Rozpoznawanie i przetwarzanie sygnałów	
Etap 2	
Numer Grupy	10
Skład grupy	Marta Sobas 252742
	Philip Zalewski 252825

Czym jest “proof of concept”?

Jest to implementacja jakiegoś rozwiązania, która ma za zadanie udowodnić, że to co chcemy wprowadzić ma sens, spełnia nasze oczekiwania i jest możliwe do zrealizowania. Jest to rozwiązanie tymczasowe i pisane jak najszybciej, aby zapewnić że rozważane rozwiązanie ma prawo zadziałać.

Opis POC oprogramowania do rozpoznawania rowerów przed pojazdem

Link do repozytorium wstępnej wersji oprogramowania:
https://github.com/aerlevsedi/BicycleDetection.git

```python
import cv2
import numpy as np
from gui_buttons import Buttons

buttons = Buttons()
buttons.add_button("person", 20, 20)
buttons.add_button("cup", 20, 80)
buttons.add_button("bicycle", 20, 140)
buttons.add_button("car", 20, 200)
```

Zgodnie z wymaganiami projektu wstępna wersja oprogramowania jest wykonana z użyciem biblioteki OpenCV. Dodatkowo poza rowerami jest możliwość wykrywania 3 innych obiektów w celach testowania działania programu.

```python
# Opencv DNN
net = cv2.dnn.readNet("dnn_model\\yolov4-tiny.weights", "dnn_model\\yolov4-tiny.cfg")
model = cv2.dnn_DetectionModel(net)
model.setInputParams(size=(320, 320), scale=1/255)
```

W tym miejscu konfigurujemy nasz model sieci neuronowej. Korzystamy z modułu Deep Neural Network (DNN), który jest dostępny w bibliotece OpenCV. Moduł ten dostarczany jest z przeszkoloną, splotową siecią neuronową (CNN) do wykrywania twarzy.

```
18  # Load class list
19  classes = []
20  with open("dnn_model\\classes.txt", "r") as file_object:
21      for class_name in file_object.readlines():
22          class_name = class_name.strip()
23          classes.append(class_name)
24
25  print(classes)
```

Tu następuje zapisanie w tablicy pobranych z modułu DNN wszystkich dostępnych klas obiektów. Tablica ta zostanie w późniejszej części kodu wykorzystana do wyświetlania nazw obiektów nad ramką.

```
27  # Initialize camera
28  # cap = cv2.VideoCapture("highway.mp4")
29  cap = cv2.VideoCapture("IMG_6048.MOV")
30  # cap = cv2.VideoCapture(0) # this takes view from the camera
31
32  cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1280)
33  cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 720)
34  # Full HD 1920 x 1080
```

Ustalamy czy będziemy korzystać w wcześniej nagranego filmiku w celu rozpoznania na nim obiektów, wówczas przekazujemy jego ścieżkę do funkcji VideoCapture, w przypadku kiedy przekażemy do tej funkcji 0 program zacznie rozpoznawać obiekty z kamery na żywo.

```
36  button_person = False
37
38  def click_button(event, x, y, flags, params):
39      global button_person
40      if event == cv2.EVENT_LBUTTONDOWN:
41          print(x, y)
42          buttons.button_click(x, y)
```

funkcja, która obsługuje klikanie przycisków do wyboru klas obiektów.

```
46  #Create window
47  cv2.namedWindow("Frame")
48  cv2.setMouseCallback("Frame", click_button)
49
50  while True:
51      # Get frames
52      ret, frame = cap.read()
```

Początek pętli, w której będzie się wykonywać główna część programu czyli wyświetlanie klatek, obsługa przycisków oraz rozpoznawanie obiektów w klatce.

```
54      # Get active buttons list
55      active_buttons = buttons.active_buttons_list()
56      print("Active buttons", active_buttons)
```

Program pobiera na bieżąco listę aktywnych przycisków.

```
58      # Object detection
59      (class_ids, scores, bboxes) = model.detect(frame)
60
61      for class_id, score, bbox in zip(class_ids, scores, bboxes):
62          (x, y, w, h) = bbox
63
64          class_name = classes[class_id]
65          color = colors[class_id]
66
67          if class_name in active_buttons:
68              cv2.putText(frame, class_name, (x, y - 10), cv2.FONT_HERSHEY_PLAIN, 1, color, 2)
69              cv2.rectangle(frame, (x, y), (x + w, y + h), color, 3)
```

W powyższej pętli program robi rozpoznanie obiektu, który będzie zaznaczony odpowiedniej wielkości ramką (bbox z ang. bounding box) oraz będzie podpisany.
Obiekt będzie rozpoznany jedynie kiedy przycisk odpowiadający konkretnej klasie obiektów, będzie włączony.

```
71      # Display buttons
72      buttons.display_buttons(frame)
73
74      cv2.imshow("Frame", frame)
75
76      key = cv2.waitKey(1)
77      if key == 27: # Press Escape to exit
78          break
79
80  cap.release()
81  cv2.destroyAllWindows()
```

Wyświetlanie przycisków, klatek oraz obsługa wyjścia z programu poprzez wciśnięcie przycisku Escape na klawiaturze.