VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2020 Kristian Barna

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

VYUŽITÍ AKCELERÁTORU NEURONOVÝCH SÍTÍ NA RASPBERRY PI
VYUŽITÍ AKCELERÁTORU NEURONOVÝCH SÍTÍ NA RASPBERRY PI

BAKALÁRSKA PRÁCA
BACHELOR'S THESIS

AUTOR PRÁCE — **Kristián Barna**
AUTHOR

VEDÚCI PRÁCE — **doc. Ing. Vašíček Zdeněk, Ph.D.**
ADVISOR

BRNO 2020

Ústav počítačových systémů (UPSY) Akademický rok 2019/2020

Zadání bakalářské práce

Student: **Barna Kristian**
Program: Informační technologie
Název: **Využití akcelerátoru neuronových sítí na Raspberry PI**
Application of Neural Accelerators on Rapsberry PI
Kategorie: Umělá inteligence

Zadání:

1. Seznamte se s problematikou akcelerace hlubokých neuronových sítí pomocí dedikovaných hardwarových akcelerátorů jako je např. Intel Neural Compute Stick (NCS), Google Edge TPU (TPU), apod. Zaměřte se na princip využití akcelerátorů, zejména způsob převodu existujícího modelu neuronové sítě z typických frameworků jako je TensorFlow či Caffe tak, aby jej bylo možné akcelerovat daným akcelerátorem. Seznamte se s platformou Raspberry PI a možnostmi akcelerace neuronových sítí s využitím multimediálního koprocesoru VideoCore.
2. Zvolte vhodnou aplikaci zpracovávající obrazové sekvence, pomocí které bude možné demonstrovat způsob využití alespoň jednoho z akcelerátorů a vyhodnotit jeho výkonnost ve spojení s platformou Raspberry PI.
3. Navrženou aplikaci implementujte formou prototypu postaveného na platformě Raspberry PI vybavené kamerovým modulem. Model neuronové sítě se pokuste implementovat v prostředí TensorFlow případně Caffe tak, aby jej bylo možné spustit také přímo na CPU a případně GPU.
4. Experimentálně vyhodnoťte výkonnost a spotřebu hardwarového akcelerátoru na zvolené klasifikační úloze a proveďte srovnání s výkonností CPU / GPU a to na běžném PC i na Raspberry PI.
5. Dosažené výsledky analyzujte.

Literatura:

- Dle pokynů vedoucího.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Vašíček Zdeněk, doc. Ing., Ph.D.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 31. července 2020
Datum schválení: 25. října 2019

ABSTRAKT

Predkladaná bakalárska práca sa zoberá štatistickým vyhodnotením výkonnosti hardwardového akcelerátora hĺbkových neurónových sieti. Opisuje Konvolučné neurónové siete spolu s matematickými výpočtami. Vysvetľuje ich akceleráciu a prevod do formátu vhodného pre akcelerátor Intel Movidius NCS. Experimentálne sa porovnalo 8 hardvérových platforiem a 22 náročností neurónovej sieti. Bolo demonštrované až 105 násobné zlepšenie pre izolovanú inferenciu MobileNetV2 siete na platformu Raspbery Pi za pomoci akcelrátora. Výkon medzi testovanými platformami sa vyhodnocoval aj z energetického hľadiska. Aplikáciou na rozpoznanie identity tváre sa demonštrovali podmienky reálneho použitia. Odkryli sa možné limity akcelerácie CNN na zariadeniach s obmedzeným výkonom (Raspberry Pi), najmä v súvislosti s nevhodným výberom vstupného rozlíšenia obrazu. Všetky merania boli vyhodnocované štatistickými postupmi.

KĽÚČOVÉ SLOVÁ

počítačové videnie, konvolučné neurónové sieťe, hlboké učenie, rozpoznanie tváre, akcelerácia CNN, Intel Movidius NCS, OpenVino, Raspberry Pi

ABSTRACT

The presented bachelor thesis deals with the statistical evaluation of performance for hardward accelerator of deep neural networks. Describes convolutional neural networks along with mathematical calculations. Explains their acceleration and conversion to a format suitable for the Intel Movidius NCS accelerator. 8 hardware platforms and 22 neural network difficulties were compared experimentally. Up to 105-fold improvement was demonstrated in isolated inference of the MobileNetV2 network for the Raspber Pi platform using an accelerator. Performance between the tested platforms was also evaluated from an energy point of view. The application of facial identity demonstrated the conditions during real use. Possible limits of CNN acceleration on power-limited devices (Raspberry Pi) have been uncovered, especially due to improper selection of input image resolution. All measurements were evaluated by statistical procedures.

KEYWORDS

computer vision, convolutional neural networks, deep learning, face recognition, CNN acceleration, Intel Movidius NCS, OpenVino, Raspberry Pi

BARNA, Kristián. *Application of Neural Accelerators on Rapsberry PI*. Brno, 2020, 78 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií, Ústav počítačových systémů. Vedúci práce: doc. Ing. Vašíček Zdeněk, Ph.D.

VYHLÁSENIE

Vyhlasujem, že svoju bakalársku prácu na tému „Application of Neural Accelerators on Rapsberry PI“ som vypracoval samostatne pod vedením vedúceho bakalárskej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno ....

podpis autora

POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce pánovi Ing. Zdeněkovi Vašíčkovi, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Obsah

Zoznam obrázkov

Úvod

Vďaka prelomom vo výskume oblasti hĺbkových neurónových sietí došlo v posledných rokoch k výraznému rozmachu aplikácii s umelou inteligenciou. Od osobného hlasového asistenta cez systémy odporúčania produktov až po video monitorovanie a rozpoznávanie objektov je za každou z týchto aplikácii nejaká forma umelej inteligencie. Keďže tieto aplikácie využívajú relatívne zložité modely dáta sa často spracúvajú na centrálnych serveroch alebo na cloud platforme. Tento prístup má niekoľko rizík. Jedným je riziko narušenia súkromia. Ďalšou nevýhodou môže byť zvýšená latencia, to platí hlavne pre aplikácie spracúvajúce väčšie objemy dát alebo vyžadujúce real-time vyhodnocovanie. Navyše sa vďaka napredovaniu IoT neustále zvyšuje počet dát vygenerovaných na okraji site. Spracúvanie týchto dát na okrajoch siete vie efektívne znížiť alebo odstrániť tieto riziká. Navyše presunutie vyhodnocovania dát bližšie k ich zdroju zníži alebo eliminuje latenciu spojenú s rýchlosťou lokálnej siete. Schopnosť efektívne akcelerovať vyhodnotenie údajov na mieste ich záznamu, či uz je to mobil, chytrá kamera alebo analýza dát z IoT senzorov, je kľúčová pri návrhu takýchto aplikácii. Globálny posun týmto smerom ukazuje aj vývoj čoraz viac neurónových sietí určených pre mobilné zriadenia akou je napríklad aj MobileNet. Zároveň posun zaznamenali aj hardvérové zariadenia. V posledných rokoch vzniklo hneď niekoľko dostupných akcelerátorov ako Google Coral alebo Intel Movidius NCS, ktoré umožňujú ďalšie urýchlenie behu týchto sieti.

Prvá časť bakalárskej práce rozoberá základné prvky neurónových sietí, ich vznik a ich jednotlivé časti. Ďalej práca pokračuje v problematike predstavením Konvolučných neurónových sietí, ich architektúrou, vlastnosťami a procesom akým fungujú. V nasledujúcej časti sa rozobrala potreba akcelerovať tieto siete predstavili sa viaceré možnosti kacelerácia a preskúmali sa schopnosti vybraného zariadenia spolu so softvérovými možnosťami. Po oboznámení sa s problematikou mala práca 2 hlavé ciele. Experimentálne vyhodnotiť výkon a spotrebu vybraného akcelerátoru a porovnať výsledky s inými hw zariadeniami podľa zadania. Ďalším cieľom bolo bolo navrhnúť a implementovať aplikáciu na spracovanie obrazu, ktorá podporuje vybraný akcelerátor v spojení s platformou Raspbery Pi.

1 Úvod do neurónových sietí

1.1 Počiatok odvetvia umelých neurónových sieti

V roku 1943 neuropsychológ Warren McCulloch a matematik Walter Pitts uverejnili článok v buletínu matematickej biofyziky (The bulletin of mathematical biophysics) s názvom logický Ä Logical Calculus of Ideas Immanent in Nervous Activity". V ňom navrhli, že by nervová aktivita mohla fungovať podľa pravidiel dvojhodnotovej (binárnej) logiky a vytvoril, na tomto základe, jednoduchý matematický model neurónu. Tento princíp nazvali "všetko-alebo-nič"(all-or-none) a demonštrovali ho na modele jednouchej neurónovej sieti z elektrického obvodu.[1] Tento princíp v zásade vraví o tom, že sila odpovede neurónu na stimul je od neho nezávislá. Ak stimul prekročí aktivačnú hranicu neurón poskytne úplnú odpoveď. Neskôr v druhej polovici 50.-tych rokoch minulého storočia posun vo výpočetných schopnostiach vtedajších počítačov umožnil prvé digitálne simulácie neurónu a umelých neurónových sieti (artificial neural networks - ANN). Prelomový a najznámejší bola F. Rosenblattová simulácia Perceptronu ktorý bol simulovaný na počítači IBM 704[2] a preukázal schopnosť učenia sa (na rozdiel od Pittsovho modelu, ktorý túto schopnosť nemal) ako jeho biologický vzor.[3] Od vtedy vývoj v oblasti ANN ide ruka v ruke s vývojom na poli počítačového hardwaru. Vystriedalo sa niekoľko období vedeckého a publikačného záujmu s obdobiami zmenšenia financovania a výskumu tzv. zima umelej inteligencia (AI winter). Pravdepodobným dôvodom boli jasne preukázateľné nedostatky vtedajších návrhov, napríklad neschopnosť naučiť sa klasifikovať nelineárne rozdeliteľné problémy napr. xor funkcia, ale ja limity výpočetného výkonu. Prelom nastal po publikácií algoritmu spätnej propagácie chyby (backpropagation) v 70.-tych rokoch S. Linnainmaaom.[4] Algoritmom, ktorý umožnil efektívne učenie viacvrstvových neurónových sieti a ktorý sa používa dodnes. Do tohto typu a architektúry neurónových sieti, okrem iných, patria aj Hĺbkovéneurónové siete.

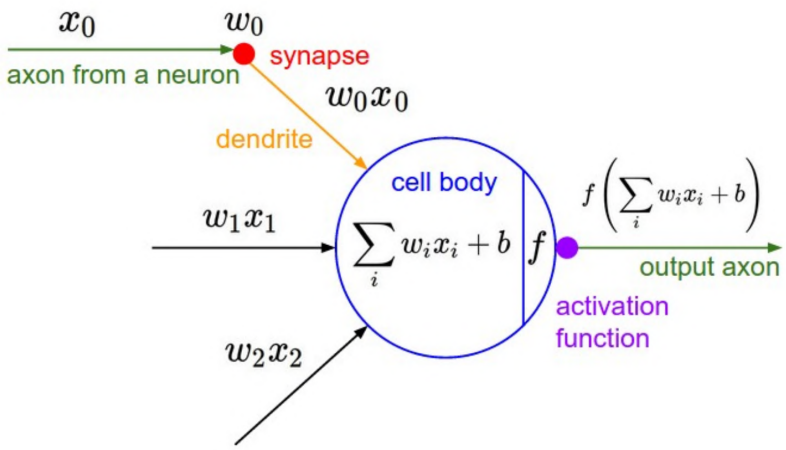
V tejto kapitole bude potupne vysvetlené tipy umelých neurónov, aké matematické operácie sa v nich používajú, ďalej bude vysvetlené čo je to Perceptrón, základné princípy fungovania neurónových sietí a na záver sa rozoberie problematika hĺbkových a špecificky konvolučných neurónových sieti .

1.2 Umelý neurón - základná stavebný blok umelých neurónových sieti

Umelý neurón, ako už bolo spomenuté, je v zásade matematickou aproximáciu jeho biologického originálu (obr. 1.1).

1.2.1 Biologický neurón

Biologický neurón môže byť popísaný ako jednotka ktorá prijíma vstupné signály cez *Dendridy*, spracúva ich v *Some* (v tele) a výstupná odpoveď sa prenáša cez Axon. Na rozdiel od *Dendridov*, ktorých je niekoľko, *Axon* je len jeden a na konci sa rozdeluje na na terminály ktoré sa viažu na *Dendridy* iných neurónov pomocou *Synapsie*. Pokiaľ kombinácia vstupných signálov na *Dendridoch* dosiahne aktivačný potenciál neurón vyšle jeden signál, ktorý sa prenesie cez Axon a *Synapsy* na pripojené *Dendridy* iných neurónov.[6] V realite sú tieto procesy zložitejšie a zatiaľ niesu známe všetky podrobnosti, preto je matematický model väčšinou výrazne zjednodušený.



Obr. 1.1: Diagram matematického popisu biologického neurónu[1]

1.2.2 Umelý neurón

Základnou funkciou umelého neurónu je prijať jeden alebo viac *vstupných signálov*, ktoré predstavujú čiastkové informácie. Ich vyhodnotenia s ohľadom na *prechodovú funkciu* a vyvolania zodpovedajúcej reakcie na výstupe podľa hodnoty *aktivačného prahu*. Ako je možné vidieť na obrázku (1.1) vstupné signály sú reprezentované vektorom $\vec{x}$. Každý zo vstupov má svoju váhu w obsiahnutú vo váhovom vektore $\vec{w}$. Účel tohto vektoru je určiť silu s akou ovplyvní príslušná hodnota x výsledný výstup umelého neurónu. Posledným parametrom je bias b. Je to konštanta ktorá umožňuje posun aktivačnej funkcie tak aby lepšie sedela pre dáta.

[1] <http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/>

1.2.3 Základná funkcia

Základná funkcia umelého neurónu sa vypočíta vynásobením vstupného vektoru $\vec{x}$ a váhového vektrou $\vec{w}$. Následne sa hodnoty výsledného vektoru sčítajú medzi sebou. Kvôli uľahčeniu si matematických operácii sa bias b pridáva do výpočtu ako vstup s váhou 1. V tomto kroku výpočtu sa používajú najmä **operácie sčítania a násobenia** aj keď v niektorých prípadoch sa namiesto sumy môže použiť iná operácia (Max, Min, Average, And, ...)[7].

$$p = \sum_{i=0}^{n} w_i x_i + b \tag{1.1}$$

Výsledná hodnota p (vzorec 1.1) reprezentuje Základnú funkciu neurónu a je ďalej poslaná do aktivačnej funkcie. Funkcia potom vypočíta výstup umelého neurónu na základe tohto čísla.

1.2.4 Aktivačná funkcia

Existuje mnoho druhov umelých neurónov. Najvýznamnejšou odlišnosťou je použitá prechodová / aktivačná funkcia. Aktivačná funkcia určuje schopnosť vyjadrenia neurónových sieti. Tento fakt naberá na dôležitosti, čím sú zložitejšie neurónové siete (Konvlučné neurónové siete) a čím komplexnejšie dáta. [8] Aktivačná funkcia sa dá rozdeliť do troch skupín[8]:

1. Binárna kroková funkcia
2. Lineárna aktivačná funkcia
3. Nelineárna aktivačná funkcia

Binárna kroková funkcia (obr. 1.2) je aktivačná funkcia založená na aktivačnom prahu. Podľa toho či sú vstupné hodnoty nad alebo pod hodnotou prahu, neurón je aktivovaný a vyšle signál s presne rovnakou intenzitou. Princíp all-or-none[1].

$$y = f(p) = \begin{cases} \text{a} & \text{p} < 0 \\ \text{b} & \text{p} \geq 0 \end{cases} \tag{1.2}$$

Problém tejto funkcie je, že neumožňuje viac hodnotové výstupy, teda neumožňuje iný ako binárny klasifikátor.

Lineárne aktivačné funkcie (obr. 1.3) má formu:

$$f(p) = kp + q \tag{1.3}$$

Táto funkcia berie Základnú funkciu p (vzorec 1.1) a vytvára výstupný signál proporčný ku vstupnému. Výhodou oproti Binárnej krokovej funkcii je, že umožňuje

viaceré výstupy nie len binárne hodnoty. Veľkou nevýhodou je, že viacvrstvové[5] siete výhradne s lineárnou aktivačnou funkciou sa správajú ako jednovrstvové. Vychádza to z faktu, že posledná vrstva je lineárnou funkciu prvej vrstvy (lineárna kombinácia lineárnych funkcií je stále lineárna funkcia). To znamená, že hlboká lineárna neurónová sieť je de facto lineárny regresný model a nedokáže správne vyriešiť nelineárne problémy.[8] Ďalšou nevýhodou je, že sa pri trénovaní nemôže použiť metóda spätnej propagácie chyby[2].

Nelineárna aktivačná funkcia je často používaná v moderných neurónových sieťach. Umožňuje komplexné mapovania medzi vstupmi a výstupmi siete. Toto mapovanie je základom pre naučenie a modelovanie komplexných dát ako obraz, video, audio a ďalšie dáta ktoré majú vysokú dimnezionalitu. Nelineárne aktivačné funkcie teda riešia vyššie zmienené nevýhody Lineárnych aktivačných funkcií. Umožňujú vytváranie hlbokých neurónových sietí skladaním vrstiev na seba a ich trénovanie pomocou spätnej propagácie chyby. To ich robí vhodnými kandidátmi pre klasifikačné problémy. Nelineárnych aktivačných funkcii je niekoľko avšak v poslednej dobe najpoužívanejšie sú ReLU a Softmax.

1. **ReLU** (Rectified Linear Unit) (obr. 1.4) sa v súčastnosti používa skoro v každej konvolučnej alebo hĺbkovej neurónovej sieti. To ju pravdepodobne robí momentálne najpoužívanejšou aktivačnou funkciou. Výhodou je jej výpočtová efektivita čo ju robí rýchlou pri učení. Je veľmi používaná pri spracovaní obrazu keďže ignoruje záporné hodnoty, čo je žiadúce pri práci s hodnotami pixlov. Je často kombinovaná s Softmax funkciou v klasifikačných problémoch.

$$ReLU(p) = max(0, p) \tag{1.4}$$

   - p je základná funkcia umelého neurónu (vzorec 1.1)

2. **SoftMax** využíva sa často v poslednej vrstve, klasifikačných sieti s viacerými triedami, kde normalizuje vstupné hodnoty do pravdepodobnostného rozloženia v intervale $(0, 1)$. Táto funkcia je definovaná nasledovne:

$$Softmax(p)_j = \frac{e^{p_j}}{\sum_{k=1}^{n} e^{p_j}};\ pre\ j = 1, ..., n \tag{1.5}$$

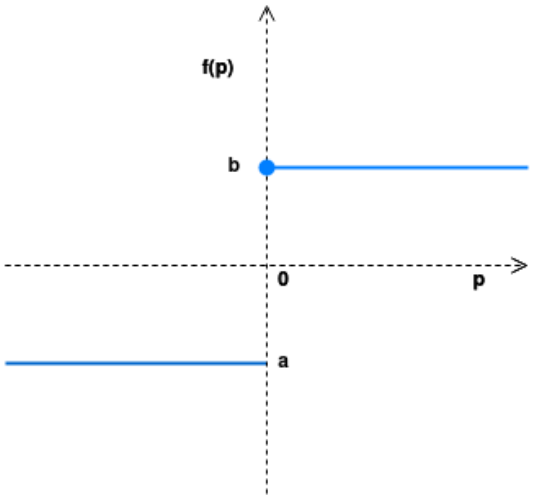
   - p je základná funkcia umelého neurónu (vzorec 1.1)
   - n je veľkosť vstupného vektoru

   Číslo z intervalu reprezentuje pravdepodobnosť s akou patrí vstup do jednotlivých kategórií / tried, teda suma výstupného vektoru je 1 (viď. rovnica 1.6).

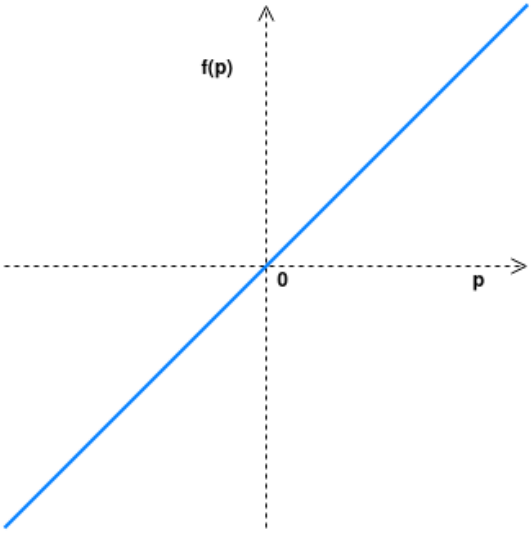
[2]Presný dôvod prečo sa nemôže použiť backpropagation je nad rámec tejto práce. Avšak vychádza to z faktu, že derivácie lineárnej funkcie je konštanta a tým pádom algoritmus nieje schopný určiť, ktoré váhy treba zmeniť

$$\sum_{j=1}^{n} \sigma(z)_j = 1 \tag{1.6}$$

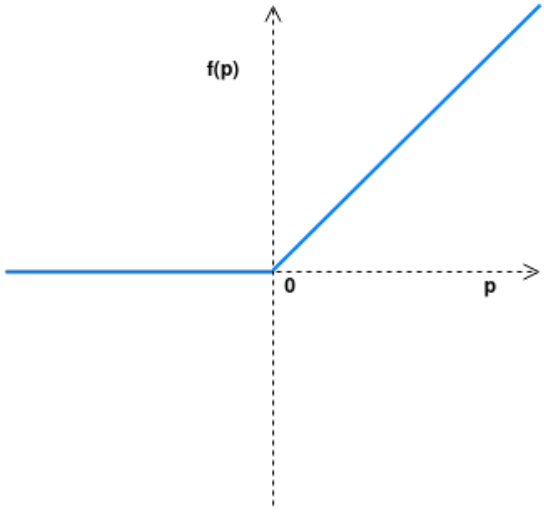
1.2.5 Grafy aktivačných funkcií



Obr. 1.2: Binárna kroková funkcia



Obr. 1.3: Linearna aktivačna funkcia



Obr. 1.4: Grafy ReLU funkcie

1.2.6 Zhrnutie

Ukázali sme, že umelý neurón je matematickou aproximáciou biologického neurónu. Ďalej sme ukázali, že táto aproximácia je založená na výpočte dvoch funkcií *zá-*

kladnej funkcie (1.2.3) a *aktivačnej funkcie* (1.2.4). Pri výpočte základnej funkcie sa pre každý vstupný "dendrit"vykonávajú operácie násobenia a sčítania. A potom pri výpočte aktivačnej funkcie sa vypočíta jedna (lineárna alebo nelineárna) rovnica. Vidíme teda, že s narastajúcou zložitosťou (narastajúci počet umelých neurónov a prepojení) siete narastá násobne viac výpočtová zložitosť základnej funkcie oproti aktivačnej funkcii. Tento jav je najväčší pre plne prepojené neurónové siete.

1.3 Neurónové siete

Exituje veľké množstvo rôznych druhov modelov neurónových sietí. Prvím a najjednoduchším modelom je Percetpron. Patrí do kategórie modelov s lineárnou aktivačnou funkciou, čiže značne obmedzený (viď. 1.2.4). Ďalej existujú modely vhodné pre tzv. učenie bez učiteľa, tieto modely sa učia na neoznačených dátach. Cieľom je aby sieť rozdelila vstupné dáta do skupín podľa podobnosti bez toho aby poznala správnu odpoveď[9]. Takouto sieťou je napríklad autoenkóder. Alebo Rekurentné modely, ktoré majú vo svojej architektúre minimálne jedno cyklické spojenie. A mnoho iných. V tejto kapitole budú popísané Konvolučné neurónové siete, ktoré sú špeciálny prípadom viacvrstvových neurónových sieti.

1.3.1 Viacvrstvový Perceptron

Viacvrstvový Perceptron (angl. Multy Layer Perceptron - MLP) je model, ktorý sa delí na tri základné časti. Každá časť reprezentuje minimálne jednu vrstvu z modelu.

1. Vstupnú vrstvu - Prijíma vstupné dáta. Jej úlohou je sprostredkovať vstupné dáta pre ostatné vrstvy bez zmeny. Počet neurónov tejto vrstvy sa rovná počtu vlastností vstupných dát. Za ňou nasledujú skryté vrstvy.
2. Skryté vrstvy - Tieto vrstvy niesu v priamom kontakte s okolím siete. Táto vrstva môže byť reprezentovaná jedným neutrónom alebo veľkým počtom vrstiev neurónov. Čím má sieť viacej vrstiev, tým je sofistikovanejšia a lepšie rieši zložité problémy ale zároveň trvá dlhšie jej tréning.
3. Výstupná vrstva - Je to posledná vrstva siete. Tu dochádza k finálnym výpočtom a in interpretácii výstupu zo skrytých vrstiev. Forma, ktorou prezentuje dáta sa mení z jedného tipu problému na druhý. Napríklad binárny klasifikátor bude mať výstup vo forme 1 al. 0 oproti tomu viac-triedny klasifikátor bude mať viac výstupných neurónov, jeden pre každú triedu.

Základnou vlastnosťou týchto sieti je, že sa vedia natrénovať na riešenie nelineárnych problémov.

Avšak niesu veľmi vhodné pre analýzu obrazu. Keďže vstupná vrstva má jeden neurón pre každý vstup pre obrázok to znamená $h * w * 3$ vstupných neurónov.

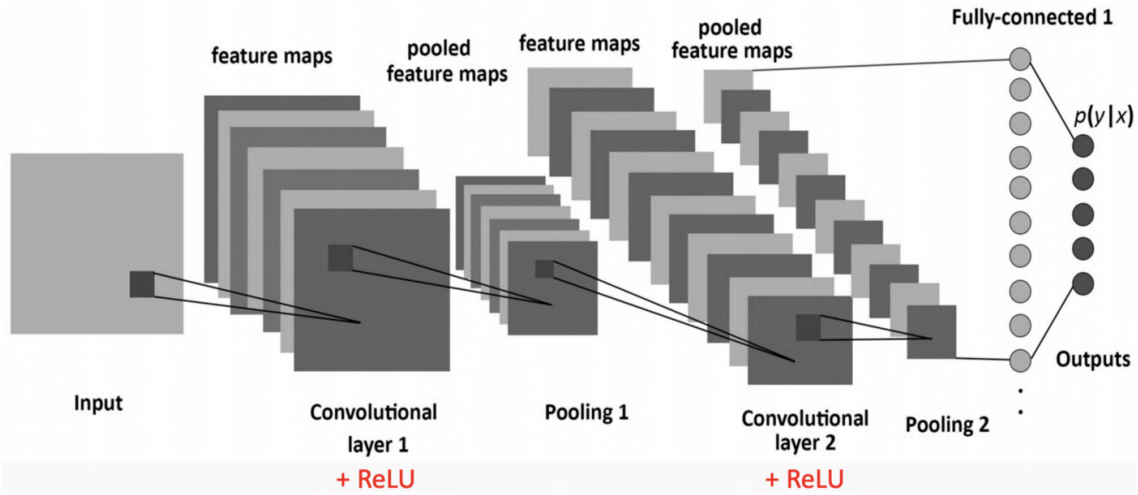
Keďže každý neurón má niekoľko prepojení počet váh začne rapídne narastať pre väčšie obrázky (pozri 1.2.6)(napr. 255x255 pixlov s 3 farbami si vyžaduje više 195 000 váh na trénovanie)

Ďalšou nevýhodou v spracovaní obrazu je, natrénované siete niesu pozične invariantné k objektom na obrázku. To je problém, keďže sieťe natrénované napríklad na dátach obsahujúcich stoličku v dolnej časti obrazu ju nedokážu v praxi rozoznať ak je napríklad v hornej časti. Ďalší problém je dedičný z povahy architektúry MLP. Keďže sú 2D dáta z obrázka sploštené na jednu dimenziu stráca sa informácia o polohe.

Ako odpoveď na tieto problémy vznikli Konvolučné neurónové site.

2 Konvolučné neurónové siete

Vo všeobecnosti sa môžme pozerať na Konvolučné neurónové siete (CNN - Convolutional Neural Network) ako na umelú neurónovú sieť, ktorá obsahuje nejaký typ špecializácie pre detekciu vlastností a priradení významu týmto vlastnostiam. Pri tejto detekcii vlastností sa využíva fakt, že pozícia pixelov na obrázku vo vsťahu k okolitým pixlom má sémantický význam[10]. Preto sa pixel pri detekcii nevystupuje samostatne ale berie sa do úvahy aj jeho okolie. Táto detekcia vlastností je jedna z vecí, čo robí CNNs tak užitočné pri analýze obrazu. Ďalšou vlastnosťou je, že tieto vlastnosti dokáže detektovať priestorovo invariantne, čiže nezávisle na ich polohe vrámci obrázka. Za tieto vlastnosť vďačí skrytým konvolučným vrstvám. Konvolučné vrstvy sú prvkom ktorým sa Konvolučné neurónové siete odlišujú od štandardného viacvrstvového perceptronu (1.3.1). Taktiež sa odlišujú tým, že nemajú plne prepojené vrstvy. To znižuje veľkosť a výpočetnú zložitosť siete na trénovanie a používanie (viď. 1.2.6). Každá sieť, ktorá obsahuje aspoň jednú konvolučnú vrstvu sa považuje za konvolučnú neurónovú sieť. Vďaka týmto vlastnostiam sú tieto siete veľmi úspešné oblasti klasifikácie obrazu. Aj keď analýza obrazu je najrozšírenejšie využite CNN, tento typ siete sa môže použiť tiež na iné analýzy dát a klasifikačné problémy. Napríklad spracovania zvuku, ale aj v zdravotníctve na predikovanie zdravia či pri modelovaní klimatu a jeho zmien alebo vo finančnom sektore ako prevencie pred finančnými podvodmi[12, 13].



Obr. 2.1: Príklad architektúry CNN s dvomi konvolučnými vrstvami, dvomi pooling vrstvami a jednou plne prepojenou vrstvou[11]

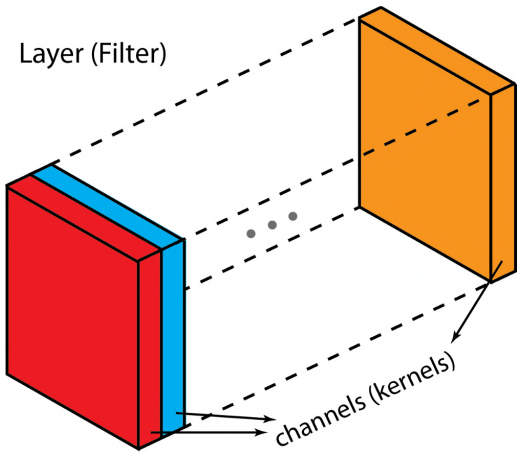
2.1 Architektúra konvolučných neurónových sietí

Ako jedná z prvých konvolučných neurónových sietí bola v roku 1998 predstavená sieť LetNet. Pôvodne sa využívala na rozpoznávanie znakov a textu. Za uplynulé roky boli síce navrhnuté viaceré nové architektúry ale všetky využívajú rovnaké hlavné princípy ako boli použité v tejto sieti. Príklad jednoduchej konvolučnej neurónovej siete je možné vidieť na nasledujúcom obrázku (obr. 2.1). Architektúra konvolučných neurónových sietí, ako už bolo čiastočne spomenuté, sa skladá zo štyroch hlavných operácií:

1. Konvolúcie
2. Nelineárnej aktivačnej funkcie (ReLU)
3. Združovania ("Pooling")
4. Klasifikácie (Plne prepojená vrstva)

2.1.1 Konvolučné filtre

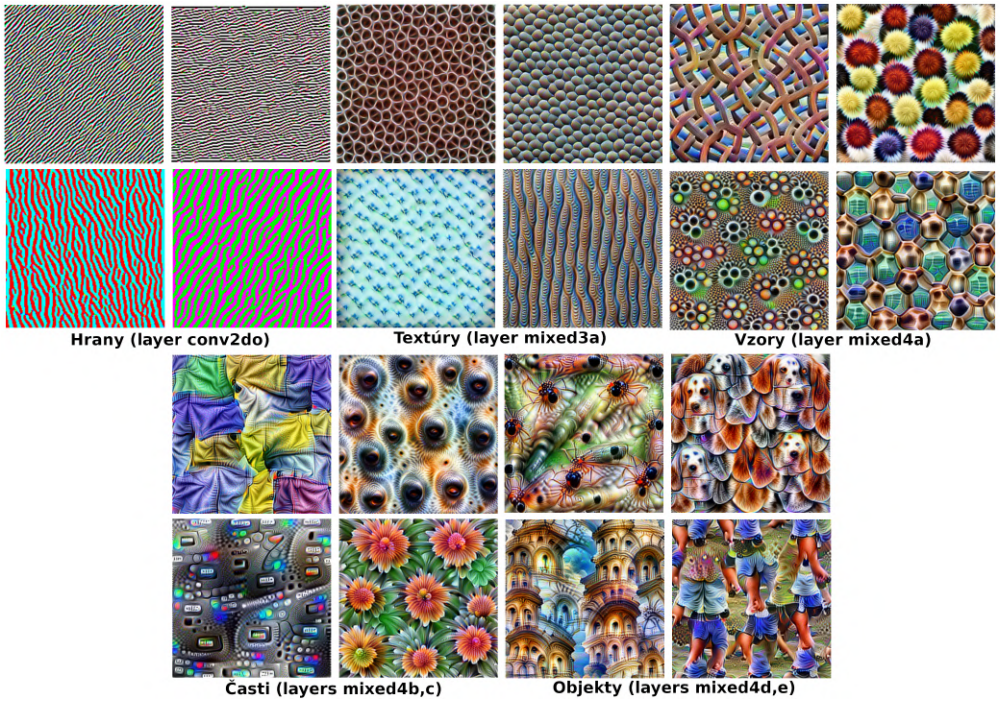
Pred vysvetlením samotnej operácie Konvolúcie je potrebné pochopiť jej prvky. Konvolučné filtre sa skladajú z jadier (kernelov) rôznej veľkosti naskladaných na sebe. Jadrá sú v zásade 2D matice ktoré obsahujú v sebe váhy. Z toho vyplýva, že konvolučné filtre majú tvar 3D matice rôznych rozmerov podľa rozmeru ich kernelov (obr. 2.2). Vrámci jedného filtru majú kernely rovnakú veľkosť. Ich veľkosť sa určuje pri návrhu siete ešte pred jej trénovaním, narozdiel od ich váh, ktoré sú náhodne inicializované a počas tréningu získajú užitočné hodnoty. Najčastejšie sa používajú kernely tvaru $3x3$ alebo $5x5$. Rozmery Konvolučných filtrov teda budú v rovnaké ako rozmery ich kernelov len o 1 dimenziu väčšie. Veľkosť pridanej dimenzie závisí od počtu kanálov vstupných dát. Farebné obrázky majú väčšinou 3 kanály, čiernobiele lne 1 (pre čiernobiele obr. je filter 2D).



Obr. 2.2: Rozdiel medzi konvolučným filtrom a kernelom

2.1.2 Konvolúcia

Ako už bolo spomenuté, Konvolúcia extrahuje vzory al. vlastnosti zo vstupu a zároveň zachováva relatívnu priestorovú informáciu. To je veľmi užitočné hlavne pri úlohách spojených s počítačovým videním, pretože tieto úlohy často zahŕňajú identifikáciu objektov kde je nutné vziať do úvahy určité priestorové vzťahy (napr. torzo človeka je spojené s hlavou, rukami a nohami). Tieto informácie sa v CNN detekujú pomocou konvolučných filtrov (2.1.1. Konvolúcia sa vykonáva v Konvolučných vrstvách neurónovej siete. Týchto vrstiev je v každej sieti niekoľko. Ako aj vo viacvrstvovej neurónovej sieti aj v CNN sa výstup z jednej konvolučnej vrstvy dostane (po transformácii v pooling vrstve 2.1.4) na vstup nasledujúcej vrstvy. V každej vrstve sa vykonáva extrahovanie vlastností. Čím sa filtre nachádzajú v hlbšej vrstve, tým sú citlivé na abstraktnejšie a komplexnejšie vlastnosti (obr. 2.3). V prvej konvolučnej vrstve s môžu detektovať vlastnosti nízkej zložitosti ako horizontálne, zvislé čiary alebo oblúky. V hlbších vrstvách sa na základe už extrahovaných jednoduchých vlastností detekujú komplexnejsie (napr. roh, kruh alebo štvorec). Tu vidíme dôležitosť zachovania lokality dát. Napríklad detekcia rohu by nebola možná bez toho aby sa zachovala relatívna pozícia dvoch navzájom kolmých hrán. Nakoniec je sieť schopná rozpoznať oko alebo tvár či celého človeka.



Obr. 2.3: Vizualizácia filtrov (GoogLeNet[15], trénovaný na ImageNet[16] datasete). Demonštruje postupné komplexnejšie chápanie obrazových vstupov neurónovej siete[14].

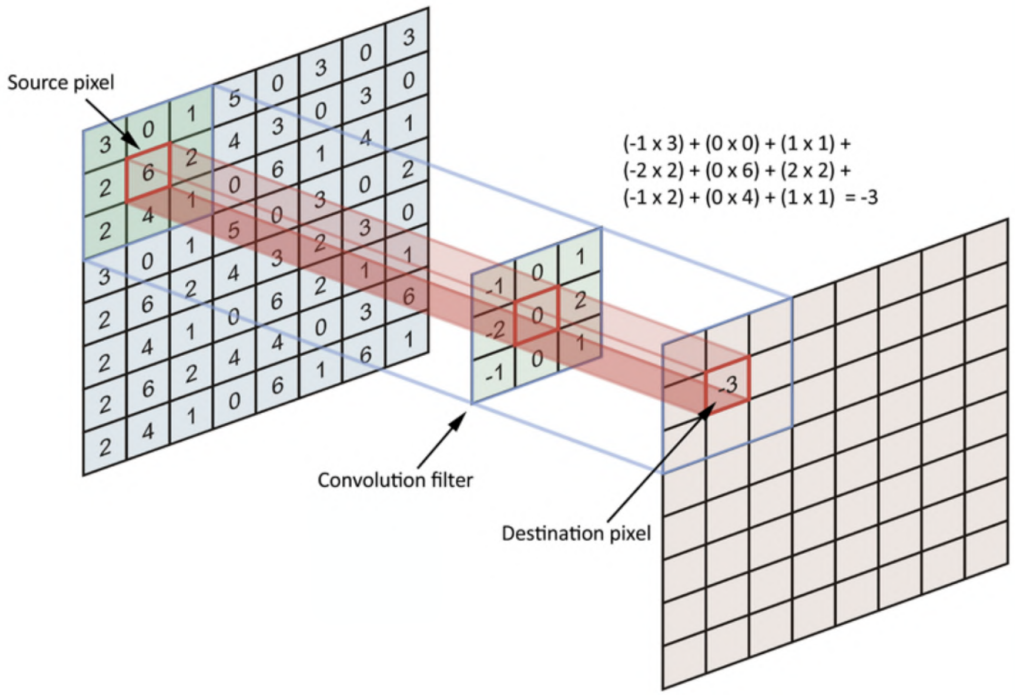
Proces konvolúcie

Pri vyhodnotení konvolučnej vrstvy sa aplikujú postupne filtre na všetky vstupné dáta. Aplikácia spočíva vo vypočítaní tzv. 2D diskrétnej konvolučnej operácie, ktorá spočíva vo vynásobení časti matice vstupu s maticou filtru a následne v sčítaní prvkov výslednej matice do jedného čísla (obr. 2.4)[17]. Táto operácia sa vykoná pre každý prvok vstupných dát. Teda kernel filtru sa posúva po vstupných dátach a vykonáva operáciu nad aktuálnymi dátami. Tým Konvolučná vrstva konvertuje vstupnú 2D maticu vlastností na inú 2D maticu komplexnejších vlastností.

$$C(m,n) = (I \times K)(m,n) = \sum_j \sum_k K(j,k) I(m-j, n-k) \tag{2.1}$$

Kde:

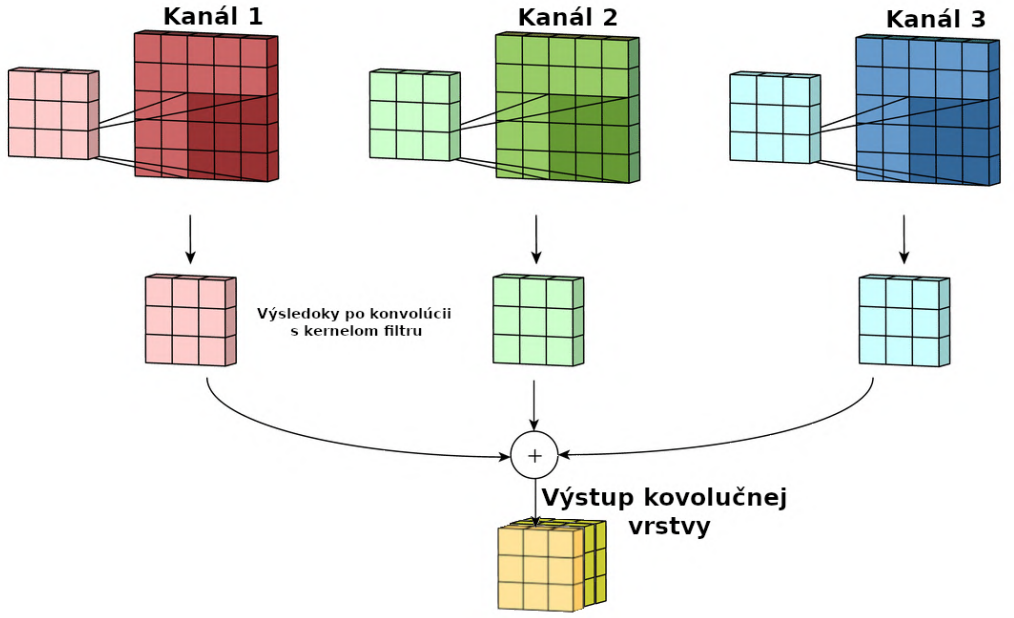
C(m,n) = je operácia 2D konvolúcie na obraze
K = je kernel s ktorým sa vykonáva konvolúcia
I = je kanál nad ktorý sa vykonáva konvolúcia



Obr. 2.4: 2D konvolučná operácia na jednom kanály vstupu (popísaná rovnicou (2.1)). Z ľava do prava: Zdrojový pixel, Konvolučný filter, Cieľový pixel[11].

Tento postup sa vykoná pre každý kanál vstupných dát. Čo má za následok 3D maticu na výstupe pre jeden filter. Potom sa dáta pre jednotlivé kanály sčítajú jednoduchým maticovým sčítaním teda vznikne jedna 2D matica pre jeden filter.

To sa opakuje pre každý z filtrov pre danú Konvolučnú vrstvu a teda na výstupe dostaneme 3D maticu s hĺbkou rovnajúcou sa počtu filtrov (čo sa môže drasticky meniť od vrstvy k vrstve napr. $32x32x16$ -> $32x32x128$ pri vrstve zo 128 filtrami).[11, 18]



Obr. 2.5: Znázornenie procesu konvolúcie. Konvolúcia pre každý kanál, sčítanie a znázornenie výstupu tzv. mapy vlastností (angl. feauture map) pre viacej filtrov (3 filtre).[18]

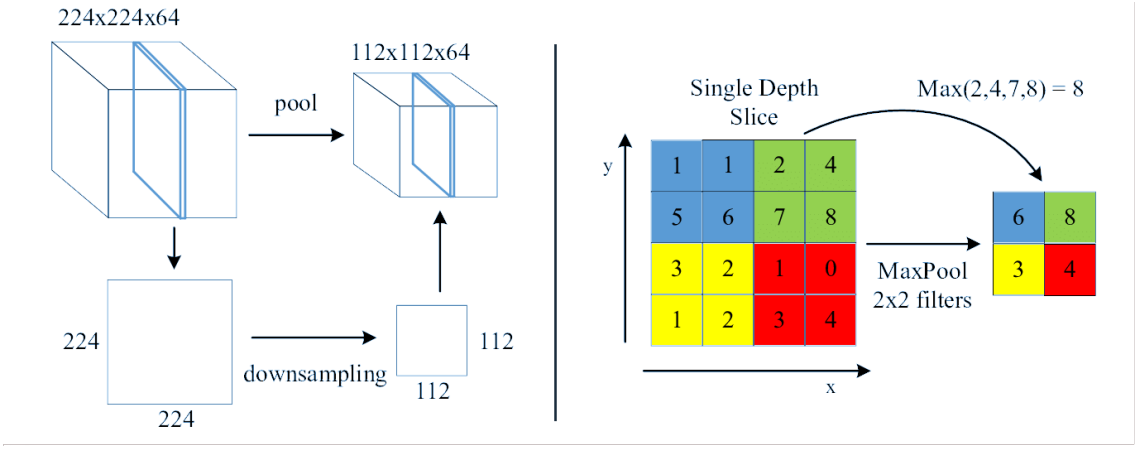
Výstup konvolučnej vrstvy sa niekedy volá mapa vlastností (angl. feauture map) a jeho veľkosť ovplyvňujú tri parametre.

1. Hĺbka konvolučnej vrstvy. Tá je určená počtom aplikovaných filtrov na danej vrstve. Ako už bolo spomenuté na konci odseku o konvolúcii (2.1.2) tento parameter sa môže medzi vrstvami meniť veľmi výrazne.
2. Krok konvolúcie. Je počet prvkov vstupných dát (napr. pixely) o ktoré sa filter medzi krokmi posunie. Ak sa nastaviť krok Konvolučného filtra na hodnotu vyššiu ako 1 filter už neprejde každý vstup, ale niektoré prvky pri každom kroku vynechá. Nutne však nepríde o informáciu o preskočených dátach, lebo väčšinou filtre obsiahnú aj okolie dát. Napríklad ak je filter rozmerov $3x3$ tak obsahuje informácie aj o dátach ktoré sú v okolí ± 1, čiže ak sa zvolí krok s hodnotou 1 vo výsledku bude stále v nejakej forme reprezentovaná informácia o každom bode zo vstupných dát.
3. Výplň nulami (Zero-padding). Ako môžme pozorovať na obrázku 2.5 výstupná matica má menšie prvé dve dimenzie (výšku a šírku) ako mala vstupná. Je to dôsledok toho, že vždy bol celý filter prekrytý vstupnými dátami a teda na

hraničné časti dát nebol úplne aplikovaný. Ak je to potreba spraviť konvolúciu nad každým pixlelom pôže sa využiť techinka tzv *zero padding*. Vtedy začína filter zo stredom na prvej bunke dát a chýbajúce bunky sa doplnia 0. Výsledok bude rovnakej veľkosti ako vstup.[19]

2.1.3 Aktivačmá funkcia

V súčastnosti najčastejšie používanou aktivačnou funkciou v Konvolučných neurónových sieťach je ReLU funkcia (pozri 1.2.4) a jej varianty. Táto funkcia sa aplikuje na každý prvok vstupných dát (ako je pixel) a nahradzuje negatívne hodnoty v nájdených vlastnostiach nulami. Keďže Konvolúcia je lineárna operácia (zložená z násobenia a sčítavania elementov matice) hlavný dôvod tejto funkcie je zaviesť do siete nelinearitu. Väčšina dát z reálneho sveta, ktoré bude sieť klasifikovať, sú nelineárneho charakteru. Na miesto GeLU funkcie sa môže použiť aj iná nelineárna funksa ako *tanh* alebo *sigmoid*[1].



Obr. 2.6: Združovacia operácia max pre okolie $2x2$.[21]

2.1.4 Združovací krok (angl. pooling)

Operácia priestorového združovania (angl. Spatial Pooling) redukuje dimenzionalitu mapy vlastnostný (výstup Konvolučnej vrstvy). Pri tejto operácií sa strácajú menej relevantné informácie niektoré informácie a ostanú zachované len tie relevantné. Je viacej druhov tejto operácie a líšia sa kritériom vo výbere relevantných údajov (Max, Priemer, Sčítanie, ...). Postup pri Združovaní je taký, že sa najprv na vstupných dátach zadefinuje nejaké malé okolie (napr. $2x2$). Potom sa pomocou zrušovacieho kritéria vyberie dominantný prvok v danej oblasti. V praxi sa ukázalo že max funkcia

[1]Táto práca sa nimi bližšie neazaoberá

dosahuje najlepších výsledkov. V tom prípade sa pre každú oblasť vyberie vlastnosť s najsilnejšou reprezentáciou. [20]

2.1.5 Plne prepojená vrstva

Používa sa spravidla na konci Konvolučných neurónových sieti ako posledná vrstva. V tomto bode má k dispozícii na vstupe identifikované vysokoúrovňové komplexné vlastnosti. Úlohou tejto časti siete je klasifikovať vstupný obraz na základe týchto vlastností do natrénovaných tried. Podľa architekrúri je táto vrstva tradičným Viacvrstvovým Perceptronom, ktorý bol už spomenutý v kapitole 1.3.1. Aj keď väčšinou vlastnosti získané z konvolučných a pooling vrstiev sú postačujúce pre klasifikačné úlohy a teoreticky by sa ich výstup mohol pretransformovať priamo do 1D podoby a spojiť s výstupnou vrstvou. Pridanie plne prepojenej vrstvy dodáva schopnosť odhaliť nelineárne závislosti medzi vstupnými vlastnosťami. Môžme sa na to pozrieť tak, že Konvolučné a Pooling vrstvy vytvárajú na vstupe priestor relevantných a nízko dimenzionálnych vlastností a Plne prepojená vrstva sa učí tento priestor interpretovať potenciálne nelineárnou funkciou. Poslednú vrstva je výstupná, tu sa typicky využíva prechodová funkcia SoftMax (pozri 1.2.4), ktorá normalizuje detekované pravdepodobnosti. [22] Z pravidla s delí na 3 časti:

1. *vstupnú vrstvu* - prvá vrstva za konvolučnou, transformuje výstup generovaný konvolučnou vrstvou do 1D vektoru (ktorý je použitý ako vstup ďalšej vrstvy)
2. *plne prepojené vrstvy* - na základe vstupu určujú kategóriu objektu
3. *výstupná vrstva* - normalizuje výstup do pravdepodobností

2.2 MAC

Multiplier-Accumulator (MAC) Unit sa často využíva v aplikáciach spracovanie signálov. Je to valstne operácia ktorá vynásobí 2 operandy medzi sebou a výsledok pričíta ku tretiemu operátoru. Táto operácia je z hw stránky 3 stupňová. A jej najpomalšia časť je čítanie čísel z pamäte. Ako bolo ukázané pri výpočte Konvolúcie (pozri 2.4) a pri výpočte základnej funkcie neurónu (pozri 1.1) táto operácia sa nachádza vo všetkých vrstvách spracovania neurónových sieti. A ako bolo ďalej spomenuté v zhrnutí (1.2.6) zo zložitosťou siete a s počtom obsiahnutých neurónov rastie aj počet MAC operácii pri každom vyhodnotení siete. Preto je táto jednotka vhodná pre určovanie a porovnávanie obtiažnosti neurónových sieti.

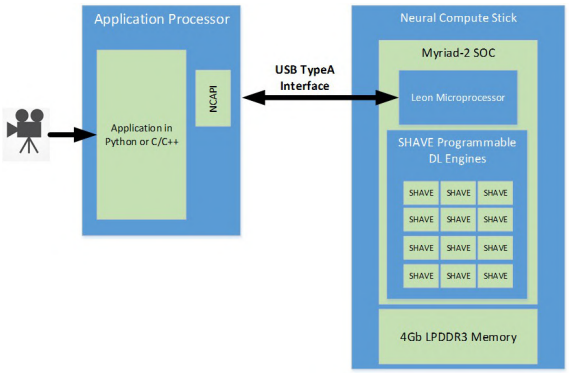
3 Akcelerovanie neurónových sietí

Na problém spracovania obrazu, videa či zvuku sa CNN stali štandardom a dosahujú v týchto oblastiach podobné výsledky ako ľudia. Takýto level presnosti však znamená čoraz zložitejšie CNN, čo má za následok veľké výpočtové nároky nielen na trénovanie ale aj na vyhodnocovanie už natrénovaných modelov tj. inferenciu. Z toho dôvodu je čoraz viacej potrebný dedikovaný hardvér na efektívne vykonávanie inferencie. Často využívanou platformou na implementovanie CNN aplikácii sú GPU, keďže majú veľmi dobrý výkon v zmysle čistej výpočtovej sily, niekedy až 11 TFLOP/s. [34] Avšak druhou stranou hrubého výkonu je vždy efektívne využite energie a v tejto kategórii sú efektívnejšie FPGA akcelerátory. Ich najväčšou výhodouje možnosť vysokej optimalizácia pre jednu konkrétnu neurónovú sieť. Vývoj a implementácia CNN riešení na poli mobilných, embeded alebo IoT zariadení robí tento aspekt čím ďalej tým dôležitejším. Navyše veľa aplikácii ktoré fungovali veľmi dobre na klaude, ako napríklad rozpoznávanie objektov, textu, tvári má tendenciu posúvať sa do hraničných, prenosných zariadení. Čo prirodzene limituje dostupné prostriedky z hľadiska pamäte ale aj množstva dostupnej energie pre výpočet. Tieto obmedzenia majú za následok zhoršenie presnosti alebo latencie. Navyše s rozšírením podpory framewokov pre hlboké učenie ako sú TensoFlow (TF Lite) alebo PyTorch a ich adopciou vývojármi vznikol dopyt po výkonných a relatívne jednoduchých na použite akcelerátorov, ktoré by ponúkli širokú podporu pre rôzne CNN . Z tohoto dôvodu sa začali objavovať akelerátori ako Neural Compute Stick (NCS)[35], NeuralCompute Stick 2 (NCS2)[35] a Google Coral[36], či Nvidia Jetson Nano[37].

Z vyššie uvedených akcelerátorov bol ďalej vybraný a použitý Intel Neural Compute Stick (NCS) najmä kvôli možnostiach knižnice OpenVino a spúšťaniu inferencie na rozličných Intel platformách.

3.1 Intel Movidius NCS

Platforma poháňaná Intel® Movidius™ Myriad™ 2 vision processing unit (VPU). Tento typ procesora patrí do kategórie, ktorá má za cieľ poskytnúť relatívne vysoký výpočetný výkon pri ultra melej spotrebe (menej ako 1W [32]). Intel VPU čip poskytuje 4Gbit pamäte LPDDR3 DRAM. Je to typ RAM určený pre mobilné zariadenia navrhnutý s ohľadom na energetickú efektívnosť, o čom svedčí aj znížené napájacie napätie na 1.8V (oproti 2.5V pre DDR3) a to pri rýchlosti 1,866 Mbps [38]. Ďalej má v sebe zabudovaných 12 SHAVE processorov (Obr3.1). SHAVE processor je mikro architektúra akcelerátoru navrhnutá pre VPU. Poskytuje výpočetný výkon 20 GFLOPS pri frekvencii 180MHz a spotrebe 300mW. Shave procesroy sú typ VLLIW procesorov (variable-length very long instruction word). Tento typ procesoru



Obr. 3.1: Architektúra akcelerátoru Intel NCS

je špeciálne optimalizovaný na efektívne vykonávanie operácii nad relatívne veľkými množinami hodnôt naraz. Medzi perifériami, ktoré podporuje aj podpora pre dve 12Mpx kamey pri vysokých prenosových rýchlostiach[39]. Vďaka týmto optimalizáciám sú tieto procesory využívané na akcelerovanie častí neurónových sietí tým, že sa spúšťajú na čipoch paralelne. US interface medzi akcelerátorom a platformou je verzie 3.0, čiže podporuje rýchlosti až po 5Gbps. VPU tiež využíva mikrokontrolér typu SPARC (redukovaná sada RISC), ktorý ovláda akcelerátor pomocou propriterného firmwaru. Ten sa pri prvom spojení akcelerátoru s počítačom stiahne pomocou Neural Compute SDK (NCSDK) knižnice (momentálne sa namiesto NCSDK využíva knižnica OpenVino pozri 3.3.1). Komunikáciu zo zariadením sprostredkováva LEON micro-processor. Okrem toho spracúva graf pre inferenciu a plánuje kernely (filtre pre jednotlivé kanaly vstupných dát pozri 2.1.1) na spracovanie pre SHAVE neural compute accelerator engine. Navyše monitoruje teplotu zaradenia a prispôsobuje výkon podľa teploty. Výstupný výsledok neurónovej siete sa prirodzene posiela naspäť tiež cez USB rozhranie a ďalej sa spracúva pomocou NCAPI. [40]

3.2 Ostatné HW zariadenia

Krátke predstavenie podstavných parametrov ďalších platforiem využívaných v tejto práci.

3.2.1 Platforma raspberry pi

Platfoma raspberry pi, od svojho debutu v roku 2012 sa stala obrovským úspechom po celom svete. Raspbery Pi Foundation je zaragistovaná ako vzdelávacia charitatívna organizácia. Ich misia (podľa raspberrypi.org) je pomôct doručiť výkon výpočtovej techniky a digitálnej tvorby do rúk ľudí na celom svete. Po distribuovaní

vyše 30 miliónov kusov počítačov sa stala štandardom v oblasti prototypovania a vývoja rozličných aplikácii.

Raspberry Pi3 model B

- 4-jadrový 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 bezdrôtový LAN
- 4 USB2 porty
- CSI kamera port pre pripojenie Raspberry Pi kamery
- MicroSD port pre načítanie OS a ukladanie údajov
- 3.7W

Raspberry Pi4

- Broadcom BCM2711, 4-jadrový Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 4GB LPDDR4-3200 SDRAM
- 2.4 GHz a 5.0 GHz IEEE 802.11ac bezdrôtový, Bluetooth 5.0, BLE
- 2 USB 3.0 porty a 2 USB 2.0 porty
- 2 Micro-HDMI porty (podporovateľné až do 4kp60)
- 2-lane(dvojradové) MIPI CSI kamra port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- MicroSD port pre načítanie OS a ukladanie údajov
- 5V DC cez USB-C connector (minimum 3A*)
- 5.1W

Raspberry Pi - Kamerový modul v1.3

- 5 Megapixlový senzor
- video módy: 1080p30, 720p60 a 640 × 480p60/90

3.2.2 Intel i7 8550u (asus ux 430u)

- 4-jadrový (8 threads) procesor s rýchlosťou od 1.8GHz do 4.0GHz
- 15-wattovej TDP
- Integrovaná grafika procesoru je Intel® UHD Graphics 620 (Intel Gen 9.5) s
- frekvencia od 300MHz až do 1.15GHz.

3.2.3 Nvidia Gforce mx150

- 1,468 MHz základná rýchlosť

- 2GB video pamäť (GDDR5)

3.3 SW používaný pri implementácii

3.3.1 OpenVino

OpenVINO™ toolkit je kompletná sada nástrojov, pre vývoj a nasadenie aplikácii zameraných na rozpoznávanie obrazu pre Intel platformy. Je to komplexný nástroj poskytujúci podporu od trénovania až po jej nasadenie a akcelerovanie. Openvino obsahuje command-line nástroj pre konvertovanie a importovanie modelov s názvom Deep Learning Model Optimizer

Deep Learning Model Optimizer

Tento cross-platformový nástoj slúži pre uľahčenie prechodu medzi fázou učenia a fázou nasadenia modelu. Konvertuje siete natrénované na podporovaných platformách (Caffe*, TensorFlow*, MXNet*, Kaldi*, ONNX*) do valstného tzv. Intermediate Representation (IR) formátu, podporovaného Inference Engin API3.3.2. Ku každej platforme je potrebné overenie podporovaných topológii siete. Pri tejto konverzií OpenVino vykonáva statickú analýzu modelu a upraví model aby bol optimálny pre Intel prostredie. Každý model, pokiaľ už nieje v IR formáte sa musí pred spustením takto konvertovať. Výsledkom sú 2 súbory, jeden súbor popisuje topológiu siete (.xml), druhý obsahuje váhy (.bin).

3.3.2 Inference Engine Python API

Je to api pre komunikáciu s hardvérom na spustenie siete. Poskytuje sadu funkcii a tried pre zjednodušenie procesu inferencie. Bola využívaná pri implementácii Demonštračnej aplikácie a ostatných experimentov. IE (Inference Engine) podporuje rôzne typy zariadení. V tejto práci boli využívaná: GPU plugin, CPU plugin a VPU plugin. Okrem nich je podporovaný ešte plugin pre FPGA, plugin na spustenie inferencie na viacerých zariadeniach naraz a pre inteligentá vyberanie najvhodnejšieho zariadenia (Heterogenous plugin) Každý plugin podporuje optimalizačné algoritmy, ktoré pokiaľ je to možné spájajú viacero vrstiev do jednej (napr. spojenie Konvolučnej a jednoduchej vrstvy, alebo plne prepojenej a aktivačnej s ReLu a iné zložitejšie). Dokonca umožňuje pred-výpočet niektorých vrstiev pri nahrávaní do zariadenia.

3.3.3 OpenCV

Spolu s ostatnými nástrojmi bola nainštalovaná knižnica OpenCV. Je to open source knižnica pre strojové učenie a spracovanie obrazu. Obsahuje sadu vysoko optimalizovaných funkcii prerácu s neurónovými sieťami, pre prácu s obrazom a jeho transformáciu. Podporuje Python interface. V balíčku bola optimalizovaná verzia pre zariadenia platformy Intel.

3.3.4 TensorFlow Lite

Je to podobne ako OpenVino sada nástrojov na spúšťanie modelov neurónových sietí pre mobilné, vstavané a IoT zariadenia. Podobne ako OpenVino má svoj nástoj na optimalizáciu a prevod siete do formátu podporovaného TF Lite. Tento prevod je však možný len z formátu pre TensorFlow, čiže iné formáty sa nedajú priamo prevádzať. Ďalej obsahuje TF interpréter, ktorý vykonáva TF Lite modely. Čiže všetky inferencie sa vykonávajú pomocou tohto modelu.

4 Implementácia využitia akcelerátoru v reálnej aplikácii

Táto kapitola sa venuje návrhu a spôsobu implementácie demonštračnej aplikácie využitia akcelerátoru Intel Movidius NCS (3.1). Problematika akcelerácie hlbokých neurónových sieti pomocou dedikovaného hardvérového akcelerátoru je úloha, ktorú je najlepšie v praxi demonštrovať pri rozpoznávaní obrazu. Keďže rozpoznávanie obrazu je problém dnes skoro výhradne riešený Hlbokými neurónovými sieťami, konkrétne Konvolučnými neurónovými sieťami (kapitola 2). V reálnych aplikáciach je často využívaná k riešeniu nie jedna ale niekoľko odlišných neurónových sieti ktoré tvoria dátovú pipelinu[9]. Týmto spôsobom každá sieť spracúva časť komplexného problému. Demonštračná aplikácia rieši problém identifikácie identity človeka v obraze. Je to komplexný problém, ktorého riešenie si tiež vyžaduje zapojenie viacerých neurónových sieti. Preto veľmi dobre demonštruje spôsob použitia akcelerátora v reálnych podmienkach. Zároveň preverí výkon a možnosti akcelerátora, poprípade limity jeho využitia.

4.1 Návrh aplikácie

Problém re-identifikácie tváre sa dá rozdeliť do troch pod-problémov. Identifikácia tváre v obraze, identifikácia významných bodov na tvary a zistenie správnej identity. Ako obrazový vstup vstup je možné zadať web kameru alebo video. Aplikácia podporuje viaceré druhy sieti na identifikáciu tváre v obraze. Pre ostatné pod problémy je implementovaná podpora len pre jednu sieť. Sú využívané už natrénované konvolučné siete z model zoo v Openvino prostredia. Siete boli sťahované z oficiálneho git depozitáru. Aplikácia bola implementovaná v Pythone3 s využitím OmenVino prostredia. Najväčší problém pri implementácii bolo správne nastavenie vývojového prostredia. Aj keď manuálové stránky boli pri samostatnej implementácii veľmi užitočné, proces inštalácie nebol až taký priamočiary ako bol popísaný na stránkach. Viackrát bolo treba nainštalovať iné verzie pre niektoré package a to napriek tomu, že proces bol opakovaný na čisto nainštalovanej podporovanej verzii Ubuntu 18.04 s virtuálnym prostredím aj bez. Nakoniec bolo prostredie rozbehnuté ako docker kontajner a aj aplikácia tak vyhynutá. Tento krok sa ukázal ako správny a umožnil totálnu kontrolu nad vývojovým prostredím.

4.1.1 Detekcia tváre

Pri detekcii tváre boli využívané siete face-detection-0100, face-detection-0102, face-detection-0104. Tieto site majú ako základ sieť MobolenetV2. Medzi sebou sa líšia ich zložitosťou a veľkosťou vstupného obrazu. Ten, vahľadom na uvedené poradie sietí, bol 256x256, 384,384, 448x448. Ďalej sa líšili zložitosťou a presnosťou (v rovnakom poradí): 86.66%, 0.786 GFlops; 91.61%, 1.767 GFlops; 92,74%, 2.406 GFlops. Obrazové transformácie boli implementované pomocou knižnice OpenCV. Zaujímavé hodnoty z výstupu boli počet detekovaných tvári a koordinácie ohraničujúcich štvorcov.

4.1.2 Rozpoznanie významných bodov tváre

Pre detekovanie bodov bola využitá ultra ľahká neurónová sieť s názvom landmarks-regression-retail-0009. Vstup bol malý výsek tváre 48x48. Náročnosť siete bola iba 0.021 GFlops a priemerná normovaná chyba (pre VGGFace2) 0.0705.

4.1.3 Identifikovanie známych tvárí

Pre re-identifikáciu tváre bola použitá sieť s názvom face-reidentification-retail-0095 tiež so základom v sieti MobileNetV2. Vstup pre sieť bol v tvare 128x128 a výstup bol vektor 256 floating point hodnôt, ktoré určovali tvár. Pre zapamätanie a re-identifikáciu bolo treba uložiť deskriptor tváre a potom vypočítať kosínusovú vzdialenosť dvoch vektorov.

5 Experimentálne vyhodnotenie výkonu akcelerátora

Experimenty boli navrhnuté tak, aby preverili vlastnosti hardvérových prvkov, a to z hľadiska izolovaného výkonu, výkonu v reálnej aplikácii a vhodnosti použitia s ohľadom na spotrebu. Následne boli zistené výsledky spracované pomocou štatistických metód, čo zaručí relevantnú interpretáciu zistení.

5.1 Matemeticko-štatistické postupy

Pri prvotnom štatistickom spracovaní získaných údajov, najmä pri diskrétnych meraniach, sa okrem určenia odhadu meranej veličiny, resp. závislosti, ktorá je cieľom merania, vyšetrí aproximácia súboru dát normálnym modelom. Mnohé postupy štatistického spracovania súboru nameraných hodnôt vychádzajú z predpokladu normálneho rozdelenia početnosti. Z teoretického hľadiska sú súbory s normálnym rozdelením najpodrobnejšie preskúmané a majú viaceré vlastnosti umožňujúce zjednodušenie mnohých výpočtov. Pre použitie analýzy variancie (ANOVA), ktorá by mohla byť vhodnou metódou pre vyhodnotenie získaných výsledkov, je potrebné presvedčiť sa, či je možne akceptovať predpoklad o normalite a homogenite analyzovaných súborov[23]. Na to sú vypracované viaceré metódy, ktoré spadajú do oblasti testovania hypotéz (testy dobrej zhody). Pri hodnotení výsledkov meraní sa ako kritériá normality najčastejšie používajú asymetria a exces rozdelenia, Kolmogorov - Smirnov test dobrej zhody [27] a iné testy. Najväčšiu silu zo všetkých testov normality má vo väčšine situácií Shapiro - Wilkov W test normality, ktorý som použil vo svojej práci [28]. Ak je štatistika W významná, zamietame nulovú hypotézu, ktorá tvrdí, že dáta pochádzajú z normálneho rozdelenia [24]. Pre homogenitu rozptylov sa najčastejšie používajú Hartleyov, Cochranov alebo Bartlettov testy [25, 26].

Na základe analýz a vykonaní testu normality sa potvrdilo, že rozdelenie výberu dát sa štatisticky významne odlišuje od normálneho rozdelenia. V prípade, že dáta nespĺňajú požadované kritéria môžu sa upraviť transformáciou. Keďže potrebnou logaritmickou transformáciou ani Box-Coxovou transformáciou sa nedosiahlo normálne rozdelenia dát v ďalšej analýze sa pokračovalo neparametrickou Kruskal-Wallisovou ANOVA, ktorá porovnáva rozdiely v troch a viac nezávislých súboroch. Je založená na poradí analyzovaných údajov a nepredpokladá sa, že dáta pochádzajú z normálneho rozdelenia. Základnou podmienkou testu je nezávislosť pozorovaných hodnôt, čo v našom prípade vieme z povahy pokusu. Kruskal-Wallisov test vychádza z mediánových hodnôt. Medián predstavuje prostrednú hodnotu usporiadaného štatistického súboru. Rovnaký počet hodnôt je nižších ako medián a rovnaký počet

hodnôt je vyšších ako medián. Teda nie je ovplyvnený extrémnymi hodnotami.

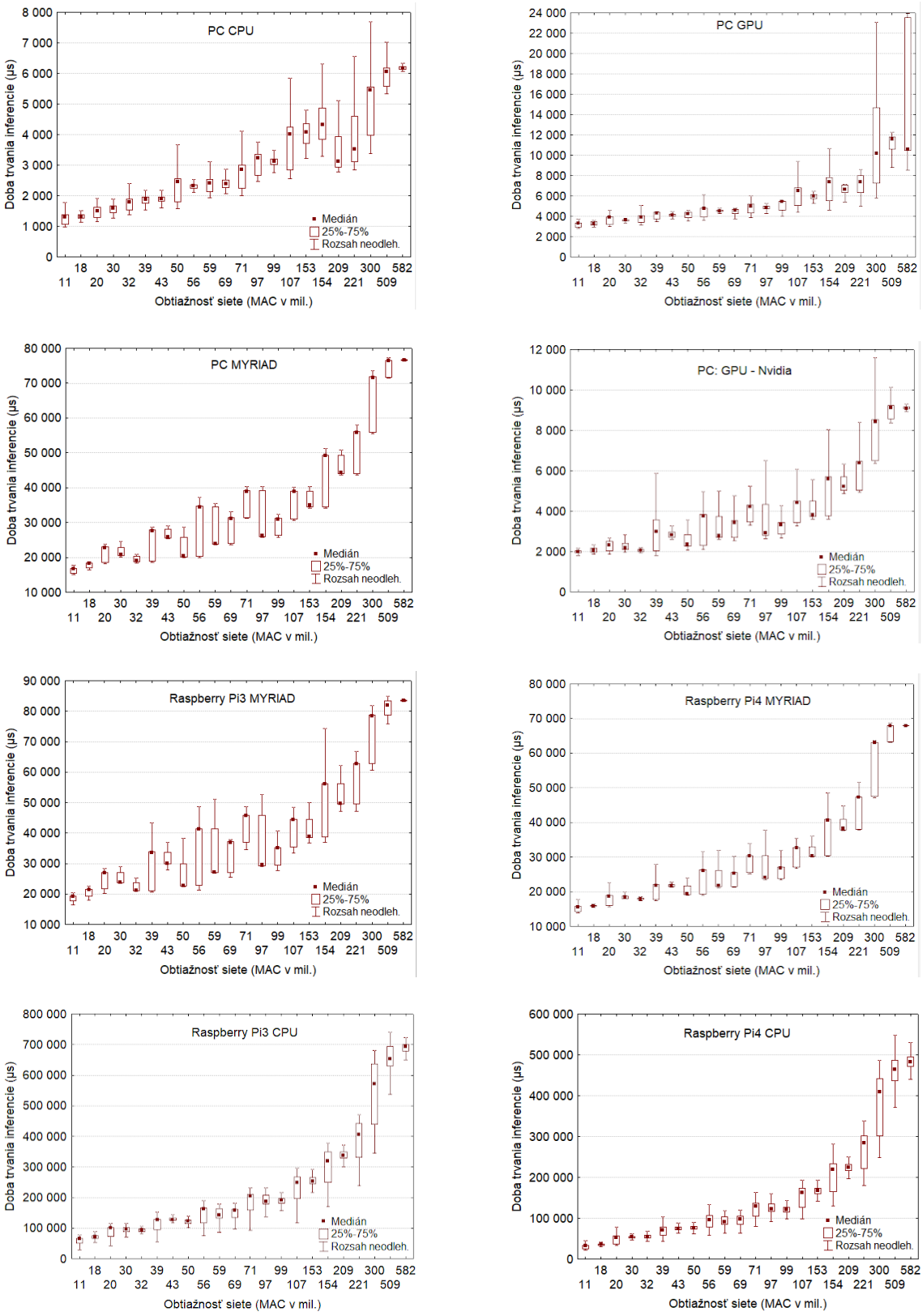
Všetky testovacie a štatisticko-matematické výpočty ako aj grafické vykreslenie bolo robené pomocou softwaru STATISTICA ver. 9.0.

5.2 Experiment 1: porovnanie rýchlosti akcelerátora s inými zariadeniami pri rôznych úrovniach zátaže

Cieľom tohto experimentu je porovnať výkon a efektívnosť akcelerátoru Movidius NCS s inými HW platformami. Ďalej ukázať mieru akcelerácie CNN na platforme Raspberry Pi a dať ju do kontextu s inými zariadeniami. Experiment sa spúšťal spolu pre týchto 8 zariadení: akcelerátor Intel Movidius NCS na PC platforme, Raspberry Pi 4, Raspberry Pi 3 a pre ARM Cortex-A72 Raspberry Pi3 a ARM Cortex A53 Raspberry Pi4 ďalej pre Intel® Core™ i7-8550U Processor, Intel® UHD Graphics 620, Nvidia Gforce mx150 na PC platforme. V experimente sa využívala sieť MobileNetV2[29]. Zariadenia sa testovali na 22 Imagenet Checkpointoch, každý korešpondujúci s inou konfiguráciou parametrov[29]. Čo malo za následok inú výpočetnú zložitosť siete vyjadrenú v MAC (viď. 2.2). To poskytlo širokú škálu podmienok pre testovanie. Pri analýze sa dbalo na to aby na testovanom systéme nebežali žiadné iné procesy čo by mohli ovplyvniť experiment. Napriek tomu sa táto možnosť nedala úplne eliminovať preto sa každý test opakoval 300 krát aby prípadné rušenie a náhodná záťaž štatisticky neovplyvnila výsledok. V experimente sa sledoval čas výpočtu jednotlivej inferencie pre rôzdne náročnosti siete. Tieto časy boli použité ako kritérium pre vyhodnotenie výsledkov. Ďalej sa porovnal a vyhodnotil experiment z hľadiska spotreby energie pri výpočtoch.

5.2.1 Výsledky a analýza experimentu 1A - časová náročnosť

Výsledky analýzy časového výkonu zariadení sú grafický zobrazené krabicovými grafmi (Obr. 5.1) pre medián, dolný a horný kvartil, a rozpätie neodľahlých dát. Medián je vhodný z toho dôvodu, že zodpovedá povahe Kruskal-Walisovho testu. Všetky vybrané spomenuté štatistiky veľmi dobre poukazujú na charakteristiku rozdelenia skúmaných súborov pre jednotlivé zariadenia a ich podskupiny. V prípade normálneho rozdelenia početností súboru, je medián zhodný s aritmetickým priemerom. Vo väčšine skúmaných súborov to tak nebolo. Podľa hodnôt y-ovej osi je vidieť veľkosť nameraných hodnôt a ich rozptyl. Najnižšie hodnoty vykazuje PC CPU (škálovanie do 8 000), nasleduje PC CPU Nvidia (do 12 000). Naopak najväčšie hodnoty s najväčšími rozptylmi boli na Raspberry Pi3 CPU (do 800 000) a Raspberry Pi4 CPU (do 600 000 µs).



Obr. 5.1: Výsledné časové údaje pre jednotlivé zariadenia spolu s mediánom, percentilovým a rozsahom neodľahlých dát

Testované zariadenia		Kruskal-Wallisov test
PC	CPU	K-W test: H (21, N= 6600) =6264,539 p =0,000
	GPU	K-W test: H (21, N= 6600) =5571,029 p =0,000
	MYRIAD	K-W test: H (21, N= 6600) =6573,404 p =0,000
	GPU - Nvidia	K-W test: H (21, N= 6600) =6331,237 p =0,000
Raspberry	Pi3: CPU	K-W test: H (21, N= 6600) =6452,957 p =0,000
	Pi3: MYRIAD	K-W test: H (21, N= 6600) =6564,217 p =0,000
	Pi4: CPU	K-W test: H (21, N= 6600) =6502,456 p =0,000
	Pi4: MYRIAD	K-W test: H (21, N= 6600) =6507,435 p =0,000

Obr. 5.3: Výsledky Kruskal-Walisovho testu pre analyzované zariadenia

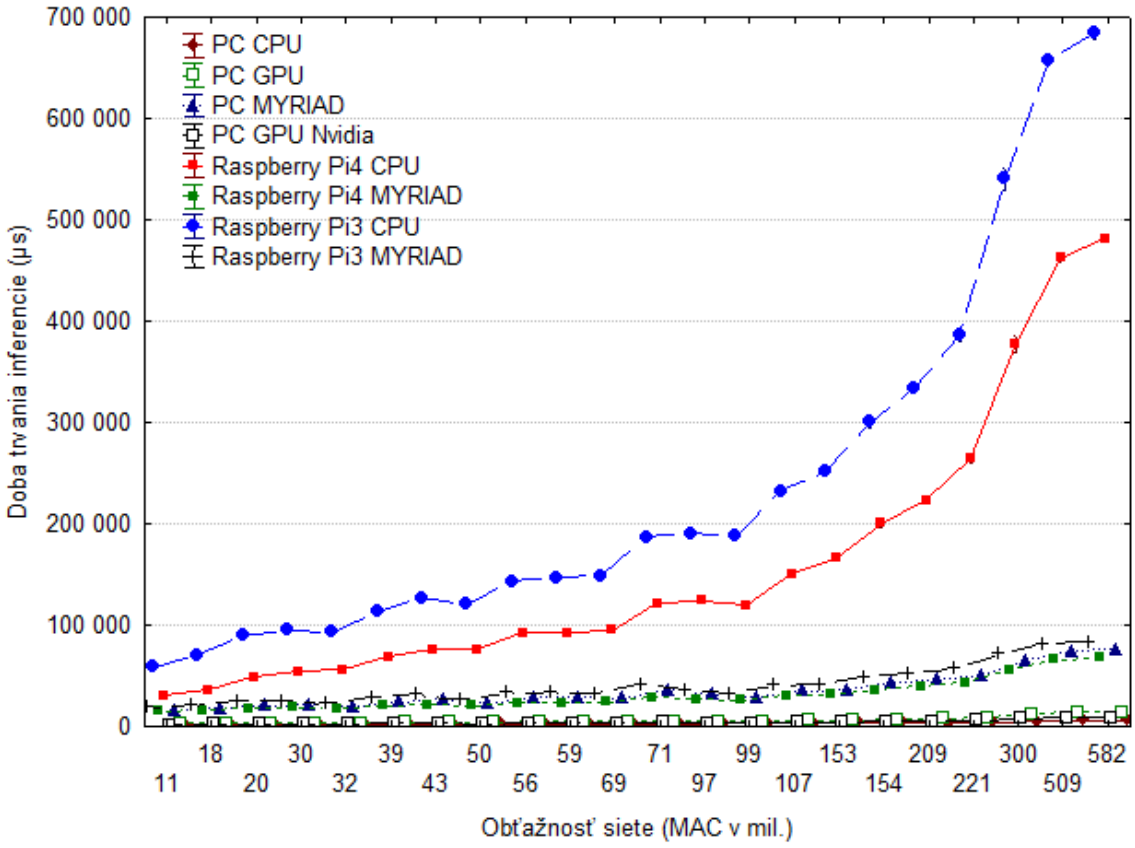
Analýza štatistických vlastností dát Z hodnôt Shapiro - Wilkovho W testu normality pre zariadenie PC CPU (Obr. 5.2 test normality pre Inferenciu (µs) kategorizovanú Obťažnosťou - MAC (mil.)) vidíme, že v každej skupine bol porušený predpoklad normality, aj keď hodnoty testu sú večšinou vysoké, blízke 1. P-hodnota vyšla veľmi nízka ($P < 0{,}05$). Podobné hodnoty Shapiro - Wilkovho W testu vyšli pre všetky testované zariadenia. Čo znamená, že výsledky neodpovedajú normálnemu rozloženiu, preto v ďalšom postupe sa namerané hodnoty podrobovali Kruskal-Walisovému testu vhodnému pre toto rozloženie. Na základe dosiahnutej hladiny významnosti ($P < 0{,}05$) sa môže povedať, že bol dokázaný štatisticky významný rozdiel v časovom vyhodnotení úloh pre rôzne zložitosti siete MobilenetV2 (Obr. 5.3) vo všetkých zariadeniach.

Obťažnosť siete (MAC v mil.)	Shapiro - Wilkov W test
11	SW-W = 0,8662; p = 0.0000
18	SW-W = 0,9207; p = 0.0000
20	SW-W = 0,912; p = 0.0000
30	SW-W = 0,7037; p = 0.0000
32	SW-W = 0,771; p = 0.0000
39	SW-W = 0,6651; p = 0.0000
43	SW-W = 0,8137; p = 0.0000
50	SW-W = 0,5141; p = 0.0000
56	SW-W = 0,8099; p = 0.0000
59	SW-W = 0,6467; p = 0.0000
69	SW-W = 0,7166; p = 0.0000
71	SW-W = 0,6751; p = 0.0000
97	SW-W = 0,6558; p = 0.0000
99	SW-W = 0,655; p = 0.0000
107	SW-W = 0,6507; p = 0.0000
153	SW-W = 0,8489; p = 0.0000
154	SW-W = 0,8133; p = 0.0000
209	SW-W = 0,9265; p = 0.0000
221	SW-W = 0,633; p = 0.0000
300	SW-W = 0,6863; p = 0.0000
509	SW-W = 0,4669; p = 0.0000
582	SW-W = 0,4573; p = 0.0000

Obr. 5.2: S-W pre PC CPU

Hodnoty súčtu poradia poukazujú na obtiažnosť spracovania úlohy pre jednotlivé skupiny a na rozdiely medzi nimi. Najrýchlejšie spracovanie bolo pri zložitosti siete 11 mil. MAC vo všetkých zariadeniach. Hodnota súčtu poradia je tu najnižšia. Pre PC MYRIAD je najnižšia zo všetkých zariadení (45150, 5.7). Potom pre všetky testované zariadenia okrem PC CPU nasledovala zložitosť 18 mil. MAC. Pre PC CPU mala sieť s 20 mil. MAC lepšie výsledky. Najdlhšie trvajúci výpočet pri všetkých zariadeniach bol zaznamenaný pri zložitosti siete 582 mil. MAC, potom nasledovala 509 mil. MAC. Tieto údaje

zatiaľ splnili predbežné očakávania. Aby sa posúdilo, či rozdiely časového trvania výpočtu pri rôznych zložitostiach siete sa od seba významne odlišujú, bola použitá metóda mnohonásobného porovnávania pre všetky skupiny (Obr. 4). Tým sa ukázalo ktoré výsledky sa zo štatistického hľadiska môžu považovať za rozdielne a ktoré nie. Ak P-hodnota je nižšia ako 0,05 (červené čísla) znamená to významný rozdiel medzi dvoma porovnávanými skupinami. Výsledky mnohonásobného porovnania pre ďalšie skúmané zariadenia (PC: CPU, PC: GPU – Nvidia, PC: GPU – Intel, PC: MYRIAD, Raspberry Pi4: MYRIAD, Raspberry Pi3: MYRIAD, Raspberry Pi4: CPU, Raspberry Pi3: CPU) vidíme v Tabuľkách A.1a – 5.8b (Príloha).

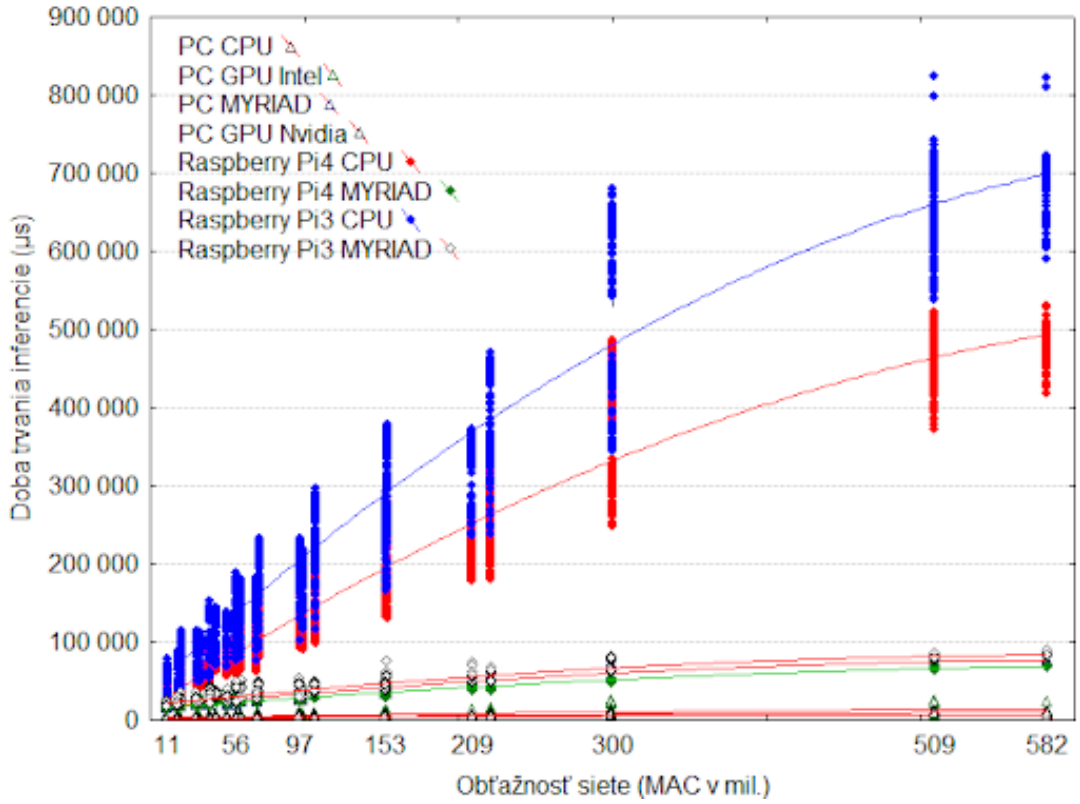


Obr. 5.4: Priebeh časových inferencií (µs) podľa obťažnosti v mil. MAC na jednotlivých skúmaných zariadeniach. (N=300, Priemer; Svorka: Priemer ±0,95 Interval spoľahlivosti).

Porovnanie času spracovania inferencie pri rôznych zložitostiach sietí medzi skúmanými zariadeniami

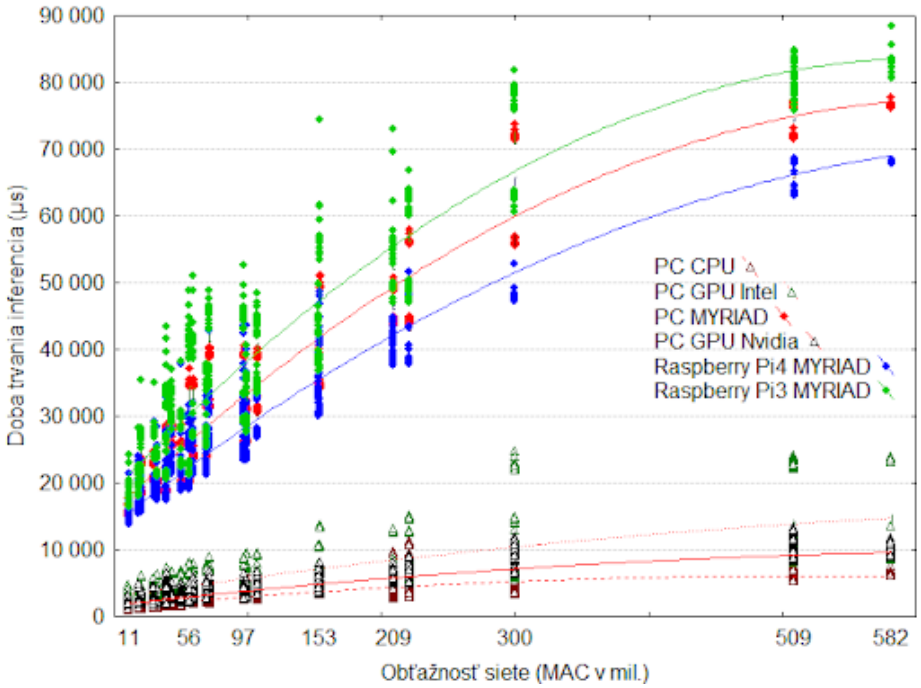
Na Obr. 5.4 sú znázornené priemerné časové hodnoty inferencie na všetkých zariadeniach. Pre lepšie vyhodnotenie výsledkov z 22 stupňov zložitosti siete (11 – 582 MAC v mil.) bola upravená perspektíva x-ovej os podľa skutočných vzdialeností

(skutočná perspektíva x-ovej osi; Obr. 5.5, 5.6a, 5.6b) a následne namerané údaje sú preložené krivkou polynómu druhého rádu, ktorá najlepšie poukazovala na trendy výsledkov. Vidno, že nárast doby trvania inferencie po obťažnosti 300 miliónov (resp. 200) MAC mal miernejšiu tendenciu nárastu. Najviac sa to prejavilo pri Raspberry Pi3 CPU (Obr. 5.5), Raspberry Pi3 MYRIAD (Obr. 5.6a) a PC CPU (Obr. 5.6b).

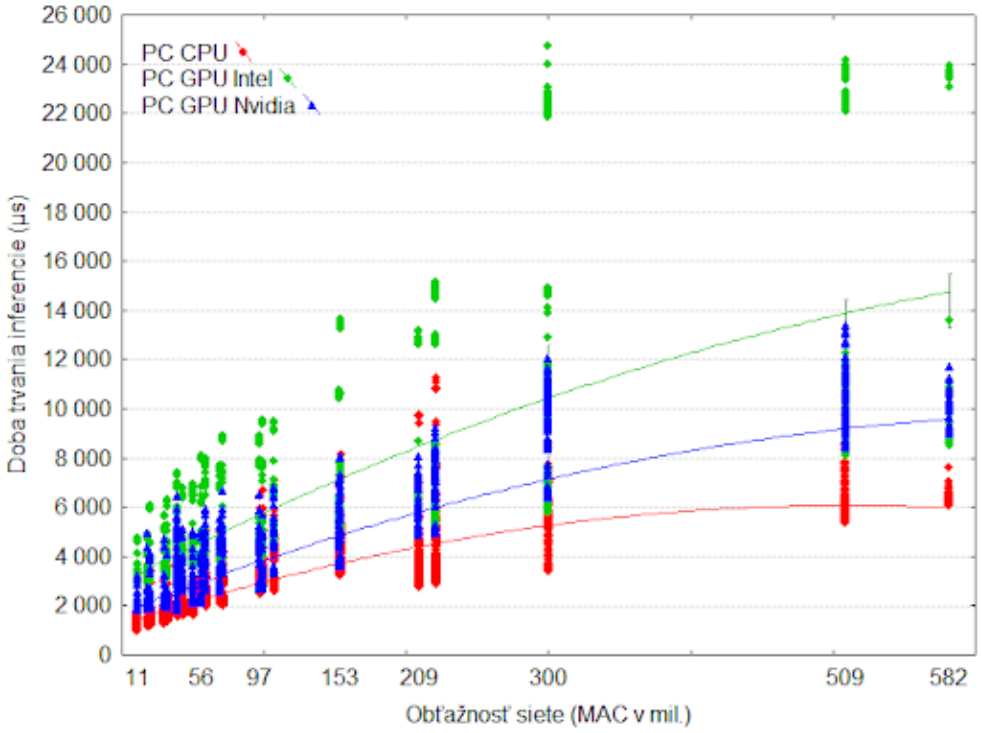


Obr. 5.5: Časové inferencie (µs) podľa obťažnosti v MAC (Multiplication and Accumulation operation, v miliónoch) na jednotlivých skúmaných zariadeniach. Súbory nameraných hodnôt sú preložené čiarami druhého polynómu.

Diskusia Experimenty neukázali Gaussovo (normálne) rozdelenie v analyzovaných súboroch dát napriek tomu, že by bolo na prvý pohľad očakávané. Dôvod je pravdepodobne ten, že testovanie síce prebehlo za najlepších možných podmienok ale nie za ideálnych. Môžme si všimnúť v grafoch 5.1 excentricitu (najmä pri výpočtoch robených na CPU a GPU) medzi mediánom a 2. ,3. kvartilom alebo maximálnymi hodnotami. Ďalej je veľmi zaujímavý fakt, že skoro vždy bola abnormálnou hodnotou maximálna. Keďže z povahy testu vyplýva maximálne zaťaženie hw častí výpočtom, akékoľvek zdržanie výpočtu, či už priame (CPU, GPU - bol zapnutý systém aj s grafickým rozhraním) alebo nepriame, presun dát do periférie MYRIAD mohlo drasticky ovplyvniť aktuálne meraný výpočet. Ako už bolo spomenuté, tak



(a) Časové inferencie (µs) podľa obťažnosti v MAC bez Pi3 a Pi4 CPU kvôli lepšej vizualizácií.



(b) Časové inferencie (µs) 3 najrýchlejších zariadení podľa obťažnosti v MAC.

Obr. 5.6

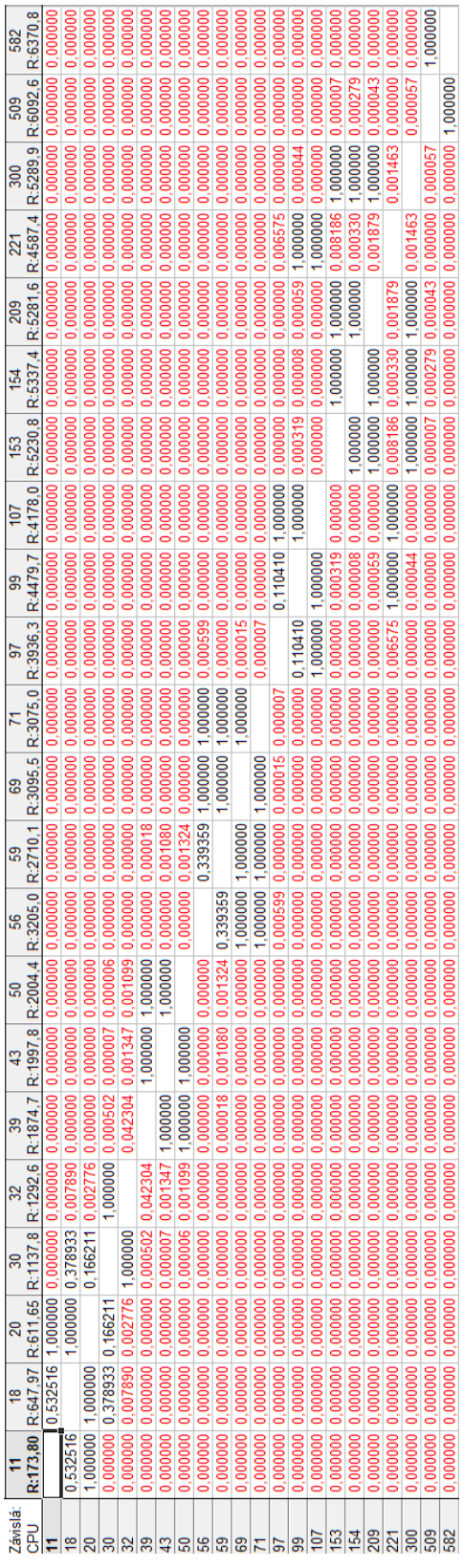
Doba trv.	Počet	PC				Raspberry			
infer.	mer.	CPU	GPU	MYRIAD	GPU	Pi3		Pi4	
(µs)	N				Nvidia	CPU	MYRIAD	CPU	MYRIAD
11	300	52141	86325	45150	126720	54766	45223	46764	53390
18	300	194390	185403	135150	202407	143964	137834	147549	167708
20	300	183496	231394	233095	283838	230522	310197	212416	205889
30	300	341344	516789	585705	590493	487981	646273	420704	498764
32	300	387793	335123	484962	344403	350614	491561	360254	402552
39	300	562405	546532	307806	221977	406928	235342	438612	334498
43	300	599333	799433	944594	1036869	749526	950193	705272	846665
50	300	601330	662328	770286	765603	700717	848029	672245	740910
56	300	961493	735921	414739	511262	610275	406363	693051	572016
59	300	813040	1082923	1265123	1248605	1011852	1307335	1009911	1118460
69	300	928663	889176	674626	714938	817923	617165	818216	710260
71	300	922499	963130	1117147	1085169	967703	1113537	999199	1035422
97	300	1180889	1246243	1455635	1387513	1285595	1480299	1286131	1368879
99	300	1343908	1076463	850541	804913	1116733	767882	1128487	942354
107	300	1253409	1276174	1043153	1030698	1212466	1040418	1195729	1208887
153	300	1569242	1399627	1424665	1449503	1473109	1398079	1463544	1479614
154	300	1601231	1325701	1255178	1228428	1396302	1212992	1410254	1324510
209	300	1584471	1557052	1665150	1660749	1651110	1665789	1642318	1663096
221	300	1376233	1518387	1575150	1564851	1581826	1574267	1597360	1574569
300	300	1586956	1646889	1755150	1745879	1753092	1754226	1754988	1754564
509	300	1827793	1842433	1845150	1866498	1855429	1845502	1850950	1845450
582	300	1911246	1859858	1935150	1911989	1924871	1934798	1929350	1934850

Obr. 5.7: Hodnoty súčtu poradia pre jednotlivé skupiny Kruskal-Wallisovho mediánového testu (počet pozorovaní pod a nad spoločným mediánom v jednotlivých skupinách) pre všetky testované zariadenia

tento fenomén bol minimalizovaný vyšším počtom meraní. Preto sa mediány s vysokou pravdepodobnosťou podobajú ideálnym hodnotám. Avšak celkové rozloženie to mohlo posunúť do jednej strany natoľko že sa už nejednalo o Gaussovo rozloženie (priemer $\neq$ medián).

Experimenty ukázali globálne (v rozsahu testovaných záťaží) stúpajúci charakter potrebného času na vykonanie inferencie pri všetkých zariadeniach. Takisto pre tento výskum časovej premennej platí, že medzi všetkými skúmanými zariadeniami bol významný rozdiel a to v tomto poradí – od najrýchlejšieho po najpomalšie: PC: CPU < PC: GPU – Nvidia < PC: GPU – Intel < Raspberry Pi4: MYRIAD < PC: MYRIAD < Raspberry Pi3: MYRIAD < Raspberry Pi4: CPU < Raspberry Pi3: CPU. Bol pozorovaný extrémny nárast rýchlosti pri akcelerácii výpočtu na platforme Raspberry Pi. Raspberry Pi 4 CPU bez akcelerátora malo až 116x horší čas oproti PC CPU a 50x horší oproti PC Intel GPU (porovnaj Obr. 5.5 a 5.6a, 582 mil. MAC). Po aplikácii akcelerátora sa rozdiel zlepšil v poradí na 11x a 5x násobný (porovnanie koľko krát pomalšie sa vykonal výpočet). Takéto zlepšenie surového výkonu je veľmi významné a v praxi to znamená že mnohé aplikácie si môžu presunúť výpočet priamo na koncové zariadenie. To môže hrať veľkú rolu pri vyhodnocovaní dát v reálnom alebo skoro reálnom čase. Na druhej strane z tohto hľadiska výkonu akcelerátor nekonkuruje bežnému počítaču.

MAC metrika Ako už bolo spomenuté z globálneho hľadiska celého rozsahu náročnosti bol časový nárast spracovávania pre všetky zariadenia približne lineárny. Tento priebeh sám o sebe bol vopred očakávaný a nieje prekvapivý. Očakávaný priebeh sa však zmení ak sa na dáta pozrieme na menších intervaloch. Rýchlosť na intervaloch 20-32, 39-50, 43-56 klesá napriek tomu, že hodnota MAC si zachováva svoj stúpajúci charakter (porovnaj 5.1). Tento jav je najlepšie pozorovateľný pre všetky zapojenia akcelerátoru MYRIAD avšak skoro nepatrný alebo nevýznamný pre CPU spustenie. Toto pozorovanie podporuje aj štatistika. Kruskal-Walisovho test z tabuľky 5.3 nám vraví, že namerané hodnoty pre úrovne rôzne úrovne vrámci jedného zariadenia sú štatisticky odlišné ($p < 0.05$). Podľa toho je jasné, že nejde o zlú interpretáciu grafov. Hodnoty súčtu poradia pre jednotlivé skupiny 5.7 z tohto testu nám ďalej potvrdzujú zmenu na pozorovaných záťažiach. Podstatné je že podľa mnohonásobného porovnávania nameraných hodnôt vrámci jedného zariadenia (každé s každým) tieto rozdiely sú štatisticky významne (5.8b a 5.8a). Pravdepodobne je to tým, že výkon akcelerátora nieje jednoznačne závislý len od počtu operácii MAC ale existuje ešte iný skrytý parameter. Checkpointy pre sieť MobileNetV2 sa okrem počtu operácií MAC líšia ešte 2 ďalšími parametrami, rozmerom spracovávaného obrázku a Dropout hodnotou. Tieto dve hodnoty sú reprezentované v MAC operáciách ale jedna z nich spôsobuje nerovnomerne väčšiu záťaž pre akcelerátor.



Závislá: CPU	11	18	20	30	32	39	43	50	56	59	69	71	97	99	107	153	154	209	221	300	509	582
	R:173,80	R:647,97	R:611,65	R:1137,8	R:1292,6	R:1874,7	R:1997,8	R:2004,4	R:3205,0	R:2710,1	R:3095,5	R:3075,0	R:3936,3	R:4479,7	R:4178,0	R:5230,8	R:5337,4	R:5281,6	R:4587,4	R:5289,9	R:6092,6	R:6370,8
11		0,532516	1,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
18	0,532516		1,000000	0,378933	0,007890	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
20	1,000000	1,000000		0,166211	0,002776	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
30	0,000000	0,378933	0,166211		1,000000	0,000502	0,000007	0,000006	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
32	0,000000	0,007890	0,002776	0,000000		0,042304	0,001347	0,001099	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
39	0,000000	0,000000	0,000000	0,000502	0,042304		1,000000	1,000000	0,000000	0,000018	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
43	0,000000	0,000000	0,000000	0,000007	0,001347	1,000000		1,000000	0,000000	0,001080	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
50	0,000000	0,000000	0,000000	0,000006	0,001099	1,000000	1,000000		0,000000	0,001324	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
56	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000		0,339359	1,000000	1,000000	0,000599	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
59	0,000000	0,000000	0,000000	0,000000	0,000000	0,000018	0,001080	0,001324	0,339359		1,000000	1,000000	0,000015	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
69	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	1,000000		1,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
71	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	1,000000	1,000000		0,000007	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
97	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000599	0,000000	0,000015	0,000007		0,110410	1,000000	0,000319	0,000008	0,000059	0,006575	0,000044	0,000000	0,000000
99	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,110410		1,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
107	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	1,000000		0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
153	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000319	0,000000	0,000000		1,000000	1,000000	0,000000	0,008186	0,000007	0,000000
154	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000008	0,000000	1,000000	1,000000		1,000000	0,000330	1,000000	0,000279	0,000000
209	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000059	0,000000	1,000000	1,000000	1,000000		0,001879	1,000000	0,000043	0,000000
221	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,006575	1,000000	1,000000	0,008186	0,000330	0,001879		0,001463	0,000000	0,000000
300	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000044	1,000000	1,000000	1,000000	1,000000	1,000000	0,001463		0,000057	0,000000
509	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000007	0,000279	0,000043	0,000000	0,000057		1,000000
582	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	

(a) PC CPU

Závislá: PC MYRIAD	11	18	20	30	32	39	43	50	56	59	69	71	97	99	107	153	154	209	221	300	509	582
	R:150,50	R:450,50	R:776,98	R:1952,3	R:1616,5	R:1026,0	R:3148,6	R:2567,6	R:1382,5	R:4217,1	R:2248,8	R:3723,8	R:4852,1	R:2835,1	R:3477,2	R:4748,9	R:4183,9	R:5550,5	R:5250,5	R:5850,5	R:6150,5	R:6450,5
11		1,000000	0,013057	0,000000	0,000000	0,000004	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
18	1,000000		1,000000	0,000000	0,000000	0,049943	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
20	0,013057	1,000000		0,000000	0,000016	1,000000	0,000000	0,000000	0,022978	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
30	0,000000	0,000000	0,000000		1,000000	0,000001	0,000000	0,017694	0,057562	0,000000	1,000000	0,000000	0,000000	0,000003	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
32	0,000000	0,000000	0,000016	1,000000		0,034007	0,000000	0,000000	1,000000	0,000000	0,011157	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
39	0,000004	0,049943	1,000000	0,000001	0,034007		0,000000	0,000000	1,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
43	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,043412	0,000000	0,000000	0,000002	0,050377	0,000000	1,000000	1,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
50	0,000000	0,000000	0,000000	0,000000	0,017694	0,000000	0,043412		0,000000	0,000000	1,000000	0,000000	0,000000	0,000001	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
56	0,000000	0,000000	0,022978	0,057562	1,000000	1,000000	0,000000	0,000000		0,000000	0,000006	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
59	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000		0,000000	0,000000	0,000000	0,000000	0,351527	0,010319	0,000000	0,000456	0,145543	1,000000	0,000000	0,000000
69	0,000000	0,000000	0,000000	0,000000	0,111157	0,000000	0,000002	1,000000	0,000006	0,000000		0,000000	0,000000	0,037842	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
71	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,050377	0,000000	0,000000	0,351527	0,000000		0,000000	0,000003	1,000000	0,000000	0,000000	0,716617	0,000000	0,000000	0,000000	0,000000
97	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,010319	0,000000	0,000000		0,000000	1,000000	0,000000	0,000000	0,004036	0,001653	1,000000	0,000000	0,000000
99	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	1,000000	0,000000	0,000000	0,037842	0,000003	0,000000		0,008494	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
107	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	0,000001	0,000000	0,000456	1,000000	0,000000	0,000000	0,008494		0,000000	0,001282	0,000000	0,000000	0,000000	0,000000	0,000000
153	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,145543	0,000000	0,000000	1,000000	0,000000	0,000000		0,065112	0,000059	0,291740	0,000000	0,000000	0,000000
154	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	0,000000	0,716617	0,004036	0,000000	0,001282	0,065112		0,000000	0,000000	0,000000	0,000000	0,000000
209	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,001653	0,000000	0,000000	0,000000	0,000059	0,000000		1,000000	1,000000	0,026553	0,000002
221	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	0,000000	0,000000	0,291740	0,000000	1,000000		0,026553	0,000002	0,000000
300	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	0,026553		0,026553	0,000000
509	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,026553	0,000002	1,000000		1,000000
582	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,026553	0,000000	0,026553	1,000000	

(b) PC MYRIAD

Obr. 5.8: Výsledok mnohonásobného porovnávania spotreby zariadenia všetkých skupín zložitosti siete (11 – 582) (Kruskal- Walisov test). P-hodnota nižšia ako 0,05 (červené čísla) znamená významný rozdiel medzi dvomi kategóriami.

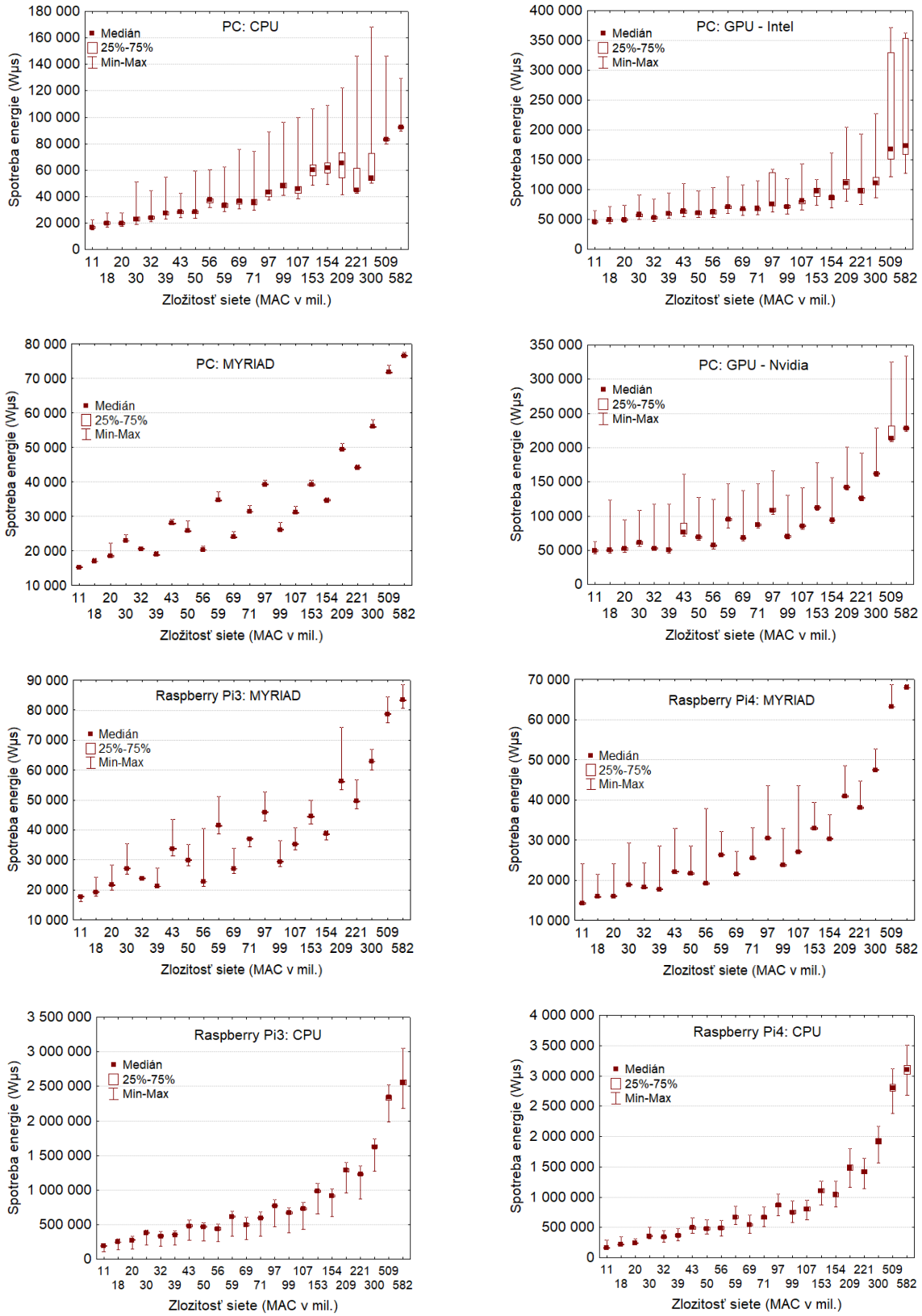
5.2.2 Výsledky a analýza experimentu 1B - energetická náročnosť

Výsledky analýzy spotreby pri výpočte jednotlivých zariadení, podobne ako vo výstupoch predchádzajúcej analýzy, sú grafický zobrazené krabicovými grafmi (Obr. 5.9). Sú tu znázornené popisné štatistiky: minimum, prvý kvartil, medián, tretí kvartil a maximum. Zaujímavosťou pri minimách a maximách je väčší rozptyl na stranu maximálnych hodnôt takmer vo všetkých zariadeniach. Prejavuje sa tu ešte viditeľnejšie ako pri experimente 1A(pozri 5.2.1). Rozmedzie hodnôt na y-ovej osi znázorňuje, že najnižšie hodnoty spotreby vykazujú všetky zariadenia MYRIAD: a Raspberry Pi4 (do 70 000), PC (do 80 000) a Raspberry Pi3 (do 90 000Wµs). Najväčšie hodnoty s najväčšími rozptylmi stále boli na Raspberry Pi3 a Pi4 CPU (do 3 000 000 – 3 500 000 Wµs).

Presné množstvo spotrebovanej energie pre všetky zariadenia sme nevedeli spoľahlivo a konzistentne zmerať. Preto tento údaj bol odvodený od spotreby uvedenej v špecifikáciách pre jednotlivé zariadenia a času výpočtu. Taktiež bol predpoklad, že zariadenie počas výpočtu pracovalo s maximálne možným výkonom. Čo bolo priamo pozorované rôznymi monitorovacími utilitami alebo nepriamo potvrdené v maximách meraní. pri výpočtoch sa vychádzalo z týchto hodnôt zariadení:

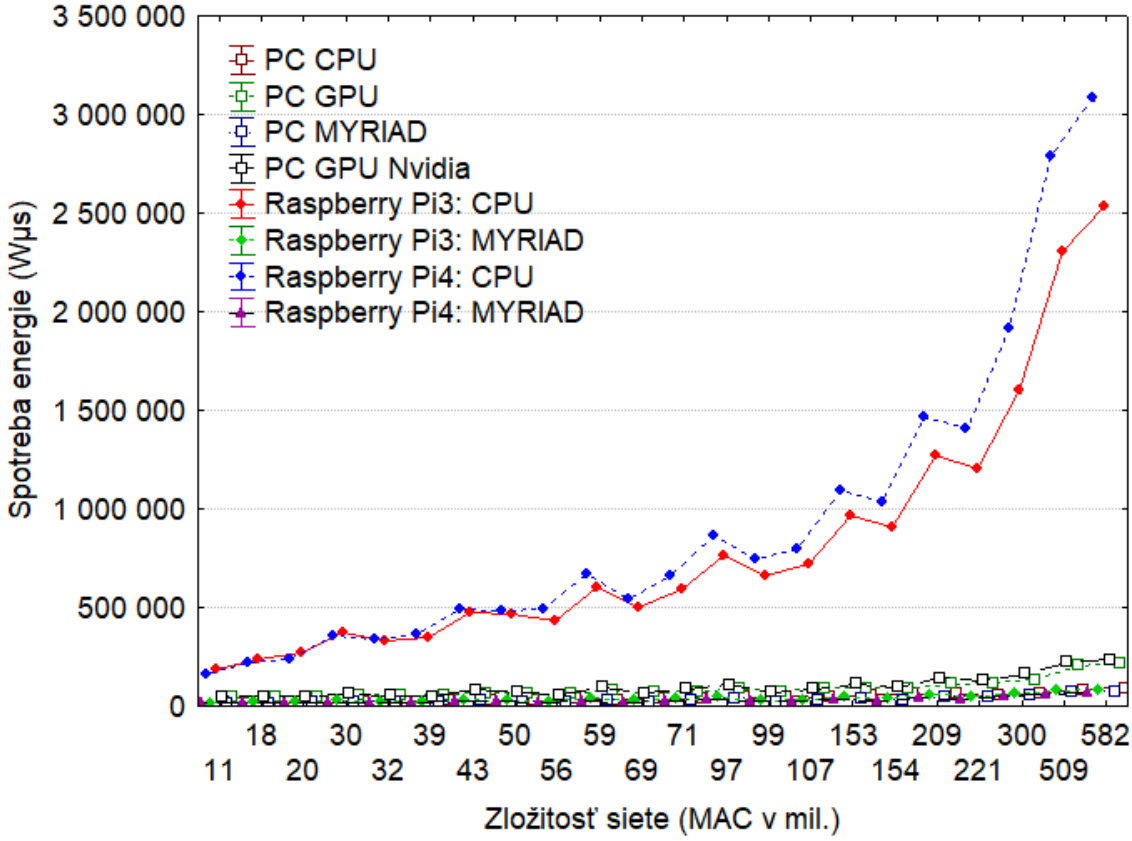
1. Intel®Core™i7-8550U Processor, Intel®UHDGraphics 620 15W [30]
2. Nvidia Gforce mx150 25W [31]
3. Intel Movidius NCS 1W [32]
4. Raspberry Pi 4 5.1W [33]
5. Raspberry Pi 3 3.7W [33]

Pre overenie typu rozdelenia skúmaných dát bol použitý Shapiro - Wilkov W test normality. Pre homogenitu rozptylu – Cochranov a Bartlettov testy. Pri všetkých skúmaných súboroch vyšli rovnaké výsledky ako v Experimente 1. Hladina významnosti testov bola veľmi nízka ($P < 0{,}05$) a skúmané dáta nemajú normálne rozdelenie ani homogénne. Ďalším skúmaním Kruskal-Walisovym testom sa potvrdilo, že celková variabilita v spotrebe energie pre všetky skúmané zariadenia v závislosti od zložitosti siete MobileNetV2 prekročila kritickú hodnotu, čím zamietame hypotézu o zhode rozptylov a tým aj nulovú hypotézu (H0) analýzy rozptylu, že mediánové hodnoty sledovaných skupín sa nelíšia.V prípade zamietnutia H0 platí alternatívne hypotéza H1: Nie všetky stredné hodnoty sú rovnaké (t.j. aspoň jedna zo stredných hodnoty sa líši od ostatných). Ak analýzou rozptylu sa zamietla nulová hypotéza (H0: µ1 = µ2 = µ3 = ... = µm, m-počet porovnávaných skupín) o vplyve pôsobiaceho faktora (zložitosť siete), pokračujeme v ďalšom testovaní pomocou mnohonásobného porovnávania. Takto určíme štatistickú významnosť jednotlivých rozdielov mediánových hodnôt u všetkých možných párov porovnávaných skupín zložitosti siete (11



Obr. 5.9: Priebeh spotreby energie (Wµs) podľa obťažnosti siete v MAC (v miliónoch) na jednotlivých skúmaných zariadeniach (N=300, medián, kvartily, neodľahlé dáta)

– 582 mil. MAC). V Obr. A.1-A.3 vidíme výsledky P-hodnôt pre jednotlivé testy spotreby energie. Ako aj v predchládzajúcom Experimente platí, že P-hodnota nižšia ako 0,05 (červené čísla) znamená významný rozdiel medzi dvomi skupinami.

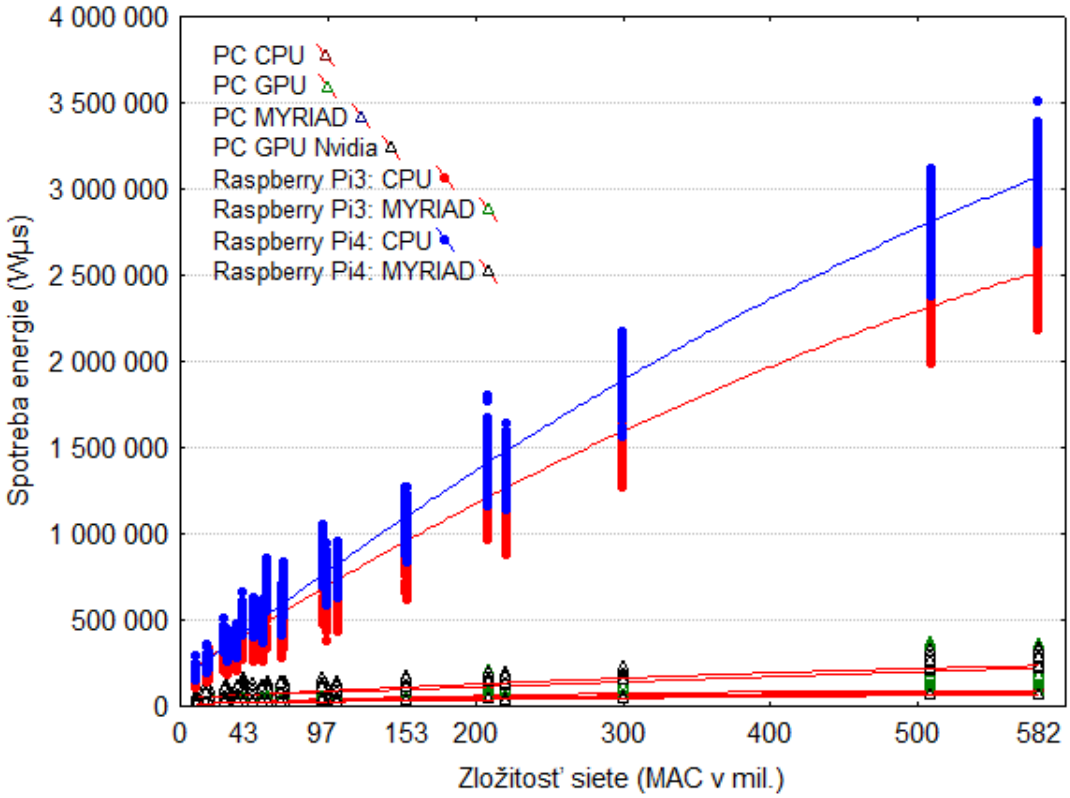


Obr. 5.10: Priebeh spotreby energie (Wµs) podľa obťažnosti v mil. MAC na jednotlivých skúmaných zariadeniach (N=300, Priemer; Svorka: Priemer ±0,95 Interval spoľahlivosti)

5.2.3 Porovnanie spotreby jednotlivých testovaných zariadení medzi sebou

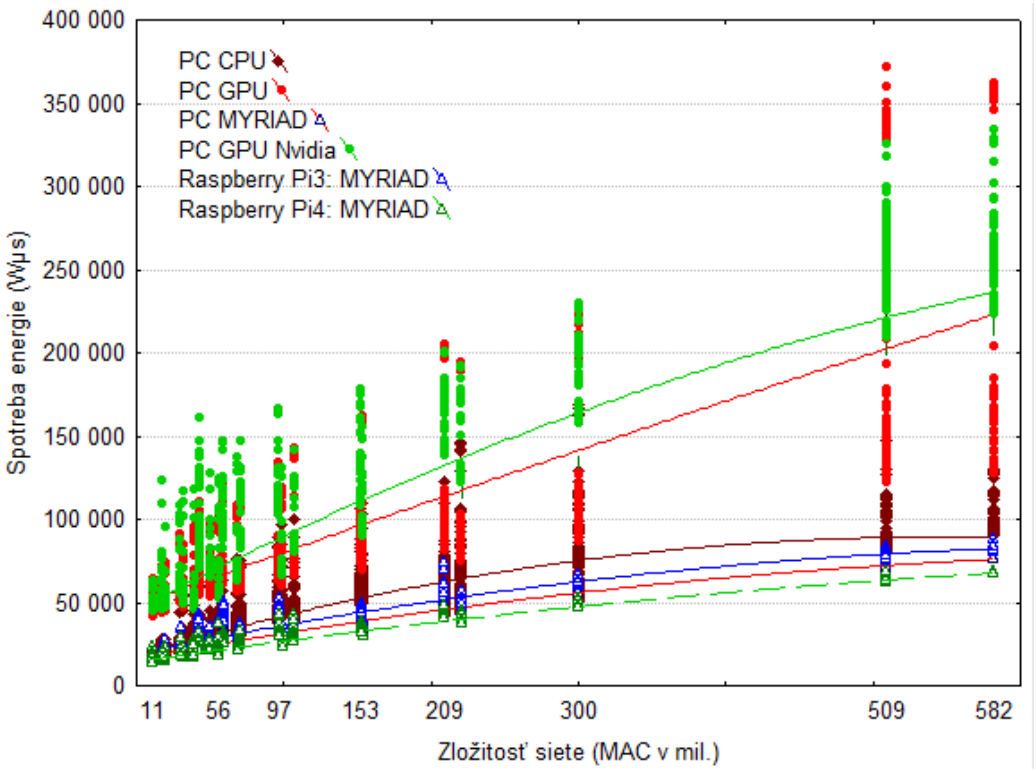
Na Obr. 5.10 sú znázornené priemerné hodnoty spotreby energie (Wµs) na všetkých zariadeniach. Ako už bolo spomenuté, najväčšiu spotrebu majú Raspberry Pi3 a Pi4 CPU. Dokonca aj priebeh podľa zložitosti siete majú totožný, s tým že hodnoty Pi4 na začiatku boli menšie, pri 32 sa vyrovnali Pi3 ale postupne so zložitosťou siete rozdiel narastal. Pre lepšie vyhodnotenie priebehu analyzovaných výsledkov 22 stupňov zložitosti sieti (11 – 582 MAC v mil.) bola upravená perspektíva x-ovej rovnako ako v experimente A (reálna x-ová os; Obr. 5.11, 5.12a, 5.12b). Namerané údaje boli tiež preložené krivkou polynómu druhého radu, ktorá najlepšie poukazovala na trendy

výsledkov. Je vidieť, že nárast spotreby energie so zložitosťou siete je u všetkých skúmaných zariadeniach približné lineárny, okrem PC CPU. Na tomto zariadení po obtiažnosti 300 miliónov MAC je nárast spotreby miernejší (Obr. 5.12b). Zariadenia s najlepšou spotrebou, ktoré sme odhadovali podľa krabicových grafov (Obr. 5.9) sme potvrdili preloženými krivkami na Obr. 5.12b - od najnižšej spotreby: a Raspberry Pi4 MYRIAD (hnedá), PC MYRIAD (zelená) a Raspberry Pi3 MYRIAD (modrá).

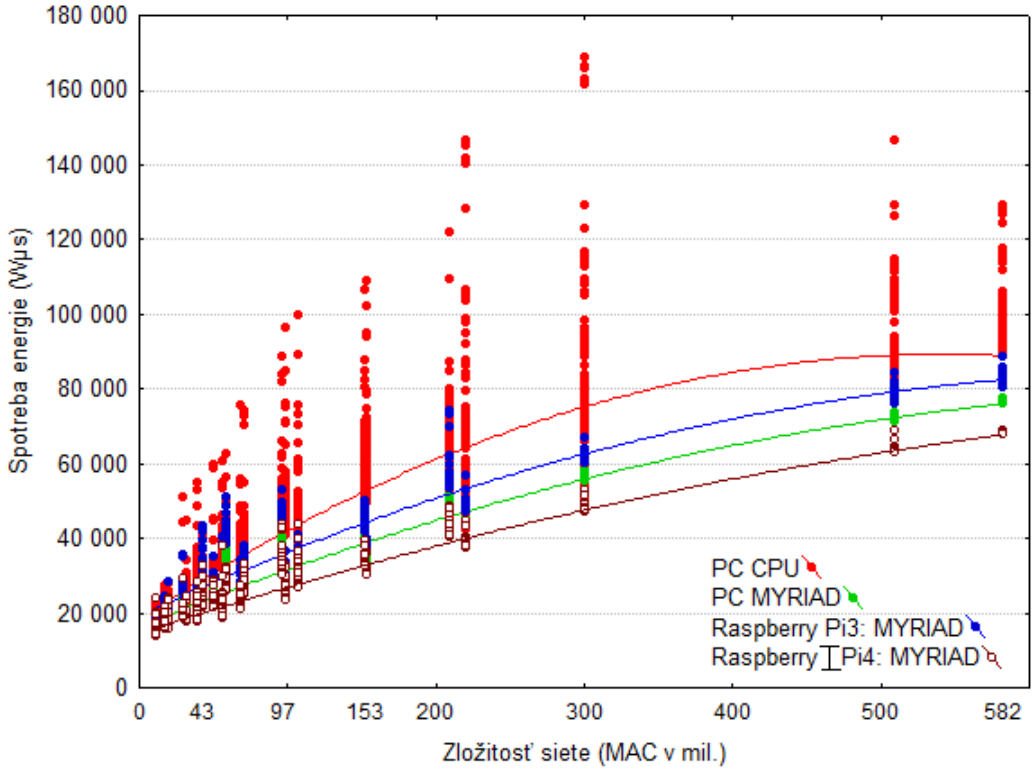


Obr. 5.11: Spotreba energie (Wµs) podľa obťažnosti v mil. MAC na jednotlivých skúmaných zariadeniach. Súbory nameraných hodnôt sú preložené čiarami polynómu druhého stupňa.

Diskusia Druhá časť experimentov ukázala silné stránky akcelerátora a oblasť kde jasne dominuje medzi ostatnými zariadeniami. Kým v experiment 1 A sme síce zaznamenali značné urýchlenie výpočtu toto urýchlenie sa podľa očakávaní vzťahovalo výhradne na platformu Raspberry Pi. Experiment 1 B ukázal pre Raspberry Pi4 až **46 násobný** rozdiel v spotrebe energie (grafy 5.11 a 5.12b,582 mil. MAC). Akcelerátor je úspornejší aj ako PC CPU. Úspora bola 25Wms, čo je zaujímavý poznatok. Avšak táto úspora nebola dostatočná na prekrytie výkonnostnej penalizácie z experimentu 1 A. Je potrebné znova poznamenať, že akcelerátor je určený pre úsporné



(a) Spotreba energie (Wμs) podľa obťažnosti v mil. MAC. Bez znázornenia údajov pre Raspberry Pi 3 a 4 CPU kvôli lepšej vizualizácií.



(b) Časové inferencie (μs) podľa obťažnosti v mil. MAC na jednotlivých skúmaných zariadeniach (PC: CPU, MYRIAD a Raspberry: Pi3 MYRIAD, Pi4 MYRIAD). Bez znázornenia údajov pre Raspberry: Pi3 CPU, Pi4 CPU a PC: GPU a GPU Nvidia kvôli lepšej vizualizácií.

Obr. 5.12

používanie, kedy sa dáva prednosť výdrži batérie pred absolútnym výkonom. Prekvapivým výsledkom je výsledná spotreba PC CPU, ktorá je vzhľadom na výkon veľmi nízka. Aj keď sa tento modle procesora radí do úspornej triedy dosiahol spotrebu relatívne porovnateľnú s akcelerátorom. V tabuľke 5.13 sú zhrnuté poradia zariadení podľa nameraných výsledkov.

Por. č.	Experiment 1	Experiment 2
1	PC: CPU	Raspberry Pi4: MYRIAD
2	PC: GPU – Nvidia	PC: MYRIAD
3	PC: GPU – Intel	Raspberry Pi3: MYRIAD
4	Raspberry Pi4: MYRIAD	PC: CPU
5	PC: MYRIAD	PC: GPU – Intel
6	Raspberry Pi3: MYRIAD	PC: GPU – Nvidia
7	Raspberry Pi4: CPU	Raspberry Pi3: CPU
8	Raspberry Pi3: CPU	Raspberry Pi4: CPU

Obr. 5.13: Vyhodnotenie skúmaných zariadení od najrýchlejších (Experiment 1A) resp. najekonomickejších (Experiment 1B)

5.3 Experiment 2: Vyhodnotenie výkonu pri spracovaní videa v reálnej aplikácii

Keďže surový výkon pri výpočte ingerencie má síce podstatný ale zároveň nie výlučný vplyv na rýchlosť reálnych aplikácii (využívajúcich CNN). Cieľom tohto experimentu je podrobiť akcelerátor čo najranejšej zátaži s pomocou demonštračnej aplikácie (kap. 4.1), aby sme zistili reálny prínos akcelerátora na urýchlenie aplikácii na platforme Raspberry Pi.

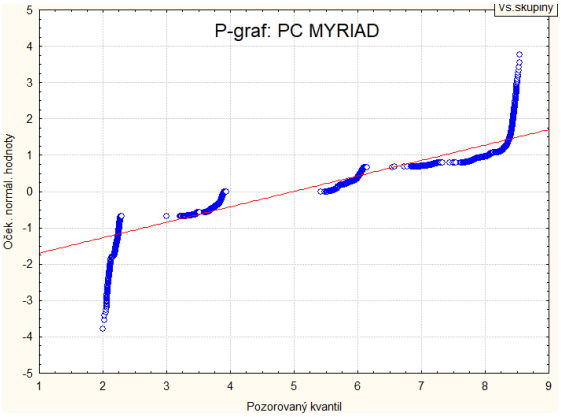
Experimenty prebehli vo vzťahu k 2 parametrom. Prvý parameter bol počet tvári. Ten priamo ovplyvňoval zložitosť celkového výpočtu ingerencie. Zložitosť sa sťažovala s narastajúcim počtom tvárí tým, že čím viacej tvárí bolo na videu tým viacej krát as musel vykonávať 2. a 3. krok pipeliny spracúvajúcej obraz (extrakcia významných bodov z tváre 4.1.2, a identifikácia tváre 4.1.3). Druhý parameter sa týkal rozlíšenia vstupných dát a nepriamo ovplyvňoval výkon zariadení, keďže so zvyšujúcim sa rozlíšením sa museli transformácie matíc (vstupného obrazu) robiť pre väčšie veľkosti dimenzií vstupných dát.

Na to aby sa dosiahla konzistencia medzi spusteniami ako vstupné dáta sa používali krátke videoklipy. Tieto testovacie videoklipy mali dĺžku 5s s frame rate-om 60fps, čiže jeden klip obsahoval 300 vstupných framov. Testy sa vykonali na 100 rôznych vzorkách. Povaha a cieľ testu nevyžadoval väčšiu databázu testovacích dát. Test sa snažil simulovať reálne nasadenie akcelerátoru v možnej aplikácii. Keďže nieje žiaden dôvod predpokladať výkonnostný rozdiel medzi statickým a meniacim obrazom, video pozostávalo z jedného obrazu tváre. Pri kódovaní videa neboli použité žiadne kompresné algoritmy. Testovacie videá boli v siedmich rozlíšeniach od 240p až po 2160p (240p, 360p, 480p, 720p, 1080p, 1440p, 2160p). V každom z uvedených rozlíšení boli videá s 1, 2, 4 a 8 tvárami. Sumárne sa tento experimente spúšťal na 28 rôznych videách. Výsledky boli porovnávané medzi sebou podľa oboch aspektov (rozlíšenie a počet tvárí). Experiment bol zameraný na platformu Intel Movidius NCS (MYRIAD) (pozri 3.1) a knižnicu OpenVino (pozri 3.3.1). Preto bol aplikovaný len na zariadenia podporujúce túto knižnicu: PC CPU, PC MYRIAD a Raspberry Pi4 MYRIAD.

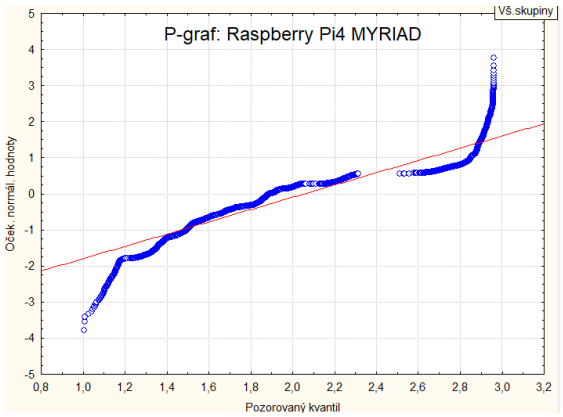
5.3.1 Výsledky a analýza experimentu 2 - demo aplikácia

Na začiatku som opäť analyzované súbory podrobil testu normality a homogenity. Ako v predchádzajúcich experimentoch ani v tomto sa nepotvrdila zhoda s normálnym rozdelením (Graf 5.14) ani s homogenitou rozptylu (Tab. 5.15), ktoré sú podmienkou použitia parametrickej ANOVA. V experimente som pokračoval skúmaním pomocou Kruskal-Walisovho testu. Pri každom testovaní som zistil význam-

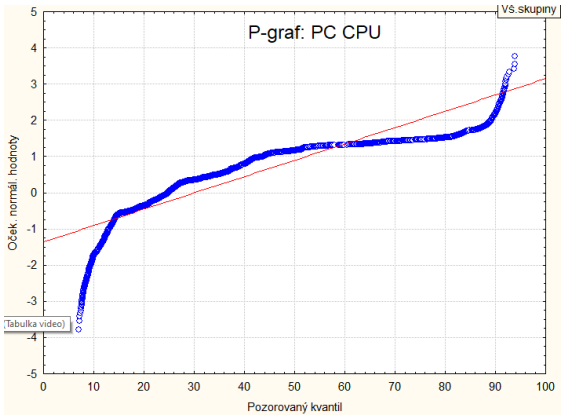
nosť rozdielov medzi skupinami, preto v ďalšom testovaní som pokračoval pomocou mnohonásobného porovnávania.



(a)



(b)



(c)

Obr. 5.14: Vyšetrenie normálneho pravdepodobnostného rozloženia. Grafy pre dáta z PC CPU, PC MYRIAD a Raspberry Pi4 MYRIAD.

	Testy homogenity rozptylu (Tabulka video) Efekt: "Počet tvári"*"Rozlišenie"				
	Hartleyů F-max	Cochranů C	Bartlett Chí-kv.	ův SV	p
PC CPU	470,4319	0,251852	11122,23	27	0,00
PC GPU	88,8869	0,147316	6469,43	27	0,00
PC MYRIAD	115,5782	0,254586	6191,58	27	0,00
Raspberry Pi4 MYRIAD	19,8546	0,187097	2233,20	27	0,00

Obr. 5.15: Výsledky Hartleyovho, Cochranovho a Bartlettovho testov pre homogenitu rozptylov

5.3.2 Vyhodnotenie výkonu pri spracovaní videa v reálnej aplikácii pre rozdielny počet tvári

Na Obr. 5.16 vidíme výsledky P-hodnôt mnohonásobného porovnávania výkonnosti vybraných zariadení pri spracovaní údajov rôzneho počtu tvári. Týmto sa potvrdil vplyv počtu tvárí na náročnosť spracovania pre všetky zariadenia. S rastúcim množstvom tvári významné klesá počet spracovaných snímkov na všetkých analyzovaných zariadeniach (FPS, Obr. 5.17a). Podrobnejšia analýza vplyvu faktora počet tvári pre jednotlivé rozlíšenia je vykreslená na Obr. 5.18a, kde vidieť klesajúci trend počtu FPS s narastajúcim rozlíšením. Avšak pri analýze grafu Obr. 5.18b, ktorý detailnejšie zobrazuje zariadenie MYRIAD na PC a Raspberry Pi4 je pozorovateľný rozdiel vplyvu rozlíšenia. Pri nižšom rozlíšení rozdiel nieje prítomný. Od rozlíšenia 1080p PC MYRIAD vykazuje väčší počet spracovaných FPS.

Závislá: Rasp.Pi4 M.	1 R:1997,1	2 R:1798,4	4 R:1264,1	8 R:542,41
1		0,000026	0,00	0,00
2	0,000026		0,00	0,00
4	0,000000	0,000000		0,00
8	0,000000	0,000000	0,00	

(a)

Závislá: PC MYRIAD	1 R:7350,5	2 R:5250,5	4 R:3150,5	8 R:1050,5
1		0,00	0,00	0,00
2	0,00		0,00	0,00
4	0,00	0,00		0,00
8	0,00	0,00	0,00	

(b)

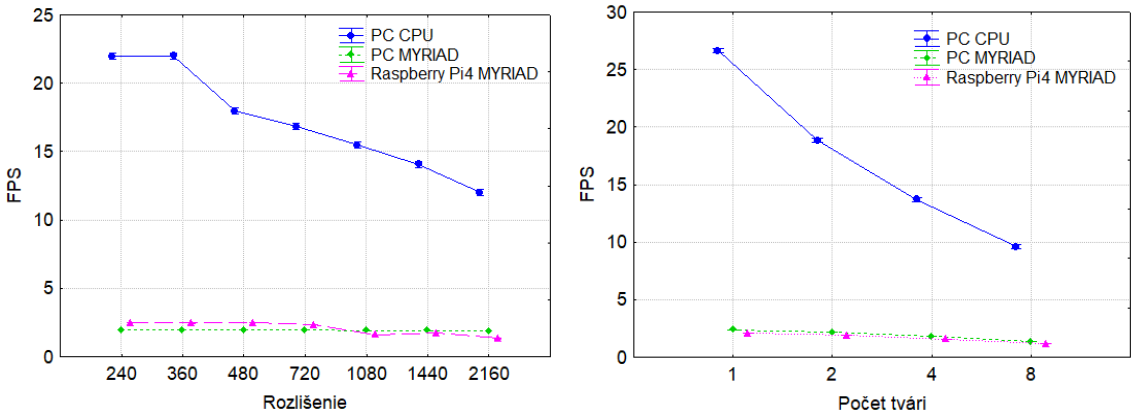
Závislá: PC CPU	1 R:6794,9	2 R:5593,9	4 R:3206,5	8 R:1206,7
1		0,00	0,00	0,00
2	0,00		0,00	0,00
4	0,00	0,00		0,00
8	0,00	0,00	0,00	

(c)

Obr. 5.16: Mnohonásobné porovnávanie priemerného poradia pre všetky skupiny podľa počtu tvári (triediaca premenná) na skúmaných zariadeniach (PC CPU, PC MYRIAD a Raspberry Pi4 MYRIAD). P-hodnota nižšia ako 0,05 (červené čísla) znamená významný rozdiel medzi dvomi skupinami.

5.3.3 Vyhodnotenie výkonu pri spracovaní videa v reálnej aplikácii pre rozdielne rozlíšenie snímok

Z výsledkov P-hodnôt mnohonásobného porovnávania výkonnosti vybraných zariadení pri spracovaní údajov rôzneho rozlíšenia snímok (Obr. 5.19) vidíme, že pri najnižších rozlíšeniach (240p, 360p, 480p) nie sú štatisticky významné rozdiely. Avšak postupne nastáva pokles v množstve spracovaných snímok (Obr. 5.17b, Obr. 5.19).



(a) (b)

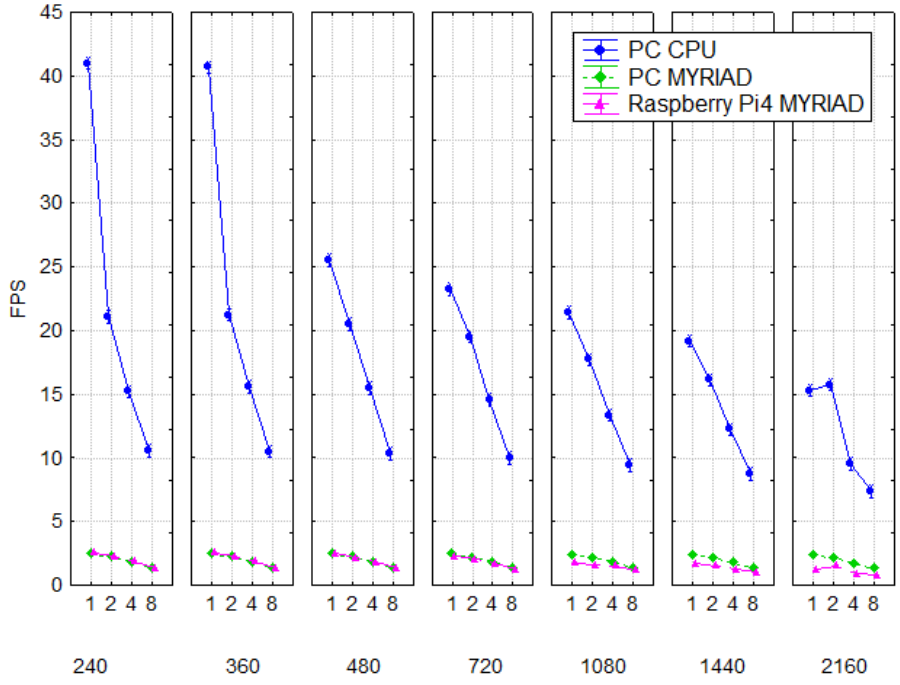
Obr. 5.17: Priemerné dosahované FPS (Frames Per Second) na skúmaných zariadeniach pre: a) rôzny počet tvári (1, 2, 4 a 8 tvári), b) rôzne rozlíšenie (240p, 360p, 480p, 720p, 1080p, 1440p, 2160p)

Podrobnejšia analýza pre vplyv faktora rozlíšenie snímok je vykreslená na Obr. 5.18. Sú tu vykreslené trendy množstva spracovania snímok podľa rozlíšenia snímok pre rôzne počty tváre. Vidíme, že so stúpajúcim počtom tvári trend poklesu analyzovaných snímok pre PC CPU čiastočne aj pre Raspberry Pi4 MYRIAD je miernejší. Zaujímavosťou je výstup pre PC MYRIAD kde vplyv rozlíšenia na počet FPS je nepatrný. Avšak faktor počet tvári mal vplyv na všetky zariadenia – s počtom tvári klesá počet spracovaných FPS. Takýto výsledok sa očakával.

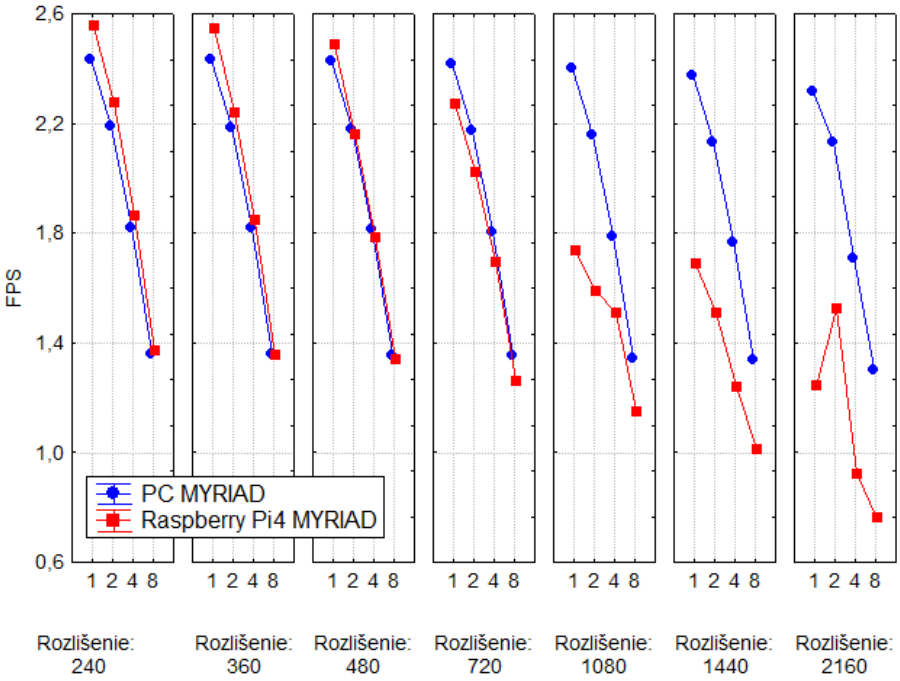
5.3.4 diskusia

Experiment potvrdil očakávanie, že zo stúpajúcim počtom osôb stúpa latencia systému. Dôvodom je fakt, že s každou novou tvárou sa vykonávajú navyše 2 z troch inferencii v aplikácii (spomenuté v kapitole 5.3) Experiment však odhalil zaujímavú skutočnosť, ktorá je v grafe 5.20 jasne vidieť. Vplyv rozlíšenia nieje rovnaký pre každú veľkosť rozlíšenia. Najmä pri porovnaní akcelerátoru na 2 platformách grafe 5.20b je vidno, prudkú zmenu rýchlosti spracovávania (FPS) jednotlivých rozlíšeniach po prekročení hranice 480p. Tento jav je možné vidieť v najväčšej mire na platforme Raspberry Pi4 a je potvrdený aj výsledkom z analýzy na Obr. 5.19. Aj keď sa dá pozorovať do určitej mierý aj na PC CPU zariadení s výnimkou grafu pre počet tvárí 1, avšak menej dramatický (z hľadiska percentuálneho poklesu rýchlosti oproti maximálnej rýchlosti).

V aplikácii dochádza k úprave a normalizácii vstupného obrazu pred tým ako sa vykoná inferencia. Táto normalizácia zahŕňa transformáciu matíc pomocou kniž-



(a) Porovnanie počtu FPS pre rôzny počet tvári (1, 2, 4, 8) na skúmaných zariadeniach (PC CPU, PC MYRIAD a Raspberry Pi4 MYRIAD). Porovnanie je vykreslené zvlášť pre každé rozlíšenie (240p, 360p, 480p, 720p, 1080p, 1440p, 2160p). N=300, Priemer; Svorka: Priemer ±0,95 Interval spoľahlivosti.



(b) Porovnanie počtu FPS pre rôzny počet tvári (1, 2, 4, 8) na zariadeniach PC MYRIAD a Raspberry Pi4 MYRIAD. Porovnanie je vykreslené zvlášť pre každé rozlíšenie (240p, 360p, 480p, 720p, 1080p, 1440p, 2160p). Údaje ako na Obr. 5.18a, len bez PC CPU kvôli lepšej vizualizácií. N=300, Priemer; Svorka: Priemer ±0,95 Interval spoľahlivosti.

Obr. 5.18

Závislá: Rasp.Pi4 M.	**240** R:1977,7	360 R:1921,8	480 R:1809,7	720 R:1631,6	1080 R:1118,5	1440 R:859,64	2160 R:484,63
240		1,000000	0,069056	0,000000	0,000000	0,000000	0,000000
360	1,000000		1,000000	0,000008	0,000000	0,000000	0,000000
480	0,069056	1,000000		0,038752	0,000000	0,000000	0,000000
720	0,000000	0,000008	0,038752		0,000000	0,000000	0,000000
1080	0,000000	0,000000	0,000000	0,000000		0,000125	0,000000
1440	0,000000	0,000000	0,000000	0,000000	0,000125		0,000000
2160	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	

(a)

Závislá: PC MYRIAD	**240** R:1643,6	360 R:1600,1	480 R:1526,4	720 R:1422,4	1080 R:1302,6	1440 R:1196,4	2160 R:1112,1
240		1,000000	0,848933	0,002289	0,000000	0,000000	0,000000
360	1,000000		1,000000	0,039464	0,000004	0,000000	0,000000
480	0,848933	1,000000		1,000000	0,001894	0,000000	0,000000
720	0,002289	0,039464	1,000000		0,759162	0,001627	0,000001
1080	0,000000	0,000004	0,001894	0,759162		1,000000	0,018106
1440	0,000000	0,000000	0,000000	0,001627	1,000000		1,000000
2160	0,000000	0,000000	0,000000	0,000001	0,018106	1,000000	

(b)

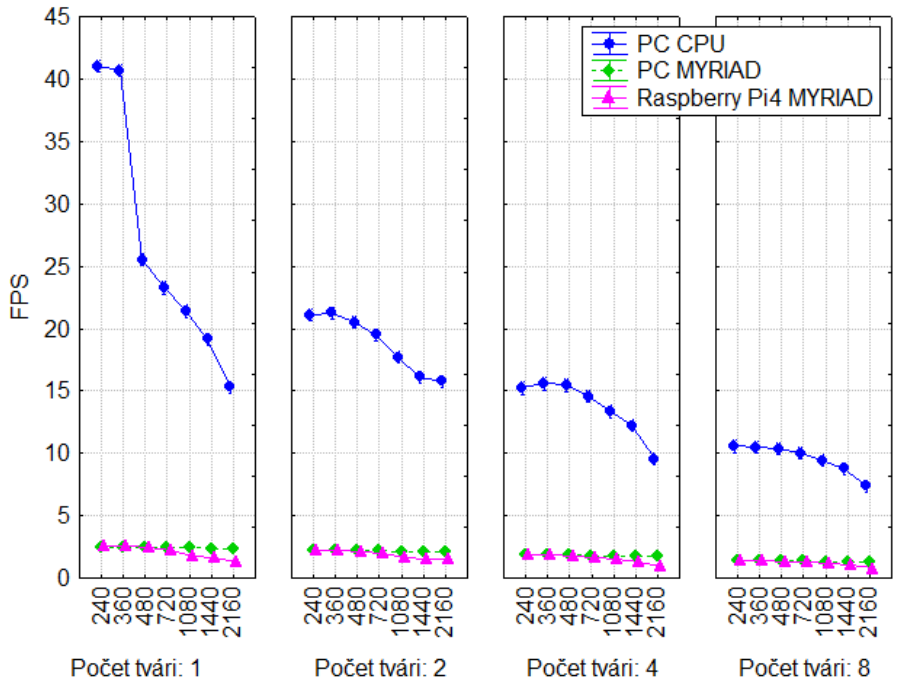
Závislá: PC CPU	**240** R:1696,4	360 R:1712,9	480 R:1615,2	720 R:1500,1	1080 R:1324,3	1440 R:1132,4	2160 R:822,16
240		1,000000	1,000000	0,012478	0,000000	0,000000	0,000000
360	1,000000		1,000000	0,004141	0,000000	0,000000	0,000000
480	1,000000	1,000000		0,923583	0,000008	0,000000	0,000000
720	0,012478	0,004141	0,923583		0,044302	0,000000	0,000000
1080	0,000000	0,000000	0,000008	0,044302		0,016449	0,000000
1440	0,000000	0,000000	0,000000	0,000000	0,016449		0,000001
2160	0,000000	0,000000	0,000000	0,000000	0,000000	0,000001	

(c)

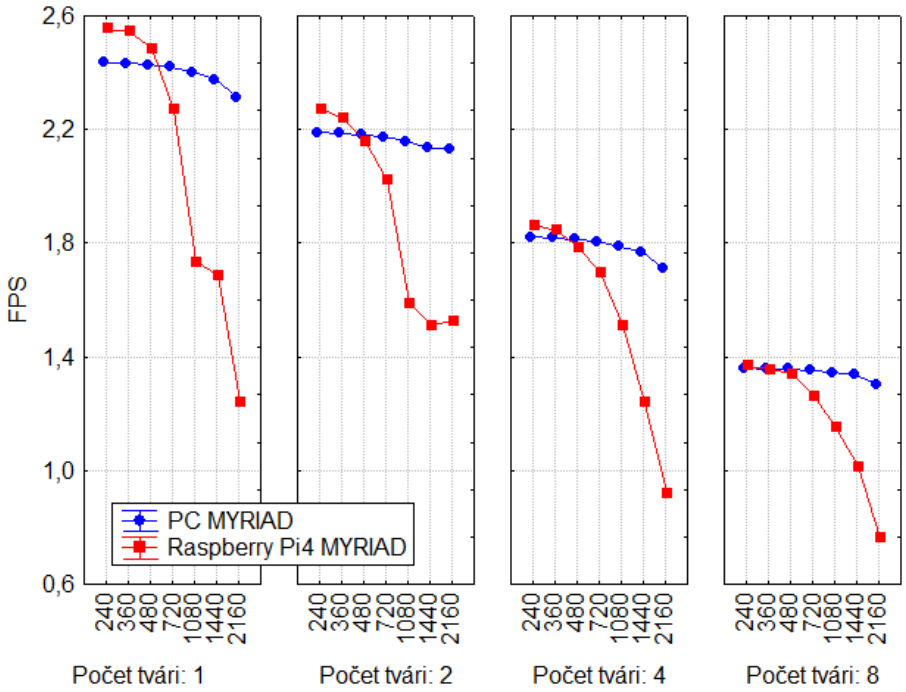
Obr. 5.19: Mnohonásobné porovnávanie priemerného poradia pre všetky skupiny podľa rozlíšenia snímok (triediaca premenná) na skúmaných zariadeniach (PC CPU, PC MYRIAD a Raspberry Pi4 MYRIAD). P-hodnota nižšia ako 0,05 (červené čísla) znamená významný rozdiel medzi dvomi skupinami.

ničných funkcii OpenCV. Možným dôvodom poklesu výkonu môže byť narastajúca zložitosť týchto úprav. Ďalšou možnou príčinou môže byť čítanie a dekódovanie videa s vysokým rozlíšením. Pravdepodobnejšia je však predchádzajúca hypotéza, hlavne s dôvodu že ten istý jav je v obmedzenej miere pozorovaný aj na PC CPU platforme. To by mohlo vysvetliť prečo PC MYRIAD nieje ovplyvnený týmto javom. Aplikácia beží na oveľa nižších FPS ako je schopny PC CPU zvládať.

Toto odhaľuje výkonnostné limity pri použití akcelerátora. Na záver treba dodať, že kvôli chybe v OpenVino frameworku a jeho spúšťaniu na ARM platforme nebolo možné spustiť test s menej náročnou sieťou na detekciu tváre (nesprávna interpretácia niektorých vrstiev konvertovanej CNN siete). Test bol bolo možné spustiť pre PC MYRIAD a aplikácia dosahovala približne 4x vyšších výstupov (okolie 8 - 9 FPS). Táto chyba je známa a v čase odovzdávania BP ešte nevyriešená.



(a) Porovnanie počtu FPS podľa kvality rozlíšenia snímok (240p, 360p, 480p, 720p, 1080p, 1440p, 2160p) na skúmaných zariadeniach (PC CPU, PC MYRIAD a Raspberry Pi4 MYRIAD). N=300, Priemer; Svorka: Priemer ±0,95 Interval spoľahlivosti.



(b) Porovnanie počtu FPS podľa kvality rozlíšenia snímok (240p, 360p, 480p, 720p, 1080p, 1440p, 2160p) na zariadeniach PC MYRIAD a Raspberry Pi4 MYRIAD. Údaje ako na Obr. 5.20a, len bez PC CPU kvôli lepšej vizualizácií. N=300, Priemer; Svorka: Priemer ±0,95 Interval spoľahlivosti.

Obr. 5.20

Závěr

Po oboznámení sa s problematikou a možnosťami akcelerácia bol vybraný akcelerátor Intel Movidius NCS a bola navrhnutá aplikácia pre jeho využitie. Aplikácia riešila problém identifikácie tváre. Je bežné že jedna CNN rieši jeden špecifický problém, preto pri implementácii komplexných aplikácii sa často využívajú viaceré siete. Z tohto dôvodu sa tento problém v návrhu rozdelil na pod problémy lokalizácie tváre v obraze, identifikácie kľúčových bodov na tvári a následné priradenie identity. V aplikácii sa využíva pre každý problém iná konvolučná neurónová sieť. Spolu tvoria dátovú pipelinu, využitím ktorej sa podarilo úspešne odhalila identitu osoby. Pri implementácii využili už pretrénované siete z depozitára open_model_zoo knižnice OpenvVino. Aplikácia bola implementovaná v súlade s vytýčenými cieľmi a spúšťaná na platforme Raspberry Pi aj PC, testovaná s aj bez akceleračného zariadenia. Okrem aplikácie bol navrhnutý a implementovaný test zameraný na izolované vyhodnotenie efektivity akcelerátora. Tento experiment sa zameral na izolovaný výkon a spotrebu pri výpočte inferencie. Test sa vykonal dokopy pre á rôzdných hw nastavený a 22 obtiažnosti testovanej inferencie. V poslednej časti sa vykonali experimenty ktoré preverili schopnosti jednotlivých platforiem. Následne boli výsledky meraní štatisticky vyhodnotené. Pri porovnávaní s ohľadom na spotrebu sa potvrdila jasná dominancia akcelerátora. V teste na rýchlosť, kde sa pre jednotlivé hw zariadenia spúšťala inferencia neurónovej siete MobileNetV2 s pôdnymi úrovňami MAC, síce akcelerátor nedosiahol najlepšie výsledku spomedzi porovnávaných zariadení, jeho výkon bol význame lepši na všetkých platformách v porovnaní s Raspberri Pi bez neho. Potvrdilo sa nám, že akcelerátor NCS má význam pri akcentovaní výpočtu na re platformu Raspberry Pi a jej podobné.

Literatúra

[1] *McCulloch W.S. Pitts W.: A logical calculus of the ideas immanent in nervous activity.* Bulletin of Mathematical Biophysics 5, 115–133 (1943)

[2] *704 electronic dataprocessing machine* International Business Machines Corporation, IBM Dostupné z URL: <http://www.bitsavers.org/pdf/ibm/704/24-6661-2_704_Manual_1955.pdf>.

[3] *978-3-642-70911-1 Frank Rosenblatt: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Springer, Berlin, Heidelberg, 1986.

[4] *Griewank Andreas.: Who invented the reverse mode of differentiation* strany 389–400. Documenta Mathematica Extra Volume ISMP, 2012

[5] Dostupné z Url: <https://www.machinecurve.com/index.php/2019/06/11/why-you-shouldnt-use-a-linear-activation-function/#the-problem-with-linear-activation-functions>.

[6] *ISBN 80-247-0512-5 TROJAN, Stanislav. Lékařská fyziologie.* Vyd. 4., přeprac. a dopl. Praha: Grada Publishing, 2003.

[7] *Daniel, G., 2013.Principles of artificial neural networks* (Vol. 7). World Scientific.

[8] *Wang, Yingying, et al.: The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition.* Applied Sciences 10.5 (2020): 1897.

[9] *Aurlien Gron.: Hands-On Machine Learning with Scikit-Learn and TensorFlow:* Concepts, Tools, and Techniques to Build Intelligent Systems (1st. ed.). O'Reilly Media, Inc. 2017.

[10] *Simard, P.Y., Steinkraus, D. and Platt, J.C., Best practices for convolutional neural networks applied to visual document analysis.* In Icdar (Vol. 3, No. 2003). 2003, August.

[11] *Matthew Stewart, PhD: Introduction to Convolutional Neural Networks* Towardsdatascience, cit 27.7.2020 [online] Dostupné z URL: <https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac>

[12] Valerii Filipets: *Business applications of Convolutional Neural Networks* The app solutions, cit 27.7.2020 [online]. Dostupné z URL: <https://theappsolutions.com/blog/development/convolutional-neural-networks/>

[13] *artificiallyintelligentclaire.com : 27 Unexpected Innovative Uses of Deep Neural Networks* cit 27.7.2020 [online]. Dostupné z URL: <https://www.artificiallyintelligentclaire.com/deep-neural-networks/>

[14] *Olah, Chris and Mordvintsev, Alexander and Schubert, Ludwig: Feature Visualization* Journal: Distill, 2017 Dostupné z URL: <https://distill.pub/2017/feature-visualization>

[15] *C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich.: Going deeper with convolutions* Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9. 2015.

[16] *J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei.: Imagenet: A large-scale hierarchical image database* Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 248–255. 2009. DOI: 10.1109/cvprw.2009.5206848

[17] *Cope, Ben. Ïmplementation of 2D Convolution on FPGA, GPU and CPU."* Imperial College Report (2006): 2-5.

[18] *Irhum Shafkat: Understanding Convolutions for Deep Learning* Towards Data Science, cit 27.7.2020 [online]. Dostupné z URL: <https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1>

[19] *Denny Britz: Understanding Convolutional Neural Networks for NLP* WILDML, platné od November 7, 2015 cit 27.7.2020 [online]. Dostupné z URL: <http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/>

[20] *CS231n Convolutional Neural Networks for Visual Recognition* prednášky Stanford Jar 2020 Dostupné z URL: <https://cs231n.github.io/convolutional-networks/>

[21] *Kamencay, Patrik & Benco, Miroslav & Mizdos, Tomas & Radil, Roman. A New Method for Face Recognition Using Convolutional Neural Network. Advances in Electrical and Electronic Engineering.* (2017) Žilina, Slovensko. 15. 10.15598/aeee.v15i4.2389.

[22] *Ujjwal Karn: Explanation of Convolutional Neural Networks* palstné od August 11, 2016 cit 27.7.2020 [online]. Dostupné z URL: <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>

[23] *Myslivec, V., 1957: Statistické metody zemědělského a lesnického výzkumnictví.* ČSAZV Praha, 555 pp.

[24] *Thadewald, T., Büning, H. (2007):Jarque-Bera Test and its Competi-tors for Testing Normality - A Power Comparison.* Journal of AppliedStatistics34, 87–105.

[25] *BLATNÁ, D., 1996.Neparametrické metody : testy založené na pořádkovýcha pořadových statistikách.* Praha: Vysoká škola ekonomická, ISBN 80-7079-607-3.

[26] *BORŮVKOVÁ, J., P. HORÁČKOVÁ a M. HANÁČEK, 2013.Statistica: úvod do zpracování dat.Jihlava: Vysoká škola polytechnická Jihlava.* ISBN 978-80-87035-79-5.

[27] *Matejka, F., Huzulák, J., 1987: Analýza mikroklímy porastu.* Bratislava, VEDA, Vydavateľstvo SAV, 228 pp.

[28] *Shapiro, S. S., Wilk M. B. 1965. An Analysis of Variance Test ofNormality (Complete Samples).* Biometrika52, 591–611.

[29] *Mark Sandler and Andrew Howard and Menglong Zhu and Andrey Zhmoginov and Liang-Chieh Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks* arXiv, 2018, eprint: 1801.04381 Použité Imagenet Checkpointy dostupné z URL: <https://github.com/tensorflow/models/tree/master/research/slim/nets/mobilenet#mobilenet-v2-imagenet-checkpoints>

[30] *Intel® oficiálna stránka* Dostupné z URL: <https://www.intel.com/content/www/us/en/products/processors/core/i7-processors/i7-8550u.html?wapkw=intelcorei7-8550u

[31] *NOTEBOOK CHECK* Dostupné z URL: <https://www.notebookcheck.net/NVIDIA-GeForce-MX150-Benchmark-and-Specs-of-the-GT-1030-for-Laptops.223530.0.html>

[32] *Sergio Rivas-Gomeza, Antonio J. Pe nab, David Moloneyc, Erwin Laurea, Stefano Markidisa: Exploring the Vision Processing Unit as Co-processor for Inference* KTH Royal Institute of Technology, Barcelona Supercomputing Center (BSC), Intel Ireland Ltd. arXiv:1810.04150v1, 9 Oct 2018

[33] *Power Consumption Benchmarks* Raspberry Pi Dramble Dostupné z URL: <https://www.pidramble.com/wiki/benchmarks/power-consumption>

[34] *Eriko Nurvitadhi a spol. : Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks?* authors: Eriko Nurvitadhi, Ganesh Venkatesh, Jaewoong Sim, Debbie Marr, Randy Huang, Jason Ong Gee Hock, Yeong Tat Liew, Krishnan Srivatsan, Duncan Moss, Suchit Subhaschandra, and Guy Boudoukh. 2017. In Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '17). Association for Computing Machinery, New York, NY, USA, 5–14.

[35] *Neural compute Stick Documentation.* Dostupné z URL: <https://software.intel.com/en-us/movidius-ncs.>

[36] *Google Coral Datasheet* Dostupné z URL: <https://coral.withgoogle.com/tutorials/accelerator-datasheet/.

[37] *Nvidia Jetson Nano Developer Kit* Dostupné z URL: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit

[38] *samsung* Dostupné z URL: <https://www.samsung.com/semiconductor/dram/lpddr3/

[39] *WikiChip* Dostupné z URL: <https://en.wikichip.org/wiki/movidius/microarchitectures/shave_v2.0

[40] *Intel® Movidius™ Neural Compute SDK* Dostupné z URL: <https://movidius.github.io/ncsdk/ncs.html

[41] *TensorFlow Lite guide* Dostupné z URL: <https://www.tensorflow.org/lite/guide

[42] *OpenVINO™ Toolkit* Dostupné z URL: <https://docs.openvinotoolkit.org/latest/openvino_docs_IE_DG_Introduction.html

Zoznam príloh

A Prílohy k štatistickej analýze

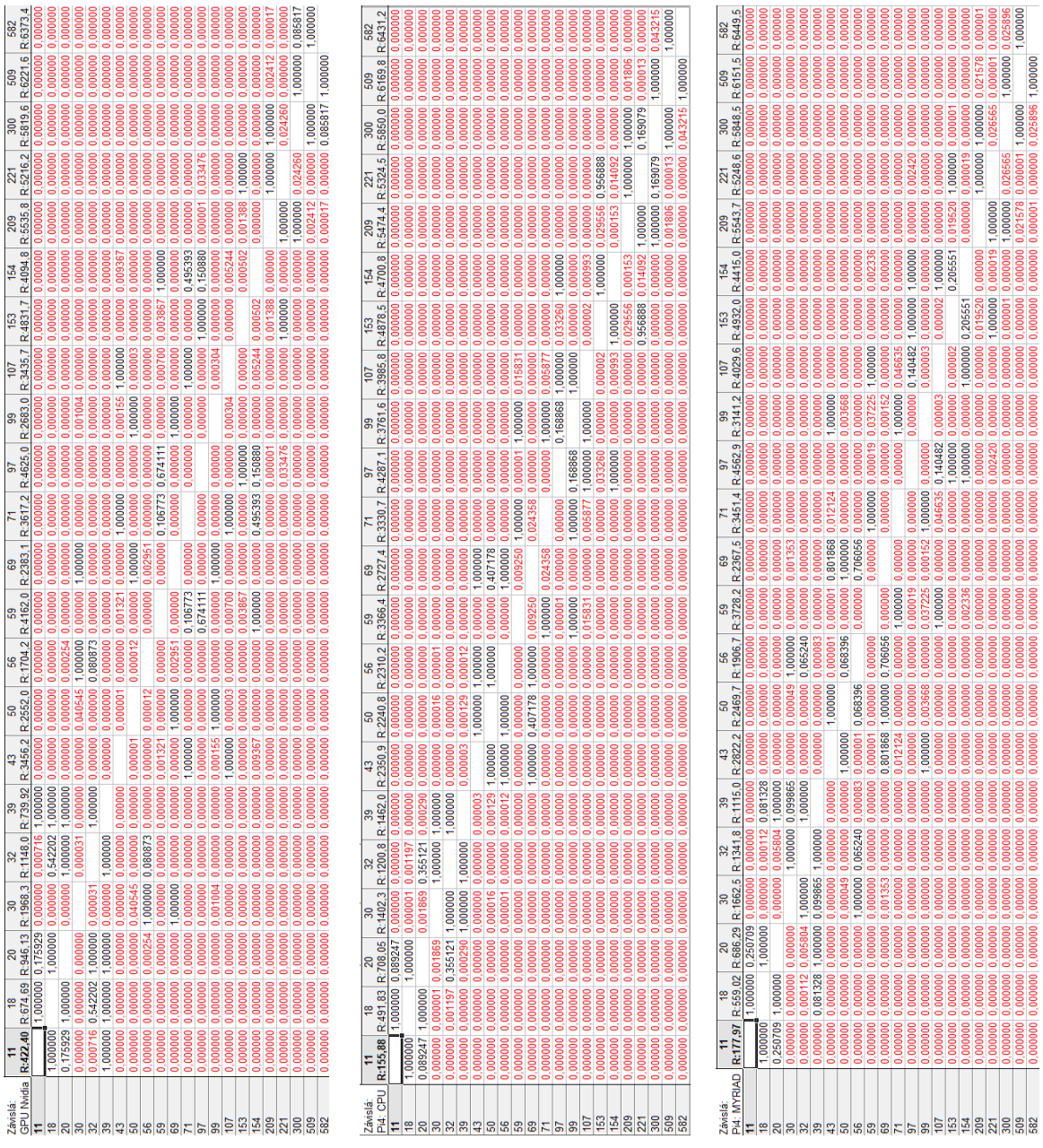
A.1 Výsledky mnohonásobného porovnávania časových inferencií všetkých záťažových úrovní

(a) Raspberry pi 3 - CPU

(b) Raspberry pi 3 - MYRIAD

(c) PC Intel GPU

Obr. A.1: Výsledok mnohonásobného porovnávania spotreby zariadenia všetkých skupín zložitosti siete (11 – 582) (Kruskal- Walisov test). P-hodnota nižšia ako 0,05 (červené čísla) znamená významný rozdiel medzi dvomi kategóriami.



(a) PC Nvidia GPU

(b) Raspberry pi 4 - CPU

(c) Raspberry pi 4 - MYRIAD

Obr. A.2: Výsledok mnohonásobného porovnávania spotreby zariadenia všetkých skupín zložitosti siete (11 – 582) (Kruskal- Walisov test). P-hodnota nižšia ako 0,05 (červené čísla) znamená významný rozdiel medzi dvomi kategóriami.

Závislá: CPU	11 R:173,80	18 R:647,97	20 R:611,65	30 R:1137,8	32 R:1292,6	39 R:1874,7	43 R:1997,8	50 R:2004,4	56 R:3205,0	59 R:2710,1	69 R:3095,5	71 R:3075,0	97 R:3936,3	99 R:4479,7	107 R:4178,0	153 R:5230,8	154 R:5337,4	209 R:5281,6	221 R:4587,4	300 R:5289,9	509 R:6092,6	582 R:6370,8
11		0,532516	1,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
18	0,532516		1,000000	0,378933	0,007890	0,002776	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
20	1,000000	1,000000		0,166211	0,002776	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
30	0,000000	0,378933	0,166211		1,000000	0,000502	0,000007	0,000006	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
32	0,000000	0,007890	0,002776	1,000000		0,042304	0,001347	0,001099	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
39	0,000000	0,000000	0,000000	0,000502	0,042304		1,000000	1,000000	0,000000	0,000018	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
43	0,000000	0,000000	0,000000	0,000007	0,001347	1,000000		1,000000	0,000000	0,001080	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
50	0,000000	0,000000	0,000000	0,000006	0,001099	1,000000	1,000000		0,000000	0,001324	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
56	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000		0,339359	1,000000	1,000000	0,000599	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
59	0,000000	0,000000	0,000000	0,000000	0,000000	0,000018	0,001080	0,001324	0,339359		1,000000	1,000000	0,000015	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
69	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	1,000000		1,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
71	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	1,000000	1,000000		0,000007	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
97	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000599	0,000015	0,000007			0,110410	1,000000	0,000319	0,000008	0,000059	0,006575	0,000044	0,000000	0,000000
99	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,110410		1,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
107	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	1,000000		0,000000	1,000000	1,000000	1,000000	0,000000	0,000000	0,000000
153	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000319	0,000000	0,000000		1,000000	1,000000	1,000000	0,008186	1,000000	0,000007
154	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000008	0,000000	1,000000	1,000000		1,000000	0,000330	1,000000	0,000279	0,000000
209	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000059	0,000000	1,000000	1,000000	1,000000		0,001879	1,000000	0,000043	0,000000
221	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,006575	1,000000	1,000000	0,008186	0,000330	0,001879		0,001463	0,000000	0,000000
300	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000044	1,000000	1,000000	1,000000	1,000000	1,000000	0,001463		0,000057	0,000000
509	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000007	0,000279	0,000043	0,000000	0,000057		1,000000
582	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	

(a) PC CPU

Závislá: PC MYRIAD	11 R:150,50	18 R:450,50	20 R:776,98	30 R:1952,3	32 R:1616,5	39 R:1026,0	43 R:3148,6	50 R:2567,6	56 R:1382,5	59 R:4217,1	69 R:2248,8	71 R:3723,8	97 R:4852,1	99 R:2835,1	107 R:3477,2	153 R:4748,9	154 R:4183,9	209 R:5550,5	221 R:5250,5	300 R:5850,5	509 R:6150,5	582 R:6450,5
11		1,000000	0,013057	0,000000	0,000000	0,000004	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
18	1,000000		1,000000	0,000000	0,000000	0,049943	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
20	1,000000	1,000000		0,000000	0,000016	1,000000	0,000000	0,000000	0,022978	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
30	0,013057	0,000000	0,000000		1,000000	0,000001	0,000000	0,017694	0,057562	0,000000	1,000000	0,000000	0,000000	0,000003	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
32	0,000000	0,000000	0,000016	1,000000		0,034007	0,000000	0,000000	1,000000	0,000000	0,011157	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
39	0,000004	0,049943	1,000000	0,000001	0,034007		0,000000	0,000000	1,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
43	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000		0,043412	0,000000	0,000000	0,000002	0,050377	0,000000	1,000000	1,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
50	0,000000	0,000000	0,000000	0,000000	0,017694	0,000000	0,043412		0,000000	0,000000	1,000000	0,000000	0,000000	1,000000	0,000001	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
56	0,000000	0,000000	0,022978	0,057562	1,000000	1,000000	0,000000	0,000000		0,000000	0,000006	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
59	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000		0,000000	0,351527	0,010319	0,000000	0,000456	0,145543	1,000000	0,000000	0,000000	0,000000	0,000000	0,000000
69	0,000000	0,000000	0,000000	0,000000	0,111157	0,000000	0,000002	1,000000	0,000006	0,000000		0,000000	0,000000	0,037842	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
71	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,050377	0,000000	0,000000	0,351527	0,000000		0,000000	0,000003	1,000000	0,000000	0,716617	0,000000	0,000000	0,000000	0,000000	0,000000
97	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,010319	0,000000	0,000000		0,000000	1,000000	1,000000	0,004036	0,001653	1,000000	0,000000	0,000000	0,000000
99	0,000000	0,000000	0,000000	0,000000	0,000003	0,000000	1,000000	1,000000	0,000000	0,000000	0,037842	0,000003	0,000000		0,008494	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
107	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	0,000000	0,000001	0,000456	0,000000	1,000000	0,000000	0,008494		0,000000	0,001282	0,000000	0,000000	0,000000	0,000000	0,000000
153	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,145543	0,000000	0,000000	1,000000	0,000000	0,000000		0,065112	0,000059	0,291740	0,000000	0,000000	0,000000
154	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	0,000000	0,716617	0,004036	0,000000	0,001282	0,065112		0,000000	0,000000	0,000000	0,000000	0,000000
209	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,001653	0,000000	0,000000	0,000000	0,000059	0,000000		1,000000	1,000000	0,026553	0,000002
221	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	0,000000	0,000000	0,291740	0,000000	1,000000		0,026553	0,000002	0,000000
300	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000	0,026553		1,000000	0,026553
509	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,026553	0,000002	1,000000		1,000000
582	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,026553	1,000000	

(b) PC MYRIAD

Obr. A.3: Výsledok mnohonásobného porovnávania spotreby zariadenia všetkých skupín zložitosti siete (11 – 582) (Kruskal- Walisov test). P-hodnota nižšia ako 0,05 (červené čísla) znamená významný rozdiel medzi dvomi kategóriami.