

Artist classifier

[Document subtitle]



April 19, 2019

SAi

vuppalsa@mail.uc.edu | phone:7348812859

**Description:**

The goal of this project is to train a model that can identify whether a work of art was created by Pablo Picasso or Vincent van Gogh.

**Details about Dataset:**

Given a dataset of 2000 images.

With 1000 images for each artist.

**Assignment Walkthrough:**

I used cloud to do this assignment,

So most of the cells at the beginning will setup the data.

I randomly taken 20% of images into test directory from test directory.

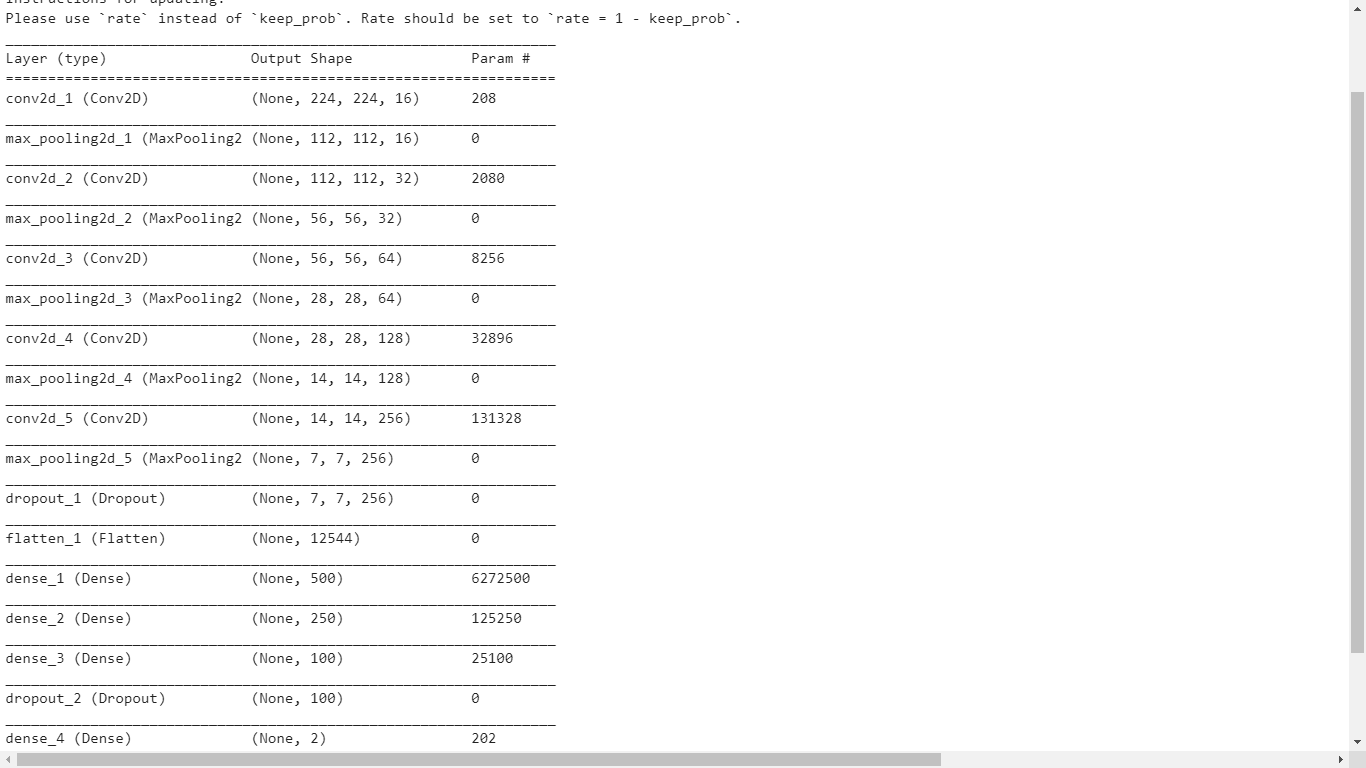
I randomly taken 10% of images into validation directory from test directory.

**Preprocessing data:**

* Firstly, I labeled all the data using np\_utils in keras
* I converted the image into 4d tensor to feed it into keras.
* Images come in different sizes, and should not be feed directly into network, So we resize that images into a square size(generally 224\*224 or 512\*512)
* we divide the tensor with 255(0-254 intensity levels) to normalize the range we input into network.

**Creating a CNN:**

**Below shown is my architecture:**



**Hyperparameters:**

**No of filters(**to increases number of nodes in convolutional layers**)**

I started with 16 and went up till 512,

Finally came back to 256.

**Size of the filters(**size of patterns to be detected**)**

2\*2 was kept constant

**Stride:**

Decides how convolutional filter moves on image horizontally/vertically at a time.

If Stride is 1 then then output will also have same height and width

,depth depends on no of filters we set.

If we increase strides then height and width decreases.

For convolution layers I took stride=1, for pooling I took 2 to reduce dimensionality.

**Padding-**

If filter is extending outside image we either can leave the pixels at corners.

(loss of information) –‘VALID’

Or

Padd with zeros so that filter covers all the pixels-‘SAME’.

Ive choose to keep padding to same so that covnet wont miss any regions of image.

**maxpooling**

since convolution layers produce filter stacks of large depths, the parameters used to compute this will go on increasing which causes to overfit.

So by using max pooling we will select a pooling filter where it takes feature maps and gives a weight which I max in that feature map.

Some of parameters:

* Poolsize:filter size
* Strides:same as pool size, in general we will take it as 2.
* Padding

**MY CONV NET:**

The convolution nets are useful to discover not only feature but also spatial patterns.

Our aim is to make the network learn features from spatial patterns.

At beginning the convolution layers of 16 filters will learn the object shapes etc

If we increase the depth of this layers and decreasing spatial dimensions they will start learning features like artist strokes, touches in our dataset.

To do that we need to increase depth of our network which can be achieved by adding a series of convolution layers with no of filters starting from 16 to 256.

We use maxpooling layers to curb overfitting and to reduce spatial dimensions and also decreases parameters.

My first layer I have taken 16 filters , with each size 0f 2\*2, strides=1

Activation as relu which is proved to be best in convolutional nets.

Followed by adding max pooling layer.

Increased depth till 256 where I got max test accuracy.

I tried 512 but the model overfits so I got back to 256.

I used dropout to curb overfitting, it took in such a way that there is 30 % prob that a node gets dropped out from training which forces other nodes to learn features.

Flattened the max pooling output followed by a series of dense layers to classify.

Used SoftMax to predict class.

After some experimenting I was able to achieve a **test accuracy of 80%.**

I am not satisfied with result, so I tried Transfer learning.

**Transfer Learning:**

There are architectures which were experimented and proved to be efficient convolutional architectures.

We can import that architectures along with their weight, which were trained on

Image net .

Image net is 1000 classes classification problem, whereas our dataset has only 2 classes.

We will pass our data set through this architecture, freezing weights and we will obtain features and we either flatten/pool this outputs and take it as input to train it on deep neural network to build a classifier.

I choose vgg19 pretrained on image net,

Chopped dense layers , passed data through it and got features.

Now I passed this features through globalavgpooling layer since there are nearly 3million parameters if we try to flatten it, and my cloud gpu cannot handle that so used global avg pooling which will decrease parameters by a lot.

After that added couple of dense layers and was able to get a **accuracy close to 89%.**

I expected more than that.

Even though inception is of high complexity and not needed for this small dataset, I gave it a shot.

But even it gave same accuracy.

After some experimenting and reading I realized this convolution layers are learning high level features of image net which makes that architectures to poorly perform on different dataset like ours.

So we need to only consider convolution nets with low level features like till ‘block4 pool’ so that they will not learn more high level features like detecting cars and other objects in ImageNet.

After chopping them off and training our data **I got a test accuracy of 93%**

As we know covnets suffer from translation invariance, and data augmentation can solve it.

I experimented with data augmentation on same model which gave me **a finale test accuracy of 95.22%.**

Note:

Thank you for taking the time to go through this document.