

## Sprawozdanie

### Zaawansowane przetwarzanie sygnałów

Wojciech Adamek, 226337, Czwartek 19

### Easter Egg

#### ***Cel ćwiczenia:***

Badanie wpływu jasności obrazu na zaszumienie zmienną losową z rozkładu Poissona. Sprawdzenie działania algorytmów wykrywania krawędzi oraz obiektów dla zaszumionych obrazów.

#### ***Przebieg ćwiczenia:***

Wartość zmiennej losowej z rozkładu Poissona wyraża się następującym wzorem:

$$P(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Gdzie lambda to jasność danego piksela.

Według wzoru, można spodziewać się, że wraz z zmniejszeniem jasności pikseli rośnie wartość zaszumienia ze względu na obecność jasności w wykładniku eksponenty.

Wygenerowane zostały szumy dla każdego obrazu osobno, w których zmniejszono jasność pikseli według następujących wartości:

$$\lambda = \frac{\lambda}{2}, \frac{\lambda}{4}, \frac{\lambda}{8}, \dots, \frac{\lambda}{128}$$

Wygenerowane obrazy wyglądają następująco:



Wszystkie obrazy są ciemniejsze od oryginalnego i każdy następny jest coraz bardziej zaszumiony.

Następnie użyto algorytmów wykrywania krawędzi według implementacji matlaba:

Canny:



Sobel:



Roberts:



Zdecydowanie najgorszy rezultat otrzymano poprzez zastosowanie filtru Robertsa. Filtr Canny jest bardziej czuły na szum w porównaniu do Canny, co niekoniecznie jest złą cechą, ponieważ dzięki temu jesteśmy w stanie lepiej zobaczyć zarys szczegółów łabędzia.

Oba filtry najlepiej pracują dla  $\lambda = 1/32$  oraz  $1/64$ . Może to mieć związek z faktem, że łabędź, który jest na pierwszym planie jest o wiele jaśniejszy od tła, które dla takiej wartości  $\lambda$  zbliża się do jednej barwy.

Niestety nie można dokładnie porównać działania filtrów i wywnioskować, który lepiej się nadaje do detekcji krawędzi, ponieważ wymagałoby to znajomości macierzy spłotu jaka jest używana w algorytmie według implementacji Matlaba.

Następnie użyto narzędzia Microsoft Computer Vision, aby sprawdzić na którym obrazie łabędź nie będzie już rozpoznawany.

Dla obrazu wejściowego aplikacja rozpoznaje łabędzia z pewnością niemalże 100%.

```
[ { "name": "bird", "confidence": 0.999770463 }, { "name": "animal", "confidence": 0.9997662 }, { "name": "swan", "confidence": 0.999734938 }, { "name": "aquatic bird", "confidence": 0.998846531 }, { "name": "outdoor", "confidence": 0.9872344 }, { "name": "close", "confidence": 0.2848639 } ]
```

Dla pierwszego obrazu ( $\lambda = 1/2$ ) aplikacja rozpoznaje łabędzie z pewnością 95%.

```
[ { "name": "aquatic bird", "confidence": 0.994567156 }, { "name": "animal", "confidence": 0.993533731 }, { "name": "bird", "confidence": 0.9901338 }, { "name": "swan", "confidence": 0.9498539 } ]
```

Dla drugiego obrazu ( $\lambda = 1/4$ ) łabędź jest rozpoznany z pewnością 73%.

```
[ { "name": "aquatic bird", "confidence": 0.981366158 }, { "name": "animal", "confidence": 0.9739015 }, { "name": "bird", "confidence": 0.9451338 }, { "name": "swan", "confidence": 0.731797755 } ]
```

Dla trzeciego obrazu ( $\lambda = 1/8$ ) aplikacja nie rozpoznaje już łabędzia. Twierdzi jednak, że jest to wodny ptak z pewnością 79%.

```
[ { "name": "animal", "confidence": 0.8091498 }, { "name": "aquatic bird", "confidence": 0.793554246 }, { "name": "bird", "confidence": 0.719985 } ]
```

Na czwartym obrazie nie zostaje zidentyfikowany żaden obiekt. Aplikacja zgaduje że jest to screenshot człowieka.

```
{ "tags": [ "photo", "man", "black", "monitor", "sitting", "white", "screen", "red", "room", "standing" ], "captions": [ { "text": "a screen shot of a person", "confidence": 0.5521608 } ] }
```