

# Projekt zaliczeniowy

## Analiza obrazów

Mikhail Lemiasheuski  
Mateusz Nogaj  
Rafał Gonet

January 22, 2023

## 1 Opis projektu

Celem projektu jest stworzenie aplikacji rozpoznającej pandy na podanym obrazku. I.e. wyświetlenie informacji o tym, czy na podanym obrazku znajduje się kształt pandy. W celu komunikacji z użytkownikiem opracowany został interfejs graficzny, umożliwiający użytkownikowi podstawową komunikację z programem (załadowanie/podanie ścieżki do pliku oraz wyświetlenie wyniku)

## 2 Założenia wstępne

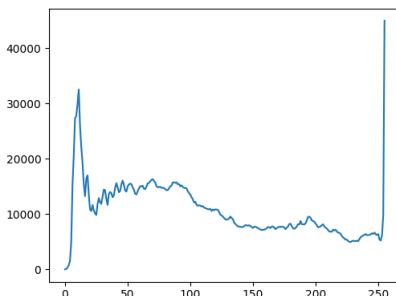
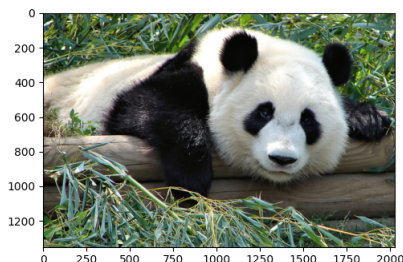
W wersji końcowej stworzona została następująca funkcjonalność:

- Projekt pozwala na załadowanie obrazka w każdym formacie, wspieranym przez bibliotekę OpenCV, są to między innymi - .JPG, JPEG, .PNG, .BMP
- Wielki przycisk "Sprawdz", uruchamiający cały kod programu
- Wyświetlenie wyniku analizy obrazka (obrazek w raz z informacją o obecności na nim pandy)

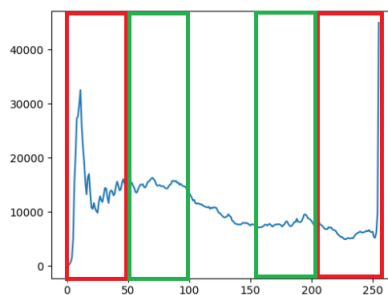
## 3 Opis rozwiązania

### 3.1 Ogólny zarys metody rozwiązania

Rozpoznanie kształtu pandy oparliśmy głównie o skład kolorowy obrazka oraz kształty na nim obecne. Zauważyliśmy, że histogram rozkładu wartości pikseli 0-255 obrazka z panda w skali szarości ma dosyć charakterystyczną postać



Jednym z parametrów modelu wybraliśmy stosunek 50 pikseli "najbardziej czarnych" do ilości następnych 50 "mniej czarnych" w celu wyznaczenia, czy istnieje na histogramie obrazka pokazany "szczyt".



Drugim parametrem była liczba Eulera dla zbinaryzowanego i odszumowanego obrazka. Celem tego parametru jest pokazanie ilości czarnych plam na obrazie, dlatego obrazek jest binaryzowany niskim progiem - 25/255.

W celu uzyskania bardziej dokładnej wartości danego parametru w programie są stosowane algorytmy domknięcia oraz otwarcia binarnego dostarczanych przez bibliotekę *SciPy* oraz funkcja *findContours* z biblioteki *OpenCV* do obliczenia bezpośrednio liczby Eulera.

## 3.2 Perceptron

Jadrem programu jest klasa *Perceptron* implementująca podstawowe funkcje do modyfikacji wag, zapisania ich do pliku oraz testowania "nauczonego" modelu.

W wersji "realisowej" klasa jest inicjowana wczytanymi z pliku .json wartościami wag w celu przyspieszenia uruchomienia aplikacji.

## 3.3 Zbiór danych uczących

## 3.4 Testowanie modelu

# 4 Podsumowanie, analiza błędów

Zastosowana w projekcie metoda uczenia maszynowego nie odpowiada celom działania programu ze względu na poniższe fakty:

- zbiór danych na których opiera się proces "uczenia się" algorytmu nie jest podzielny liniowo w żadnym z wymiarów. I.e.
- wybrane kryteria rozpoznawania pandy są "lekko" prymitywne

W trakcie pracy nad projektem dowiedzieliśmy się o istnieniu modelu **MLP (multilayer perceptron)** używanego do tworzenia algorytmów klasyfikacji obrazów. Ze względu na poziom skomplikowania podobnego rozwiązania i ograniczony czas na stworzenie projektu pozostaliśmy przy opisanym we wcześniejszej części prostym algorytmie.

# 5 Podział pracy i analiza czasowa