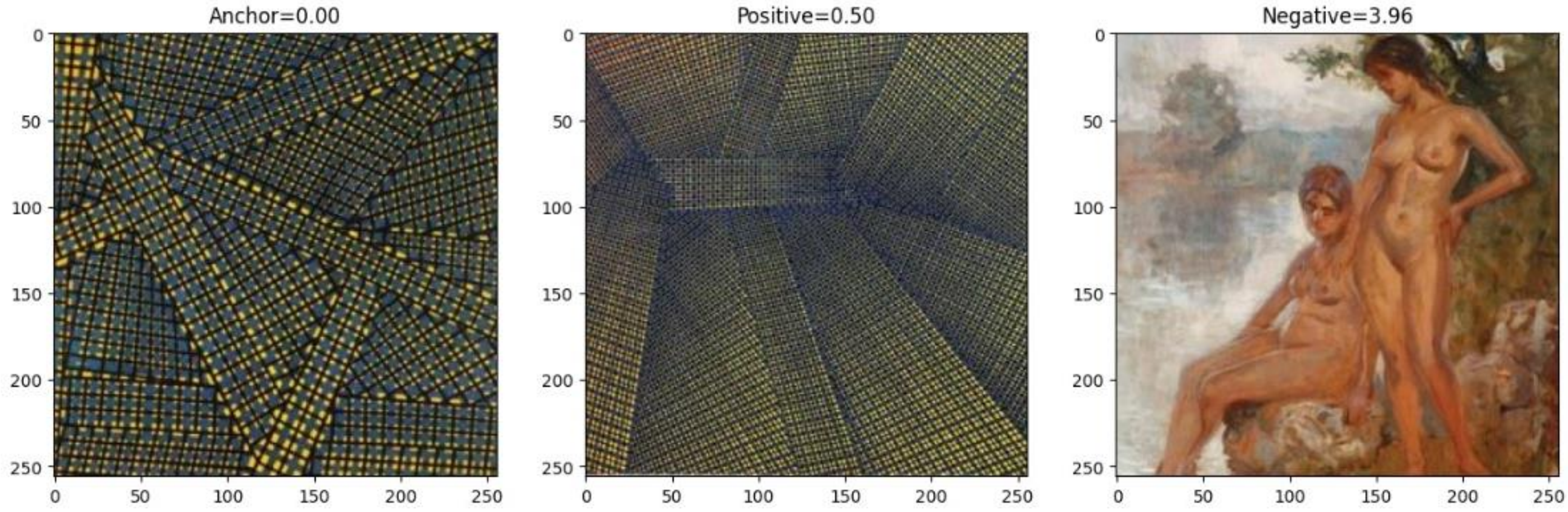
The same approach of calculating loss was used to measure test's accuracy:

Given 3 images (anchor, positive and negative), success was counted when distance between anchor and positive is smaller than distance between anchor and negative.

- **Reminder: the test set consists of paintings whose artists are not in the training.** the test set is a proof that the model **extrapolates** new painting styles!
- <u>Test accuracy - 0.8928057553956834</u>

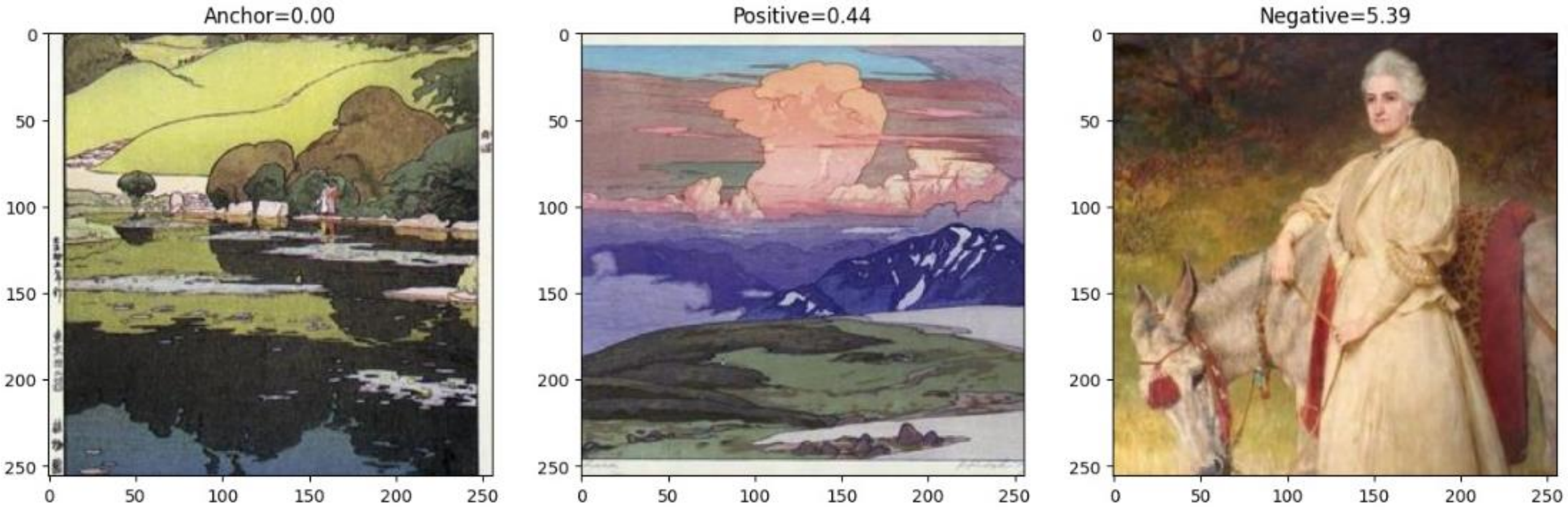
Visualization of test samples

Distance (between anchor and image) is in the title:



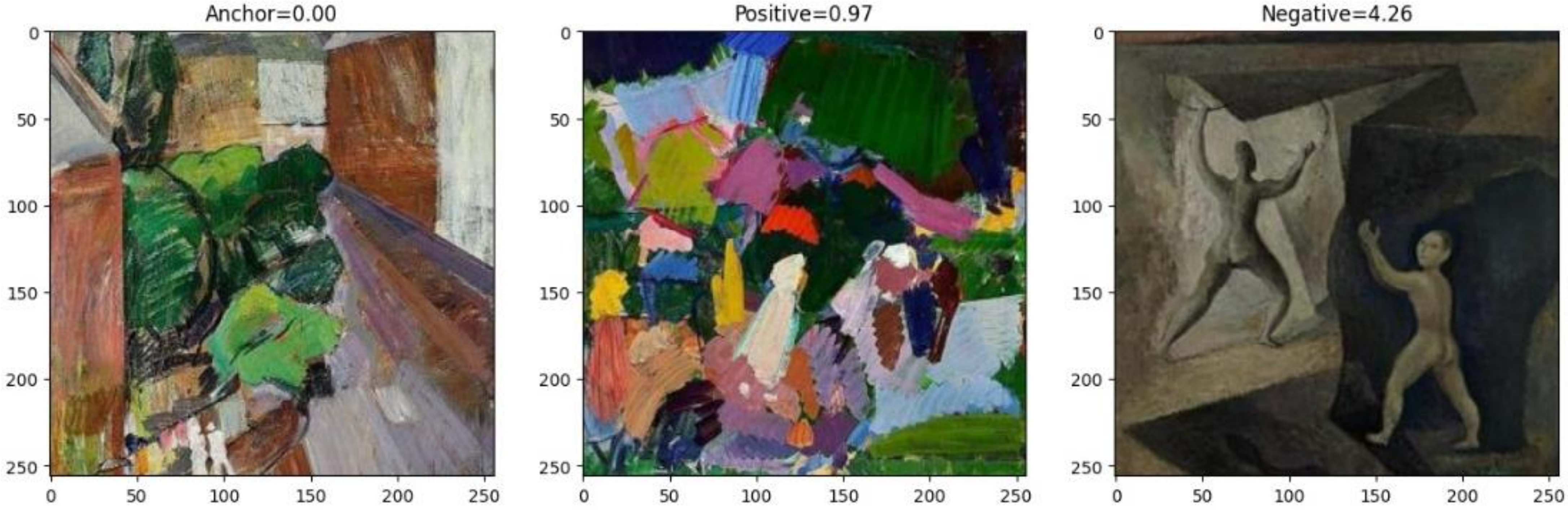
Visualization of test samples

Distance (between anchor and image) is in the title:



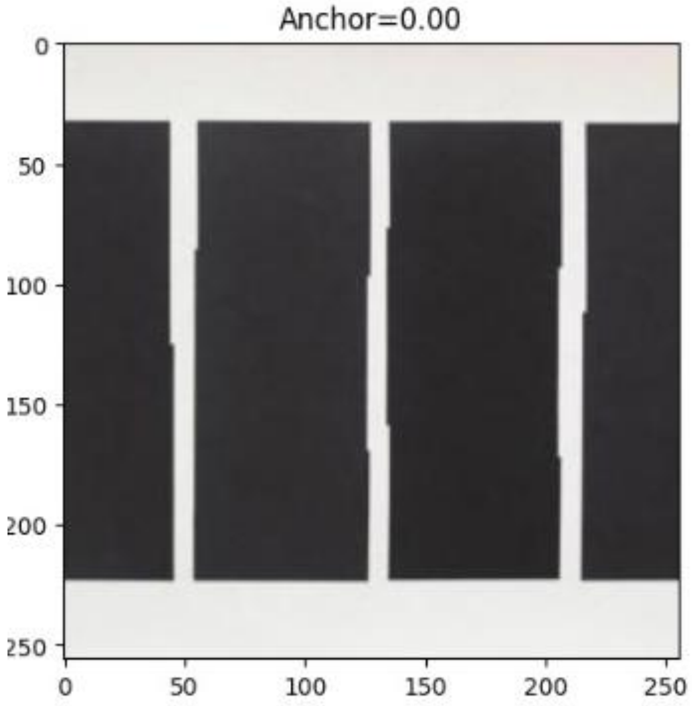
Visualization of test samples

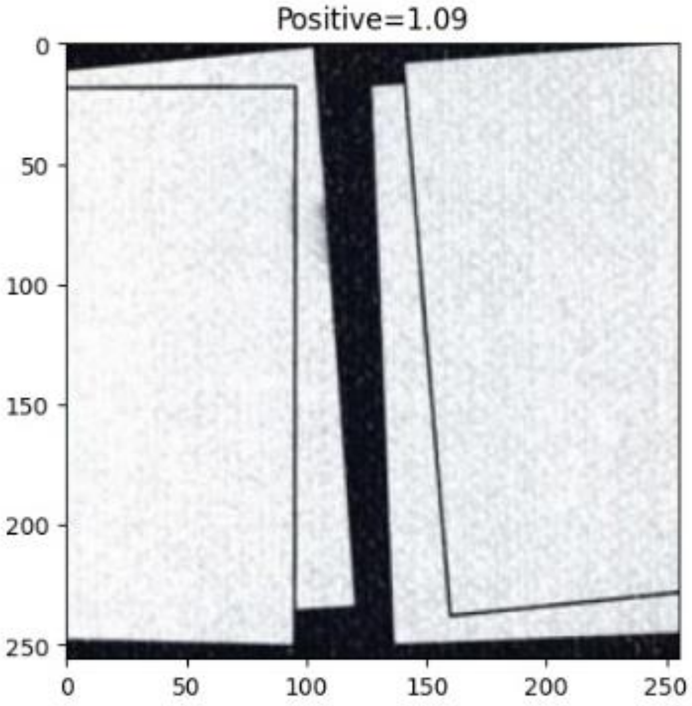
Distance (between anchor and image) is in the title:

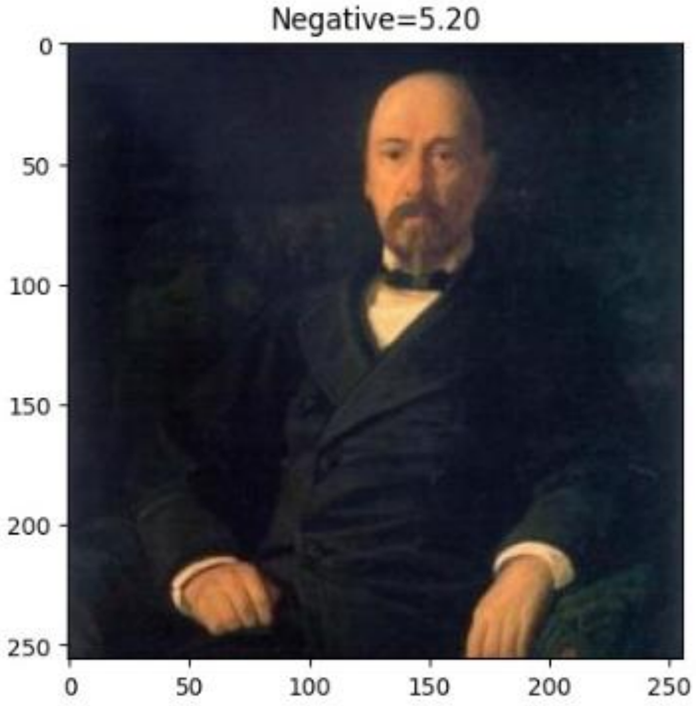


Visualization of test samples

Distance (between anchor and image) is in the title:

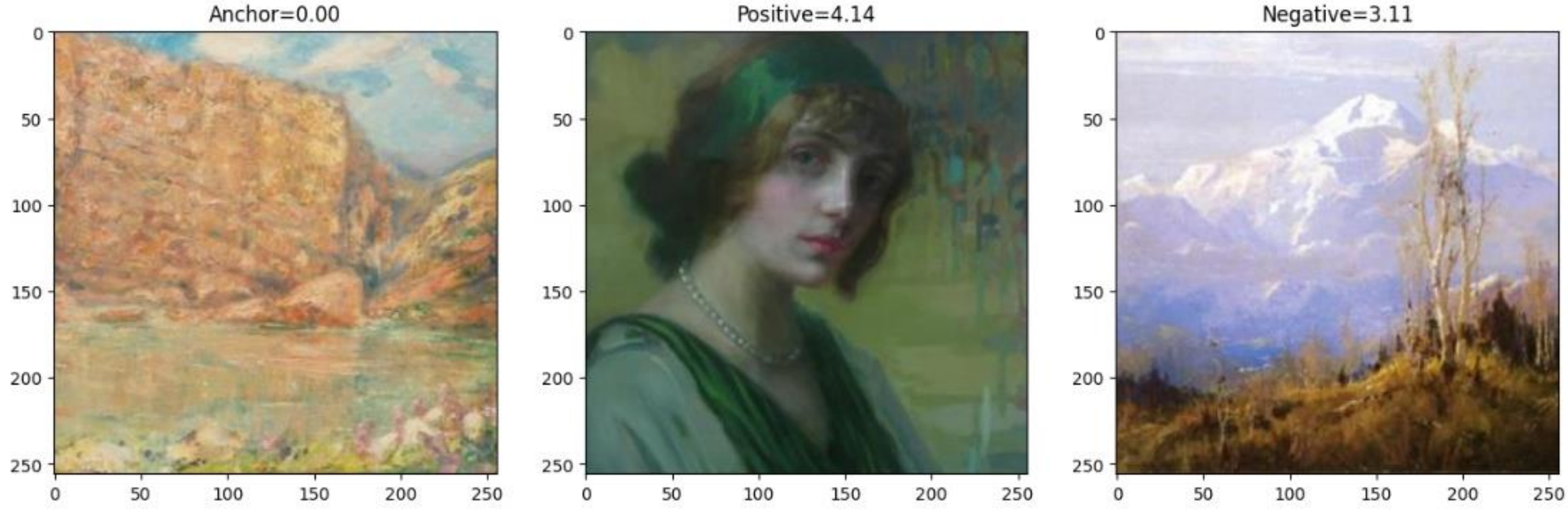




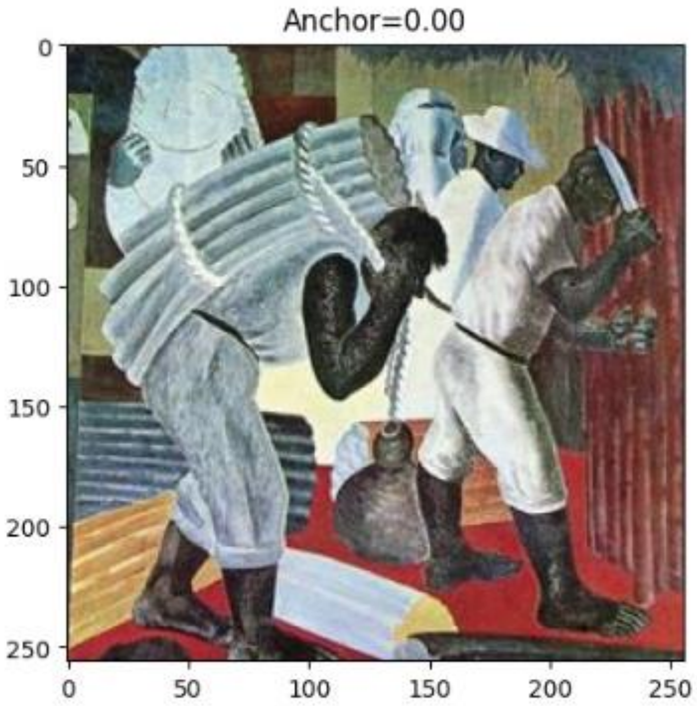


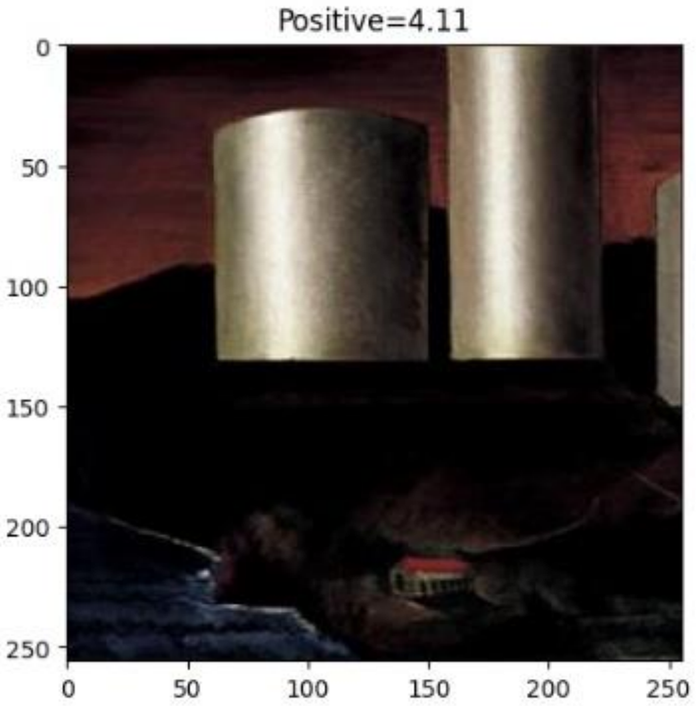
Erroneous predictions

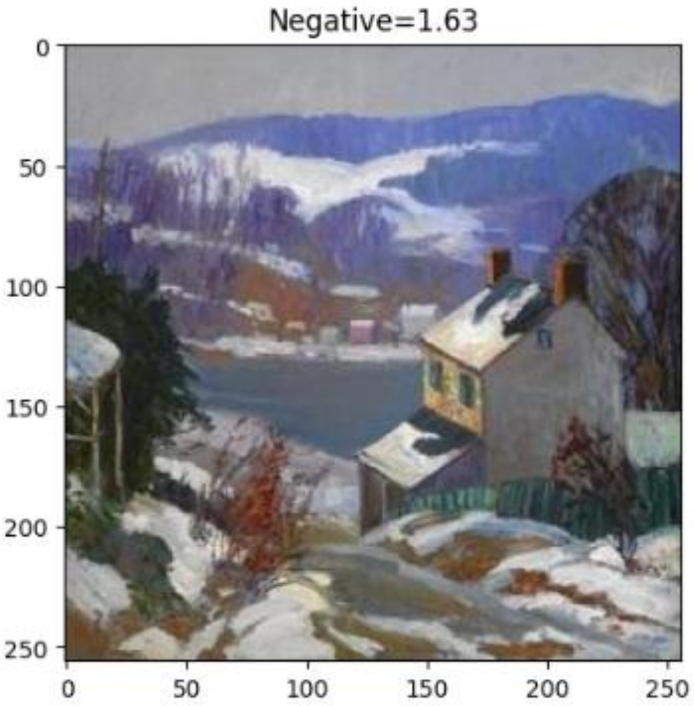
Admittedly, some paintings are confusing even for the human eye.



Erroneous predictions







Conclusion

- Based on the results of the experiment, we conclude that CNN networks are great for exctracting features from visual images (artwork in our case).
- Metric learning (like siamese networks) enables extrapolation to unseen classes (new painters in our case - not even one sample (painting) of these classes (painters) are seen during the training process). This is largly due to the fact that the network is not being fit to specific classes, but rather learning which features are important (and should be considered) when comparing paintings of two painters in order to distinguish an artist's unique painting style.
- We implemented two approaches to deal with building the feature space for the paintings. Both of them can be used in our code by turning a simple flag in the code: **pair_triplet = True/False (False = two nets, contrastive loss ; True = Three nets, Triplet loss)**.


Questions