



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

**AKCELERACE NEURONOVÉ SÍTĚ PRO DETEKCII OB-
LIČEJE VE ZHORŠENÝCH SVĚTELNÝCH PODMÍNKÁCH**
ACCELERATION OF A NEURAL NETWORK FOR FACE DETECTION IN LOW LIGHT CONDITI-
ONS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VOJTECH ORAVA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ GOLDMANN

BRNO 2023

Abstrakt

Cílem této práce je vytvořit neuronovou síť pro rozpoznávání obličejů ve špatných světelných podmínkách, tuto síť akcelerovat a porovnat s existujícími řešeními. Problém detekce je řešen konvoluční neuronovou sítí natrénovanou s využitím dat z datasetů obličejů a akcelEROvanou akcelerátorem Intel Neural Compute Stick 2. Práce obsahuje summarizaci dosavadních řešení a algoritmů detekce (jak klasických metod, tak těch využívajících neuronové sítě) a poskytuje porovnání těchto řešení.

Abstract

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v anglickém jazyce.

Klíčová slova

Detekce obličeje, akcelerace neuronových sítí, NCS 2, detekce v reálných podmínkách, neuronové sítě, Python, počítačové vidění

Keywords

Face detection, Neural Networks acceleration, NCS 2, Detection in Real World Conditions, Neural Networks, Python, Computer Vision

Citace

ORAVA, Vojtěch. *AKCELERACE NEURONOVÉ SÍTĚ PRO DETEKCI OBLIČEJE VE ZHORŠENÝCH SVĚTELNÝCH PODMÍNKÁCH*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Goldmann

AKCELERACE NEURONOVÉ SÍTĚ PRO DETEKCI OBLIČEJE VE ZHORŠENÝCH SVĚTELNÝCH PODMÍNKÁCH

Prohlášení

Prohlašuji, že jsem tuto semestrální práci vypracoval samostatně pod vedením pana Ing. Tomáše Goldmanna. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Vojtěch Orava
21. ledna 2023

Poděkování

Chtěl bych poděkovat panu Ing. Tomáši Goldmannovi za vedení práce a poskytnuté konzultace.

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 2 |
| 2 | Detekce obličeje v reálných podmínkách | 3 |
| 2.1 | Detekce obličeje | 4 |
| 2.2 | Problémy a omezení | 5 |
| 2.3 | Algoritmy detekce obličeje | 7 |
| 3 | Neuronové sítě pro detekci obličeje | 12 |
| 3.1 | Neuronové sítě | 12 |
| 3.2 | Datasetsy | 15 |
| 3.3 | Detektory obličeje | 17 |
| 3.4 | Frameworky pro neuronové sítě | 22 |
| 4 | Systémy pro detekci obličejů a akcelerace detekce | 23 |
| 4.1 | Kamery | 23 |
| 4.2 | Dostupná řešení | 23 |
| 4.3 | Akcelerace detekčních algoritmů | 24 |
| 5 | Závěr | 27 |
| | Literatura | 28 |

Kapitola 1

Úvod

Neuronové sítě (anglicky neural networks) mají v dnešním světě mnoho využití. Jelikož se jedná o jednu z aplikací umělé inteligence (anglicky artificial intelligence), lze neuronové sítě použít například k rozpoznávání řeči, zpracování přirozeného jazyka či k detekci obličejů. Tyto činnosti jsou pro běžného člověka poměrně snadné, avšak pro počítače znamenají relativně náročnou činnost.

Lidé jsou schopni velmi dobře rozeznat obličeje jiných lidí, bez ohledu na světlé podmínky, úhel natočení či částečné zakrytí tváře. Počítačové algoritmy a neuronové sítě zaměřující se na detekci obličejů v těchto neideálních podmínkách musí být těmto jevům přizpůsobeny.

Aby byly počítače schopné tyto akce vykonávat v rozumném čase (případně v reálném čase), je potřeba, aby neuronové sítě byly dostatečně rychlé. Zrychlení neuronové sítě lze dosáhnout buď optimalizací kódu, vylepšením procesu trénování neuronové sítě nebo také využitím speciálních hardwarových zařízení.

Tato práce se zabývá problematikou výše zmíněných fenoménů, konkrétně optimalizací a zrychlením neuronových sítí pro detekci obličejů v neoptimálních světelnychých podmínkách. Z oblasti neuronových sítí byly použity konvoluční neuronové sítě, k jejichž trénování je využito jak dat z běžných datasetů tváří, tak dat ze specializovaných datasetů obličejů a osob ve špatném osvětlení.

O akceleraci neuronové sítě se stará zařízení Intel Neural Compute Stick 2 (NCS2) s knihovnou OpenVINO. V práci jsou také zmíněny dostupné komerční a nekomerční systémy a řešení pro detekci obličejů.

V rámci kapitoly 2 je popsána problematika detekce obličeje v reálných podmínkách, včetně problémů, které detekci ztěžují či přímo znemožňují. Tato část také popisuje klasické přístupy k detekci obličejů jako jsou algoritmy Viola–Jones nebo Local Binary Patterns.

Kapitola 3 se věnuje neuronovým sítím jak obecně (perceptron, aktivační funkce, učení), tak konkrétně konvolučním neuronovým sítím. Dále jsou v kapitole popsány datasety pro učení neuronových sítí a existující řešení detekce obličeje založené na neuronových sítích (YOLO, MTCNN, SSD), včetně jejich porovnání a porovnání algoritmů zaměřených na detekci ve špatných světelnychých podmínkách.

Popis používaných řešení a systémů k detekci a popis akcelerace detekčních algoritmů tvoří obsah kapitoly 4. Konkrétně je zde popsáno zařízení NCS2.

Kapitola 2

Detekce obličeje v reálných podmírkách

Detekce obličeje (anglicky face detection) [23, 34] je technologie, která umožňuje v digitálním obrázku lokalizovat lidský obličej. Detekce obličeje patří do skupiny technologií HCI (Human–Computer interaction). Detektovat obličeje je poměrně jednoduchý úkol pro lidi, ale zároveň se jedná o relativně náročný úkol pro počítače. Detekce obličeje je výchozím bodem pro další algoritmy analyzující lidský obličej, jako je například (v závorce za pojmem následuje anglický výraz):

- rozpoznávání obličeje (face recognition),
- zarovnání obličeje (face alignment),
- autentizace pomocí obličeje (face verification/authentication),
- sledování pohybu hlavy (head pose tracking),
- určování věku nebo pohlaví (age/gender recognition),

a mnoho dalších.

Samotná detekce obličeje se v realním prostředí využívá například v oblasti fotografování (automatické ostření na tvář), marketingu (zjišťování zájmu zákazníku o produkty podle počtu výskytu obličejů) nebo bezpečnosti (bezpečnostní kamery a systémy).

Následující podkapitoly se zabývají principem fungování detekce obličeje v reálných podmírkách a problémy a omezeními, které se v běžném světě vyskytují a detekce by si s nimi měla umět poradit (špatné světelné podmínky, příliš členité pozadí, přílišný počet obličejů v obrázku, barva kůže, nízké rozlišení atd.). Na konci této kapitoly se nachází popis algoritmů a detektorů, které k detekci přímým způsobem nepoužívají neuronové sítě.



Obrázek 2.1: Příklad detekce obličeje

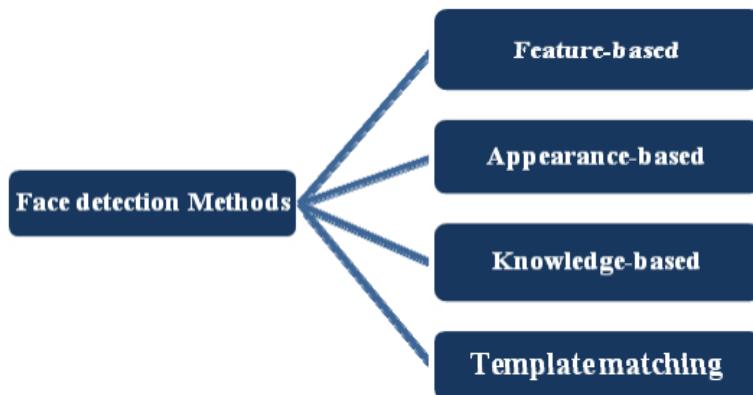
2.1 Detekce obličeje

Způsoby detekce obličeje lze rozdělit do několika skupin, vědecké práce na toto téma se liší a nelze jasně říci zda to či ono dělení je jediné korektní.

Dle [23] existují 2 různé přístupy k hledání tváří v obrázcích, a to **přístup založený na vlastnostech** (anglicky feature based approach) a **přístup založený na obrázku** (anglicky image based approach). **Přístup založený na vlastnostech** nepoužívá přímo k detekci obličeje neuronové síť. Využívá vlastnosti obličeje jako takového (rysy, pozice očí, usí, obočí, barva kůže...). Efektivita tohoto přístupu se snižuje s výskyty problémů popsaných v sekci 2.2, protože může docházet například k zakrytí nebo špatné viditelnosti některých vlastností obličeje.

Naproti tomu **obrazový přístup** uplatňuje schopnosti neuronových sítí a umělé inteligence k natrénování modelu neuronové sítě a následné přímé detekci pomocí tohoto modelu.

Podle [35] lze rozdělit metody detekce obličeje do 4 základních kategorií (viz obrázek 2.2) a 2 zvláštních kategorií (Haarovy vlastnosti a umělá inteligence).



Obrázek 2.2: Dělení metod detekce obličeje dle [35]

Feature-based methods (metody založené na vlastnostech) opět pracují s vlastnostmi obličeje. Vyhledávají v obraze rysy obličeje. Detekci může výrazně ztížit či znemožnit nevi-

ditelnost některých rysů. Výhodou těchto metod je rychlosť v porovnání s ostatními metodami.

Appearance-based methods (metody založené na vzhledu/obrázku) využívají klasifikace vlastností tváře do 2 tříd v podobrázku celého obrázku. Klasifikátory podle nichž se daná metoda rozhoduje, zda se jedná o tvář či nikoli, mají různé váhy, metoda postupuje od slabých klasifikátorů k silnějším.

Metody založené na znalostech (**Knowledge-based methods**) se uplatňují při detekci obličeje v obrázku s členitým/komplexním pozadím. Znalosti, které k detekci pomáhají jsou například: obličeji má 2 uši, jeden nos, jedny ústa nebo známé vzdálenosti mezi jednotlivými rysy obličeje.

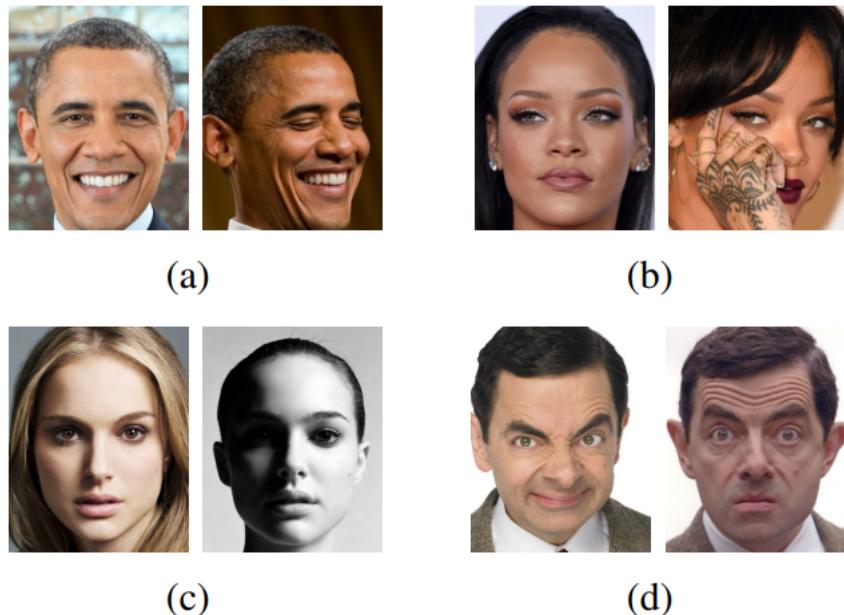
Templatematching (šablonování, maskování) aplikuje na obrázek předem danou masku obličeje a snaží se detektovat obličeji pomocí postupného maskování. Tato metoda je snadná na implementaci, jejím nedostatkem je však závislost na přímém pohledu tváře na obrázku.

Haarovými vlastnostmi a neuronovými sítěmi se zabývají sekce 2.3, respektive 3.1.

2.2 Problémy a omezení

Algoritmy pro detekci obličeji čelí několika výzvám a omezením spojených s ne vždy perfektním zobrazením obličeje v obrázku. Lidské obličeje na fotografiích a obrázcích mohou být částečně zakryté (např. sluneční brýle), fotografie mohou být pořízené za nevhodných světelných podmínek (např. zastínění části tváře) nebo mohou obrázky nabývat nedostatečné kvality (nízké rozlišení).

Jelikož tedy vstupní obrázek detekce obličeje nemusí být vždy ideální, nemusí být obličeji vždy správně detekován. V této sekci jsou popsány některé problémy [35, 23], které mohou bránit v úspěšné detekci. Minimalizace dopadu těchto jevů na detekci je klíčem k navýšení uspěšnosti detekce. Mezi problémy a omezení (viz obrázek 2.3) pro detekci patří pozice hlavy, zakrytí části/částí obličeje, špatně osvětlená scéna nebo výraz tváře.



Obrázek 2.3: Vybrané problémy při detekci obličejů [34]. (a) Pozice hlavy; (b) Zakrytí části obličeje; (c) Špatné světelné podmínky; (d) Výraz tváře

Pozice hlavy

Hlava může být na fotografii různě natočena, takže obličeje nemusí být zachycen v přímém pohledu do kamery, ale může být zaznamenán z profilu nebo ze šikma (poloprofil) jako na obrázku 2.3 část (a).

Zakrytí části obličeje

Výsledek detekce může být ovlivněn i zakrytím části obličeje (rukou, brýlemi, vlasy, šátkem apod.).

Výraz tváře

Výraz lidského obličeje mohou ovlivnit emoce a nálady. Detektor [27] zabývající se detekcí a rozpoznáváním emocí dosahuje přesnosti 96 %.

Orientace obrázku

Problémem pro detekci může být různá orientace obrázku (vzhůru nohama, zrcadlově otočený, natočený do strany apod.). Na obrázek je tak nutno aplikovat některé transformační operace pro zarovnání.

Nedostatečně výkonná detekce

Velmi důležitým faktorem při detekci obličejů, zvláště v real-timových aplikacích, je rychlosť detekce. Pokud má algoritmu vysokou přesnost, ale je příliš pomalý pro vybranou aplikaci, stává se nepoužitelným. Akcelerací detekčních algoritmů se mj. zabývá i tato práce.

Příliš členité pozadí

Pokud se v obrázku nachází příliš mnoho objektů, může dojít ke snížení přesnosti a rychlosti detekce.

Přílišný počet obličejů v obrázku

Výskyt velkého počtu obličejů v jednom obrázku, často překrývajících se, může představovat výzvu pro detekční algoritmus.

Nízké rozlišení

Obrázky a fotografie s nízkým rozlišením nemusejí obsahovat dostatek informace nutné ke správnemu detekování tváře.

Špatné světelné podmínky

Na detekci mohou mít vliv světelné podmínky panující při pořizování zkoumané fotografie či videa. Aspekty jež světlo ovlivňuje jsou mj. jas, kontrast, barvy, stíny, ostrost. Tato práce se zabývá detekcí obličejů v záznamech, v nichž některý z těchto faktorů omezuje detekci.



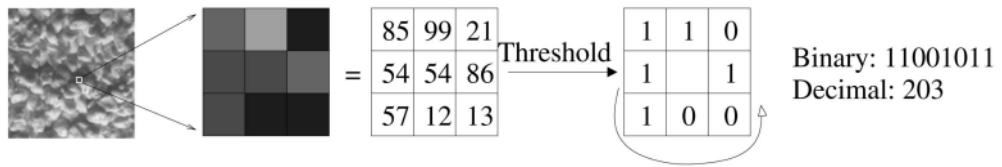
Obrázek 2.4: Příklad stejného obličeje vyfoceného při různých světelných podmínkách [35]

2.3 Algoritmy detekce obličeje

Existuje několik různých přístupů k detekci obličeje. Tato sekce se zaměřuje na detekci s využitím detektorů založených primárně ne na neuronových sítích (neuronová síť není použita vůbec, nebo není použita k přímé detekci). Detektory obličeje využívající principy neuronových sítí k přímé detekci jsou popsány v sekci 3.3.

Local Binary Patterns

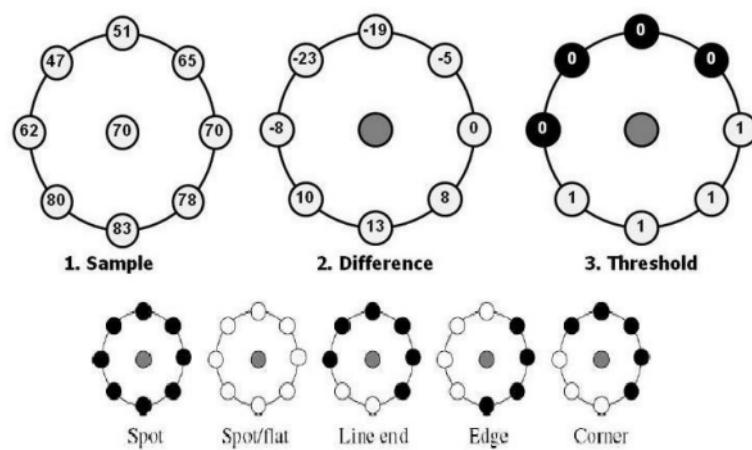
Algoritmus využívající **lokální binární vzory** (anglicky Local Binary Patterns – dále jen LBP) pro popis struktury/textury obrázku má mnoho aplikací [3]. Jedná se o jeden z nejvýkonnějších algoritmů pro popis textur v obrázcích. LBP algoritmus má vysokou účinnost detekce (89 %) [21] a nízkou výpočetní náročnost. LBP pracuje s černobílými (anglicky gray-scale) obrázky a v původní verzi funguje tak, že každému pixelu přiřadí binární číselnou hodnotu, vypočítanou dle hodnot pixelů v 3×3 okolí daného pixelu. Každý takovýto pixel v okolí je ohodnocen hodnotou buď 1 nebo 0 v závislosti na tom, zda jeho hodnota překročila stanovený práh (anglicky threshold), kterým je hodnota prostředního pixelu (viz obrázek 2.5).



Obrázek 2.5: Ukázka ohodnocení pixelu algoritmem LBP [3]

Algoritmus byl později vylepšen tak, aby uměl pracovat s různě velkým okolím, které navíc nemusí mít čtvercový tvar. Používané okolí ve tvaru kružnice je popsáno dvojicí (P, R) , kde P je počet vzorkovacích bodů a R je poloměr kružnice. Další vylepšení LBP algoritmu se týkalo definování tzv. uniformních vzorů (anglicky uniform patterns). Tyto vzory jsou ty vzory v nichž se vyskytuje nejvýše 2 přechody z 1 na 0 a opačně. Příkladem uniformního vzoru na okolí $(8, 2)$ je 11100000 (1 přechod), či 00111000 (2 přechody).

Lokální primitiva (obrázek 2.6) a histogramy vytvořené z takto získaných hodnot se využívají mj. k detekci obličejů.

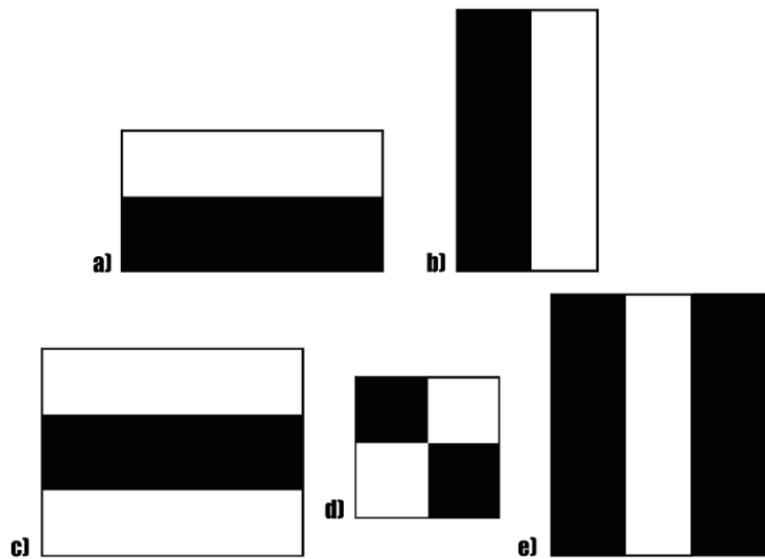


Obrázek 2.6: Lokální primitiva detekované algoritmem LBP [21]

Viola–Jones algoritmus

Algoritmus **Viola–Jones** [7, 40], někdy také nazývaný **Haar Cascades**, je obecně technika pro detekci objektů, vytvořená před používáním metod hlubokého učení. Používá se k detekci obličejů, částí těla, očí, úst atd. Dosahuje přesnosti detekce kolem 90 % [40].

Princip fungování algoritmu spočívá v detekci hran a čár (obecně vlastností) v černobílém obrázku (hodnoty pixelů jsou na intervalu $< 0; 1 >$). Vybere se jedna z vlastností (některé z nich zobrazuje obrázek 2.7) a vypočítá se průměrná hodnota pixelů ve všech obdélnících (obdélníky jsou dva, tři nebo čtyři). Rozdíl těchto hodnot pak určuje zda se jedná o hranu, čáru tzn. zda je daná vlastnost detekována. Pokud například budeme hledat vlastnost b z obrázku 2.7 a vypočtený rozdíl hodnot je blízký 1, detekce byla úspěšná (viz ukázka v obrázku 2.8).



Obrázek 2.7: Haarovy vlastnosti [7] a) horizontální hrana; b) horizontální hrana; c) horizontální čára d) diagonální e) vertikální čára

| | | | | | |
|-----|-----|-----|-----|-----|-----|
| 0.1 | 0.2 | 0.2 | 0.4 | 0.2 | 0.4 |
| 0.1 | 0.8 | 0.7 | 0 | 0.1 | 1 |
| 0.3 | 1 | 0.6 | 0.2 | 0 | 0.4 |
| 0.5 | 0.7 | 0.9 | 0.1 | 0 | 0.5 |
| 0.7 | 1 | 1 | 0.1 | 0.3 | 0.6 |
| 0.2 | 0.3 | 0.7 | 0.4 | 0.5 | 1 |

Suma pixelů v černém obdélníku =
 $0.8 + 0.7 + 1 + 0.6 + 0.7 + 0.9 + 1 + 1 = 6.7$

 Suma pixelů v červeném obdélníku =
 $0 + 0.1 + 0.2 + 0 + 0.1 + 0 + 0.1 + 0.3 = 0.8$

 Průměrná hodnota v černém obdélníku =
 $6.7 / 8 = 0.8375$

 Průměrná hodnota v červeném obdélníku =
 $0.8 / 8 = 0.1$

 $0.8375 - 0.1 = 0.7375$
DETEKOVÁNA HRANA

Obrázek 2.8: Příklad výpočtu detekce hrany dle vlastnosti b) z obrázku 2.7. Bílý obdélník je nahrazen červeným pro lepší viditelnost.

Algoritmus postupně prochází celý obrázek a hledá výskyt některé z vlastností. Toto procházení u obrázků s velkým počtem pixelů znamená enormní výpočetní nároky, protože je potřeba vždy počítat s hodnotami všech dotčených pixelů. Proto Viola a Jones [40] navrli vylepšení nazvané anglicky **Integral Image**. To spočívá v tom, že všechny pixely stačí projít pouze 1x, a každý tento pixel lze ohodnotit sumou hodnot pixelů směrem nalevo a nahoru (důsledkem je, že pixel s nejnižším ohodnocením se nachází v levém horním rohu a pixel s nejvyšším ohodnocením v pravém dolním rohu). Následný výpočet průměrné hodnoty ukazuje obrázek 2.9.

| | | | | | | | |
|-----|-----|------|------|------|------|------|------|
| 0.4 | 1.1 | 2.0 | 2.7 | 3.1 | 3.6 | 4.6 | 4.9 |
| 0.7 | 2.4 | 3.8 | 5.3 | 6.4 | 7.3 | 8.4 | 9.1 |
| 1.6 | 3.7 | 5.2 | 6.9 | 8.5 | 10.2 | 11.5 | 13.1 |
| 1.9 | 4.6 | 6.9 | 9.6 | 11.5 | 13.9 | 15.7 | 17.6 |
| 2.1 | 5.7 | 8.1 | 11.3 | 13.3 | 16.1 | 18.7 | 21.4 |
| 2.6 | 6.3 | 9.0 | 12.9 | 15.6 | 19.2 | 22.8 | 25.7 |
| 3.4 | 7.5 | 11.2 | 15.2 | 18.9 | 22.6 | 26.3 | 29.6 |
| 3.8 | 8.8 | 13.1 | 17.8 | 21.6 | 26.3 | 30.5 | 34.7 |

SUM OF THE PIXELS IN DARK AREA/NUMBER OF PIXELS
= $(26.3 - 15.3 \cdot 4.6 + 2.7) / 18 = 9.1 / 18 = 0.51$

Obrázek 2.9: Ukázka výpočtu hodnoty obdélníku v Haarově vlastnosti za pomocí vylepšení **Integral Image** [7].

Jelikož Haarových vlastností je velké množství, bylo vybráno 6000 nejvíce vyhovujících, které se k detekci obličejů používají. Detekce je rozdělena na několik etap, v každé etapě je vyhledáván v části obrázku výskyt několika vlastností (počet s každou etapou roste), pokud se vyhledání nezdaří, není již dále daná část obrázku prohledávána.

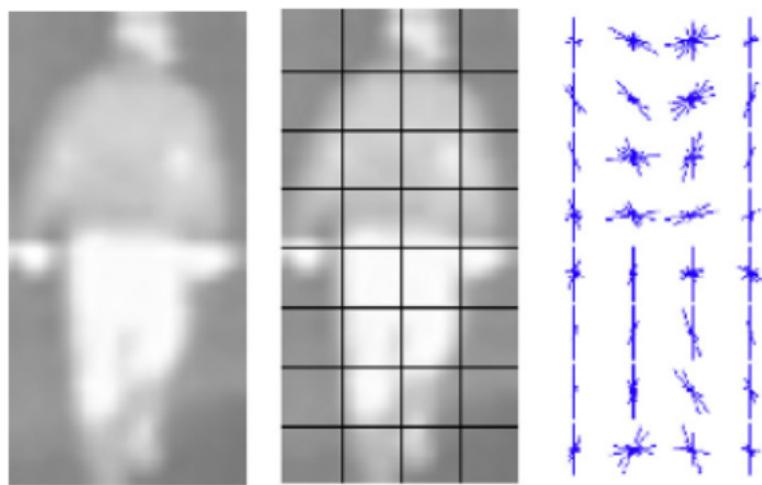
Histogram orientací gradientů

Metoda HOG (anglicky Histograms of Oriented Gradients) [12, 31] používá k detekci obličejů či postav histogramy orientovaných přechodů v obrázku, rozděleném dle mrázky na několik bloků (často například 8×8 nebo 4×4 pixelů). Pro každý pixel v takovémto bloku jsou vypočítány hodnoty gradientů – velikost (magnitude) a směr (direction). Tyto hodnoty jsou pak přiřazeny do jednoho či více sloupců v histogramu popisujícím celý blok.

Tento histogram bývá rozdelen na 9 sloupců (rozmezí – anglicky bin) vyjadřujících směr gradientu v úhlu na škále od 0° do 180° , kdy každý sloupec odpovídá intervalu 20° . Výslednému vektoru, který udává velikosti gradientů v jednotlivých úhlech se říká HOG deskriptor.

HOG využívá pro zlepšení detekce při zhoršených světelných podmínkách normalizaci HOG deskriptorů. Tato normalizace je prováděna na vektorech 4 sousedících bloků. Pokud je tedy vektor složen z 9 hodnot, vypočítá se normalizační vektor z $9 \times 4 = 36$ hodnot.

Výsledkem metody je obrázek (viz obrázek 2.10 reprezentovaný orientovanými gradienty, který se využívá k detekci osoby nebo obličeje).



Obrázek 2.10: Zleva výřez originálního obrázku převedený do šedotónové reprezentace, rozdělený obrázek na bloky, výstup metody HOG [31].

Kapitola 3

Neuronové sítě pro detekci obličeje

Tato kapitola popisuje neuronové sítě a jejich využití pro detekci obličejů. Sekce 3.1 se věnuje popisu neuronových sítí obecně, v sekci 3.2 jsou popsány datasety a jejich využití k trénování neuronových sítí. Sekce 3.3 se zabývá konkrétními detektory obličejů s využitím neuronových sítí. Konkrétní programovací frameworky pro práci s neuronovými sítěmi vyobrazuje sekce 3.4.

3.1 Neuronové sítě

Neuronové sítě [16, 8] umožňují nalezení neznámého řešení problému pomocí naučení se z podobných problémů u nichž známe řešení. Tyto umělé sítě jsou inspirovány biologickou nervovou soustavou lidí (lidským mozkem). Síla neuronových sítí se projevuje v úlohách zaměřených na detekci, rozpoznávání (objektů, lidí, obličejů, obecně vzorů – anglicky patterns) a zpracování dat. Existuje řada druhů neuronových sítí (konvoluční, hluboké, dopředné, rekurentní), pro detekci obličejů v obrázcích se nejčastěji používají konvoluční neuronové sítě (anglicky Convolutional Neural Networks – CNN).

Na princip fungování neuronových sítí může být nahlíženo jako na nelineární matematickou funkci, která převádí X vstupů na Y výstupů. Proces transformace vstupních informací na výstup je ovlivňován váhováním hodnot vně sítě. Tyto hodnoty vah jsou určovány při tzv. učení/trénování neuronových sítí. Pro správné natrénování neuronové sítě je potřeba dostatečného počtu vstupních trénovacích dat (obrázků, textů, hodnot) a dostatečně výkonný hardware.

Biologický neuron

Jak již bylo zmíněno, inspirací neuronových sítí je biologická nervová soustava. V lidském mozku se nachází okolo 10^{11} neuronů (elektricky aktivních buněk spracovávajících signály). Vstupem těchto neuronů jsou tzv. dendrity, výstupy pak nazýváme axony. Jednotlivé neurony jsou navzájem propojeny tisíci spojí pojmenovanými synapse. Synapse zajišťují komunikaci mezi neurony.

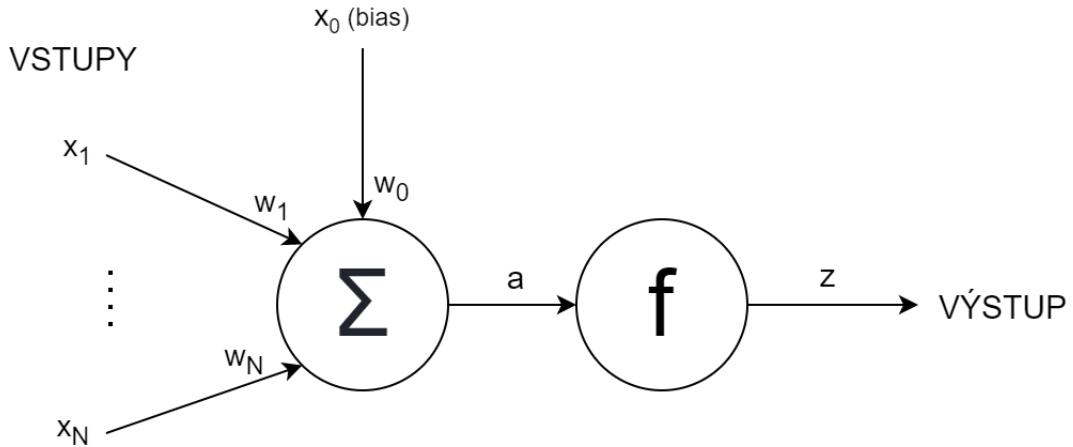
Takto vytvořený paralelismus poskytuje mozku schopnost rychle zpracovávat informace. Neurony biologické i umělé pracují s váhovanými vstupy. Po překročení určitého prahu na vstupech je adekvátně upraven výstup neuronu. Klíčovou vlastností potom je způsobilost měnit hodnoty jednotlivých vah na základě externích vlivů. Tím dochází k učení sítě neuronů.

Perceptron

Nejjednoduším modelem umělého neuronu je perceptron. Perceptron má N vstupů, jejichž hodnoty x jsou vynásobeny váhmi w a sesumovány dle vzorce 3.1.

$$a = \sum_{i=1}^N (w_i * x_i) + w_0 * x_0 \quad (3.1)$$

Hodnota x_0 se nazývá bias a jedná se o neměnnou vstupní hodnotu (často má hodnotu +1). Váhy a vstupy (včetně biasu) mohou být jak kladné, tak i záporné. Hodnota a je po vypočtení předána tzv. **aktivační funkci**.



Obrázek 3.1: Perceptron s N vstupy, biasem x_0 a aktivační funkcí f

Aktivační funkce

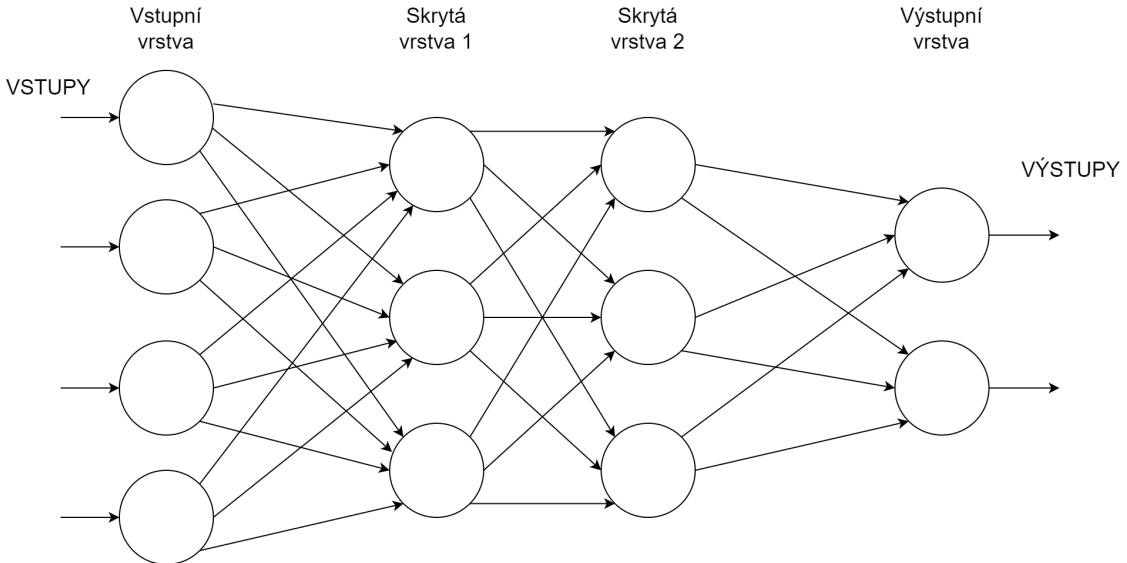
Aktivační funkce f je matematická funkce, která určuje výstup neuronu (vzorec 3.2). Na výběru vhodné aktivační funkce závisí přesnost neuronové sítě. Funkce může být různá, pro příklad je zde uveden výstup rozlišující zda se jedná o kladné či záporné číslo nebo nulu (viz vzorec 3.3). V praxi se často používá aktivační funkce nazvaná **rectified linear unit** (ReLU) [16] definovaná jako $f(a) = \max\{0, a\}$.

$$z = f(a) \quad (3.2)$$

$$f(a) = \begin{cases} 0 & \text{pro } a = 0 \\ 1 & \text{pro } a > 0 \\ -1 & \text{pro } a < 0 \end{cases} \quad (3.3)$$

Spojování neuronů

Spojením několika neuronů lze vytvořit tzv. vrstvu (anglicky layer) perceptronů. Tyto vrstvy se dělí na vstupní (anglicky input), výstupní (anglicky output) a skryté (anglicky hidden). Konkatenací vstupní, několika skrytých a výstupní vrstvy je možno vytvořit neuronovou síť připravenou k trénování.



Obrázek 3.2: Propojení vrstev neuronů do neuronové sítě

Učení

Aby neuronová síť mohla fungovat, musí se natrénoval (tzn. nastavit co nejpřesněji váhy na vstupech neuronů). K trénování se používají data z datasetů (viz sekce 3.2). Trénování neuronových sítí lze rozdělit do 3 kategorií [16]:

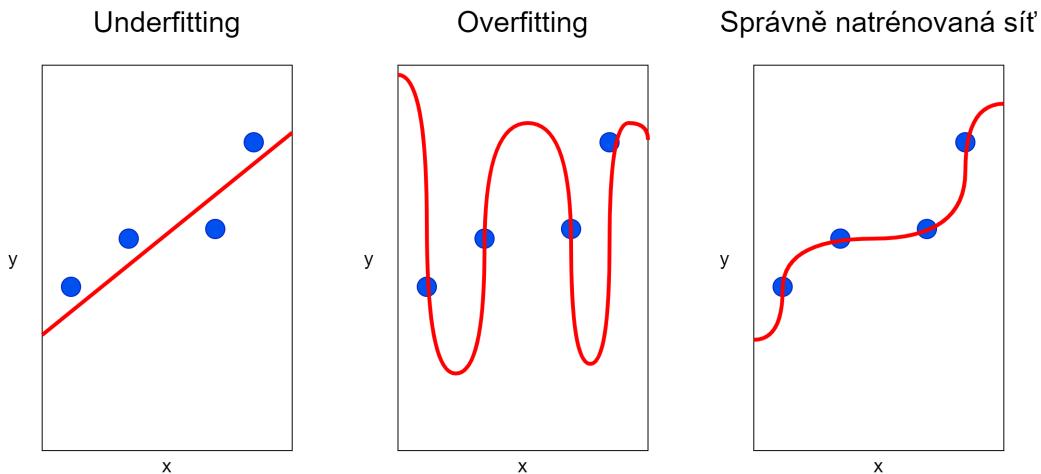
- **Učení bez učitele** (anglicky unsupervised learning) – tyto algoritmy procházejí data z datasetu a provádějí nad nimi shlukování do tzv. clusterů.
- **Učení s učitelem** (anglicky supervised learning) – každým datům z datasetu je přiřazena informace o požadovaném výstupu. Vstupy jsou zpracovány neuronovou sítí a podle chyby (rozdílu vstup/výstup) jsou upraveny parametry v neuronové síti.
- **Posilované učení** (anglicky reinforcement learning) – tento druh učení není vázán pouze na data z datasetu, ale navíc interaguje s prostředím.

Chyby

Abychom byli schopní aktualizovat váhy vstupů jednotlivých neuronů, je potřeba měřit chybovost výstupu neuronové sítě. Jednou z možností měření chyby (využitelné například při lineární regresi) je *mean squared error*. Dle vztahu 3.4 je vypočítána chybovost a jsou upraveny váhy. Snahou je chybu co nejvíce zmenšit.

$$MSE = \frac{1}{N} \sum_i^N (vystupNS - spravnyVystup)^2 \quad (3.4)$$

Při trénování může dojít k situaci, kdy neuronová síť zvládá zpracovávat trénovací data s velmi vysokou přesností, ale při použití testovacích (validačních) dat se chybovost zvyšuje. V takovémto případě mluvíme o **přetrénování** (anglicky overfitting). Druhým nežádoucím jevem, který může nastat je **nedotrénovanost** (anglicky underfitting) – síť není dostatečně natrénovala a dochází tak k vysoké chybovosti (příčinou je například málo trénovacích dat).



Obrázek 3.3: Ukázka výstupu neuronové sítě při nedoučení, přeучení a dostatečně dobrém množství dat pro učení

3.2 Datasetsy

Datasetem rozumíme soubor podobných dat (například obrázků obličejů, číslic, předmětů nebo textů). Pro detekci obličejů se využívají datasetsy obsahující fotografie a videa lidí z veřejně dostupných zdrojů (internet, televize) nebo jsou datasetsy přímo účelně vytvářeny (fotografování lidí) a následně je možné si je koupit nebo stáhnout. Data v datasetsech je nutné tzv. anotovat (označit co na daném obrázku je). V případě tváří se může jednat například o věk osoby, pohlaví, rasu, aby bylo možné určit zda po klasifikaci neuronovou sítí výstup odpovídá požadovanému výsledku. Následující podsekce popisují datasetsy používané k učení neuronových sítí pro detekci obličejů.

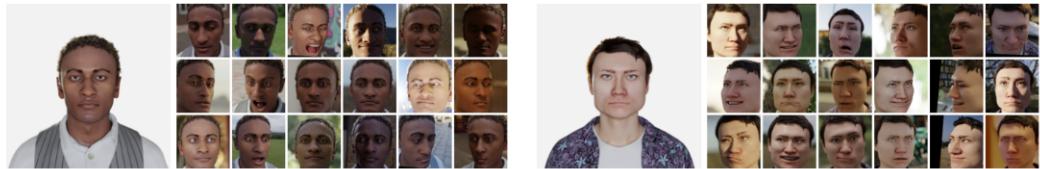
Datasetsy lidských obličejů

Dataset **CelebFaces Attributes Dataset** (CelebA) [25] obsahuje více než 200000 obrázků obličejů známých osobností. Obrázky obsahují fotografie pořízené z různých úhlů a jsou velmi dobře anotovány. Tento dataset je volně přístupný pro nekomerční účely.



Obrázek 3.4: Příklady obličejů z datasetu CelebA

Dataset **DigiFace1M** [6] se skládá z 1 milionu snímků digitálně vytvořených obličejů (viz obrázek 3.5), címž se vyhýbá případným právním a etnickým problémům, které mohou být spojeny s využití tváří reálných osob. Při použití tohoto datasetu umělých tváří společně s 200 až 2000 fotografiemi tváří reálných lidí pro učení, lze dosáhnout podobných výsledků detekce jako s datasyty tvořenými čistě reálnými obličeji.



Obrázek 3.5: Příklady vygenerovaných obličejů pod různými úhly a různým osvětlením z datasetu DigiFace1M [6]

Multi-PIE [17] je dataset zaměřený na fotografie obličejů pořízených za různých světelných podmínek a pod různými úhly. Obsahuje přes 750000 snímků, které byly vytvořeny vyfotografováním 337 lidí po dobu několika měsíců. Tento dataset je placený.

Dataset **UTKFace** [36] obsahuje přes 20000 fotek obličejů lidí různých věků, pohlaví a ras z různých úhlů a za rozličných světelných podmínek. Dataset je volně dostupný pro nekomerční použití. Data jsou podrobně anotovány dle vzorce `[věk]_[pohlaví]_[rasa]_[datum a čas].jpg`, kde věk je v rozmezí 0–116 let, pohlaví muž/žena, rasa je jedna z výčtu běloch, černoch, asiat, ind, ostatní a datum a čas uchovává informaci o okamžiku zařazení fotografie do datasetu.

Dataset **DARK FACE** [42] obsahuje 6000 fotografií různých prostředí s lidmi, pořízených za špatných světelních podmínek. Používá se mj. při hodnocení účinnosti detektorů obličejů. Dataset obsahuje jak anotované tak neanotované obrázky.

3.3 Detektory obličeje

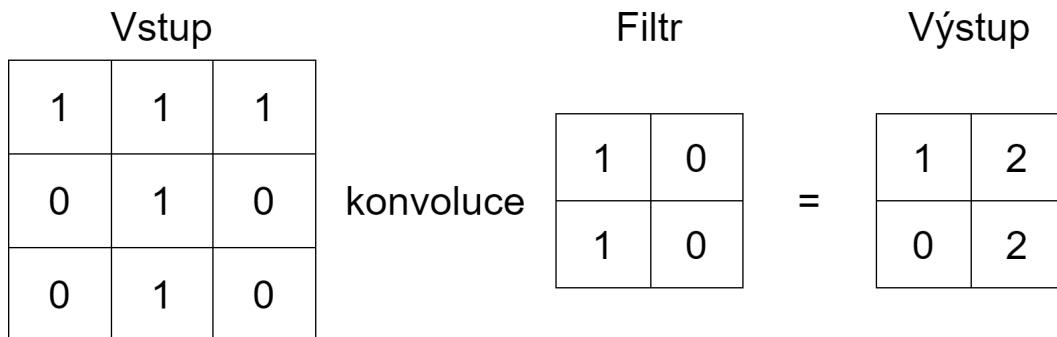
Jak již bylo zmíněno, detekce obličeje je jak samostatnou disciplínou, tak i vstupním bodem několika dalších analyzačních úkonů (rozpoznávání tváře, ověřování pomocí obličejů nebo identifikace obličejů). Zpracovávaný obrázek bývá rozdělen na několik menších oblastí (výřezů) a tyto výřezy bývají zpracovávány jednotlivě. Existuje řada různých druhů a architektur neuronových sítí využívaných k detekci obličejů [4]. Patří mezi ně například:

- **Rotation invariant neural network** (RINN, rotačně invariantní neuronová síť) – dokáže detektovat obličeje bez ohledu na úhel natočení tváře (není nutná normalizace v preprocessingu). Systém tohoto typu neuronové sítě obsahuje několik sítí, s tím že první se nazývá *router network*. Tato síť určuje orientaci (natočení) výřezu obrázku a připravuje (normalizuje) tak výřez pro 1 či více detekčních sítí.
- **Fast neural network** (FNN, rychlá neuronová síť) – obrázek je rozdělen na malé podobrázky a každý podobrázek je zvlášť otestován na výskyt tváře rychlou neuronovou sítí. Cílem je snížení výpočetního času nutného pro nalezení obličeje.
- **Polynomial neural network** (PNN, polynomiální neuronová síť) – oblasti v obrázku jsou klasifikovány jako tvář/ne-tvář pomocí binomické projekce této oblasti do prostoru vlastností obličeje naučeného tzv. základní analýzou komponent (anglicky principal component analysis, PCA). Zkoumáním vlivu PCA na vzorky lze detektovat zda se jedná o obličej.
- **Convolutional neural network** (CNN, konvoluční neuronové sítě) – samostatně popsány v další podsekci.

Konvoluční neuronové sítě

Konvoluční neuronové sítě (CNN) [9, 28, 30] jsou použity ve většině systémů v oblasti počítačového vidění. Jelikož můžeme hodnoty pixelů vyjádřit čísly (používá se šedotónová varianta obrázku → při 8 bitové barevné hloubce může pixel nabývat hodnot 0 – 255), lze tyto čísla použít jako vstupy neuronové sítě. Při vysokém rozlišení obrázku však nastává problém. Pokud bychom chtěli každou hodnotu pixelu použít na vstupu neuronové sítě pro celý obrázek naráz, měla by neuronová síť již u obrázku s rozměry 500×500 pixelů celkem 25000 vstupních neuronů, což je příliš výpočetně náročné. Proto se využívá konvoluce.

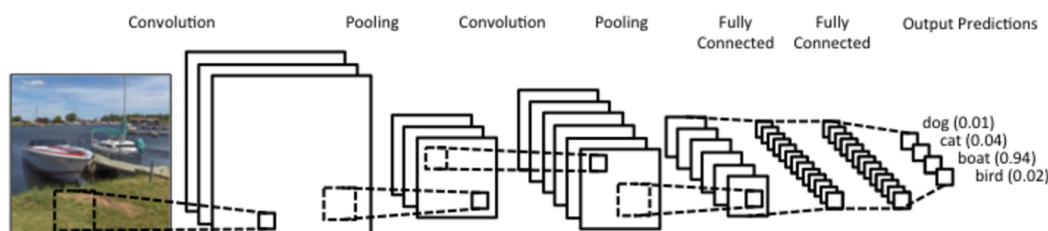
Konvoluce je okenní matematická funkce, při níž je na každou část obrázku (okno, *window*) násobením aplikována matici (filtr, kernel, konvoluční jádro). Princip konvoluce zachycuje následující obrázek.



Obrázek 3.6: Princip konvoluce 2D obrázku s rozměry 3×3 a filtru 2×2

Konvoluce dovoluje zásadně snížit počet nutných vstupů neuronové sítě. Navíc jsou konvoluční operace rychlé, protože tato funkce je hardwarově implementována na grafické kartě.

Jak lze vidět na obrázku 3.7 konvoluční neuronová síť se skládá z několika vrstev.



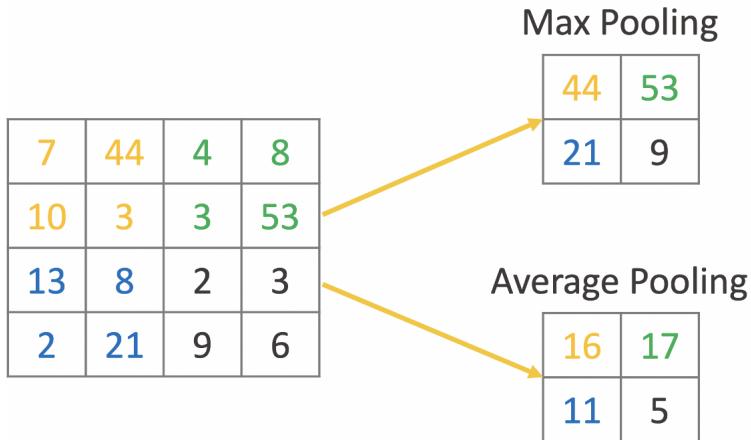
Obrázek 3.7: Řetězec vrstev kónvoluční neuronové sítě [9]

Konvoluční vrstva provádí výše popsanou konvoluci. Otázkou zůstává jak zpracovávat okraje obrázku. Existují dva přístupy:

- *Wide convolution* – přidání nul kolem dokola obrázku a provedení konvoluce krajních hodnot (výstup bude stejně velký jako vstup).
 - *Narrow convolution* – provedení konvoluce jako na obrázku 3.6. Dojde ke zmenšení velikosti na výstupu.

Dalším parametrem konvolučních neuronových sítí je tzv. **stride** velikost. Určuje o kolik hodnot se konvoluční jádro posune při každém kroku. Na obrázku 3.6 je použit stride 1. Větší hodnota stride vede k menšímu výstupu. Běžně se používají hodnoty 1 nebo 2.

Poolingová vrstva (anglicky pooling layer) je vrstva aplikovaná po konvoluční vrstvě. Používá se ke snížení výpočetní náročnosti. Tím, že je aplikována statistická funkce (maximum, průměr) na výstup konvoluční vrstvy (většinou filtrem o velikosti 2×2 se stride = 2), dojde ke zredukování počtu hodnot výsledné matici (viz obrázek 3.8).



Obrázek 3.8: Řetězec vrstev konvoluční neuronové sítě [28]

V reálné aplikaci jsou konvoluční a poolingové vrstvy poskládány za sebe. Na konci řetězce (obrázek 3.7) se pak nachází část plně propojených vrstev neuronů (fully-connected layer).

Detektory

Tato část popisuje vybrané existující detektory obličejů založené na neuronových sítích a porovnání úspěšnosti detekce těchto detektorů s detektory popsanými v sekci 2.3.

MTCNN (Multi-Task Cascaded Convolutional Network) [18, 1] provádí detekci obličejů pomocí kaskády konvoluční neuronových sítí. Algoritmus má dvě části. První část vytváří několik verzí vstupních obrázků, každa taková verze má jiné rozlišení. Druhá část složená z kaskády neuronových sítí se stará o detekci obličeje v různých verzích obrázku. Použití více verzí obrázku umožňuje zvýšit efektivitu detekce.

Single shot detector **SSD** [24] je konvoluční neuronová síť opět složená ze 2 částí, sloužící obecně k detekci objektů. Tyto části jsou: a) natrénovaný detekční model (VGG–16 nebo jiný) a b) část SSD – další konvoluční vrstva, která rozděluje obrázek pomocí mřížky (*grid*) s rozměry 4×4 nebo 8×8 . V každém takto vytvořeném bloku je pak hledán objekt (obličej).

Již výše zmíněný detekční/rozpoznávací algoritmus/neuronová síť **VGG–16** [37] používá k detekci konvoluční filtry o rozměrech 3×3 v několika konvolučních vrstvách za sebou (počet filtrů se s každou další vrstvou navýšuje). Používá se zde stride o hodnotě 1.

Detekční model **RetinaFace** [38] používá multitasking pro učení. Detekce probíhá určením pozice obličeje podle vyhledání pixelů, které tvář obsahuje. Model má 2 struktury: *Multi-task loss* – minimalizace chyb modelu a *Dense Regression Branch* – renderer, který filtrouje obličeje a získává pixely z vyrenderovaných obličejů.

YOLO (You Only Look Once) [15] je stejně jako SSD tzv. single shot detektor – rozděluje vstupní obrázek do mřížky a provádí detekci případně rekognici nad jednotlivými bloky mřížky. Pro každý blok je určena hodnota jistoty výskytu objektu. Na vstupu sítě YOLO jsou barevné obrázky o velikosti 448×448 pixelů. Architektura neuronové sítě je složena ze 7 konvolučních vrstev následovaných pooling vrstvami a 3 plně propojenými vrstvami na výstupu. YOLO má několik verzí, které se liší složením neuronové sítě a úspěšností detekce.

Porovnání výkonnosti detektorů

Práce [18] se zabývá porovnáním detektorů Viola–Jones, Histogram of Oriented Gradients ze sekce 2.3 a algoritmů založených na konvolučních neuronových sítích MTCNN a SSD (detektor VGG–16 je nahrazen detektorem MobileNet) ze sekce 3.3. Testování bylo prováděno na datasetech AFW a WIDER FACE. Co se týká úspěšnosti detekce, nejlépe vyšel z testu detektor MTCNN, nejrychlejší pak byl detektor SSD (viz obrázek 3.9). Z detektorů založených na klasických metodách měl vyšší úspěšnost HOG, rychlejší však byl detektor Viola–Jones. Hodnota AP (average precision) udává průměrnou přesnost detekce a hodnota FPS (frames per second) počet snímků zpracovaných za sekundu.

| Algorithm | AFW | | WIDER FACE | |
|---------------|---------------|-------------|---------------|--------------|
| | AP | FPS | AP | FPS |
| Viola-Jones | 40.80% | 4.16 | 9.09% | 7.52 |
| HOG | 69.88% | 0.33 | 18.05% | 0.71 |
| MTCNN | 89.62% | 0.75 | 44.66% | 1.22 |
| MobileNet-SSD | 89.28% | 9.46 | 25.36% | 10.89 |

Obrázek 3.9: Výsledek testování detekčních algoritmu [18].

Detektory zaměřující se na špatné světelné podmínky

Detektory zaměřující se na detekci ve špatných světelných podmínkách jsou buď specializované detektory, kdy je neuronová síť detektoru učena na datasetu vhodných obrázků (obrázky pořízené za špatných světelných podmínek) nebo je použit detektor obličeje natrénovaný na obyčejných snímcích. V obou případech je nutno nejprve provést *low-light image enhancement* (vylepšení obrázku) a až poté přejít k detekci.

Pro vylepšení obrázku existují různé metody [38]:

- Mirnet,
- Adaptive Gamma Correction (AGC),
- Retinex,
- RetinexNet,
- a další.

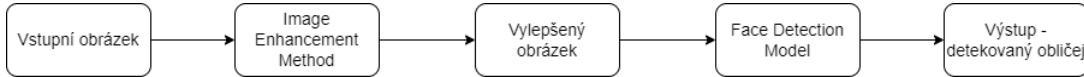
Mirnet je metoda postavená na konvolučních neuronových sítí a jejím cílem je zvýšení kvality obrazu pomocí několika úprav v konvolučních vrstvách.

AGC je metodou využívající gamma korekci k vylepšení nasvícení scény (obrázku). Metoda navíc rozděluje obrázky podle intenzity gamma na tmavé (hodnota pod 0,5) a světlé (hodnota nad 0,5).

Koncept **Retinex** se snaží upravit kvalitu obrazu tak, aby odpovídala kvalitě lidského vidění. Dosaženo toho je tak, že je prováděno odmazávání odrazů světla z obrázku.

RetinexNet vychází z Retinexu a provádí dekompozici obrázku tak, aby došlo k oddělení odrazů nezávislých na osvětlení a osvětlení se známou strukturou. Metoda také celkově upravuje nasvícení pro dosažení co největší konzistence světla v obrázku.

Pro porovnání účinnosti *image enhancement* metod lze použít 2 veličiny. *Peak signal-noise ratio* (PSNR) – poměr užitečného signálu k šumu a *Structural similarity* (SSIM) – index podobnosti 2 obrazů.



Obrázek 3.10: Řetězec postupu detekce obličeje ve zhoršených světelných podmínkách. Vychází z [38].

Obrázek 3.10, který vychází ze článku [38] popisuje potřebný postup pro úpravu obrázku (preprocessing) a následné kroky vedoucí k detekci obličeje. V rámci tohoto článku jsou zpracovány výsledky detekce pomocí výše zmíněných metod vylepšení obrazu a detektoru RetinaFace (viz sekce Detektory). Jako testovací dataset byl použit dataset DARK FACE, popsaný v sekci 3.2. V rámci testování preprocessingových metod vylepšení obrazu bylo zjištěno, že nejlepší hodnoty PSNR (19,48 dB) a SSIM (0,74) dosahuje metoda Mirnet (hodnoty dalších metod viz obrázek 3.11).

| Model | PSNR (dB) | SSIM |
|------------|-----------|------------|
| Raw | 7.83 | 0.20185421 |
| AGC | 13.60 | 0.25519422 |
| Retinex | 13.40 | 0.47176218 |
| RetinexNet | 16.77 | 0.41909298 |
| MirNet | 19.48 | 0.7411703 |

Obrázek 3.11: Hodnoty PSNR a SSIM pro metody vylepšení obrazu [38].

Výsledky detekce však ukázaly, že v detekci obličejů si nejlépe vedl Retinex s 0,43% *mAP* (mean average precision – průměrná přesnost detekce). Z toho vyplývá, že lepší preprocessingová metoda nutně neznamená úspěšnější detektor.

Detekcí ve špatných světelných podmínkách se zabývá také článek [41]. V rámci tohoto článku byl vytvořen samoučící se (self-supervised) detektor dosahující 44,4 % *mAP* také na datech z datasetu DARK FACE.

Práce [43] se rovněž zabývá detekcí nad datasetem DARK FACE při špatném osvětlení. Navržený algoritmus dosahuje 82,3 % *mAP*. Algoritmus využívá 3 hlavní části umožňující dosáhnutí takto vysoké přesnosti detekce. Jsou to: *Multi-scale retinex with color restoration* – metoda podobná Retinexu, ale zachovává konzistenci barev, architektura konvolučních vrstev *PyramidBox* a *Multi-scale test module* – úpravy velikosti vstupního obrázku pro lepší detekci.

Velké rozdíly v hodnotách *mAP* jsou způsobeny tím, že neuronové sítě z prací [41, 43] jsou trénovány přímo na datasetu obličejů ve zhoršených pozorovacích podmírkách (specializoavné detektory), zatímco [38] používá model učený na běžných snímcích tváří.

3.4 Frameworky pro neuronové sítě

Při programování neuronových sítí nebo obecně aplikací strojového učení se využívají k tomu určené frameworky. Každý framework je trochu jiný a používá se k jiným účelům. Všechny zde zmíněné frameworky podporují programovací jazyk Python. Mezi nejpoužívanější frameworky patří: TensorFlow, PyTorch, Keras, ONNX, Caffe [5].

TensorFlow je framework pro jazyky R, C++, Python a další. Má několik modulů s rozličnými funkcionalitami. Je velmi rozšířený a implementuje jej například Google překladač. Dokumentace je velmi podrobná a framework podporuje paralelismus napříč GPU.

PyTorch je vědecký framework použitelný v jazyce Python, vhodný na prototypování. Navíc podporuje GPU paralelismus.

Dalším frameworkem je **Keras**, v němž se programují konvoluční a rekurentní neuronové sítě. Je minimalistický a snadno se integruje spolu s TensorFlow.

Framework **ONNX** je open source framework pro hluboké učení. Modely vytvořené v ONNX lze konvertovat do jiných frameworků (TensorFlow, Keras).

Caffe je framework podporovaný napříč různými programovacími jazyky (C, C++, Python, MATLAB) a je vhodný pro konvoluční neuronové sítě. Umožňuje nastavit parametry sítě a natrénovat síť, není nutné ji přímo vytvářet.

V rámci této práce jsou použity frameworky TensorFlow a Keras.

Kapitola 4

Systémy pro detekci obličejů a akcelerace detekce

Detekci obličejů s využitím počítačových programů lze provádět nad snímkem (fotografií, obrázkem), sadou snímků, videozáznamem nebo tzv. real-timově pomocí kamer a kamerových systémů. V této kapitole jsou popsány aktuálně využívané prostředky pro vytváření obrázků k detekci obličejů (kamery) a také dostupná řešení zabývající se detekcí, a to jak placené komerční, tak neplacené open-source systémy.

4.1 Kamery

Nezbytnou součástí oboru detekce obličejů jsou kamery a kamerové systémy. Existují kamery specializované k detekci či rozpoznávání obličejů a kamery obyčejné, které pouze zprostředkovávají obraz dále ke zpracování.

Specializované kamery se používají například k zabezpečení objektů nebo jako domovní videozvonky, kdy kamera (respektive její software) v zachyceném obrazu detekuje a rozpozná obličej, a následně může vykonat přiřazenou akci (spustit alarm, poslat notifikaci, umožnit osobě vstup...) [32].

Další oblastí, kde se kamery s detekcí a rozpoznáváním obličejů uplatní je bezesporu dohled ve veřejných prostorách (anglicky CCTV surveillance). Detekce obličeje může sloužit k hledání podezřelých osob v záznamech z dohledových kamer. Tyto záznamy jsou shromážďovány na serveru, pomocí detekce jsou z obrazu vyřezány fragmenty s obličeji lidí a poté jsou tyto fragmenty porovnány rozpoznávacím algoritmem s obličeji hledaných osob. Při shodě dochází k informování administrátora systému, který podnikne další kroky [39].

Kamery tedy mají v doméně detekce obličejů nezastupitelnou roli, na jejich praktické využití v systémech a řešení pro detekci se zaměřuje následujících podkapitol.

4.2 Dostupná řešení

Řešení umožňující detekci (a často i rekognici) obličejů lze rozdělit do dvou kategorií: komerční (placené, profesionální) a nekomerční (zdarma, open-source, amatérské). V následujících dvou podsekčích jsou popsány konkrétní systémy umožňující detekci obličejů včetně jejich výhod a nevýhod.

Komerční

Komerčně využívaná zařízení pro detekci, případně rekognici lze běžně zakoupit a používat ve firemním nebo domácím prostředí. Mezi zástupce těchto zařízení patří například produkty firem Netatmo, Google, Nest [33].



Obrázek 4.1: Nest Hello [33]

Nekomerční

Nekomerční řešení pro detekci obličejů zahrnují frameworky s otevřeným zdrojovým kódem (open-source). Tyto frameworky a řešení využívají neuronové sítě, které jsou trénovány pomocí datasetů a následně jsou využity k detekci obličeje [13]. Mezi open-source řešení patří například TinaFace [2] nebo MTCNN [1], další volně použitelné frameworky popisuje sekce 3.4.

4.3 Akcelerace detekčních algoritmů

S rozmachem využívání strojového učení a neuronových sítí se objevila potřeba zrychlení těchto sítí. Proto byly vytvořeny specializované zařízení s cílem urychlit výpočty a snížit energetickou náročnost výpočtů ve srovnání s běžnými (univerzálními) výpočetními prostředky. Vznikly tak hardwarové akcelerátory neuronových sítí. Příkladem jsou Intel Neural Compute Stick 2 [19], Coral [11] nebo Nvidia Jetson [29].

Existují různé typy architektur akcelerátorů [10], mezi něž patří architektury popsané v tomto odstavci. **NPU** (neural processing unit) – k provádění matematických operací využívají speciální NPU obsahující PE (processing engines). NPU používá hardwarové verze MLP (multi-layer perceptron), pro dosažení zrychlení obecného výpočtu (nejen neuronových sítí). Pokud je část programu určena ke zrychlení spouštěna často a jsou dobré známy vstupy a výstupy, lze tuto část programu zrychlit pomocí NPU. Příkladem algoritmu pro zrychlení je rychlá Fourierova transformace (FFT).

Architektura **RENO** využívá memristory (ReRAM) – speciální druh paměti jejíž struktura umožňuje urychlit maticové a vektorové násobení.

Mezi další akcelerátory patří série akcelerátorů **DianNao** využívaná na akademické půdě. Tyto akcelerátory obsahují NFU (neural functional unit), vstupní, výstupní a synaptický buffer a kontrolní procesor.

Průmyslovým akcelerátorem od Google je **TPU** (tensor processing unit). Je použitelný i přes cloud.

Kromě ReRAM jsou v akcelerátor používány i paměti typu HMC (hybrid memory cube). Obě paměti (ReRAM a HMC) umožňují tzv. *processing-in-memory*, takže snižují čas výpočtu tím, že není nutno data přesouvat mezi procesorem a pamětí. Příklady akcelerátorů postavených na HMC je Neurocube [22] nebo Tetris [26].

Většina akcelerátorů je zaměřena na výpočty s již natrénovanou neuronovou sítí, jen pár jich podporuje učení neuronových sítí. V některých případech (edge computing) se ke zrychlení neuronové sítě používají cloudové služby – v datacentrech provádějí náročné výpočty grafické karty a výsledky spolu s nenáročnými výpočty jsou zpracovány například na IoT (Internet of Things – internet věcí) nebo mobilních zařízeních.

Intel Neural Compute Stick 2

Intel Neural Compute Stick 2 (NCS2) [19] (obrázek 4.2) je akcelerátor neuronových sítí zaměřený na počítačové vidění. Obsahuje VPU (vision processing unit) Intel Movidius X. Připojuje se k počítači přes rozhraní USB 3.0 a umožňuje urychlení neuronové sítě bez použití cloutu a s nízkými energetickými nároky (například v kombinaci s Raspberry Pi). Podporuje mj. frameworky TensorFlow, Caffe, PyTorch, Keras popsané v sekci 3.4. Pro práci s NCS2 se používá knihovna/framework OpenVINO [20].



Obrázek 4.2: Intel Neural Compute Stick 2 [19]

Experimenty provedené v [14] porovnávají výkonnost akcelerátorů Nvidia Jetson Nano, Nvidia Jetson Xavier NX a Raspberry Pi 4B s NCS2. K porovnání posloužily frameworky YOLOv3 popsaný v sekci 3.3 a YOLOv3-tiny. Experimenty byly založeny na detekci objektů ve dvou videích.

- *Video1* – rozlišení 768×436 pixelů, celkem 1596 framů
- *Video2* – rozlišení 1920×1080 pixelů, celkem 960 framů

Naměřené výsledky zobrazuje obrázek 4.3.

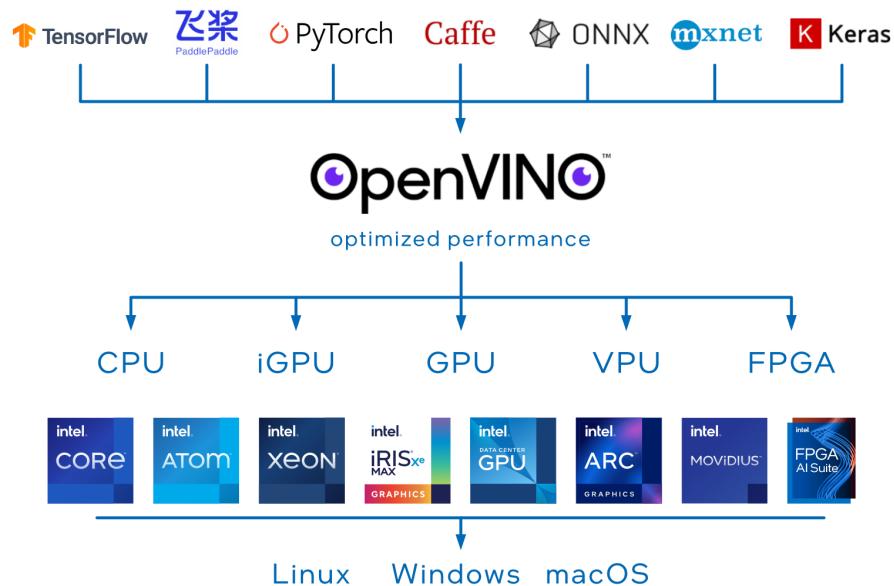
| | Models | Accelerator-based SBCs | Mean Confidence(%) | FPS | CPU Usage (%) | Memory Usage (GB) | Power (W) | Time (s) |
|--------|-------------|------------------------|--------------------|------|---------------|-------------------|-----------|----------|
| Video1 | YOLOv3 | RPI+NCS2 | 99.3 | 2.5 | 4.3 | 0.33 | 6.0 | 690 |
| | | Nano | 99.7 | 1.7 | 26.5 | 1.21 | 7.9 | 967 |
| | | NX | 99.7 | 6.1 | 22.5 | 1.51 | 15.2 | 256 |
| | YOLOv3-tiny | RPI+NCS2 | 0 | 18.8 | 15.5 | 0.11 | 6.5 | 121 |
| | | Nano | 57.9 | 6.8 | 28.8 | 1.00 | 7.2 | 236 |
| | | NX | 57.9 | 41.1 | 30.5 | 1.33 | 13.5 | 46 |
| Video2 | YOLOv3 | RPI+NCS2 | 85.8 | 2.5 | 9.8 | 0.41 | 6.2 | 496 |
| | | Nano | 71.5 | 1.6 | 28.8 | 1.36 | 8.0 | 587 |
| | | NX | 71.5 | 5.9 | 26.8 | 1.69 | 15.2 | 162 |
| | YOLOv3-tiny | RPI+NCS2 | 61.5 | 19.0 | 24.8 | 0.18 | 6.8 | 162 |
| | | Nano | 54.1 | 6.6 | 37.3 | 1.16 | 7.4 | 152 |
| | | NX | 54.1 | 35.6 | 55.5 | 1.47 | 13.2 | 31 |

Obrázek 4.3: Výsledky testu akcelerátorů neuronových sítí s modelem YOLOv3 a YOLOv3-tiny [14]

Z tabulky v obrázku 4.3 můžeme vidět, že NCS2 dosahuje podobných výsledků detekce ve Video1 jako Jetson Nano a zároveň nejméně zatěžuje procesor. U Video2 má NCS2 nejlepší výsledky detekce. Využití paměti s větším vstupem roste, stále je však mezi konkurenčními nejnižší, to samé lze říci o spotřebě energie. S dostatečně dobrým modelem neuronové sítě je tak NCS2 úsporným akcelerátorem s vysokou úspěšností detekce.

OpenVINO

Framework/knihovna OpenVINO se stará o konverzi a optimalizaci natrénovaného modelu neuronové sítě v jednom z podporovaných frameworků (viz obrázek 4.4) pro použití na NCS2 nebo jiném zařízení (procesory, grafické karty). OpenVINO se nezaměřuje pouze na oblast počítačového vidění, ale na neuronové sítě obecně.



Obrázek 4.4: Využití frameworku OpenVINO [20]

Kapitola 5

Závěr

V rámci tohoto textu byla představena problematika detekce obličejů v reálných podmínkách, dále byly popsány problémy omezující detekci a klasické algoritmy detekce obličejů. V samostatných kapitolách a sekčích byly popsány neuronové sítě (obecně, konvoluční) a algoritmy pro detekci založené na neuronových sítích. Dále pak existující komerční a nekomerční řešení a systémy detekce a možnosti akcelerace neuronových sítí specializovaným hardwarem.

V této práci bude pokračováno návrhem neuronové sítě pro detekci ve špatných světelních podmínkách, návrhem Python aplikace, její implementací a provedením experimentů s vytvořeným řešením a existujícími řešeními.

Literatura

- [1] *MTCNN* [online]. [cit. 2022-10-19]. Dostupné z: <https://github.com/ipazc/mtcnn>.
- [2] *TinaFace: Strong but Simple Baseline for Face Detection* [online]. [cit. 2022-10-19]. Dostupné z:
<https://github.com/Media-Smart/vedadet/tree/main/configs/trainval/tinafac>.
- [3] AHONEN, T., HADID, A. a PIETIKAINEN, M. Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2006, sv. 28, č. 12, s. 2037–2041. DOI: 10.1109/TPAMI.2006.244.
- [4] AL-ALLAF, O. N. A. Review of Face Detection Systems Based Artificial Neural Networks Algorithms. *CoRR*. 2014, abs/1404.1292. Dostupné z:
<http://arxiv.org/abs/1404.1292>.
- [5] ARIWALA, P. *Top 8 Deep Learning Frameworks* [online]. [cit. 2022-01-17]. Dostupné z: <https://marutitech.com/top-8-deep-learning-frameworks/>.
- [6] BAE, G., LA GORCE, M. de, BALTRUSAITIS, T., HEWITT, C., CHEN, D. et al. DigiFace-1M: 1 Million Digital Face Images for Face Recognition. In: IEEE. *2023 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2023.
- [7] BEHERA, G. S. *Face Detection with Haar Cascade* [online]. [cit. 2022-11-06]. Dostupné z: <https://www.aitimejournal.com/@alessandro.eric.russo/best-wifi-security-cameras-with-face-recognitionhttps://towardsdatascience.com/face-detection-with-haar-cascade-727f68daff08>.
- [8] BISHOP, C. M. Neural networks and their applications. *Review of Scientific Instruments*. 1994, sv. 65, č. 6, s. 1803–1832. DOI: 10.1063/1.1144830. Dostupné z: <https://doi.org/10.1063/1.1144830>.
- [9] BRITZ, D. *Understanding Convolutional Neural Networks for NLP* [online]. [cit. 2022-12-12]. Dostupné z: <https://dennybritz.com/posts/wildml/understanding-convolutional-neural-networks-for-nlp/>.
- [10] CHEN, Y., XIE, Y., SONG, L., CHEN, F. a TANG, T. A Survey of Accelerator Architectures for Deep Neural Networks. *Engineering*. 2020, sv. 6, č. 3, s. 264–274. DOI: <https://doi.org/10.1016/j.eng.2020.01.007>. ISSN 2095-8099. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S2095809919306356>.
- [11] CORAL. *Build beneficial and privacy preserving AI* [online]. [cit. 2022-12-25]. Dostupné z: <https://coral.ai>.

- [12] DALAL, N. a TRIGGS, B. Histograms of oriented gradients for human detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. 2005, sv. 1, s. 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.
- [13] DAVID WANYONYI, T. C. Open-Source Face Recognition Frameworks: A Review of the Landscape. Květen 2022, sv. 10. DOI: 10.1109/ACCESS.2022.3170037. Dostupné z: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9762647>.
- [14] FENG, H., MU, G., ZHONG, S., ZHANG, P. a YUAN, T. Benchmark Analysis of YOLO Performance on Edge Intelligence Devices. In: *2021 Cross Strait Radio Science and Wireless Technology Conference (CSRSWTC)*. 2021, s. 319–321. DOI: 10.1109/CSRSWTC52801.2021.9631594.
- [15] GARG, D., GOEL, P., PANDYA, S., GANATRA, A. a KOTECHA, K. A Deep Learning Approach for Face Detection using YOLO. In: *2018 IEEE Punecon*. 2018, s. 1–4. DOI: 10.1109/PUNECON.2018.8745376.
- [16] GOODFELLOW, I., BENGIO, Y. a COURVILLE, A. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [17] GROSS, R., MATTHEWS, I., COHN, J., KANADE, T. a BAKER, S. Multi-PIE. In: *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*. 2008, s. 1–8. DOI: 10.1109/AFGR.2008.4813399.
- [18] GUO, Y. a WÜNSCHE, B. C. Comparison of Face Detection Algorithms on Mobile Devices. In: *2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ)*. 2020, s. 1–6. DOI: 10.1109/IVCNZ51579.2020.9290542.
- [19] INTEL. *Intel® Neural Compute Stick 2 (Intel® NCS2)* [online]. [cit. 2022-12-25]. Dostupné z: <https://www.intel.com/content/www/us/en/developer/articles/tool/neural-compute-stick.html>.
- [20] INTEL. *OpenVINO* [online]. [cit. 2022-12-25]. Dostupné z: <https://docs.openvino.ai/latest/home.html>.
- [21] KADIR, K., KAMARUDDIN, M. K., NASIR, H., SAFIE, S. I. a BAKTI, Z. A. K. A comparative study between LBP and Haar-like features for Face Detection using OpenCV. In: *2014 4th International Conference on Engineering Technology and Technopreneurship (ICE2T)*. 2014, s. 335–339. DOI: 10.1109/ICE2T.2014.7006273.
- [22] KIM, D., KUNG, J., CHAI, S., YALAMANCHILI, S. a MUKHOPADHYAY, S. Neurocube: A Programmable Digital Neuromorphic Architecture with High-Density 3D Memory. In: *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*. 2016, s. 380–392. DOI: 10.1109/ISCA.2016.41.
- [23] KUMAR, A., KAUR, A. a KUMAR, M. Face detection techniques: a review. *Artificial Intelligence Review*. Aug 2019, sv. 52, č. 2, s. 927–948. DOI: 10.1007/s10462-018-9650-2. ISSN 1573-7462. Dostupné z: <https://doi.org/10.1007/s10462-018-9650-2>.
- [24] LIU, W., ANGUELOV, D., ERHAN, D., SZEGEDY, C., REED, S. E. et al. SSD: Single Shot MultiBox Detector. *CoRR*. 2015, abs/1512.02325. Dostupné z: <http://arxiv.org/abs/1512.02325>.

- [25] LIU, Z., LUO, P., WANG, X. a TANG, X. *Large-scale CelebFaces Attributes (CelebA) Dataset* [online]. [cit. 2022-12-10]. Dostupné z: <https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>.
- [26] LU, H., WEI, X., LIN, N., YAN, G. a LI, X. Tetris: Re-architecting Convolutional Neural Network Computation for Machine Learning Accelerators. In: *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 2018, s. 1–8. DOI: 10.1145/3240765.3240855.
- [27] MEHENDALE, N. Facial emotion recognition using convolutional neural networks (FERC). *SN Applied Sciences*. Feb 2020, sv. 2, č. 3, s. 446. DOI: 10.1007/s42452-020-2234-1. ISSN 2523-3971. Dostupné z: <https://doi.org/10.1007/s42452-020-2234-1>.
- [28] MELCHER, K. *Convolutional Neural Networks & Computer Vision* [online]. [cit. 2022-12-12]. Dostupné z: <https://www.knime.com/blog/convolutional-neural-networks-computer-vision>.
- [29] NVIDIA. *Embedded Systems with Jetson* [online]. [cit. 2022-12-25]. Dostupné z: <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/>.
- [30] O'SHEA, K. a NASH, R. An Introduction to Convolutional Neural Networks. *CoRR*. 2015, abs/1511.08458. Dostupné z: <http://arxiv.org/abs/1511.08458>.
- [31] RAMIREZ CERNA, L. Face Detection: Histogram of Oriented Gradients and Bag of Feature Method. Říjen 2013.
- [32] RUSSO, A. E. *Are Face Recognition Security Cameras Worth It?* [online]. [cit. 2022-10-18]. Dostupné z: <https://www.aitimejournal.com/@alessandro.eric.russo/are-face-recognition-security-cameras-worth-it>.
- [33] RUSSO, A. E. *Best Home Security Cameras with Face Recognition* [online]. [cit. 2022-10-18]. Dostupné z: <https://www.aitimejournal.com/@alessandro.eric.russo/best-wifi-security-cameras-with-face-recognition>.
- [34] SAEZ-TRIGUEROS, D., MENG, L. a HARTNETT, M. Face Recognition: From Traditional to Deep Learning Methods. *CoRR*. 2018, abs/1811.00116. Dostupné z: <http://arxiv.org/abs/1811.00116>.
- [35] SHARIFARA, A., MOHD RAHIM, M. S. a ANISI, Y. A general review of human face detection including a study of neural networks and Haar feature-based cascade classifier in face detection. In: *2014 International Symposium on Biometrics and Security Technologies (ISBAST)*. 2014, s. 73–78. DOI: 10.1109/ISBAST.2014.7013097.
- [36] SONG, Y. a ZHANG, Z. *UTKFace* [online]. [cit. 2022-12-10]. Dostupné z: <https://susanqq.github.io/UTKFace/>.
- [37] TEAM, G. L. *Introduction to VGG16 / What is VGG16?* [online]. [cit. 2022-12-19]. Dostupné z: <https://www.mygreatlearning.com/blog/introduction-to-vgg16/>.

- [38] THEOFFILUS, R. B., DHARMADINATA, O. J. a KUSUMA, G. P. LOW-LIGHT FACE DETECTION USING DEEP LEARNING. *Jatit.* 2022, sv. 100. Dostupné z: <http://www.jatit.org/volumes/Vol100No10/28Vol100No10.pdf>.
- [39] V D, A. K., KUMAR, V., SUBRAMANIAN, M., VENGATESAN, K. a RAMAKRISHNAN, M. Facial Recognition System for Suspect Identification Using a Surveillance Camera. *Pattern Recognition and Image Analysis.* Červenec 2018, sv. 28, s. 410–420. DOI: 10.1134/S1054661818030136.
- [40] VIOLA, P. a JONES, M. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.* 2001, sv. 1, s. I–I. DOI: 10.1109/CVPR.2001.990517.
- [41] WANG, W., YANG, W. a LIU, J. HLA-Face: Joint High-Low Adaptation for Low Light Face Detection. *CoRR.* 2021, abs/2104.01984. Dostupné z: <https://arxiv.org/abs/2104.01984>.
- [42] YANG, W. *DARK FACE: Face Detection in Low Light Condition* [online]. [cit. 2022-12-19]. Dostupné z: <https://flyywh.github.io/CVPRW2019LowLight/>.
- [43] YU, J., HAO, X. a HE, P. Single-stage Face Detection under Extremely Low-light Conditions. In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW).* 2021, s. 3516–3525. DOI: 10.1109/ICCVW54120.2021.00392.