Capstone Project

CLASSIFICATION OF ART PIECES

POOJITHA ANUMUKONDA

Monday, February 18 ,2019

I. Definition

Project Overview

- An art is something that is created with imagination and skill.
- Art in its broadest sense is a form of communication. It means whatever the artist intended and feel is represented in the form of art.
- Art is an act of expressing feelings, thoughts, and observations.
  The most common concept of art as pieces of work whether paintings or sculptures. Indeed art surrounds life, every people in every location. Since time is immemorial, art has existed as long as man. It is a huge part of our culture which shapes our ideas, provides us with a deeper understanding of emotions, self-awareness and more.
  Art work is used for various purposes now-a-days.
  - Decoration
  - Hobby
  - Expressing emotions
  - Religious purpose
  - Social activities i.e from ancient times, people uses paintings and art works to express their protest against their rulers.
    http://www.totalartsoul.com/index.php?option=com_content&task=view&id=2510&Itemid

History:

Drawing:- Drawing goes back at least 16,000 years to Paleolithic cave representations of animals such as those at Lascaux in France and Altamira in Spain. In ancient Egypt, ink drawings on papyrus, often depicting people, were used as models for painting or sculpture. Drawings on Greek vases, initially geometric, later developed to the human form with black-figure pottery during the 7th century BC.

With paper becoming common in Europe by the 15th century, drawing was adopted by masters such as Sandro Botticelli, Raphael, Michelangelo, and Leonardo da

Vinci who sometimes treated drawing as an art in its own right rather than a preparatory stage for painting or sculpture.

Painting:- Like drawing, painting has its documented origins in caves and on rock faces. The finest examples, believed by some to be 32,000 years old, are in the Chauvet and Lascaux caves in southern France. In shades of red, brown, yellow and black, the paintings on the walls and ceilings are of bison, cattle, horses and deer.

Paintings of human figures can be found in the tombs of ancient Egypt. In the great temple of Ramses II, Nefertari, his queen, is depicted being led by Isis.[7] The Greeks contributed to painting but much of their work has been lost. One of the best remaining representations are the Hellenistic Fayum mummy portraits. Another example is mosaic of the Battle of Issus at Pompeii, which was probably based on a Greek painting. Greek and Roman art contributed to Byzantine art in the 4th century BC, which initiated a tradition in icon painting.

https://en.wikipedia.org/wiki/Visual_arts

Application

A) can used by kids in schooling to Cleary differentiate between different art works based on medium. And it is easy to use.

B) Can be used for research purposes on different categories of art pieces.

We can also expect good performances around 80% accuracy with this project.

FUTURE: We can even extend this algorithm to remaining categories of arts.

In this project, I have taken 5 different art works to classify and their relevant description

Drawings: Drawing is a form of visual art in which a person uses various drawing instruments to mark paper or another two-dimensional medium. ... A drawing instrument releases a small amount of material onto a surface, leaving a visible mark.

Works of paintings: Painting is the practice of applying paint, pigment, color or other medium to a solid surface (support base). The medium is commonly applied to the base with a brush, but other implements, such as knives, sponges, and airbrushes, can be used. The final work is also called a painting

Sculpture: A sculpture is a work of art that is produced by carving or shaping stone, wood, clay, or other materials.

Graphic Art: The fine and applied arts of representation, decoration, and writing or printing on flat surfaces together with the techniques and crafts associated with them.

Iconography: Iconography, as a branch of art history, studies the identification, description, and the interpretation of the content of images: the subjects depicted, the particular compositions and details used to do so, and other elements that are distinct from artistic style. The word iconography comes from the Greek εικών ("image") and γράφειν ("to write").

A secondary meaning (based on a non-standard translation of the Greek and Russian equivalent terms) is the production of religious images, called "icons", in the Byzantine and Orthodox Christian tradition.

Datasets and Inputs

The dataset I am working is downloaded from [1]

The dataset contains about 9000 different styles of art's images

Problem Statement

The aim of this project is to predict or classify the type art pieces it belongs to by visualizing image. In this project I am going to use keras by adding convolutional layers.

- My solution is to apply Keras by adding Convolution Layers ,using optimizer as 'adam',activation function as 'relu' and two maxpooling layers with window size (2,2).

Metrics

- Since this is a Classification problem, for this project I used Accuracy as a Metrics in order to calculate the performance of the model and loss metric as 'categorical_crossentropy'.

II. Analysis

Data Exploration

The dataset contains about 9000 different styles of art works. The dataset contains train and validation sets separately. The images are divided into classes: drawings, paintings, sculpture, graphic arts such as engraving and iconography. For each class there are 1000 to 2000 images. Photos are not reduced to a single size, they have different proportions. [1]

Exploratory Visualization

There are total 5 image categories.

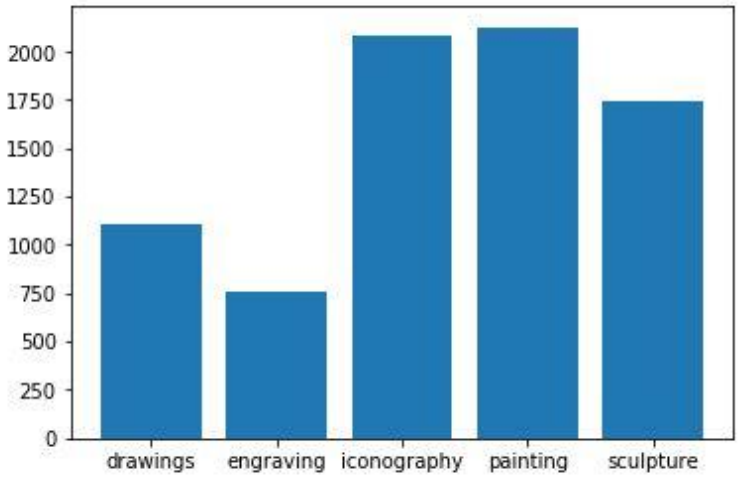
There are 8,685 total art images.

There are 7,819 training images.

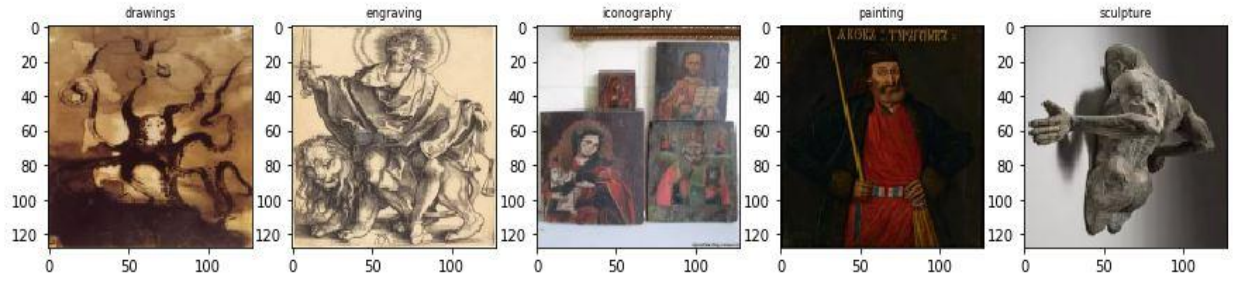
There are 866 testing images.

```
Found 7819 images belonging to 5 classes.
Found 866 images belonging to 5 classes.
```

The data that present in training category:-



Sample images from the training data set



Algorithms and Techniques

Deep learning is a subfield of machine learning with algorithms inspired by the working of the human brain. These algorithms are referred to as artificial neural networks. Examples of these neural networks include Convolutional Neural Networks that are used for image classification, Artificial Neural Networks and Recurrent Neural Networks.

Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) are everywhere. It is arguably the most popular deep learning architecture. The recent surge of interest in deep learning is due to the immense popularity and effectiveness of convnets. The interest in CNN started with AlexNet in 2012 and it has grown exponentially ever since. In just three years, researchers progressed from 8 layer AlexNet to 152 layer ResNet.

CNN is now the go-to model on every image related problem. In terms of accuracy they blow competition out of the water. It is also successfully applied to recommender systems, natural language processing and more. The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision. For example, given many pictures of cats and dogs it learns distinctive features for each class by itself.

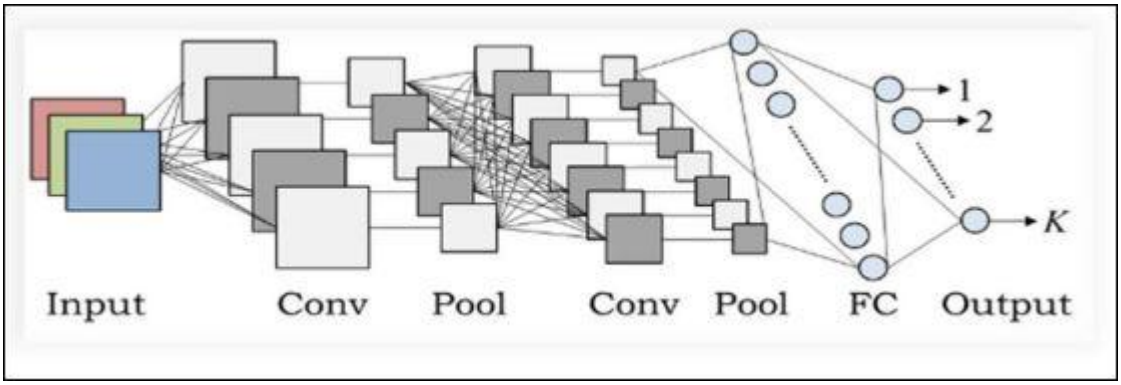
CNN is also computationally efficient. It uses special convolution and pooling operations and performs parameter sharing. This enables CNN models to run on any device, making them universally attractive.

All in all this sounds like pure magic. We are dealing with a very powerful and efficient model which performs automatic feature extraction to achieve superhuman accuracy (yes CNN models now do image classification better

than humans). Hopefully ,CNN trains the data efficiently with less training time.

Architecture

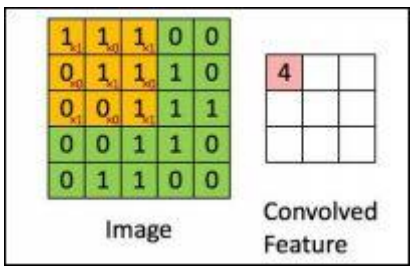
There is an input image that we're working with. We perform a series convolution + pooling operations, followed by a number of fully connected layers. If we are performing multiclass classification the output is softmax. We will now dive into each component.



Convolution

The main building block of CNN is the convolutional layer. Convolution is a mathematical operation to merge two sets of information. In our case the Convolution is applied on the input data using a convolution filter to produce a feature map. There are a lot of terms being used so let's visualize them one by one.

On the left side is the input to the convolution layer, for example the input Image. On the right is the convolution filter, also called the kernel, we will use these terms interchangeably. This is called a 3x3 convolution due to the shape of the filter.
We perform the convolution operation by sliding this filter over the input. At every location, we do element-wise matrix multiplication and sum the result. This sum goes into the feature map. The green area where the convolution operation takes place is called the receptive field. Due to the size of the filter the receptive field is also 3x3.



Here the filter is at the top left, the output of the convolution operation “4” is shown in the resulting feature map. We then slide the filter to the right and perform the same operation, adding that result to the feature map as well.

Pooling

After a convolution operation we usually perform pooling to reduce the dimensionality. This enables us to reduce the number of parameters, which both shortens the training time and combats over fitting. Pooling layers down sample each feature map independently, reducing the height and width, keeping the depth intact.

The most common type of pooling is max pooling which just takes the max value in the pooling window. Contrary to the convolution operation, pooling has no parameters. It slides a window over its input, and simply takes the max value in the window. Similar to a convolution, we specify the window size and stride.

In CNN architectures, pooling is typically performed with 2x2 windows, stride2
and no padding. While convolution is done with 3x3 windows, stride 1 and with padding.

Hyperparameters

Let’s now only consider a convolution layer and pooling, and go over the hyperparameter choices we need to make. We have 4 important hyperparameters to decide on:

• **Filter size:** we typically use 3x3 filters, but 5x5 or 7x7 are also used depending on the application. There are also 1x1 filters which we will explore in another article, at first sight it might look strange but they have interesting applications. Remember that these filters are 3D and have a depth dimension as well, but since the depth of a filter at a given layer is equal to the depth of its input, we omit that.

• **Filter count:** this is the most variable parameter; it’s a power of two anywhere between 32 and 1024. Using more filters results in a more powerful model, but we risk overfitting due to increased parameter count. Usually we start with a small number of filters at the initial layers, and progressively increase the count as we go deeper into the network.

• **Stride:** we keep it at the default value 1.

• **Padding:** we usually use padding.

Implementation

Structurally the code looks similar to the ANN we have been working on.

There are 4 new methods we haven't seen before:

• **Conv2D:** this method creates a convolutional layer. The first parameter is the filter count, and the second one is the filter size. For example in the first convolution layer we create 32 filters of size 3x3. We use relunon-linearity as activation. We also enable padding. In Keras there are two options for padding: same or valid. Same means we pad with the number on the edge and valid means no padding. Stride is 1 for convolution layers by default so we don't change that.

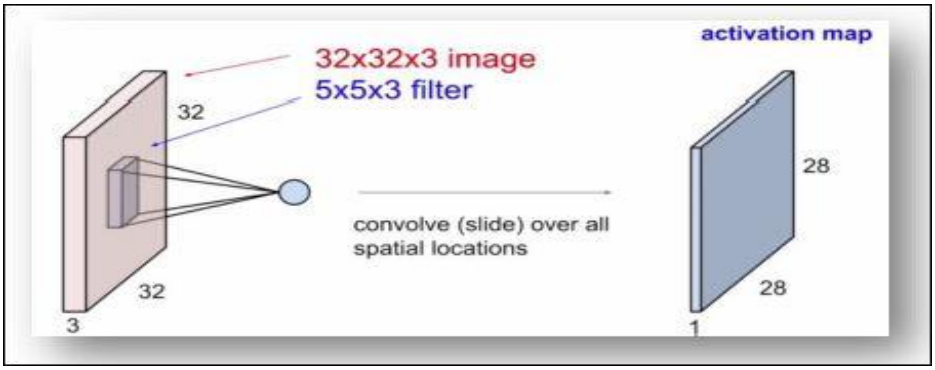
• **MaxPooling:** creates a maxpooling layer, the only argument is the window size. We use a 2x2 window as it's the most common. By default stride length is equal to the window size, which is 2 in our case, so we don't change that.

• **Flatten:** After the convolution + pooling layers we flatten their output to feed into the fully connected layers

• **Dropout:** Dropout is used to prevent overfitting and the idea is very simple. During training time, at each iteration, a neuron is temporarily "dropped" or disabled with probability p. This means all the inputs and outputs to this neuron will be disabled at the current iteration.

My Model Architecture

- In this project to solve the problem , I used convolution network as a technique (i.e),adding convolution Layers which consists of input layer, hidden layer and output layer. I added the Convolution Layer at first so I mentioned the input shape as (64,64,3) .The Convolution Layer is represented as



- The activation function used is 'relu' which is defined as positive part of the argument, Since we are dealing with images this activation function should be used because the pixel values always lies within the positive range

- The two max pooling layers are used in order to reduce the dimension more specifically defined as dimensionality reduction. The Pooling window size I considered is 2*2 matrix.
- At last the dense layer is added with 5 nodes and with sigmoid function as activation function.

Benchmark

- The Benchmark model I considered is one convolution Layer with one maxpooling layer, flatten layer, dense layer with 5 units ,activation of 'softmax' and also 50 number of epochs which gives me testing accuracy as 65%.
- In order to beat the accuracy of Benchmark model ,I added one more Convolution layer and Maxpooling Layer with same number of epochs as 50 .
- Successfully I increased the accuracy around 80% which can be considered as best model to evaluate

Architecture:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 31, 31, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 31, 31, 32)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 30752)             0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                1968192
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 325
=================================================================
Total params: 1,969,413
Trainable params: 1,969,413
Non-trainable params: 0
_________________________________________________________________
```

It gives me the training accuracy of 76.48%

It gives me the testing accuracy of 75.90%

III. Methodology

Data Preprocessing

- In case of Image Processing ,it is very important to perform the data augmentation .I have performed augmentation by using ImageDataGenerator() method which will rescales the images and flips images horizontally
- This process will play major role because there will be chance of images where the size can be different and images represented in horizontal direction, so data augmentation will helps to evaluate the model even the images are not organized.
- The class have a method named flow_from_directory () which will load the train and test datasets into different directories.
- The Images height and width for model is considered as 64,64 and with 3 channels, so that every Image will have same dimensions.

Implementation

As an implementation, I considered two convolution layers are with 2 maxpooling layers and activation function as 'relu' and a dense layer with 5 units and softmax as activation function. Dropouts are the regulation techniques which are also added to avoid the overfitting .

Architecture:

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d_2 (MaxPooling2	(None, 31, 31, 32)	0
dropout_3 (Dropout)	(None, 31, 31, 32)	0
conv2d_3 (Conv2D)	(None, 29, 29, 32)	9248
max_pooling2d_3 (MaxPooling2	(None, 14, 14, 32)	0
dropout_4 (Dropout)	(None, 14, 14, 32)	0
flatten_2 (Flatten)	(None, 6272)	0
dense_3 (Dense)	(None, 500)	3136500
dropout_5 (Dropout)	(None, 500)	0
dense_4 (Dense)	(None, 128)	64128
dropout_6 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 5)	645

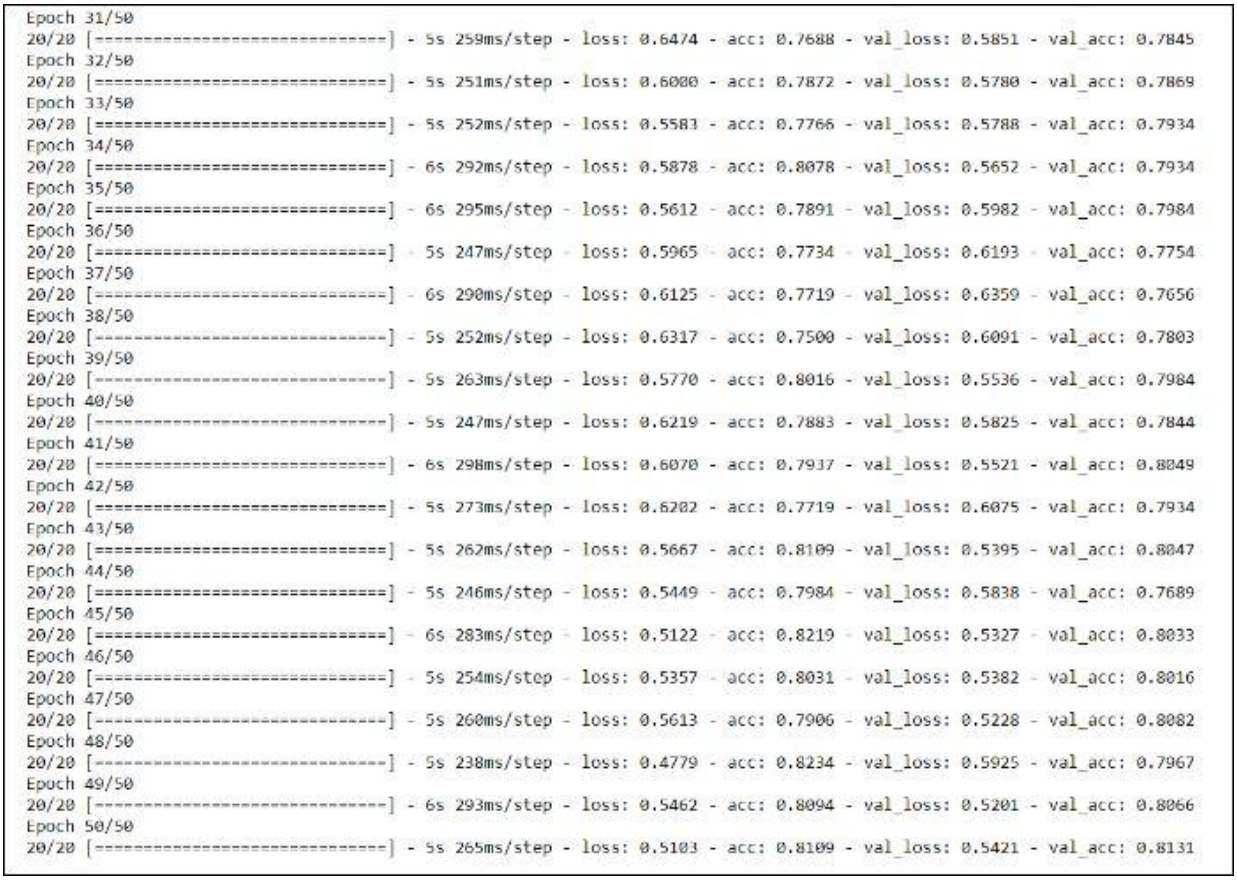
Total params: 3,211,417
Trainable params: 3,211,417
Non-trainable params: 0

It gives the training accuracy of 81.09%.

It gives the testing accuracy of 81.31%.

Model training

The training is done on both bench mark model and on our target model with batch size of 20 and epoch 50. Once the training starts it will be as follows



We can observe here during training as the training and validation loss decreases our training and validation accuracy increases .Here the validation dataset is treated as testing dataset. So the testing accuracy is equal to validation accuracy.

Refinement

- First model gets an accuracy around 65% ,so to improve my accuracy I have added one more convolution layer and maxpooling layer. In dense layer iI have changed the batch size.
- These changes in the benchmark model give me the better accuracy about 79% which is better than the accuracy of benchmark model.

IV. Results

Model Evaluation and validation

- In Final model, the accuracy of training samples have nearly 81% accuracy. Any ways the testing accuracy is 79% which can be noted as good model for performance .
- Consider the Loss for training and testing where testing images has low loss that means it is performing good predictions at testing.

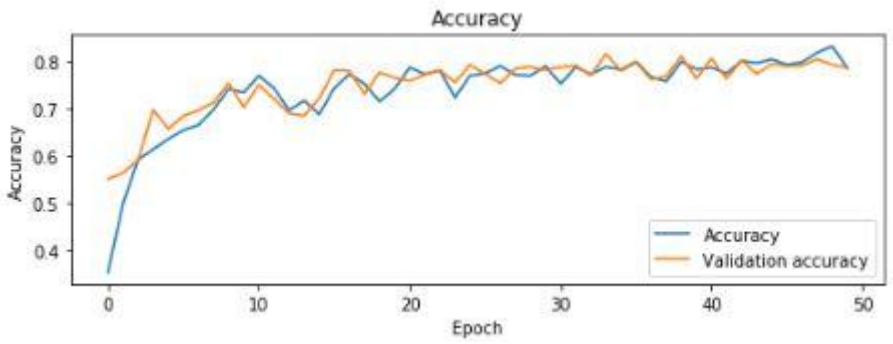
Justification

The final results occurred from the model found stronger than the benchmark model reported earlier in terms of improving the accuracy and model implementation time. For benchmark model it takes lot of time while fitting the data after optimizing the parameters final model did not take as much of time to evaluate. So I consider 2nd model as best model for classifying the art pieces

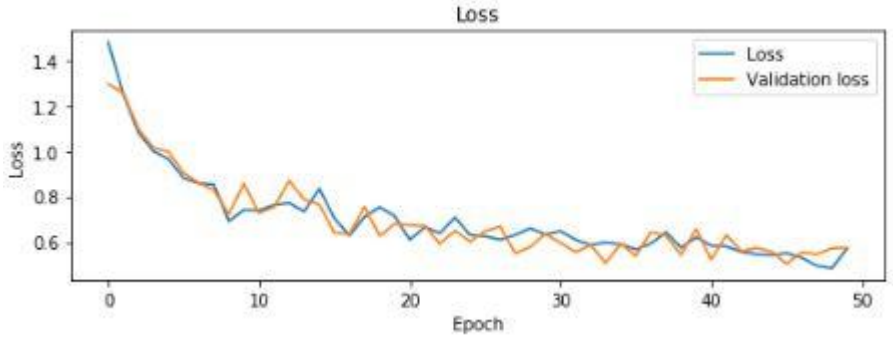
V. Conclusion

Free Form Visualization

- Accuracy Results



- Loss Result



Reflection

1. First I have gone through many problems in kaggle and found that “The Classification of Art Pieces” more interesting.

2. Then I gained knowledge about Keras and their libraries.

3. I downloaded the dataset and plots of art images in python by using matplotlib and keras from the kaggle.

4. I have also learned how to use kaggle kernals and how to commit it and reproduce my work.

5. Then, I thought to use Deep Learning approach to implement this model, so I used Convolution Neural Network by using Keras.

6. Starting I added one convolution Layer and one maxpooling layer and also some dense layers to create any benchmark model which doesn’t yield the good results.

7. To yield good results I added one more convolution, a maxpooling layer and changed some of the parameters which increase my model’s accuracy.

8. I learn that Image resizing is more important in image classification because we can’t expect every image of same size.

9. From all of the above steps ,I felt more difficult at parameter tuning to get good optimized model.

10. Last but not least without visualizing results we can’t trust the robustness of a model.

Improvement

In order to improve my model performance , Transfer Learning technique can be more effective .I think GridSearch mechanism can also be applied here for improvement in case of parameter tuning. we can include many art pieces categories to it . Rather than differentiating between different art works/pieces, we can also train model to differentiate between what is an art work and what is not.

References

[1]: https://www.kaggle.com/thedownhill/art-images-drawings-painting-sculpture-engraving