

# Metody sztucznej inteligencji 2

## Projekt 2. — Rozpoznawanie kształtów w czasie rzeczywistym

### Konspekt

Bartłomiej Dach, Tymon Felski

27 maja 2018

Niniejszy dokument zawiera informacje wstępne na temat drugiego projektu, którego celem jest zaimplementowanie dwóch podejść do rozpoznawania prostych kształtów — z wykorzystaniem biblioteki **OpenCV** i sieci neuronowej — oraz ich analiza i porównanie na podstawie wybranych danych treningowych.

## 1. Skład grupy

Projekt realizowany będzie w dwuosobowej grupie, w składzie:

1. Bartłomiej Dach,
2. Tymon Felski.

## 2. Opis problemu badawczego

Uczenie maszynowe znajduje szerokie zastosowania w dziedzinie rozpoznawania obiektów. Istnieje niezliczona ilość podejść do tego problemu, nierzadko różniących się znacząco wykorzystywanymi technikami i technologiami. Wszystkie bazują jednak na znajdowaniu charakterystycznych cech obrazów wejściowych i stwierdzaniu czy na obu z nich mamy do czynienia z tą samą cechą.

Niniejszy projekt będzie ograniczał się jedynie do rozpoznawania prostych kształtów geometrycznych takich jak: trójkąt, kwadrat, koło, gwiazda. Kształty będą mogły znajdować się na obrazach w różnych miejscach, w innej skali i orientacji oraz po kilka na jednym obrazie. Docelowo planowane jest umożliwienie nie tylko wczytania i rozpoznania kształtu na pojedynczym obrazie, ale również rozpoznawania kształtów na strumieniu wideo w czasie rzeczywistym. Będzie to także jedno z kryteriów oceny rozwiązania.

## 3. Cel badań

Ideą projektu jest weryfikacja jakości i wydajności sposobów rozpoznawania prostych kształtów różnymi metodami. Wyselekcjonowane podejścia zostaną porównane z użyciem jednakowych danych wejściowych, co pozwoli na eksperymentalne wyłonienie najlepszego.

## 4. Planowane do wykorzystania technologie

Językiem programowania, który zdecydowano się wykorzystać, jest język skryptowy **Python** w wersji 3.5.2. Jest to uwarunkowane między innymi sporymi możliwościami tego języka w zakresie przetwarzania i analizy danych, wygodą programowania i przenośnością stworzonych rozwiązań.

Planowane jest zaimplementowanie dwóch podejść pozwalających na rozwiązanie postawionego problemu. Jedno z nich będzie wykorzystywać wbudowane funkcjonalności biblioteki **OpenCV** w celu analizy obrazów i znajdowania kształtów. Drugie rozwiązanie będzie bazować na sieci neuronowej skonstruowanej z użyciem biblioteki **Keras**. Model sieci zostanie stworzony i nauczony od zera na wybranych na potrzeby projektu danych.

W celu uproszczenia pracy z danymi oraz ich analizy, przydatne mogą okazać się moduły Pythona takie jak: **pandas**, **NumPy** oraz **Matplotlib**.

W poniższej tabeli zestawiono wspomniane biblioteki wraz z ich wersjami oraz określono licencje, na których zostały udostępnione.

Nr	Komponent, wersja	Opis	Licencja	
1	Keras, 2.1.4	Biblioteka udostępniająca API do sieci neuronowych	MIT License	[3]
2	Matplotlib, 2.1.0	Umożliwia tworzenie wykresów	Matplotlib License	[5]
3	NumPy, 1.13.3	Używana do efektywnych obliczeń na wektorach $n$ -wymiarowych	BSD License	[9]
4	OpenCV, 3.3.0	Biblioteka do obróbki obrazów	New BSD License	[4]
5	pandas, 0.21.0	Wspomaga ładowanie danych z plików CSV oraz ich analizę	BSD License	[6]

Tablica 1: Wykorzystane biblioteki wraz z określeniem licencji

## 5. Opis danych

Do realizacji projektu wybrano zbiór danych **Four Shapes** [7]. Zbiór ten zawiera 16 tysięcy zdjęć w rozdzielczości  $200 \times 200$ , na których znajdują się cztery kształty: trójkąt, kwadrat, koło i gwiazda.

Dane zostały zebrane przy pomocy kamery. Autor wyciął kartonowe szablony każdego z kształtów, pomalował je na zielono, a następnie nagrał cztery dwuminutowe filmy, w których podnosił i obracał każdy z kształtów. Przy pomocy biblioteki OpenCV w języku Python kształty zostały wycięte i przeskalowane do docelowych rozmiarów. Na koniec nagrany został piąty, testowy film, który posłuży do weryfikacji poprawności i wydajności działania nauczonych modeli.

Poza wspomnianym powyżej zbiorem treningowym, w trakcie projektu zostaną wykorzystane także autorskie obrazy zawierające proste kształty — zarówno szukane jak i inne — w celu sprawdzenia stworzonych rozwiązań.

## 6. Sposób weryfikacji rezultatów

Jednym z kryteriów jakości klasyfikacji kształtów przez każde z rozwiązań będzie ich dokładność (ang. *accuracy*), czyli stosunek poprawnie zidentyfikowanych oraz poprawnie odrzuconych kształtów do wszystkich rozważanych. Rozsądnym jest stwierdzenie, że oczekujemy jak najlepszych wyników.

Dodatkowym aspektem, który będzie brany pod uwagę jest wydajność. Docelowo projekt przewiduje zaadoptowanie każdej z metod na analizę obrazów do użycia ze strumieniem wideo w czasie rzeczywistym. Na tym etapie nie jest możliwe stwierdzenie, czy będzie to możliwe w przypadku obu metod, jednak zostanie to zbadane i opisane po przeprowadzeniu testów.

## Literatura

- [1] Abe, S.: *Support Vector Machines for Pattern Classification*, Second Edition. Springer: Nowy Jork, 2010.
- [2] C.M. Bishop: *Neural Networks for Pattern Recognition*. Oxford University Press: Nowy Jork, 1995.
- [3] Chollet F., Keras Team: Keras – Deep Learning for humans.  
Oficjalna strona: <https://keras.io> [Dostęp 17 marca 2018]
- [4] Intel Corporation, Willow Garage, Itseez: OpenCV – Open Source Computer Vision.  
Oficjalna strona: <https://opencv.org>. [Dostęp 17 marca 2018]
- [5] Matplotlib Development Team: Matplotlib.  
Oficjalna strona: <https://matplotlib.org>. [Dostęp 17 marca 2018]
- [6] McKinney W.: pandas – Python Data Analysis Library.  
Oficjalna strona: <https://pandas.pydata.org>. [Dostęp 17 marca 2018]
- [7] Meschke S.: Four Shapes dataset.  
Dostępny: <https://www.kaggle.com/smeschke/four-shapes>. [Dostęp 17 marca 2018]
- [8] Murty M.N., Devi V.S.: *Introduction to Pattern Recognition and Machine Learning*. World Scientific Publishing: Singapur, 2015.
- [9] Oliphant T.: NumPy.  
Oficjalna strona: <http://www.numpy.org>. [Dostęp 17 marca 2018]