

# Jupyter Notebook Execution Report

**Name:** Siddharth Karmokar

**ID:** 123cs0061

**Project Title:** Lab1

**Project SubTitle:** Home Assignment

**Date:** January 05, 2026

---

## Cell 1: ■ Markdown

### Lab1

---

### Assignment 1

## Cell 2: ■ Markdown

### Question 1

## Cell 3: ■ Code

```
import matplotlib.pyplot as plt  
import cv2 as cv  
import numpy as np  
  
%matplotlib inline
```

### Error:

```
Traceback (most recent call last):  
  File "c:\Users\Cynix\.vscode\extensions\ganeshkumbhar.nb2pdf-1.1.9\scripts\nb2pdf.py", line 403  
    result = eval(lines[-1], glb)  
               ^^^^^^^^^^^^^^^^^^  
File "<string>", line 1  
  %matplotlib inline  
  ^  
SyntaxError: invalid syntax  
During handling of the above exception, another exception occurred:  
Traceback (most recent call last):  
  File "c:\Users\Cynix\.vscode\extensions\ganeshkumbhar.nb2pdf-1.1.9\scripts\nb2pdf.py", line 403
```

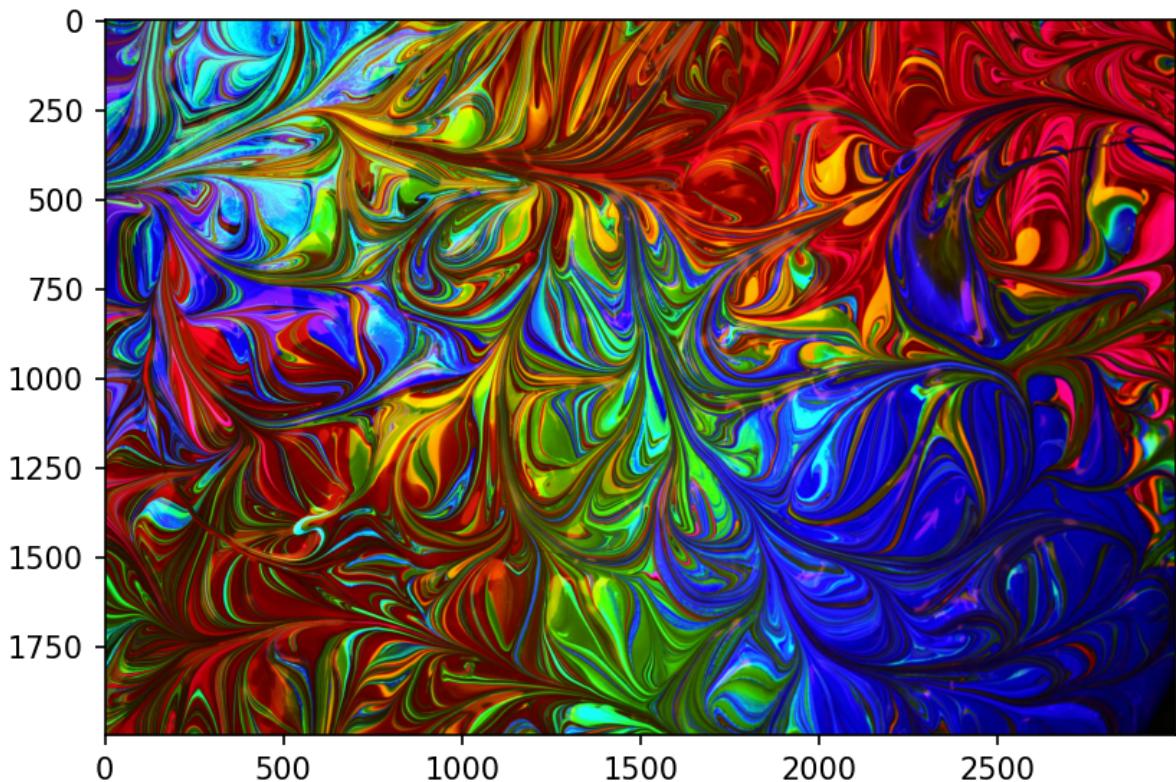
```
exec(source, glb)
File "<string>", line 5
    %matplotlib inline
^
SyntaxError: invalid syntax
```

#### Cell 4: ■ Code

```
X = cv.imread("./art.jpg")
plt.imshow(X)
```

#### Output:

```
<matplotlib.image.AxesImage object at 0x00000295F64E7090>;
```



#### Cell 5: ■ Code

```
if cv.imwrite("art.png", X):
    print("Saved image as a png")
else:
    print("Failed to save image as png")
```

#### Output:

```
Saved image as a png
```

## Cell 6: ■ Markdown

### Question 2

## Cell 7: ■ Code

```
x.shape
```

### Output:

```
(2000, 3000, 3)
```

## Cell 8: ■ Code

```
x.shape[-1]
```

### Output:

```
3
```

## Cell 9: ■ Code

```
x.dtype
```

### Output:

```
dtype('uint8')
```

## Cell 10: ■ Markdown

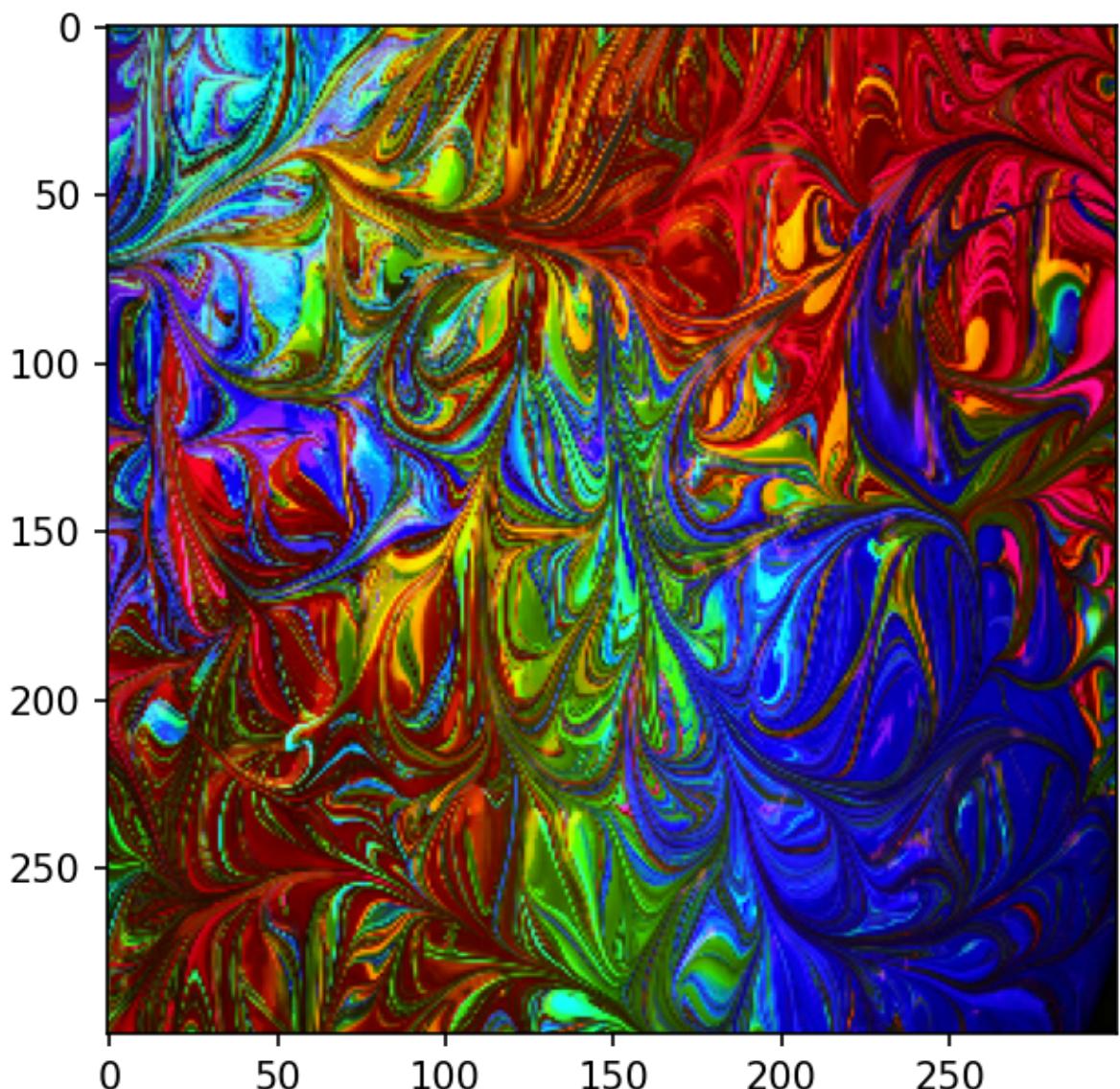
### Question 3

## Cell 11: ■ Code

```
nimg = cv.resize(X, (300, 300))  
plt.imshow(nimg)
```

### Output:

```
&lt;matplotlib.image.AxesImage object at 0x00000295F6BED9D0&gt;
```



### Cell 12: ■ Markdown

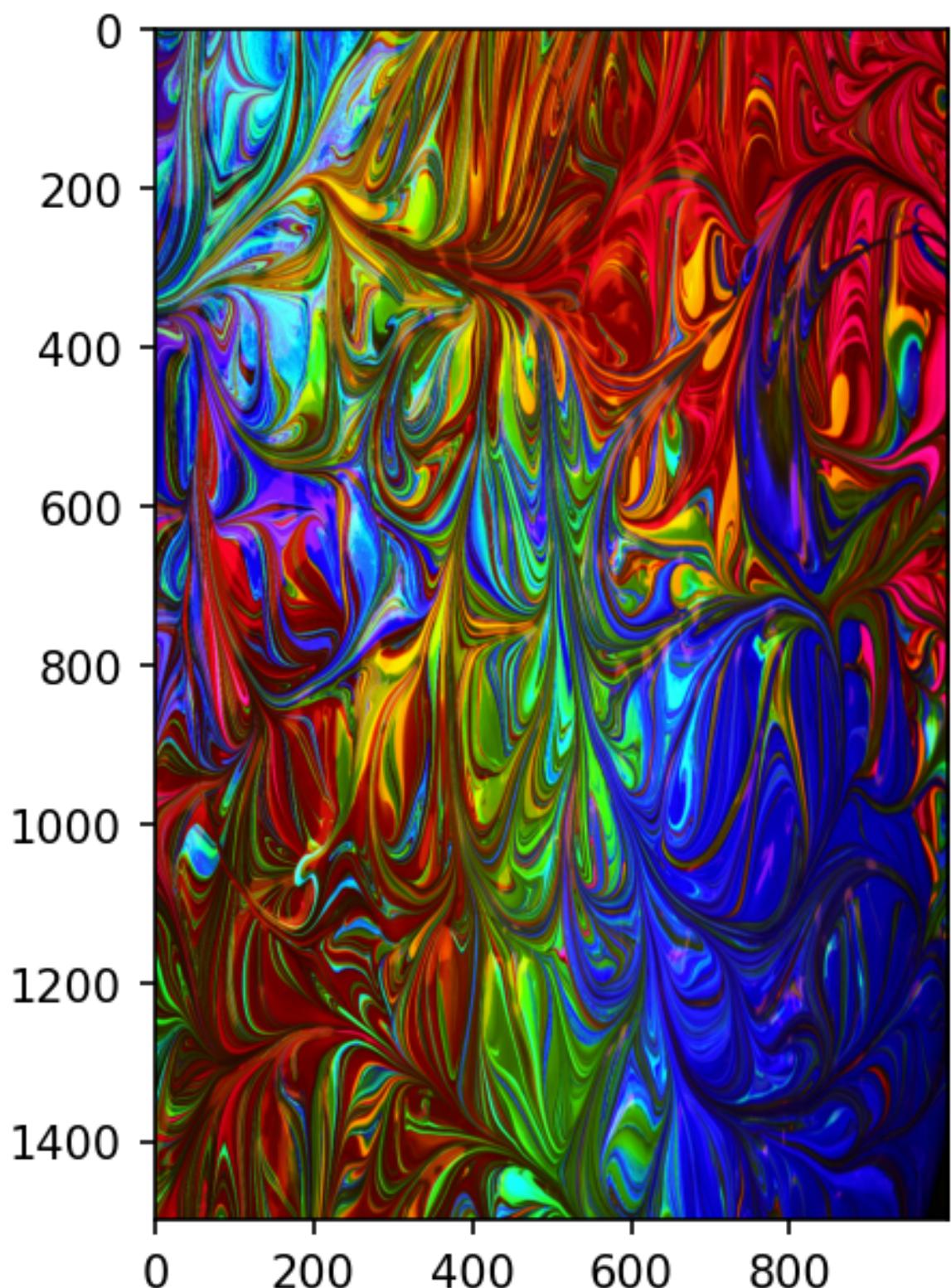
#### Question 4

### Cell 13: ■ Code

```
nimg = cv.resize(X, (X.shape[0]//2, X.shape[1]//2))  
plt.imshow(nimg)
```

#### Output:

```
&lt;matplotlib.image.AxesImage object at 0x00000295F6C49E10&gt;
```



Cell 14: ■ Markdown

## Question 5

### Cell 15: ■ Code

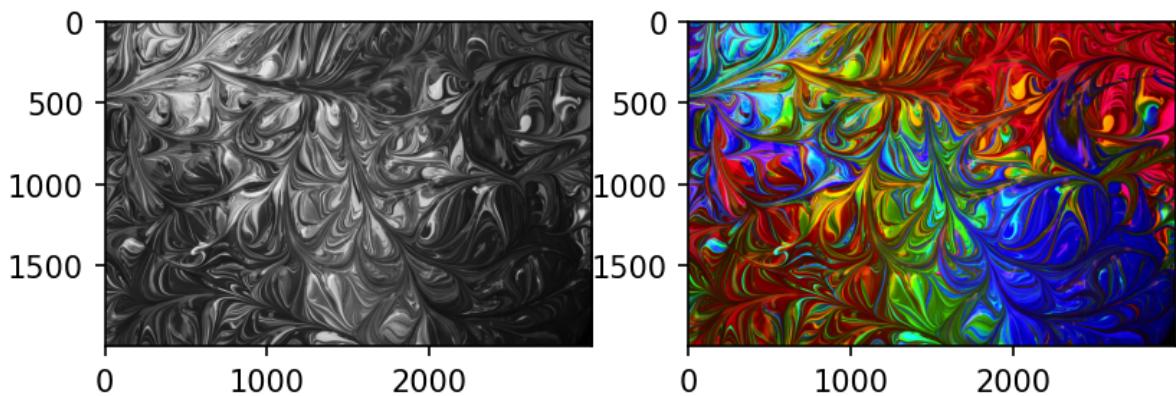
```
R = cv.cvtColor(X, cv.COLOR_RGB2GRAY)

plt.subplot(1, 2, 1)
plt.imshow(R, cmap='gray')

plt.subplot(1, 2, 2)
plt.imshow(X)
```

### Output:

```
<matplotlib.image.AxesImage object at 0x00000295F6CA7450>;
```



### Cell 16: ■ Markdown

## Question 6

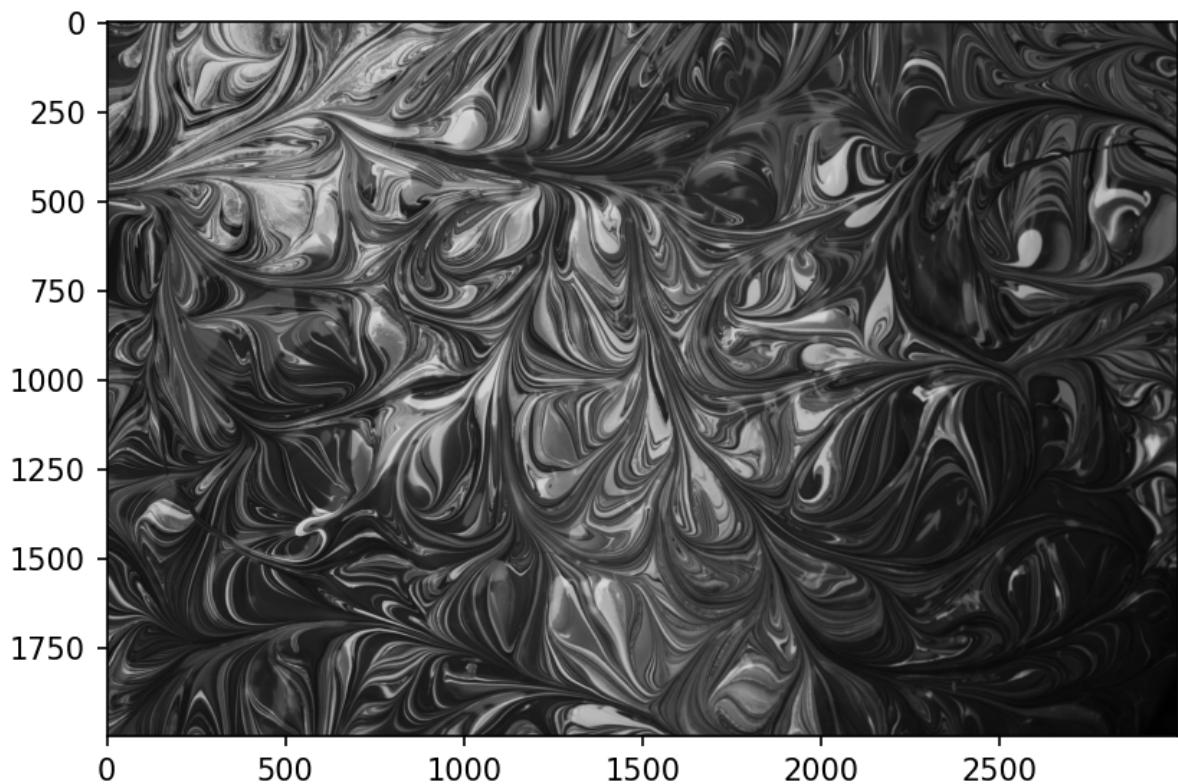
### Cell 17: ■ Code

```
nR = cv.cvtColor(R, cv.COLOR_GRAY2BGR)

plt.imshow(nR)
```

### Output:

```
<matplotlib.image.AxesImage object at 0x00000295F687CD90>;
```



### Cell 18: ■ Markdown

#### Question 7

### Cell 19: ■ Code

```
gray = cv.cvtColor(X, cv.COLOR_RGB2GRAY)

gray1d = gray.ravel()

plt.hist(gray1d, bins=256, range=[0, 256], color='gray')

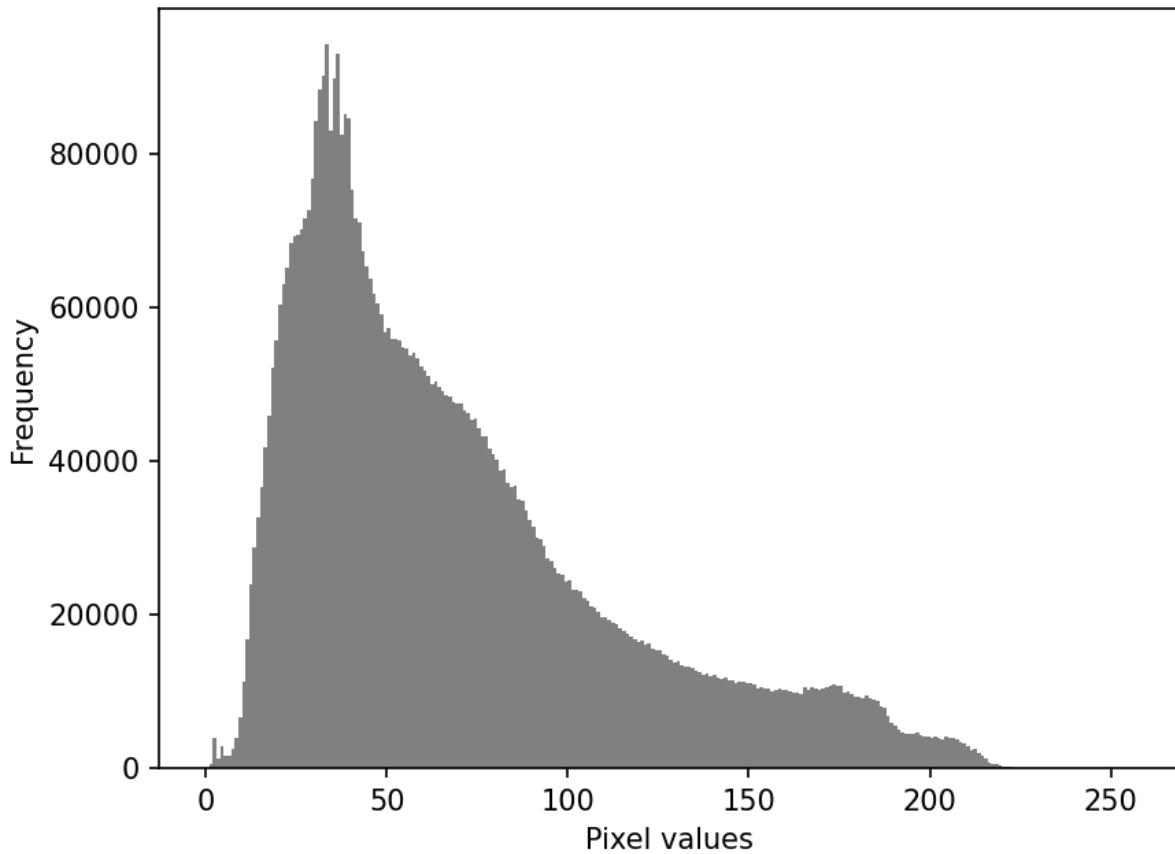
plt.xlabel("Pixel values")

plt.ylabel("Frequency")

plt.show()
```

#### Output:

```
[ STDERR ]
<string>:1: UserWarning: FigureCanvasAgg is non-interactive, and thus cannot be shown
```



### Cell 20: ■ Markdown

#### Question 8

### Cell 21: ■ Code

```
f32 = np.array(X).astype(np.float32)  
meanval = np.mean(f32)  
stdval = np.std(f32)  
print(meanval, stdval)
```

#### Output:

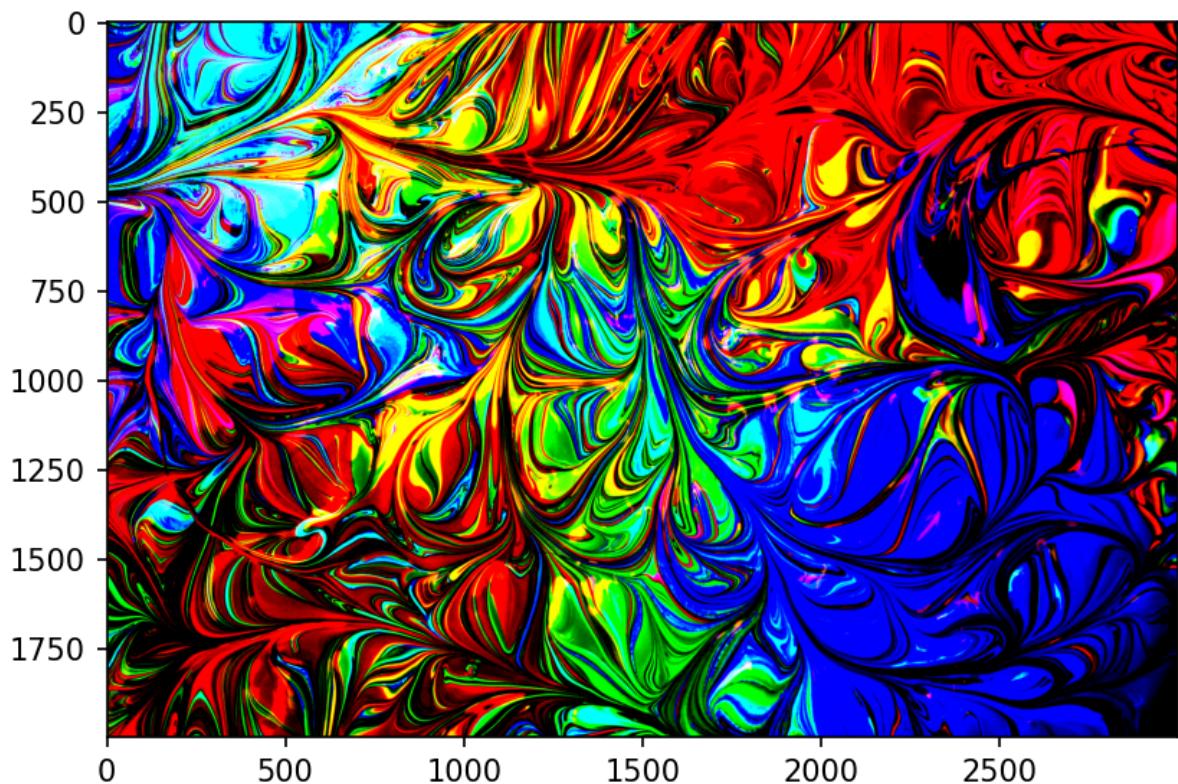
```
71.21748 72.728905
```

### Cell 22: ■ Code

```
norm = (f32 - meanval)/stdval  
plt.imshow(norm)
```

## Output:

```
&lt;matplotlib.image.AxesImage object at 0x0000029582402390&gt;  
[ STDERR ]  
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for
```



## Cell 23: ■ Markdown

### Question 9

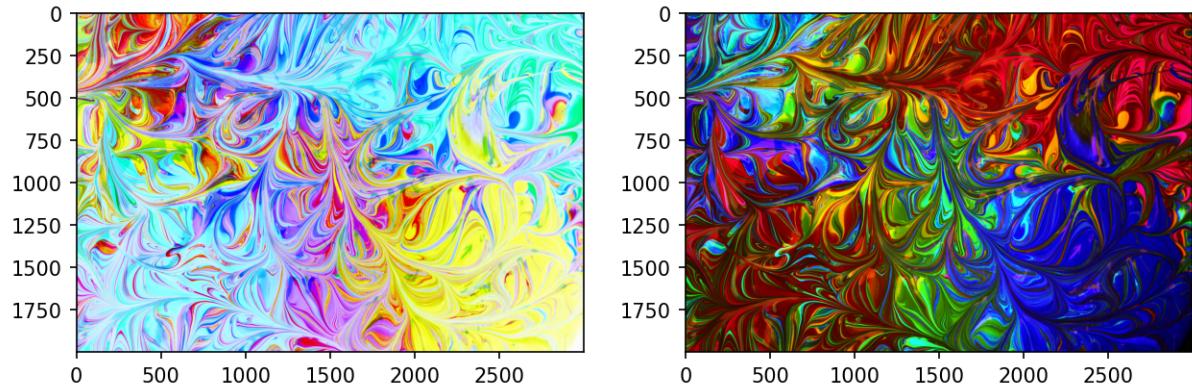
## Cell 24: ■ Code

```
nimg = X.copy()  
  
dim = X.shape  
  
for i in range(dim[0]):  
    for j in range(dim[1]):  
        nimg[i, j] = 255 - nimg[i, j]  
  
plt.figure(figsize=(10, 6))
```

```
plt.subplot(1, 2, 1)
plt.imshow(nimg)
plt.subplot(1, 2, 2)
plt.imshow(X)
```

### Output:

```
<matplotlib.image.AxesImage object at 0x00000295825380D0>;
```



### Cell 25: ■ Markdown

#### Question 10

### Cell 26: ■ Code

```
nimg = X.copy()

dim = X.shape

for i in range(dim[0]):
    for j in range(dim[1]):
        if np.any(nimg[i, j] > 127):
            nimg[i, j] = [255, 255, 255]
        else:
            nimg[i, j] = [0, 0, 0]

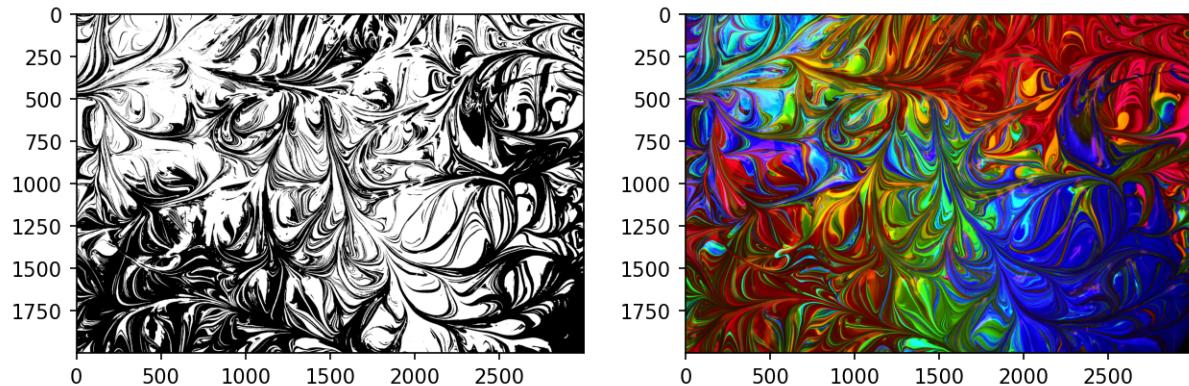
plt.figure(figsize=(10, 6))

plt.subplot(1, 2, 1)
plt.imshow(nimg)

plt.subplot(1, 2, 2)
plt.imshow(X)
```

**Output:**

```
&lt;matplotlib.image.AxesImage object at 0x00000295F6894910&gt;
```



**Cell 27: ■ Code**