

RecognizeMe  
rozpoznawanie twarzy Amazon Rekognition



**Politechnika  
Śląska**

*Autorzy:*

Marta Lewandowska,

Krystian Barczak,

Aleksander Boronowski,

Wydział Matematyki Stosowanej

Kierunek Informatyka

VI semestr - grupa KUTAR3

# Spis treści

1	Opis programu . . . . .	2
2	Instrukcja obsługi . . . . .	2
3	Specyfikacja techniczna . . . . .	5
4	Szczegóły techniczne . . . . .	6

## 1 Opis programu

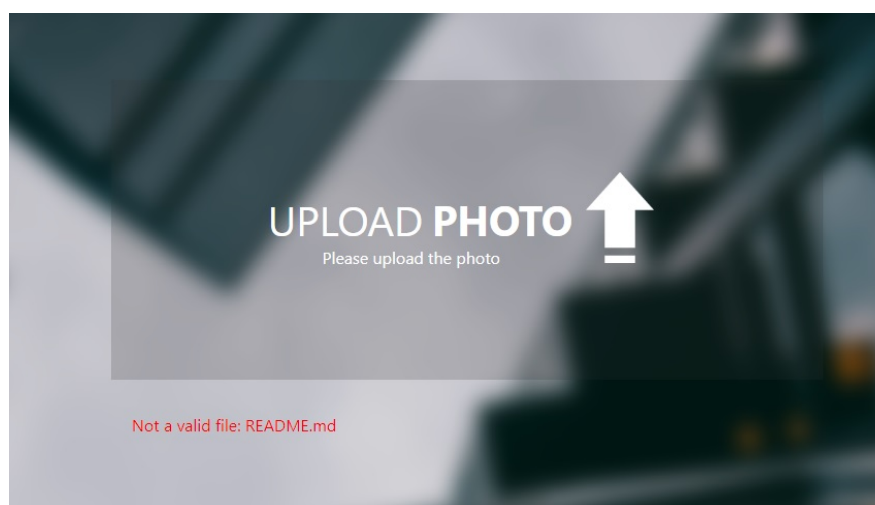
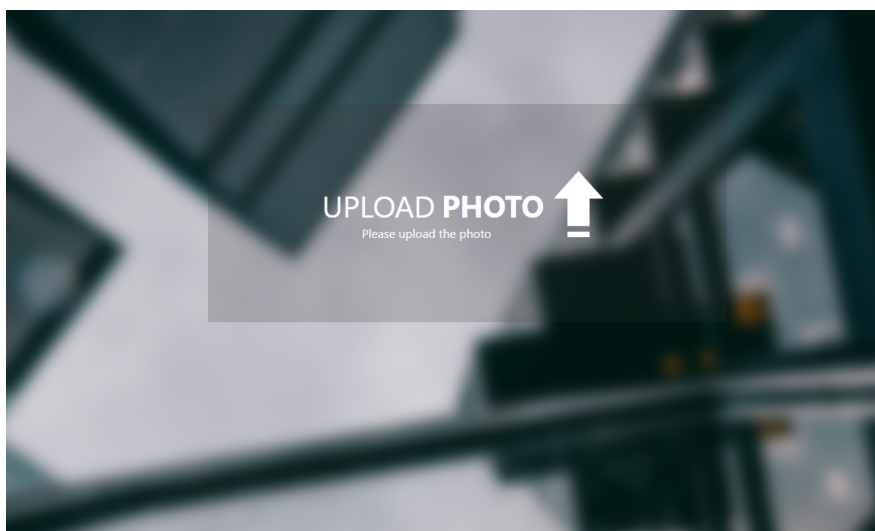
Webowa aplikacja pozwalająca na rozpoznanie twarzy wraz ze szczegółami z podanego przez użytkownika zdjęcia.

Program został wykonany w celu projektu zaliczeniowego z przedmiotu Projektowanie rozwiązań chmurowych z wykorzystaniem Amazon Web Services.

## 2 Instrukcja obsługi

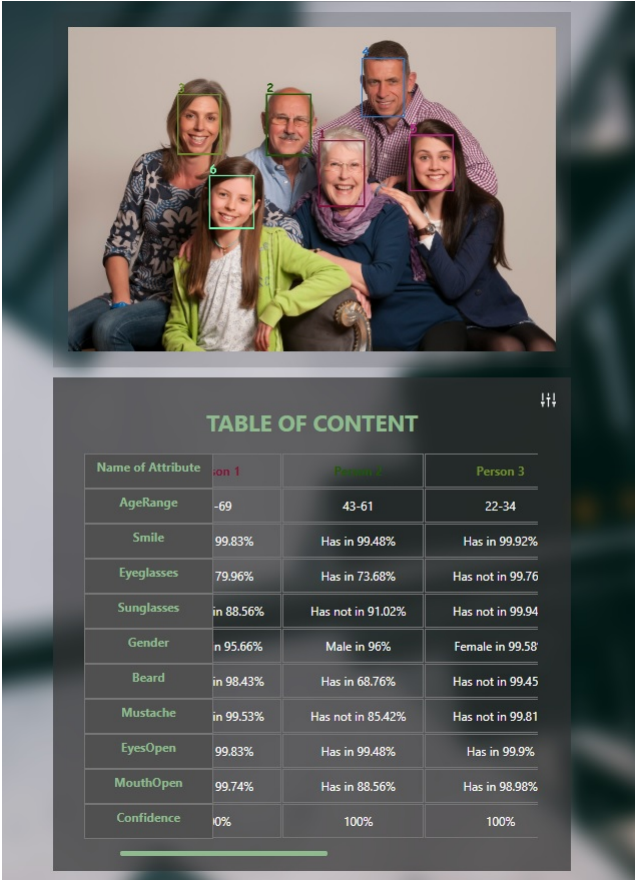
### 1. Upload własnego zdjęcia

W celu dodania zdjęcia, które użytkownik chce sprawdzić programem, należy kliknąć przycisk **"Upload photo"**, a następnie wybrać plik obrazu. Jeśli plik nie jest obrazem, wyświetli się błąd z nazwą podanego pliku.



## 2. Odczytywanie danych

Po poprawnym załadowaniu zdjęcia pod wyświetlonym obrazem użytkownika wyświetli się tabela z uzupełnionymi danymi. Pokazuje ona ilość wykrytych osób na zdjęciu oraz procent w jakim dany atrybut jest prawdopodobny dla zaznaczonej osoby. Jeśli na zdjęciu znajdują się dużo osób, tabele można przesuwac za pomocą scroll bara aby zobaczyć wszystkie dane.



The screenshot displays a group photo of five people. Each person's face is enclosed in a colored bounding box: green for Person 1 (top left), blue for Person 2 (top right), pink for Person 3 (bottom left), purple for Person 4 (bottom center), and red for Person 5 (bottom right). Below the photo is a table titled "TABLE OF CONTENT" with a scroll bar at the bottom. The table lists various attributes and their detection probabilities for the three people identified in the photo (Person 1, Person 2, and Person 3).

Name of Attribute	Person 1	Person 2	Person 3
AgeRange	-69	43-61	22-34
Smile	99.83%	Has in 99.48%	Has in 99.92%
Eyeglasses	79.96%	Has in 73.68%	Has not in 99.76
Sunglasses	in 88.56%	Has not in 91.02%	Has not in 99.94
Gender	n 95.66%	Male in 96%	Female in 99.58
Beard	in 98.43%	Has in 68.76%	Has not in 99.45
Mustache	in 99.53%	Has not in 85.42%	Has not in 99.81
EyesOpen	99.83%	Has in 99.48%	Has in 99.9%
MouthOpen	99.74%	Has in 88.56%	Has in 98.98%
Confidence	0%	100%	100%

### 3. Wyświetlanie danych

Użytkownik może wybrać, które dane mają wyświetlić się w tabeli. Wystarczy kliknąć przycisk opcji po prawej stronie oraz zaznaczyć lub odznaczyć interesujące go funkcje.

## TABLE OF CONTENT

☒ Age Range  
☒ Sunglasses  
☒ Mustache  
☒ Confidence  
☐ Sad  
☒ Fear

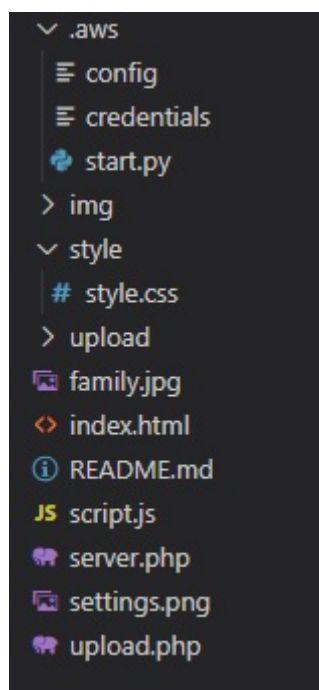
☒ Smile  
☒ Gender  
☒ Eyes Open  
☐ Happy  
☐ Surprised  
☒ Angry

☒ Eyeglasses  
☒ Beard  
☒ Mouth Open  
☐ Calm  
☐ Disguisted  
☐ Confused

Name of Attribute	Person 2	Person 3	Person 4
AgeRange	18-61	22-34	32-48
Smile	99.48%	Has in 99.92%	Has in 96.46%
Eyeglasses	73.68%	Has not in 99.76%	Has not in 98.39%
Sunglasses	in 91.02%	Has not in 99.94%	Has not in 99.42%
Gender	in 96%	Female in 99.58%	Male in 96.58%
Beard	68.76%	Has not in 99.45%	Has not in 89.59%
Mustache	in 85.42%	Has not in 99.81%	Has not in 98.84%
EyesOpen	99.48%	Has in 99.9%	Has in 99.71%
MouthOpen	88.56%	Has in 98.98%	Has in 97.55%
Confidence	100%	100%	100%
Fear	27%	3.44%	0.39%
Angry	52%	0.19%	2.65%

### 3 Specyfikacja techniczna

Podział na pliki:



**Kompilacja projektu:**

Do stworzenia projektu wykorzystany został program Microsoft Visual Studio Code oraz przeglądarki Google Chrome i Opera. Aplikacja napisana została w języku HTML oraz CSS. Funkcjonalność aplikacji natomiast napisana została w języku JavaScript, PHP i Python oraz framework jQuery.

## 4 Szczegóły techniczne

### 1. Funkcja odpowiedzialna za pobieranie danych JSON

```
function getJSON() {
    xmlhttp = new XMLHttpRequest();
    xmlhttp.open("GET", fileNameJSON, true);

    xmlhttp.onreadystatechange = function () {
        if (xmlhttp.readyState === XMLHttpRequest.DONE) {
            var status = xmlhttp.status;
            if (status === 0 || (status >= 200 && status < 400)) {
                document.getElementsByClassName("functionBox")[0].style.display = "none";
                document.getElementsByClassName("loader")[0].style.display = "none";
                clearInterval(refreshIntervalId);

                myArr = JSON.parse(this.responseText);
                lenghtOfPeople = myArr.FaceDetails.length;
                sendJSON(myArr);
            } else {
                var refreshIntervalId = setInterval(() => {
                    if (status === 0 || (status >= 200 && status < 400)) {
                        document.getElementsByClassName("functionBox")[0].style.display = "none";
                        document.getElementsByClassName("loader")[0].style.display = "none";
                        clearInterval(refreshIntervalId);
                    } else {
                        status = 0;
                        getJSON();
                    }
                }, 1500);
            }
        }
    };
    xmlhttp.send();
}
```

### 2. Funkcja odpowiedzialna za wysyłanie danych JSON do serwera PHP

```
function sendJSON(myArr) {
    xmlhttp = new XMLHttpRequest();
    xmlhttp.open("POST", "server.php", true);
    xmlhttp.onreadystatechange = function () {
        if (xmlhttp.readyState === XMLHttpRequest.DONE) {
            var status = xmlhttp.status;
```

```

        if (status === 0 || (status >= 200 && status < 400)) {
            document.getElementsByClassName("inner")[0].innerHTML = this.response;
            document.getElementById("photo").width = img.width;
            document.getElementById("photo").height = img.height;
            drawAll();
        } else {
            console.log("Oh no! There has been an error with the request!");
        }
    }
};
xmlhttp.send(
    JSON.stringify({
        send: true,
        fileNameToPy: file.name,
        length: myArr.FaceDetails.length,
        json: myArr,
    })
);
}

```

### 3. Funkcja odpowiedzialna za rysowanie Canvas oraz zaznaczanie na nim twarzy

```

function drawAll() {
    clearVariables();
    for (var i = 0; i < lenghtOfPeople; ++i) {
        imgHeight = document.getElementById("photo").height;
        imgWidth = document.getElementById("photo").width;

        var canvas = document.getElementById("photo");
        var ctx = canvas.getContext("2d");
        ctx.beginPath();
        ctx.lineWidth = "3";

        if (imgWidth > 500 && imgHeight > 300) {
            ctx.lineWidth = "6";
            ctx.font = "Bold 48px Comic Sans MS";
            ctx.fillText(i + 1, imgWidth * ratioX, imgHeight * ratioY - 10);
        }

        var getColor = randomColor();
        document.getElementById("colName_" + i).style.color = getColor;
        ctx.strokeStyle = getColor;
        ctx.fillStyle = getColor;
    }
}

```



```
var json = myArr["FaceDetails"][i]["BoundingBox"];

var ratioX = 0;
ratioX = json.Left;

var ratioY = 0;
ratioY = json.Top;
var ratioWidth = 0;
ratioWidth = json.Width;
var ratioHeight = 0;
ratioHeight = json.Height;

ctx.rect(imgWidth * ratioX, imgHeight * ratioY, imgWidth * ratioWidth,
ctx.stroke();
ctx.closePath();
}
document.getElementsByClassName("canvas-photo")[0].style.display = "block";
document.getElementsByClassName("main-tab")[0].style.display = "block";
}
```

#### 4. Funkcja odpowiedzialna za ustawienie obrazka

```
function setBackgroundAndName(file) {
    xmlhttp = new XMLHttpRequest();
    xmlhttp.open("post", "upload.php", true);
    xmlhttp.onreadystatechange = function () {
        document.getElementById("photo").style.backgroundImage = "url(upload/"
    };
    var data = new FormData();
    data.append("file", file);
    fileWithoutExt = file.name.replace(/\.[^/.]+$/, "");
    fileNameJSON = "upload/" + fileWithoutExt + ".json";
    xmlhttp.send(data);
}
```

## 5. Funkcje odpowiedzialne za generowanie koloru

```
function random(min, max) {
    var num = Math.floor(Math.random() * (max - min)) + min;
    return num;
}

function randomColor() {
    return "rgb(" + random(0, 255) + ", " + random(0, 255) + ", " + random(0, 255) + ")";
}
```

## 6. Funkcja odpowiedzialna za nasłuchiwanie zmian elementu Input

```
document.getElementById("file").addEventListener(
    "change",
    function () {
        file = this.files[0];
        if (file.type.split("/")[0] === "image") {
            document.getElementsByClassName("functionBox")[0].style.display = "block";
            document.getElementsByClassName("loader")[0].style.display = "block";
            document.getElementsByClassName("errorFormatter")[0].style.display = "block";
            setBackgroundAndName(file);
            fileName = file.name;
            getJSON();
            var _URL = window.URL || window.webkitURL;
            if ((file = this.files[0])) {
                img = new Image();
                img.src = _URL.createObjectURL(file);
            }
        } else {
            document.getElementsByClassName("loader")[0].style.display = "none";
            document.getElementsByClassName("functionBox")[0].style.display = "block";
            document.getElementsByClassName("errorFormatter")[0].style.display = "block";
            document.getElementsByClassName("errorFormatter")[0].style.color = "red";
            document.getElementsByClassName("errorFormatter")[0].innerHTML = "Not an image";
        }
    },
    false
);
```

## 7. Funkcja odpowiedzialna za wywołanie pliku Python

```
<?php

$target_path = "upload/" ;
if (!file_exists($target_path)) {
    mkdir($target_path, 0755, true);
}

$filename = basename($_FILES['file']['name']);
$tmp_name = $_FILES['file']['tmp_name'];
$target_path = $target_path . basename( $_FILES['file']['name']);

if(move_uploaded_file($tmp_name, $target_path)) {
    $procedure = ".aws/start.py ". $filename;
    exec($procedure);
} else{
    echo "There was an error uploading the file , please try again!";
}

exit;
?>
```

## 8. Funkcja odpowiedzialna za dynamiczne generowanie tabeli z przetworzonymi danymi

```
<?php
$f = fopen(" uneditable","a");
flock($f,LOCK_EX);

$rawdata = file_get_contents("php://input");
$dataJSON = json_decode($rawdata,true);
$ok = true;

if($dataJSON == null) {
    $result = array('status' => false , 'code' => 1, 'value' => 'Bad format'
    $ok = false;
}

$dataJSON['json']['length'] = $dataJSON['length'];

if($dataJSON['send'] == true){
    foreach($dataJSON['json']['FaceDetails'] as $chunk) {
        unset($chunk["Landmarks"],$chunk["Pose"],$chunk["BoundingBo
    }

    foreach ($dataJSON['json'] as $key => $value) {
        if($key == "FaceDetails"){
            $intcols = count($value);
        }
    }

    $array = [ 'AgeRange', 'Smile', 'Eyeglasses', 'Sunglasses', 'Gender', 'Bo
    'Mustache', 'EyesOpen', 'MouthOpen', 'Confidence' ];
    $arrayEmotions = [ 'Happy', 'Calm', 'Sad', 'Surprised', 'Disguisted', 'Fe

    echo "<table id='table'>";
    echo "<td class='headcol'>Name of Attribute</td>";
    for ($j=0;$j<$intcols;$j++) {
        $id = $j +1;
        echo "<td class='long' id='colName_". $j."'>Person ".$id."</td>";
    }

    for ($i = 0;$i<10;$i++) {

        echo "<tr id='". $i."'>";
        echo "<td class='headcol'>".$array[$i]."</td>";
```

```
        foreach($dataJSON['json']['FaceDetails'] as $chunk) {
            if ($chunk[$array[$i]] == $chunk['AgeRange']){
                echo "<td class='long'>". $chunk['AgeRange']
                ['High'] . "</td>";
            }
            else if($chunk[$array[$i]] == $chunk['Gender']){
                if($chunk['Gender']['Value'] == "Male"){
                    echo "<td class='long'>Male in ".ro
                }
                else {
                    echo "<td class='long'>Female in ".ro
                }
            }
            else if($chunk[$array[$i]] == $chunk['Confidence']){
                echo "<td class='long'>" . round($chunk['Co
            }
            else if($chunk[$array[$i]] != $chunk['AgeRange']){
                if($chunk[$array[$i]]['Value'] == 1){
                    echo "<td class='long'>Has in ".rou
                }
                else {
                    echo "<td class='long'>Has not in "
                }
            }
        }
        echo "</tr>";
    }

    for ($j = 0;$j<8;$j++) {
        $rowNumber = $j+10;
        echo "<tr id='". $rowNumber ." ' style='display:none;'>";
        echo "<td class='headcol'>". $arrayEmotions[$j] . "</td>";

        foreach($dataJSON['json']['FaceDetails'] as $chunk) {
            echo "<td class='long'>".round($chunk['Emotions'][$
        }

        echo "</tr>";
    }
    echo "</table>";
}

flock($f, LOCK_UN);
fclose($f);
unlink('uneditable');
```

?>

## 9. Skrypt Python odpowiedzialny za komunikację z serwerem AWS

```
import boto3
import json
import sys

reko = boto3.client('rekognition')

pic = str(sys.argv[1])

pic_w_ext = pic[: -4]
pic_json = "upload/" + pic_w_ext + ".json"

in_file = open("upload/" + pic, "rb")
pic_binary = in_file.read()
in_file.close()

response_binary = reko.detect_faces(
    Image={
        'Bytes': pic_binary
    },
    Attributes=[
        'ALL',
    ]
)

with open(pic_json, 'w') as f:
    json.dump(response_binary, f)
```

# Bibliografia

- [1] *<https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html>*
- [2] *<https://aws.amazon.com/rekognition/>*
- [3] *<https://docs.aws.amazon.com/AWSJavaScriptSDK/latest/AWS/Rekognition.html>*
- [4] *<https://stackoverflow.com>*