

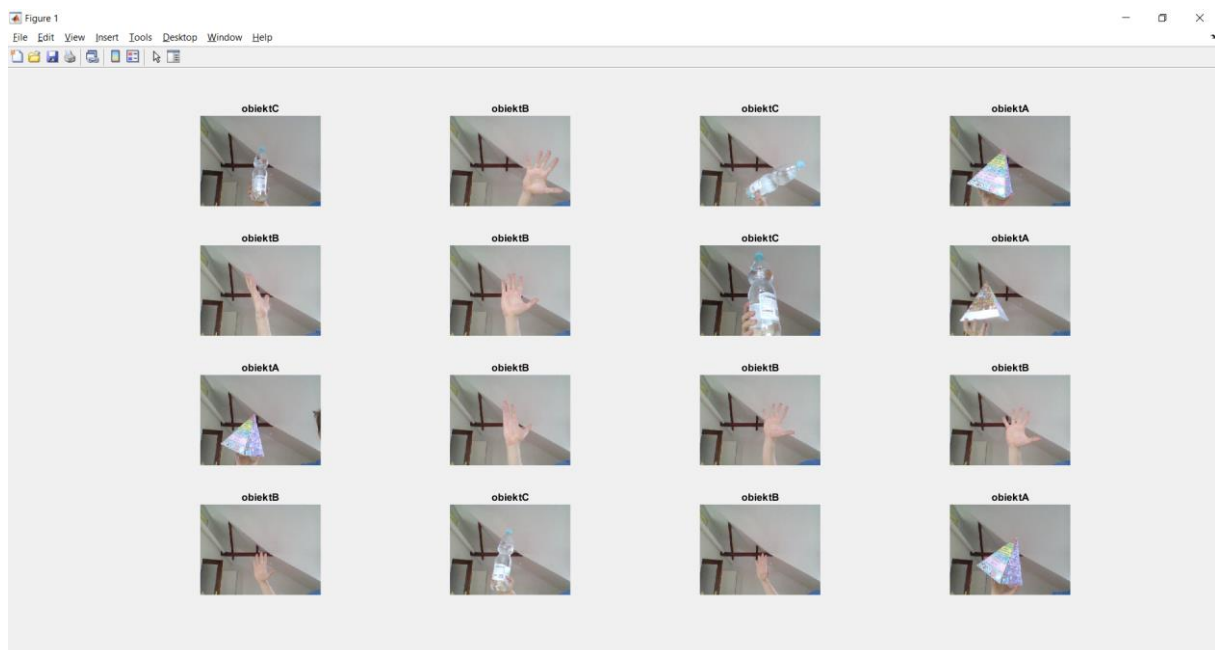
**Data:** 23.05.2020

**Imię i nazwisko:** Marek Matys

# Rezultaty

## *Część II – sieci konwolucyjne CNN*

*Wizualizacje przykładowych danych uczących:*



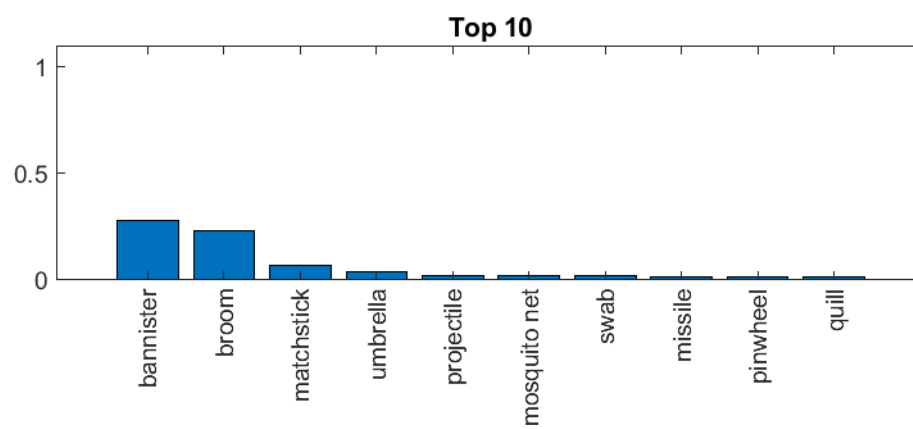
*Rozmiar obrazu wejściowego akceptowanego przez sieć Alexnet:*  
**227 x 227 x 3**

*Odkomentowany kod:*

```
%-eksport wykresu do pliku (nazwa ustawiana automatycznie wg nazwy pliku  
% wejściowego)  
% - można to także zrobić "recznie" - w oknie Figure, Manu, Plik>Save As... wybrać PNG  
  
[~,fname1,~]=fileparts(imFileName);  
saveas(fh,[fname1 '_output.png'])  
  
%% Wizualizacja działania sieci (wagi, aktywacje) dla wybranego obrazu
```

*Kilka rezultatów rozpoznawania przy użyciu sieci Alexnet:*

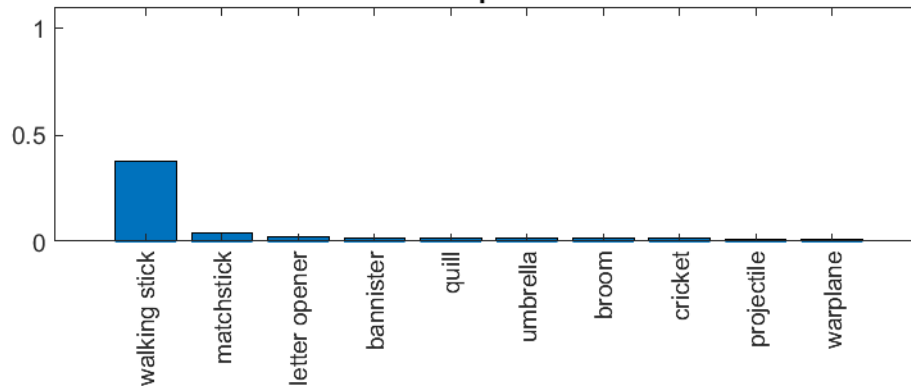
**Rozpoznano: bannister, P = 0.27909**



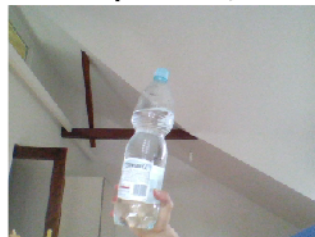
Rozpoznano: walking stick, P = 0.37667



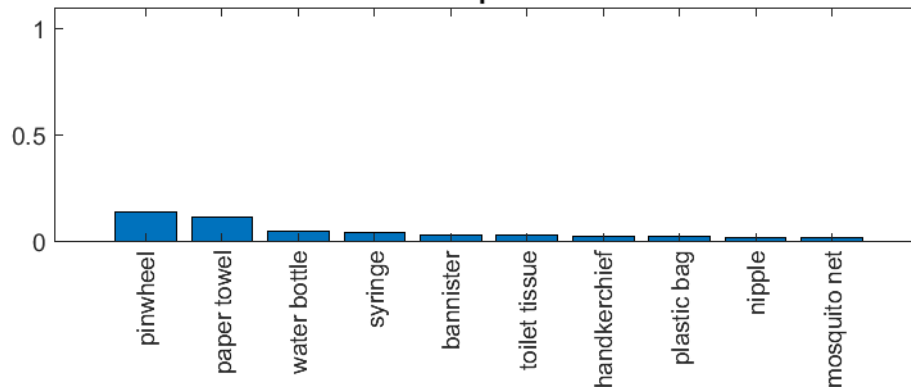
Top 10



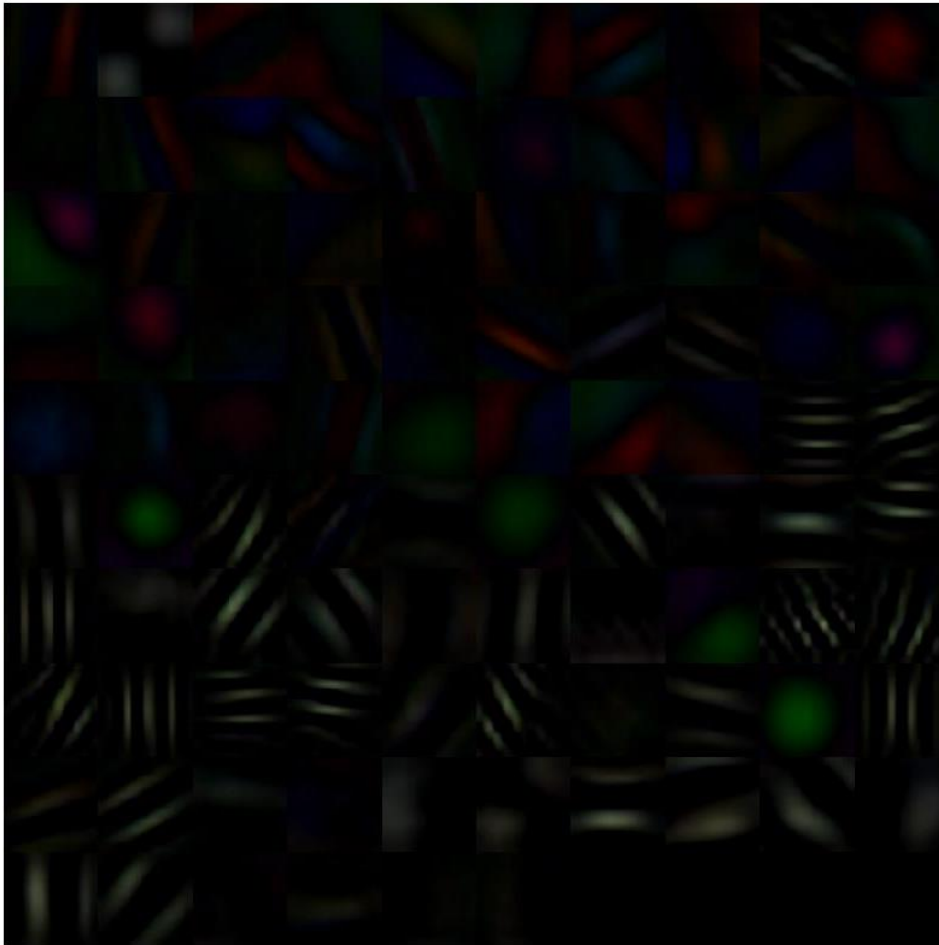
Rozpoznano: pinwheel, P = 0.14333



Top 10



tablica filtrow/wag warstwy CONV1)



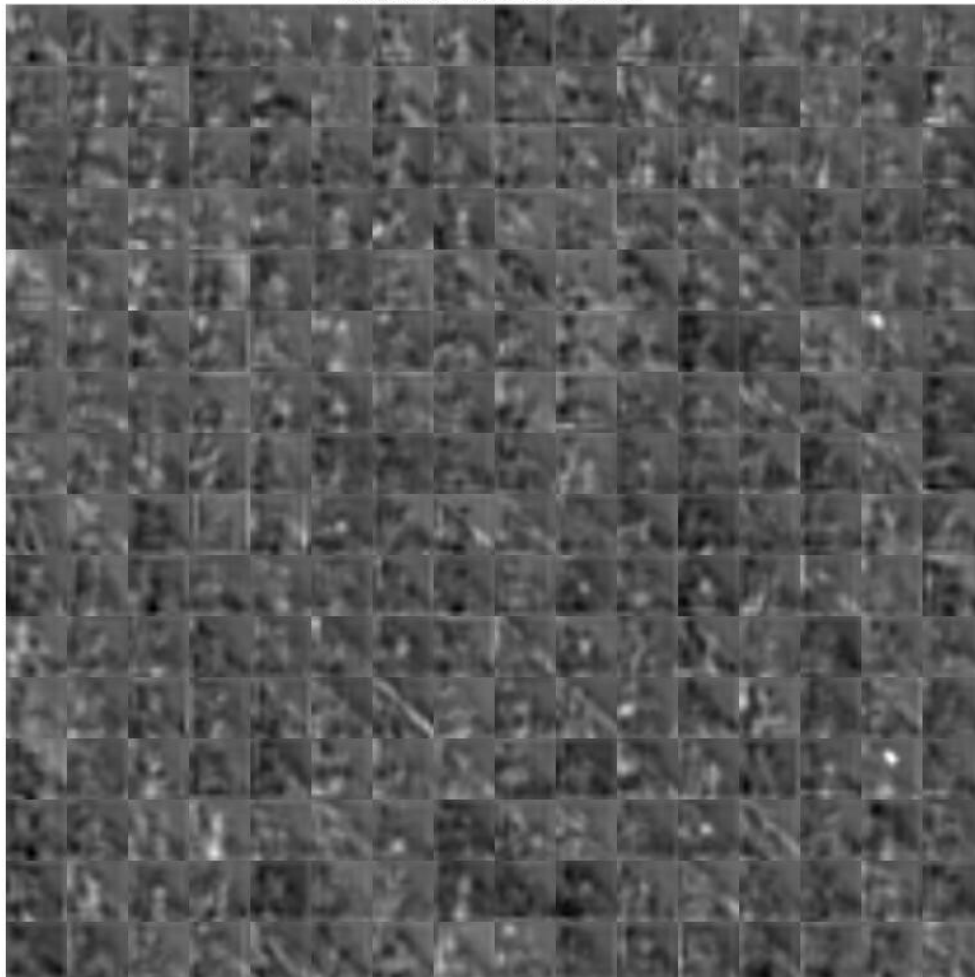
Tablica filtrów 1 warstwy. Filtry w dolnej części zdjęcia wyglądają na filtry bardziej reagujące na cienkie, równoległe linie. Początkowe filtry wykrywają bardziej pewne obszary i ich położenie.

Najsilniejsza aktywacja warstwy CONV1 nastąpiła na krawędziach butelki i drewnianych elementów w tle.

najsilniejsza aktywacja warstwy CONV1



aktywacje warstwy CONV5



## najsilniejsza aktywacja warstwy CONV5



Dokładność klasyfikacji sieci: **92.6%**

```
200
201 % - uczenie (proces ten moze potrwać długo - w zależności od ilości danych)
202 netTransfer = trainNetwork(augimdsTrain, layers, options);
203
204 %- eksport sieci do pliku MAT
205 save nauczonaSiec netTransfer
206
207 %% Testowanie sieci na danych walidacyjnych
208 %- zanotuj dokładność klasyfikacji
209 [YPred, scores] = classify(netTransfer, augimdsValidation);
210
211 YValidation = validationImages.Labels;
212 accuracy = 100*mean(YPred == YValidation);
213 disp(['Dokładność (zbior walidujący) = ' num2str(accuracy, '%2.1f'), '%'])
214
215 %% Testowanie sieci - osobny zbior testowy
216 %- nagrać kilka dodatkowych obrazów testowych dla każdej klasy (do
```

	2	11.61	3
	3	10.39	3
	4	11.78	3
	5	12.31	3
	6	12.71	3
	7	13.18	3
	8	18.12	3
	9	14.66	3
	10	15.34	3

```
Warning: Image is too big to fit on screen; displaying at 67%
Dokładność (zbior walidujący) = 92.6%
fx >>
```

Macierz pomyłek:

True class	obiektA	23	7	
	obiektB		30	
	obiektC	1	9	20
		obiektA	obiektB	obiektC
		Predicted class		

Model bezbłędnie rozpoznawał rękę. Najczęściej mylił butelkę z ręką (9 razy), piramidkę z ręką (7 razy) i 1 raz została butelka pomyłona z piramidką.

## Analiza i wnioski

*Zanotuj wnioski z eksperymentów realizowanych przy pomocy skryptu `uczenie_transfer_learning.m`*

### **Omówienie rezultatów rozpoznawania:**

Tło na którym robione były zdjęcia było bardzo niejednolite- zawiera wiele pionowych elementów (belek), które mogą być wykryte podczas rozpoznawania obrazu i przez to negatywnie rzutują na wynik operacji rozpoznawania.

Dodatkowo obiekty A i C były trzymane w ręce - która za każdym razem była w kadrze i przez to do algorytmu nie dostawały się “czyste” zdjęcia samych obiektów do wykrycia, ale zdjęcia obiektu do wykrycia i ręki, co zaburzało wynik rozpoznawania.

**Wniosek:**

Przy następnych próbach wykrywania należałoby zadbać o stosunkowo gładkie tło i starać się uchwycić na obrazie tylko przedmiot który chcemy aby sieć rozpoznała.

## Pytania

*Wymień rodzaje warstw sieci konwolucyjnych i ich rolę*

- Warstwa Konwolucji (Convolution) - wyostanie pożądanych cech, ekstrakcja cech
- Warstwa ReLU - eliminacja problemu zanikającego gradientu
- Warstwa krzyżowej normalizacji kanałów
- Warstwa MaxPoolingu - stosowana aby uniknąć przeuczenia modelu oraz redukuje ilość cech i złożoność obliczeniową sieci

*Znajdź w internecie informacje na temat "Deep Dream Visualization"*

Jest to projekt open source założony przez Google wykorzystujący w swoim działaniu konwolucyjne sieci neuronowe.

Działa to na zasadzie znajdowania wzorców w obrazie i uwypukleniu ich, aż do przesady. Działa z wykorzystaniem zjawiska **pareidolii**- czyli właśnie dopatrywania się znanych kształtów w przypadkowych przedmiotach.

Z racji, że jest to projekt open source, można w internecie znaleźć różnego rodzaju generatory takich obrazów np.:

<https://deepdreamgenerator.com/>