In [1]:
```
#Import Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import cv2
from IPython.display import Image
```

In [2]:
```
#Load The Image
img = cv2.imread("cuteCat.png", 0) #0 for Grayscale Image
img_Color = cv2.imread("cuteCat.png", 1) #1 for Colored Image. This is the Default Output
img_Alpha = cv2.imread("cuteCat.png", -1) #-1 for Alpha Channel Image
```

In [3]:
```
print("Image Shape is: ", img.shape)
```

```
Image Shape is:  (695, 700)
```

In [4]:
```
print("Image Shape is: ", img_Color.shape)
```

```
Image Shape is:  (695, 700, 3)
```

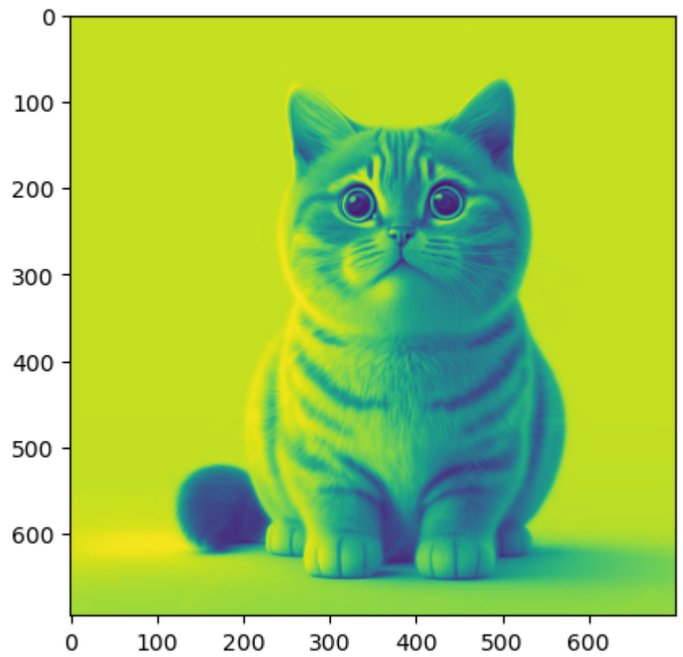
In [5]:
```
print("Image Shape is: ", img_Alpha.shape)
```

```
Image Shape is:  (695, 700, 3)
```

In [6]:
```
#Open Image with cv2
Image = cv2.namedWindow("Image")
cv2.imshow("Image", img)
cv2.waitKey(4000)
cv2.destroyAllWindows()

#Open Image with cv2
Color_Image = cv2.namedWindow("Color Image")
cv2.imshow("Color Image", img_Color)
cv2.waitKey(7000)
cv2.destroyAllWindows()

#Open Image with cv2
Alpha_Image = cv2.namedWindow("Alpha Image")
cv2.imshow("Alpha Image", img_Alpha)
cv2.waitKey(9000)
cv2.destroyAllWindows()

#3 Images will open in 3 different windows
```

In [7]:
```
#Open Image with Matplotlib
plt.imshow(img)
```

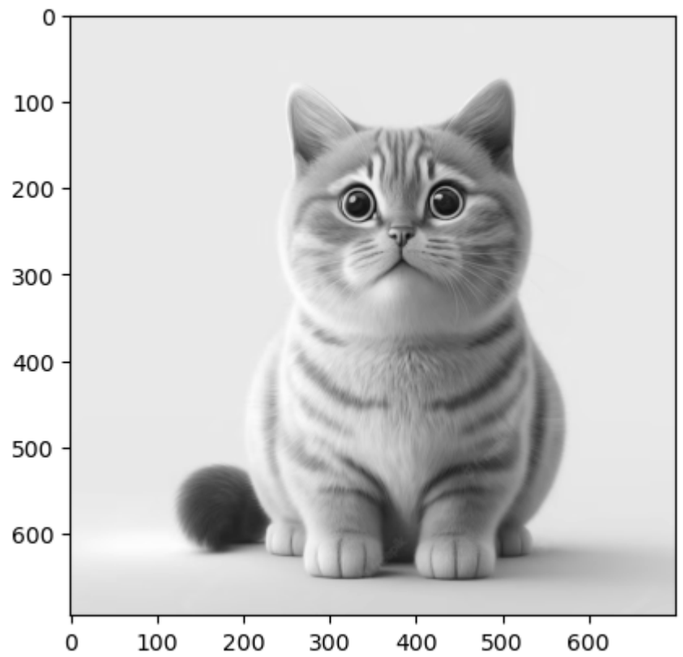
Out[7]: <matplotlib.image.AxesImage at 0x19ee2da74f0>



What Happened? Even though the image was read in as a grayscale image, it won't necessarily display in gray scale when using imshow(). Cause, Matplotlib uses different color maps and it's possible that the gray sale color map is not set.

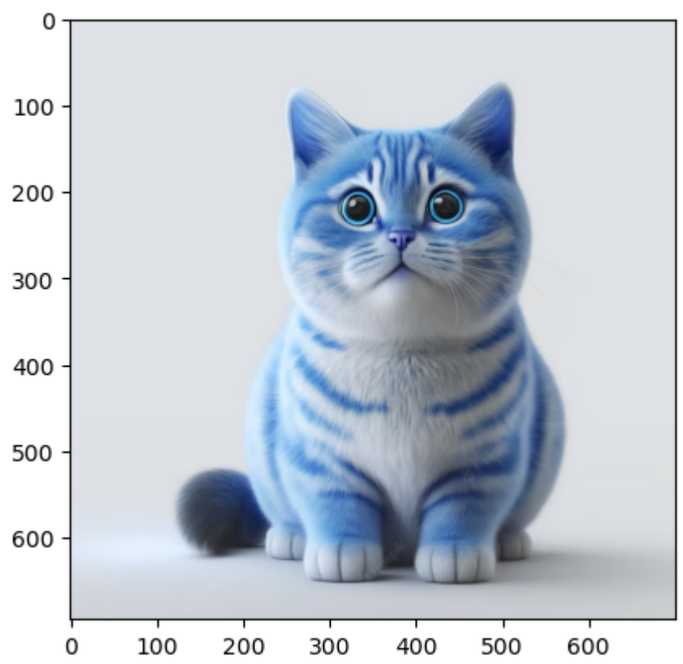
In [8]:
```
#Using Colormap Gray to get Gray Scale Image
plt.imshow(img, cmap = "gray")
```

Out[8]: <matplotlib.image.AxesImage at 0x19ee2fece50>



In [9]:
```
#Open Image with Matplotlib
plt.imshow(img_Color)
```

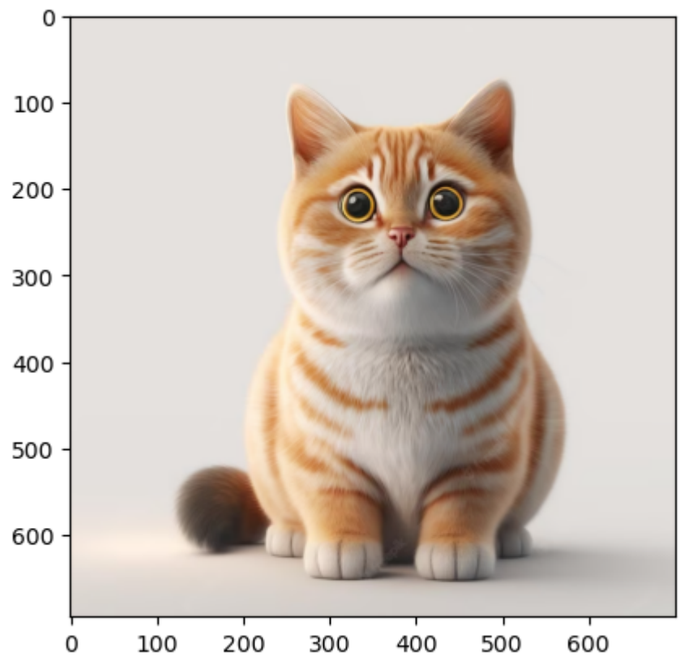
Out[9]: <matplotlib.image.AxesImage at 0x19ee30681c0>



What Happened? The color displayed above is different from actual image. This is because Matplotlib expects the image in RGB format whereas OpenCV Stores image in BGR format. Thus, for correct display, we need to reverse the channels of the image.

In [10]:
```
#To Get the Colored Image Using Matplotlib, Reverse the Channel of Image
img_Color_Channel_reveresed = cv2.cvtColor(img_Color, cv2.COLOR_BGR2RGB) #or Using -> img_Color[:, :, ::-1]
plt.imshow(img_Color_Channel_reveresed)
```

Out[10]:
```
<matplotlib.image.AxesImage at 0x19ee30c9d90>
```

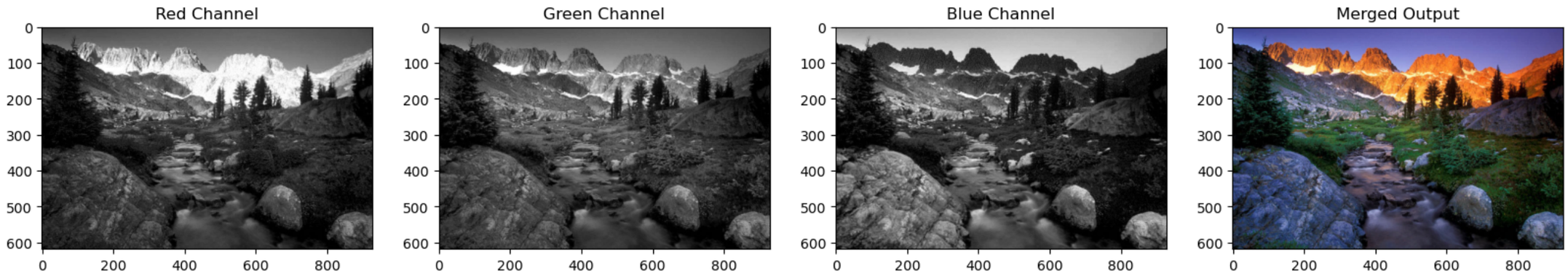


Splitting and Merging Color Channels

In [11]:
```
#cv2.split() --> Divides a Multichannel Array into Several Single Channel Arrays
#cv2.merge() --> Merges Several Arrays to Make a Single Multi-Channel Array. All The Input Matrices Must Have The Same Size.
```

In [12]:
```
#Split the Image into BGR Components
LandscapeImg = cv2.imread("LandscapeImg.png", cv2.IMREAD_COLOR) #Or, Use 1 to Get Colores Image
b,g,r = cv2.split(LandscapeImg) #Split The Image Into BGR Channel & Store to b,g,r Respectively

#Show The Channels
plt.figure(figsize = (20,5))

plt.subplot(141)
plt.imshow(r, cmap = 'gray')
plt.title("Red Channel")

plt.subplot(142)
plt.imshow(g, cmap = 'gray')
plt.title("Green Channel")

plt.subplot(143)
plt.imshow(b, cmap = 'gray')
plt.title("Blue Channel")

#Merge The Individual Channels into a BGR Image
imgMerged = cv2.merge((b,g,r))

#Show The Merged Output
plt.subplot(144)
plt.imshow(cv2.cvtColor(imgMerged, cv2.COLOR_BGR2RGB)) #or Using -> imgMerged[:, :, ::-1]
plt.title("Merged Output")

#Show Image Shape
print("Landscape Image Shape", LandscapeImg.shape)
```

```
Landscape Image Shape (619, 928, 3)
```



Converting Image To Different Color Spaces

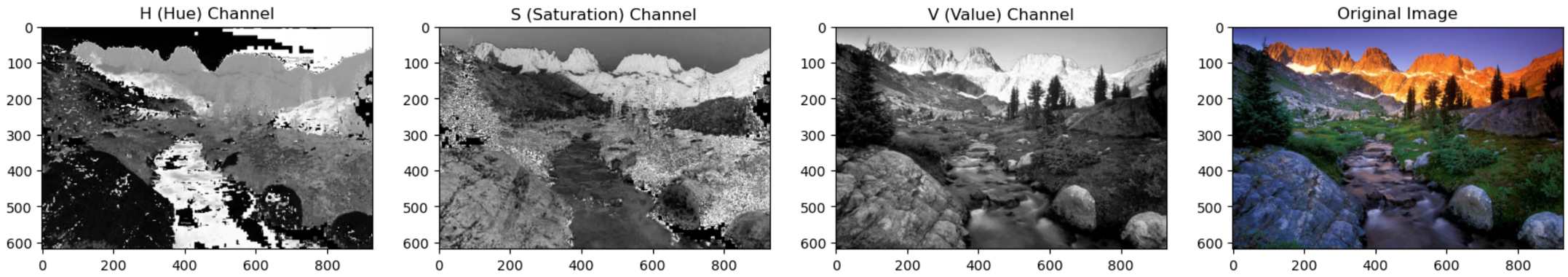
In [13]:
```
#cv2.cvtColor() -->
#Converts an Image From One Space to Another. THe Default COlor Format is RGB but actually BRG(Bytes are reversed).
#This function Takes Two Arguements (1: Source of the image, 2:Color Conversion Codes)
```

In [14]:
```
#Changing from BRG to RGB
LandscapeImg = cv2.cvtColor(LandscapeImg, cv2.COLOR_BGR2RGB)
plt.imshow(LandscapeImg)
```

Out[14]: <matplotlib.image.AxesImage at 0x19ee4703460>

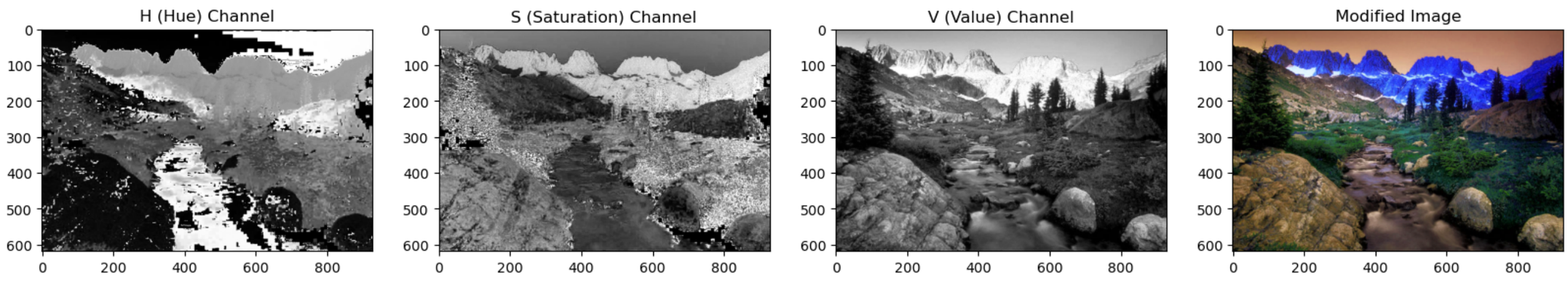
In [15]:
```
#Changing to HSV (Hue, Saturation, Value) Color Space
Landscape_HSV_Img = cv2.cvtColor(LandscapeImg, cv2.COLOR_BGR2HSV) #Or, Use 1 to Get Colores Image
h,s,v = cv2.split(Landscape_HSV_Img) #Split The Image Into HSV Channel & Store to h,s,v Respectively

#Show The Channels
plt.figure(figsize = (20,5))

plt.subplot(141)
plt.imshow(h, cmap = 'gray')
plt.title("H (Hue) Channel")

plt.subplot(142)
plt.imshow(s, cmap = 'gray')
plt.title("S (Saturation) Channel")

plt.subplot(143)
plt.imshow(v, cmap = 'gray')
plt.title("V (Value) Channel")

#Show The Original Image
plt.subplot(144)
plt.imshow(LandscapeImg)
plt.title("Original Image")
```

Out[15]: Text(0.5, 1.0, 'Original Image')



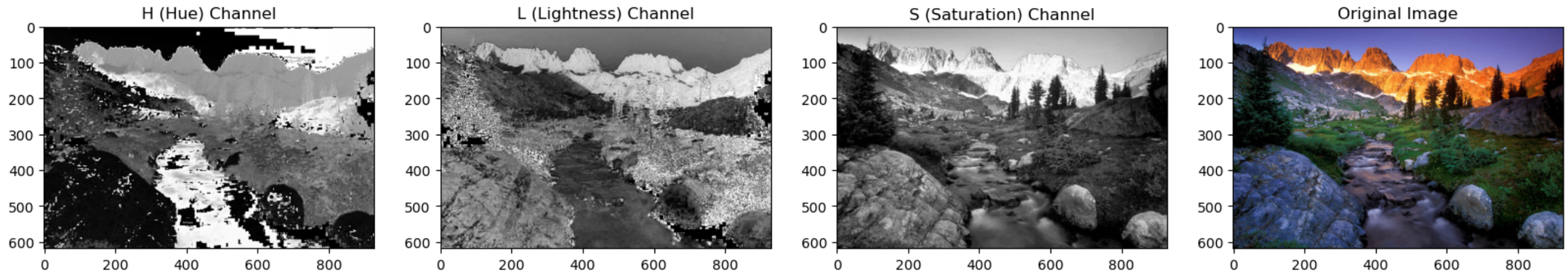
In [16]:
```
#Modifying Individual Channel
h_new = h + 10
imgMerged = cv2.merge((h_new,s,v)) #Merge The Image Into New HSV Channel Where h = h_new & Store to h,s,v Respectively
Landscape_RGB = cv2.cvtColor(imgMerged, cv2.COLOR_HSV2RGB) #Or, Use 1 to Get Colores Image

#Show The Channels
plt.figure(figsize = (20,5))

plt.subplot(141)
plt.imshow(h, cmap = 'gray')
plt.title("H (Hue) Channel")

plt.subplot(142)
plt.imshow(s, cmap = 'gray')
plt.title("S (Saturation) Channel")

plt.subplot(143)
plt.imshow(v, cmap = 'gray')
plt.title("V (Value) Channel")

#Show The Modified Image
plt.subplot(144)
plt.imshow(Landscape_RGB)
plt.title("Modified Image")
```

Out[16]: Text(0.5, 1.0, 'Modified Image')



In [17]:
```
#Changing to HSL (Hue, Saturation, Lightness) Color Space
Landscape_HLS_Img = cv2.cvtColor(LandscapeImg, cv2.COLOR_BGR2HLS) #Or, Use 1 to Get Colores Image
h,l,s = cv2.split(Landscape_HSV_Img) #Split The Image Into HLS Channel & Store to h,l,s Respectively

#Show The Channels
plt.figure(figsize = (20,5))

plt.subplot(141)
plt.imshow(h, cmap = 'gray')
plt.title("H (Hue) Channel")

plt.subplot(142)
plt.imshow(l, cmap = 'gray')
plt.title("L (Lightness) Channel")

plt.subplot(143)
plt.imshow(s, cmap = 'gray')
plt.title("S (Saturation) Channel")

#Show The Original Image
plt.subplot(144)
plt.imshow(LandscapeImg)
plt.title("Original Image")
```

Out[17]: Text(0.5, 1.0, 'Original Image')



Saving Images

In [18]:
```
#cv2.imwrite() -->
#This function Saves the Image to the Specified File. (jpg, jpeg, png)
#This function Takes Two Arguements (1: Path/File Name, 2:Image Object)
```

In [19]:
```
#Save the Image
cv2.imwrite("Landscape.png", Landscape_RGB) #Image Saved to the Directory
```

Out[19]: True