Painter by Numbers Project

Classifying Image Pairs by Artists

Murad Khoury – 206538928
Ram Khoury - 211912993

Introduction

The primary goal of our project, is to develop a model capable of determining whether two given images belong to the same artist or not.

Key Components of the Project:

- **Dataset Preprocessing:** Cleaning, organizing, and augmenting the dataset to enhance model performance and generalization.

- **Neural Network Architecture:** Implementing a Siamese neural network to learn embeddings for image pairs, facilitating similarity comparison and choosing an appropriate loss function.

- **Data Augmentation Techniques:** Employing transformations like flipping, rotation, color jittering, random perspective to augment the dataset, improving the model's ability to learn diverse artistic styles and variations preventing overfitting.

Data Preprocessing

- First, we downloaded all the data from Kaggle, we made sure to buy a Google Drive cloud so that we could upload the zipped data there.
- We wrote a code where we go through all the images, unzip them, and then perform a CenterCrop for all the images.
- Downsizing our data by filtering out bad datapoints.
- Building the database in a way that maximizes the number of different artists for the model to learn from.
- Making sure that there is no intersection between train, validation and test artists.

Data Augmentation

Data Augmentation was necessary for diversifying the training data and preventing overfitting.

We used these augmentations:

- ❑ Center Crop
- ❑ Random Vertical Flip
- ❑ Random Horizontal Flip
- ❑ Random Rotation
- ❑ Color Jitter
- ❑ Random Perspective
- ❑ Normalize

Data Augmentation Examples

Original Anchor

Anchor transformed with Normalize

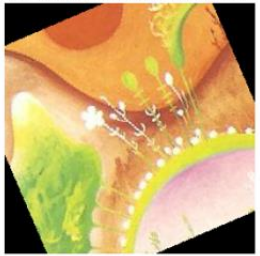
Anchor transformed with RandomAffine

Anchor transformed with ColorJitter

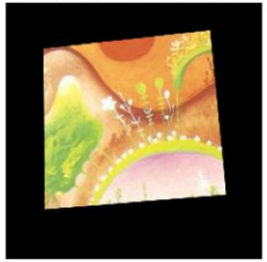
Original Positive

Positive transformed with RandomHorizontalFlip

Positive transformed with RandomRotation



Positive transformed with RandomPerspective



PyTorch's Dataset Class

This custom dataset class facilitates the loading and preprocessing of image data for the project using Data Loaders.

This allows us to load the data into memory dynamically in the form of batches, allowing the use of a larger dataset.

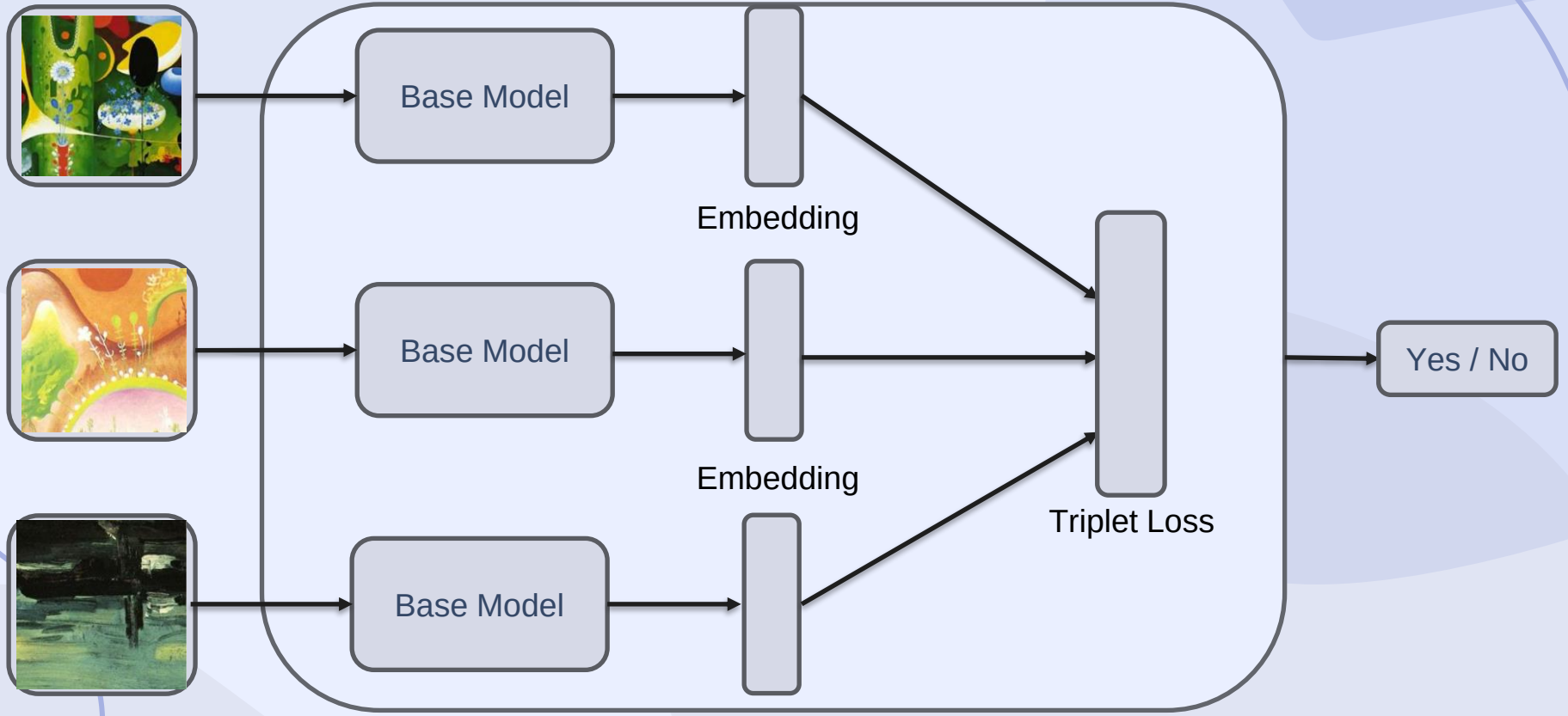
Siamese Network

- The Siamese Network is designed to learn embeddings for pairs of images and classify them based on the similarity of their artists.

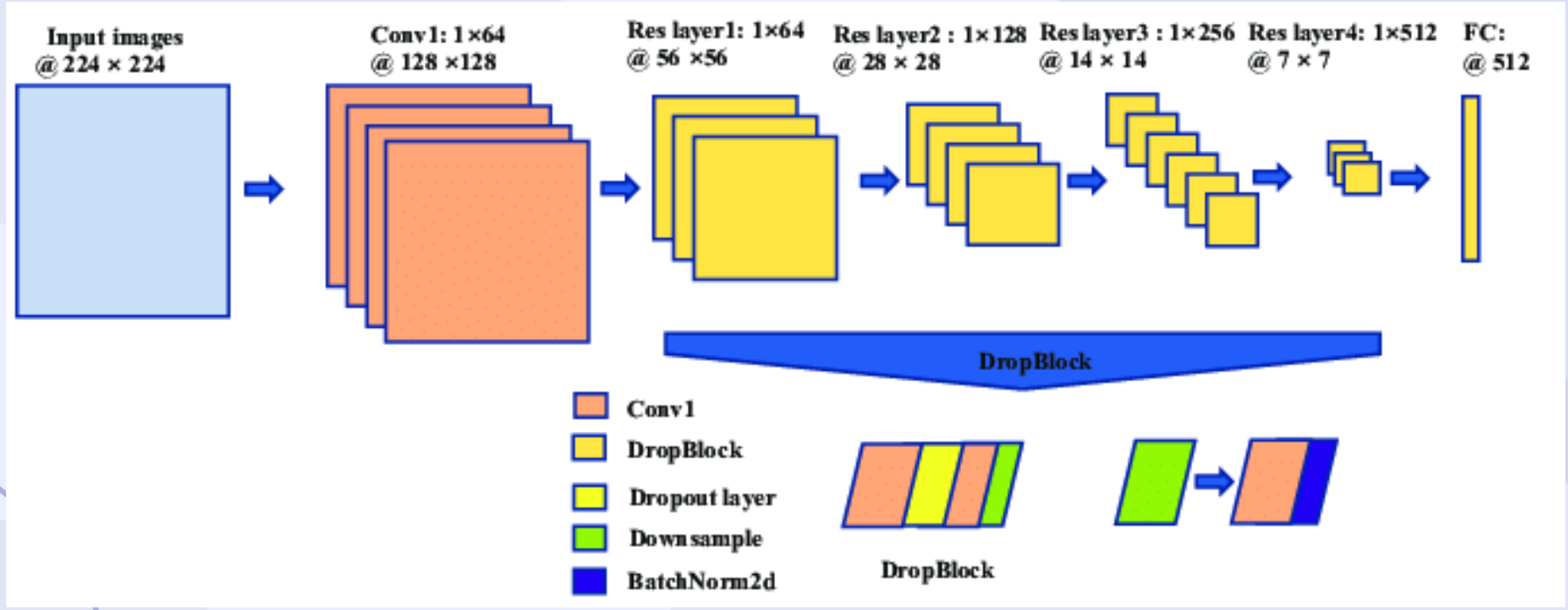
Architecture of the Network:

- **Base model:** Pre-trained Resnet34 model
- **Embedding Layer:** Last layer of base model was modified to output 256- dimensional embeddings.
- **Forward:** takes triplets of images (anchor, positive, negative) as input and passed through the base model to extract embeddings.

Siamese Network

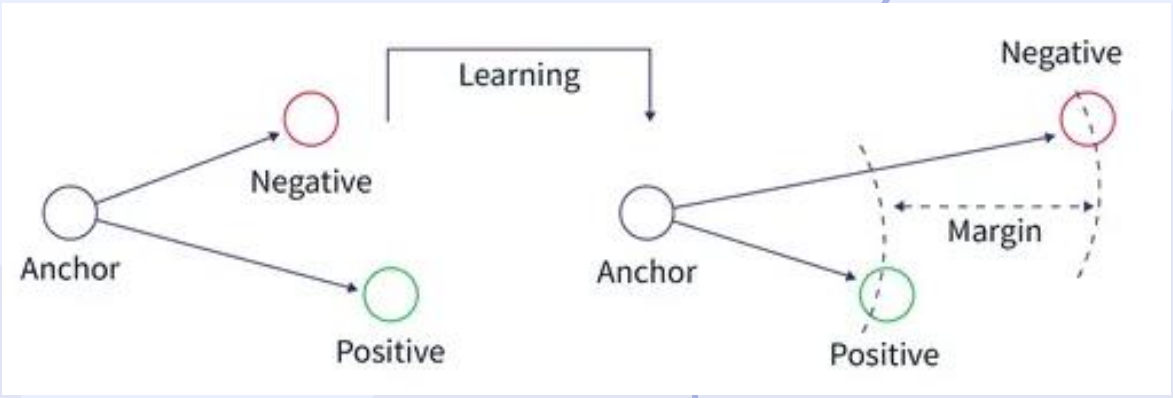


Resnet34 – Base Model



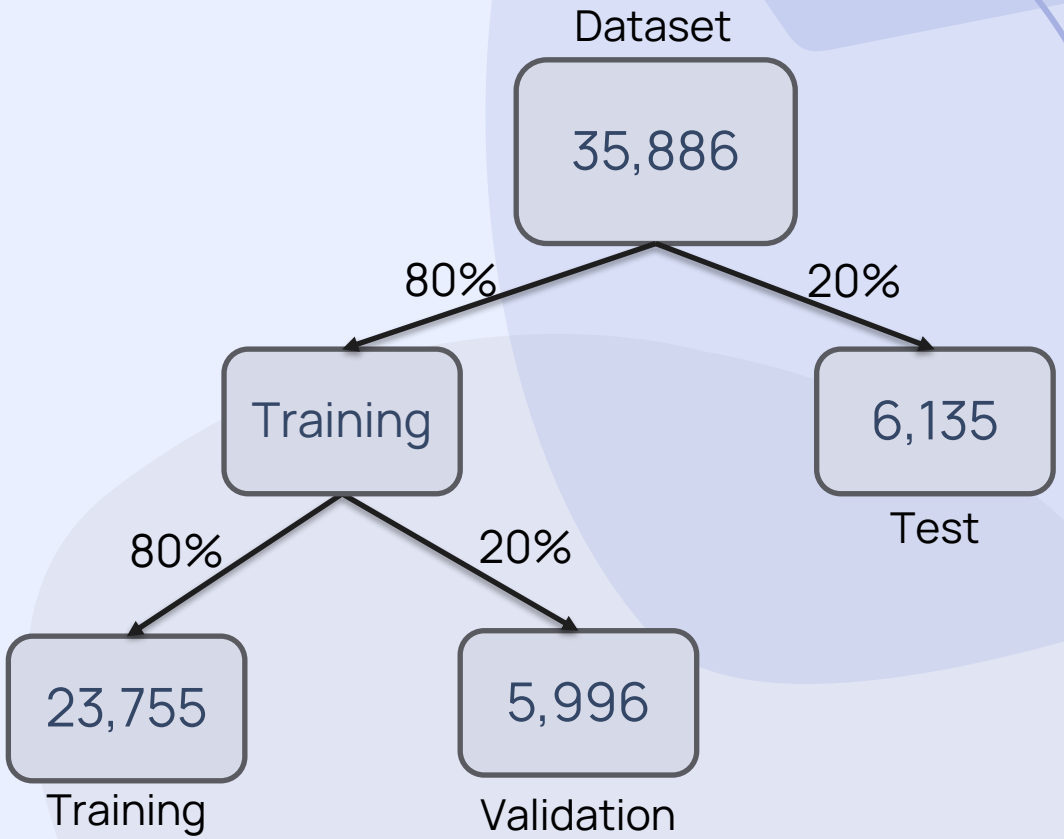
Triplet Loss

- Encourages the model to map anchor and positive samples closer while pushing anchor and negative samples apart.
- Margin determines the minimum difference required between the positive and negative pairs to contribute to the loss.

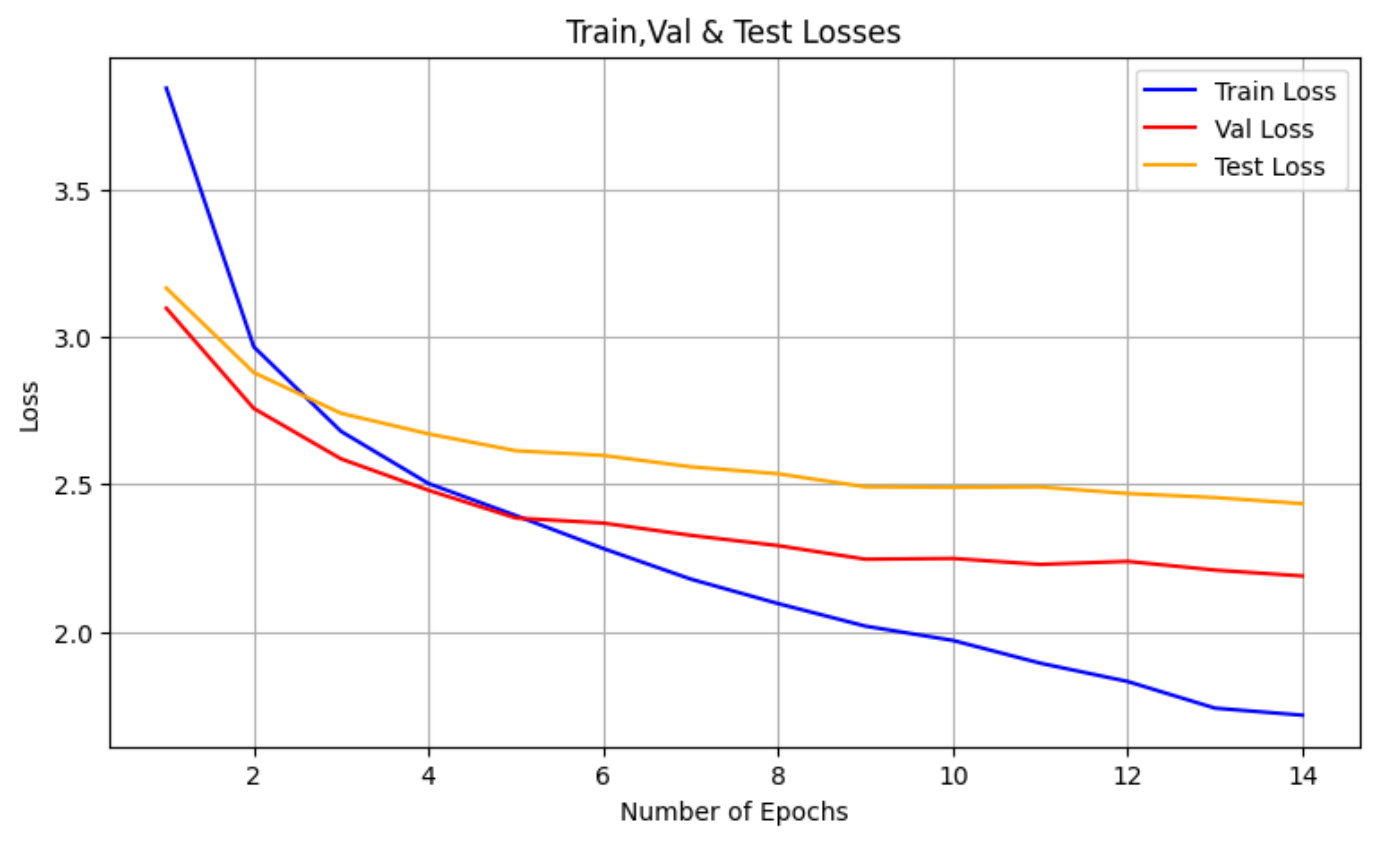


Data Split & Parameters

- Number of epochs: 14
- Batch size: 64
- Margin: 5
- Learning rate: 0.000006



Train, Val and Test Losses



Accuracy Results

- Train Accuracy : 86.1587%
- Validation Accuracy : 81.3209%
- Test Accuracy : 79.7555%

Thank You !

Questions?