Artist Identification Description

Pre-processing

The given dataset has 1991 images in total. I have separated randomly the dataset as follows:

- Training images – 1440 images
- Validation images – 144 images
- Testing images – 407 images

Each of the images are resized to sizes of 256x256 before they are fed into the CNN.

During training of the CNN, data augmentation is done only on the training images. Data augmentation is required because the networks are data hungry/driven models and hence more the data better is the performance. Data augmentation additionally makes the network more robust and helps it generalise.

Data augmentation techniques used in this implementation are:

- Adding Gaussian noise
- Horizontal flip
- Vertical flip

Few more augmentation techniques are possible:

- Horizontal shift
- Vertical shift
- Both horizontal and vertical flip
- Adding different amounts of noise to different channels
- Random rotations of the image
- Contrast normalisation

The images are normalized by dividing by its maximum value (255). The normalization is done so that the values are kept in the range 0 and 1, and they don't explode while training.

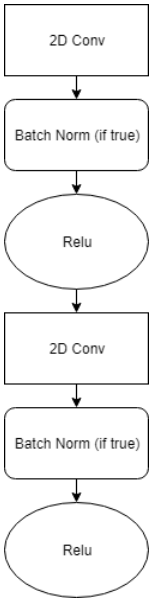
The labels are made for each image as follows

- Picasso = 1
- vanGogh = 0

Model

The model used was inspired by the VGG architecture. The model consists of repeated application of convolutions, each followed by a ReLU activation unit and a max pooling operation.

Below is the diagram for a single 2D convolution block.



2D Convolution Block

The architecture of my model follows below:

2D Conv Block (8 filters) → Max Pooling → Dropout → 2D Conv Block (16 filters) → Max Pooling → Dropout → 2D Conv Block (32 filters) → Max Pooling → Dropout → 2D Conv Block (16 filters) → Max Pooling → Dense layer (1024 units) → Dense layer (256 units) → Dense layer (1 unit)[sigmoid activation]

Training

The model was trained for 15 epochs using a batch size of 8. Due to data augmentation, each epoch takes longer than normal to complete. The training did not use K-fold cross validation but that is something I would like to have tried. More epochs can be tried, but I found that the model tends to overfit after 15 epochs. The trained model is used to predict on the test data.

Evaluation metrics and loss function

The evaluation metrics used is accuracy. The loss function used is binary cross entropy since we have binary classification model whose output is a probability value between 0 and 1.

Performance

Without Data augmentation:

- Train accuracy = 90.76388888888889 %
- Validation accuracy = 83.33333333333334 %
- Test accuracy = 84.75 %

With Data augmentation:

- Train accuracy = 91.875 %
- Validation accuracy = 83.33333333333334 %
- Test accuracy = 82.25 %

Experimentation

Following are the experiments on the model architecture that I tried with, or experimented with hyperparameters:

- Changing number of filters in each successive 2D Conv Block as follows:
    - 8 filters, 16 filters, 32 filters, 64 filters
    - 8 filters, 16 filters, 16 filters
    - 16 filters, 32 filters, 56 filters
    - 16 filters, 32 filters, 48 filter, 56 filters
- Using only 1 dense layer of 512 units and then a dense layer of 1 unit
- Increasing max pooling filter size to 3x3
- Using dropout rate greater than 0.10
- Adding gaussian noise as the only data augmentation technique

After trying all these experiments, I found that the model described previously gives the best results.