# Picture Recognizer for Prado Museum Pictures

Ali Safaei

November 2024

## Abstract

This project is aimed at designing and implementing an image classifier that can recognize pictures within a subset of the Prado Museum Pictures dataset (an available dataset on the Kaggle website). The project dataset is classified based on the four most frequent techniques used in the artworks in the Prado Museum Pictures dataset. These techniques include oil painting, coinage, sculpture and composite pencil. Three different models using convolutional neural networks have been introduced, and their performances have been analysed. All the models are based on transfer learning, utilizing three different pre-trained feature extractors.

From a high-level perspective, each model consists of a pre-trained feature extractor followed by a fully connected layer with 128 neurons, and then a softmax layer. Dropout and data augmentation techniques, which can be helpful to prevent overfitting, are employed. The pre-trained feature extractors are based on the VGG16, ResNet50 and InceptionV3 models, which have been previously trained on the ImageNet dataset. Among the models, the one using the pre-trained feature extractor based on InceptionV3 outperformed the others, achieving an 98.57% accuracy on the test set.

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work, and including any code produced using generative AI systems. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

## 1.Introduction

In this project, the aim is to design and implement an image classifier capable of classifying images acquired from the Prado Museum Pictures dataset into four different categories of “oil painting”, “coinage”, “sculpture” and “composite pencil”.  Three different models with different architectures have been introduced. The models use transfer learning, with the first model using VGG16-based pre-trained feature extractor, the second model using a ResNet50-based pre-trained feature extractor, and the third model using an InceptionV3-based pre-trained feature extractor, while in all the models the feature extractor is followed by a fully connected layer with 128 neurons and then a softmax layer. The pre-trained feature extractors have already been trained on the ImageNet dataset. Also, data augmentation and dropout techniques have been used in all three models. The results for all three models are compared and analysed.

This report is organized as follows: Section 2 provides a description of the dataset and its preprocessing. Section 3 presents some of the theoretical concepts and the model components. Section 4 elaborates on all the six different models, their experimental results and analysis. Finally, Section 5 presents the project’s conclusion.

## 2. Dataset and Preprocessing

The Prado Museum Pictures dataset, which is an available dataset on the Kaggle website, consists of 13487 pictures of different artworks belonging to the Prado Museum in Spain. While different kinds of techniques have been used for different artworks, pictures from the four most frequent techniques have been used for this project. The dataset used for the project contains 6286 images and all the images have been converted from JPEG to RGB. For the normalization purpose all the pixel values for each RGB channel have been scaled from [0, 255] to [0, 1]. Also, each RGB channel has a resolution of 224 by 224 pixels.

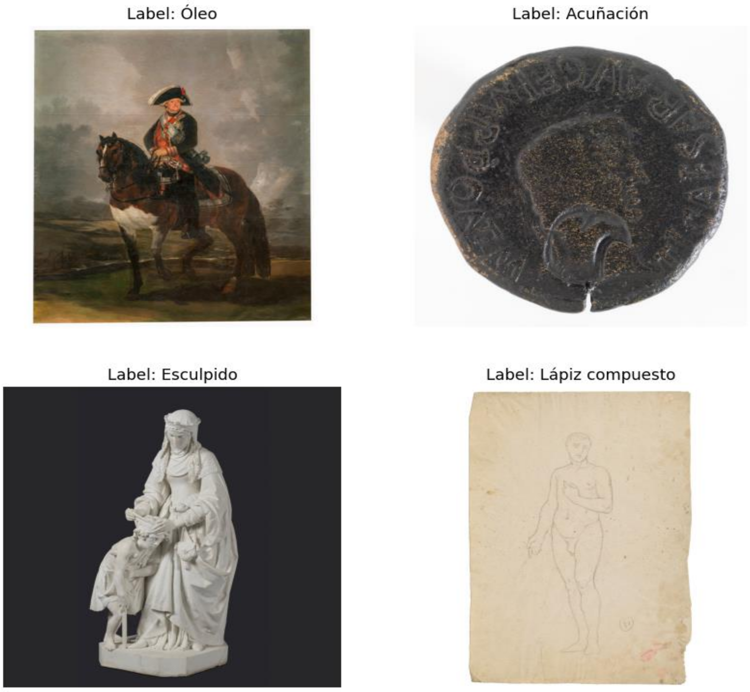

Figure 1: A sample from each of the four techniques

1

# 3. Models Components and Training Process

## 3.1. VGG16 Architecture

VGG16 is a convolutional neural network architecture which has 16 layers. This architecture consists of 13 convolutional layers with (3, 3) filters and 3 fully connected layers. Additionally, max pooling layers are used to reduce the spatial dimensions of feature maps produced by the convolutional layers in the architecture. VGG16 provides a deep architecture which is relatively simple, with having a uniform structure that stacks multiple convolutional layers before each max pooling layer.

In this project, while using the pre-trained version of VGG16 which is already trained on the ImageNet dataset, the top fully connected layers of the architecture have been removed and the rest of the architecture has been used as a pre-trained feature extractor to be used with other components in the first introduced model. This pre-trained feature extractor has 14,714,688 non-trainable parameters.

## 3.2. ResNet50 Architecture

ResNet50 is a convolutional neural network architecture, which has mitigated the vanishing gradient problem in deep networks by introducing the concept of residual learning. It consists of different layers, including convolutional layers, batch normalization layers, a global pooling layer and a fully connected layer. The innovative idea in this architecture is the use of skip connections (residual connections), which bypass certain layers and allow gradients to flow more easily through the network during backpropagation.

In this project, while using the pre-trained version of ResNet50 which is already trained on the ImageNet dataset, the top global average pooling layer and fully connected layer of the architecture have been removed and the rest of the architecture has been used as a pre-trained feature extractor to be used with other components in the second introduced model. This pre-trained feature extractor has 23,587,712 non-trainable parameters.

## 3.3. InceptionV3 Architecture

InceptionV3 is a convolutional neural network architecture that employs inception modules capable of multiscale feature extraction by using filters of different sizes in parallel. By applying these parallel convolutional operations, the network can process the information in different spatial resolutions, helping it extract features at multiple scales simultaneously.

In this project, while using the pre-trained version of InceptionV3 which is already trained on the ImageNet dataset, the top global average pooling layer and fully connected layer of the architecture have been removed and the rest of the architecture has been used as a pre-trained feature extractor to be used with other components in the third introduced model. This pre-trained feature extractor has 21,802,784 non-trainable parameters.

## 3.4. Data Augmentation Layer

To generate new images and increase the diversity of the model’s training data, we can use various data augmentation techniques. In this project, image transformation techniques, including Horizontal Flip, Random Rotation (with a factor of 0.1), and Random Zoom (with a factor of 0.1), have been used to create new images from the existing images in the training dataset. Generating new images for training the classification model can be helpful to mitigate overfitting.

### 3.5. Dropout Layer

Dropout technique is a method that can help mitigate overfitting in the model while being utilized during the training process of neural networks. Using this technique, in each iteration of the training process, some neurons in a specific layer will be randomly dropped out. The fraction of neurons to be dropped out in each iteration is controlled by a parameter called the dropout rate, which is a value between 0 and 1.

### 3.6. Global Average Polling

To reduce the spatial dimensions of a feature map, a Global Average Pooling layer can be used at the end of convolutional layers in convolutional neural networks. Using the global average pooling technique, each feature map is averaged across its entire spatial dimension producing a single value per feature map. As a result, applying this technique to a multi-dimensional feature map will be a vector that has a size equal to the number of channels in the feature map while all the information inside the feature map will be used for the obtained result. This technique is also useful for reducing the risk of overfitting.

### 3.7. Training Process

Considering four classes that are supposed to be detected using the model, this project is concerned with a multiclass classification problem, so the loss function used here is categorical cross-entropy loss function. A mathematical description for this loss function is presented below:

$$L = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{C} y_{ij}\log(p_{ij})$$

N is the number of samples, $y_{ij}$ and $p_{ij}$ refer to the true label and predicted probability for class $i$ of sample $j$, and C is the number of classes.

The Adam (Adaptive Moment Estimation) optimizer is used in this project to find the optimal set of parameters that minimize the categorical cross-entropy loss function.

In the training process for all the models in this project, the batch size has been set to 32. Although the number of epochs for all the models’ training setup is set to 20, the models stopped being trained before arriving in epoch 20 due to an early stopping mechanism which is devised in the training process.

The early stopping mechanism can be used to prevent the model from overtraining and can be helpful to reduce overfitting. Using this mechanism, the model stops being trained when a specific monitored metric fails to improve for a specific number of epochs which is called “patience”. In this project, the monitored metric for the early stopping mechanism is “val_loss” and the “patience” is 3. It means that if the monitored “val_loss” does not improve after 3 epochs, the training process will be stopped.

# 4. Models Development and Performance Analysis

## 4.1. Model 1

The order of layers for this model is described sequentially below:

1. Data augmentation layer

2. VGG16-based pre-trained feature extractor

3. Global Average Pooling layer

4. Dense layer with 128 units and ReLU activation function

5. Dropout layer with a dropout rate equal to 0.25

6. Dense layer with four units and Softmax activation function

In the training process of this model, the number of epochs for the training process is set to 20 but due to the use of early stopping technique the training process stopped after 7 epochs.

| Train Loss | Train Accuracy | Test Loss | Test Accuracy |
|---|---|---|---|
| 0.0385 | 0.9894 | 0.1044 | 0.9666 |

Table 1: Model 1 performance on the training set and the test set after 7 epochs

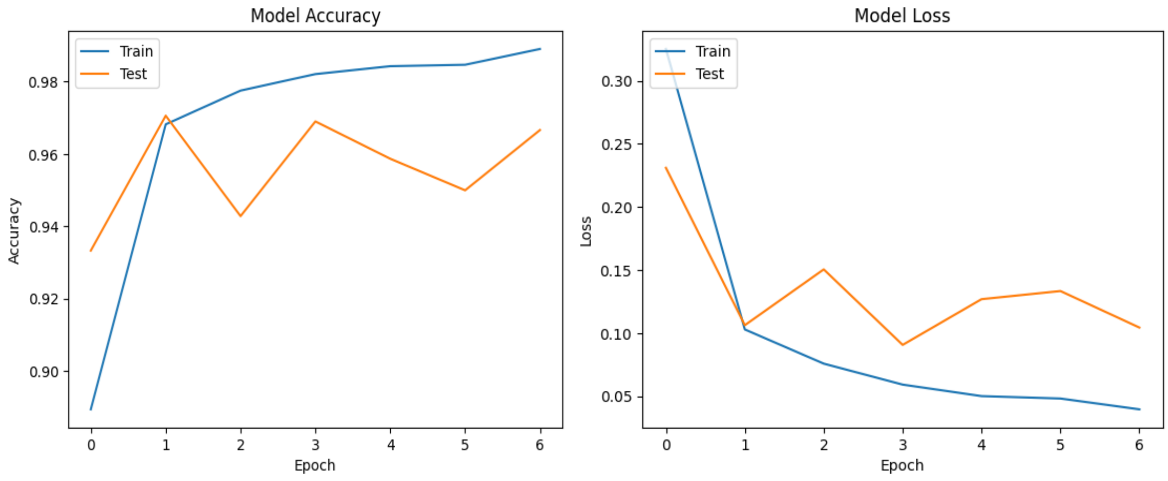

Figure 2: Model 1 performance on the training set and the test set at each epoch

## 4.2. Model 2

The order of layers for this model is described sequentially below:

1. Data augmentation layer

2. ResNet50-based pre-trained feature extractor

3. Global Average Pooling layer

4. Dense layer with 128 units and ReLU activation function

5. Dropout layer with a dropout rate equal to 0.25

6. Dense layer with four units and Softmax activation function

In the training process of this model, the number of epochs for the training process is set to 20 but due to the use of early stopping technique the training process stopped after 10 epochs.

| Train Loss | Train Accuracy | Test Loss | Test Accuracy |
|---|---|---|---|
| 0.2597 | 0.9204 | 0.4434 | 0.8380 |

Table 2: Model 2 performance on the training set and the test set after 10 epochs

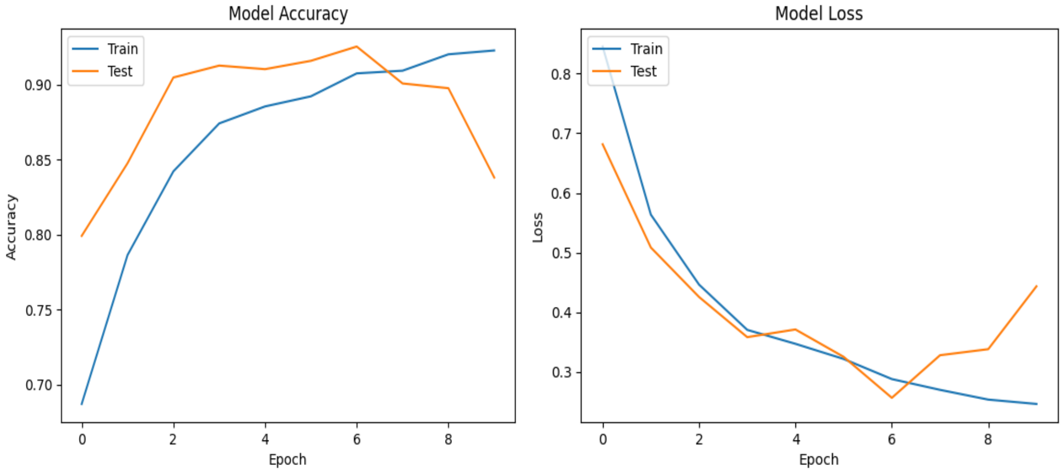

Figure 2: Model 2 performance on the training set and the test set at each epoch

## 4.3. Model 3

The order of layers for this model is described sequentially below:

1. Data augmentation layer

2. InceptionV3-based pre-trained feature extractor

3. Global Average Pooling layer

4. Dense layer with 128 units and ReLU activation function

5. Dropout layer with a dropout rate equal to 0.25

6. Dense layer with four units and Softmax activation function

In the training process of this model, the number of epochs for the training process is set to 20 but due to the use of early stopping technique the training process stopped after 7 epochs.

| Train Loss | Train Accuracy | Test Loss | Test Accuracy |
|---|---|---|---|
| 0.0166 | 0.9944 | 0.0456 | 0.9857 |

Table 3: Model 3 performance on the training set and the test set after 7 epochs

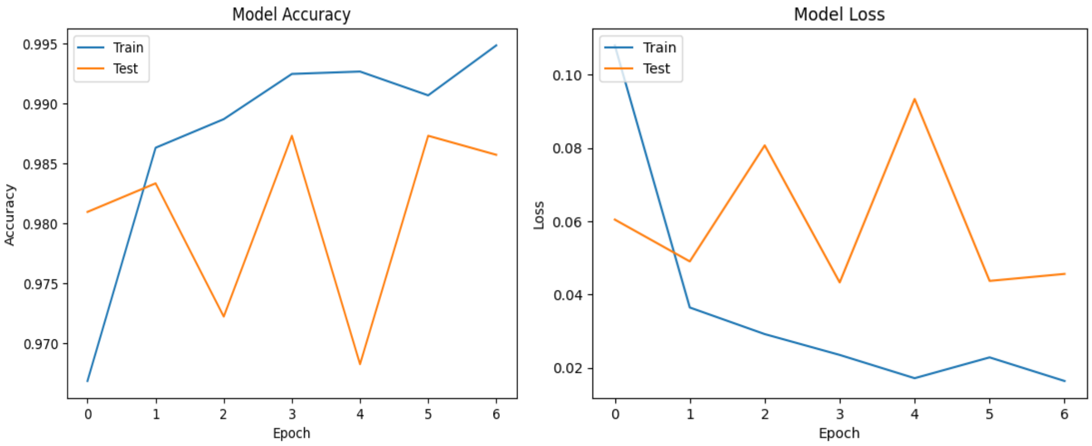

Figure 5: Model 3 performance on the training set and the test set at each epoch

## 4.4. Models Performance Analysis

Based on the obtained results, the third model is the highest-performing model while the first model is also performing very well but not as well as the third one. Also, the second model is the lowest performing one with a huge difference between its performance on the test set and the training set that is indicating a high variance error and the overfitting problem for the model.

Some hypothetical reasons may help explain the observed results. Providing a comparison between the first model and the second one, the first model outperforms the second model, even though the second model, which uses a ResNet50-based pre-trained feature extractor, is much deeper than the first one, which uses a VGG16-based pre-trained feature extractor. As a feature extractor, the one that is based on the ResNet50 can capture very complex features which may be unnecessary for the specific task in this project. On the other hand, the VGG16-based feature extractor is a shallower network, that can perform well in extracting relatively simpler features that seem sufficient to distinguish different artwork techniques effectively. So, maybe the very deep architecture of ResNet50-based feature extractor with its many stacked convolutional layers can lead to the extraction of overly complex features that are probably not necessary for the image classification task in this project.

When comparing the second model and the third one, the third model outperforms the second one. Both models utilize very deep feature extractors, but the third model, which employs an InceptionV3-based pre-trained feature extractor, is capable of capturing features in multiple scales using Inception modules, that apply convolution filters of different sizes. Also, the InceptionV3-based pre-trained feature extractor has 21,802,784 parameters, which is less than the 23,587,712 parameters in the ResNet50-based pre-trained feature extractor used in the second model. The mentioned multiscale feature extraction capability of InceptionV3-based pre-trained feature extractor may lead to extraction of essential features at different scales while providing a deep model with fewer parameters than the second model which uses ResNet50-based pre-trained feature extractor.

If the dataset scales up both in terms of the number of images and the number of classes, we may be able to use the third architecture among the introduced ones, which is deep and capable of extracting features at multiple scales using an InceptionV3-based pretrained feature extractor. Additionally, we can unfreeze some layers at the top of the InceptionV3-based feature extractor and fine-tune them using our dataset. Data augmentation can also help by providing more samples to train the model, considering some probable scenarios in which the number of samples per each class may not be sufficient to train the model properly.