

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Návrh vestavěných systémů

Detekce objektů v obraze pomocí metod AI

Obsah

1	Úvod	1
2	Prerekvizity	1
3	Struktura projektu	1
4	Postup řešení	1
4.1	Klasifikátor	1
4.2	Ovladač videí	2
5	Limitace a slepé cesty	2
5.1	Klasifikace	2
5.2	Ovladač videí	3
6	Závěr	3

1 Úvod

Jako téma projektu bylo vybráno ovládání videí na Youtube prostřednictvím sady gest, jež má nahradit potřebu fyzické klávesnice. Cílem implementace bylo zkusit poskytnout možnost pohodlnější práce s tanečními videi během tréninků.

2 Prerekvizity

Pro funkcionální ovladače videí

- *Python 3.x* (Testováno na Python 3.8)
- *Mozilla firefox browser*
- *Geckodriver*

Pro kompilaci a upload kódu na ESP32-EYE

- *Arduino IDE*

3 Struktura projektu

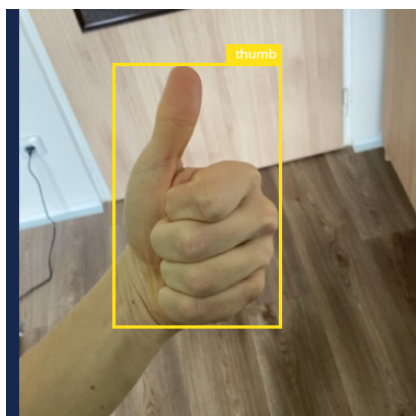
/	
├	esp32_code zdroje na kompilaci pro ESP-EYE
│	├ ei-inari007-project-1-arduino-1.0.3.zip knihovna pro klasifikátor
│	└ esp32_camera.ino Arduino kód
├	song_controller_code zdroje pro interpret na funkci ovladače videí
│	├ config.txt soubor s nastavením portu
│	├ file_parser.py zpracování vstupních souborů
│	├ links.txt přednastavené odkazy na videa s názvy pro UI
│	├ main.py hlavní soubor ke spuštění pro interpret
│	├ user_interace.py UI na selekci aktuálního videa
│	└ youtube_player.py ovladač videí na YouTube
├	doc.tex zdrojový kód dokumentace
└	doc.pdf dokumentace

4 Postup řešení

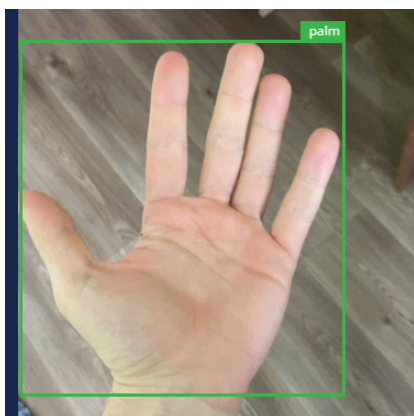
4.1 Klasifikátor

Klasifikátor byl vytvořen za pomoci doporučené platformy na trénování modelů **Edge Impulse**. Nejprve se vytvořil dataset o 3 klasifikačních třídách o velikosti 225 prvků. Ty byly sesbírány pomocí fotoaparátu běžného telefonu a na webové platformě jednotlivě označeny. Jedná se o třídy **thumb**, která představuje zvednutý palec, **palm** neboli vztyčenou dlaň a **left** jež jsou 2 prsty ukazující směrem doleva orientovaným od uživatele ke kameře. Dataset obsahuje pro každou třídu právě 75 označených fotografií.

Před trénováním se obrázky pomocí Edge Impulse převedly na rozměry 96x96 kvůli rychlosti vyhodnocování a paměťovým nárokům mikrokontroleru. Zároveň se převedly na grayscale opět pro výkonostní účely a také zlepšení generalizace. Pro samotné trénování byl zvolen model **FOMO MobileNetV2 0.1** a po několika testovacích bězích jsem konvergoval k zdá se neoptimálnějšímu parametrům a to 50 trénovacím cyklům s learning rate 0.001. Následně vygenerované knihovny s klasifikátorem se nachází ve složce *esp32_code* a pro správný průběh kompilace musí být naimportovány do Arduino IDE.



(a) Palec nahoře (thumb).



(b) Vztyčená dlaň (palm)



(c) Prsty doleva (left)

Knihovna poskytuje exemplární svazky kódu, jechž jeden z nich byl využit jako šablona pro implementaci a adekvátně modifikován, aby poskytoval na výstupu pouze název klasifikační třídy. Ty posílá v půl vteřinových intervalech na svůj USB port.

4.2 Ovladač videí

Pro ovladač videí byl zvolen skriptovací jazyk **Python** z důvodu podpory řady knihoven jež výsledná aplikace vyžaduje. Hlavní smyčka programu odposlouchává na sériovém portu, jež se definuje v souboru *config.txt*. V něm se také definují cesta k *firefox.exe* a cesta ke *geckodriver.exe*. Kromě něj se načte i soubor *links.txt*, která obsahuje uživatelův seznam libovolného výběru písniček ve tvaru:

[název] - [YouTube URL]

Po spuštění skriptu dojde k otevření nového okna prohlížeče. Pro manipulaci s ním byla využita knihovna **Selenium**. Skript reaguje na výstupní data z portu a volí akce na základě aktuální klasifikační třídy. Pro hodnotu **thumb** se video spustí či odpozastaví, pro **palm** dojde k pozastavení a pro **left** k přetočení obsahu zpátky o 5s.

Krom prohlížeče je uživateli po spuštění poskytnu malé uživatelské rozhraní běžící na dalším vlákne. Jeho funkcí je selekce z předdefinovaných písniček. Rozhraní obsahuje řadu tlačítek s názvy písniček, které po stisknutí přesměrují spuštěný prohlížeč na odpovídající URL. UI je implementováno pomocí knihovny **Tkinter**.

5 Limitace a slepé cesty

5.1 Klasifikace

Implementace klasifikátoru nejprve obsahovala 4. třídu a to **right** neboli přetáčení dopředu čítaje tak původní dataset o hodnotu 300 prvků. Natrénovaný model však poskytoval zavádějící a nekonzistentní výsledky. Tyto nešvary by byly s větší trénovací sadou omezeny, avšak pro naše řešení jsme zvolili tuto třídu eliminovat, jelikož se jednalo o nejméně důležitou funkci.

Pro vytvoření trénovací sady bylo využíváno řady prostředí, ale pouze ruky jednoho jedince. Pro jasnější klasifikaci byla veškerá gesta trénována pro rozpoznávání na **levé ruce**. Pro lehké obohacení testovací sady bylo využíváno odrazu zrcadla ruky pravé.

Klasifikace jednotlivých gest není jednoznačná, tudíž je pro minimalizaci chybovosti pro vykonání akce vyžadováno získání 2x stejného gesta po sobě. Toto se alespoň pro domácí prostředí osvědčilo jako spolehlivé řešení. Samotné vyhodnocení klasifikace trvá okolo 1s a výstup klasifikátoru je k dispozici přibližně každou 0.5s. Minimální doba programu ke zpracování klasifikace je tedy cca 1.5s.

5.2 Ovladač videí

Implementace se nejprve pokoušela o zpracování offline videí a manipulace s nimi a to pomocí knihoven jako **OpenCV**, **Pygame** nebo **pyvidplayer2**. Všechna řešení vyžadovala konvertovat video na externí zvukové stopy, jež probíhalo za běhu programu pomocí nástroje **FFmpeg**, avšak koordinace výsledného zvuku s videem byl hlavní klíč neúspěchu. Pro online videa bylo nejprve přihlíženo na prohlížeč **Google Chrome** využívající *chromedriver.exe*, avšak ten vyžaduje starší verze prohlížeče, jehož downgrade by způsobil odstranění cookies, což se jevilo bolestivě.

6 Závěr

Výsledné řešení funguje na omezené vzdálenosti v domácím prostředí relativně spolehlivě, po instalaci mikrokontroleru na specifickém vyvýšeném místě by dle mého názoru průběh domácích tréninků zefektivnil.

Pro zdokonalení řešení se nabízí především rozšíření datové sady a zprovoznění funkce přetáčení dopředu. Za dalšími užitečnými funkcemi by stálo rozšíření gest o změny rychlosti, které je v této oblasti zřídka využíváno a případné kreativní řešení přesunutí výběru písniček z uživatelského rozhraní do gestikulační podoby.