

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



Grafická a zvuková rozhraní a normy (GZN)
Dokumentace k projektu: Odstranění pozadí pomocí OpenCV a
TensorFlow

Duben 2020

Daniel Kolínek (xkolin05)

1 Zadání

Výsledkem práce bude aplikace realizující odstranění pozadí pomocí knihoven OpenCV a Tensorflow, přičemž jako popředí je brána lidská postava.

2 Segmentace osoby

Pro implementaci řešení je zapotřebí použít techniky segmentace osob v obraze. Pro vykonání operace segmentace byla využita knihovna BodyPix, která využívá konvoluční neuronovou síť MobilNetv1 a v nové verzi 2.0 ResNet50.

2.1 Bodypix

V počítačovém vidění se segmentace obrazu vztahuje na techniku seskupování pixelů v obraze do sémantických oblastí, které obvykle slouží k lokalizaci objektů a hranic. Model BodyPix je trénován tak, aby to zvládl pro osobu a dvacet čtyři částí těla (části jako levá ruka, přední pravá dolní část nohy nebo zadní část trupu). Jinými slovy, BodyPix může klasifikovat pixely obrázku do dvou kategorií: 1. pixely, které představují osobu, a 2. pixely, které představují pozadí. Může dále klasifikovat pixely představující osobu do kterékoli z dvaceti čtyř částí těla. [4] Bodypix je implementován pro použití v javascriptových aplikacích s pomocí Tensorflow.js.



Obrázek 1: Ukázka možného zobrazení výstupu Bodypix, při zvolení segmentace jednotlivých částí těla.

3 Implementace

Jelikož jsem se rozhodl použít Python ve spojení s knihovnami numpy a OpenCv bylo pro mě fakt, že je Boudypix pro použití pomocí javascriptové aplikace. Při průzkumu technologií vhodných pro projekt neexistovala knihovna tf-boudypix, o čemž svědčí i datum jejího vydání (16.10. 2020). Dvě možnosti, které se mi zdály nejvhodnější byly:

• 3.1 1. čistý Python

Stažení modelu pro Tensorflow.js v podobě json a vložení k němu. Načtení modelu a jeho následný převod do grafu vhodného pro tensorflow. Následný převod api z Boudypix pro neuronovou síť. Pro část s prací nad neuronovou sítí jsem využil a mírně upravil soubor utils.py z githubového repozitáře [3]. Samostatná implementace poté nebyl problém, ovšem jedná se o velice pomalý algoritmus a doporučuji její použití maximálně pro vyzkoušení. Pomalý je nejspíše z důvodu neoptimální implementace.

• 3.2 2. Python + nodejs

Jelikož je knihovna Boudypix pro javascript a chtěl jsem aby výpočet konvoluční sítě byly dostatečně rychlé pro použití v online telefonátech, zvolil jsem použití na nodejs. Je zapotřebí brát v potaz také informaci, že pokud použijeme tensorflowjs v pythonu, tak si buďto tahá implementaci z tensorflow nikoli z tensorflow-gpu nebo přímo z balíčku tensorflowjs, který ovšem také není smýšlený pro běh na grafické kartě (tuhle informaci neuvádí v dokumentaci, nebo jsem ji nenašel. Dohledal jsem ji až na fórech).

Vytvoří se tedy v první řadě pomocí nodejs server sloužící pro vyhodnocení segmentace a na něj se posílají dotazy z pythonu. Aby mohl server pracovat s grafickou kartou je zapotřebí nainstalovat si ovladač pro vaši grafickou kartu a následně cuda verze 10. Zde nastal další problém. Jelikož aktuálně používám Ubuntu verze 20.04 a cuda verze 10 je pouze maximálně pro Ubuntu 18.04, nezbyla jiná možnost, kromě downgradu na verzi Ubuntu, zkusit nainstalovat cudu verze 10 pro 18.04. Vše se zdá, že jede v pořádku ze strany tensorflow, ovšem po zadání příkazu nvidia-smi mi hlásí cuda verzi 11 a po zadání nvcc -version raději, že balíček neexistuje (toolkit je nainstalovaný).

3.3 tf-boudypix

Jelikož jsem při implementaci verze s nodejs chtěl celou situaci vzdát, než se mi podařilo vše rozběhnout na grafické kartě, hledal jsem, zda náhodou není nějaké api, které jsem přehlédnul. Našel jsem knihovnu tf-boudypix, díky které je celá implementace segmentace velice jednoduchá. Pouze se načte model pomocí `tf_boudypix.api.load_model` a následně vyhodnotit pomocí `boudypix_model.predict_single`. Stahování modelů je možné pomocí `tf_boudypix.api.download_model`, zjištění dostupných modelů v příkazovém řádku pomocí `python -m tf_boudypix list-models`.

4 Terminálová aplikace

Jak je nejspíše z popisu zřejmé, byly implementovány 3 metody, které jsou spojeny jedním spouštěcím skriptem, kde každá potřebuje různé balíčky. Tudíž první vyšší požadavky pro jednotlivé verze, co je potřeba udělat před spuštěním a následně až spuštění aplikace jako takové. Celá aplikace byla ozkoušena na:

- Python: Python 3.8.5, tensorflow 2.3.1, opencv-python 4.3.0.36, numpy 1.18.5, tf-bodypix 0.3.1
- OS: Ubuntu 20.04
- Nvidia: NVIDIA-SMI 455.32.00, Cuda 10

4.1 1. čistý Python

- **Požadavky** se vztahují pouze na balíčky pro Python 3. Neboli numpy, opencv, argparse, tensorflow, tensorflow-gpu.
- **Před spuštěním** je zapotřebí si stáhnout model do složky only_python pomocí scriptu download_model.sh.

4.2 2. Python + nodejs

- **Požadavky:**
 - **Python 3:** numpy, opencv, argparse, requets, pyfakewebcam
 - **Nodejs:** tensorflow-models/body-pix, tensorflow/tfjs-node, tensorflow/tfjs-node-gpu
 - **Nvidia:** pro případ zpuštění na grafice je zapotřebí si nainstalovat driver pro svou grafickou kartu a následně cuda 10. Pro případ rozjetí na CPU stačí v souboru python+node/node_part/app.js změnit načítání require('@tensorflow/tfjs-node-gpu') na require('@tensorflow/tfjs-node').
 - Pro spuštění na **web-kameře** nainstalujte balíčky v4l2loopback-dkms, modprobe.
- **Před spuštěním** je zapotřebí zapnout server pomocí nodejs node_part/app.js 0.75, kde 0.75 je práh pro konvoluční síť. Pokud chcete spustit webkameru, je zapotřebí si prvně vytvořit "fake" kameru, která bude sloužit výstup aplikace a tuto kameru následně v aplikacích pro stream videa nastavíte jako výchozí. Příkazy pro vytvoření "fake" kamery: modprobe -r v4l2loopback a následně sudo modprobe v4l2loopback devices=1 video_nr=5 card_label="v4l2loopback" exclusive_caps=1.

4.3 Bodypix-api

- **Požadavky:**
 - **Python 3:** numpy, opencv, argparse, pyfakewebcam, tf_bodypix, tensorflow, tensorflow-gpu
 - **web-kamera** v4l2loopback-dkms, modprobe
- **Před spuštěním** Pouze pokud zapínáte možnost pro webkameru, použijte kroky stejné jako v sekci 4.2.

4.4 Spuštění aplikace

```
remove_bg.py [-h] [--o option] [--img /path/to/image] [--video /path/to/video]
--bg /path/to/bg [--save filename] [--camerain videoIn] [--cameraout
videoOut] [--height heightForCamera] [--width widthForCamera] [--fps
fpsForCamera] [--threshold thresholdForDetection] [--starwars]
```

přímý popis jednotlivých parametrů naleznete v souboru README.md, který je přiložený ke zdrojovým souborům.

5 Závěr

Aplikace pracuje vcelku nad má očekávání. Výsledky jsou přiloženy u každé verze ve složce results a fungující verze pro webovou kameru bude předvedena u prezentace projektu. Výsledný ukázkový obrázek, kde je Harrison Ford dosazen na Tatooine byl vytvořen pomocí použití verze python+nodejs při nastavení prahu 0.75.



Obrázek 2: Python+nodejs, práh 0.75 [2][1]

Na zhodnocení přichází i import tensorflow na gpu, jelikož tato část mi zabrala docela velkou část času strávenou na projektu. Nemyslím si, že dokumentace k tensorflow-gpu a jeho sprovoznění na vlastní grafice je dostačující, jelikož jsem zjistil až po několika dnech, že má grafická karta Nvidia 940mx nemá dostačující výkon pro rozjetí tensorflow-gpu. Snažil jsem se o to hlavně kvůli tomu, že síť Resnet50 dosahuje lepších výsledků, ovšem se jedná o větší model, než je Mobilenetv1 a tudíž je výpočetně náročnější a i při rozlišení 640x480 nejsem schopen na svém procesoru Intel Core i5-6200U přesáhnout 10 fps.

Použitá literatura

- [1] e8e1e65b9b6d5f32a739e76ea058c062.jpg (1920×1080). <https://i.pinimg.com/originals/e8/e1/e6/e8e1e65b9b6d5f32a739e76ea058c062.jpg>. (Accessed on 11/08/2020).
- [2] O grito da selva — harrison ford pode ser estrela do longa - manual dos games. <https://manualdosgames.com/o-grito-da-selva-harrison-ford-pode-ser-estrela-do-longa/>. (Accessed on 11/08/2020).
- [3] simple_bodypix_python/utils.py at master · ajaichemmanam/simple_bodypix_python. https://github.com/ajaichemmanam/simple_bodypix_python/blob/master/utils.py. (Accessed on 11/05/2020).
- [4] [updated] bodypix: Real-time person segmentation in the browser with tensorflow.js — the tensorflow blog. <https://blog.tensorflow.org/2019/11/updated-bodypix-2.html>. (Accessed on 11/05/2020).