

Praca Inżynierska  
Politechnika Śląska

Szymon Ciemala

Październik 2021

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
1.1	Ogólnie . . . . .	2
1.2	Wykorzystane technologie . . . . .	2
1.2.1	OpenCV . . . . .	2
1.2.2	MediaPipe . . . . .	2
1.2.3	SciKit Learn . . . . .	3
<b>2</b>	<b>Założenia projektowe</b>	<b>4</b>
2.1	Budowa modułu . . . . .	4
2.2	Dostęp . . . . .	4
<b>3</b>	<b>Część techniczna/praktyczna</b>	<b>5</b>
3.1	Rozpoznawanie dłoni . . . . .	5
3.1.1	Elementy charakterystyczne . . . . .	5
3.1.2	OpenCV - przygotowanie obrazu z kamery . . . . .	5
3.1.3	Generowanie grafiki dłoni . . . . .	6
3.2	SciKit Learn - uczenie maszynowe . . . . .	6
3.2.1	Budowa programu . . . . .	6
3.2.2	Zebrańie danych . . . . .	6
3.2.3	Metody klasyfikacji - uczenie maszynowe . . . . .	6
3.2.4	Ponowne wykorzystanie modelu . . . . .	6
3.3	Paczka PyPi . . . . .	6
3.3.1	Budowa paczki . . . . .	6
3.3.2	Plik setup.py . . . . .	6
3.3.3	Załadowanie paczki do repozytorium . . . . .	6
<b>4</b>	<b>Możliwości wykorzystania</b>	<b>7</b>
4.1	Kioski samoobsługowe . . . . .	7
4.2	Obsługa komputera . . . . .	7
4.3	Sterowanie Robotami Mobilnymi . . . . .	7

# Rozdział 1

## Wstęp

### 1.1 Ogólnie

Praca przedstawia wykorzystanie bibliotek OpenCV, MediaPipe oraz metod uczenia maszynowego biblioteki SciKit Learn do stworzenia modułu dla języka Python, który umożliwi sterowanie dowolnymi aplikacjami przy użyciu gestów oraz ruchu dłońmi.

### 1.2 Wykorzystane technologie

#### 1.2.1 OpenCV

Biblioteka, dzięki której można wykorzystać obraz z kamery oraz wstępnie przetworzyć obraz, który zostanie wykorzystany przez bibliotekę MediaPipe.

#### 1.2.2 MediaPipe

Biblioteka MediaPipe o otwartym źródle, udostępnia wieloplatformowe oraz konfigurowalne rozwiązania wykorzystujące uczenie maszynowe w dziedzinie rozpoznawania, segmentacji oraz klasyfikacji obiektów wizji komputerowej. Niektórymi z rozwiązań są:

- Rozpoznawanie twarzy
- Segmentacja włosów oraz twarzy
- Rozpoznawanie oraz określanie rozmiarów obiektów trójwymiarowych na podstawie obrazu dwuwymiarowego.

Platformy/Języki programowania obsługiwane przez MediaPipe:

- Android
- IOS
- JavaScript
- Python
- C++
- Coral

???? Pozwoli na rozpoznanie dłoni oraz jej elementów charakterystycznych, takich jak nagarstek, stawy oraz końcówki palców. ???

### 1.2.3 SciKit Learn

SciKit Learn to biblioteka, która oferuje różnego typu metody uczenia maszynowego. Biblioteka zawiera algorytmy klasyfikacji, regresji oraz analizy skupień. Biblioteka pozwala na wykorzystanie metod klasyfikacji uczenia maszynowego. Co pozwoli na rozpoznanie gestów dłoni.

# Rozdział 2

## Założenia projektowe

### 2.1 Budowa modułu

Moduł został napisany przy pomocy paradygmatu programowania obiektowego, co pozwala na przystępne wykorzystanie biblioteki w dowolnych projektach wymagających obsługi gestów.

### 2.2 Dostęp

Całość projektu będzie dostępna na platformie GitHub wraz z możliwością pobrania przy pomocy programu pip ze zdłanego repozytorium PyPi, dzięki czemu pozwoli to na sprawne i proste wykorzystanie modułu w dowolnym projekcie.

# Rozdział 3

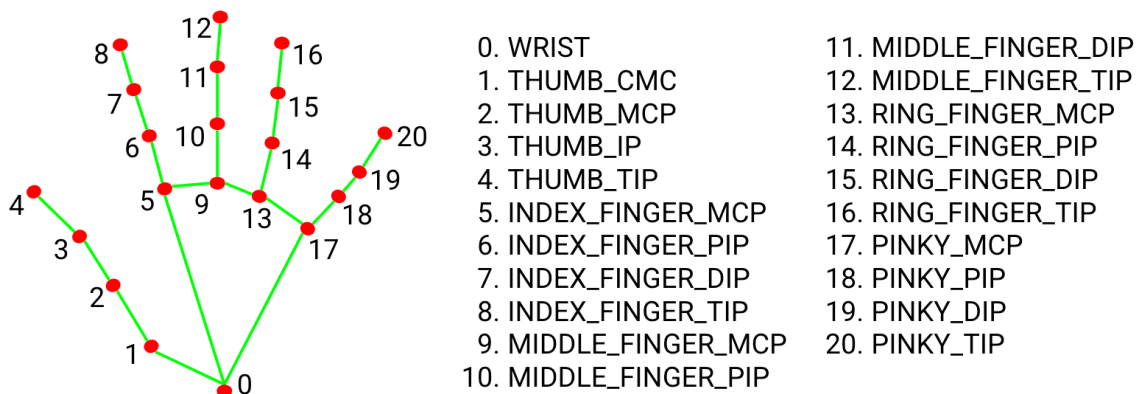
## Część techniczna/praktyczna

### 3.1 Rozpoznawanie dłoni

Pierwszym elementem projektu jest rozpoznanie dłoni poprzez wyznaczenie pozycji elementów charakterystycznych. Pozycja każdego z tych elementów jest względna według pozycji nadgarstka. ????

#### 3.1.1 Elementy charakterystyczne

Model modułu MediaPipe pozwala na wyznaczenie pozycji 21 elementów charakterystycznych dłoni. Współrzędne X i Y są znormalizowane względem rozdzielczości obrazu kamery. Współrzędna X względem liczby pikseli w osi X, a współrzędna Y względem liczby pikseli w osi Y. Oś Z jest prostopadła do osi X i Y, z punktem początkowym w punkcie określającym pozycję nadgarstka. Współrzędna Z jest znormalizowana względem szerokości obrazu kamery, tak jak współrzędna X.



Rysunek 3.1: Elementy charakterystyczne dłoni

Pozycje nadgarstka, paliczków oraz stawów dłoni zostaną wykorzystane do obliczenia obrotu dłoni względem punktu 0 oraz do wytrenowania modeli uczenia maszynowego, których zadaniem będzie rozpoznawanie wybranych gestów.

#### 3.1.2 OpenCV - przygotowanie obrazu z kamery

Poprawne działanie modelu MediaPipe wymaga odpowiedniego przygotowania obrazu kamery.

```

272 def main(self):
273     """
274     Main function that runs the core of the program.
275     """
276
277     hand_type = None
278     if self.cap.isOpened():
279         ret, frame = self.cap.read()
280
281         #BGR to RGB
282         image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
283
284         #Flip horizontal
285         image = cv2.flip(image, 1)
286
287         #Set flag
288         image.flags.writeable = False
289
290         #Detections
291         self.results = self.hands.process(image)
292
293         #Set flag back to True
294         image.flags.writeable = True
295
296         #RGB to BGR
297         image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

```

### 3.1.3 Generowanie grafiki dłoni

## 3.2 SciKit Learn - uczenie maszynowe

### 3.2.1 Budowa programu

### 3.2.2 Zebranie danych

### 3.2.3 Metody klasyfikacji - uczenie maszynowe

### 3.2.4 Ponowne wykorzystanie modelu

## 3.3 Paczka PyPi

### 3.3.1 Budowa paczki

### 3.3.2 Plik setup.py

### 3.3.3 Załadowanie paczki do repozytorium

# Rozdział 4

## Możliwości wykorzystania

4.1 Kioski samoobsługowe

4.2 Obsługa komputera

4.3 Sterowanie Robotami Mobilnymi