

```
#!/usr/bin/python

import os
import numpy as np
import matplotlib.pyplot as plt
import cv2
import pdb

def colormapArray(X, colors):
    """
    Basically plt.imsave but return a matrix instead

    Given:
        a HxW matrix X
        a Nx3 color map of colors in [0,1] [R,G,B]
    Outputs:
        a HxW uint8 image using the given colormap. See the Bewares
    """
    H, W = X.shape
    N = colors.shape[0]

    vmin = np.nanmin(X)
    vmax = np.nanmax(X)

    if not np.any(np.isfinite(X)):
        result_img = np.zeros((H, W, 3), dtype=np.uint8)
        return result_img

    if not np.isfinite(vmin) or not np.isfinite(vmax) or vmax == vmin:
        base = (np.clip(colors[0], 0, 1) * 255).astype(np.uint8)
        return np.tile(base, (H, W, 1))

    norm = (X - vmin) / (vmax - vmin)
    norm = np.where(np.isfinite(norm), norm, 0.0)
    norm = np.clip(norm, 0.0, 1.0)

    idx = np.minimum((norm * (N - 1)).astype(np.int64), N - 1)
    rgb = colors[idx]
    result_img = np.clip(rgb * 255.0, 0, 255).astype(np.uint8)
    return result_img

if __name__ == "__main__":
    colors = np.load("mysterydata/colors.npy")
    data = np.load("mysterydata/mysterydata.npy")

    mystery_data4 = np.load("mysterydata/mysterydata4.npy")
    # print(f"Shape mystery data 4 {mystery_data4.shape}")

    for i in range(9):
        mystery4_result_img = colormapArray(mystery_data4[:, :, i], colors)
        plt.imsave("3_4/vis_%d.png" % i, mystery4_result_img, cmap='plasma')

    # pdb.set_trace()
```

