**INFO6105 15218 Data Sci Eng. Methods**

**SEC 02 Fall 2020**

**Final Project Report**

**Identity a Painting’s Author by Learning**

**from the Author’s Paintings**

**Instructor:** Dr. Liu Handan

**Team Number:** Team 2

**Team Members:**

Yi Ren 001050300 Fan Ji 001350642

Table of Contents

[Introduction 3](#_Toc58012078)

[Background: 3](#_Toc58012079)

[Motivation: 4](#_Toc58012080)

[Goal: 4](#_Toc58012081)

[Key Words: 5](#_Toc58012082)

[Methodology: 5](#_Toc58012083)

[Algorithms & Methodology: 5](#_Toc58012084)

[Problems and solutions: 7](#_Toc58012085)

[Import Packages: 7](#_Toc58012086)

[Visualization: 8](#_Toc58012087)

[Data Processing: 9](#_Toc58012088)

[Data Augmentation: 13](#_Toc58012089)

[Build Module: 15](#_Toc58012090)

[Train Model: 16](#_Toc58012091)

[Description of Dataset (Link): 18](#_Toc58012092)

[Results and Analysis: 20](#_Toc58012093)

[Batch\_size=16, epoch = 10 and epoch=5 20](#_Toc58012094)

[Result\_1: 20](#_Toc58012095)

[Result\_2: 21](#_Toc58012096)

[Result\_3: 21](#_Toc58012097)

[Result\_4: 21](#_Toc58012098)

[Result\_5: 22](#_Toc58012099)

[Result\_6: 23](#_Toc58012100)

[Result\_7: 23](#_Toc58012101)

[Batch\_size=32, epoch = 10 and epoch=25 23](#_Toc58012102)

[Result\_8: 24](#_Toc58012103)

[Result\_9: 24](#_Toc58012104)

[Result\_10: 24](#_Toc58012105)

[Result\_11: 25](#_Toc58012106)

[Result\_12: 26](#_Toc58012107)

[Result\_13: 26](#_Toc58012108)

[Result\_14: 26](#_Toc58012109)

[Conclusion: 28](#_Toc58012110)

[Reference: 29](#_Toc58012111)

# Introduction

## Background:

Artistic creation is one of the highest forms of expression of human thought and imagination. The ability to convey the imagination sets us apart from all others. Since the dawn of civilization, painting has been an expression of the visual language that has attracted and connected the brilliant human minds, from the earliest paintings on cave walls to modern paintings on paper or glass.

The study of the artistic style of art works is an interesting research activity. Its purpose is to make our computer program "perceive" the artistic picture submitted to her like a person, extract its "artistic style characteristics" from it, and assign this characteristic to another painting. So that this picture reflects the information of characteristic of itself and the artistic style. The current art style learning research is mostly limited to the "mechanical transfer" technology of pixel-by-pixel and sample-by-sample features, but there is not much research and discussion about identifying the paintings through learning artistic style.

Identifying the artist based on his or her artistic works is not that difficult for an enthusiast of arts. As you go from painting to painting, especially different artists, the discrepancies in style can be recognized, and trained art historians can catch even the most subtle of brush strokes and identify a certain artist or period. but most people find that it is sometimes confusing and difficult to identify famous painters by their art.

For a computer, this kind of analysis can be extremely difficult to undertake, but can a computer do the same? Can a machine without emotions identify who the genius is behind a mind-blowing painting? We travelled for a long time and finally we reached a stage where not only humans but computers, another brilliant creation of human minds, is identifying paintings.

This project focuses on 50 artists’ paintings art works, and according to its characteristics, it makes a deeper exploration of directly learning the artistic style characteristics of these paintings. The project firstly analyzes a large number of art works, and initially defines the local artistic characteristics of painting, that is, the technical features based on style, and simple classification of ink features. Then classic image preprocessing techniques, such as algorithm that image segmentation based on region growth, and skeleton extraction based on subdivided, so we can extract local data (graphic elements) of an art work Finally, a more mature machine learning model-neural network algorithm is introduced to learn the technical style characteristics of each part of the painting, so as to achieve the artistic style learning of the entire paintings.

Results show that the art style learning of painting based on neural network proposed in this project can effectively determine and learn the artistic style and identify the artist, it provides a new model of art style learning. It increases the accuracy of solution and improves the processing technology. The analysis of art also has positive significance for people who want to get to know the art in more detail without help of human being.

The artists we have been used in the project:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Vincent  van Gogh | Edgar Degas | Pablo  Picasso | Pierre-Auguste Renoir | Albrecht  Dürer | Paul  Gauguin |
| Francisco  Goya | Rembrandt | Alfred Sisley | Titian | Marc Chagall | Rene  Magritte |
| Amedeo Modigliani | Paul Klee | Henri  Matisse | Andy  Warhol | Mikhail  Vrubel | Sandro  Botticelli |

## Motivation:

With the development of economic and society, there is a huge increasing amount of people who started to strengthen their ability of art appreciation. As a result of this, there is a demand that people need a way to learn the story behind the painting, including the artist, the period and the emotion. It’s not difficult for an enthusiast of arts to identify these from a painting, but we want to let most people who don’t have this background still can get to know everything behind the painting. But how can we do this? Nowadays high technologies including Artificial Intelligence and Machine learning can help human beings to do most of the things that cannot be imagined in the past. We are trying to implement these technology and methodology to the art, which we expect the technology can partially and gradually replace human being in this field.

Artists have different styles that are unique to themselves. While some styles are much different from each other, others are subtle and hard to tell apart. We are curious whether computers can identify the artists based on particular paintings they created through learning the existing paintings. Our motivation is to develop an application in the future, which can be used in the gallery and museum to help people get to know the paintings in greater detail. When this app is used for business purposes, its cost is relatively low, and the price-performance ratio is high. It is expected that this app will be widespread used in the future for people who visit museum and gallery, and this invention has an immeasurable prospect.

## Goal:

Our goal here is not to identify the objects in a painting but to understand the overall style of the painting and establish the connection between style and artist and create a model that learn to identify the artist and analyze new pictures. We will use TensorFlow and ResNet50 as our main library and algorithm to analyze 50 artists and their paintings. We use TensorFlow to achieve the goal that identifying what an image represents, which is called *image classification*. An image classification model is trained to recognize various classes of images. TensorFlow provides optimized pre-trained models that we can learn the exiting painting with different style, different background and artists, then when given the new paintings, we can identify who is the artist for this painting. When this this project is implemented in the real life, it will help people without any artistic background can still appreciate and get to know the art.

Moreover, since the classification problem in this project is in a different category from the problems we did in assignments, we believe that this project will give us opportunities of exploring areas that are outside of coursework and deepening our understanding of the application of machine learning in computer vision.

Our dataset is obtained from a publicly available task on Kaggle, where the creator of the dataset scrapes images of artworks from [artchallenge.ru](http://artchallenge.ru/?lang=en) . We apply ResNet50 model, where we mainly focus on training shallow layers rather than deep layers. Eventually, from the observation, we conclude that shallow layers in deep learning models are mainly responsible for understanding the overall style of paintings.

## Key Words:

Identity, Machine Learning, Deep supervised learning, Convolutional Neural Network, Image preprocessing, ResNet50, TensorFlow.

# Methodology:

## Algorithms & Methodology:

In this project, we use convolutional neural network-based approach, with a pre-defined architecture as baseline and ResNet50 to achieve the CNN. Why we choose to use the Keras in TensorFlow: Keras in TensorFlow has a lot of advantage and disadvantage. But as a beginner it is strong enough to finish the project. Keras is a more advanced and user-friendly API with a configurable backend, written and maintained by Francis Chollet, a member of the Google Brain team.

Advantage of Keras in TensorFlow:

1. Provide high-level APIs to build deep learning models to make them easy to read and use.
2. Write standardized documents.
3. Large, active community.
4. On top of other deep learning libraries (such as Theano and TensorFlow, configurable).
5. Object-oriented design is used, so everything is treated as an object (such as network layers, parameters, optimizers, etc.). All model parameters can be accessed as object properties.

Disadvantage of Keras in TensorFlow:

1. Due to its very common use, it lacks in performance.
2. There are performance issues when used with the TensorFlow backend (because it is not optimized for it), but it works well when used with the Theano backend.
3. Not as flexible as TensorFlow or PyTorch.

Why we use ResNet50 to as a model train the data: Image classification refers to the classification of a given picture into one of several predefined categories. The traditional process of image classification involves two modules: feature extraction and classification.

As the network depth increases, the accuracy of the network should increase simultaneously. Of course, attention should be paid to the problem of overfitting. However, one problem with the increase in network depth is that these added layers are signals for parameter update, because the gradient propagates from back to front. After increasing the network depth, the gradient of the higher layer will be small. This means that the learning of these layers basically stagnates, which is the vanishing gradient problem. The second problem of the deep network is training. When the network is deeper, it means the parameter space size, and the optimization problem becomes more difficult. Therefore, simply increasing the network depth will cause higher training errors. Residual network ResNet designed a residual module so that we can train a deeper network.

The training problem of the deep network is called the degradation problem. The logic behind the residual unit can solve the degradation problem is this: imagine a network A with a training error of x. Now build network B by stacking more layers on top of A. These new layers do nothing, just copy the output of the previous A. These new layers are called C. This means that network B should have the same training error as A. Then, if training network B, its training error should not be worse than A. But in fact, it is worse. The only reason is that it is not easy for the added layer C to learn the identity mapping. In order to solve this degradation problem, the residual module establishes a direct connection between input and output, so that the newly added layer C only needs to learn new features on the basis of the original input layer, that is, learning residuals, which is easier.

Advantage of ResNet50:

Unlike traditional sequential network architectures (such as AlexNet, OverFeat, and VGG), it adds a y=x layer (identity mapping layer), which allows the network to increase in depth without degrading. The following figure shows a build block. The input passes through two weight layers, and finally is added to the input to form a micro-architecture module. ResNet ultimately consists of many micro-architecture modules.

From the observation, we come up with hypothesis that relates the performance of models to the number of layers and the nature of architecture. Also, we conclude that shallow layers in deep learning models are mainly responsible for understanding the overall style of paintings.

We first split the raw dataset into training set and validation set by a ratio of 4:1. The x coordinate is expanded five times while keeping the y coordinate unchanged copy of every painting was then added into the dataset. After data augmentation, the dataset contains 5576 paintings in total. Paintings in both sets are grouped under different subfolders based on their creators and labelled properly. Image pixels were normalized, and all images were cropped to a 224x224 resolution by performing random crops. Models we used in our work were fine-tuned ResNet50.

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers---8x deeper than VGG nets but still having lower complexity.

An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers.

Diagram

Description automatically generated

The objective is to identify artist and not objects in the images. So, the model should understand the style of the image better rather than the final output. Hence, training of shallow layers is more important than the deeper layers.

The above statement is based on my understanding of the problem and experiments and observations.

Training the model for more iterations might improve the performance, at the cost of computation resource.

## Problems and solutions:

### Import Packages:

In this project, we use the NumPy and pandas to process the data, and the time to be a timer to calculate the processing time. We use the metrics from the Sklearn to generate the Confusion Metric, and the seaborn to draw some diagram to make the data visualized. We also use the os to make sure to path of the images and use json to link the different paths. We import the pyplot to print some picture about the analyzing results. We use the random to print some random images as test.

TensorFlow is a good tool for us to use for machine learning and deep learning. Keras is a deep learning library based on TensorFlow and Theano (a machine learning framework developed by the University of Montreal, Canada). It is a high-level neural network API written in pure python and only supports python development. It is a re-encapsulation of TensorFlow or Theano in order to support rapid practice, so that we can quickly convert ideas into results without paying too much attention to the underlying details. We also use many Keras package to achieve the machine learning part.



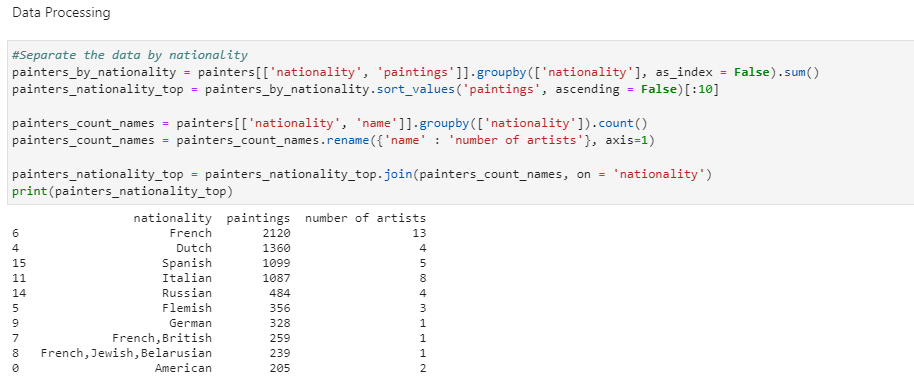
### Visualization:

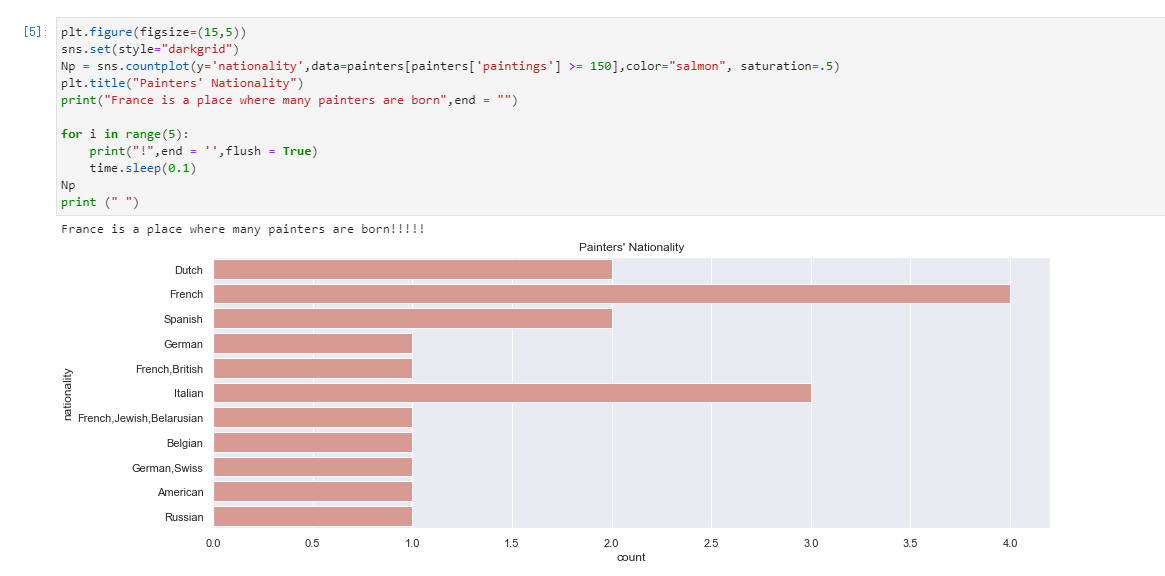
After importing the packages, we make the data visualization. At first, we want to find out how many documents in the data, so we list the directory of the data, and choose the painters whose paintings are more than 150. We just need to use the two factors of the data which are ‘name’ and ‘paintings’, the shape of the original data and the used data are (50,8) and (18,2) respectively.



### Data Processing:

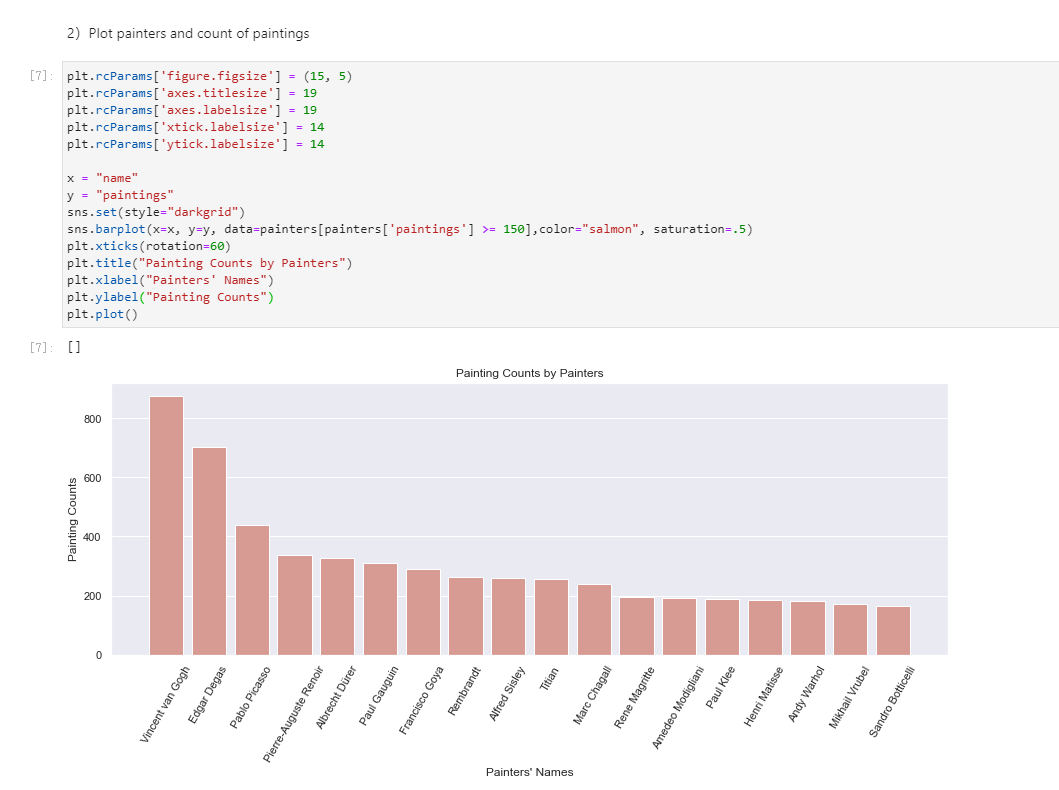
We want to find out the nationality of the painters and which country has the most world-class painters. So, we separate the data by the factor ‘nationality’ and print a graph for it.





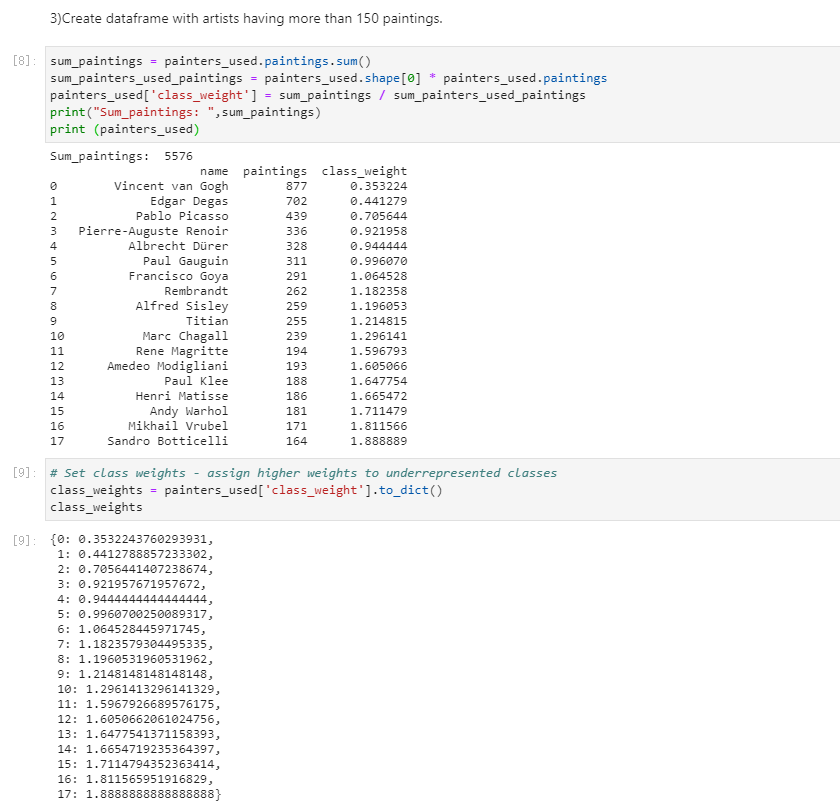
As we can see that no matter the original data or the used data that France has the most world-class painters. It can be seen that, at that time, France was the world's leading country in artistic attainments, and its economic and cultural development was amazing and far ahead.

Then we counted the number of paintings of each artist and made a chart. As we can see that Van Gosh in this data has the most paintings and Botticelli has the least paintings.



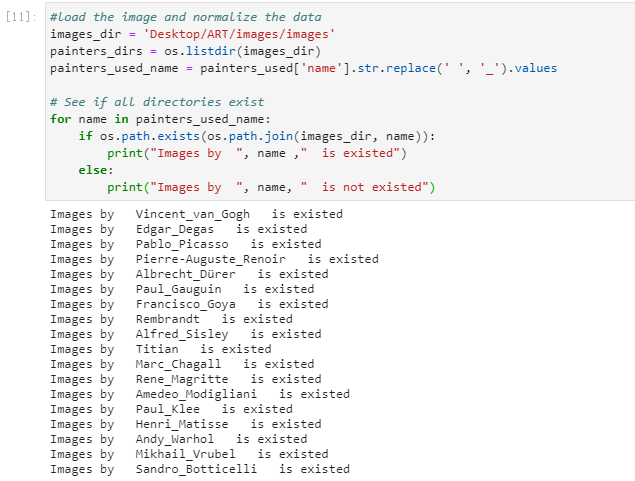
The topic of our project is to identify the author of the world famous painting by learning the author’s paintings, so the first problem we face is to select the appropriate research object from the 50 painters given in the data set as our learning goal, because the data set Given that the number of paintings of 50 painters is different and they are very different.

In order to reduce the problem of poor learning results due to the data set, our group adopted two methods. The first is to select paintings from 50 painters. More than 150 paintings are used as research objects, the second is to use class\_weights, which can add a weight to each category in the training set. If the number of samples in the category is large, then its weight is low, otherwise the weight is high. This is our class\_weights of the different painters.

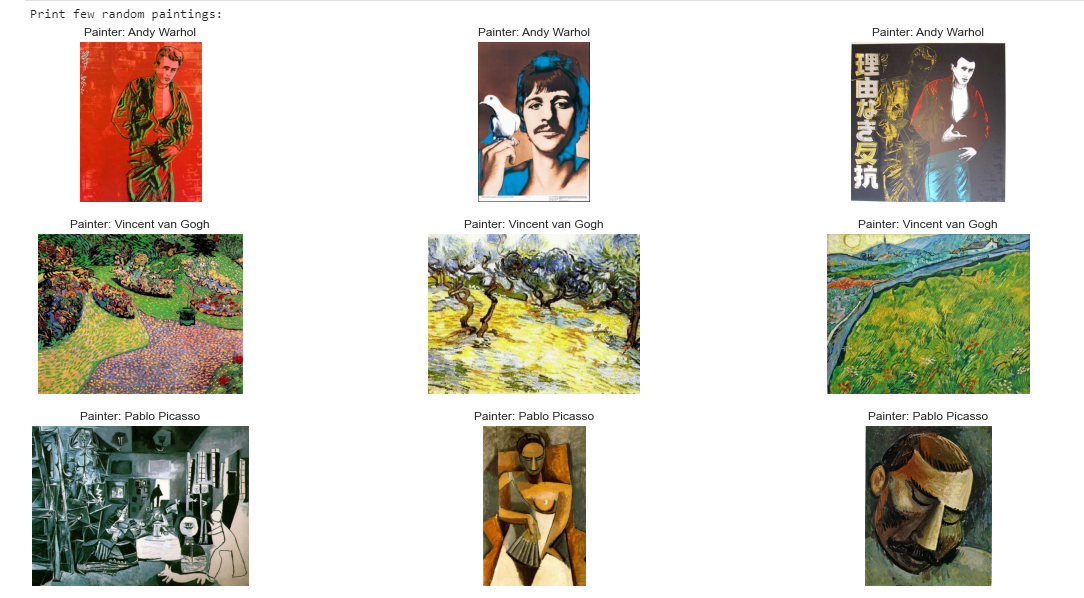


When we verify our paths of image whether are available, we find out that the fourth painter’s name has some problem to be identified so we changed his name format and then can be identified.

And we define a function to print some random pictures to verify our picture is available.



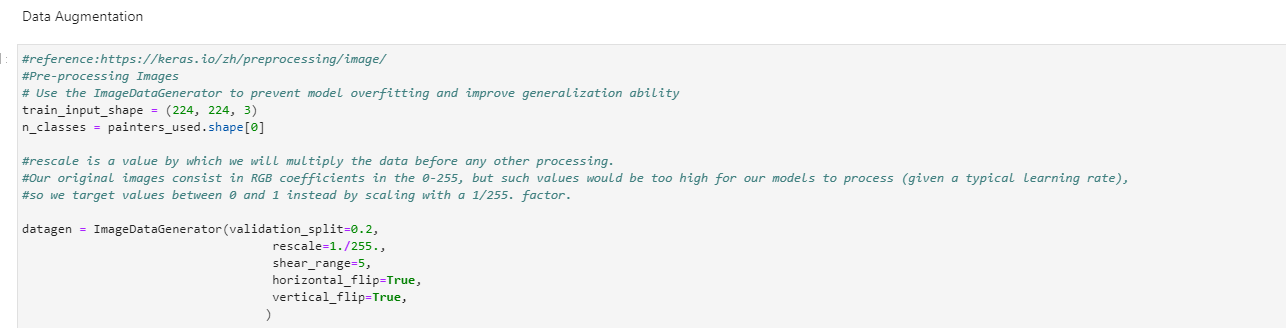




### Data Augmentation:

At the same time, we are also facing problems such as uneven painting size, uneven proportions and pixels.

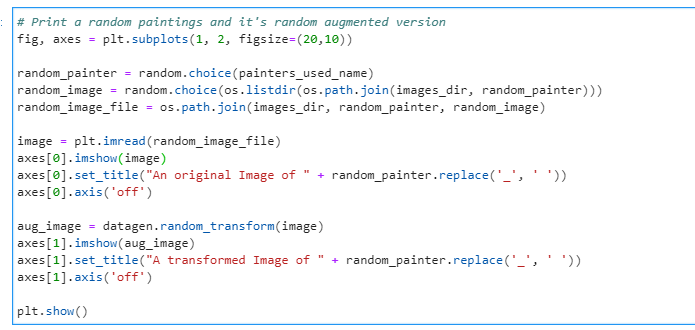
We use image normalization processing, use ImageDataGenerator in Keras in TensorFlow to process, divide all given images into training set and prediction set, divide it at the ratio of 80% training set and 20 prediction set, and draw The pixels of is set to (224,224), and the parameters of the training set and prediction set are adjusted accordingly.

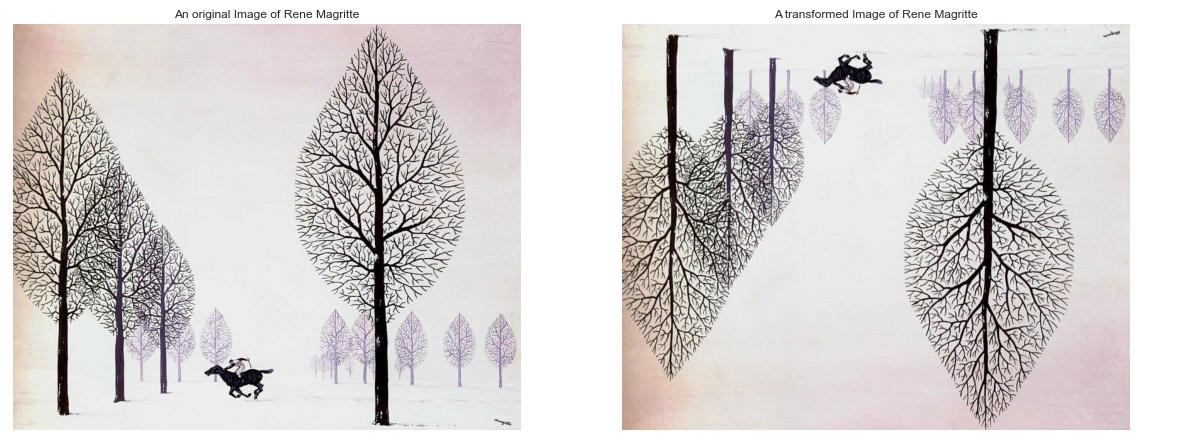


Set the train datagen and the valid datagen to normalize the data and make it work well in the model by the way the batch size is 32: As we can see the step size of train is 139, and the step size of valid is 34.



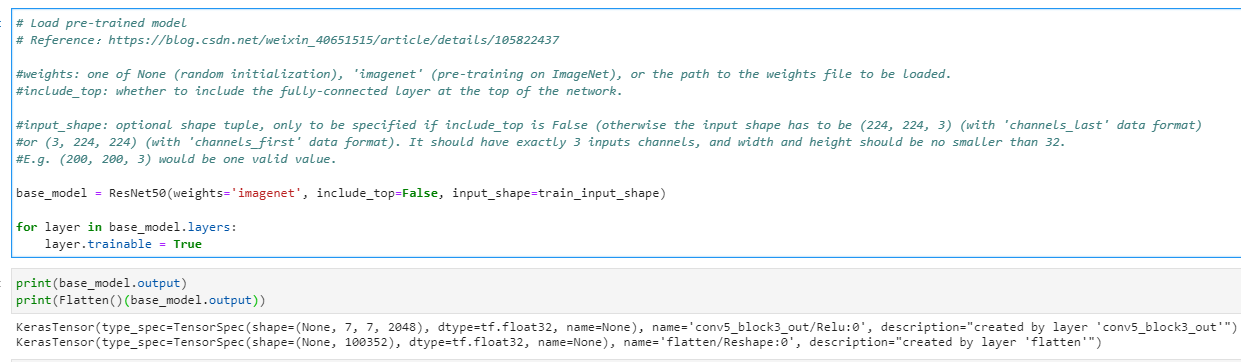
After data augmentation I try to make sure the augmentation is successful, so we print an original picture and a random picture.



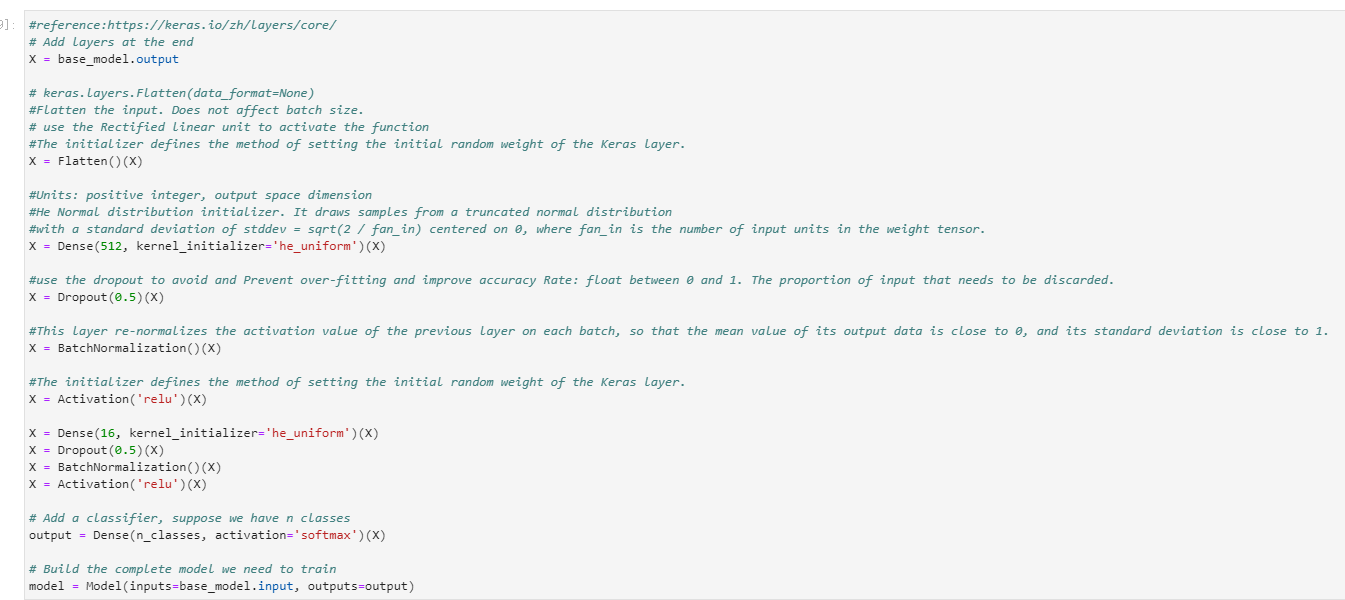


### Build Module:

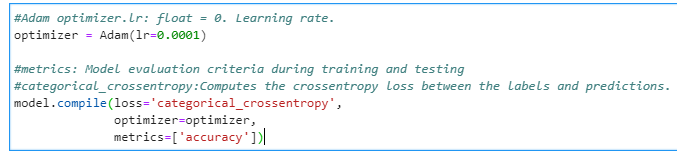
We use the ResNet as our base model and train the data twice using the all layers is on and first fifty layers is on respectively.



Recall the ResNet model according to our data, set the output and input and the initializer defines the method of setting the initial random weight of the Keras layer. Use the dropout to avoid and Prevent over-fitting and improve accuracy Rate: float between 0 and 1. The proportion of input that needs to be discarded. This layer re-normalizes the activation value of the previous layer on each batch, so that the mean value of its output data is close to 0, and its standard deviation is close to 1. The initializer defines the method of setting the initial random weight of the Keras layer. Add a classifier, suppose we have n classes. Build the complete model we need to train.



Adam optimizer.lr: float = 0. Learning rate. metrics: Model evaluation criteria during training and testing categorical\_crossentropy: Computes the crossentropy loss between the labels and predictions.

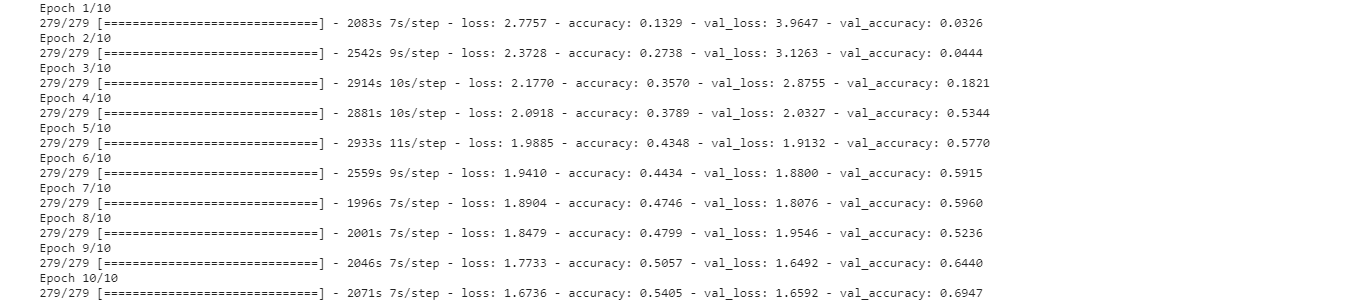


### Train Model:

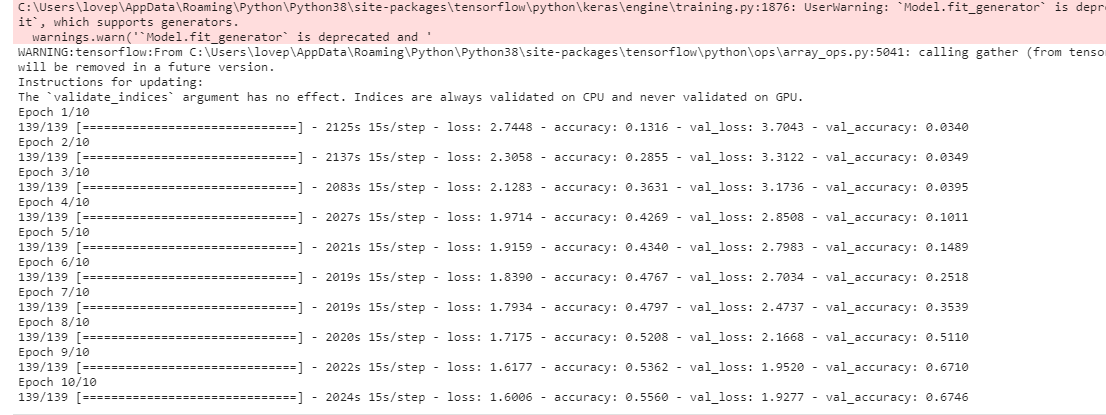
We use two methods to train the model. At first use all the layers turn on to train the model and then use the first fifty layers to train the model then add the results together as the final results. In order to improve the accuracy and to find out the influence of the n\_epoch and the batch\_size of the train result. We trained twice of both two method.



The first train results of all layers: Time: 6.674h



The second train results of the all layers: Time: 5.694h



The first train results of the first fifty layers: Time: 1.982h

A picture containing table

Description automatically generated

The second train results of the first fifty layers: Time: 10.748h

Table

Description automatically generated

# Description of Dataset (Link):

<https://www.kaggle.com/ikarus777/best-artworks-of-all-time>

We use the dataset Best Artworks of All Time create by Icaro, who gathered a collection of artworks of the 50 most influential artists of all time. The data was scraped from [artchallenge.ru](http://artchallenge.ru/?lang=en) during the end of February 2019. All download images are low resolution copies of the original artwork and are unsuitable for commercial use and follow the copyright term. We added a dataset with basic information retrieved from Wikipedia. We planned to create a convolutional neural network to identify the artworks.

We choose this dataset because it covers all the attributes, we need including artist information and images, we don’t have to do much extra data processing to improve the model. And the data size is suitable for our model, it will improve the running speed and the accuracy of result. In addition, the neural networks algorithms will probably be sensitive to the scale and distribution of your numerical input variables, as well as the presence of irrelevant and redundant variables.

This dataset contains three files:

1. artists.csv: dataset of information for each artist, attributes include id, name, years, genre, nationality, bio, Wikipedia and paintings.
2. images.zip: collection of images (full size), divided in folders and sequentially numbered
3. resized.zip: same collection but images have been resized and extracted from folder structure

After finishing collecting all data from the website, we conduct data cleaning, missing labels completing and easy accessing on the dataset for three main purpose : processing limiting the dataset size and thus accelerating the model learning process, removing noise from the data and thus improving the model predictive capabilities and making the data interpretation easier by humans.

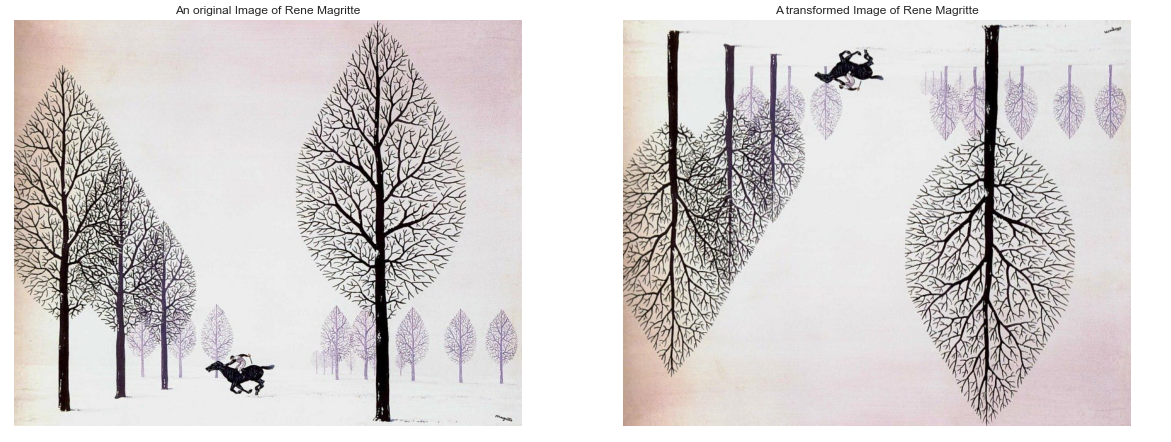
In the data cleaning process, we identified and corrected mistakes or errors in the data. We found some problem There is some problem recognizing 'Albrecht\_Dürer’, So we'll update this string as directory name:

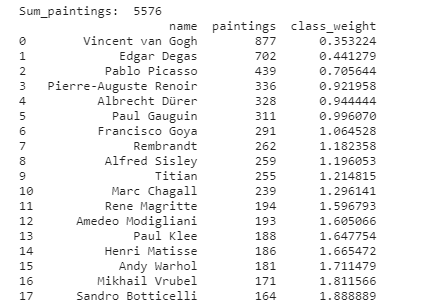
updated\_name = "Albrecht\_Dürer”. replace ("\_", " ")

artists\_top. iloc[4, 0] = updated\_name

For feature selection, we identified those input variables that are most relevant to the task. So, in the artist table, we choose to select all the attributes they have including include id, name, years, genre, nationality, bio and paintings. We don’t do any data transforms which changes the scale or distribution of variables nor any dimensionality reduction. But we are adding new variables into the existing data, which is Wikipedia, this will add more detail for the painting and will help for further analyzing and predicting.

To further look at the data we found that there are paintings of 50 artists in the dataset. However only 18 artists have more than 150 paintings available here. To reduce computation and better training, we decided to use the paintings of these 18 artists only. Since this is an imbalanced dataset (Van Gogh has 877 paintings whereas Sandro Botticelli has only 164), class\_weights is important. Infect, it improved model performance substantially. We used Keras ImageDataGenerator for data augmentation. This is not a traditional object detection problem; hence the augmentation approach should be used very carefully. After further researching and different testing, we conclude that far zoom range worked well.





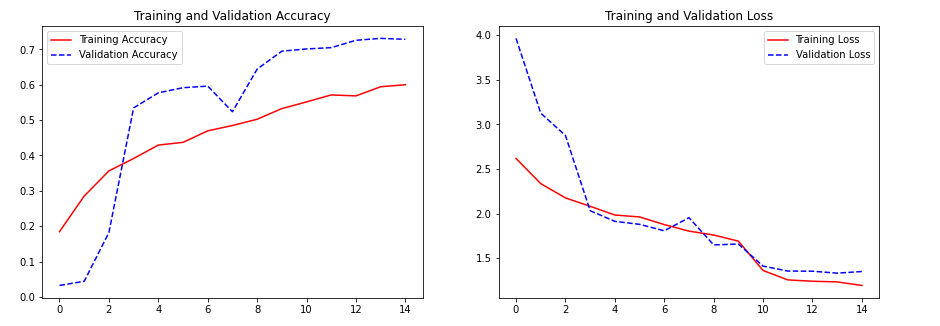
Make sure all the data(images) can be used successfully. And we use the ImageDataGenerator to prevent model overfitting and improve generalization ability.



# Results and Analysis:

We define a function to print a training graph and merge the two training results as a history result.

## Batch\_size=16, epoch = 10 and epoch=5



### Result\_1:

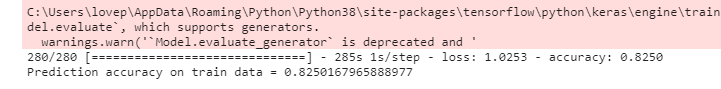
In this picture we can see that the training and the validation accuracy trend is linear rise, and there is a little backlash, which means that the training is not over-fitting and proves the effectiveness of the model.

### Result\_2:

In this picture we can see that the training and the validation loss trend is linear decline, and the validation loss changes significantly, and the changes are large, indicating that as the training deepens, the accuracy of the final prediction set is also rising.

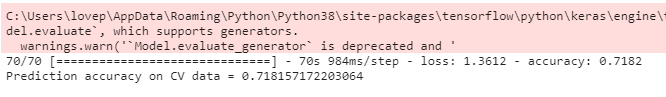
### Result\_3:

We use evaluate function to evaluate the accuracy of the training\_generator and the result is *82.5%*, it is a not bad result for us.



### Result\_4:

We use evaluate function to evaluate the accuracy of the valid\_generator and the result is *71.8%*, it is not as good as the training\_generator.



I think there are some reasons can cause this result:

The validation data set is less than the training data set so there is not enough for the model to learn the validation data. The data split of the dataset may be adjusted for better result.

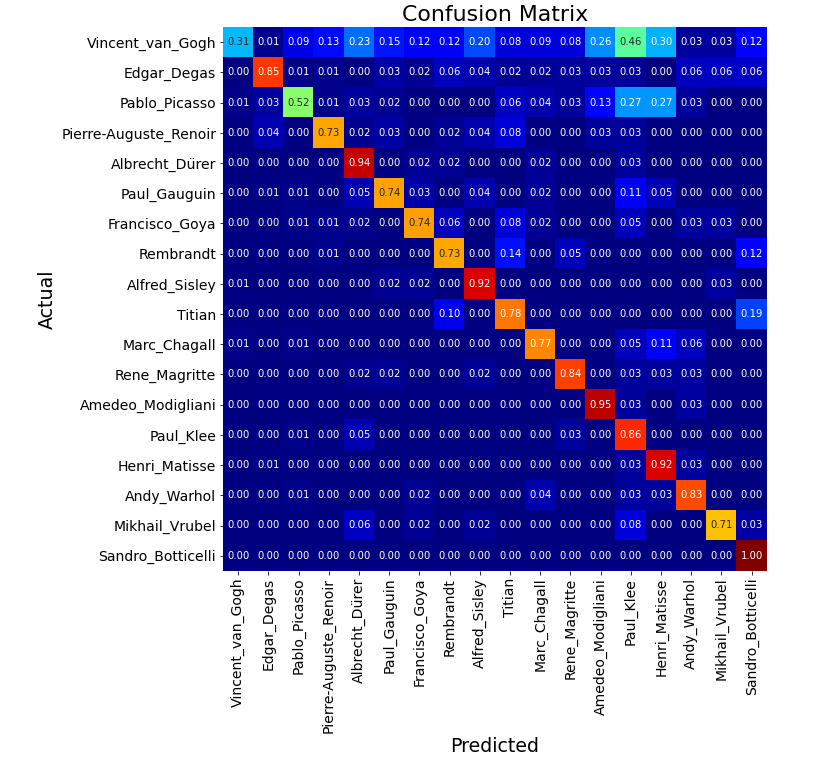
We print the confusion matrix of the validation dataset use the recall rate.

Recall attempts to answer the following question: The probability of being predicted as a positive sample in a sample that is actually positive.

What proportion of actual positives was identified correctly?

Mathematically, recall is defined as follows:



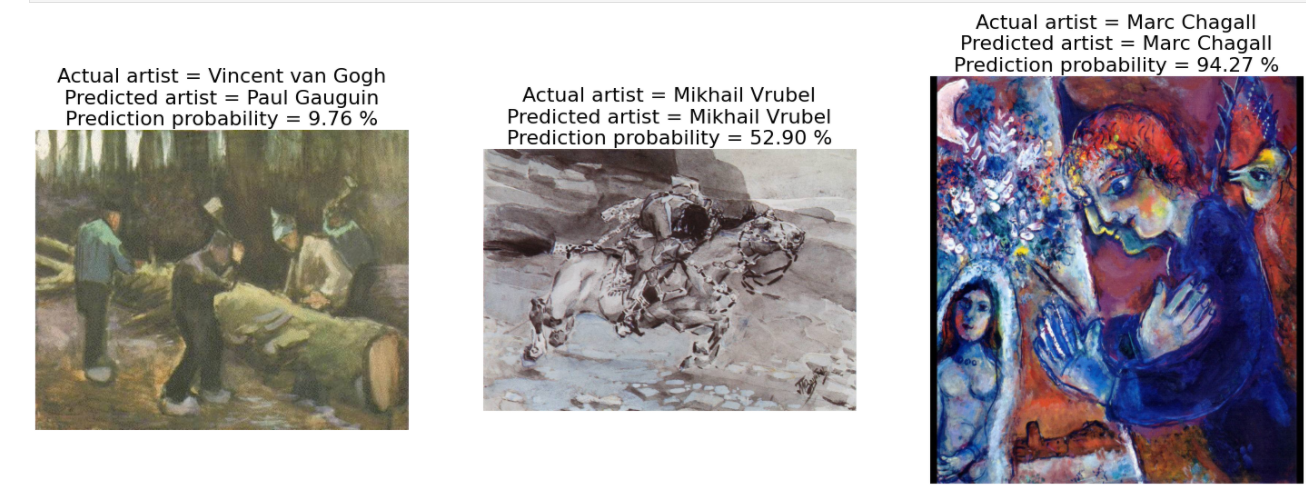


### Result\_5:

In this confusion matrix we can see that Sandro\_botticelli’s image has the best prediction result and Van Gosh has the worst prediction result. I think there are three reasons: The first reason is that Van Gogh doesn’t have a uniform style, and each painting has no fixed pheromone, and is different. The second reason is that Van gosh has the most images but after use the class\_weights, the model learns not as many as other painters’ paintings cause the recall rate is low. The third reason is a little dramatic that may be Van Gogh is not one person, may be a group of people use the Van Gogh as a fake name to paint paintings.

### Result\_6:

In this matrix we can see that Van Gogh and Picasso’s recall rate is very low compared to others. And the value of the two painters is higher than others. We have a bold assumption that the lower the recall rate the higher the paintings value.



### Result\_7:

As we can see in the results of the prediction that the first image painted by Van Gogh is wrong but the other two is right and has relatively high accuracy. There are three reasons, the first one is that there are too few learning samples. The second is that there is a problem with the training model. The third is that the paintings of these two painters are very similar. I think our model has no problems so the other reasons may cause this result.

## Batch\_size=32, epoch = 10 and epoch=25

Graphical user interface, chart, line chart

Description automatically generated

### Result\_8:

With batch\_size=32, epoch =10 and epoch =25 we can see that the line of the accuracy of validation is smoother and after 6 epochs the line is closer which indicates as the epoch increases, the training effect of the model is better. The loss line is different from the line with batch\_size=16, epoch =5 and epoch =10, because the different batch\_size cause the validation loss is always higher than the training loss.

### Result\_9:

We use evaluate function to evaluate the accuracy of the training\_generator and the result is *86.6%*, it is better than the last training rate.

Graphical user interface, text

Description automatically generated

### Result\_10:

We use evaluate function to evaluate the accuracy of the valid\_generator and the result is *73.0%*, it is better than the last validation rate.

Text

Description automatically generated

I think the reason why the accuracies of training\_generator and valid\_generator are both improved because that the epoch is more than before. So, we can conclude that the more epochs the higher accuracy.

Graphical user interface, chart

Description automatically generated

### Result\_11:

The confusion matrix’s overall trend has not changed, and the specific values ​​are slightly different. Alfred\_Sisley’s image has the best prediction result and Van Gosh has the worst prediction result. I think this is the inevitable result of more training epochs, but the overall trend has not changed to prove that our model is correct. In this matrix we also can see that Van Gogh and Picasso’s recall rate is very low compared to others. And the value of the two painters is higher than others. We have a bold assumption that the lower the recall rate the higher the paintings value.

A picture containing text, picture frame

Description automatically generated

### Result\_12:

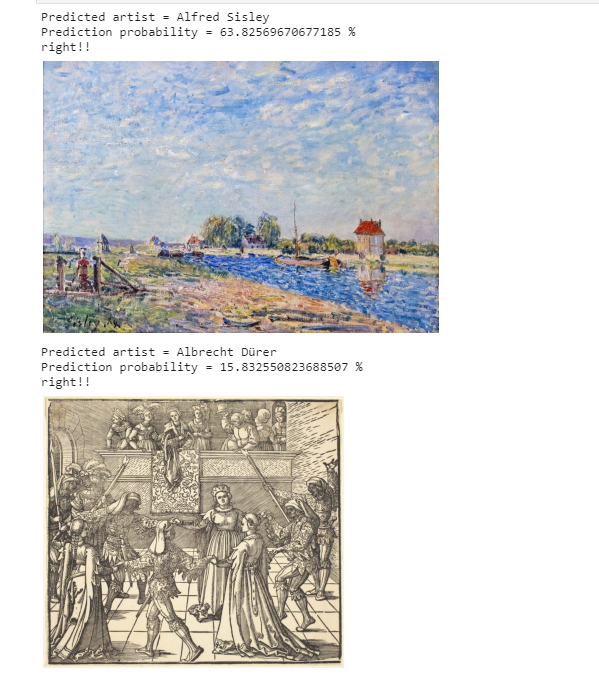
As we can see in the results of the prediction that the second image painted by Pablo Picasso is wrong but the other two is right and has relatively high accuracy. There are two reasons, the first one is that there are too few learning samples. The second is that there is a problem with the training model. I think our model has no problems so the other reasons may cause this result.

### Result\_13:

Van Gogh and Picasso is great painters in the world, but Van Gogh is the best one.

### Result\_14:

In addition to using the given test set to test the prediction results, we also use some online image resources for testing, and the selected online images have correct prediction results, so our model and learning method are generally Correct and valuable.



# Conclusion:

Our group has achieved all the preset functions and goals, as well as the corresponding algorithms, and through 8.5 hours of model learning one time and another 16.5 hours of model learning, the final machine model can master the feature extraction and classification of the corresponding pictures, and not only can give accurate predictions on the data in a data set, you can also make predictions on pictures on the Internet and get relatively ideal results.

In this project, we used a larger data set as the research object. It was also the first time that our group tried image-related machine learning and deep learning, applied data visualization, and learned how to augment images and how to normalize them. Analyze data, and study and understand the tuning and application of resnet50 in the keras package, as well as unsupervised learning related content.

In our project, we used a lot of knowledge that we had never understood and seen before, but after our in-depth research, we got the results we wanted. I am also very grateful for the opportunity of this project, which allowed me to learn about images in machine learning. The principle of processing and analysis, the following are the corresponding conclusions obtained by our group:

1. We use Resnet as our basic model. This method is correct.
2. We used the control variable method to study the factors that affect the accuracy and computing time of the model and found that the size of batch\_size can affect the computing time of the model. The larger the batch\_size, the less computing time (based on CPU), and the more epochs, the more accurate the model prediction. The higher the degree.
3. Van Gogh and Picasso’s paintings have very few similarities. Therefore, the results obtained by studying their paintings and predicting paintings are relatively inaccurate. This shows that each painting of the two painters is different, and even there are big differences. Especially Van Gogh's.
4. At the same time, we found that Van Gogh and Picasso’s paintings are generally more valuable than other painters. If the internal reason is explained by data science, it may be that each of their paintings has different pheromones, which also shows that their painting ideas are changeable. Make the machine unable to extract information and also unable to classify and learn
5. Van Gogh and Picasso is great painters in the world, but Van Gogh is the best one.

# Reference:

[1] https://www.cnblogs.com/bonelee/p/8972139.html

[2] https://keras.io/zh/initializers/

[3] https://www.cnblogs.com/wj-1314/articles/9621901.html

[4] https://keras.io/zh/activations/

[5] https://keras.io/zh/layers/core/

[6] https://keras-cn.readthedocs.io/en/latest/layers/normalization\_layer/

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions, 2014.

[8] Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1. IEEE, 539–546.

[9] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. Machine learning 20, 3 (1995), 273–297.

[10] E. J. Crowley and A. Zisserman. 2014. In Search of Art. In Workshop on Computer Vision for Art Analysis, ECCV.

[11] E. J. Crowley and A. Zisserman. 2016. The Art of Detection. In Workshop on Computer Vision for Art Analysis, ECCV.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 248–255.

[13] Bert Dodson. 1990. Keys to drawing. North Light Books.

[14] Leon Gatys, Alexander S Ecker, and Matthias Bethge. 2015. Texture synthesis using convolutional neural networks. In Advances in Neural Information Processing Systems. 262–270.

[15] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2414–2423.

[16] Elad Hoffer and Nir Ailon. 2015. Deep metric learning using triplet network. In International Workshop on Similarity-Based Pattern Recognition. Springer, 84–92.

[17] James M Hughes, Daniel J Graham, and Daniel N Rockmore. 2010. Quantification of artistic style through sparse coding analysis in the drawings of Pieter Bruegel the Elder. Proceedings of the National Academy of Sciences 107, 4 (2010), 1279– 1283.

[18] C Richard Johnson, Ella Hendriks, Igor J Berezhnoy, Eugene Brevdo, Shannon M Hughes, Ingrid Daubechies, Jia Li, Eric Postma, and James Z Wang. 2008. Image processing for artist identification. IEEE Signal Processing Magazine 25, 4 (2008).

[19] Fahad Shahbaz Khan, Shida Beigpour, Joost van de Weijer, and Michael Felsberg. 2014. Painting-91: a large scale database for computational painting categorization. Machine vision and applications 25, 6 (2014), 1385–1397.

[20] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. Neural computation 1, 4 (1989), 541–551.