

Sieć AlexNet:

Liczba warstw Conv: 5

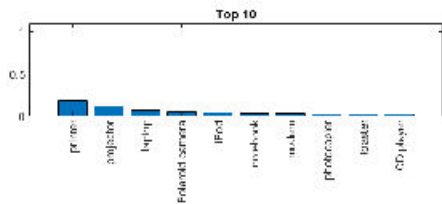
Wymiary wejścia: 227x227x3

Rozpoznaje 1000 obiektów

img1

```
I = imread('test_img1.png');  
figure, imshow(I);
```

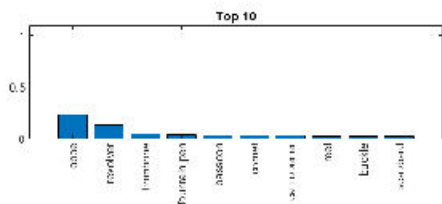
Rozpoznano: printer, F = 0.18153



img2

```
I = imread('test_img2.png');  
figure, imshow(I);
```

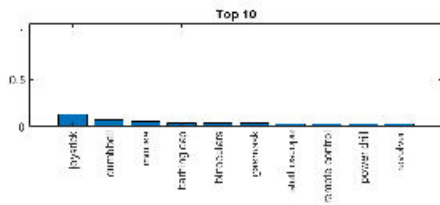
Rozpoznano: obus, P = 0.23394



img3

```
I = imread('test_img3.png');  
figure, imshow(I);
```

Rozpoznano: Joystick, P = 0.13035



Rozmiar filtrow: 11x11

FilterSize - rozmiar filtru

NumChannels - liczba kanałów które posiadają dane wejściowe np dla RGB - 3, GRAY - 1

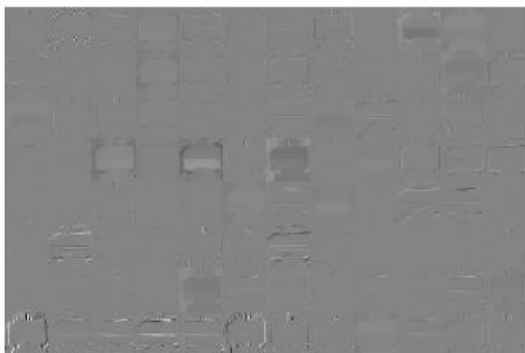
Stride - ile pikseli jest omijane przez kernel, podczas przechodzenia po obrazie wejściowym

Liczba filtrów - 96

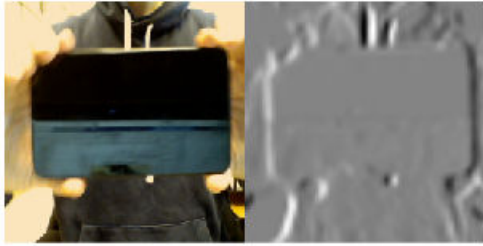
Będzie wykrywał linie/krawędzie

img1

```
imshow(I1a)
```

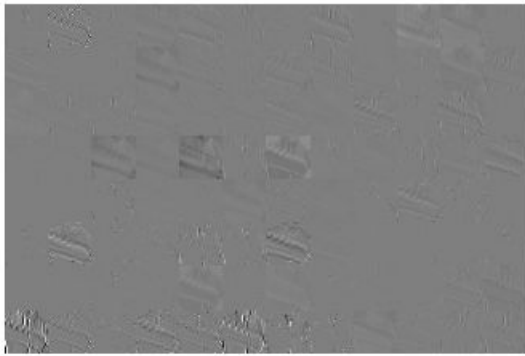


```
imshow(I2a)
```



img2

```
imshow(I1b)
```

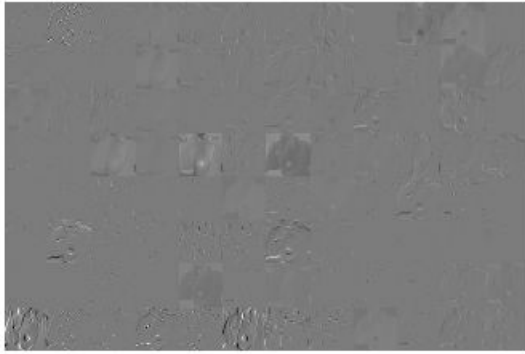


```
imshow(I2b)
```



img3

```
imshow(I1c)
```

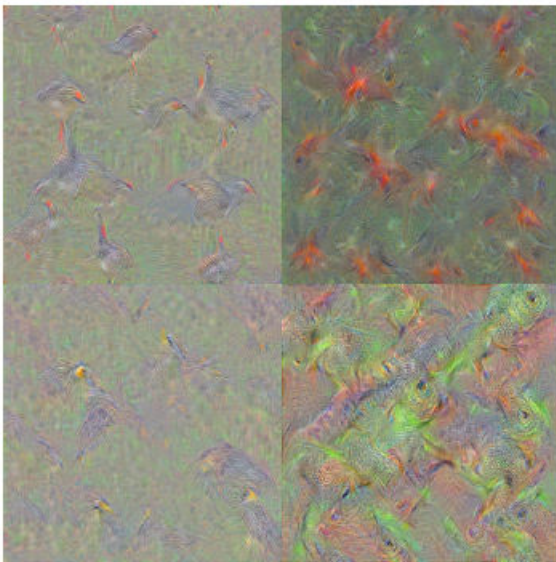


```
imshow(I2c)
```



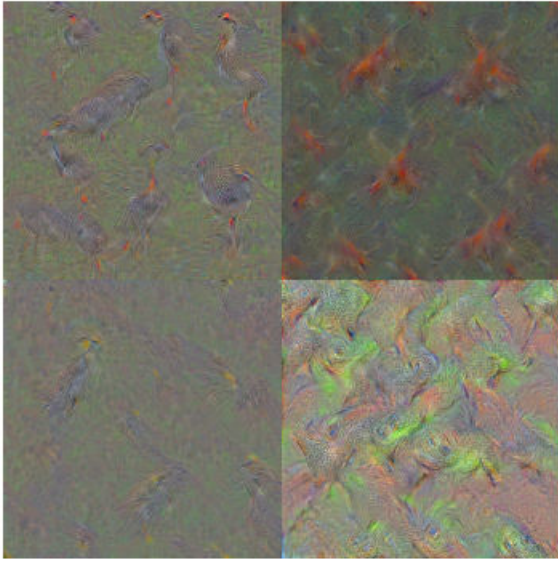
img1

```
imshow(I1a)
```



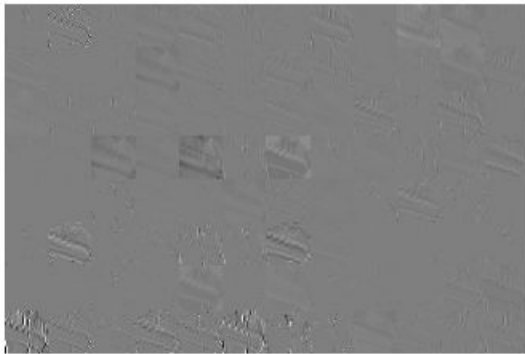
img2

```
imshow(I1b)
```



img3

```
imshow(I1c)
```

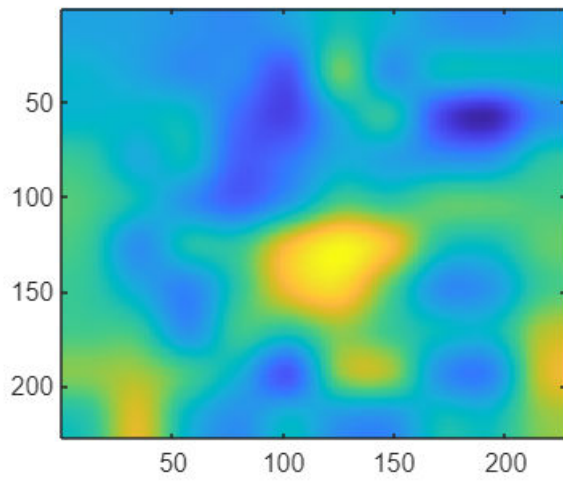


img1

```
imshowpair(IM1a, mapa)
```



```
imagesc(mapa)
```

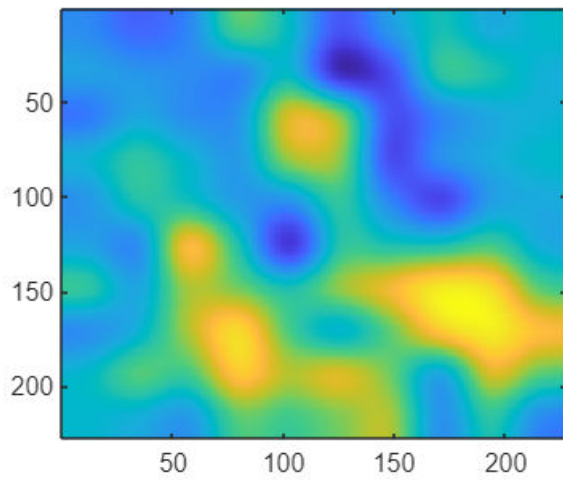


img2

```
imshowpair(IM1b, mapb)
```



```
imagesc(mapb)
```

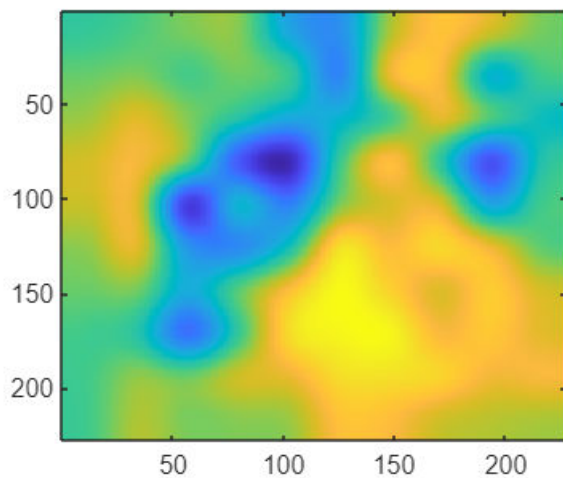


img3

```
imshowpair(IM1c, mapc)
```



```
imagesc(mapc)
```



transfer learning - bierzemy gotowy model nauczony dla innego problemu, podmieniamy warstwę klasyfikującą, albo kilka górnych warstw na takie, które będą odpowiednie dla nowego (naszego), problemu, przyuczamy sieć powstałą w ten sposób na nowych danych, wcześniej zamrażając warstwy już nauczone.

augmentacja danych - zwiększa ilość dostępnych próbek uczących, co pozwala lepiej nauczyć sieć i zmniejszyć prawdopodobieństwo overfittingu.

Rodzaje warstw:

Convolution layer - odpowiedzialna za tworzenie filtrów, które identyfikują cechy na obrazie

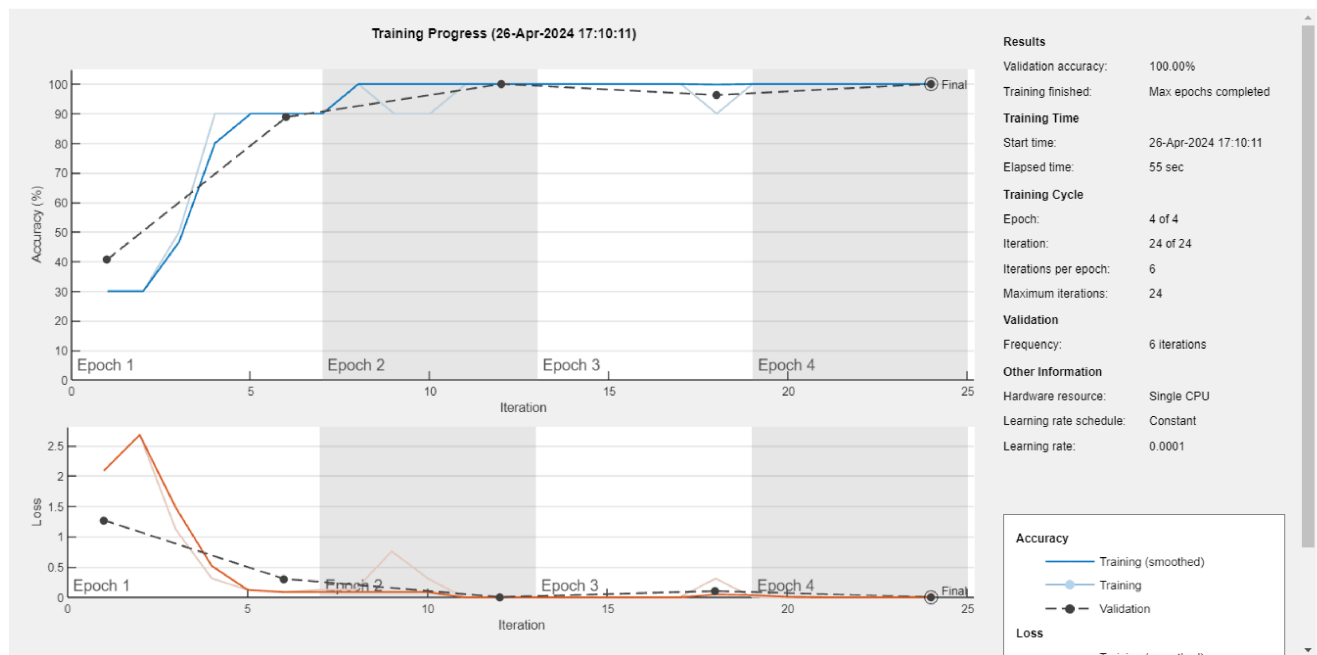
Pooling layers (Max/mean) - łączą grupy pixeli, wyciągając najważniejsze cechy i zmniejszając objętość danych

Activation layers - warstwy odpowiedzialne za okrojenie wyjść z neuronów to określonych przez nas ram

Dense - warstwy łączące neurony każdy z każdym, wykorzystywane na samym końcu do klasyfikacji obrazu

6) - wyniki uczenia Acc - 100%, loss - 0


```
figure;
imshow('training.png')
```



7) Zbiór walidacyjny

```
disp(['Dokladnosc (zbior walidujacy) = ' num2str(accuracy,'%2.1f'), '%'])
```

Dokladnosc (zbior walidujacy) = 100.0%

8) zbiór testowy

```
disp(['Dokladnosc (zbior testowy) = ' num2str(accuracyT,'%2.1f'), '%'])
```

Dokladnosc (zbior testowy) = 86.7%

```
confusionchart(YValidationT,YPredT)
```

True Class	obiektA	15	8	2
	obiektB		25	
	obiektC			25
		obiektA	obiektB	obiektC
		Predicted Class		

Mylone są telefon z bidonem, obydwie te obiekty są bardzo zbliżone wizualnie (kolor, kształt), różni je trochę budowa (wcięcia w bidonie) i rozmiar, nie dziwne, że klasyfikator ma z nimi problem. Obiekt C (myszka) różni się dużo bardziej co widać na wykresie.

Podsumowanie

Rezultaty i analiza - powyżej

Pytania

1. Struktura sieci AlexNet - widoczne bloki tworzone z Conv, Pool, Activation. Istotność parametrów takich jak stride, filterSize - niezbędne do zrozumienia jak powstają pojedyncze filtry w sieci Conv. Idea transfer learningu - przydatne narzędzie, np gdy nie posiadamy wystarczającej liczby danych treningowych.
2. Dobieranie parametrów sieci do problemu. Inne sposoby radzenia sobie z przeuczeniem sieci
3. Czemu do tego problemu została wykorzystana akurat sieć AlexNet, a nie jedno z nowszych rozwiązań