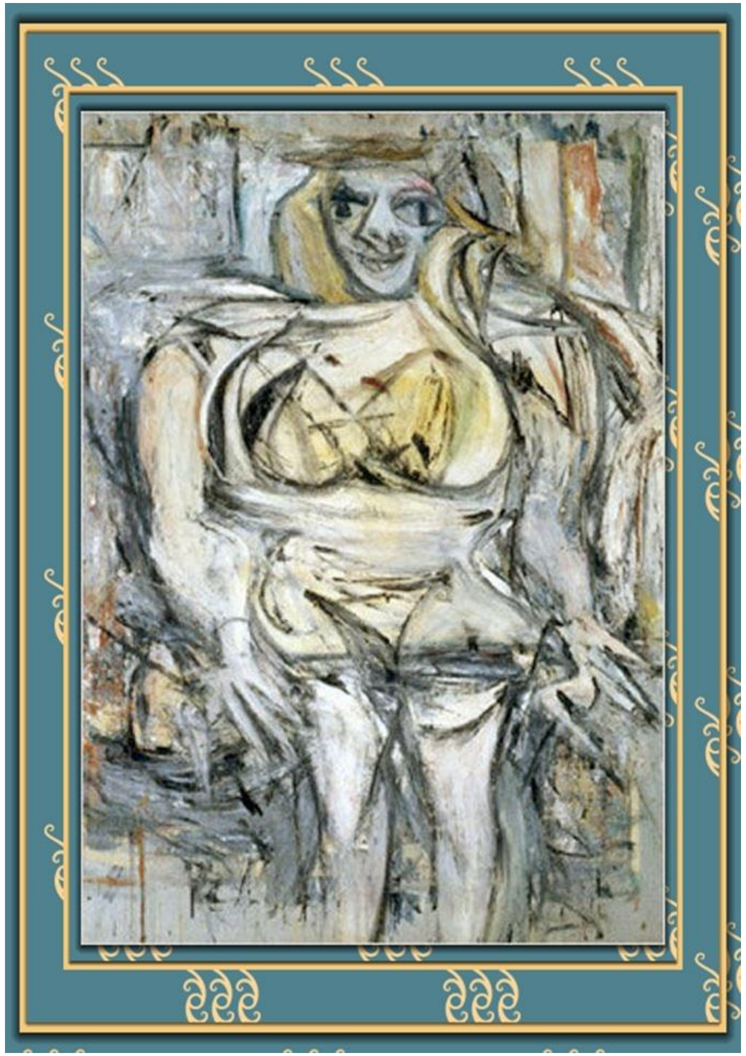
Identify Artist from Art

Perceptual Computing Final Pre:
PRML Problems Challenge

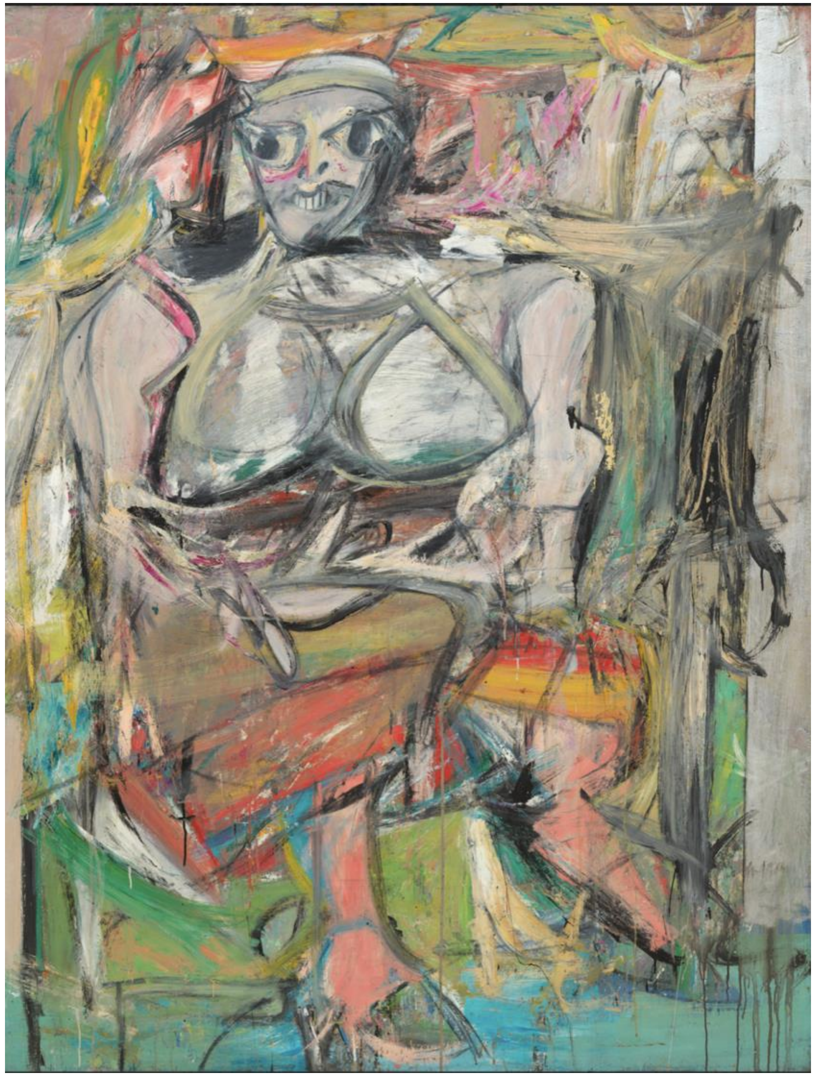
Content



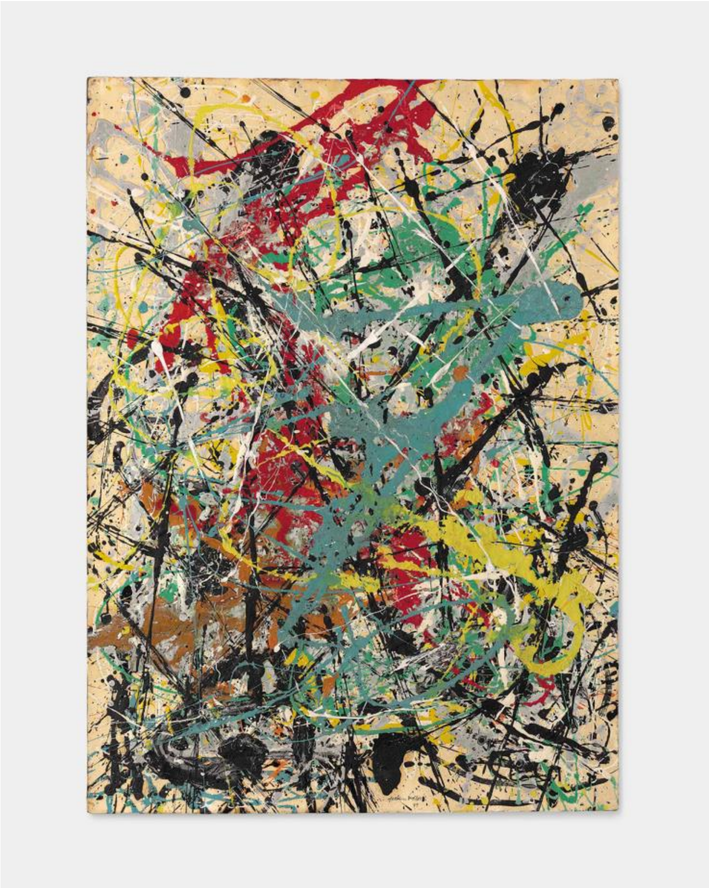
No.5, 1948 by Jackson Pollock
165 million dollar

Woman III by Willem de Kooning
162 million dollar

What do you think about these paintings？



Woman I by Willem de Kooning



No.16, 1949 by Jackson Pollock

32.7 million dollar

What do you think about these paintings？

MARKETSINSIDER WATCHLIST

MARKETS STOCKS INDICES COMMODITIES CRYPTOCURRENCIES CURRENCIES ETFS NEWS BI PR

A hedge fund-backed art dealer just lost an $11 million ruling to Sotheby's over an allegedly fake painting

Forged pieces of art or **fake** paintings are becoming an **increasing** problem for museums and **art galleries** around the world.

It is **embarrassing** to find out that one of the **masterpieces** you show in an **exhibition** is a **fake**.

Such **awareness** can be expensive as well.

A British museum paid £440,000 (about $700,000) for a **forged** Egyptian **statue** in 2003.

What about if they bought the painting in 1893 ?

Instead of subjecting works to lengthy and hugely expensive materials analysis, hoping a forger has made a tiny mistake.

Using Neural Network technique is so powerful that it doesn’t even need access to the original work: A digital photograph will do.

Every single gesture – shape, curvature, the velocity with which a brush-or pencil-stroke is applied – reveals something about the artist who made it. Those features are part of the artist`s style.

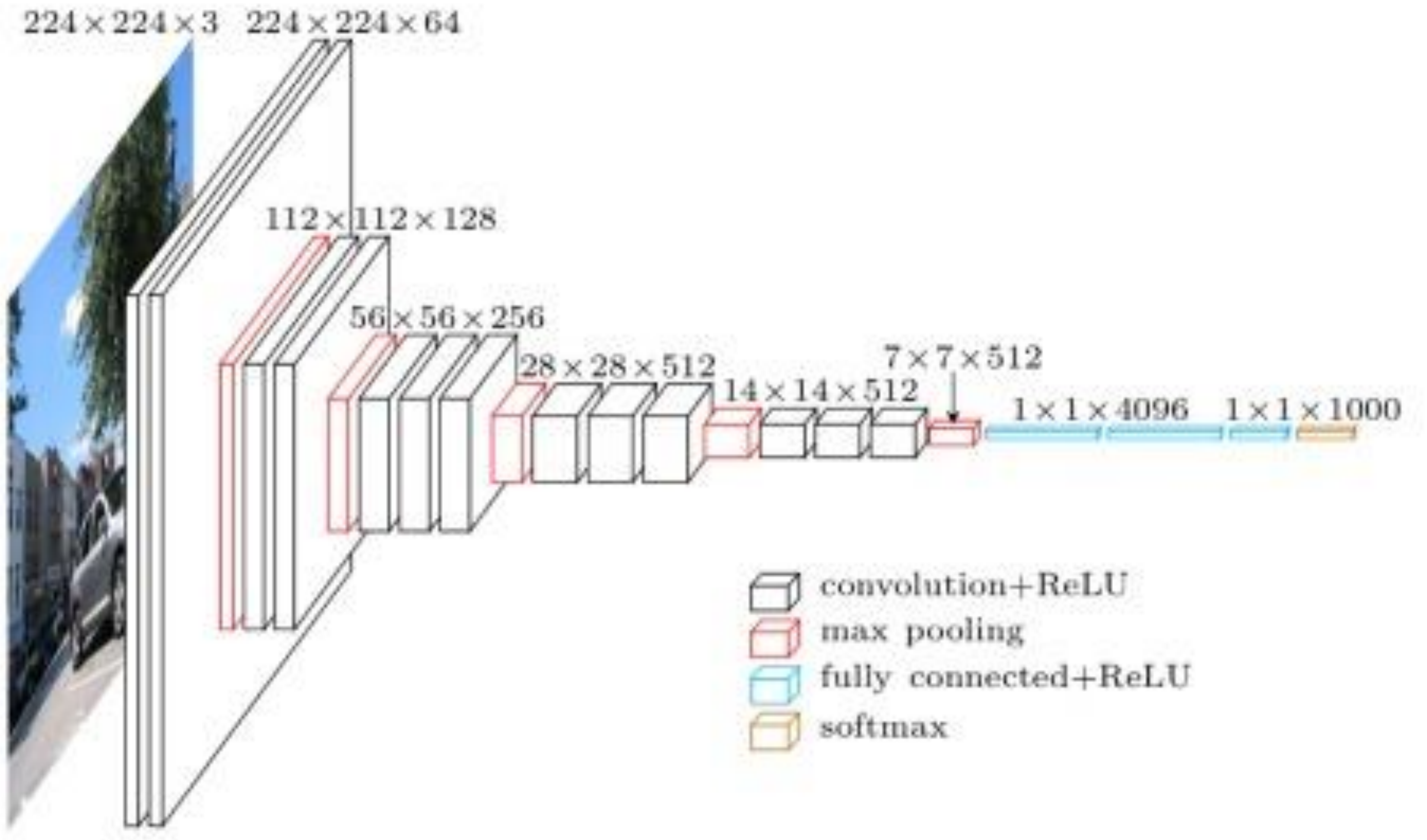
Related Work

- The earliest work in this field to **learn different painting styles** through deep neural networks was the Bethge Laboratory at the University of Tubingen, Germany, which was mainly studied by Gatys L A, Ecker A S, Bethge M. these three researchers.[1]
- In early 2016, Russian computer engineers developed the APP called "Prisma", which successfully commercialized the technology for the first time. Speeding each photo's processing time to only 20 seconds.

[1] Gatys L A, Ecker A S, Bethge M. A Neural Algorithm of Artistic Style[J]. Computer Science, 2015.

- [1] Gatys L A, Ecker A S, Bethge M. A Neural Algorithm of Artistic Style[J]. Computer Science, 2015.

VGG-Network

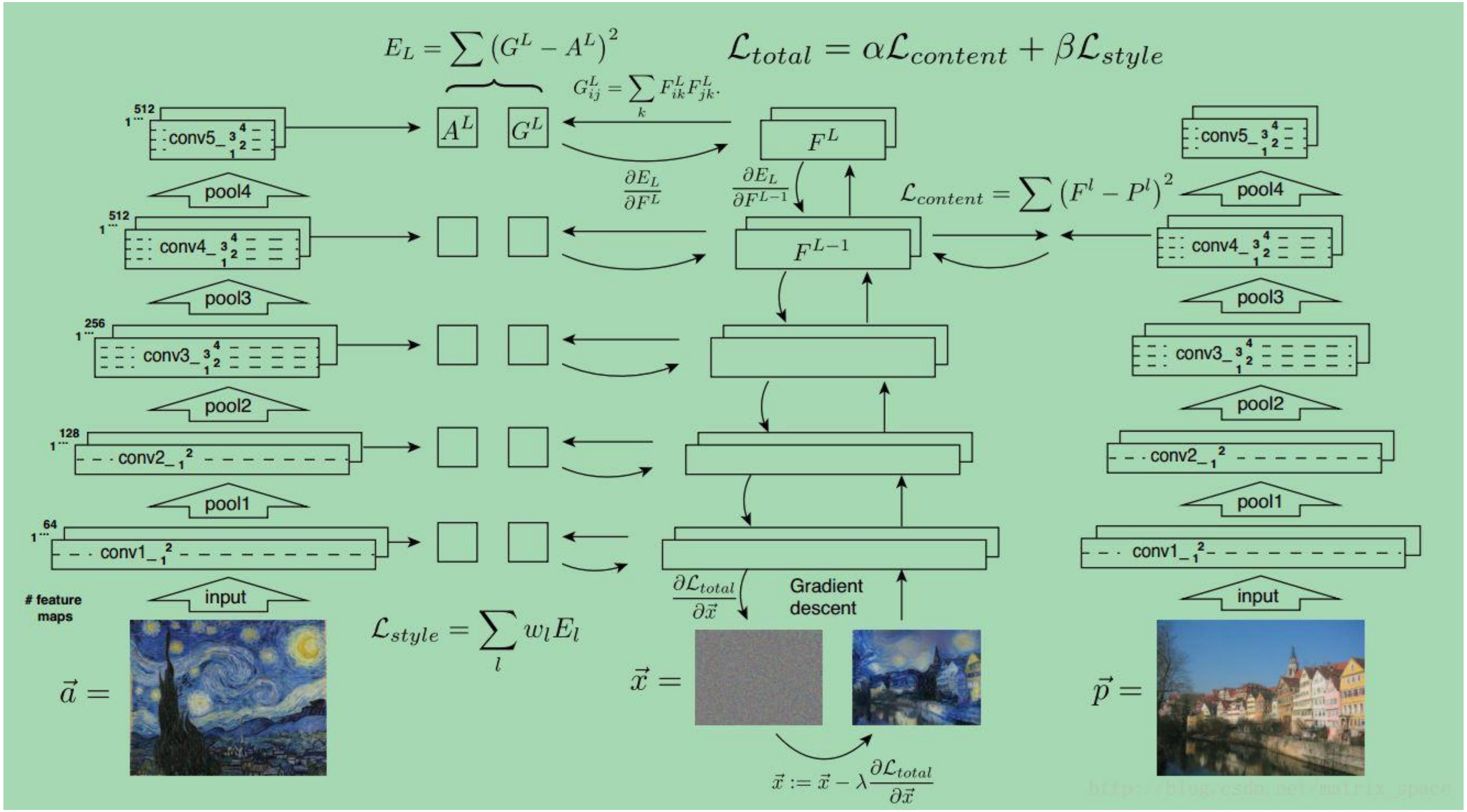


- The work in paper[1] were generated on the basis of the VGG-Network[2].

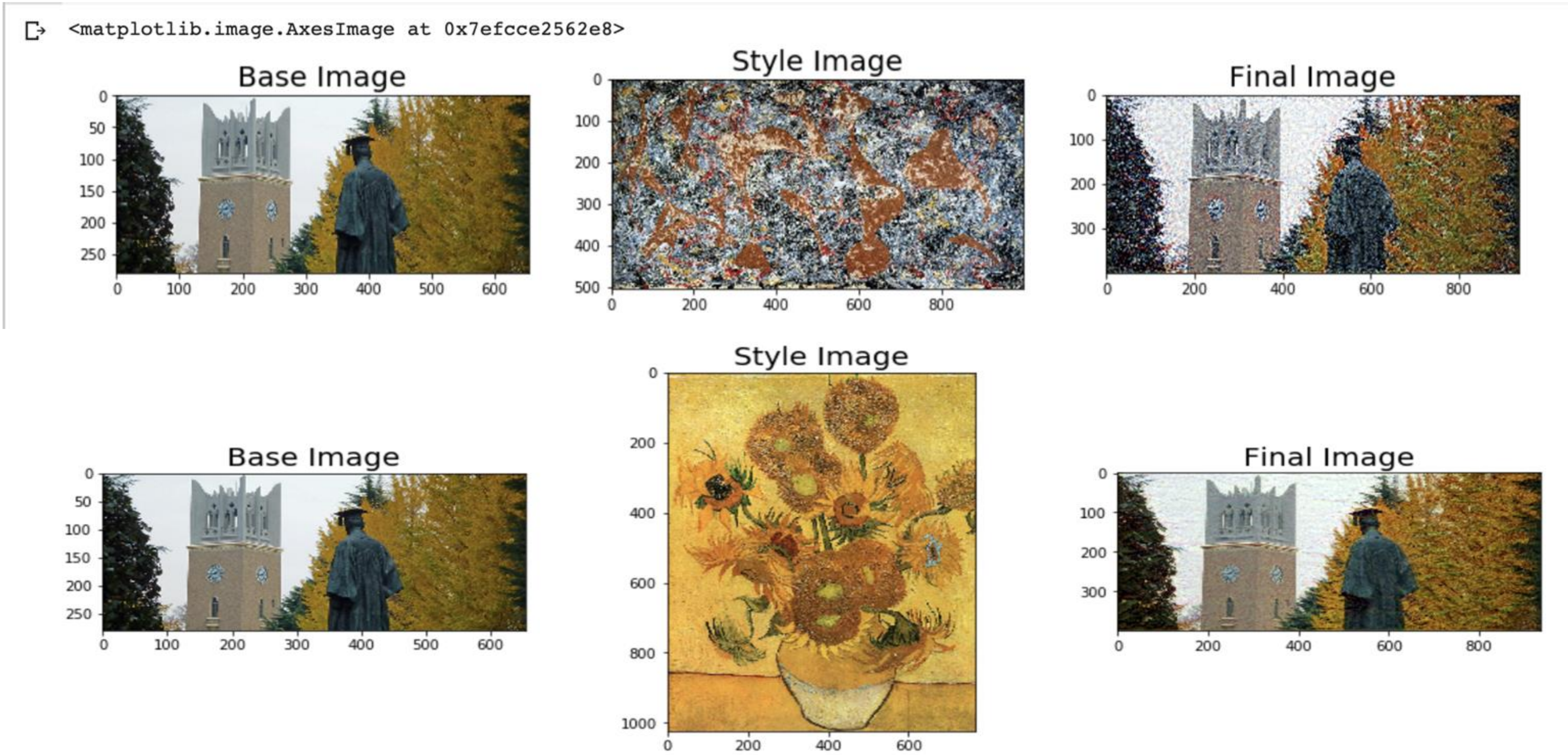
- VGG is known for:
- Simple Structure
- **Multiple small convolution** Kernels perform better than a single large convolution kernel.
- Effectively improve performance by increasing depth

[2] Simonyan K , Zisserman A . Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.

VGG in Style Transfer[1]



Experiment of Style Transfer



Style for Human

1. how the artist expresses his or her vision.
2. the way the artist employs form
3. color
4. composition
5. name just a few.
6. the way the artist handles the medium
7. the method or technique that the artist uses.
8. philosophy or driving force behind the artwork.

Style for Machine

1. texture
2. color
3. shape
4. curvature
5. the velocity with which a brush- or pencil-stroke is applied

Task Description

- Inspired by the Style Transfer task, and to resolve the problem mentioned above, we'd like to do classifications to the artworks through neural networks.
- **Objective**: Develop an algorithm which will identify the artist when provided with a painting.

- In pre-research, the basic principle of learning painter style based on deep neural networks is through **CNN**, so we try our challenge by adopting CNN networks.

- CNN works well on feature extraction due to the special organizational structure of it. The combined effect of the convolutional layer and the pooling layer allows CNN to extract better characteristics from the image.

Dataset

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

Description:

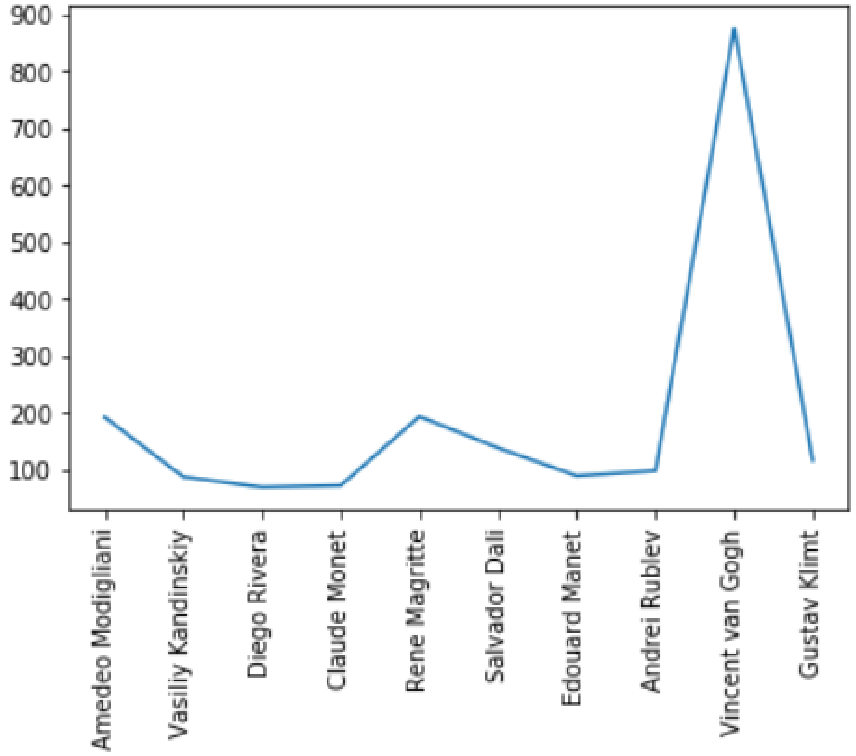
Artist	50
Artworks	8446
Style	29

The dataset includes a collection of artworks of the **50** most influential artists of all time.

Difficulty:
a. Most artists have different numbers of works.

b. Artists have different painting styles in different periods.

c. An artist has different painting styles in a painting.



Fine-tuning Method

Data Preprocessing:

1. Input requirements of the model is 224*224. Resizing the image randomly.

2. Take an middle part of the image.

3. Calculating the mean and standard deviation and do the batch normalization.

Parameters:

Method:	Fine-tuning method
Model:	ResNet18,ResNet50
Loss_fuction:	CrossEntropy
Optimizer:	SGD
Learning_rate :	0.001
Momentum:	0.9
Epoch:	25

Model

ResNet18

ResNet50

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Evaluation metrics

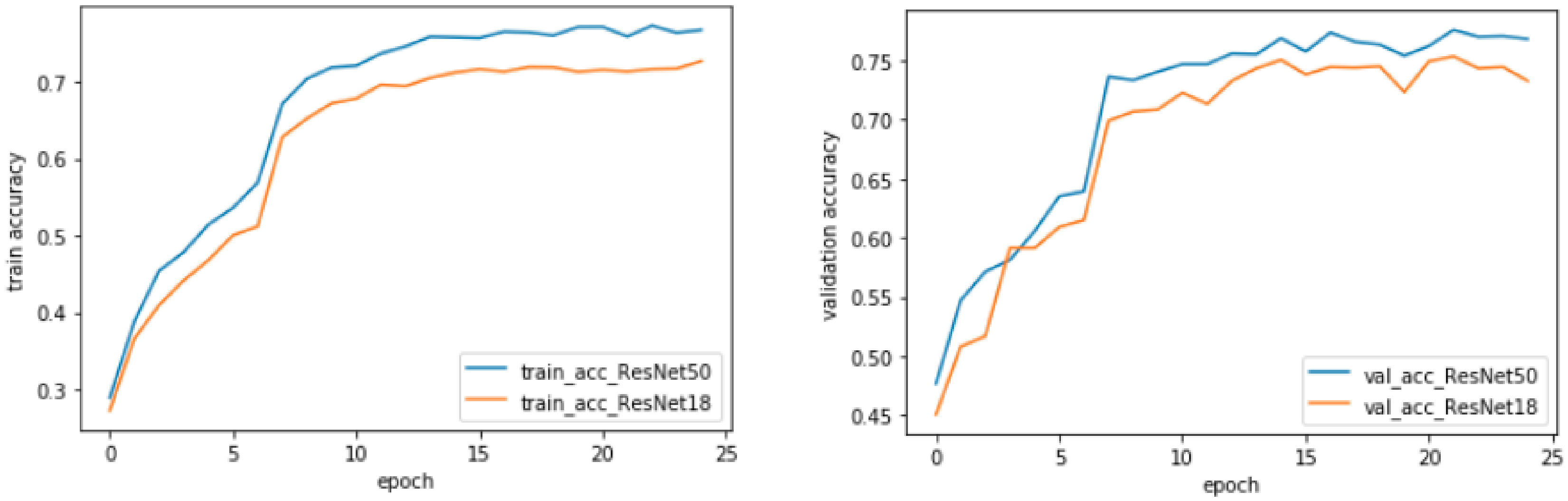
MCA(mean class accuracy) —— the average of the accuracies for all artists

This makes sure that the overall performance is not heavily biased by the performance on a single artist.

Experiment environment

OS: Ubuntu 18.04 LTS
CPU: Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz
GPU: Tesla V100, CUDA10.0, cuDNNv7.5.1
Memory: 96GB
Language: python3
Tool: Pytorch

Result

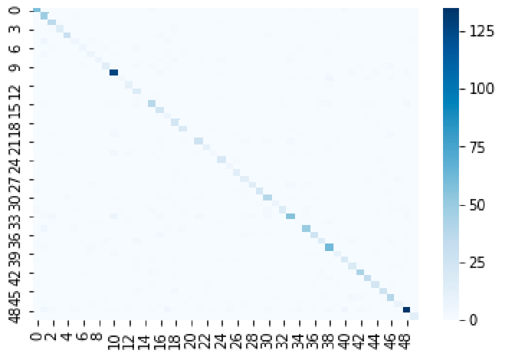


Best val Acc: 0.753700(ResNet18) 0.776199(ResNet50)

Confusion matrix

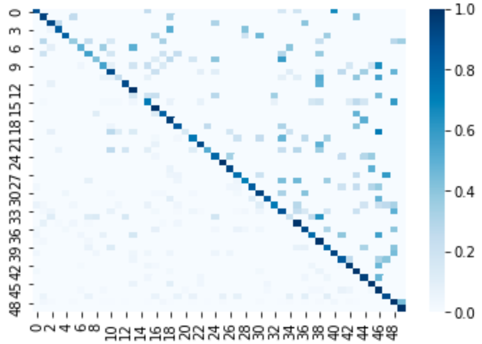
ResNet18

```
tensor([[ 59.,   0.,   0.,  ...,   0.,   0.,   0.],
        [  0.,  49.,   0.,  ...,   0.,   0.,   0.],
        [  0.,   0.,  36.,  ...,   0.,   0.,   0.],
        ...,
        [  0.,   1.,   0.,  ...,   8.,   0.,   0.],
        [  0.,   5.,   2.,  ...,   0., 135.,   1.],
        [  0.,   0.,   0.,  ...,   0.,   0.,  13.]])
```



Normalization →

```
[[0.8939394  0.         0.         ... 0.         0.         0.        ]
 [0.         0.9423077  0.         ... 0.         0.         0.        ]
 [0.         0.         0.9230769  ... 0.         0.         0.        ]
 ...
 [0.         0.05555556 0.         ... 0.9138933  0.         0.        ]
 [0.         0.02857143 0.01176273 ... 0.         0.9908286  0.44634214]
 [0.         0.         0.         ... 0.         0.         1.        ]]
```

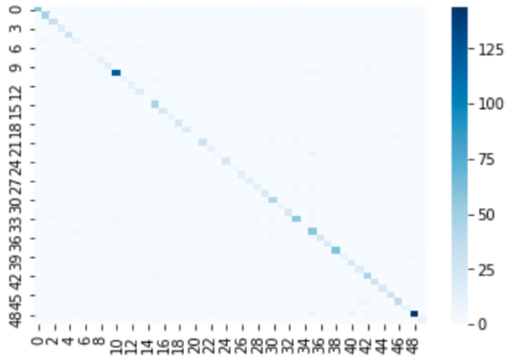


High percision

Artist name	Precision
Edgar_Degas	92.93%
Vasiliy_Kandinskiy	92.86%
Andrei_Rublev	92.47%
Jackson_Pollock	92.31%
Pierre-Auguste_Renoir	92.11%

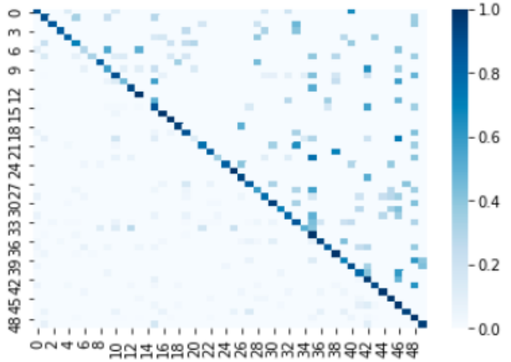
ResNet50

```
tensor([[ 59.,   0.,   0.,  ...,   0.,   0.,   0.],
        [  0.,  47.,   0.,  ...,   0.,   1.,   0.],
        [  0.,   0.,  34.,  ...,   0.,   1.,   0.],
        ...,
        [  0.,   1.,   0.,  ...,  10.,   0.,   0.],
        [  1.,   2.,   0.,  ...,   0., 144.,   0.],
        [  0.,   1.,   0.,  ...,   0.,   0.,   9.]])
```



Normalization →

```
[[0.8939394  0.         0.         ... 0.         0.         0.        ]
 [0.         0.90384614 0.         ... 0.         0.361558   0.        ]
 [0.         0.         0.8717949  ... 0.         0.3687206  0.        ]
 ...
 [0.         0.05555556 0.         ... 0.9478171  0.         0.        ]
 [0.00571429 0.01149388 0.         ... 0.         0.9986661  0.        ]
 [0.         0.07692308 0.         ... 0.         0.         0.9635809 ]]
```



Low percision

Artist name	Precision
Georges_Seurat	60.00%
Camille_Pissarro	62.39%
Gustav_Klimt	63.80%
Edouard_Manet	65.85%
Claude_Monet	65.96%

Comparison of different artists

We want to explore whether a artist's style affects his identification

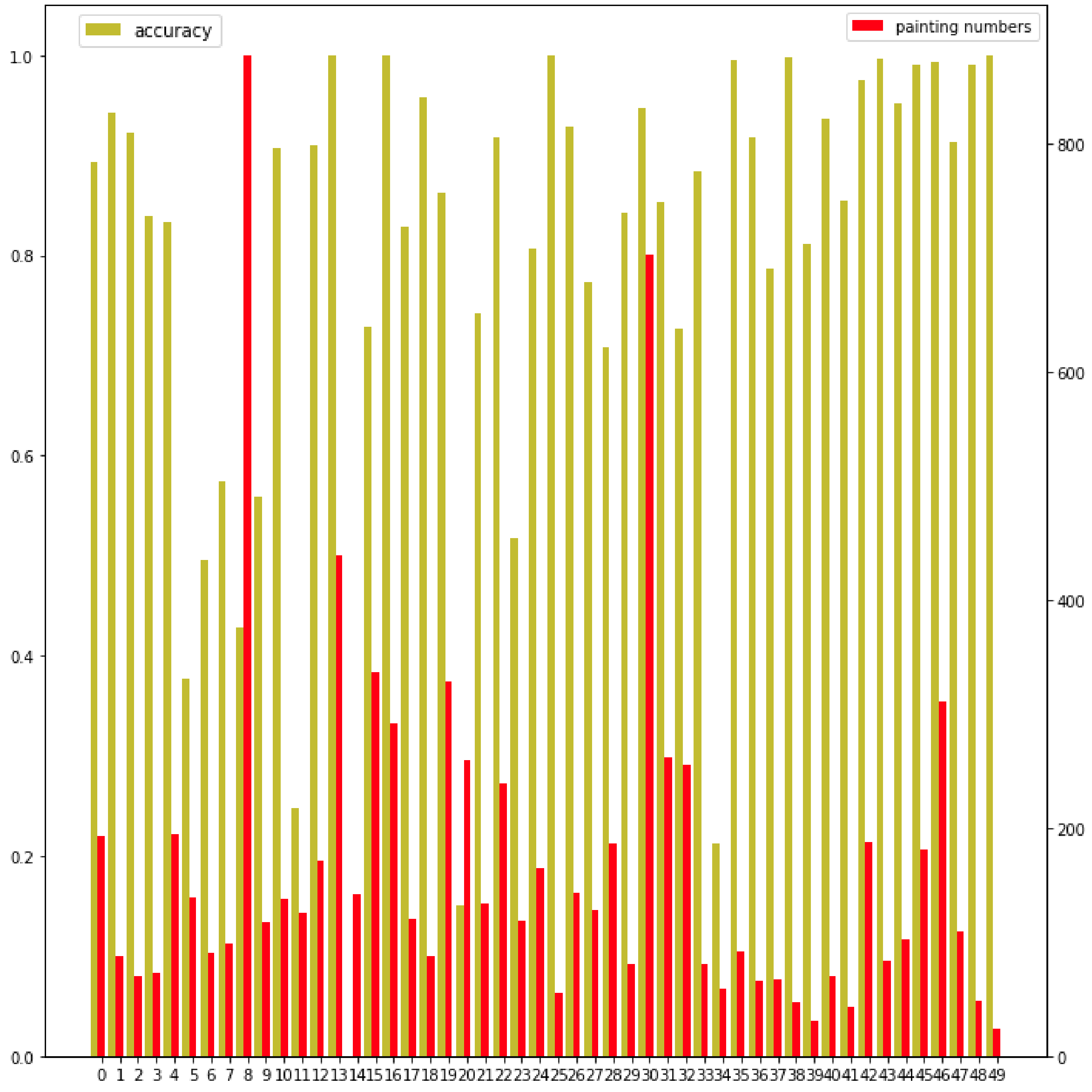
High recall

Artist name	Recall
Georges_Seurat	97.67%

Low recall

Artist name	Recall
Gustave_Courbet	30.51%

Statistics on painting numbers and accuracy

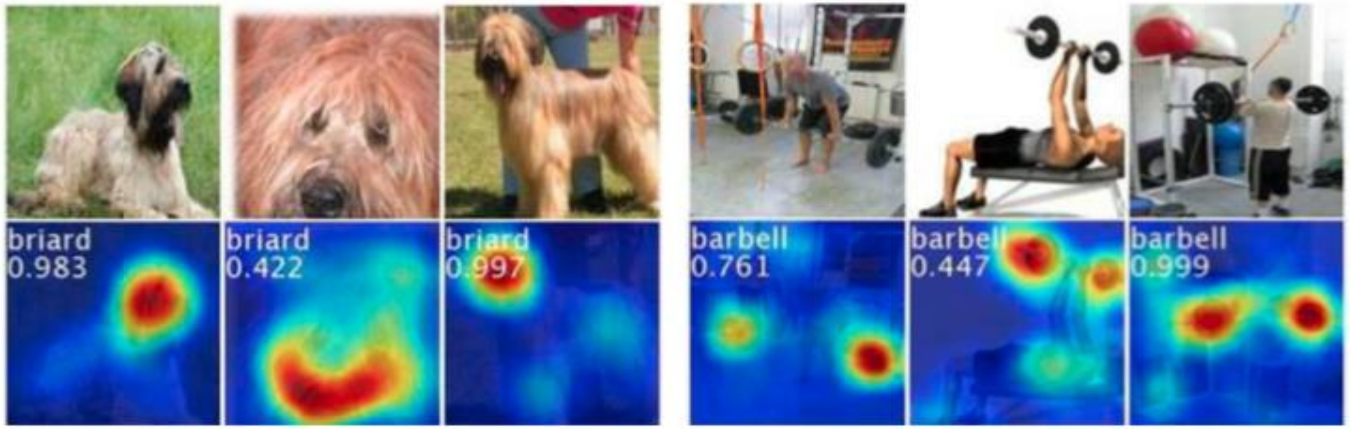


The results are not very satisfactory WHY？

We should find a way to visualize our model

Class activation mapping (CAM)

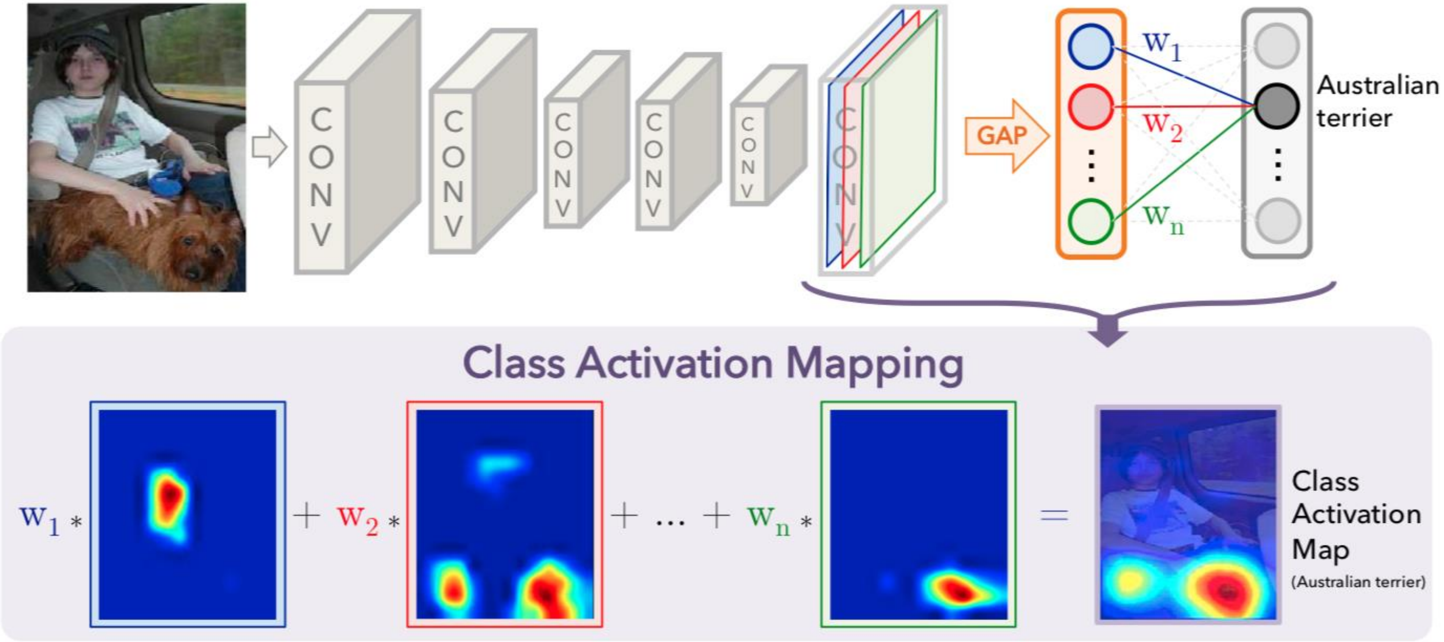
[3] Learning Deep Features for Discriminative Localization, *Bolei Zhou.etc.*; The IEEE CVPR, 2016, pp. 2921-2929



The maps highlight the discriminative image regions used for image classification, the head of the animal for briard and the plates in barbell.

Class activation mapping (CAM)

[3] Learning Deep Features for Discriminative Localization, *Bolei Zhou.etc.*; The IEEE CVPR, 2016, pp. 2921-2929



$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y} \sum_k w_k^c f_k(x,y).$$

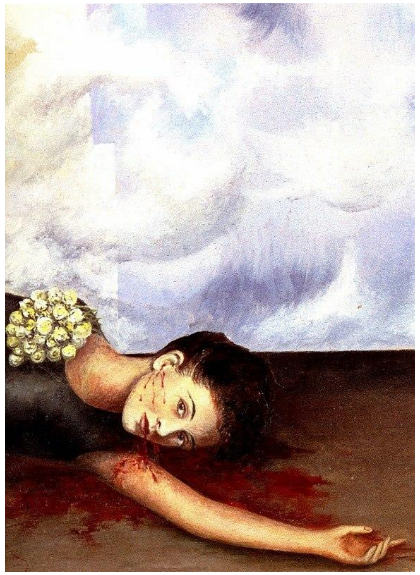
$f_k(x, y)$ represent the activation of unit k in the last convolutional layer at spatial location (x,y).

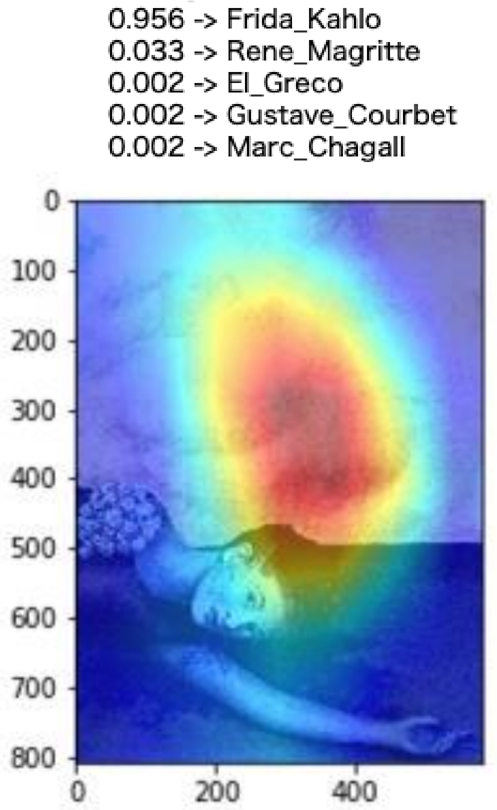
w_k^c indicates the importance of F_k for class c

Comparison of experimental results

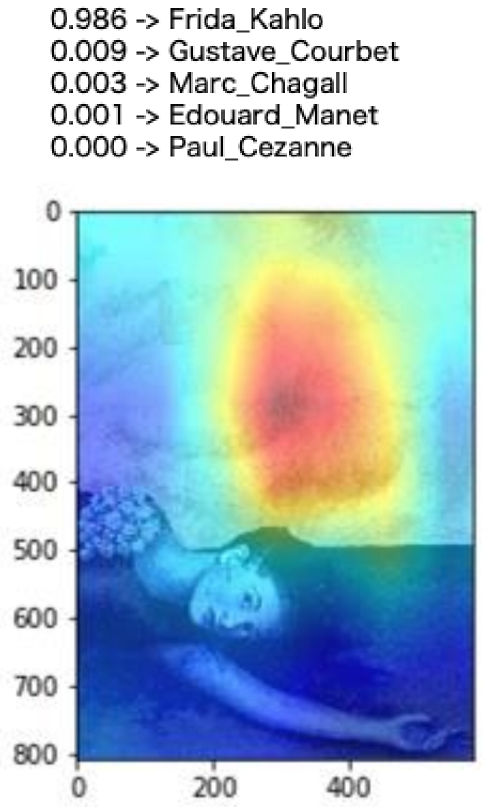
True

Frida_Kahlo





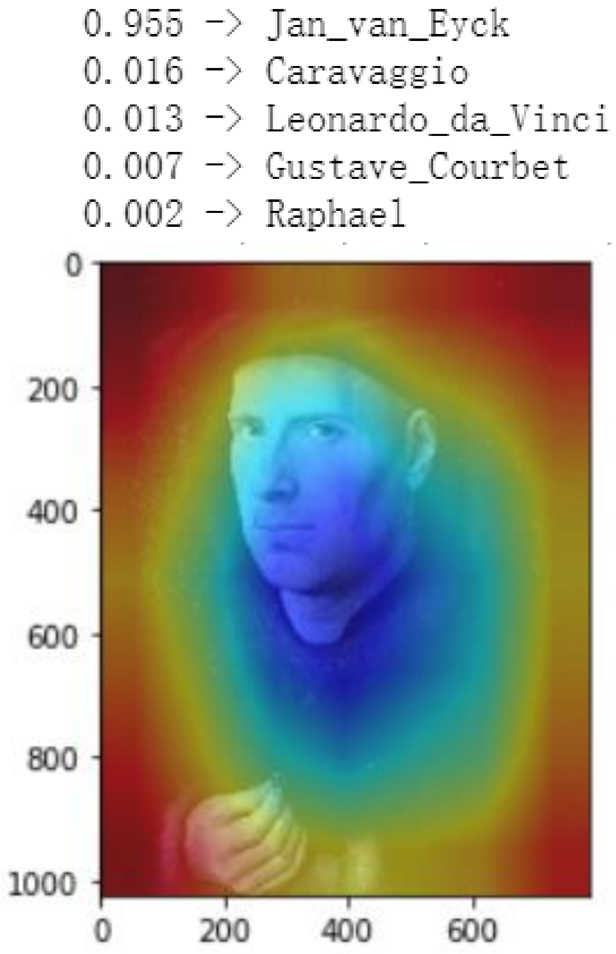
ResNet18



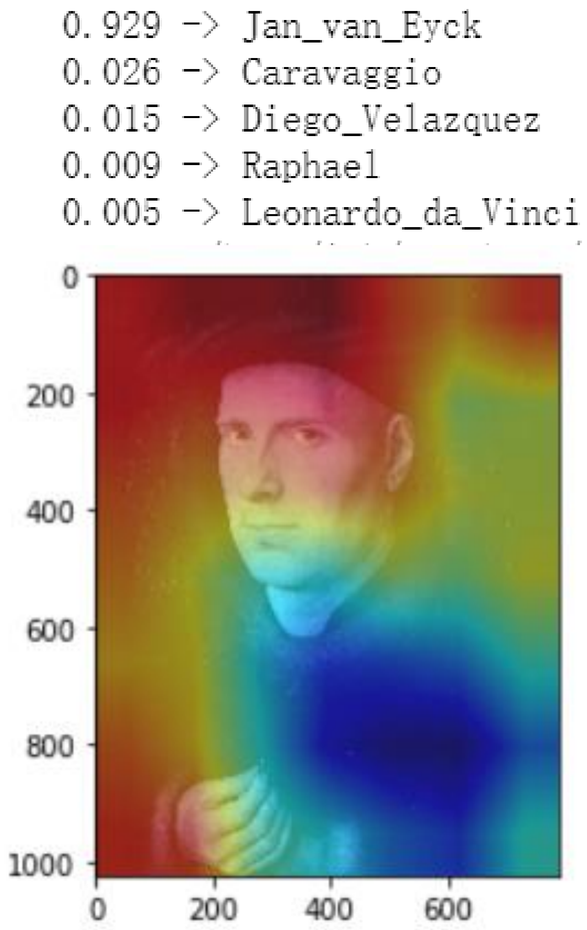
ResNet50

Comparison of experimental results

True
Jan van Eyck



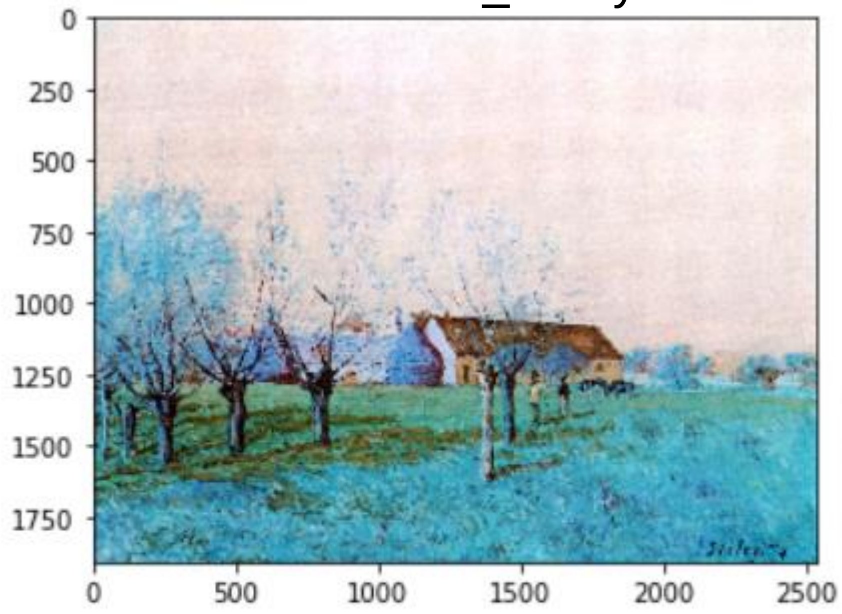
ResNet18



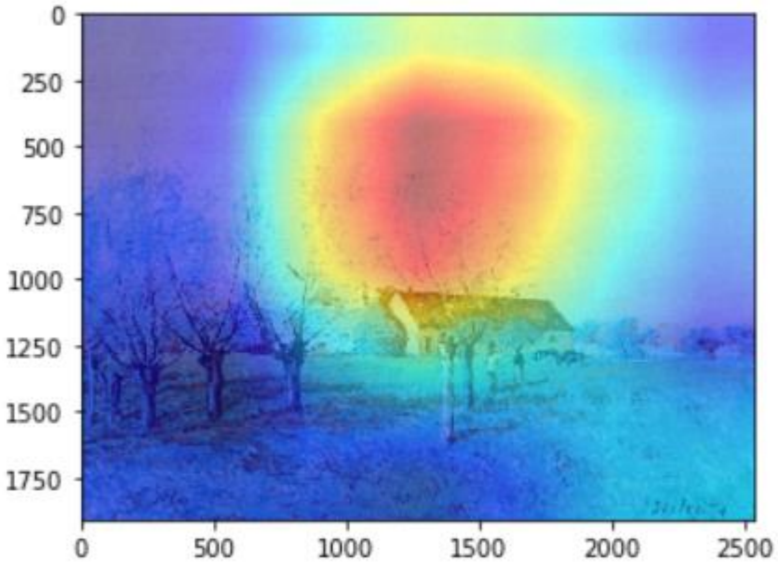
ResNet50

Comparison of experimental results

True
Alfred_Sisley

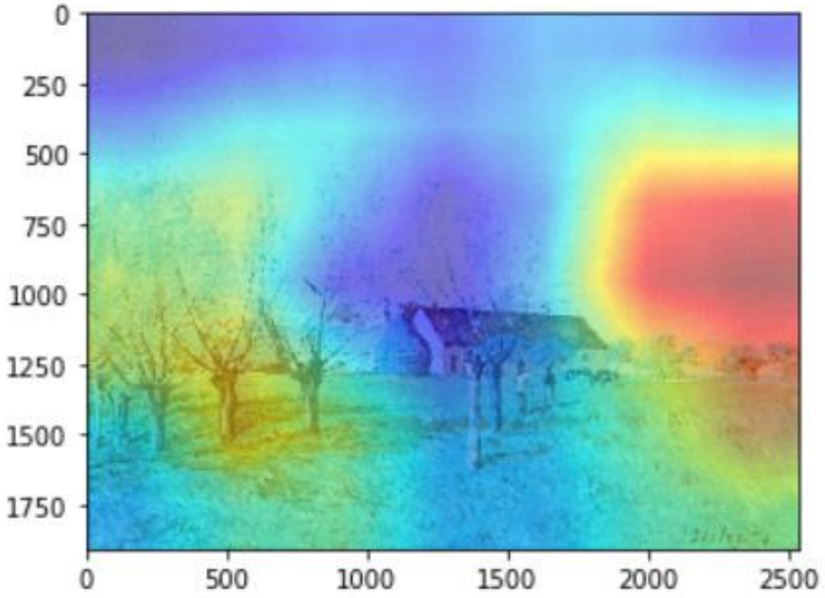


0.975 -> Alfred_Sisley
0.017 -> Camille_Pissarro
0.007 -> Vincent_van_Gogh
0.000 -> Georges_Seurat
0.000 -> Claude_Monet



ResNet18

0.544 -> Camille_Pissarro
0.375 -> Alfred_Sisley
0.060 -> Vincent_van_Gogh
0.015 -> Claude_Monet
0.002 -> Georges_Seurat



ResNet50

Conclusion

References

[1] Gatys L A, Ecker A S, Bethge M. A Neural Algorithm of Artistic Style[J]. Computer Science, 2015.

[2] Simonyan K , Zisserman A . Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.

[3] Learning Deep Features for Discriminative Localization, Bolei Zhou.etc.; The IEEE CVPR, 2016, pp. 2921-2929