Downloading Images using the Kaggle API and slipting images into train and test folders

We can directly download datasets from https://www.kaggle.com/ikarus777/best-artworks-of-all-time (https://www.kaggle.com/ikarus777/best-artworks-of-all-time); However, we can also do this work by following codes with the help of Kaggle API.

To use the Kaggle API, sign up for a Kaggle account at https://www.kaggle.com (https://www.kaggle.com). And other crucial steps please refer to https://github.com/Kaggle/kaggle-api (https://github.com/Kaggle/kaggle-api).

In [1]:
```
import os
from zipfile import ZipFile
import datetime

exists = os.path.isdir('./kaggle')
if exists:
    print("Database is ready")
    print(os.listdir('./kaggle'))
else:
    os.environ['KAGGLE_USERNAME'] = "mingshengyin"
    os.environ['KAGGLE_KEY'] = "a8763018b2b00591157c4fdda973a88b"
    import kaggle
    kaggle.api.authenticate()
    kaggle.api.dataset_download_files('ikarus777/best-artworks-of-all-time',
```

```
Database is ready
['artists.csv', 'images.zip', 'resized.zip']
```

Read the information of zip files

In [2]:
```
# specifying the zip file name
file_name = './kaggle/images.zip'

# opening the zip file in READ mode
with ZipFile(file_name, 'r') as zip:
    for info in zip.infolist():
            print(info.filename)
            print('\tModified:\t' + str(datetime.datetime(*info.date_time)))
            print('\tSystem:\t\t' + str(info.create_system) + '(0 = Windows,
            print('\tZIP version:\t' + str(info.create_version))
            print('\tCompressed:\t' + str(info.compress_size) + ' bytes')
            print('\tUncompressed:\t' + str(info.file_size) + ' bytes')
```

```
images/
        Modified:       2019-03-01 09:32:10
        System:         3(0 = Windows, 3 = Unix)
        ZIP version:    21
        Compressed:     0 bytes
        Uncompressed:   0 bytes
images/Leonardo_da_Vinci/
        Modified:       2019-03-01 08:47:20
        System:         3(0 = Windows, 3 = Unix)
        ZIP version:    21
        Compressed:     0 bytes
        Uncompressed:   0 bytes
images/Leonardo_da_Vinci/Leonardo_da_Vinci_78.jpg
        Modified:       2019-03-01 08:46:30
        System:         3(0 = Windows, 3 = Unix)
        ZIP version:    21
        Compressed:     121451 bytes
        Uncompressed:   121790 bytes
images/Leonardo_da_Vinci/Leonardo_da_Vinci_79.jpg
```

Unzip and extract all files

In [3]:
```
# opening the zip file in READ mode
file_name = './kaggle/images.zip'
with ZipFile(file_name, 'r') as zip:
    # printing all the contents of the zip file
    zip.printdir()

    # extracting all the files
    print('Extracting all the files now...')
    zip.extractall()
    print('Done!')
```

```
File Name                                             Modified
Size
images/                                        2019-03-01 09:32:10
0
images/Leonardo_da_Vinci/                      2019-03-01 08:47:20
0
images/Leonardo_da_Vinci/Leonardo_da_Vinci_78.jpg 2019-03-01 08:46:30
121790
images/Leonardo_da_Vinci/Leonardo_da_Vinci_79.jpg 2019-03-01 08:46:32
205020
images/Leonardo_da_Vinci/Leonardo_da_Vinci_8.jpg 2019-03-01 08:45:36
364658
images/Leonardo_da_Vinci/Leonardo_da_Vinci_80.jpg 2019-03-01 08:46:32
172620
images/Leonardo_da_Vinci/Leonardo_da_Vinci_81.jpg 2019-03-01 08:46:34
91047
images/Leonardo_da_Vinci/Leonardo_da_Vinci_82.jpg 2019-03-01 08:46:34
163159
images/Leonardo_da_Vinci/Leonardo_da_Vinci_83.jpg 2019-03-01 08:46:34
```

Each folder belongs to one artist, so we can tell #classes in total.

In [4]:
```
artist_name = os.listdir('./images')
len(artist_name)
```

Out[4]: 50

Delete useless folders

In original dataset, under each artist folder, there is an empty folder named "resized". We don't need it in this project, then delete it.

In [5]:
```
for i in range (len(artist_name)):
    folder_name = './images/' + artist_name[i]
    if os.path.isdir(folder_name + '/resized'):
        os.rmdir(folder_name + '/resized')
print("Done")
```

```
Done
```

Split data into train and test folders

If it does not exist, create two folders for storing train and test data.

In [6]:
```
if os.path.isdir('./images_test') == False:
    os.mkdir('./images_test')
    print('Done')
else:
    print('Has Test Folder')
if os.path.isdir('./images_train') == False:
    os.mkdir('./images_train')
    print('Done')
else:
    print('Has Train Folder')
```

```
Done
Done
```

In [7]:
```
image_path = './images/'
test_path = './images_test/'
train_path = './images_train/'
print("{0:^25} {1:^10} {2:^10} {3:^10}".format('Name','Total','Train','Test'
for i in range (len(artist_name)):
    artist_path = image_path + artist_name[i]
    image_name = os.listdir(artist_path)
    ntest = len(image_name) // 4
    ntrain = len(image_name) - ntest
    print("{0:^25} {1:^10} {2:^10} {3:^10}".format(artist_name[i],len(image_
```

Name	Total	Train	Test
Albrecht_Du ╞êrer	328	246	82
Alfred_Sisley	259	195	64
Amedeo_Modigliani	193	145	48
Andrei_Rublev	99	75	24
Andy_Warhol	181	136	45
Camille_Pissarro	91	69	22
Caravaggio	55	42	13
Claude_Monet	73	55	18
Diego_Rivera	70	53	17
Diego_Velazquez	128	96	32
Edgar_Degas	702	527	175
Edouard_Manet	90	68	22
Edvard_Munch	67	51	16
El_Greco	87	66	21
Eugene_Delacroix	31	24	7
Francisco_Goya	291	219	72
Frida_Kahlo	120	90	30
Georges_Seurat	43	33	10
Giotto_di_Bondone	119	90	29
Gustave_Courbet	59	45	14
Gustav_Klimt	117	88	29
Henri_de_Toulouse-Lautrec	81	61	20
Henri_Matisse	186	140	46
Henri_Rousseau	70	53	17
Hieronymus_Bosch	137	103	34
Jackson_Pollock	24	18	6
Jan_van_Eyck	81	61	20
Joan_Miro	102	77	25
Kazimir_Malevich	126	95	31
Leonardo_da_Vinci	143	108	35
Marc_Chagall	239	180	59
Michelangelo	49	37	12
Mikhail_Vrubel	171	129	42
Pablo_Picasso	439	330	109
Paul_Cezanne	47	36	11
Paul_Gauguin	311	234	77
Paul_Klee	188	141	47
Peter_Paul_Rubens	141	106	35
Pierre-Auguste_Renoir	336	252	84
Pieter_Bruegel	134	101	33
Piet_Mondrian	84	63	21
Raphael	109	82	27
Rembrandt	262	197	65
Rene_Magritte	194	146	48
Salvador_Dali	139	105	34

Sandro_Botticelli	164	123	41
Titian	255	192	63
Vasiliy_Kandinskiy	88	66	22
Vincent_van_Gogh	877	658	219
William_Turner	66	50	16

Split train images and test images (80% & 20%), and only keep the artists whose paintings more than 100.

In [8]:
```
import shutil
```

In [9]:
```
for i in range (len(artist_name)):
    artist_path = image_path + artist_name[i]
    image_name = os.listdir(artist_path)
    nsample = len(image_name)
    if nsample >= 100: # only keep paintings more than 100
        ntest = len(image_name) // 5
        ntrain = len(image_name) - ntest
        #make folder for each artists
        os.mkdir(test_path + artist_name[i])
        os.mkdir(train_path + artist_name[i])
        for j in range (nsample):
            if j < ntest:
                shutil.copy2(artist_path + '/' + image_name[j],
                             test_path + artist_name[i] + '/'+image_name[j])
            else:
                shutil.copy2(artist_path + '/' + image_name[j],
                             train_path + artist_name[i] + '/'+image_name[j])
print('Done')
```

```
Done
```

In [ ]:
```

```

Exploratory Dataset Analysis

After downloading and preliminary processing of redundant and useless data in the database, we analyze the basic situation of the data set. In this section, we will present information about 50 painters and study the artist's style and personal information.

First import the packages need to use

In [1]:
```
import os
import cv2
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Then display the dataset path

In [2]:
```
os.listdir('./')
```

Out[2]:
```
['.ipynb_checkpoints', 'artists.csv', 'Dataset_Analysis.ipynb', 'images']
```

Extract the information of csv file

In [3]:
```
df = pd.read_csv('./artists.csv', index_col=False)
print(df.head(5))
```

```
   id                 name        years                          genre  \
0   0   Amedeo Modigliani  1884 - 1920                 Expressionism
1   1  Vasiliy Kandinskiy  1866 - 1944  Expressionism,Abstractionism
2   2         Diego Rivera  1886 - 1957       Social Realism,Muralism
3   3         Claude Monet  1840 - 1926                 Impressionism
4   4       Rene Magritte  1898 - 1967      Surrealism,Impressionism

  nationality                                                bio  \
0     Italian  Amedeo Clemente Modigliani (Italian pronunciat...
1     Russian  Wassily Wassilyevich Kandinsky (Russian: Васи́...
2     Mexican  Diego María de la Concepción Juan Nepomuceno E...
3      French  Oscar-Claude Monet (; French: [klod mɔnɛ]; 14 ...
4     Belgian  René François Ghislain Magritte (French: [ʁəne...

                                      wikipedia  paintings
0  http://en.wikipedia.org/wiki/Amedeo_Modigliani (http://en.wikipedia.or
g/wiki/Amedeo_Modigliani)        193
1  http://en.wikipedia.org/wiki/Wassily_Kandinsky (http://en.wikipedia.or
g/wiki/Wassily_Kandinsky)         88
2       http://en.wikipedia.org/wiki/Diego_Rivera (http://en.wikipedia.or
g/wiki/Diego_Rivera)         70
3       http://en.wikipedia.org/wiki/Claude_Monet (http://en.wikipedia.or
g/wiki/Claude_Monet)         73
4      http://en.wikipedia.org/wiki/René_Magritte (http://en.wikipedia.or
g/wiki/René_Magritte)        194
```

In [4]:
```
df = df.drop(['bio','wikipedia','id'], axis=1)
```

In [5]:
```
artist = df[['name','paintings','nationality']]
```

Sort descendings by number and reset index

Here we found that only 30 painters have more than 100 paintings identified as their paintings. Since the remaining painters have too few paintings, we decided to use only the top 30 painters in the subsequent data classification and research.

Vincent van Gogh, Edgar Degas, and Pablo Picasso are the three painters with the most paintings.

In [6]:
```
num_paintings = artist.sort_values(by = ['paintings'], ascending = False)
num_paintings = num_paintings.reset_index(drop = True)
print(num_paintings)
```

	name	paintings	nationality
0	Vincent van Gogh	877	Dutch
1	Edgar Degas	702	French
2	Pablo Picasso	439	Spanish
3	Pierre-Auguste Renoir	336	French
4	Albrecht Dürer	328	German
5	Paul Gauguin	311	French
6	Francisco Goya	291	Spanish
7	Rembrandt	262	Dutch
8	Alfred Sisley	259	French,British
9	Titian	255	Italian
10	Marc Chagall	239	French,Jewish,Belarusian
11	Rene Magritte	194	Belgian
12	Amedeo Modigliani	193	Italian
13	Paul Klee	188	German,Swiss
14	Henri Matisse	186	French
15	Andy Warhol	181	American
16	Mikhail Vrubel	171	Russian
17	Sandro Botticelli	164	Italian
18	Leonardo da Vinci	143	Italian
19	Peter Paul Rubens	141	Flemish
20	Salvador Dali	139	Spanish
21	Hieronymus Bosch	137	Dutch
22	Pieter Bruegel	134	Flemish
23	Diego Velazquez	128	Spanish
24	Kazimir Malevich	126	Russian
25	Frida Kahlo	120	Mexican
26	Giotto di Bondone	119	Italian
27	Gustav Klimt	117	Austrian
28	Raphael	109	Italian
29	Joan Miro	102	Spanish
30	Andrei Rublev	99	Russian
31	Camille Pissarro	91	French
32	Edouard Manet	90	French
33	Vasiliy Kandinskiy	88	Russian
34	El Greco	87	Spanish,Greek
35	Piet Mondrian	84	Dutch
36	Henri de Toulouse-Lautrec	81	French
37	Jan van Eyck	81	Flemish
38	Claude Monet	73	French
39	Diego Rivera	70	Mexican
40	Henri Rousseau	70	French
41	Edvard Munch	67	Norwegian
42	William Turner	66	British
43	Gustave Courbet	59	French
44	Caravaggio	55	Italian
45	Michelangelo	49	Italian
46	Paul Cezanne	47	French
47	Georges Seurat	43	French
48	Eugene Delacroix	31	French
49	Jackson Pollock	24	American

Analysis of the artist's nationality.

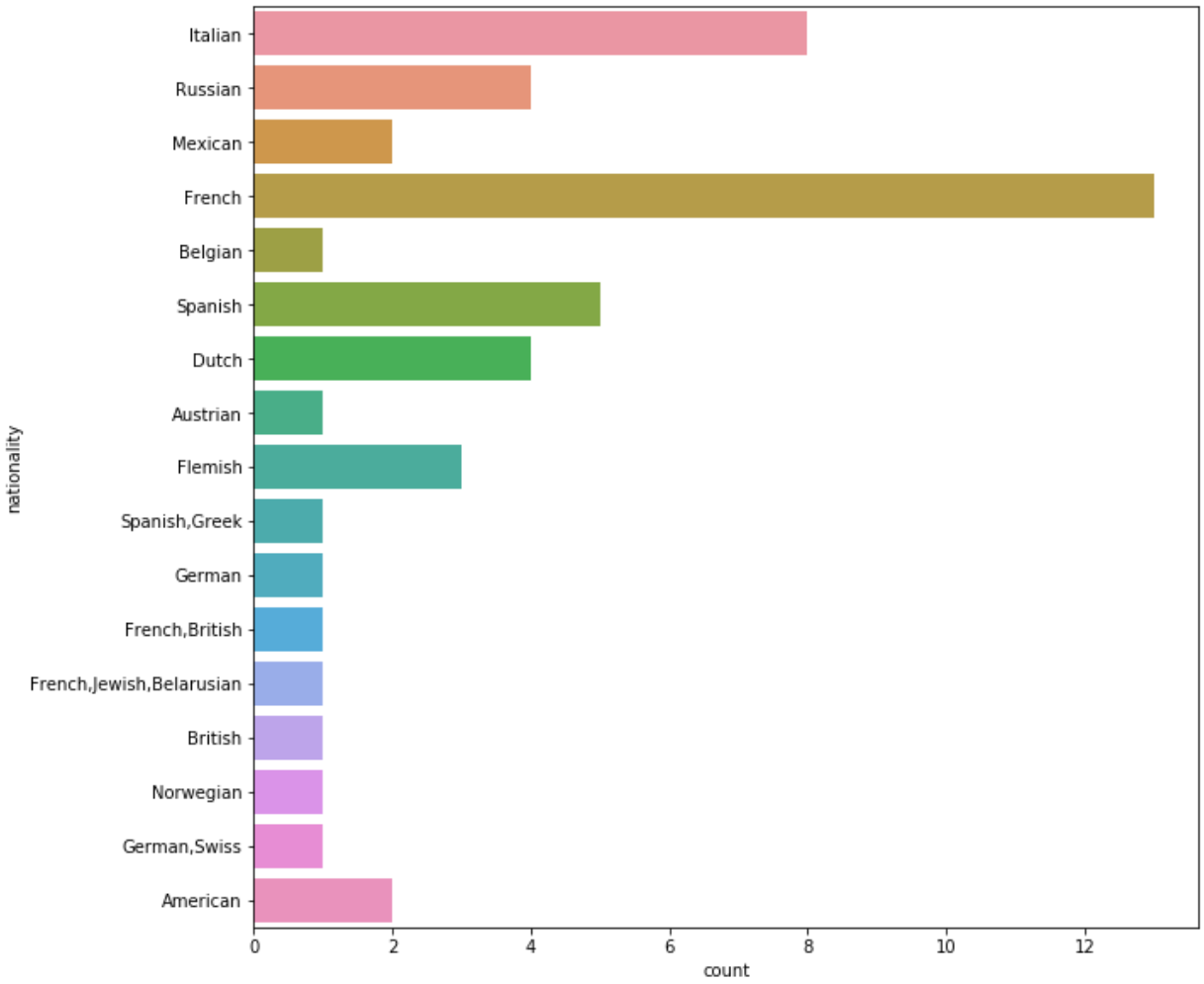
It can be seen that among the 50 most famous painters recognized by the world, the largest number of French painters

In [7]:
```
cla_national = artist.sort_values(by = ['nationality'], ascending = False)
cla_national = cla_national.reset_index(drop = True)
print(cla_national)
```

	name	paintings	nationality
0	El Greco	87	Spanish,Greek
1	Diego Velazquez	128	Spanish
2	Salvador Dali	139	Spanish
3	Joan Miro	102	Spanish
4	Francisco Goya	291	Spanish
5	Pablo Picasso	439	Spanish
6	Andrei Rublev	99	Russian
7	Vasiliy Kandinskiy	88	Russian
8	Kazimir Malevich	126	Russian
9	Mikhail Vrubel	171	Russian
10	Edvard Munch	67	Norwegian
11	Diego Rivera	70	Mexican
12	Frida Kahlo	120	Mexican
13	Leonardo da Vinci	143	Italian
14	Amedeo Modigliani	193	Italian
15	Giotto di Bondone	119	Italian
16	Titian	255	Italian
17	Raphael	109	Italian
18	Michelangelo	49	Italian
19	Sandro Botticelli	164	Italian
20	Caravaggio	55	Italian
21	Paul Klee	188	German,Swiss
22	Albrecht Dürer	328	German
23	Marc Chagall	239	French,Jewish,Belarusian
24	Alfred Sisley	259	French,British
25	Camille Pissarro	91	French
26	Henri Rousseau	70	French
27	Gustave Courbet	59	French
28	Henri de Toulouse-Lautrec	81	French
29	Georges Seurat	43	French
30	Edouard Manet	90	French
31	Edgar Degas	702	French
32	Paul Cezanne	47	French
33	Henri Matisse	186	French
34	Paul Gauguin	311	French
35	Claude Monet	73	French
36	Pierre-Auguste Renoir	336	French
37	Eugene Delacroix	31	French
38	Pieter Bruegel	134	Flemish
39	Jan van Eyck	81	Flemish
40	Peter Paul Rubens	141	Flemish
41	Vincent van Gogh	877	Dutch
42	Rembrandt	262	Dutch
43	Piet Mondrian	84	Dutch
44	Hieronymus Bosch	137	Dutch
45	William Turner	66	British
46	Rene Magritte	194	Belgian
47	Gustav Klimt	117	Austrian
48	Andy Warhol	181	American
49	Jackson Pollock	24	American

In [8]:
```
import seaborn as sns
plt.figure(figsize=(10,10))
nationality = sns.countplot(data = df, y='nationality')
nationality
```

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x241fe3bc550>



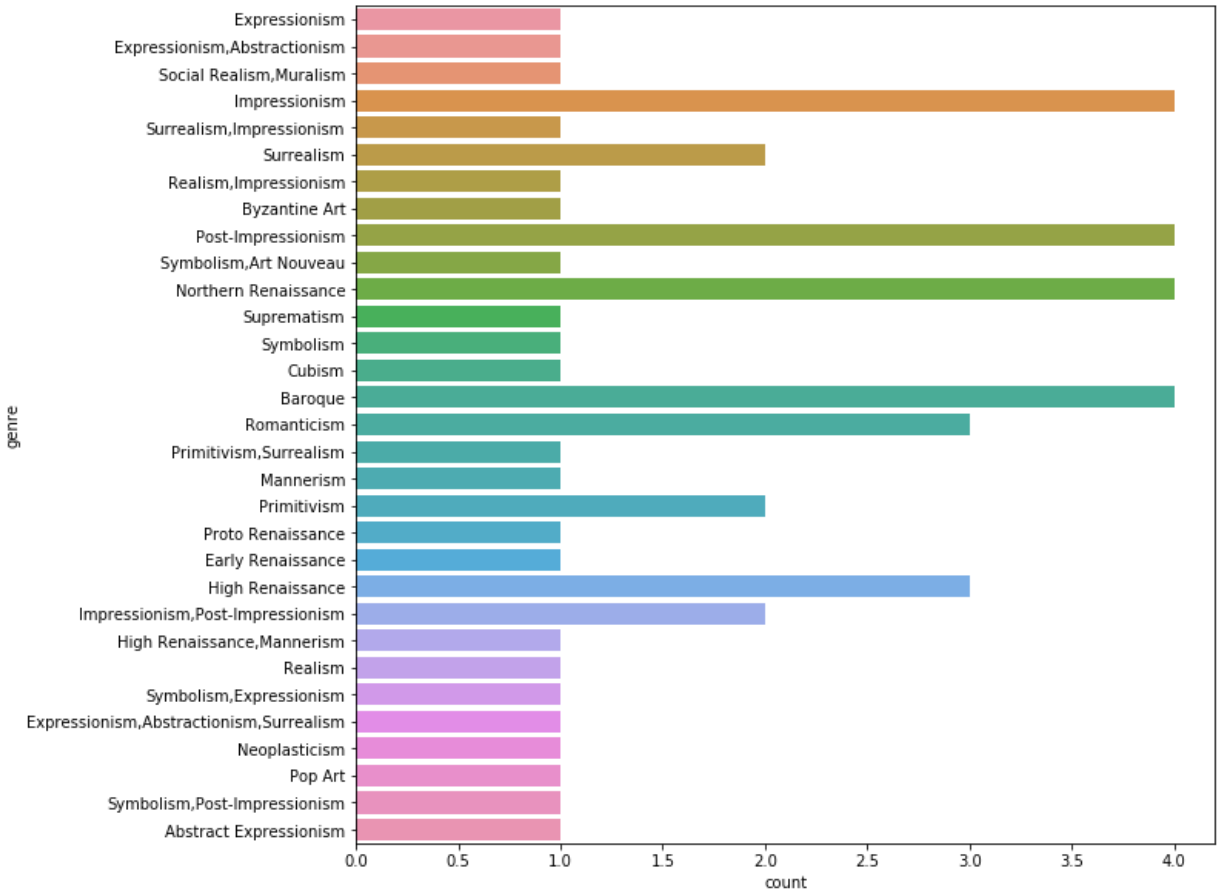
According to the style of the posterity of his paintings, classification. 10 of the 50 painters are Impressionism

In [9]:
```
genre = df[['name','genre']]
cla_genre = genre.sort_values(by = 'genre')
cla_genre = cla_genre.reset_index(drop = True)
print(cla_genre)
```

	name	genre
0	Jackson Pollock	Abstract Expressionism
1	Rembrandt	Baroque
2	Diego Velazquez	Baroque
3	Caravaggio	Baroque
4	Peter Paul Rubens	Baroque
5	Andrei Rublev	Byzantine Art
6	Pablo Picasso	Cubism
7	Sandro Botticelli	Early Renaissance
8	Amedeo Modigliani	Expressionism
9	Vasiliy Kandinskiy	Expressionism,Abstractionism
10	Paul Klee	Expressionism,Abstractionism,Surrealism
11	Raphael	High Renaissance
12	Leonardo da Vinci	High Renaissance
13	Michelangelo	High Renaissance
14	Titian	High Renaissance,Mannerism
15	Edgar Degas	Impressionism
16	Pierre-Auguste Renoir	Impressionism
17	Claude Monet	Impressionism
18	Alfred Sisley	Impressionism
19	Camille Pissarro	Impressionism,Post-Impressionism
20	Henri Matisse	Impressionism,Post-Impressionism
21	El Greco	Mannerism
22	Piet Mondrian	Neoplasticism
23	Pieter Bruegel	Northern Renaissance
24	Hieronymus Bosch	Northern Renaissance
25	Albrecht Dürer	Northern Renaissance
26	Jan van Eyck	Northern Renaissance
27	Andy Warhol	Pop Art
28	Georges Seurat	Post-Impressionism
29	Paul Cezanne	Post-Impressionism
30	Henri de Toulouse-Lautrec	Post-Impressionism
31	Vincent van Gogh	Post-Impressionism
32	Henri Rousseau	Primitivism
33	Marc Chagall	Primitivism
34	Frida Kahlo	Primitivism,Surrealism
35	Giotto di Bondone	Proto Renaissance
36	Gustave Courbet	Realism
37	Edouard Manet	Realism,Impressionism
38	William Turner	Romanticism
39	Eugene Delacroix	Romanticism
40	Francisco Goya	Romanticism
41	Diego Rivera	Social Realism,Muralism
42	Kazimir Malevich	Suprematism
43	Joan Miro	Surrealism
44	Salvador Dali	Surrealism
45	Rene Magritte	Surrealism,Impressionism
46	Mikhail Vrubel	Symbolism
47	Gustav Klimt	Symbolism,Art Nouveau
48	Edvard Munch	Symbolism,Expressionism
49	Paul Gauguin	Symbolism,Post-Impressionism

In [10]:
```
plt.figure(figsize=(10,10))
genre = sns.countplot(data = genre, y='genre')
genre
```

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x241fe9174a8>



Display paintings

In [11]:
```
from glob import glob
def plot_img(artist):
    path = './images/' + artist + '/**'
    all_img = glob(path)
    plt.figure(figsize=(15,15))
    plt.subplots_adjust(wspace=0, hspace=0)
    i = 0
    for image in all_img[:9]:
        img = cv2.imread(image)
        img = cv2.resize(img, (400, 400))
        plt.subplot(3, 3, i+1)
        plt.axis('off')
        plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
        i += 1
```

Show the paintings of two influence painters.

It is not difficult to find that Alfred Sisley and Claude Monet are two Impressionist landscape painters, and the two paint in the same period, the style is very similar. Even if human beings are not experts, it is difficult to accurately distinguish which masterpiece is a masterpiece.

Therefore, we estimate that the paintings of 30 painters are classified by convolutional neural network, and the final result should be an accuracy rate of 50%.

In [12]:
```
print(df[df['name'] == 'Alfred Sisley'])
plot_img('Alfred_Sisley')
```

```
              name        years          genre     nationality  paintings
20  Alfred Sisley  1839 - 1899  Impressionism  French,British        259
```

In [13]:
```
print(df[df['name'] == 'Claude Monet'])
plot_img('Claude_Monet')
```

```
            name        years          genre nationality  paintings
3  Claude Monet  1840 - 1926  Impressionism      French         73
```

Show baroque style paintings.

This style of painting is more similar, and basically people cannot distinguish painters by painting.

In [14]:
```
print(df[df['name'] == 'Diego Velazquez'])
plot_img('Diego_Velazquez')
```

```
                name        years    genre nationality  paintings
27  Diego Velazquez  1599 - 1660  Baroque     Spanish        128
```

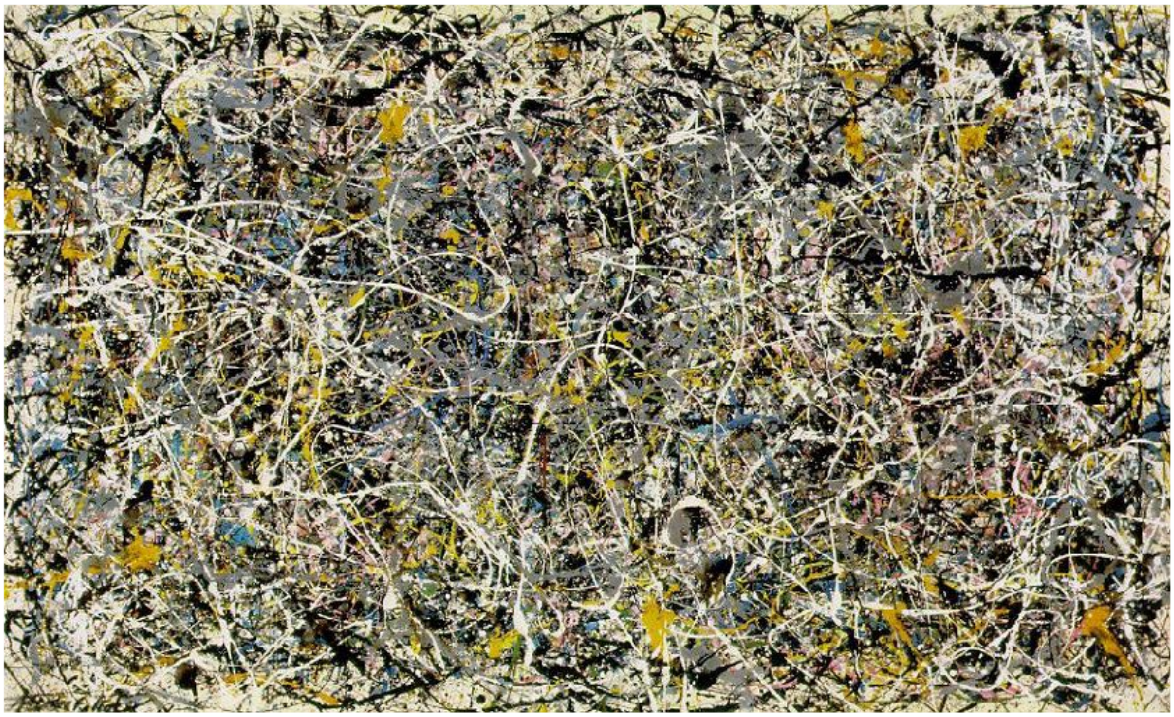
In [15]:
```
print(df[df['name'] == 'Caravaggio'])
plot_img('Caravaggio')
```

```
          name        years    genre nationality  paintings
25  Caravaggio  1571 - 1610  Baroque     Italian         55
```

Soothe your mood and see one of my favorite paintings, now in the Museum of Modern Art in New York.

In [16]:
```
print(df[df['name'] == 'Jackson Pollock'])
plt.figure(figsize=(15,15))
img = cv2.imread('./images/Jackson_Pollock/Jackson_Pollock_13.jpg')
plt.axis('off')
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.show()
```

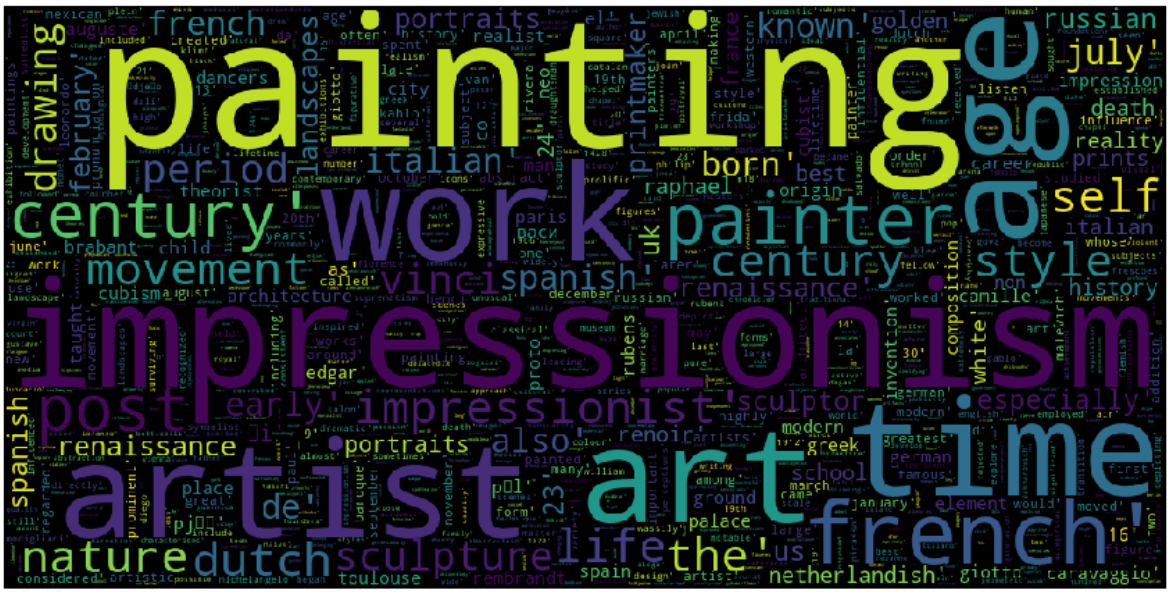
```
                name        years                    genre nationality  \
49  Jackson Pollock  1912 – 1956  Abstract Expressionism    American

    paintings
49         24
```



Finally, use wordcloud to draw a summary of all writers' information.

In [17]:
```
from wordcloud import WordCloud, STOPWORDS
from collections import Counter
from nltk.corpus import stopwords
df = pd.read_csv('./artists.csv')
column = 'bio'
nword = 1000
topic_words = [ z.lower() for y in
                [ x.split() for x in df[column] if isinstance(x, str)]
                for z in y]
word_count_dict = dict(Counter(topic_words))
popular_words = sorted(word_count_dict, key = word_count_dict.get, reverse =
popular_words_nonstop = [w for w in popular_words if w not in stopwords.word
word_string=str(popular_words_nonstop)
wordcloud = WordCloud(stopwords = STOPWORDS,
                      background_color = 'black',
                      max_words = nword,
                      width = 1000,height = 500,).generate(word_string)
plt.clf()
plt.figure(figsize=(15,15))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



In [ ]:
```
```