

BIO: Určení natočení hlavy na základě antropometrických vlastností

Brno, 2021

Radek Duchoň, xducho07

Ivana Saranová, xsaran02

1 ZADÁNÍ

Cílem projektu je vytvořit software, který na základě významných bodů v obličeji dokáže určit natočení hlavy. Součástí zadání jsou následující úkoly:

- 1.) Zprovozníte existující detektor významných bodů v obličeji.
- 2.) Vyberte vhodnou množinu bodů, kterou lze použít pro určení natočení obličeje.
- 3.) Zaměřte se především na určení *pitch*.
- 4.) Proveďte experimenty na datasetu, který vám dodá vedoucí projektu.

2 SOFTWARE

Projekt byl implementován v jazyce Python za použití knihoven `dlib`, `numpy`, `opencv`, `imutils` a `math`. Pro zpracování obrázku s obličejem a následný výpočet úhlů byla vytvořena třída `ImageAnalysis` s odpovídajícími metodami. Jednotlivé metody v této třídě a další pomocné metody budou rozebrány v dalších kapitolách.

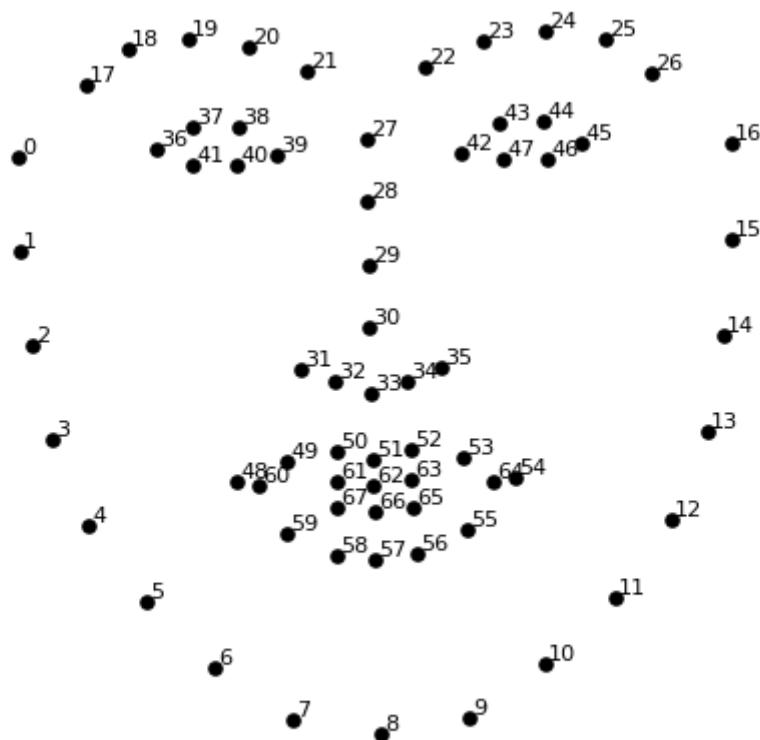
Je potřeba stáhnout prediktor¹ pro získání 68 významných bodů v obličeji a nainstalovat všechny knihovny zmíněné v souboru `requirements.txt`. Samotné spuštění programu je pak možné pomocí příkazu ``python3 main.py``. Cestu ke vstupnímu souboru lze měnit přímo v souboru `main.py`. Program předpokládá, že se střed otáčení nachází uprostřed obrázku.

3 ÚKOLY

1.1 Detektor významných bodů v obličeji

Pro získání polohy významných bodů v obličeji byl použit detektor knihovny `dlib` a prediktor pro určení 68 významných bodů na obličeji (viz Obrázek 1).

¹ http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2



The image is created by Brandon Amos from CMU that creates OpenFace

Obrázek 1 - 68 významných bodů v obličeji

Načtení významných bodů z obrázku má na starost metoda `get_landmarks()` v třídě `ImageAnalysis`. Tato metoda se volá hned při inicializaci instance třídy, takže při neúspěchu získání významných bodů není možno získat odpovídající úhly. Při mnoha větších rotacích *yaw* a *pitch* nebyl detektor schopen najít obličej, nebo získal významné body na špatných pozicích (např. úplně mimo tvář).

Jednotlivé významné body pak detektor vrátí v seznamu bodů, kde každý bod obsahuje *x*-ovou a *y*-ovou souřadnici, přičemž počáteční bod se nachází v levém horním rohu. Kód byl z velké části převzat z oficiálních stránek knihovny `dlib`.

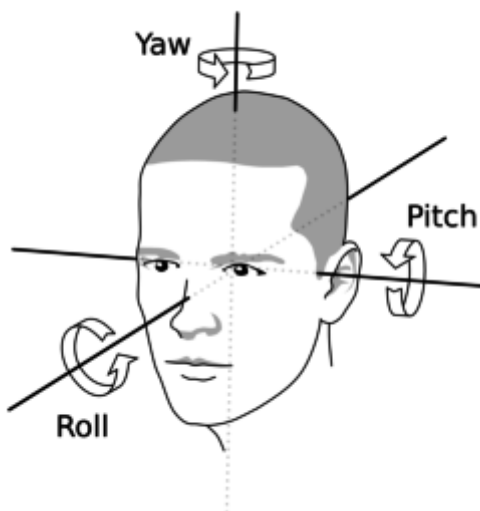
1.2 Vhodná množina bodů a určení jednotlivých rotací

1.2.1 Rotace *roll*, *yaw* a *pitch*

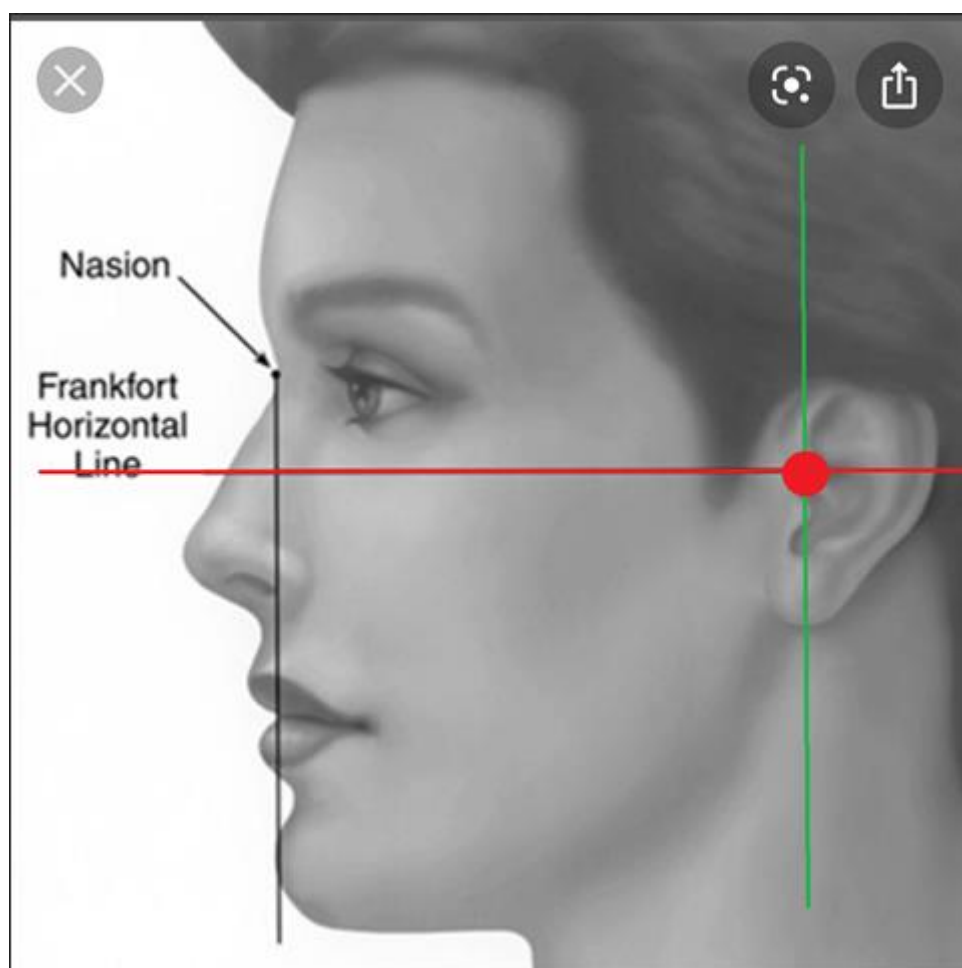
Roll, *yaw* a *pitch* označují 3D rotaci objektu kolem nějakého středu, který je nejčastěji umístěn ve středu daného objektu. V případě hlavy označuje *roll* nahnutí hlavy k jednomu nebo druhému rameni, *yaw* je otáčení hlavy doprava nebo doleva a *pitch* předklon a záklon hlavy. Náklon hlavy k pravému rameni je kladný směr, k levému záporný. Otočení hlavy doprava je záporný směr, doleva kladný. Předklon hlavy je kladný směr, záklon záporný.

Střed, podle kterého se provádí tyto rotace, se nachází v průsečíku tří rovin (viz Obrázek 3), kdy jedna vertikálně prochází středem nosu, druhá je rovnoběžná s Frankfurtskou horizontálou a třetí je kolmá k ostatním a prochází bodem *tragion* (v podstatě střed ucha).

V projektu předpokládáme, že *roll* je rotace kolem osy *z*, *yaw* kolem osy *y* a *pitch* kolem osy *x*. Pro určení všech rotací bylo využito metoda převedení problému do 2D roviny těch os, které se neúčastní dané rotace. Například *roll* je počítán ze souřadnic daných osami *x* a *y*.



Obrázek 2 - Rotace roll, yaw a pitch



Obrázek 3 - Střed otáčení

1.2.2 Určení roll

Implementaci výpočtu lze najít v metodě `get_roll_rotation()`. Tato rotace je určena poměrně jednoduše jako úhel mezi vektorem daným body očí a horizontálním vektorem (tedy vektorem rovnoběžným s osou x). Pro větší přesnost se využívá kombinace vektoru vnějších koutků očí, vnitřních koutků očí a středu očí v různých poměrech.

Důležitými významnými body v obličeji jsou pro tuto rotaci tedy oči, což je výhodné, protože ty lze poměrně dobře detekovat.

Před každým určením dalších z rotací jsou významné body v obličeji rotovány o *roll* (*alfa* v Obrázku 4) v rámci metody `fix_landmarks_to_zero_roll_rotation()`.

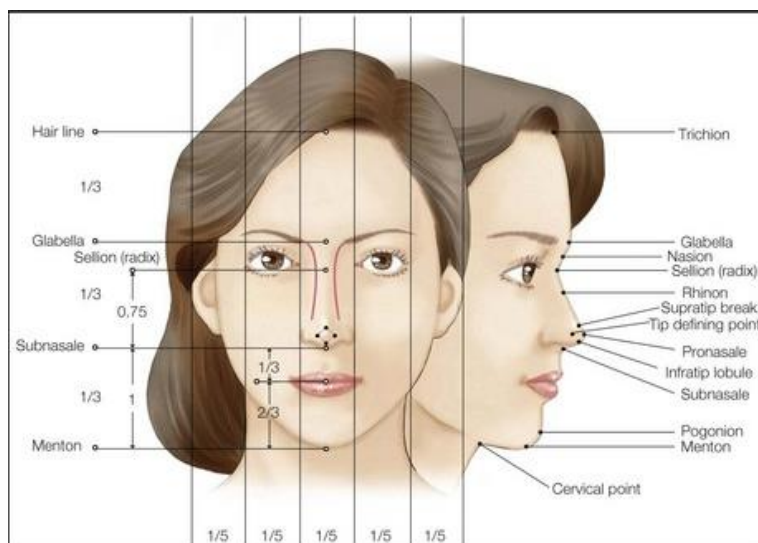


Obrázek 4 - Rotace roll, naznačení vektorů

1.2.3 Určení yaw

Implementaci výpočtu lze najít v metodě `get_yaw_rotation()`. Pro výpočet rotace yaw byl problém převeden do 2D systému s osami x a z , tedy v podstatě pohled na hlavu shora. Opět byly využity vnější koutky očí (levý koutek levého oka a pravý koutek pravého oka), ale také ještě bod *Sellion* (*radix*, kořen nosu, viz Obrázek 5). Dále bylo využito průměrných hodnot pro rozměry lidské hlavy²:

- 1.) Vzdálenost mezi týlem a *sellionem*.
- 2.) Vzdálenost mezi vnějšími koutky očí.

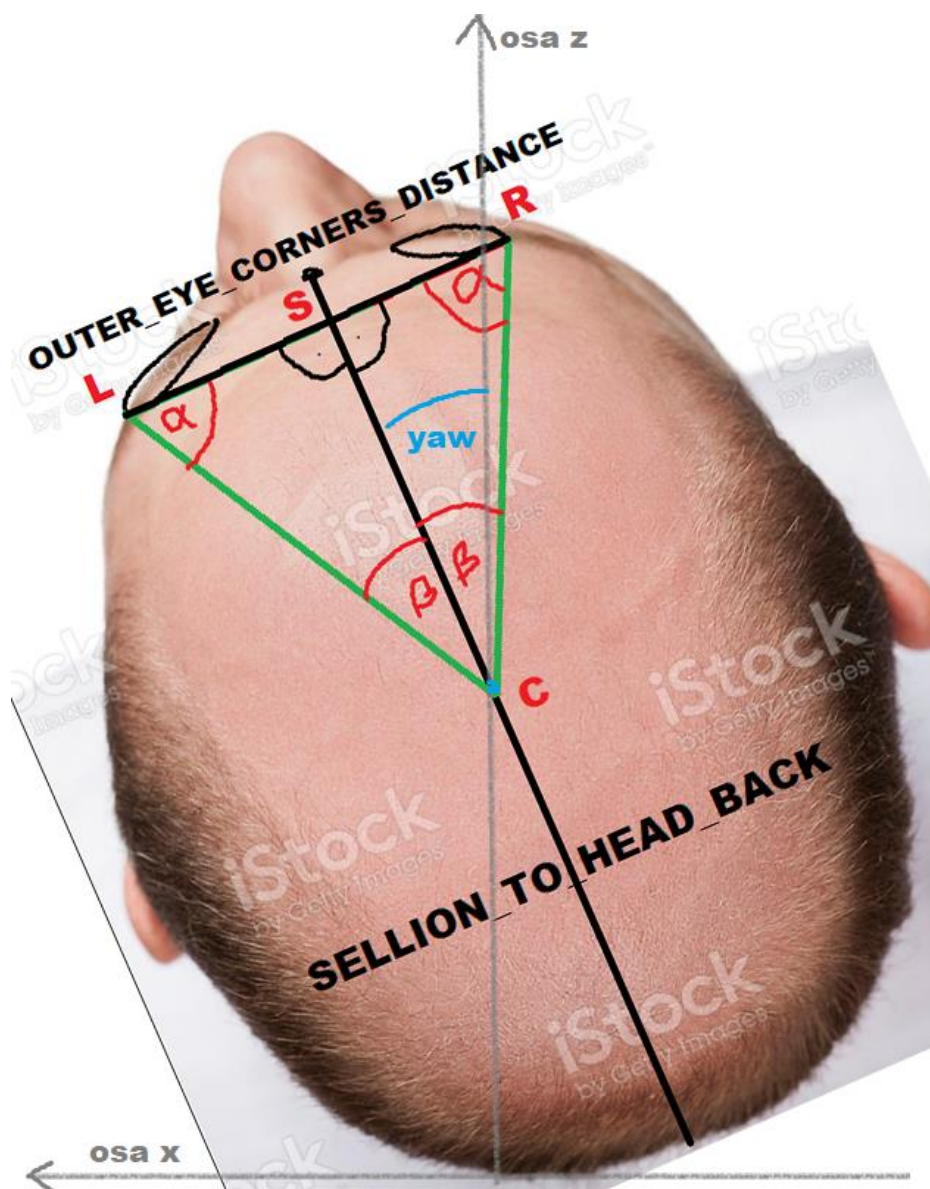


Obrázek 5 - Antropometrie obličeje

Střed otáčení je určen jako střed rozměrů obrázku (na základě datasetu, viz Obrázek 6). Měl by ležet uprostřed hlavy, tedy v polovině referenční vzdálenosti mezi týlem a *sellionem*. Dále známe x -ové souřadnice obou vnějších koutků očí a *sellionu*. Díky referenčním vzdálenostem můžeme určit poměr stran (koutek oka – *sellion* a *sellion* – střed) a očekávané úhly. Samotné určení z -ových souřadnic koutků oka a *sellionu* je iterativně odhadováno v cyklu (s přesností na jedno desetinné místo), kdy je vždy jednomu koutku oka a *sellionu* experimentálně určena souřadnice z , jsou vypočteny délky RS a CS a jejich poměr je porovnán s referenčním. Pokud je v požadovaném rozmezí chyby, považují se takto získané z -ové souřadnice jako platné a vypočítá se rotace yaw (známe SC a rozdíl x -ových souřadnic S a C). Využívá se vlastností rovnoramenného trojúhelníku, podobností trojúhelníku a goniometrických funkcí.

² https://en.wikipedia.org/wiki/Human_head

Určení, zda se jedná o kladnou nebo zápornou yaw rotaci, je v porovnání vzdálenosti pravého oka od *sellionu* a levého oka od *sellionu*. V případě, že je vzdálenost mezi pravým (levým) okem a *sellionem* větší než mezi levým (pravým) okem a *sellionem*, pak to znamená, že se hlava otáčí doleva (doprava), tedy kladným (záporným) směrem.



Obrázek 6 - Rotace yaw, naznačení způsobu řešení

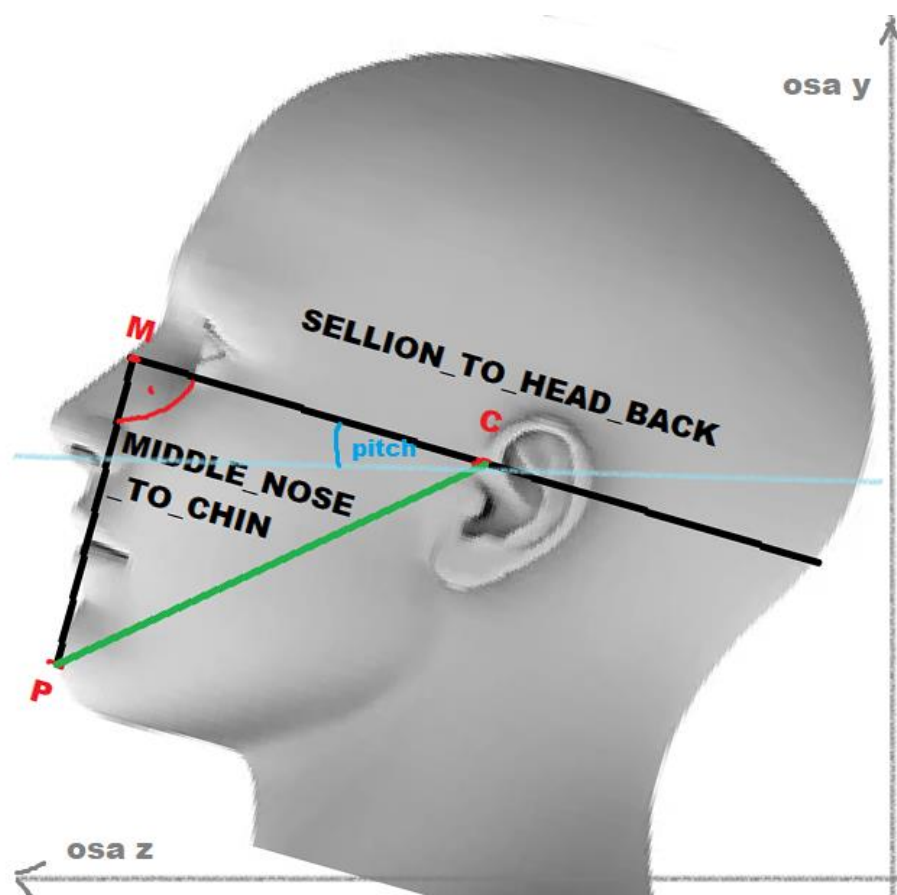
1.2.4 Určení pitch

Implementaci výpočtu lze najít v metodě `get_pitch_rotation()`. Určení rotace *pitch* je na stejném principu jako určení rotace *yaw*. Tentokrát se pohybuje mezi osami *z* a *y* a využíváme střed nosu a bradu (*pogonion*). Jako referenční vzdálenosti byly využity:

- 1.) Vzdálenost mezi týlem a *sellionem*.
- 2.) Vzdálenost mezi *sellionem* a vrcholem hlavy.
- 3.) Vzdálenost mezi bradou a vrcholem hlavy.
- 4.) Průměrná vzdálenost kořene nosu a bodem přechodu nosní přepážky v horní ret.

Ze vzdáleností 2-4 je vypočítána vzdálenost mezi středem nosu a bradou. Střed otáčení se opět nachází uprostřed hlavy na úrovni středu nosu (viz Obrázek 7). U středu nosu a brady známe pouze *y*-ové souřadnice. Díky znalosti referenčních vzdáleností můžeme vypočítat odhadovaný poměr mezi vzdálenostmi střed nosu – střed a brada – střed nosu. Následně iterativně určujeme experimentální hodnoty souřadnic *z* středu nosu a brady, vypočítáme potřebné vzdálenosti a porovnáme jejich poměr s referenčním. V případě shody je pak potřeba spočítat úhel svíraný vektorem osy *z* a vektorem danými střed a středem nosu.

Určení, zda se jedná o kladnou nebo zápornou rotaci, je v porovnání *y*-ových souřadnic středu nosu a středu, kdy v případě, že se střed nosu nachází nad (pod) středem, se jedná o záporný (kladný) směr.

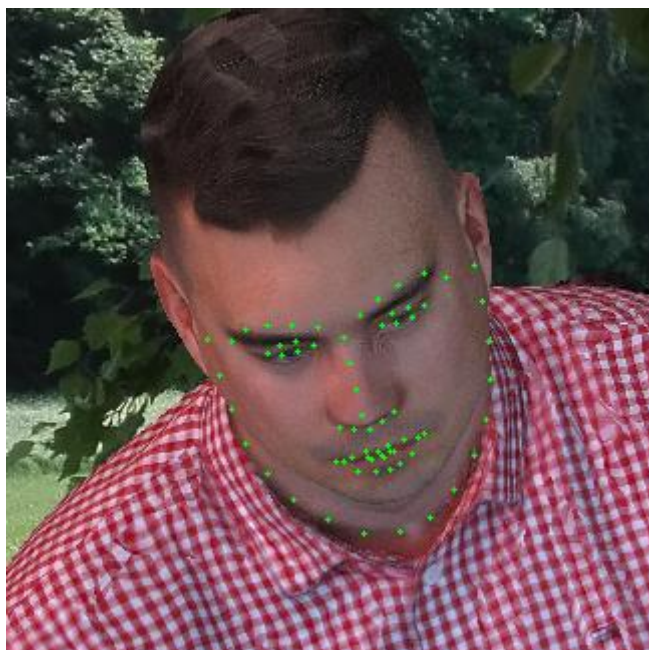


Obrázek 7 - Rotace pitch, naznačení způsobu řešení

2 EXPERIMENTY NAD DATASETEM

2.1.1 Problémy

Jak již bylo naznačeno, v některých případech si detektor a prediktor knihovny `dlib` neporadily s detekcí obličeje nebo určením významných bodů v obličeji. Obvykle se jednalo o obličeje vygenerované/neživé nebo o obličeje s větší rotací *pitch* a *yaw*.



Obrázek 8 - Chybné určení významných bodů

2.1.2 Odchyly od očekávaného řešení

Nad každým obličejem z poskytnutého datasetu byly vypočítány rotace *roll*, *yaw* a *pitch* a jejich odchylky od očekávaného řešení, které bylo součástí názvu obrázku s obličejem. Přibližně bylo zpracováno celkem 500 obrázků s různým natočením obličeje včetně chybných detekcí významných bodů (vzhledem k časové náročnosti zpracování jednoho obrázku nebylo v našich silách ručně zkontrolovat celý dataset).

Tabulka 1 - Výsledné hodnoty absolutních odchylek pro celý dataset

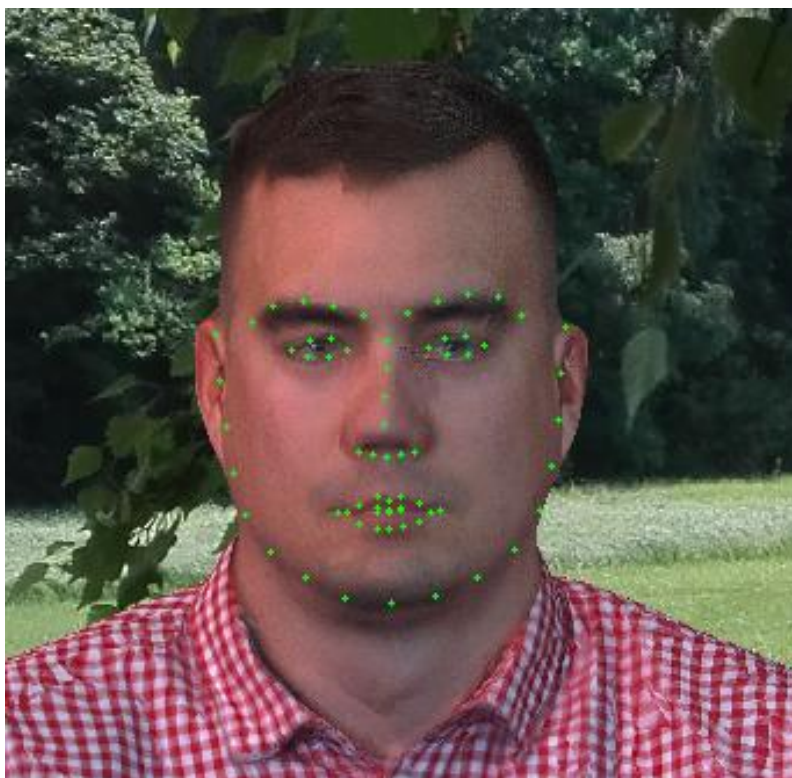
	Roll	Yaw	Pitch
Maximální absolutní odchylka (°)	21,31168579	21,75629406	40,89082476
Minimální absolutní odchylka (°)	0	0	0
Průměrná absolutní odchylka (°)	1,94796133	4,679822491	8,764041183
Medián absolutní odchylky (°)	1,023878925	3,558368578	6,597559949

2.1.3 Příklady odhadu jednotlivých rotací



```
Actual:    Roll -20.1 Yaw 16.7 Pitch 14.1  
Expected:  Roll -20.0 Yaw 10.0 Pitch 20.0
```

Obrázek 9 - dataset/Ap20.0_y10.0_r-20.0.png



```
Actual:    Roll 1.9 Yaw 0.0 Pitch 5.8  
Expected:  Roll 0.0 Yaw 0.0 Pitch 0.0
```

Obrázek 10 – dataset/Ap0.0_y0.0_r0.0.png



```
Actual:    Roll 19.5 Yaw -6.0 Pitch -10.8  
Expected:  Roll 20.0 Yaw -10.0 Pitch -10.0
```

Obrázek 11 - head/ p-10.0_y-10.0_r20.0.png

4 ZÁVĚR

Podařilo se zprovoznit detekci obličeje a prediktor významných bodů v obličeji v jazyce Python pomocí knihoven `dlib` a `opencv`. Jednotlivé rotace jsou určeny v rámci relativně dobrých odchylek pomocí knihoven `numpy`, `math` a dalších. Byly provedeny experimenty nad poskytnutým datasetem, které přinesly souhrnné výsledky i jednotlivé ukázkové příklady.