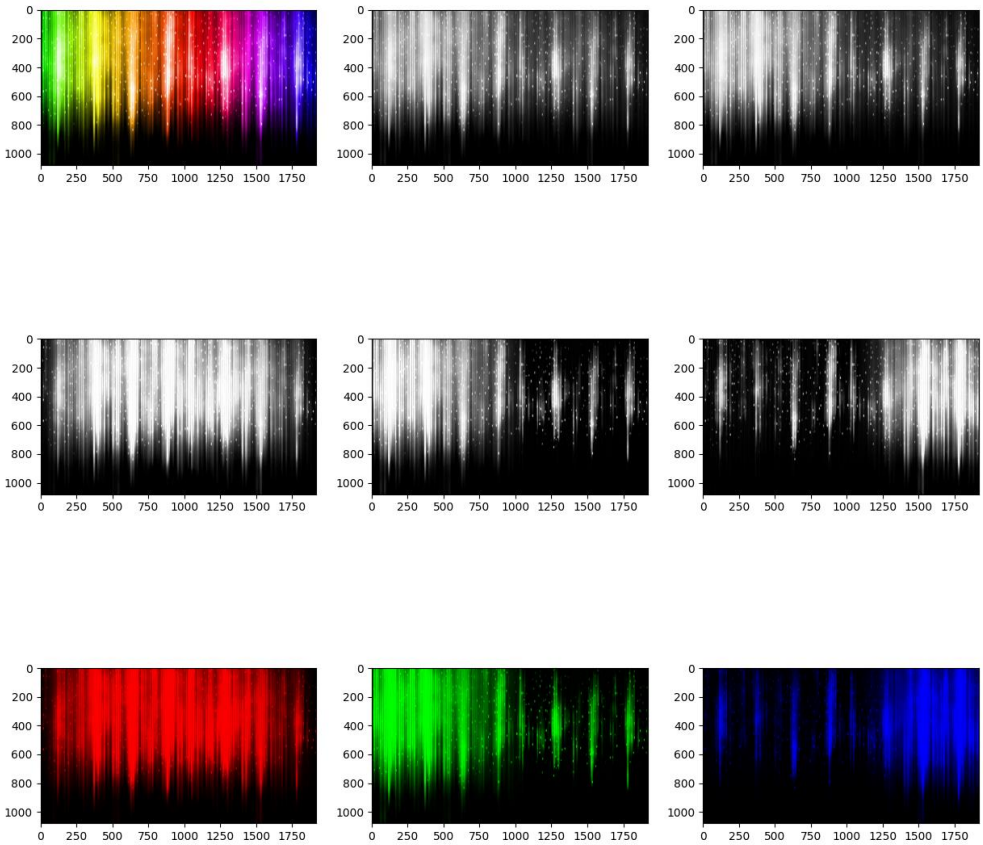
Zadanie 1

```python
def imgToUInt8(img):
    if np.issubdtype(img.dtype, np.unsignedinteger):
        return img
    elif np.issubdtype(img.dtype, np.floating):
        return (img * 255).astype(np.uint8)

    raise ValueError("Unsupported image type")


def imgToFloat(img):
    if np.issubdtype(img.dtype, np.floating):
        return img
    elif np.issubdtype(img.dtype, np.unsignedinteger):
        return img.astype(np.float32) / 255
        # return img / float(255)

    raise ValueError("Unsupported image type")
```

Zadanie 2

```python
def process_image(img):
    O = img.copy()
    R1 = img.copy()
    R2 = img.copy()
    G1 = img.copy()
    G2 = img.copy()
    B1 = img.copy()
    B2 = img.copy()

    R = img[:, :, 0]
    G = img[:, :, 1]
    B = img[:, :, 2]

    Y1 = 0.299 * R + 0.587 * G + 0.114 * B
    Y2 = 0.2126 * R + 0.7152 * G + 0.0722 * B

    R1 = R1[:, :, 0]
    G1 = G1[:, :, 1]
    B1 = B1[:, :, 2]

    R2[:, :, 1:3] = 0
    G2[:, :, [0, 2]] = 0
    B2[:, :, 0:2] = 0

    fig, axs = plt.subplots(3, 3, figsize=(15, 15))

    axs[0, 0].imshow(O)
    axs[0, 1].imshow(Y1, cmap="gray")
    axs[0, 2].imshow(Y2, cmap="gray")

    axs[1, 0].imshow(R1, cmap="gray")
    axs[1, 1].imshow(G1, cmap="gray")
    axs[1, 2].imshow(B1, cmap="gray")

    axs[2, 0].imshow(R2)
    axs[2, 1].imshow(G2)
    axs[2, 2].imshow(B2)

    return fig
```

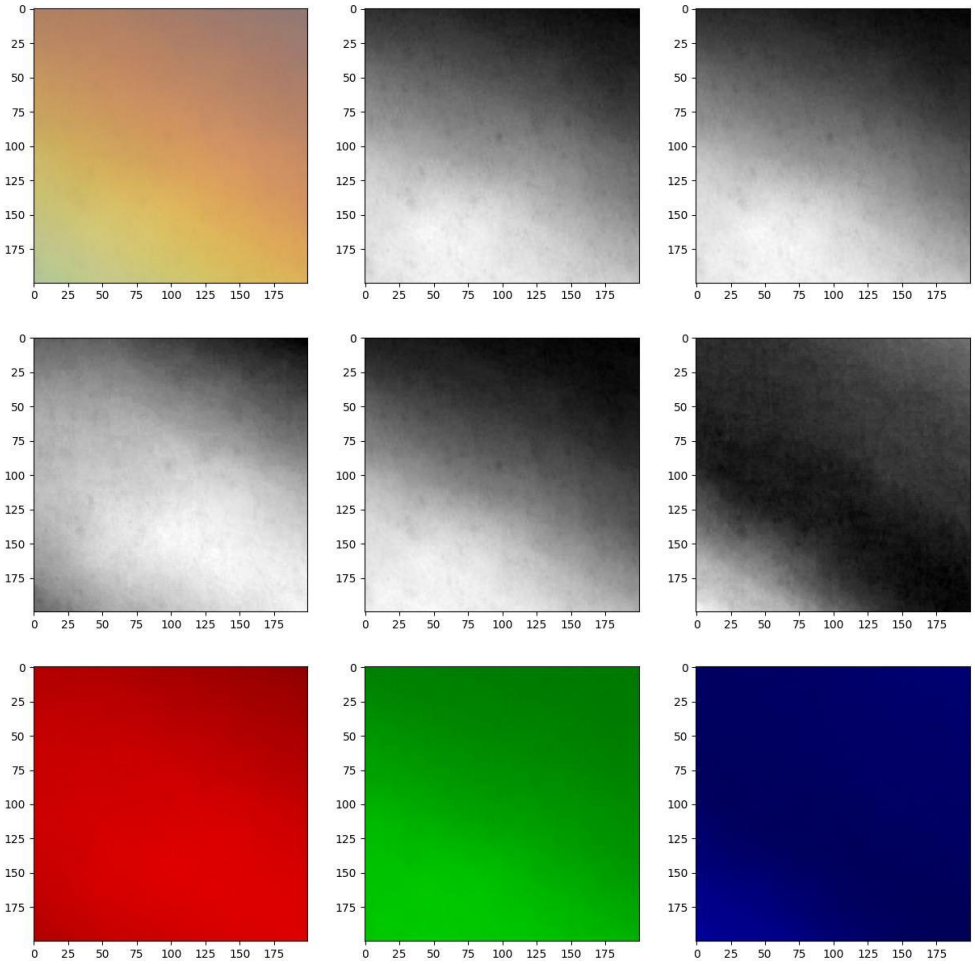
Wynik dla pliku B01.png:



Zadanie 3

```python
format = "jpg"
input_dir = "IMG_INTRO/"
output_dir = "zadanie_3_img"
df = pd.DataFrame(
    data={
        "Filename": ["B02.jpg"],
        "Grayscale": [False],
        "Fragments": [
            [[200, 200, 400, 400], [400, 400, 600, 600]],
        ],
    }
)


for index, row in df.iterrows():
    img = cv2.imread(input_dir + row["Filename"])
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    print(img.shape)

    if row["Grayscale"]:
        pass
    else:
        if row["Fragments"] is not None:
            for f in row["Fragments"]:
                fragment = img[f[0] : f[2], f[1] : f[3]].copy()

                # operacje
                fig = process_image(fragment)

                plt.savefig(
                    f"{output_dir}/{row['Filename']}_fragment_{f[0]}_{f[1]}_{f[2]}_{f[3]}.{format}"
                )

```

Wynik dla fragmentu 200-400 (200x200):



Wynik dla fragmentu 400-600 (200x200):