Rozpoznávání gest pomocí NVIDIA Jetson Nano

D. Pojhan[1], D. Majrich[2], and M. Sojka[3]

[1]Gymnázium, Plzeň, Mikulášské nám. 23, danpojhan@gmail.com
[2]SPŠSE Dukelská 13, České Budějovice, denis.majrich@gmail.com
[3]SPŠSE Dukelská 13, České Budějovice, miroslav.sojka@spssecb.cz

Abstrakt

Rozpoznávání gest, které přímo souvisí s odhadem pózy a gest, je dnes hodně důležitý obor v hlubokém učení a umělé inteligenci. Cílem našeho projektu bylo seznámit se s vývojovým prostředím NVIDIA Jetson Nano a algoritmy odhadu pózy jako je ResNet18 a DenseNet121. Výstup naší práce není jen pochopení pomůcek pro vývoj rozpoznávání gest, ale vytvořili jsme i program, který s vámi dokáže hrát kámen, nůžky, papír. Výsledky naší práce se nachází na:
https://github.com/HelloWorld7894/GestureDetection

1 Úvod

Odhad pózy a gest, který vytváří skelet objektu, je stále jeden z nejtěžších oborů počítačového vidění. Teprve nedávno (NeurIPS 2022) byl předveden nový state-of-art model ViTPose, který dosahoval pouze 81 AP * na datasetu COCO. Při našem testování jsme používali algoritmy ResNet18 a DenseNet121.
* (**A**verage **P**recision → metrika, používající se pro měření přesnosti modelů hlubokého učení)

2 NVIDIA Jetson Nano

Zařízení, na kterém jsme celý projekt testovali, byl malý jednodeskový počítač s GPU. Právě díky tomu je náš projekt přenosný a díky GPU s akcelerací výpočtů AI i dostatečně rychlý, NVIDIA zároveň nabízí SDK (software developement kit) pro deep learning.

specifikace:

GPU	128-core Maxwell GPU
CPU	Quad-core ARM Cortex-A57
paměť	4 GB LPDDR4 1600MHz
CUDA	10.2.
komunikace	GPIO, I2C, SPI, UART, USART
OS	Ubuntu 18.04 LTS

GPU - **G**raphical **P**rocessing **U**nit
CPU - **C**entral **P**rocessing **U**nit
CUDA - **C**ompute **U**nified **D**evice **A**rchitecture (nástroj pro akceleraci trénování a inference neuronové sítě)
GPIO - **G**eneral **P**urpose **I**nput/**O**utput (elektrický kontakt přes který prochází komunikace v integrovaném obvodu či jednodeskovém počítači)
I2C - **I**nter-Integrated **C**ircuit *
SPI - **S**erial **P**eripheal **I**nterface *
UART - **U**niversal **A**synchronous **R**eceiver/**T**ransmitter *
USART - **U**niversal **S**ynchronous/**A**synchronous **R**eceiver/**T**ransmitter *

* (Komunikační protokol mezi čipy a dalšími elektronickými součástkami na plošných spojích)

3 Neuronové sítě

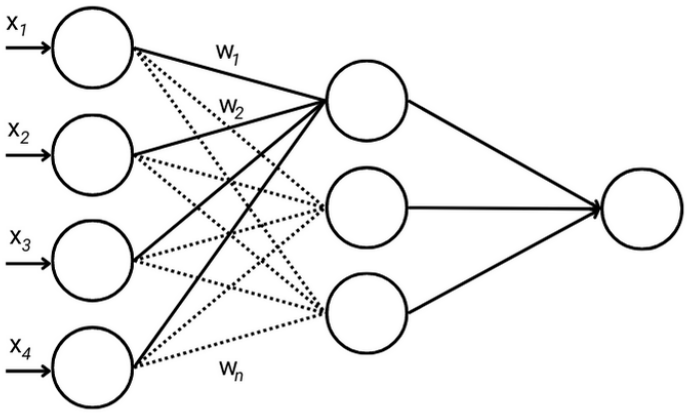
S rozšiřováním dostupných informací na internetu a komplexitou všech dat se kterými člověk musí pracovat, se také zvětšila časová náročnost různých úloh, které standardní algoritmy nezvládají. Díky těmhle problémům vznikly neuronové sítě.

Architektura těchto algoritmů je inspirovaná fugováním lidského mozku, a právě díky tomu se dokážou velmi dobře učit neurčitá data. Jejich nejběžnější aplikace jsou v klasifikaci a regresi, odkud se pak odvíjí více oborů jako např. sémantická segmentace, odhad pózy, klasifikace objektu, detekce objektu, atd.

Struktura

Stavební bloky neuronové sítě jsou tzv. neurony, které spolu vytváří vazby s různými váhami. Takže pokud bychom např. chtěli vypočítat hodnotu neuronu do kterého vedou váhy $w_1, w_2, ..., w_n$, rovnice bude vypadat:

$$y_1 = f_a(w_1 * x_1 + w_2 * x_2 + ... + w_n * x_n + b) = \sum_{i=1}^{n} w_i * x_i + b \tag{1}$$



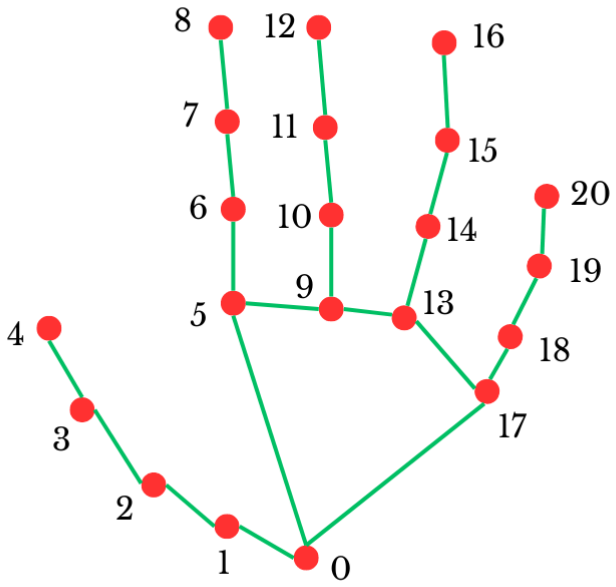
kde význam symbolů je následující:

$y_1 \leftarrow$ výstup neuronu
$f_a \leftarrow$ aktivační funkce neuronu, zavádí "nelinearitu"do neuronové sítě
$b_i \leftarrow$ bias neuronu *
$w_i \leftarrow$ váha vazby *
$x_i \leftarrow$ vstup

* (nastavovatelné parametry neuronové sítě, které se trénováním mění)

Všechny parametry, které jsou značeny *, se pak následně mění pomocí algoritmu sestupu gradientu.

4 Odhad pózy ruky



Náš přístup pro odhad pózy ruky používá pouze předdefinované matematické rovnice z výstupu neuronové sítě u některých bodů ruky. Tohle řešení má své meze, protože používá pouze přesně dané hodnoty, avšak výstup se může podle prostředí měnit. Tenhle problém by šel eliminovat natrénováním další neuronové sítě, která by brala výstup z té dosavadní a fungovala by jako nadstavba. K tomuhle řešení jsme však nestihli dojít, protože jsme neměli dostatek času na natrénování dané sítě.

Náš kód se dá zapsat následovně:

$$d_{ref} = \sqrt{|l_{5_x} - l_{6_x}|^2 + |l_{5_y} - l_{6_y}|^2} \leftarrow \text{referenční hodnota pro vzdálenost dvou kloubů}$$

$$d_{ti} = \sqrt{|l_{4_x} - l_{8_x}|^2 + |l_{4_y} - l_{8_y}|^2} \leftarrow \text{vzdálenost palce a ukazováčku}$$

$d_{li} =$ —"—(s indexy 20 a 8)

$d_{li} =$ —"—(s indexy 20 a 4)

$l_{i_x} \leftarrow$ kloub (landmark), kde i je index kloubu z celé ruky (od 0 do 20) a x je horizontální a y je vertikální souřadnice

$$f(x) = \begin{cases} 1 & \text{if } d_{li} < 3 * d_{ref} \\ 2 & \text{if } d_{ti} > 2 * d_{ref} \wedge d_{li} > 3 * d_{ref} \wedge d_{lt} > 3 * d_{ref} \\ 3 & \text{if } d_{ti} > 2 * d_{ref} \wedge d_{li} > 3 * d_{ref} \wedge d_{lt} < 3 * d_{ref} \end{cases}$$

$H_f = \{1, 2, 3\}$ ($1 \leftarrow$ kámen, $2 \leftarrow$ nůžky, $3 \leftarrow$ papír)

5 Shrnutí

Na tomto projektu jsme se seznámili s jednodeskovým počítačem NVIDIA Jetson Nano a algoritmy pro odhad pózy a gest z obrazu. Měli jsme možnost pracovat s výstupy neuronové sítě ResNet18 a výsledkem naší práce je hra kámen, nůžky, papír s počítačem, kde se detekují gesta pomocí kamery. Naše řešení není zdaleka optimální, a je zde mnoho možností jak ho vylepšit, jako např. použití další neuronové sítě která by detekovala gesta pomocí numerických výstupů z první sítě. Bohužel jsme ale na další vylepšení neměli v ohledu 2 dnů dostatek času.

6 Poděkování

Děkujeme organizátorům Týdne Vědy na Jaderce a hlavně pak také Ing. Jakubovi Klinkovskému, který byl skvělým vedoucím našeho projektu.

Reference

[1] NVIDIA *technical specifications of Nvidia Jetson Nano*. https://www.developer.nvidia.com/embedded/jetson-nano. 2019.

[2] Dustin Franklin *Hello AI World guide to deploying deep-learning inference networks and deep vision primitives with TensorRT and NVIDIA Jetson* . https://github.com/dusty-nv/jetson-inference/tree/master/docs. 2019.

[3] Ren Jie Tan *Breaking Down Mean Average Precision (mAP)*. https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52. 2019.

[4] Yufei Xu, Jing Zhang, Qiming Zhang and Dacheng Tao *ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation*. https://arxiv.org/pdf/2204.12484.pdf. 2022.