In [17]:
```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
Image=Image.open(r'C:\Users\LENOVO\Downloads\Image.jpg')
Image
```

Out[17]:

In [18]:
```
type(Image)
```

Out[18]:
```
PIL.JpegImagePlugin.JpegImageFile
```

In [19]:
```
Image=np.asarray(Image)
Image
```

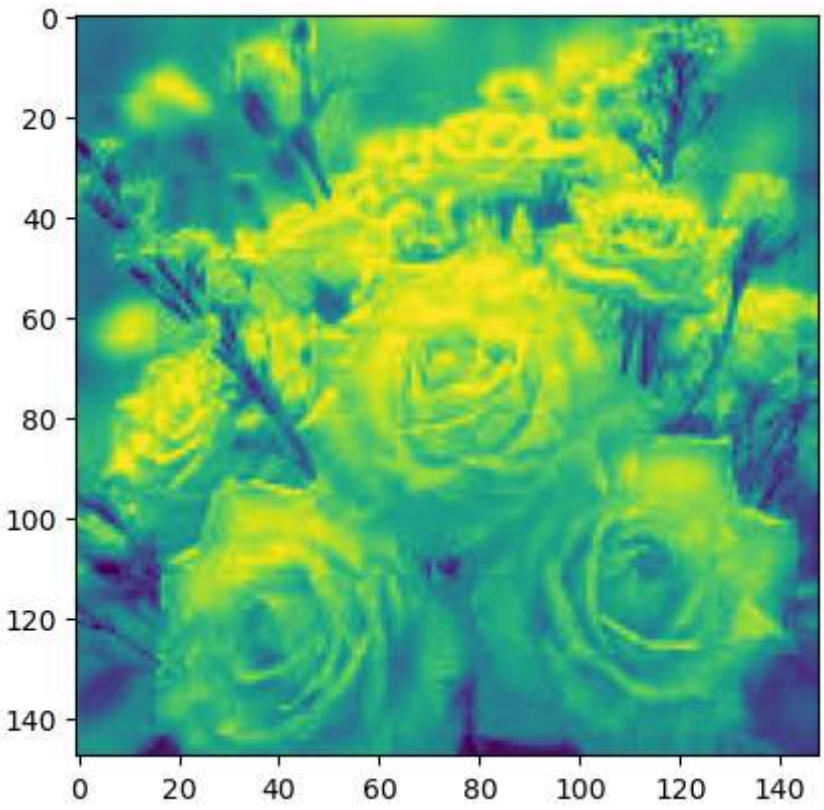
Out[19]:
```
array([[[101, 133, 172],
        [101, 133, 172],
        [100, 134, 172],
        ...,
        [178, 123, 165],
        [174, 119, 161],
        [170, 115, 157]],

       [[101, 133, 172],
        [101, 133, 172],
        [100, 134, 172],
        ...,
        [178, 125, 167],
        [173, 120, 162],
        [170, 117, 159]],

       [[102, 131, 171],
        [102, 131, 171],
        [100, 132, 171],
        ...,
        [174, 123, 164],
        [170, 119, 160],
        [167, 116, 157]],

       ...,

       [[133, 111, 114],
        [133, 110, 116],
        [133, 110, 118],
        ...,
        [ 53,  83,  81],
        [ 52,  82,  80],
        [ 51,  81,  79]],

       [[130, 108, 111],
        [130, 107, 113],
        [130, 107, 115],
        ...,
        [ 48,  80,  77],
        [ 47,  79,  76],
        [ 45,  77,  74]],

       [[125, 105, 107],
        [127, 104, 110],
        [128, 105, 113],
        ...,
        [ 44,  79,  75],
        [ 42,  77,  73],
        [ 41,  76,  72]]], dtype=uint8)
```

In [20]:
```
plt.imshow(Image)
plt.show()
```

In [21]:
```
Image.shape
```

Out[21]:
```
(148, 148, 3)
```

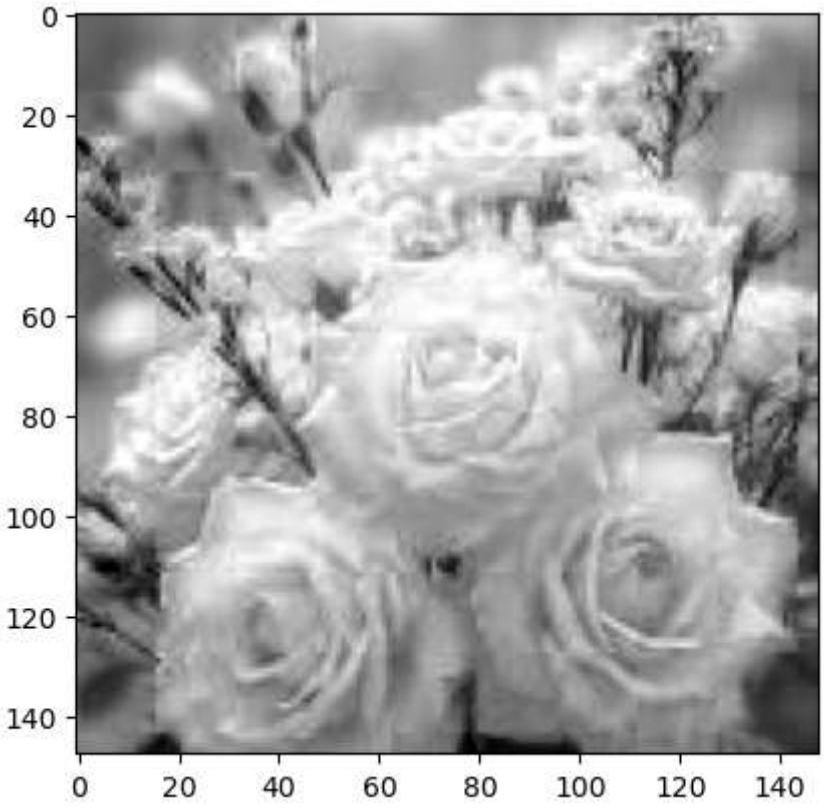
In [22]:
```
plt.imshow(Image[:,:,0])
plt.show()
```



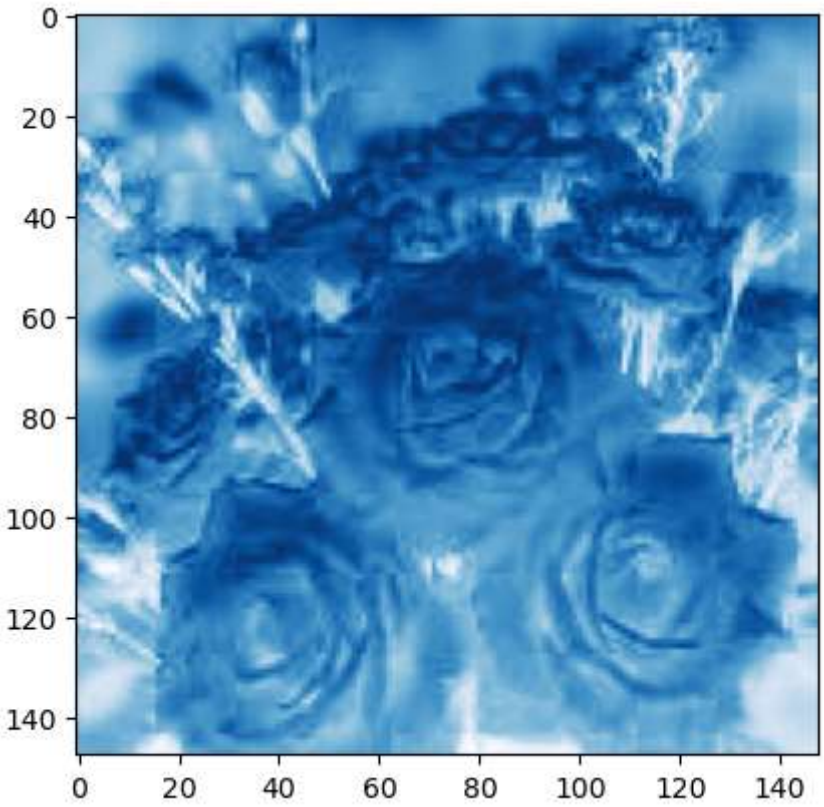
In [23]:
```
Image[:,:,0]
```

Out[23]:
```
array([[101, 101, 100, ..., 178, 174, 170],
       [101, 101, 100, ..., 178, 173, 170],
       [102, 102, 100, ..., 174, 170, 167],
       ...,
       [133, 133, 133, ...,  53,  52,  51],
       [130, 130, 130, ...,  48,  47,  45],
       [125, 127, 128, ...,  44,  42,  41]], dtype=uint8)
```

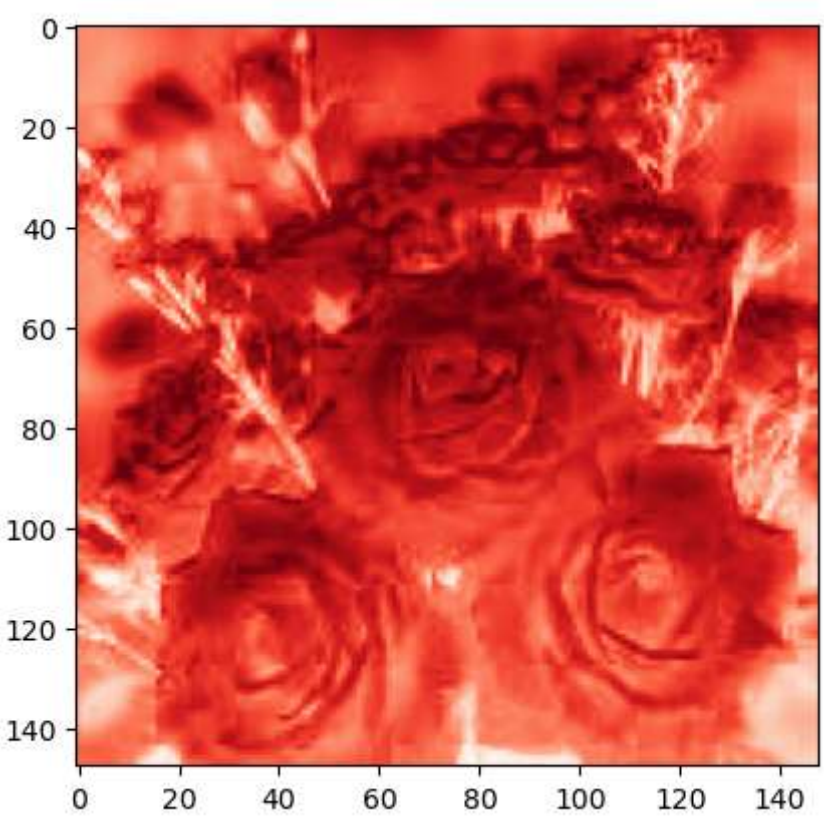
In [24]:
```
plt.imshow(Image[:,:,0],cmap='gray')
plt.show()
```



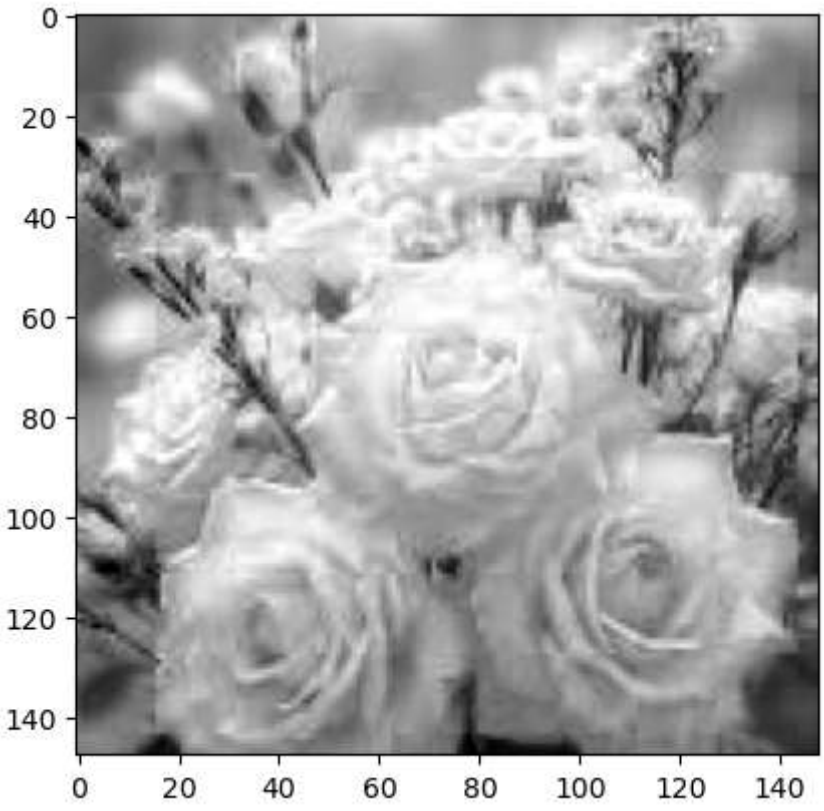
In [25]:
```
plt.imshow(Image[:,:,0],cmap='Blues')
plt.show()
```



In [26]:
```
plt.imshow(Image[:,:,0],cmap='Reds')
plt.show()
```



In [27]:
```
plt.imshow(Image[:,:,0],cmap='gray')
plt.show()
```



In [28]:
```
Image[:,:,0]
```

Out[28]:
```
array([[101, 101, 100, ..., 178, 174, 170],
       [101, 101, 100, ..., 178, 173, 170],
       [102, 102, 100, ..., 174, 170, 167],
       ...,
       [133, 133, 133, ...,  53,  52,  51],
       [130, 130, 130, ...,  48,  47,  45],
       [125, 127, 128, ...,  44,  42,  41]], dtype=uint8)
```

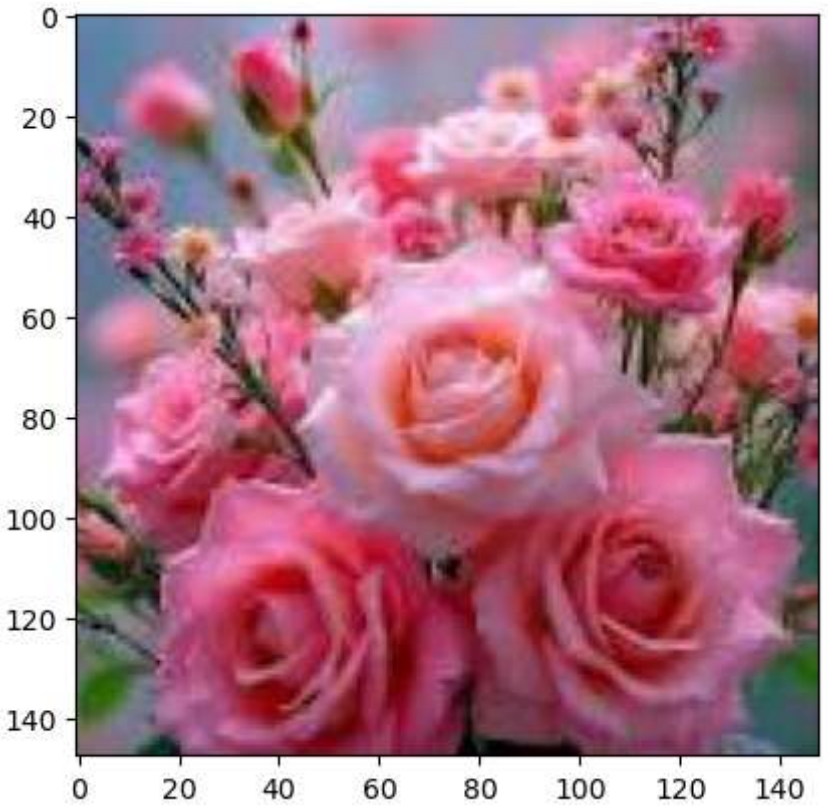
In [29]:
```
Image[:,:,1]
```

Out[29]:
```
array([[133, 133, 134, ..., 123, 119, 115],
       [133, 133, 134, ..., 125, 120, 117],
       [131, 131, 132, ..., 123, 119, 116],
       ...,
       [111, 110, 110, ...,  83,  82,  81],
       [108, 107, 107, ...,  80,  79,  77],
       [105, 104, 105, ...,  79,  77,  76]], dtype=uint8)
```

In [30]:
```
Image[:,:,2]
```

Out[30]:
```
array([[172, 172, 172, ..., 165, 161, 157],
       [172, 172, 172, ..., 167, 162, 159],
       [171, 171, 171, ..., 164, 160, 157],
       ...,
       [114, 116, 118, ...,  81,  80,  79],
       [111, 113, 115, ...,  77,  76,  74],
       [107, 110, 113, ...,  75,  73,  72]], dtype=uint8)
```

In [31]:
```
plt.imshow(Image)
plt.show()
```

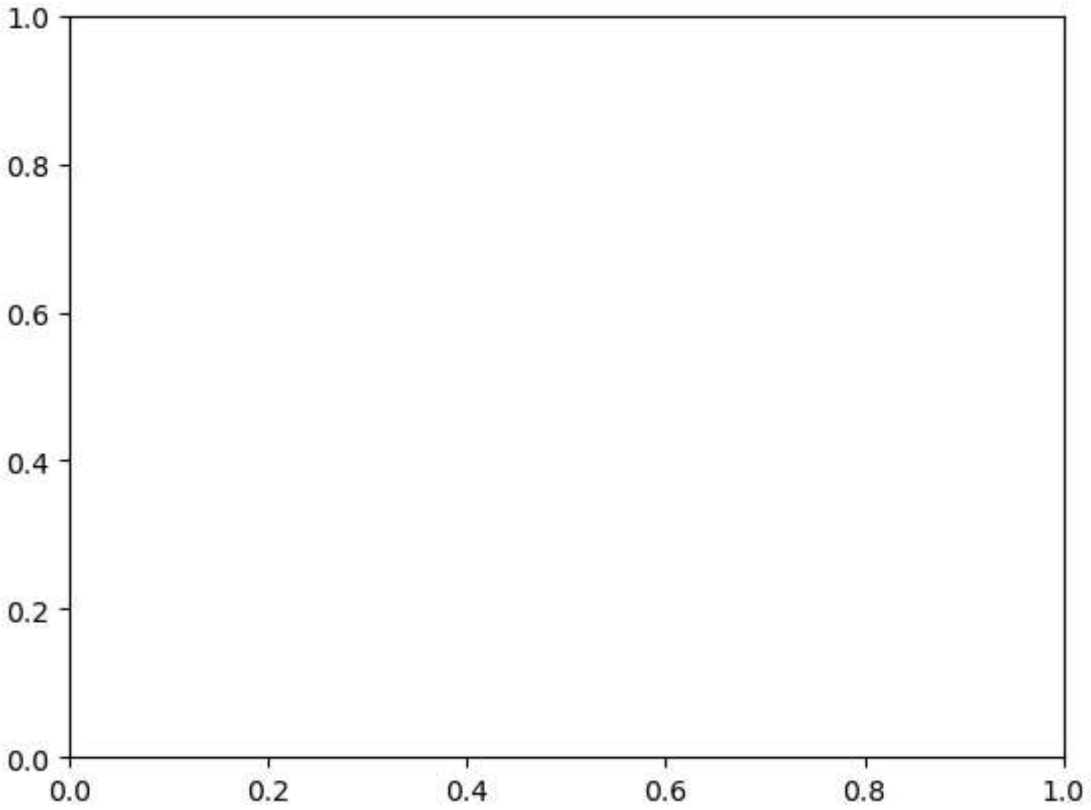


In [37]:
```
plt.imshow(Image[:,:,1],cmap='G')
plt.show()
```

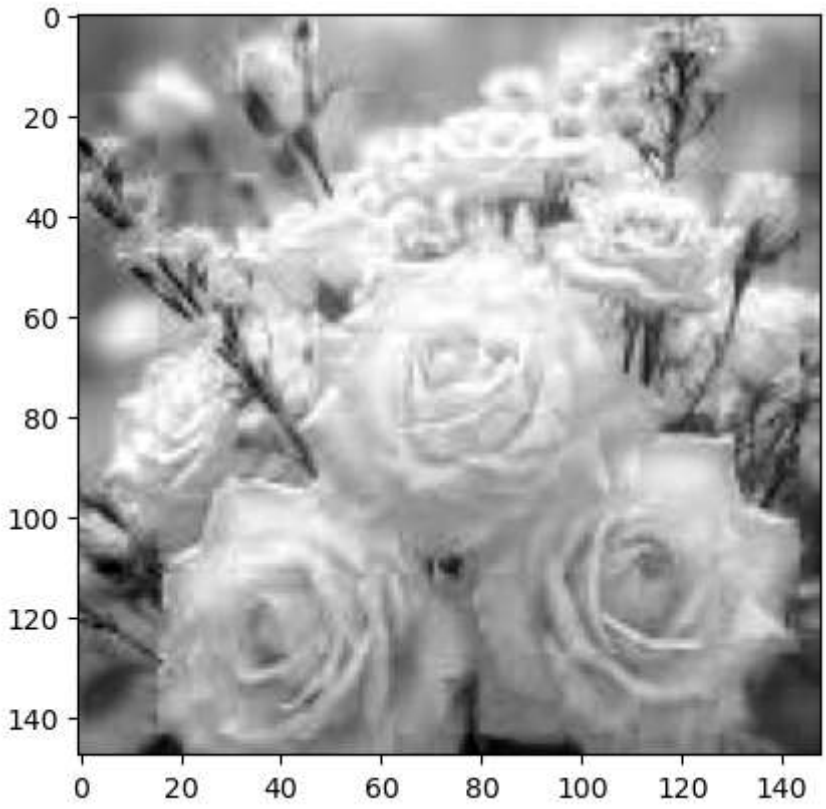
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[37], line 1
----> 1 plt.imshow(Image[:,:,1],cmap='Red')
      2 plt.show()

File ~\anaconda3\Lib\site-packages\matplotlib\pyplot.py:3358, in imshow(X, cmap, nor
m, aspect, interpolation, alpha, vmin, vmax, origin, extent, interpolation_stage, fi
lternorm, filterrad, resample, url, data, **kwargs)
   3337 @_copy_docstring_and_deprecators(Axes.imshow)
   3338 def imshow(
   3339     X: ArrayLike | PIL.Image.Image,
   (...)
   3356     **kwargs,
   3357 ) -> AxesImage:
-> 3358     __ret = gca().imshow(
   3359         X,
   3360         cmap=cmap,
   3361         norm=norm,
   3362         aspect=aspect,
   3363         interpolation=interpolation,
   3364         alpha=alpha,
   3365         vmin=vmin,
   3366         vmax=vmax,
   3367         origin=origin,
   3368         extent=extent,
   3369         interpolation_stage=interpolation_stage,
   3370         filternorm=filternorm,
   3371         filterrad=filterrad,
   3372         resample=resample,
   3373         url=url,
   3374         **({"data": data} if data is not None else {}),
   3375         **kwargs,
   3376     )
   3377     sci(__ret)
   3378     return __ret

File ~\anaconda3\Lib\site-packages\matplotlib\__init__.py:1465, in _preprocess_data.
<locals>.inner(ax, data, *args, **kwargs)
   1462 @functools.wraps(func)
   1463 def inner(ax, *args, data=None, **kwargs):
   1464     if data is None:
-> 1465         return func(ax, *map(sanitize_sequence, args), **kwargs)
   1467     bound = new_sig.bind(ax, *args, **kwargs)
   1468     auto_label = (bound.arguments.get(label_namer)
   1469                   or bound.kwargs.get(label_namer))

File ~\anaconda3\Lib\site-packages\matplotlib\axes\_axes.py:5745, in Axes.imshow(sel
f, X, cmap, norm, aspect, interpolation, alpha, vmin, vmax, origin, extent, interpol
ation_stage, filternorm, filterrad, resample, url, **kwargs)
   5540 @_preprocess_data()
   5541 @_docstring.interpd
   5542 def imshow(self, X, cmap=None, norm=None, *, aspect=None,
   (...)
   5545            interpolation_stage=None, filternorm=True, filterrad=4.0,
   5546            resample=None, url=None, **kwargs):
```

```
   5547     """
   5548     Display data as an image, i.e., on a 2D regular raster.
   5549
   (...)
   5743     (unassociated) alpha representation.
   5744     """
-> 5745     im = mimage.AxesImage(self, cmap=cmap, norm=norm,
   5746                           interpolation=interpolation, origin=origin,
   5747                           extent=extent, filternorm=filternorm,
   5748                           filterrad=filterrad, resample=resample,
   5749                           interpolation_stage=interpolation_stage,
   5750                           **kwargs)
   5752     if aspect is None and not (
   5753             im.is_transform_set()
   5754             and not im.get_transform().contains_branch(self.transData)):
   5755         aspect = mpl.rcParams['image.aspect']

File ~\anaconda3\Lib\site-packages\matplotlib\image.py:912, in AxesImage.__init__(self, ax, cmap, norm, interpolation, origin, extent, filternorm, filterrad, resample, interpolation_stage, **kwargs)
    896 def __init__(self, ax,
    897              *,
    898              cmap=None,
   (...)
    907              **kwargs
    908              ):
    910     self._extent = extent
--> 912     super().__init__(
    913         ax,
    914         cmap=cmap,
    915         norm=norm,
    916         interpolation=interpolation,
    917         origin=origin,
    918         filternorm=filternorm,
    919         filterrad=filterrad,
    920         resample=resample,
    921         interpolation_stage=interpolation_stage,
    922         **kwargs
    923     )

File ~\anaconda3\Lib\site-packages\matplotlib\image.py:261, in _ImageBase.__init__(self, ax, cmap, norm, interpolation, origin, filternorm, filterrad, resample, interpolation_stage, **kwargs)
    248 def __init__(self, ax,
    249              cmap=None,
    250              norm=None,
   (...)
    258              **kwargs
    259              ):
    260     martist.Artist.__init__(self)
--> 261     cm.ScalarMappable.__init__(self, norm, cmap)
    262     if origin is None:
    263         origin = mpl.rcParams['image.origin']

File ~\anaconda3\Lib\site-packages\matplotlib\cm.py:416, in ScalarMappable.__init__(self, norm, cmap)
```

```
    414 self.set_norm(norm)  # The Normalize instance of this ScalarMappable.
    415 self.cmap = None  # So that the setter knows we're initializing.
--> 416 self.set_cmap(cmap)  # The Colormap instance of this ScalarMappable.
    417 #: The last colorbar associated with this ScalarMappable. May be None.
    418 self.colorbar = None

File ~\anaconda3\Lib\site-packages\matplotlib\cm.py:605, in ScalarMappable.set_cmap
(self, cmap)
    596 """
    597 Set the colormap for luminance data.
    598
   (...)
    601 cmap : `.Colormap` or str or None
    602 """
    603 in_init = self.cmap is None
--> 605 self.cmap = _ensure_cmap(cmap)
    606 if not in_init:
    607     self.changed()

File ~\anaconda3\Lib\site-packages\matplotlib\cm.py:744, in _ensure_cmap(cmap)
    741 # use check_in_list to ensure type stability of the exception raised by
    742 # the internal usage of this (ValueError vs KeyError)
    743 if cmap_name not in _colormaps:
--> 744     _api.check_in_list(sorted(_colormaps), cmap=cmap_name)
    745 return mpl.colormaps[cmap_name]

File ~\anaconda3\Lib\site-packages\matplotlib\_api\__init__.py:129, in check_in_list
(values, _print_supported_values, **kwargs)
    127 if _print_supported_values:
    128     msg += f"; supported values are {', '.join(map(repr, values))}"
--> 129 raise ValueError(msg)

ValueError: 'Red' is not a valid value for cmap; supported values are 'Accent', 'Acc
ent_r', 'Blues', 'Blues_r', 'BrBG', 'BrBG_r', 'BuGn', 'BuGn_r', 'BuPu', 'BuPu_r', 'C
MRmap', 'CMRmap_r', 'Dark2', 'Dark2_r', 'GnBu', 'GnBu_r', 'Grays', 'Greens', 'Greens
_r', 'Greys', 'Greys_r', 'OrRd', 'OrRd_r', 'Oranges', 'Oranges_r', 'PRGn', 'PRGn_r',
'Paired', 'Paired_r', 'Pastel1', 'Pastel1_r', 'Pastel2', 'Pastel2_r', 'PiYG', 'PiYG_
r', 'PuBu', 'PuBuGn', 'PuBuGn_r', 'PuBu_r', 'PuOr', 'PuOr_r', 'PuRd', 'PuRd_r', 'Pur
ples', 'Purples_r', 'RdBu', 'RdBu_r', 'RdGy', 'RdGy_r', 'RdPu', 'RdPu_r', 'RdYlBu',
'RdYlBu_r', 'RdYlGn', 'RdYlGn_r', 'Reds', 'Reds_r', 'Set1', 'Set1_r', 'Set2', 'Set2_
r', 'Set3', 'Set3_r', 'Spectral', 'Spectral_r', 'Wistia', 'Wistia_r', 'YlGn', 'YlGnB
u', 'YlGnBu_r', 'YlGn_r', 'YlOrBr', 'YlOrBr_r', 'YlOrRd', 'YlOrRd_r', 'afmhot', 'afm
hot_r', 'autumn', 'autumn_r', 'binary', 'binary_r', 'bone', 'bone_r', 'brg', 'brg_
r', 'bwr', 'bwr_r', 'cividis', 'cividis_r', 'cool', 'cool_r', 'coolwarm', 'coolwarm_
r', 'copper', 'copper_r', 'cubehelix', 'cubehelix_r', 'flag', 'flag_r', 'gist_eart
h', 'gist_earth_r', 'gist_gray', 'gist_gray_r', 'gist_grey', 'gist_heat', 'gist_heat
_r', 'gist_ncar', 'gist_ncar_r', 'gist_rainbow', 'gist_rainbow_r', 'gist_stern', 'gi
st_stern_r', 'gist_yarg', 'gist_yarg_r', 'gist_yerg', 'gnuplot', 'gnuplot2', 'gnuplo
t2_r', 'gnuplot_r', 'gray', 'gray_r', 'grey', 'hot', 'hot_r', 'hsv', 'hsv_r', 'infer
no', 'inferno_r', 'jet', 'jet_r', 'magma', 'magma_r', 'nipy_spectral', 'nipy_spectra
l_r', 'ocean', 'ocean_r', 'pink', 'pink_r', 'plasma', 'plasma_r', 'prism', 'prism_
r', 'rainbow', 'rainbow_r', 'seismic', 'seismic_r', 'spring', 'spring_r', 'summer',
'summer_r', 'tab10', 'tab10_r', 'tab20', 'tab20_r', 'tab20b', 'tab20b_r', 'tab20c',
'tab20c_r', 'terrain', 'terrain_r', 'turbo', 'turbo_r', 'twilight', 'twilight_r', 't
wilight_shifted', 'twilight_shifted_r', 'viridis', 'viridis_r', 'winter', 'winter_r'
```



In [39]:
```
plt.imshow(Image[:,:,0],cmap='')
plt.show()
```



In [ ]: