

Politechnika Warszawska
Wydział Elektryczny

SYSTEMY WIZYJNE LABORATORIUM

SPRAWOZDANIE Z WYKONANEGO PROJEKTU
**ROZPOZNAWANIE PROSTYCH MIN
NA TWARZACH LUDZKICH**

Wykonawcy projektu:

Bohdan Holovko

Karol Urban

Michał Kądziołka

Opiekun:

Dr inż. Sławomir Skoneczny

Data złożenia projektu:

22.06.2021r.

1. Opis zagadnienia

Zadaniem projektowym do zrealizowania przez zespół było wykonanie w środowisku MATLAB programu pozwalającego na rozpoznawanie min na ludzkich twarzach. Wybrano cztery wyrazy twarzy: radosny, smutny, zdziwiony oraz obojętny (neutralny), które miały być rozróżniane przez program.

W ramach projektu udało się wybrać i opracować sposób rozpoznawania wyrazów twarzy oraz zaimplementować metodę w środowisku MATLAB. Następnie utworzyć bazę zdjęć twarzy z wybranymi minami, wykonać testy programu oraz opracować uzyskane wyniki.

2. Realizacja projektu

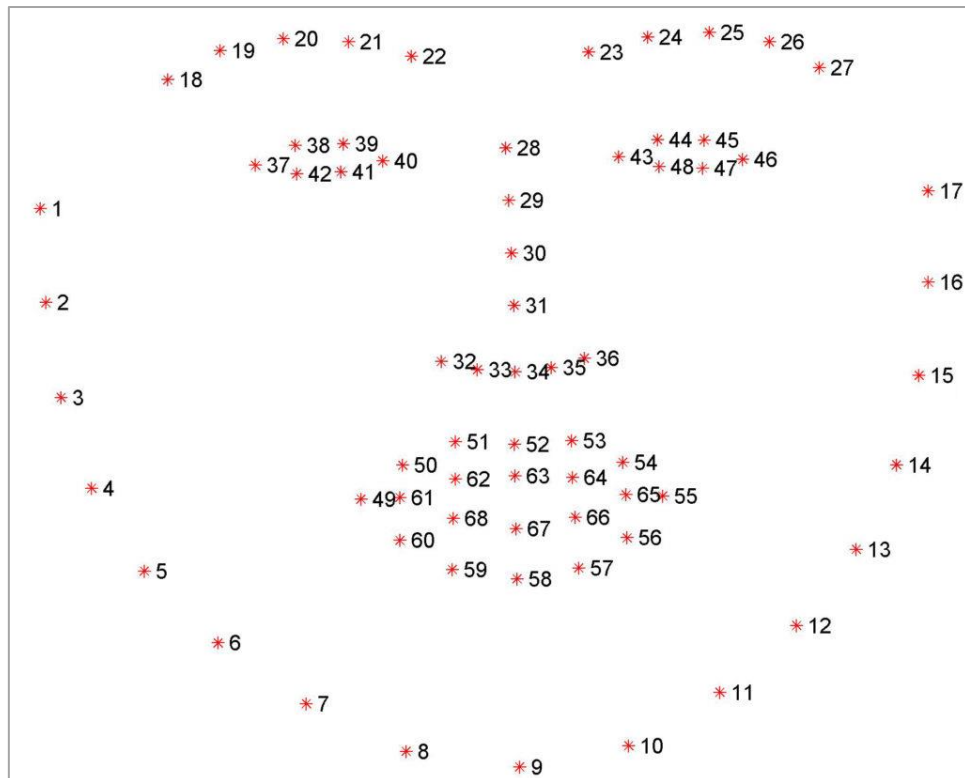
2.1. Metoda wykrywania wyrazów twarzy

Do wykonania programu zdecydowano się na wykorzystanie biblioteki **dlib**. Jest to kompleksowy zestaw algorytmów oraz narzędzi napisany pierwotnie w języku C++. Ciągłe rozwijana zawiera szeroki zestaw funkcji umożliwiający rozwiązywanie problemów z zakresu przetwarzania obrazów, uczenia maszynowego, data mining'u, czy też zaawansowanej algebry liniowej.

W projekcie zostały wykorzystane funkcje biblioteki implementowane w języku Python. Została podjęta próba implementacji biblioteki dlib w Matlabie, jednak ze względu na błąd typu przy wczytywaniu obrazu przez detektor nie udało się zrealizować problemu w tym środowisku.

W programie skorzystano z funkcji *get_frontal_face_detector* wykorzystującej głębokie sieci neuronowe do wykrywania twarzy na obrazie. Zwraca ona współrzędne dwóch wierzchołków prostokąta zawierającego wykrytą twarz.

Po tej operacji użyto funkcji *shape_predictor* pozwalającej na wykrycie 68 punktów charakterystycznych twarzy na podstawie wczytanego wcześniej modelu (http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2). Funkcja zwraca współrzędne poszczególnych punktów na obrazie.



Rys. 1 Wizualizacja 68 współrzędnych punktów charakterystycznych twarzy

W celu klasyfikacji emocji na twarzach zbudowano następujący algorytm drzewa decyzyjnego.

2.2. Implementacja kodu

```
import sys
import os
import dlib
import glob
import cv2

if len(sys.argv) != 3:
    print(
        "Give the path to the trained shape predictor model as the first "
        "argument and then the directory containing the facial images.\n"
        "For example, if you are in the python_examples folder then "
        "execute this program by running:\n"
        "  ./face_landmark_detection.py shape_predictor_68_face_landmarks.dat
    ..../examples/faces\n"
        "You can download a trained facial shape predictor from:\n"
        "  http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2")
    exit()
scale_percent = 25

predictor_path = sys.argv[1]
faces_folder_path = sys.argv[2]
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(predictor_path)
```

```

dict = {}
for count, f in enumerate(glob.glob(os.path.join(faces_folder_path, "*.jpg"))):
    print("Processing file: {}".format(f))
    img = dlib.load_rgb_image(f)
    img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
    width = int(img.shape[1] * scale_percent / 100)
    height = int(img.shape[0] * scale_percent / 100)
    dsize = (width, height)

    # Ask the detector to find the bounding boxes of each face. The 1 in the
    # second argument indicates that we should upsample the image 1 time. This
    # will make everything bigger and allow us to detect more faces.
    dets = detector(img, 1)
    print("Number of faces detected: {}".format(len(dets)))
    for k, d in enumerate(dets):
        print("Detection {}: Left: {} Top: {} Right: {} Bottom: {}".format(
            k, d.left(), d.top(), d.right(), d.bottom()))
        # Get the landmarks/parts for the face in box d.
        shape = predictor(img, d)

        d_smutek = shape.part(48).y - shape.part(66).y
        print(d_smutek)
        d_smutek2 = shape.part(54).y - shape.part(66).y
        print(d_smutek2)
        d_radosc = shape.part(48).y - shape.part(62).y
        print(d_radosc)
        d_radosc2 = shape.part(54).y - shape.part(62).y
        print(d_radosc2)
        stosunek1 = (shape.part(66).y - shape.part(62).y) / (shape.part(64).x - shape.part(60).x)
        print(stosunek1)
        stosunek2 = 0.5 * ((shape.part(40).y - shape.part(38).y) / (shape.part(39).x - shape.part(36).x) +
            (shape.part(46).y - shape.part(44).y) / (shape.part(45).x - shape.part(42).x))
        print(stosunek2)

        for i in range(68):
            img = cv2.putText(img, str(i), (shape.part(i).x, shape.part(i).y),
                cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 255), 2)

        if 0.5 * (d_smutek + d_smutek2) > 10:
            string = "Smutek"
            img = cv2.putText(img, string, (50, 150), cv2.FONT_HERSHEY_SIMPLEX, 7, (0, 0, 255), 4)
        elif d_radosc < 0 and d_radosc2 < 0:
            string = "Radosc"
            img = cv2.putText(img, string, (50, 150), cv2.FONT_HERSHEY_SIMPLEX, 7, (0, 0, 255), 4)
        elif stosunek1 > 0.21 or stosunek2 > 0.34:
            string = "Zdziwko"
            img = cv2.putText(img, string, (50, 150), cv2.FONT_HERSHEY_SIMPLEX, 7, (0, 0, 255), 4)
        else:
            string = "Neutral"
            img = cv2.putText(img, string, (50, 150), cv2.FONT_HERSHEY_SIMPLEX, 7, (0, 0, 255), 4)
        print(string)
        dict[str(f)] = string

```

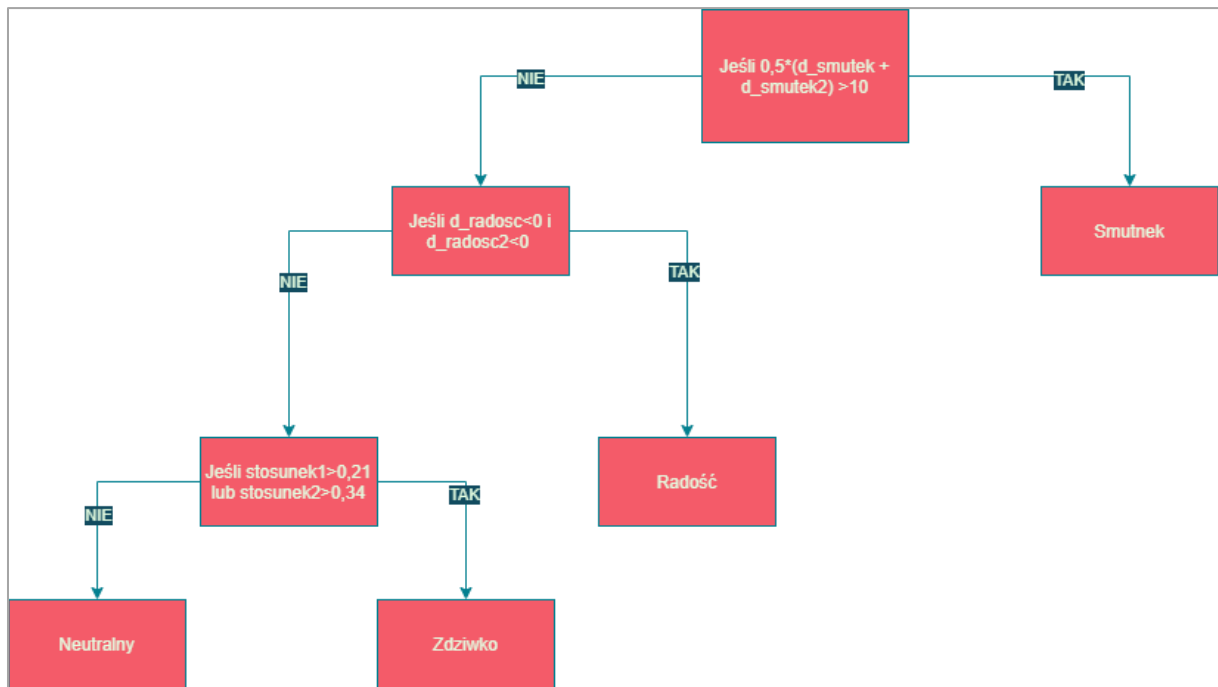
```

output = cv2.resize(img, dsize)
cv2.imshow("", output);

var = cv2.waitKey(2000)
if var == ord("q"): break

cv2.destroyAllWindows()
print(dict)

```

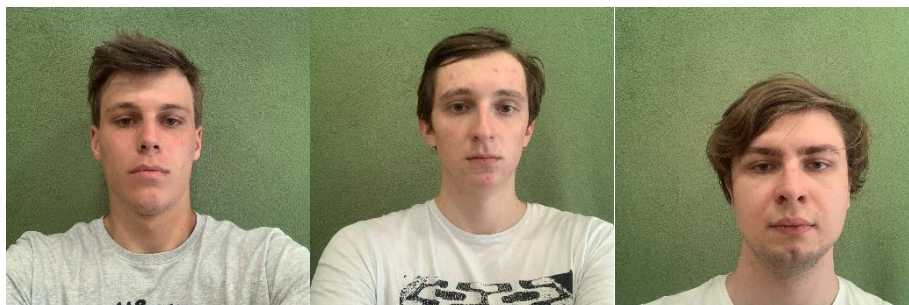


Rys. 2 Drzewo decyzyjne

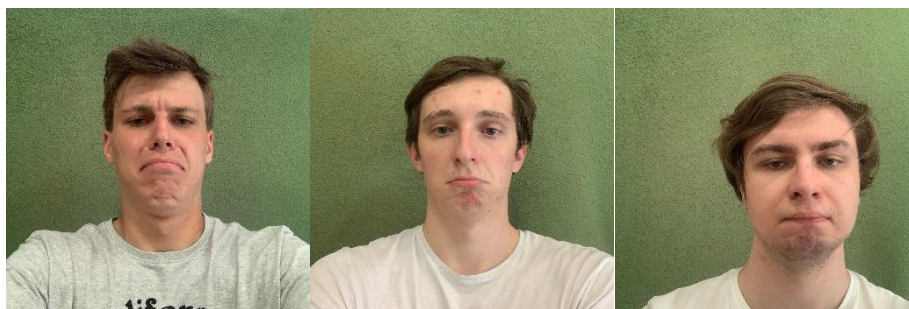
2.3. Utworzenie bazy zdjęć

W celach testowych wykonano własną bazę zdjęć złożoną z fotografii wykonanych jednym tym samym aparatem na jednakowym tle i w jednakowych warunkach oświetleniowych. Baza zawiera zdjęcia twarzy z czterema różnymi wyrazami twarzy: neutralną, radosną, smutną i zdziwioną. Wszystkie miny, prócz neutralnej, zostały przedstawione w sposób bardzo dosadny, tak aby wyraźnie zaznaczyć ukazywaną minę i emocje.

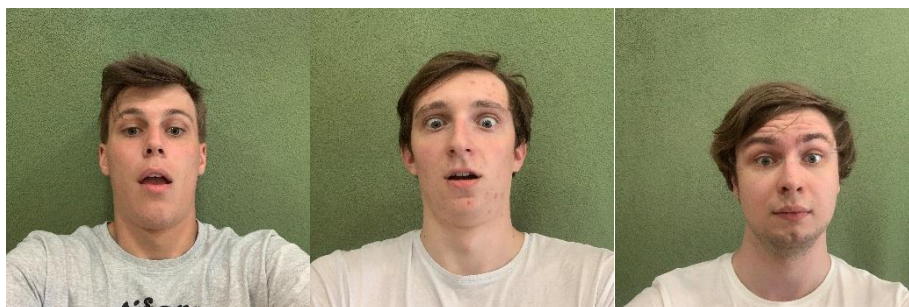
Przykładowe obrazy utworzonej bazy przedstawiono na Rys. 3, Rys. 4, Rys. 5, Rys. 6.



Rys. 3 Neutralny wyraz twarzy



Rys. 4 Smutny wyraz twarzy



Rys. 5 Zdziwiony wyraz twarzy

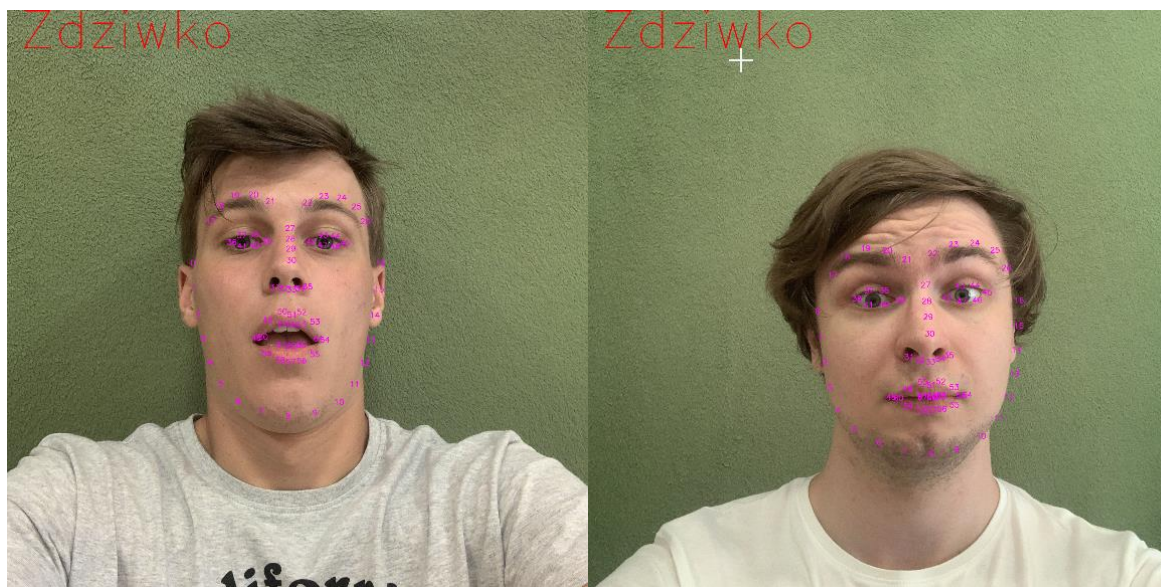


Rys. 6 Radosny wyraz twarzy

3. Uzyskane wyniki

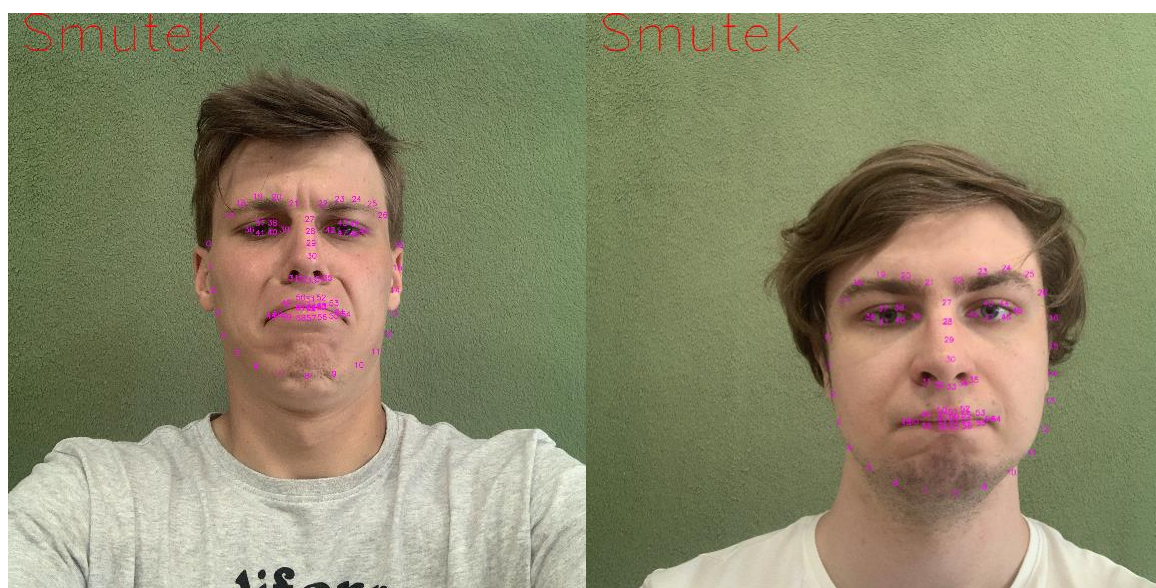
Zdjęcia z utworzonej bazy zostały poddane działaniu programu.

Wykrycie miny typu „zdziwienie” zostało realizowane na podstawie wykrycia bardziej otwartych ust lub oczu. Program wykrywa zdziwienie za pomocą stosunku odległość między brzegowymi punktami ust i odległość między centralnymi punktami warg oraz porównaniu tego stosunku z wartością, która została dobrana metodą prób i błędów. Ten sam warunek został zrealizowany dla oczu. Wynik działania przedstawiono na Rys. 7, gdzie na pierwszym zdjęciu są szeroko otwarte usta, natomiast na drugim oczu są szeroko otwarte, a usta zamknięte.



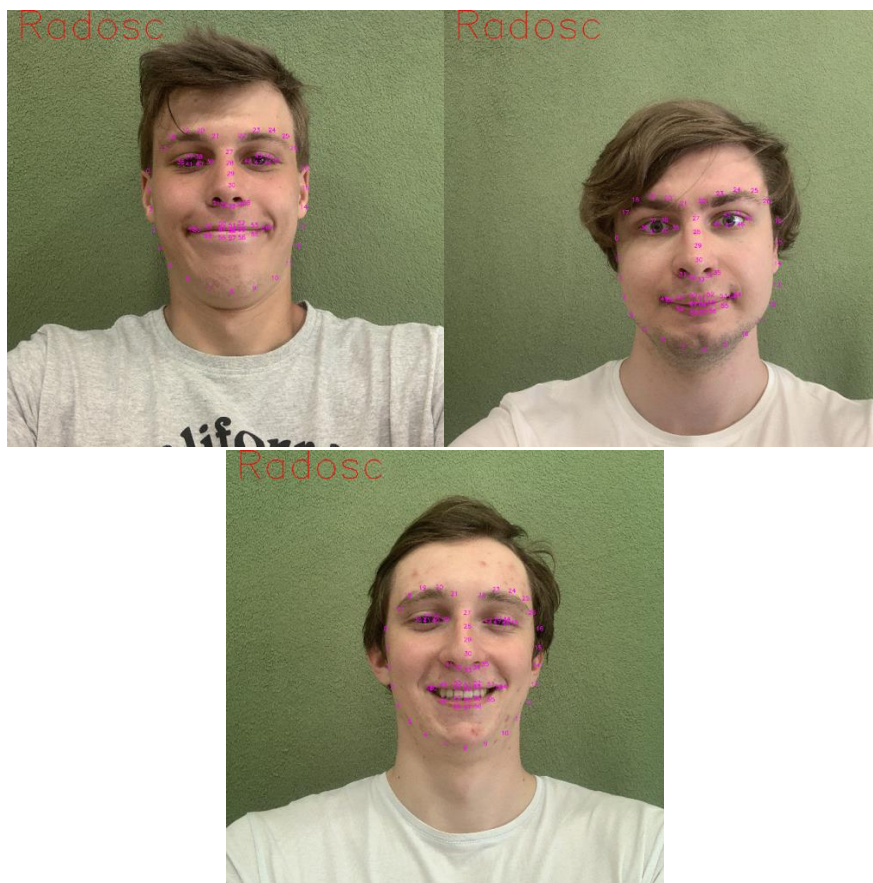
Rys. 7 Wyniki wykrywania zdziwionego wyrazu twarzy

W przypadku wykrycia miny typu „smutek” porównywane są brzegowe punkty ust z punktem na środku ust. Jeżeli brzegowe punkty znajdują się poniżej środkowego, to program wykrywa smutek. Wyniki działania przedstawiono na Rys. 8.



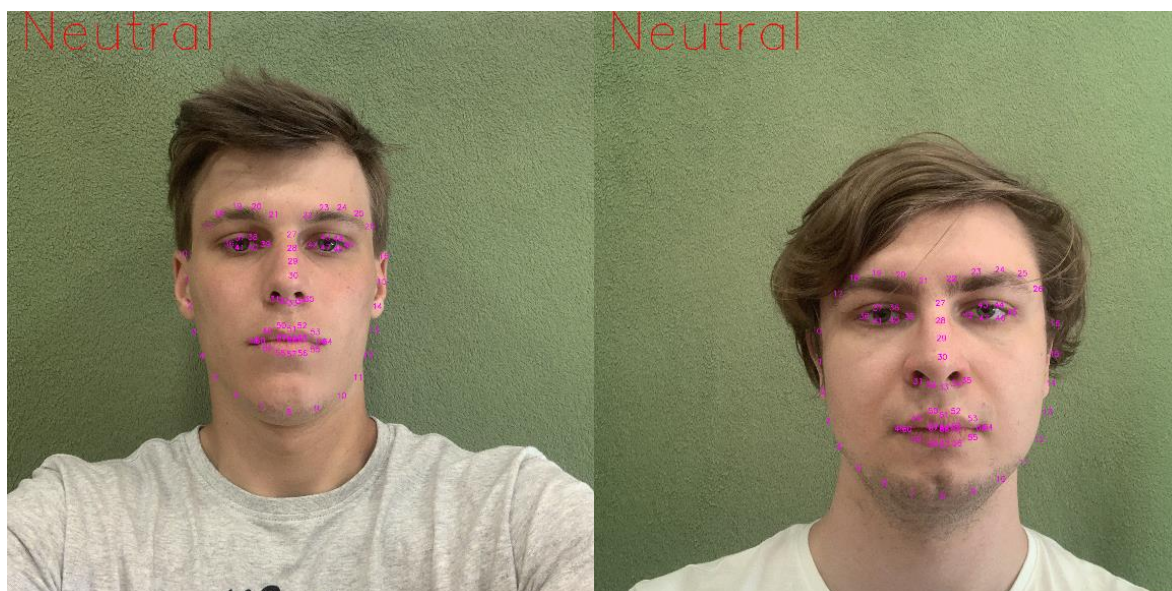
Rys. 8 Wyniki wykrywania smutnego wyrazu twarzy

Wykrycie radości zostało zrealizowane w odwrotny sposób - jeżeli brzegowe punkty znajdują się powyżej środkowego, to program wykrywa radość. Wyniki działania programu przedstawiono na Rys. 9.



Rys. 9 Wyniki wykrywania radosnego wyrazu twarzy

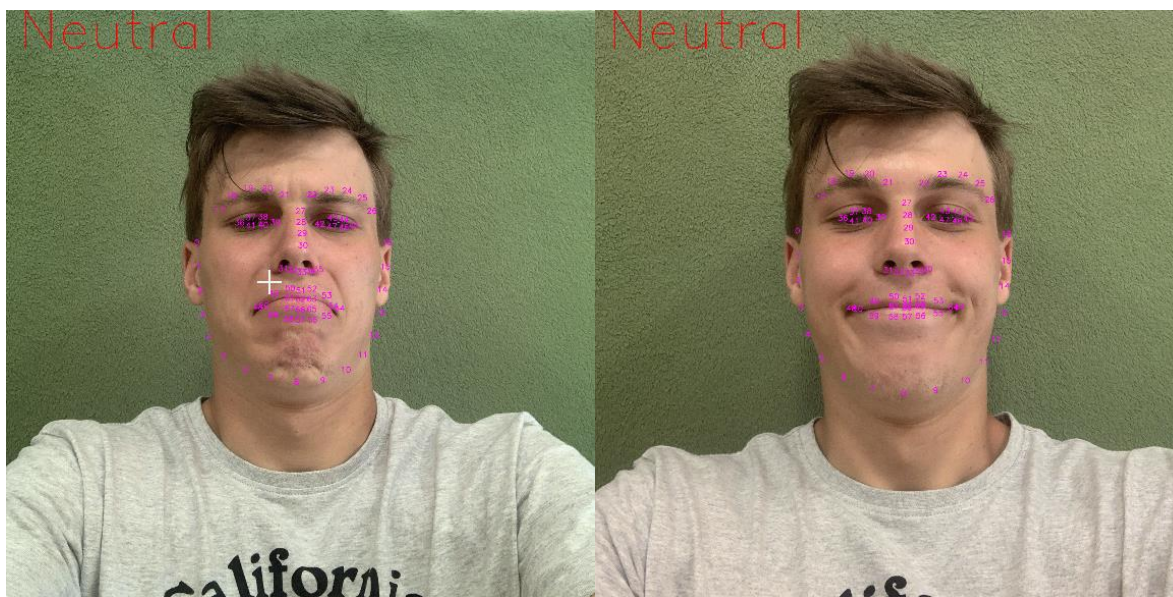
Jeżeli punkty orientacyjne na twarzy nie spełniły żadnego z przedstawionych powyżej warunków, to program wykrywa minę typu „neutralna” (Rys. 10).



Rys. 10 Wyniki wykrywania neutralnego wyrazu twarzy

Na wyniki działania programu mogą wpływać różne czynniki. Na pierwszym zdjęciu (Rys. 11) widać jak program źle wykrył punkty na ustach przez cień, znajdującą się pod dolną wargą. Problem może również występować przez to, że kolor ust jest podobny do koloru skóry na

twarzy. Na drugim zdjęciu został przedstawiony wpływ odchylenia głowy, co spowodowało błędne wykrycie współrzędnych punktów na końcach ust.



Rys. 11 Błędne wyniki wykrywania wyrazu twarzy

Rodzaj miny	Liczba obrazów	Liczba poprawnie rozpoznanych	Skuteczność rozpoznania [%]
Neutralna	25	23	92
Radosna	28	21	75
Smutna	29	14	45
Zdziwiona	28	25	90

4. Podsumowanie i wnioski

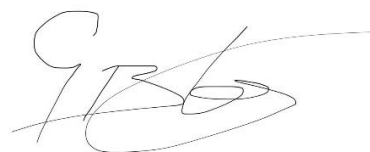
Wynik działania programu zależy od następujących czynników:

- Kolor ust i skóry,
- Występujące cienie na twarzy,
- Koloru i kontrastu tła,
- Pozycja głowy – odchylenie w stosunku do kamery,
- Charakterystyczność danego wyrazu twarzy,
- Oświetlenie

Niestety nie udało się zaimplementować programu w środowisku MATLAB z powodu niemożności wczytania poprawnego formatu plików zdjęciowych.

Oświadczenie

Oświadczamy, że niniejsza praca stanowiąca podstawę do uznania efektów uczenia się została wykonana przez nas samodzielnie.



Bohdan Holovko



Karol Urban



Michał Kądziołka