

Raport 10

Wiktor Krasiński ISI gr. 2

Zad. 1.

Stosujemy metodę resize do wybranego obrazu postac (postac.png).

Następnie tworzymy 6 obrazów, dla metod resamplingu. I przedstawiamy na jednym diagramie plt (*fig1.png*).



Oto kod do programu:

```
w,h = im.size
s_w = 0.15
s_h = 0.27
w0 = int(w * s_w)
h0 = int(h * s_h)

im_NEAREST = im.resize( size: (w0,h0) , resample: 0, reducing_gap = None)
diff_NEAREST = ImageChops.difference(im_NEAREST, im_NEAREST)
print("\nStatystyki dla: NEAREST vs NEAREST")
stat1 = statystyki(diff_NEAREST)

im_LANCZOS = im.resize( size: (w0,h0) , resample: 1, reducing_gap=None)
diff_LANCZOS = ImageChops.difference(im_NEAREST, im_LANCZOS)
print("\nStatystyki dla: NEAREST vs LANCZOS")
stat2 = statystyki(diff_LANCZOS)

im_BILINEAR = im.resize( size: (w0,h0) , resample: 2, reducing_gap = None)
diff_BILINEAR = ImageChops.difference(im_NEAREST, im_BILINEAR)
print("\nStatystyki dla: NEAREST vs BILINEAR")
stat3 = statystyki(diff_BILINEAR)

im_BICUBIC = im.resize( size: (w0,h0) , resample: 3, reducing_gap = None)
diff_BICUBIC = ImageChops.difference(im_NEAREST, im_BICUBIC)
print("\nStatystyki dla: NEAREST vs BICUBIC")
stat4 = statystyki(diff_BICUBIC)

im_BOX = im.resize( size: (w0,h0) , resample: 4, reducing_gap = None)
diff_BOX = ImageChops.difference(im_NEAREST, im_BOX)
print("\nStatystyki dla: NEAREST vs BOX")
stat5 = statystyki(diff_BOX)

im_HAMMING = im.resize( size: (w0,h0) , resample: 5, reducing_gap = None)
diff_HAMMING = ImageChops.difference(im_NEAREST, im_HAMMING)
print("\nStatystyki dla: NEAREST vs HAMMING")
stat6 = statystyki(diff_HAMMING)
```

Zad. 2.

a)

Stosujemy metodę resize z metodą resamplingu BILINEAR powiększamy fragment obrazu postac (*postac.jpg*). Kod:

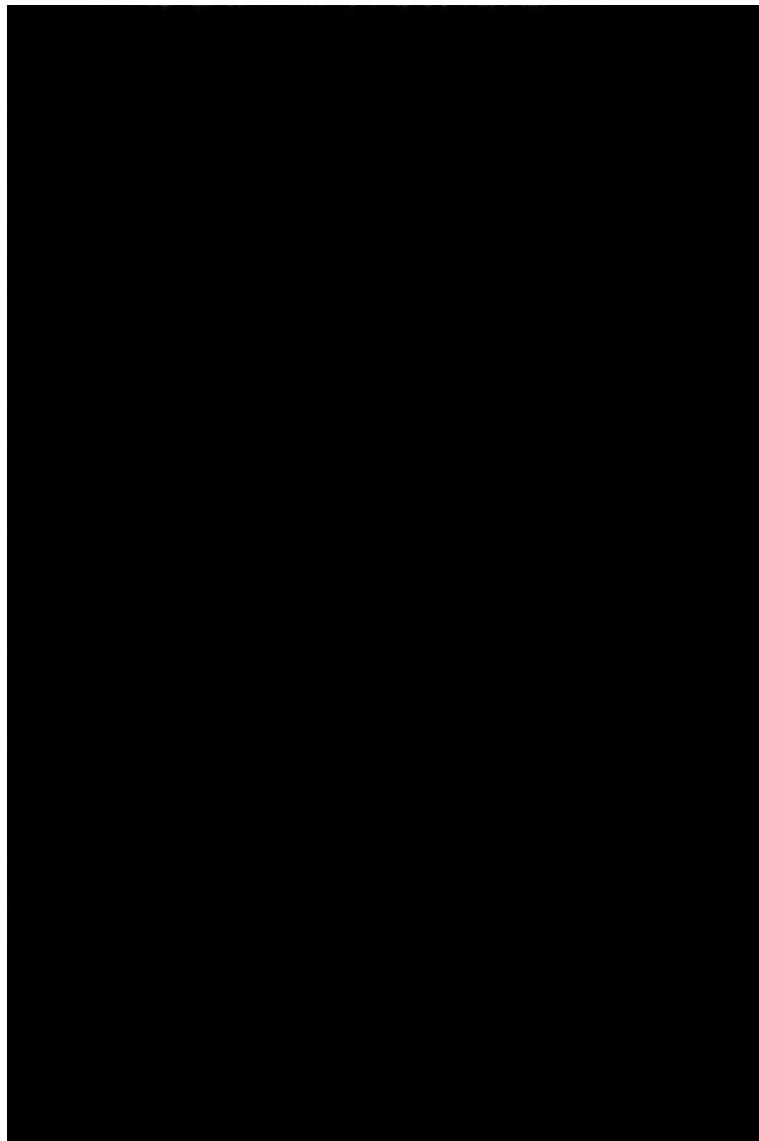
```
frag_a = im.crop(box).resize( size: (w0, h0), resample=Image.Resampling.BICUBIC)
frag_a.save("zad2a.png")
```

b)

Stosujemy metodę crop wycinając fragment obrazu postac (*postac.jpg*).

```
frag_b = im.resize( size: (w0,h0), resample=Image.Resampling.BICUBIC, box=box)
frag_b.save("zad2b.png")
```

Porównanie obrazów (*difference.png*):



Zad. 3.

a)

Obracamy obraz o 60 stopni i dopasowujemy obraz.

Kod do zadania:

```
rot_left = im.rotate(angle=60, expand=True, fillcolor=(255, 0, 0))
rot_left.save("rot_left.png")
```

Otrzymujemy następujący obraz:



b) Obracamy obraz o 60 stopni w prawo dopasowując argumenty:

```
rot_right = im.rotate(-60, expand=False, fillcolor=(0, 255, 0))
rot_right.save("rot_right.png")
```

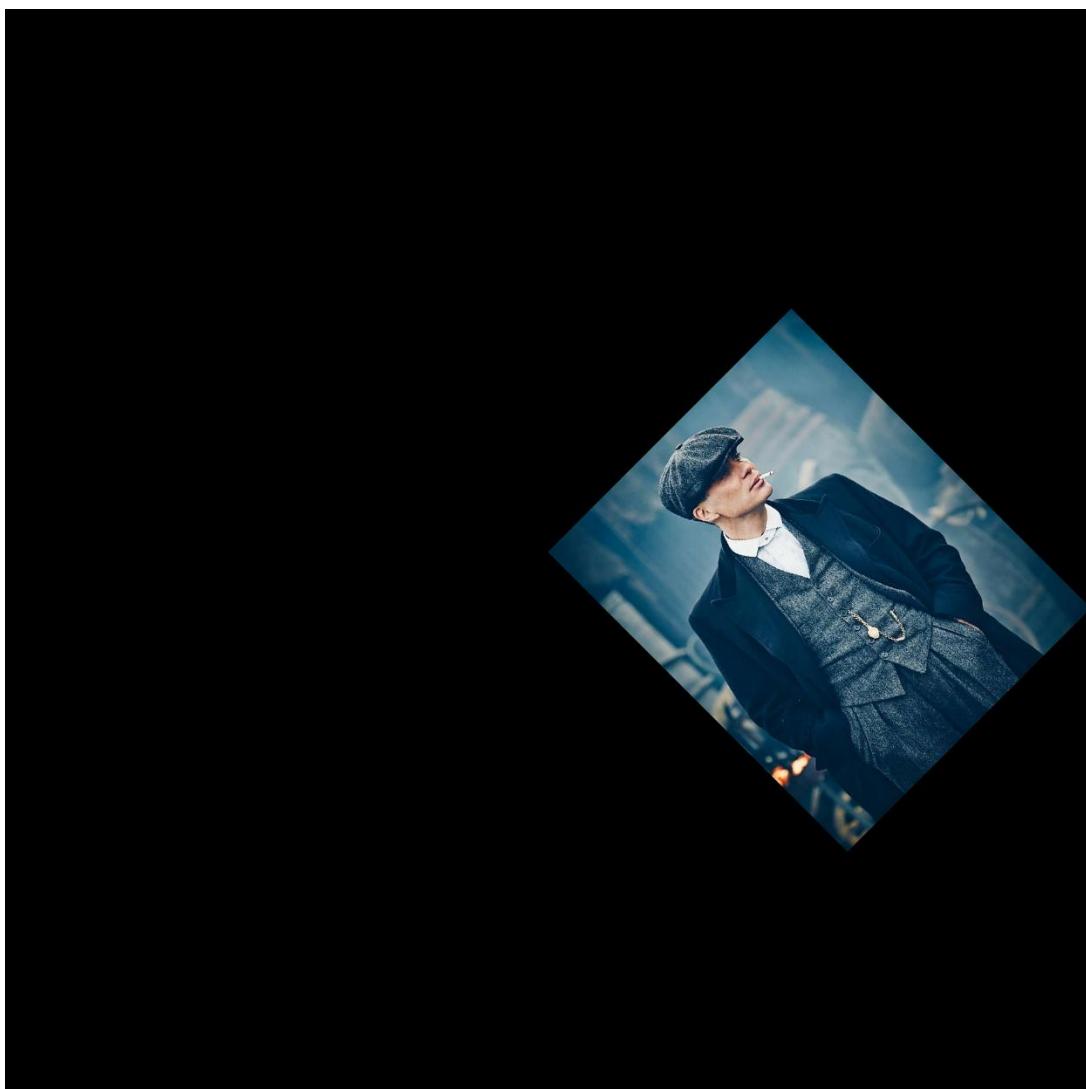


Zad. 4.

Kod do zadania, tworzymy nowy obraz o wymiarach takich, aby punkt obrotu był jego środkiem:

```
w, h = im.size  
new_size = (w * 2, h * 2)  
  
new_im = Image.new(mode: "RGB", new_size, color: (0, 0, 0))  
new_im.paste(im, box: (w, h))  
rot = new_im.rotate(angle: 45, expand=True)  
  
rot.save("obrot.png")
```

Po wykonaniu otrzymujemy obraz (*obrot.png*):



Zad. 5.

```
transpose1 = im.rotate(angle: 90, expand=True).transpose(0)
transpose2 = im.rotate(-90).transpose(0)

#transpose1.show()
#transpose2.show()
```

Odp: Tak można, uzyskujemy to dzięki obrotowi + Image flip.

Image.transpose (czyli odbicie względem głównej przekątnej)

Image.transverse (czyli odbicie względem drugiej bocznej)

Zad. 6.

Z obrazu motylek (*motylek.png*) tworzymy maskę, a następnie wklejamy estetycznie trzy motylki w różnych położeniach.

Oto kod do zadania:

```
postac = Image.open("postac.jpg").convert("RGB")
motyl_src = Image.open("motylek.png").convert("L")

maska_czysta = ImageOps.invert(motyl_src)
maska_czysta = maska_czysta.point(lambda x: 255 if x > 1 else 0)

tlo = Image.open("kolorowe_tlo.png").convert("RGB")
tlo = tlo.resize(maska_czysta.size)

kolorowy_motylek = Image.new(mode='RGB', size=maska_czysta.size)
kolorowy_motylek.paste(tlo, box=(0, 0), mask=maska_czysta)

w = postac.width // 6
prop = maska_czysta.height / maska_czysta.width
h = int(w * prop)
size = (w, h)

motyl_res = kolorowy_motylek.resize(size, Image.Resampling.LANCZOS)
maska_res = maska_czysta.resize(size, Image.Resampling.LANCZOS)

motyl_normal = motyl_res
motyl_right = motyl_res.rotate(-60, expand=True, resample=Image.Resampling.BICUBIC)
motyl_left = motyl_res.rotate(angle=60, expand=True, resample=Image.Resampling.BICUBIC)

maska_normal = maska_res
maska_right = maska_res.rotate(-60, expand=True, fillcolor=0)
maska_left = maska_res.rotate(angle=60, expand=True, fillcolor=0)

postac.paste(motyl_normal, box=(50, 400), mask=maska_normal)
postac.paste(motyl_right, box=(400, 500), mask=maska_right)
postac.paste(motyl_left, box=(300, 300), mask=maska_left)

postac.save("postac_motylki.png")
```

Po wykonaniu kodu otrzymujemy następujący obraz:

