

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2018

Bc. Dominik Štarha



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

**MEŘENÍ PODOBNOSTI OBRAZŮ S POMOCÍ HLUBOKÉHO  
UČENÍ**

IMAGE SIMILARITY MEASURING USING DEEP LEARNING

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. Dominik Štarha**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Martin Rajnoha**

**BRNO 2018**



# Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

**Student:** Bc. Dominik Štarha

**ID:** 155864

**Ročník:** 2

**Akademický rok:** 2017/18

**NÁZEV TÉMATU:**

## Meření podobnosti obrazů s pomocí hlubokého učení

### POKYNY PRO VYPRACOVÁNÍ:

Prostudujte současné možnosti měření podobnosti obrazů, zaměřte se na přístupy s použitím umělé inteligence. Teoreticky popište problematiku hlubokého učení, konvolučních vrstev a jejich možnosti a využití pro zpracování obrazu. Za pomoci frameworku Keras navrhnete, implementujete a natrénujete hlubokou neuronovou síť, která předpovídá míru podobnosti dvou obrazů. Síť navrhnete tak, aby ji bylo možné přetrénovat na specifický problém. Dosažené výsledky vhodně reprezentujte a demonstруйте na příkladech.

### DOPORUČENÁ LITERATURA:

[1] WANG, Jiang, et al. Learning fine-grained image similarity with deep ranking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014. p. 1386-1393.

[2] ZAGORUYKO, Sergey; KOMODAKIS, Nikos. Learning to compare image patches via convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015. p. 4353-4361.

**Termín zadání:** 5.2.2018

**Termín odevzdání:** 21.5.2018

**Vedoucí práce:** Ing. Martin Rajnoha

**Konzultant:**

**prof. Ing. Jiří Mišurec, CSc.**  
předseda oborové rady

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

## ABSTRAKT

Tato diplomová práce se zabývá výzkumem technologií, využívajících metod hlubokého učení, využitelných při zpracovávání obrazových dat. Konkrétním zaměřením práce je zhodnotit vhodnost a efektivnost hlubokého učení při porovnávání dvou vstupních obrazových dat. První – teoretická – část zahrnuje úvod do problematiky neuronových sítí a hlubokého učení. Obsahuje popis dostupných možností a jejich výhody a principy, vhodné při zpracování obrazových dat. Druhá – praktická – část práce obsahuje návrh vhodného modelu siamských sítí pro řešení problému problematiky porovnávání dvou vstupních obrazů a vyhodnocení jejich podobnosti. Výstupem je zhodnocení několika možných konfigurací modelu a vyzdvihnutí parametrů modelu s nejlepšími výsledky.

## KLÍČOVÁ SLOVA

Obraz, porovnání, podobnost, dataset, hluboké, učení, umělá, inteligence, strojové, učení, konvoluční, vrstva, konvoluce, sdružování, model, siamská, neuronová, síť, trénování, validace, testování, Keras, TensorFlow, Python, PyCharm, neuron, filtr.

## ABSTRACT

This master's thesis deals with the research of technologies using deep learning method, being able to use when processing image data. Specific focus of the work is to evaluate the suitability and effectiveness of deep learning when comparing two image input data. The first – theoretical – part consists of the introduction to neural networks and deep learning. Also, it contains a description of available methods, their benefits and principles, used for processing image data. The second - practical - part of the thesis contains a proposal a appropriate model of Siamese networks to solve the problem of comparing two input image data and evaluating their similarity. The output of this work is an evaluation of several possible model configurations and highlighting the best-performing model parameters.

## KEYWORDS

Image, comparsion, similarity, dataset, deep, learning, artifical, intelligence, machine, learning, convolutional, layer, convolution, pooling, model, siamese, neuron, network, training, validation, testing, Keras, TensorFlow, Python, PyCharm, neuron, filter.

ŠTARHA, Dominik *Meření podobnosti obrazů s pomocí hlubokého učení*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2017. 61 s. Vedoucí práce byl Ing. Martin Rajnoha,

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Meření podobnosti obrazů s pomocí hlubokého učení“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora(-ky)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Martinu Rajnohovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora(-ky)

## PODĚKOVÁNÍ

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno .....

.....  
podpis autora(-ky)

# OBSAH

<b>Úvod</b>	<b>12</b>
<b>1 Obraz</b>	<b>13</b>
1.1 Obraz - struktura . . . . .	13
1.1.1 Vektorová grafika . . . . .	13
1.1.2 Rastrové obrazy . . . . .	15
<b>2 Neuronové sítě</b>	<b>18</b>
2.1 Neuron . . . . .	18
2.1.1 Aktivační funkce . . . . .	19
2.2 Neuronová síť . . . . .	21
2.2.1 Učení neuronových sítí . . . . .	22
<b>3 Hluboké učení</b>	<b>24</b>
3.1 Historie . . . . .	24
3.2 Hluboké neuronové sítě . . . . .	25
3.2.1 Rekurentní neuronové sítě . . . . .	25
3.2.2 Konvoluční neuronové sítě . . . . .	26
3.2.3 Konvoluční vrstvy . . . . .	28
3.2.4 ReLu vrstvy . . . . .	31
3.2.5 Sdružovací vrstvy . . . . .	32
3.2.6 Vrstvy zahazování . . . . .	33
3.2.7 Plně propojené vrstvy . . . . .	33
3.2.8 Optimalizační algoritmy . . . . .	34
<b>4 Metody měření podobnosti</b>	<b>35</b>
4.1 Porovnání na základě struktury obrazu . . . . .	35
4.2 Využití hlubokého učení . . . . .	36
<b>5 Návrh řešení</b>	<b>37</b>
<b>6 Prostředky k ověření metody</b>	<b>38</b>
6.1 Použité programy a nástroje . . . . .	38
6.2 Použitý dataset . . . . .	39
<b>7 Realizace neuronové sítě</b>	<b>41</b>
7.1 Hlavní části a jejich proměnné parametry . . . . .	41
7.2 Struktury modelu . . . . .	42



7.2.1	Struktura 1 . . . . .	43
7.2.2	Struktura 2 . . . . .	43
7.2.3	Struktura 3 . . . . .	43
7.2.4	Struktura 4 . . . . .	44
7.2.5	Struktura 5 . . . . .	44
7.2.6	Struktura 6 . . . . .	45
7.2.7	Struktura 7 . . . . .	45
7.2.8	Struktura 8 . . . . .	46
7.2.9	Struktura 9 . . . . .	46
7.3	Syntaxe definice modelu . . . . .	47
7.3.1	Část siamských větví . . . . .	47
7.3.2	Společná část . . . . .	47
<b>8</b>	<b>Výsledky</b>	<b>49</b>
<b>9</b>	<b>Vyhodnocení výsledků</b>	<b>55</b>
9.1	Nejlepší návrh . . . . .	55
9.2	Ostatní návrhy . . . . .	55
<b>10</b>	<b>Závěr práce</b>	<b>57</b>
	<b>Literatura</b>	<b>59</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>61</b>

# SEZNAM OBRÁZKŮ

1.1	Vektorová grafika - kompletní obraz . . . . .	14
1.2	Přiblížení na písmeno „V“ . . . . .	14
1.3	Rastrová grafika - ukázkový obraz . . . . .	15
1.4	Struktura rastrového obrazu . . . . .	15
1.5	Model RGB . . . . .	16
1.6	YCbCr rozklad na složky . . . . .	17
1.7	Model 4:4:4 - bez eliminace složek . . . . .	17
1.8	Model 4:2:0 - eliminace složek Cb a Cr . . . . .	17
2.1	Model neuronu[12] . . . . .	19
2.2	Skoková přenosová funkce [20]. . . . .	19
2.3	Sigmoidální přenosová funkce [20]. . . . .	20
2.4	Funkce hyperbolické tangenty [20]. . . . .	20
2.5	Funkce radiální báze [20]. . . . .	20
2.6	Funkce ReLU [20]. . . . .	21
2.7	Abstrakce neuronové sítě[12] . . . . .	21
3.1	Neuronová síť[12] . . . . .	24
3.2	Rekurentní neuronová síť . . . . .	26
3.3	Schéma konvoluční neuronové sítě [19]. . . . .	27
3.4	Počet konvolučních vrstev.[19] . . . . .	28
3.5	Příklad realizace konvoluční neuronové sítě . . . . .	28
3.6	Konvoluční vrstvy - propojení neuronů [18]. . . . .	29
3.7	Vstupní data do konvoluční vrstvy . . . . .	29
3.8	Sada filtrů konvoluční vrstvy . . . . .	30
3.9	Blok vstupních dat 5 x 5 . . . . .	30
3.10	Konvoluce prvního bloku . . . . .	30
3.11	Aktivační mapy . . . . .	31
3.12	Příklad využití ReLU vrstvy - odstranění záporných hodnot [21] . . . . .	31
3.13	Sdružování podle maximální hodnoty [22] . . . . .	32
3.14	Plně propojené vrstvy [18] . . . . .	33
4.1	Dvě číselné matice – jednoduché porovnání . . . . .	35
4.2	Matice se společným objektem . . . . .	35
4.3	Rozdílné matice . . . . .	36
5.1	Schéma modelu siamských sítí . . . . .	37
6.1	Dataset – příklad první osoby . . . . .	39
6.2	Dataset – příklad druhé osoby . . . . .	39
6.3	Struktura použitého datasetu . . . . .	40
7.1	Obecný model . . . . .	42

7.2	Struktura 1 . . . . .	43
7.3	Struktura 2 . . . . .	43
7.4	Struktura 3 . . . . .	44
7.5	Struktura 4 . . . . .	44
7.6	Struktura 5 . . . . .	44
7.7	Struktura 6 . . . . .	45
7.8	Struktura 7 . . . . .	45
7.9	Struktura 8 . . . . .	46
7.10	Struktura 9 . . . . .	46
8.1	Graf nejlepšího výsledku struktury 1 . . . . .	52
8.2	Graf nejlepšího výsledku struktury 3 . . . . .	53
8.3	Graf nejlepšího výsledku struktury 4 . . . . .	53
8.4	Graf nejlepšího výsledku struktury 5 . . . . .	54
8.5	Graf nejlepšího výsledku struktury 6 . . . . .	54

## SEZNAM TABULEK

8.1	Struktura 1 - Přehled výsledků . . . . .	49
8.2	Struktura 2 - Přehled výsledků . . . . .	50
8.3	Struktura 3 - Přehled výsledků . . . . .	50
8.4	Struktura 4 - Přehled výsledků . . . . .	50
8.5	Struktura 5 - Přehled výsledků . . . . .	51
8.6	Struktura 6 - Přehled výsledků . . . . .	51
8.7	Struktura 7 - Přehled výsledků . . . . .	51
8.8	Struktura 8 - Přehled výsledků . . . . .	51
8.9	Struktura 9 - Přehled výsledků . . . . .	51

# ÚVOD

V dnešní době multimediálního rozmachu lze najít obrazová data v obrovském množství odvětví. Existence a postupné narůstání objemu obrazových databází otevírá mnoho nových možností ve zpracování, či jiném využití těchto dat. V některých případech lze aplikací nových metod docílit zjednodušení již existujících postupů, jindy se však může prakticky jednat o nutnost, například z hlediska bezpečnostního.

Jako příklad využití existujících databází obrazových dat lze uvést porovnání a určení míry podobnosti obrazů, kterým se zabývá tato diplomová práce. Údaje o podobnosti obrazů je možné vyžadovat z mnoha důvodů. Jedním z významných využití je vyhodnocení podobností dvou fotografií člověka a určení, zda se může jednat o stejnou osobu. Tento postup lze využít například při identifikaci osoby, zachycené bezpečnostním systémem, ať už se jedná o osobu hledanou, či oprávněnou ke vstupu do chráněných objektů. Důvodů k aplikaci těchto metod je samozřejmě nezměrné množství a obzory využití jsou téměř v nedohlednu.

Obrazové databáze však disponují jednou parazitní vlastností, která je na první pohled zřejmá - jedná se o obrovské množství dat. Zpracování těchto dat je i z hlediska struktury obrazu velmi náročné a prakticky nelze řešit jinak, než za pomoci moderních výpočetních technologií.

Pokud uvažujeme porovnání dvou obrazů, lze si představit jednoduchý algoritmus, který porovná jednotlivé pixely a na základě počtu shodných vyhodnotí podobnost. Tímto přístupem však nedosáhneme jisté modularity systému. Fotografie jedné osoby v jiném prostředí a při jiném nasvícení by byly považovány za zcela rozdílné.

Je tedy nutné vytvořit metodu s jistou adaptovatelností. Klíčem k tomu může být využití tzv. „Hlubokého učení“ za pomoci neuronových sítí a konvolučních vrstev.

V teoretické části práce se lze seznámit s vlastnostmi obrazu, zejména těmi, které jsou využitelné pro metody určení podobnosti. Následuje popis hlubokého učení, jeho principy a využití v širokém spektru oborů. Zakončení této části je věnováno popisu a rozboru existujících metod, podle nichž lze obrazy porovnávat.

Praktická část sestává z výběru vhodné metody a uskutečnění její realizace. Z výsledků budou určeny závěry o vhodnosti metody k porovnání obrazů a aspekty, ovlivňující její úspěšnost.

# 1 OBRAZ

Jedněmi z prvních obrazů, vytvořenými člověkem, jsou bezesporu nástěnné malby pračlověka. Historie tohoto umění sahá do dávné minulosti lidstva. V Evropě se malby vyskytovaly již okolo 12,000 - 17,000 let před naším letopočtem v oblasti jeskynního komplexu Altamira nedaleko Pyrenejí na území dnešního Španělska. Malby nalezené na území dnešní Francie dokonce dosahují stáří až 30,000 let[1].

Za velký milník směrem k digitální reprezentaci obrazu se dá považovat vznik prvních fotografií, datující se do 19. století. Francouzský vynálezce *Joseph Nicéphore Niépce* pomocí profesionálně zhotoveného zařízení zachytil roku 1826 výhled z okna své pracovny na cínovou desku s petrolejovým roztokem asfaltu[9]. Zajímavostí je, že Niépceho k výzkumu vedl mimo jiné i nedostatek uměleckých vloh[8].

Tato práce se však zabývá poněkud modernější reprezentací obrazů – digitální. Rozmach tohoto typu zpracování obrazu nastal s příchodem digitálních počítačů, schopných data zpracovávat. Nutnost reprezentace obrazových dat v paměti zařízení dala vzniknout digitalizované podobě obrazu. Dřívější fotografická zařízení zachycovala obraz v analogové podobě. Takto pořízená data by bylo nutné před zpracováním digitalizovat. Od roku 1990 jsou však široké veřejnosti již dostupné digitální fotoaparáty[10], které fotografie v digitální podobě již pořizují.

Ačkoliv se zpočátku jednalo o výsadu specialistů, disponujících dostatečně výkonným výpočetním strojem[2], postupem času se zpracování digitálních obrazových dat stalo dostupné i širšímu okruhu uživatelů. To především díky rozmachu výpočetní techniky v posledních letech.

## 1.1 Obraz - struktura

Digitální obraz je ve své podstatě číselná reprezentace obrazových informací do paměti zařízení, které data zpracovává. Tato reprezentace v počítačové grafice může být realizována dvěma způsoby:

- Vektorová grafika 1.1.1
- Rastrové obrazy 1.1.2

### 1.1.1 Vektorová grafika

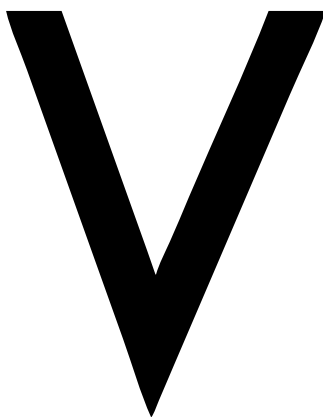
Obraz je reprezentován zdrojovým kódem, popisujícím základní útvary, které jsou matematicky definovány[3]. Mezi ně patří například body, přímky, či křivky. Výsledná podoba obrazu je následně tvořena vykreslením daných objektů. To má za

důsledek prakticky neomezenou škálovatelnost přiblížení, či oddálení obrazu. Při libovolné míře přiblížení objekty obrazu vektorové grafiky disponují hladkými okraji a nedochází k degradaci kvality obrazu[11].



Obr. 1.1: Vektorová grafika - kompletní obraz

Na následujícím obrazu je možné vidět kvalitu zobrazení dolního konce písmena „V“ ze slova „Vektorová“, uvedeném na zdrojovém obraze. Je zřejmé, že i při takto vysokém přiblížení nevznikají tzv. „zubaté hrany“ a to ani na ostrých hranách objektu.



Obr. 1.2: Přiblížení na písmeno „V“

Tyto vlastnosti předurčují vektorovou grafiku například pro tvorbu schémat, či diagramů. Nezastupitelnou roli má také v oblasti počítačové sazby a tvorbě ilustrací.

Vektorová grafika se však nehodí pro popis složitějších grafických objektů, například fotografií. Od určité meze složitosti se stává reprezentace složením geometrických objektů mnohem náročnější na výpočetní výkon stroje, než rastrové obrazy, protože se zdrojový kód stane velmi rozsáhlým pro zpracování.[11]

### 1.1.2 Rastrové obrazy

Struktura rastrových obrázků sestává z jednotlivých bodů, nazývaných „Pixely“ [3]. V závislosti na rozlišení je v obrazu přesně daný počet pixelů na výšku a na šířku. Jeden obrazový pixel je reprezentován předem určeným počtem bitů a je to nejmenší jednotka, na kterou lze rastrový obraz rozdělit.

Na první pohled nelze zpravidla rastrový obraz od vektorové grafiky odlišit (viz ukázka vektorové grafiky 1.1).



Obr. 1.3: Rastrová grafika - ukázkový obraz

Pokud ale rastrový obraz dostatečně přiblížíme, můžeme sledovat jeho strukturu. Na následujícím obrázku je zobrazen střed písmena „R“ ze slova „Rastrová“ na původním obrazu. Na hranách objektu lze jasně pozorovat rozklad na jednotlivé pixely obrazu. To je zapříčiněno tím, že pixely nejsou nekonečně malé a při dostatečné hodnotě přiblížení obrazu jsme schopni strukturu pixelů sledovat obrazech s libovolně vysokým rozlišením.



Obr. 1.4: Struktura rastrového obrazu



Zmíněná vlastnost (viditelný rozklad obrazu na jeho strukturu) se může zdát jako nevýhoda, avšak přináší mnoho možností pro manipulaci s obrazem. Jednotlivé vlastnosti pixelů jsou reprezentovány číselnou hodnotou. Jedná se o:

- Jas
- Barvu

každého jednoho pixelu. Tyto hodnoty lze jednoduše upravovat a tím ovlivňovat vlastnosti celého obrazu. Mohou sse také využít k mnoha dalším operacím, například k porovnání hodnot na příbuzném obrazu a tím zjistit míru rozdílnosti.

Existuje více druhů reprezentace obrazů. Mezi nejvýznamnější režimy patří RGB, RGBA a YCbCr. Jednotlivé modely se liší svými základními barvami a vlastnostmi. Každý z modelů má určitý způsob, jak reprezentovat přesnou hodnotu barvy každého pixelu v obrazu.

- **RGB** - Zkratka modelu vyjadřuje anglická slova Red-Green-Blue, což v překladu znamená Červená-Zelená-Modrá. Rozšířený zápis již napovídá, že se jedná o model, využívající tyto tři barvy. Kombinací těchto barev lze následně získat barvy ostatní. Jedná se o model aditivní - základní barva je černá (hodnota všech tří základních barev je 0) a pokud jsou hodnoty základních barev nastaveny na maximum, dostaneme bílou barvu[14].

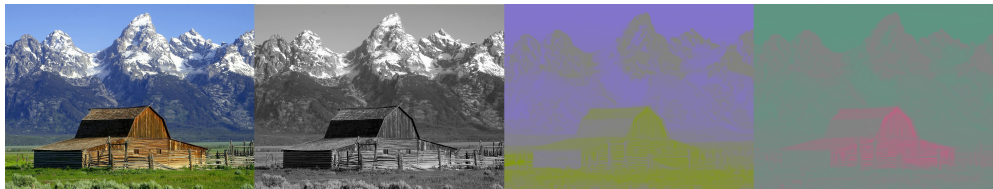


Obr. 1.5: Model RGB

Na přiloženém obrázku lze názorně vidět barevné kombinace na přechodech mezi základními barvami. Zvyšováním intenzity barev směrem do středu koláče se zvyšuje jas, uprostřed se nachází čistě bílá barva, reprezentována maximální hodnotou všech tří základních barev.

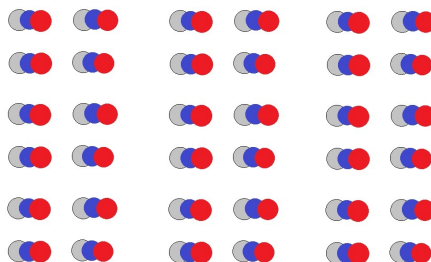
- **RGBA** - Tento model využívá stejného míchání barev jako předchozí RGB, navíc je však rozšířen o tzv. Alfa kanál (*A* ve zkratce modelu). Ten se využívá pro uchování informace o hodnotě průhlednosti pixelu. Je využíván například u formátu typu *PNG*.

- **YCbCr** - Model, využívaný právě u digitální reprezentace videa, či statických obrazů. Je zastupován opět třemi složkami. Složka  $Y$  nese jasovou informaci pixelu, kdežto  $Cb$  a  $Cr$  tvoří chrominanční složky modré a červené. Na ilustraci



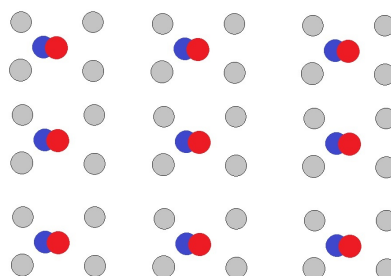
Obr. 1.6: YCbCr rozklad na složky

je zobrazen složený obraz(zleva), následuje jasová složka  $Y$ ,  $Cb$  a jako poslední složka  $Cr$ . Využívaným typem ztrátové komprese u tohoto typu modelu je podvzorkování chrominančních složek.



Obr. 1.7: Model 4:4:4 - bez eliminace složek

Nejčastěji používané podvzorkování je typu 4:2:0. Toho dosáhneme rozdělením matice obrazu do čtveřic a následně pro každou skupinu provedeme vertikální a horizontální podvzorkování složek  $Cb$  a  $Cr$ .



Obr. 1.8: Model 4:2:0 - eliminace složek  $Cb$  a  $Cr$

Zmíněné podvzorkování je sice ztrátové, ale citlivost oka není na tak vysoké úrovni, aby byla schopna podvzorkování barevných složek rozeznat jako zhoršení kvality obrazu[15].

## 2 NEURONOVÉ SÍTĚ

Jeden z výpočetních modelů, využívaných v umělé inteligenci, se nazývá „Umělá neuronová síť“. Vzorem pro návrh neuronových struktur je biologická struktura, sestávající z neuronů. Pomocí neuronové sítě jsou namodelovány neurony v umělé podobě, avšak chováním odpovídající biologické předloze[7].

Zasloužené pozornosti se neuronovým sítím dostalo v několika posledních letech. Díky svým možnostem napodobit lidský mozek k umožnění rozhodování se a vyvoznání závěrů jsou v současnosti považovány za nástroj s obrovským potenciálem. Díky jisté úrovni napodobení kreativity mozku jsou neuronové sítě v současnosti dokonce schopny tvořit nová data, či objekty[17]. Při vyhledávání na internetu lze dokonce objevit pokusy neuronových sítí o „napsání“ ještě nevydaných dílů známých knižních sérií, či pokračování, která nikdy vyjít neměla.

Tyto schopnosti jsou však zatím poměrně limitovány, nelze tedy v nejbližší očekávat rozsáhlou sbírku zatím nevydaných knih s příběhy známého mladého kouzelníka s jizvou na čele. Vývoj této technologie však směřuje pouze kupředu, a to i díky narůstající výpočetní kapacitě moderních počítačů.

### 2.1 Neuron

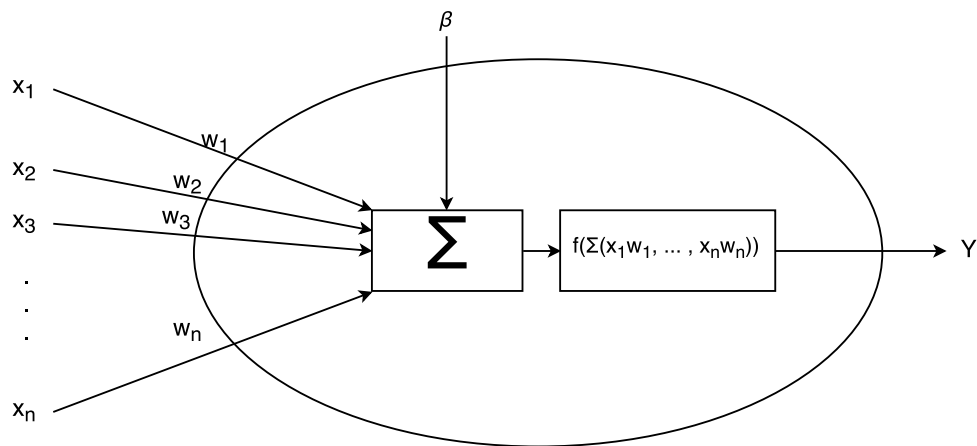
Základní stavební jednotkou je neuron. Na obrázku 2.1 lze vidět ilustrovanou strukturu tohoto objektu – McCulloch–Pittsův model[7]. Položky  $x_1$  až  $x_n$  představují vstupy, které jsou následně váženy podle jejich odpovídajících hodnot  $x_1$  až  $w_n$ . Hodnoty vah určují, do jaké míry je daný vstup významný, vzhledem k celkové množině vstupů. Následuje přičtení báze  $\beta$  (v anglické literatuře označované jako *bias*). Výsledná hodnota je poté podrobena transformaci odpovídající funkcí  $f(X)$ , čímž je získána hodnota výstupu  $Y$ , který je vždy právě jedna, nezávisle na počtu vstupů. Model neuronu lze matematicky popsat jako:

$$Y = S\left(\sum_{i=0}^N (w_i x_i) + \beta\right), \quad (2.1)$$

kde jednotlivé parametry jsou

- $x_i$  – vstupy neuronu,
- $w_i$  – váhy vstupů,
- $\beta$  – práh aktivační funkce,
- $S(\dots)$  – aktivační funkce neuronu,
- $Y$  – výstup neuronu.

Je záhodno podotknout, že ačkoliv je snaha o co nejvěrnější napodobení struktury a funkcionality biologického neuronu, nelze jejich složitosti ani v nejmenším

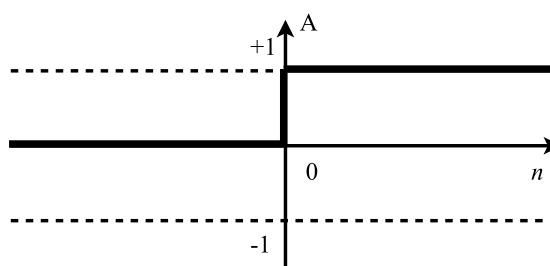


Obr. 2.1: Model neuronu[12]

srovnávat – biologický neuron má nesrovnatelně složitější strukturu, oproti modelu neuronu, využitém v neuronových sítích, a žádný biolog by se s „počítačovým“ neuronem zdaleka nespokojil. Jako při každém řešení problému je však třeba volit mezi složitostí a reálné proveditelnosti návrhu.

### 2.1.1 Aktivační funkce

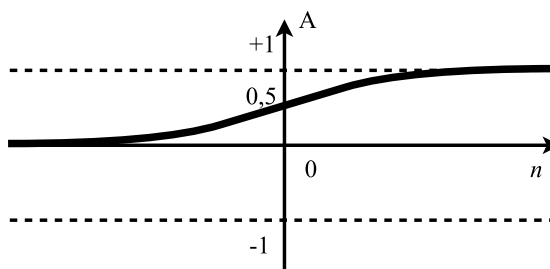
V oblasti neuronových sítí je využíváno velké množství aktivačních funkcí [20]. Tato funkce má za úkol modulaci signálu na výstupu neuronu. Výběr vhodného typu má přímý vliv na konvergenci učení sítě. Mezi nejčastěji používané aktivační funkce lze zařadit mimo jiné:



Obr. 2.2: Skoková přenosová funkce [20].

Pro skokovou přenosovou funkci platí:

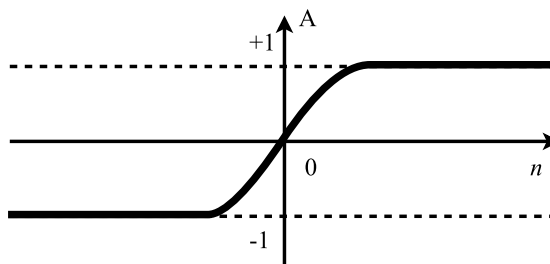
- Pro vstup menší, než daný práh –  $f(x) = 0$
- Pro vstup větší než daný práh –  $f(x) = 1$



Obr. 2.3: Sigmoidální přenosová funkce [20].

Sigmoidální přenosová funkce je velmi často využívána ve vícevrstvých neuronových sítích. Má tvar:

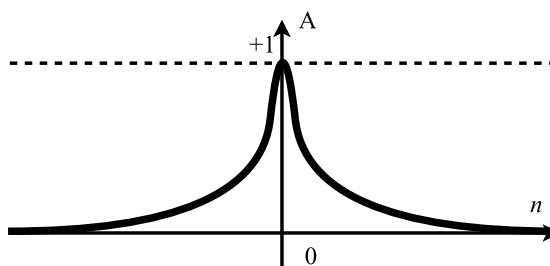
$$f(x) = \frac{1}{1 + e^{-kx}}$$



Obr. 2.4: Funkce hyperbolické tangenty [20].

Přenosová funkce hyperbolické tangenty má tvar:

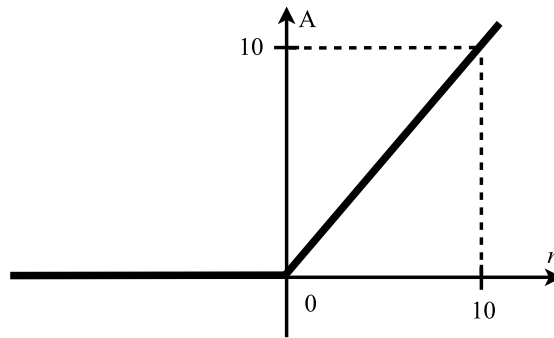
$$f(x) = \frac{2}{1 + e^{-kx}} - 1$$



Obr. 2.5: Funkce radiální báze [20].

Přenosová funkce radiální báze je ve tvaru:

$$f(x) = e^{-kx^2}$$



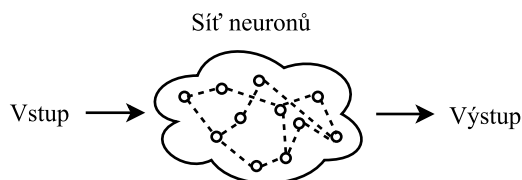
Obr. 2.6: Funkce ReLU [20].

Přenosová funkce funkce ReLU je ve tvaru:

$$f(x) = \max(0, x)$$

## 2.2 Neuronová síť

Neuronová síť představuje rozmanité propojení různého počtu objektů typu neuron 2.1 do libovolně propojené struktury. Každá neuronová síť by měla mít vstupní a výstupní část. Může se jednat o jeden, či více neuronů, tvořících vrstvu.



Obr. 2.7: Abstrakce neuronové sítě[12]

První vytvořená neuronová síť vznikla v roce 1957, vytvořena byla Frankem Rosenblattem. Jedná se o tzv. „Perceptron“, neboli síť slouženou pouze z jedné vrstvy neuronů. Jelikož taková síť dokáže rozlišit pouze lineárně separovatelné třídy[7], jsou její možnosti využití velmi omezeny.

## Druhy neuronových sítí

Druhů neuronových sítí je několik, liší se zejména určením [17]. Mezi základní patří například:

- Vícevrstvé neuronové sítě,
- Hopfieldovy sítě,
- Samoorganizující se sítě,
- Radiální báze,
- další typy.

Následující text se bude zabývat prvně jmenovanými, vícevrstvními neuronovými sítěmi. Ty jsou využívány zejména v oblastech predikce, či klasifikace.

K oživení a rozvíjení myšlenky vícevrstvných dopředných neuronových sítí dochází v několika posledních letech [17]. Změna přišla s objevem efektivního algoritmu pro stanovení sil synaptických vazeb ve vícevrstvných sítích.

### Algoritmus zpětné propagace

Zmíněnou efektivní metodou je Algoritmus zpětné propagace (anglicky označován *Backpropagation*), s jehož pomocí jsme schopni korektně upravovat jednotlivé dílčí váhy sítě[13]. Toho se dosáhne výpočtem chyby mezi výstupem modelu a očekávaným výstupem. Cílem minimalizaci právě této výstupní chyby.

První vrstva je nazvána „vstupní“. Výstupy jednotlivých neuronů se dále šíří do následující vrstvy, nazvané „skrytá“. Těchto skrytých vrstev může být větší množství, v tom případě se však již jedná o tzv. „Hluboké učení“, viz kapitola 3. Po průchodu celým systémem jsou hodnoty předány do poslední vrstvy modelu – „výstupní“.

Algoritmu zpětné propagace chyby je u tohoto modelu využito následovně:

1. Na počátku jsou váhy sítě nastaveny na náhodnou hodnotu.
2. Dopředným průchodem modelem dojde k získání výstupních hodnot
3. Rozdíl hodnot získaných a hodnot očekávaných se přenesse zpět na vstup.
4. Dojde k upravení vah modelu, aby se dosáhlo minimalizace chyby, viz 3.2.
5. Postup se opakuje, dokud není dosaženo minimální hodnoty chyby.

#### 2.2.1 Učení neuronových sítí

Jedná se o hlavní vlastnost neuronových sítí, která je využívána. Cílem je nakonfigurovat síť tak, aby se její výsledky daly považovat za nejpresnější možné [18]. K této konfiguraci se využívá parametrů neuronových sítí nazývaných „váhy“.

Jak je možné vidět na modelu neuronu – obr. 2.1, tyto váhy ovlivňují vstupní hodnoty jednotlivých neuronů. Úpravou těchto hodnot na základě nějakého parametru vlastně dochází k ukládání zkušenosti s právě prodělanou akcí sítě.

Učení neuronové sítě se následně dělí na dva další druhy. Liší se zejména zacházením s výstupem ze sítě[18]. Jedná se o:

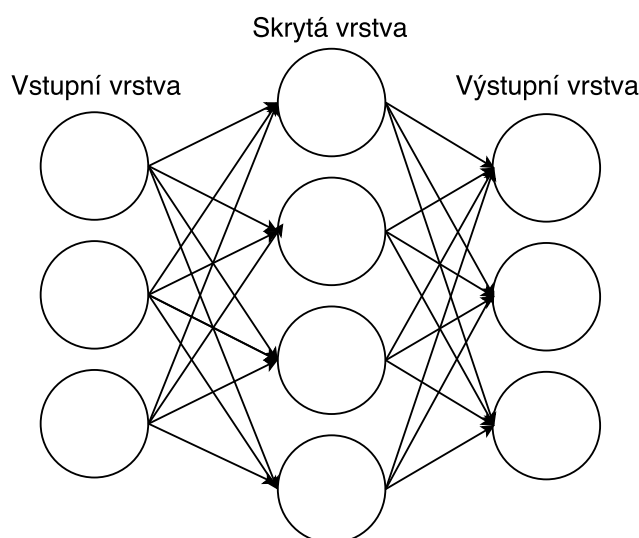
- **Učení s učitelem** – Neuronové síti se předloží vzor(data + požadovaný výsledek, označován jako „label“. Váhy sítě jsou zpočátku nastaveny na výchozí hodnotu. Po průchodu sítí je získána chyba, udávající rozdíl mezi očekávaným výstupem a reálným výstupem. S využitím zpětné vazby je následně vypočítána korekce a dojde k upravení vah, či prahů sítě, aby byla chyba minimalizována. Tento postup se opakuje, dokud není dosaženo požadované minimální chyby. Princip je nastíněn již v odstavci, věnovaném algoritmu zpětné propagace 2.2.
- **Učení bez učitele** – Při učení bez učitele není brán v potaz výstup neuronové sítě. Třídění probíhá podle předložených vzorů. Cílem je najít společné vlastnosti a skrytou strukturu v datech.



### 3 HLUBOKÉ UČENÍ

Pokud se mluví o tzv. „Hlubokém učení“ (anglicky „Deep learning“), je třeba vědět, že se ve své podstatě jedná o specifický druh strojového učení[4], spadající do oblasti vývoje umělé inteligence. Tato disciplína přináší výpočetním strojům možnost a dovednost učit se danou problematiku[5].

Cílem metod hlubokého učení je pak simulovat funkci lidského mozku. Toho je docíleno modelováním sítě navzájem propojených neuronů(viz 2.2), předávajících si signály pomocí impulzů. Co však může být na učení hlubokého ? „Hlubokost“ spočívá v existenci libovolného počtu tzv. skrytých vrstev neuronů[6].



Obr. 3.1: Neuronová síť[12]

Podle jiných zdrojů může být hlubokost odvozena spíše z faktu, že požadovaná znalost je skryta hluboko ve vstupních datech a nelze jí snadno dosáhnout[16]. Obě interpretace pojmu „Hluboké“ však vystihují podstatu problematiky a její charakter. Hluboké učení se využívá tam, kde běžné metody získání dat selhávají.

#### 3.1 Historie

Vznik první neuronové sítě je datován do roku 1957. jak již bylo popsáno v kapitole 2.2. Jednalo se však o pouhý perceptron. Pojem „hluboké učení“ (v literatuře lépe vyhledávatelné pod anglickým názvem „deep learning“, česká obdoba tohoto pojmu prozatím není zcela ustálena) byl představen Rinou Detcher roku 1986 [6]. První

experimenty s těmito vícevrstevnými sítěmi s několika skrytými vrstvami však nebyly efektivní.

Důvod neefektivnosti byl prostý – nedostatek výpočetního výkonu. Tehdejší výpočetní stroje nedokázaly v reálném čase řešit složitost takto strukturovaných neuronových sítí.

V odborné literatuře se často uvádí historie hlubokého učení v bodech, či milnících, které měnily směr vývoje tohoto odvětví strojového učení [18]. Pomáhá to získat nadhled a pochopení vývoje problematiky.

1. Hluboké učení má dlouhou historii, zpočátku se však pohybovalo v rovině filozofické. Název i definice se lišily podle různých směrů pohledu.
2. S nárůstem dostupných trénovacích dat začalo hluboké učení dostávat svůj smysl (požadavek na velký objem trénovacích dat).
3. Postupným vývojem stále výkonnějších výpočetních kapacit se stávalo hluboké učení „realizovatelnějším“.
4. Hluboké učení je postupem času schopnější řešit komplexnější a složitější problémy se zvyšující se přesností.

## 3.2 Hluboké neuronové sítě

Hluboké neuronové sítě jsou v základním principu stejné, jako jiné umělé neuronové sítě, navíc však disponují určitým počtem skrytých vrstev mezi vstupními a výstupními. To jim umožňuje modelovat komplexní nelineární problémy [16]. Vyšší počet skrytých vrstev může potenciálně umožnit tvořit složitější modely.

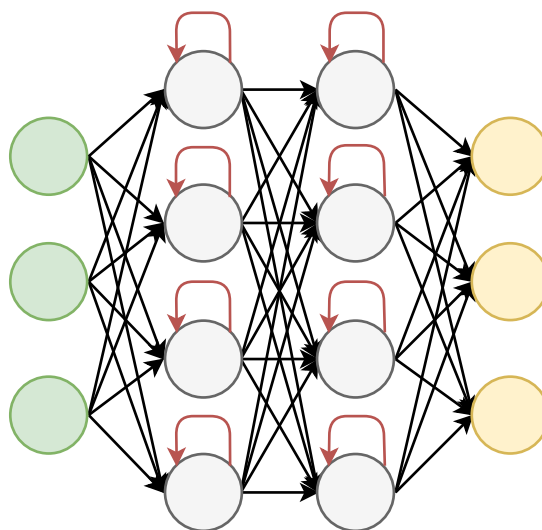
Mezi základní hluboké neuronové sítě lze zařadit:

- RNN – Rekurentní neuronové sítě (angl. Recurrent Neural Networks)
- CNN – Konvoluční neuronové sítě (angl. Convolutional Neural Networks)

### 3.2.1 Rekurentní neuronové sítě

Bezstavové neuronové sítě – to má za důsledek různé chování při prohození pořadí vstupních dat [12]. Jendnotlivé neurony získávají nejen informace z předešlé vrstvy, ale také výstupní informace jich samých (odtud označení „rekurentní“).

Využití rekurentních sítí je všude, kde nedochází k závislosti na průběhu v čase (například hudba, či video). Lze však síti předat data, reprezentována jako sekvenci. Obecně lze tento typ sítí použít například pro kompletaci dat, nebo doplnění informace (například automatické dokončování, dopočítání chybějící informace na fotografii a tím její doostření).



Obr. 3.2: Rekurentní neuronová síť

### 3.2.2 Konvoluční neuronové síť

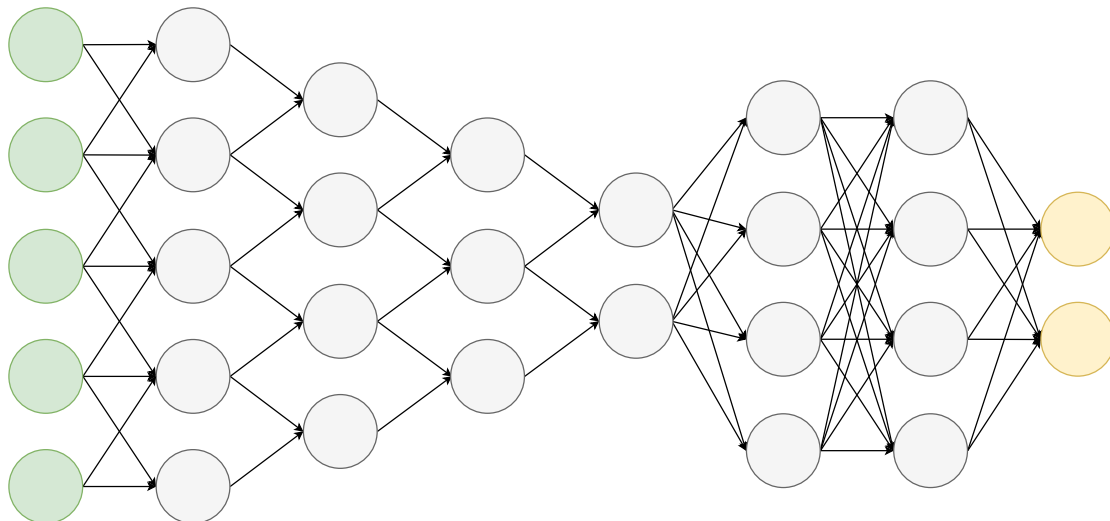
Rozpoznávání objektů v obrazech se pro klasické algoritmy zdá být věcí téměř neřešitelnou problematikou. Zajímavým porovnáním je fakt, že malé dítě je schopné identifikovat automobil na předložené fotografii, kdežto algoritmy nejsou schopny automobil nalézt, jen co se lehce změní jeho podoba, či poloha na fotografii vůči předloženému vzoru [19].

Řešením tohoto problému mohou být právě neuronové síť (ze svého principu napodobování funkce mozku).

Speciálním případem, vhodným na práci s objekty v obrazech jsou právě konvoluční síť, někdy také nazývané *hluboké* konvoluční neuronové síť se od jiných sítí částečně liší. Hlavním znakem je využití tzv. „konvolučních vrstev“, popsaných v následující kapitole 3.2.3. Na obrázku 3.3 lze vidět schématickou podobu struktury konvoluční síť. Jako každá jiná, obsahuje i tento typ síť vstupní vrstvu (vyznačená zeleně). Obdobně lze na modelu vidět vrstvu výstupní (oranžová barva). Dále jsou v modelu obsaženy skryté vrstvy (šedá), plnící požadované funkce.

- **Vstupní vrstva** – Obvykle nemá stejnou velikost, jako vstupní data (pro obrázek s rozměry 300 x 300 pixelů by musela mít 90 000 vstupních uzlů). Místo toho obsahuje počet uzlů, tvořící poměrnou část celkového počtu uzlů.
  - Pokud je počet vstupních uzlů 30 x 30, je prvním průchodem zpracována pouze odpovídající část vstupního obrazu (obvykle levý horní roh).
  - V následujícím průchodu je toto „okno“ posunuto o požadovaný počet pixelů. Tento proces se opakuje do zpracování celého vstupního obrazu.

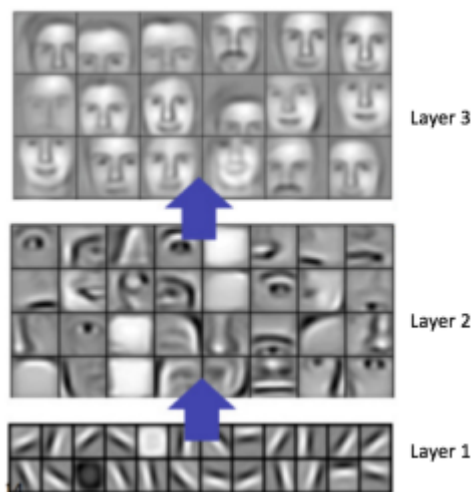
- **Skryté vrstvy** – mají za úkol zpracování, či úpravu dat podle požadavků. Mohou být další konvoluční vrstvy, nebo dále:
  - 3.2.5 Sdružovací vrstvy (angl. Pooling layers) – například vrstva MaxPooling. Plní funkci filtrace detailů.
  - 3.2.4 ReLu vrstvy (angl. ReLu layers) – nelineární/aktivační vrstvy.
  - 3.2.6 Vrstvy zahazování (angl. Dropout layers) – vrstvy „zahazující“ určitou část dat, pro předcházení tzv. *přetrénování*.
  - 3.2.7 Plně propojené vrstvy (angl. Fully connected layers) – vrstvy, kde jsou všechny neurony propojeny se všemi neurony následující vrstvy.
 Čím větší je hloubka těchto vrstev, tím se snižuje počet uzlů jednotlivých vrstev.
- **Výstupní vrstva** – Může mít mimo jiné za úkol klasifikační funkci, tzn. rozdělovat výsledky do předem definovaných tříd a následně určovat s jakou pravděpodobností objekt spadá do dané třídy.
  - Názorné provedení klasifikace je model neuronové sítě, který má rozhodnout, zda je na fotografii kočka, nebo pes. Výstupní vrstva je tvořena dvěma uzly, kde každý zastupuje jednu třídu (třída „kočka“, nebo třída „pes“). Výstupem je pravděpodobnost, se kterou objekt spadá do odpovídající třídy.



Obr. 3.3: Schéma konvoluční neuronové sítě [19].

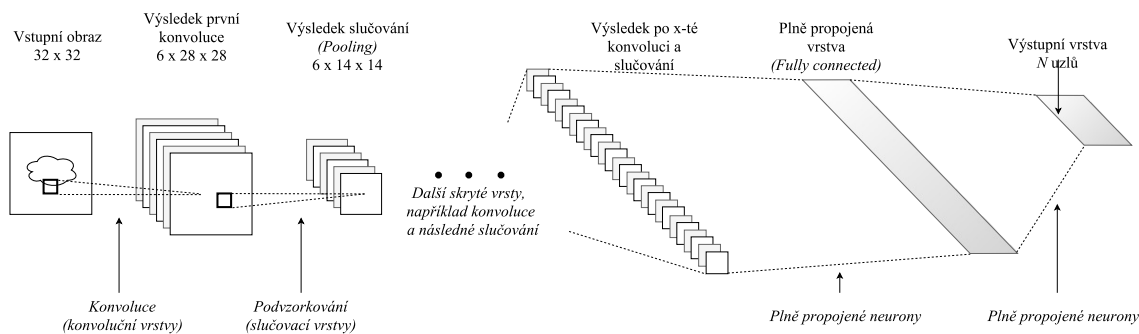
Na obrázku 3.5 lze vidět jeden z možných způsobů realizace konvoluční neuronové sítě. Vstupem do modelu je obraz o rozměru 32 x 32 (šířka x výška). Po průchodem první konvoluční vrstvou získáme 6 aktivačních map o rozměrech 28 x 28. Následuje podvzorkování s využitím slučovací vrstvy.

Model může dále obsahovat další skryté vrstvy stejného charakteru. S každou další skrytou konvoluční vrstvou se mění zpracování vstupních dat.



Obr. 3.4: Počet konvolučních vrstev.[19]

Výstup modelu je obvykle tvořen plně propojenými vrstvami. Výstupy z předchozích vrstev jsou převedeny do vektoru hodnot. Jelikož se v tomto případě jedná o model, který má za úkol klasifikaci, je poslední vrstva zaměřena právě tímto úkolem. Počet uzlů této vrstvy je  $N$ , což je počet tříd, do kterých mají být data klasifikována.

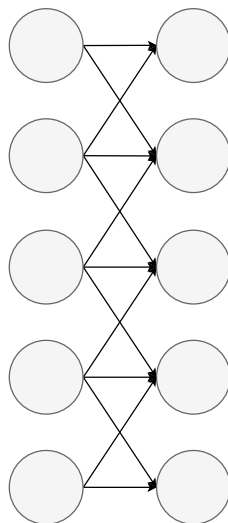


Obr. 3.5: Příklad realizace konvoluční neuronové sítě

### 3.2.3 Konvoluční vrstvy

Na této vrstvě je prováděna konvoluce vstupních dat. Jejím účelem je získání vlastností vstupního obrazu. Konvoluce zachovává prostorové vztahy mezi pixely, *naučí* se obrazové funkce z malých čtverců, na které je rozdělen vstupní obraz. Výstupem jsou aktivační mapy, vstupující do následující vrstvy modelu [18].

Typická vlastnost vrstvy je spojení dílčích uzlů. Jednotlivé neurony jsou navázány pouze na následující, s kterými přímo sousedí (na rozdíl od plně propojených vrstev, kde jsou neurony propojeny každý s každým).

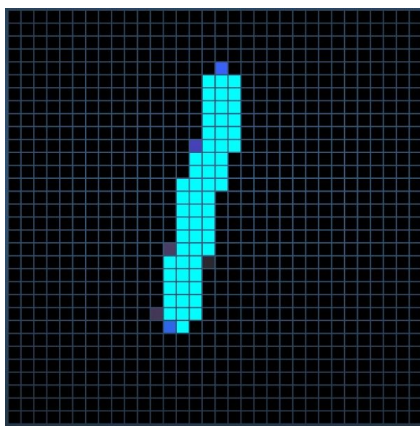


Obr. 3.6: Konvoluční vrstvy - propojení neuronů [18].

Fungování této vrstvy je založeno na klasické 2D konvoluci. Vstupními parametry jsou:

- Rozměry vstupního obrazu  $w, h, c$ 
  - $w$  – šířka
  - $h$  – výška
  - $c$  – počet kanálů (3 pro RGB obraz)
- Velikost filtru  $n$
- Dimenze jádra  $d$

Data vstupního obrázku první vrstvy jsou o velikosti  $32 \times 32$ , tedy  $w = h$ :



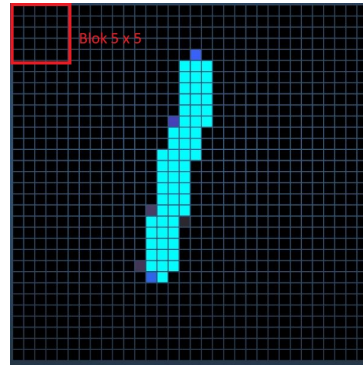
Obr. 3.7: Vstupní data do konvoluční vrstvy

Velikost filtru je určena na  $n = 5$  a máme dannou sadu filtrů:



Obr. 3.8: Sada filtrů konvoluční vrstvy

V prvním kroku je získán blok ze vstupního obrazu o rozměrech filtru  $n \cdot n$ . ( $n = 5$ ):



Obr. 3.9: Blok vstupních dat 5 x 5

Následně je spočtena konvoluce tohoto bloku a příslušného filtru (první filtr ze sady na obr. 3.2.3):



Obr. 3.10: Konvoluce prvního bloku

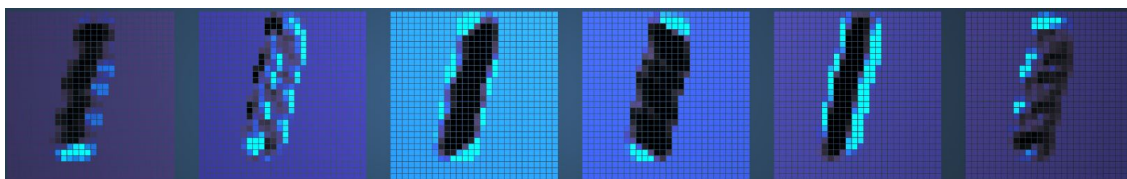
Tím je získána jediná hodnotu pixelu, která je uložena do výsledné aktivační mapy. Následně dojde k změně výběru bloku 5 x 5 ze vstupního obrazu. Výběr 5 x 5 je posunut o jeden pixel doprava. Následně je opět provedena konvoluce s dříve použitým filtrem. Při dosažení konce řádku je nápodobně posunut výběr bloku o řádek pixelů níže a proces posuvu v horizontálním směru se opakuje. Tím dostaneme aktivační mapu o rozměrech  $k \times k$ . Jelikož  $w = h$ , tak:

$$k = w - n + 1 \quad (3.1)$$

V našem případě tedy:

$$k = 32 - 5 + 1 = \mathbf{28} \quad (3.2)$$

Opakováním pro celou sadu filtrů dostáváme 6 aktivačních map pro jeden vstupní obraz. Tato sada aktivačních map je výstupem konvoluční vrstvy:



Obr. 3.11: Aktivační mapy

Ilustrované zobrazení průběhu zpracování dat v konvolučních, ale i ostatních vrstvách lze názorně prostudovat pomocí online nástroje <http://scs.ryerson.ca/~aharley/vis/conv/flat.html>, simulujícího průběh dat reálnou konvoluční sítí.

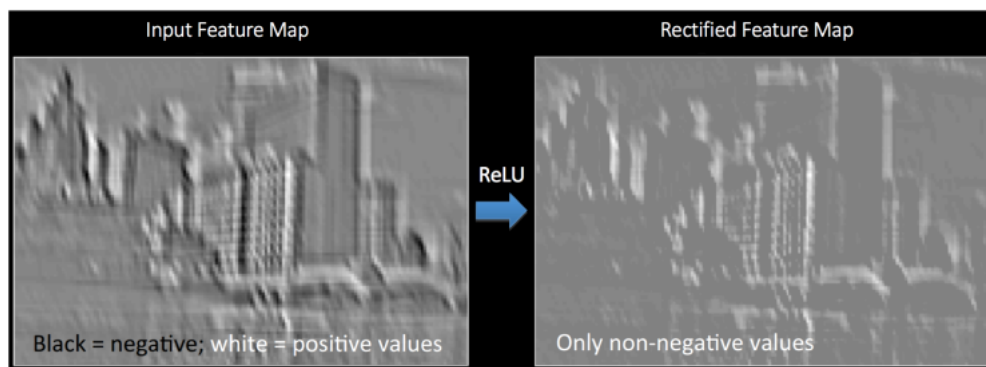
### 3.2.4 ReLu vrstvy

Jedná se o typ vrstvy typicky využívané právě po konvoluční vrstvě. Označení „ReLU“ je zkratkou anglického názvu *Rectified Linear Units*[18].

Výpočet funkce je velmi jednoduchý:

- Pro vstup funkce  $x < 0$  platí –  $f(x) = 0$
- Pro vstup funkce  $x \geq 0$  platí –  $f(x) = x$

Jinak řečeno, všechny záporné hodnoty jsou nastaveny na 0, kladné hodnoty jsou ponechány na stejné hodnotě.



Obr. 3.12: Příklad využití ReLU vrstvy - odstranění záporných hodnot [21]

Na příkladu 3.12 je možné sledovat, že provedením funkce ReLU jsou z původních dat odstraněny černé oblasti, zastupující záporné hodnoty.

Průběh je zobrazen v grafu na obrázku 2.6. Funkce je aplikována na prvek po prvku výstupu předchozí vrstvy. Jelikož nejsou prováděny žádné složitější matematické operace, jako násobení, dělení, umocňování, má jednoduchost tohoto výpočtu za následek zvýšení rychlosti průběhu trénování celé neuronové sítě.



### 3.2.5 Sdružovací vrstvy

Prostorové sdružování, prováděné sdružovacími vrstvami, snižuje dimenzi vstupní aktivační mapy. Jedná se o ztrátové podvzorkování, avšak výstupní data jsou složena z nejvíce významných informací. Tím se výsledná neuronová síť stává odolnější vůči minoritním transformacím vstupních dat. Také umožňuje síti být tzv. *equivariantní*. Díky tomu lze detekovat objekt nezávisle na jeho poloze ve vstupním obrazu [18]. Sdružování může probíhat podle:

- Maximální hodnota
- Průměrná hodnota
- Součet
- ...

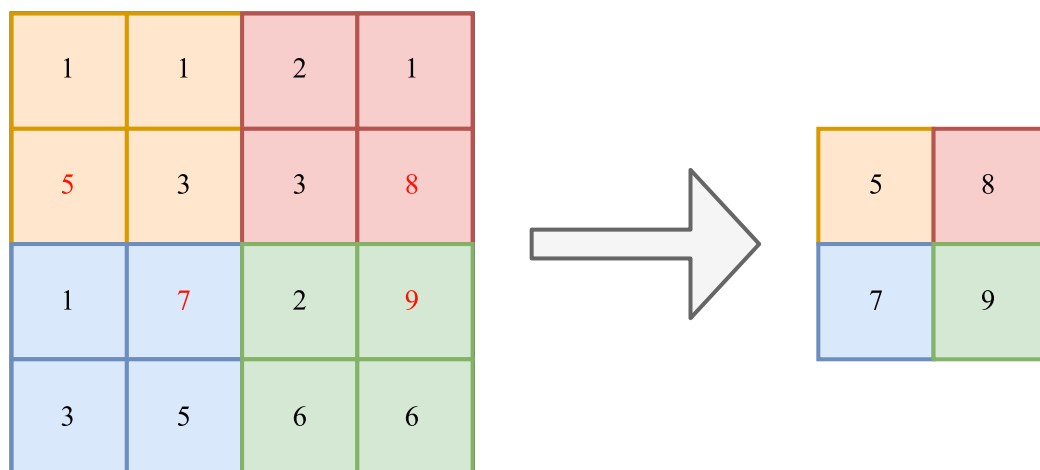
Princip fungování bude popsán na sdružování podle maximální hodnoty. Tento postup lze v odborné literatuře dohledat pod pojmem MaxPooling.

Vstupními parametry operace jsou:

- Velikost okna  $\sigma$
- Velikost kroku  $\alpha$

Pokud je uvažováno:  $\sigma = 2, \alpha = 2$ , tak je postup následující:

1. Je vybráno okno  $\sigma \cdot \sigma$  s počátkem na výchozí pozici.
2. Nejvyšší hodnota, nalezená v matici hodnot je výsledkem tohoto okna.
3. Výchozí pozice pro výběr okna je posunuta přesně o  $\alpha$  pixelů.
4. Na nové výchozí pozici je opět vybráno okno o velikosti  $\sigma \cdot \sigma$ , ve kterém je vyhledána nejvyšší hodnota.
5. Postup je opakován, dokud není zpracována celá vstupní matice.



Obr. 3.13: Sdružování podle maximální hodnoty [22]

### 3.2.6 Vrstvy zahazování

Zahazování je regulační technika, kdy jsou některé z neuronů sítě při procházení deaktivovány. Tato metoda se využívá k tzv. *přeučení* sítě [18], kdy model ztrácí generalizaci a stává se moc konkrétním. Není pak schopen reagovat na mírné změny a není schopen modulárně řešit zadané problémy.

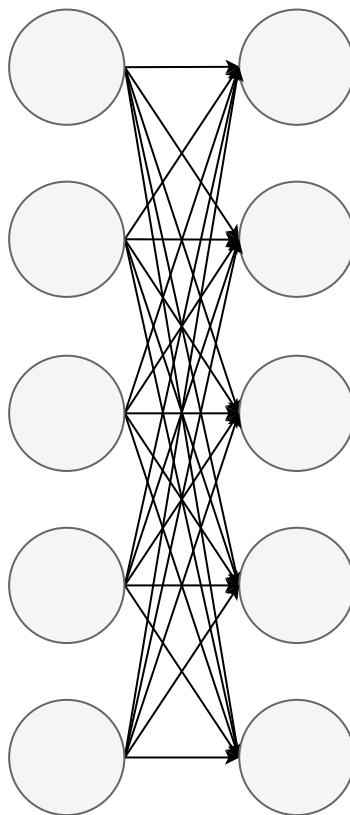
Aplikací zahazování jsou zvýšeny ztráty při trénování modelu, je však dosaženo více generalizace.

Zahazování se klasicky aplikuje na plně propojené vrstvy modelu 3.2.7. Lze jej však také aplikovat za vrstvy sdružování 3.2.5, pro simulaci šumu.

### 3.2.7 Plně propojené vrstvy

Jedná se o klasický vícevrstvý perceptron. Pojem „plně propojené“ znamená, že všechny neurony vrstvy jsou spojeny se všemi neurony vrstvy předchozí a vrstvy následující.

Výstupem konvolučních vrstev 3.2.3 a vrstev sdružování 3.2.5 jsou prvky vysoké úrovně vstupního obrazu. Plně propojené vrstvy mají za úkol klasifikaci vstupního obrazu do tříd na základě trénovacího datasetu viz příklad modelu na obrázku 3.5.



Obr. 3.14: Plně propojené vrstvy [18]

### 3.2.8 Optimalizační algoritmy

Optimalizační algoritmy v hlubokých neuronových sítích slouží k úpravám vah a bází 2.1. Při trénování modelu sítě je na výstupu vypočítána chyba, představující rozdíl mezi hodnotou očekávanou a hodnotou skutečnou. Váhy a báze systému představují „učitelné“ parametry. S využitím těchto algoritmů lze docílit minimalizace chyby na výstupu modelu úpravou zmíněných vlastností sítě. Mezi využívané algoritmy mimo jiné patří:

- Gradient descent
- Adam
- ADADELTA
- Stochastic
- ...

#### Gradient descent[23]

Česky algoritmus gradientního sestupu. využívá smyčku, ve které jsou vyhledávány směry s hodnotu, blížící se cíli. Pokud existuje více stejných cest, je vybrána náhodně jedna z nich.

Algoritmus se může nesprávným náhodným výběrem dostat do lokálního minima, ačkoliv při výběru jiné cesty za lepším výsledkem by mohl dosáhnout globálního minima, které by mělo lepší hodnotu.

Při dosažení lokálního minima lze využít opětovného hledání cesty z náhodného nižšího bodu. Tím je možné získat sérii různě úspěšných výsledků.

Počet iterací se může lišit podle toho, zda již bylo dosaženo požadované prahové hodnoty, nebo může být omezen uživatelem. Vlastností algoritmu gradientního sestupu je konstantní míra učení během trénování modelu.

#### Adam[23]

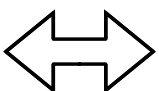
Zkratka názvu *Adaptive moment estimation*. Jedná se o další algoritmus, umožňující úpravu parametrů neuronové sítě k dosažení nejlepšího výsledku. Oproti algoritmu gradientního sestupu, disponujícím konstantní mírou učení je algoritmem Adam upravována každá váha sítě odděleně.

## 4 METODY MĚŘENÍ PODOBNOSTI

### 4.1 Porovnání na základě struktury obrazu

Porovnávat několik obrazů lze na základě znalosti struktury, jakou jsou obrazy reprezentovány v datech. Pokud je obraz uvažován jako matice číselných hodnot, nazývaných pixely 1.1, lze o jejich podobnosti velmi jednoduše rozhodnout.

1	1	2	3
5	8	13	21
34	55	89	144
233	377	610	987

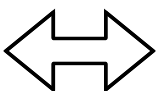


1	1	2	3
5	8	13	21
34	55	89	144
233	377	610	987

Obr. 4.1: Dvě číselné matice – jednoduché porovnání

Když je třeba řešit problém vyhledávání objektu v obrazu (například podmatici  $[(8,13)(55,89)]$ ), jeví se následující matice jako částečně rozdílné, hledaný objekt však stále obsahují obě:

1	1	2	3
5	8	13	21
34	55	89	144
233	377	610	987



8	13	2	3
55	89	1	21
34	5	8	144
233	377	610	987

Obr. 4.2: Matice se společným objektem

Zde však pořád lze aplikací klasických algoritmů vyhledat požadovaný objekt a stanovit, že obě matice hledanou podmaticí obsahují.

Pokud se však například změní osvětlení, jeví se obě matice naprosto rozdílně:

1	1	2	3
5	8	13	21
34	55	89	144
233	377	610	987

↔

2	2	3	4
6	9	14	22
35	56	90	145
234	378	611	988

Obr. 4.3: Rozdílné matice

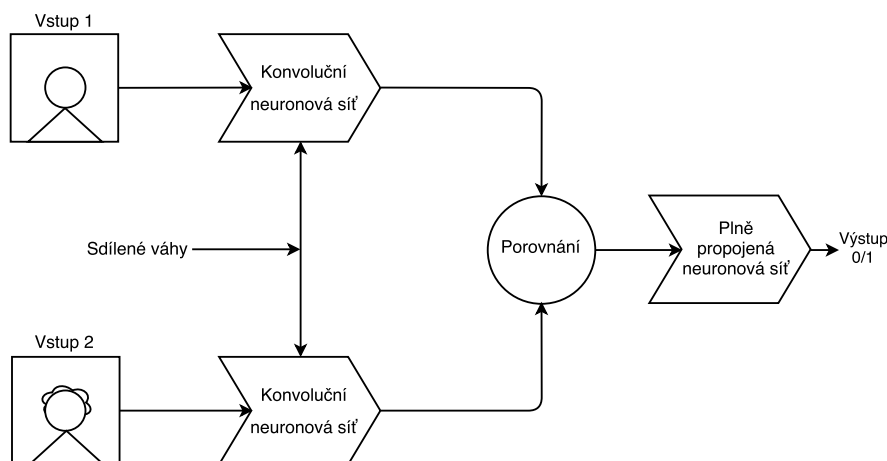
## 4.2 Využití hlubokého učení

Vhodným řešením ke generalizaci porovnávání obrazu je vytvoření modelu hluboké konvoluční neuronové sítě 3.2.2. Vytvořením vhodného modelu složeného z konvolučních vrstev 3.2.3 a vrstev sdružování 3.2.5, zakončených plně propojenými vrstvami 3.2.7 lze získat 1D vektor hodnot, uvádějící významné vlastnosti vstupního obrazu.

Pokud je tomuto postupu podrobena více obrazových dat, je možné jejich výsledné vektory významných vlastností za použití vhodné metody porovnat. Tím je možné stanovit míru podobnosti daných dat.

## 5 NÁVRH ŘEŠENÍ

Navržené řešení sestává z využití tzv. *Siamských neuronových sítí*. Tento typ modelu pracuje na principu sdílených vah. Sdílené filtry jsou aplikovány na různá vstupní data, čímž jsou všechny obrazy, vstupující do systému zpracovány stejně.



Obr. 5.1: Schéma modelu siamských sítí

Na schématu představují objekty „Vstup 1“ a „Vstup 2“ rozdílná obrazová data. Může se jednat o stejné fotografie té samé osoby. Stejně tak se však může jednat o naprosto rozdílné obrazy, kdy na jednom bude fotografie obličeje a na druhém bude zachycena krajina.

Data vstupují do příslušných větví shodných konvolučních neuronových sítí, tvořených sekvencí konvolučních vrstev 3.2.3 a vrstev sdružování 3.2.5. Tyto vrstvy mají sdílené hodnoty filtrů mezi větvemi. Jednotlivé větve jsou pak ukončeny plně propojenými vrstvami 3.2.7. Ty mají za úkol jednotlivé aktivační mapy převést na výstupní vektory hodnot s významnými vlastnostmi.

Hlavním aspektem této části celkového modelu je aplikace shodného postupu zpracování na (potenciálně) rozdílná vstupní data. Z principu fungování sdružovacích vrstev budou výstupní vektory dat obsahovat významné aspekty vstupního obrazu.

Podle struktury výstupních dat jednotlivých větví konvolučních sítí se již nabízí další krok celkového modelu a to porovnání těchto výstupních vektorů hodnot.

Zvolením vhodné porovnávací metody jsou výstupní vektory předchozích vrstev porovnávány.

Následná síť plně propojených vrstev má za úkol minimalizovat kontrastní ztráty mezi vektory. Výsledkem je číslo (0 nebo 1), udávající podobnost vstupních dat. Pro hodnotu 0 platí, že vstupní obrazy nejsou podobné, 1 znamená, že vstupní obrazy jsou podobné.

## 6 PROSTŘEDKY K OVĚŘENÍ METODY

K ověření zvolené metody je třeba využít nástroje, pomocí kterých bude model sítě namodelován, otestován a výsledky budou vizuálně prezentovány. Nástroji se rozumí programy, umožňující tvorbu a práci s hlubokými neuronovými sítěmi.

Trénování navrženého modelu je třeba provést na určitém počtu vzorových obrazových dat, nazývaných *dataset*.

Testování modelu (tj. jak úspěšně byl natrénován) je prováděno na nezávislých validačních datech. Je nutné ošetřit, aby trénovací a validační data byly dvě oddělené entity a nedocházelo k jejich průniku. To by mohlo ovlivnit důvěryhodnost výstupního hodnocení navržené a realizované metody.

Výsledné hodnoty, určující kvalitu modelu jsou tedy:

- **Přesnost odhadu na validačních datech** [%]
- **Přesnost dohadu na testovacích datech** [%]

, kde nejdůležitější parametr je zejména přesnost na testovacích datech. Ta určuje, zda je model využitelný i u obecného řešení problému a není specializován pouze na množinu, kterou byl trénován.

### 6.1 Použité programy a nástroje

K realizaci, testování a následné vizualizaci vlastností modelu jsou použity následující programy a nástroje:

- Nástroje programovacího jazyka Python <https://www.python.org/>
- Pycharm - Vývojové prostředí pro programovací jazyk Python. Na potřeby tohoto projektu byla využita Community verze programu. Více informací na internetové adrese <https://www.jetbrains.com/pycharm/>.
- TensorFlow – Open-source knihovna pro práci s vícerozměrnými datovými maticemi (tenzory). Tato dovednost je široce využívána při vývoji aplikací umělé inteligence. Dostupné na <https://www.tensorflow.org/>.
- Keras – Aplikační rozhraní, vytvořené pomocí jazyka Python, pracující nad knihovnami TensorFlow, Theano a CNTK. Bylo vyvinuto se zaměřením na umožnění rychlého experimentování. Umožňuje práci jak s rekurentními, tak i konvolučními neuronovými sítěmi. Podporuje běh na CPU i GPU. Více informací a kompletní dokumentace na adrese <https://keras.io/>.

## 6.2 Použitý dataset

Jako vhodný dataset byla zvolena sada obrazů obličejů osob s názvem ExtendedYaleB. Jedná se o fotografie 28 osob. Ke každé, z osob náleží 585 fotografií. Vzorčky jednotlivých osob se různí v několika aspektech:

- Osvětlení osoby na fotografii (mírné změny)
- Pozadí za osobou (výrazné změny)
- Výraz osoby (mírné změny)

Celkem je tedy dostupných 16 380 obrazových dat, využitelných k trénování modelu, či validaci již natrénovaného modelu.



Obr. 6.1: Dataset – příklad první osoby



Obr. 6.2: Dataset – příklad druhé osoby

Uvedené příklady představují prvních 6 vzorků pro první 2 osoby. Celkový náhled na dataset nelze z pochopitelných důvodů uvést.

Pro účely trénování a následného testování vytvořeného modelu byl počet osob rozdělen na tři skupiny: Skupiny, využitě v průběhu trénování modelu:

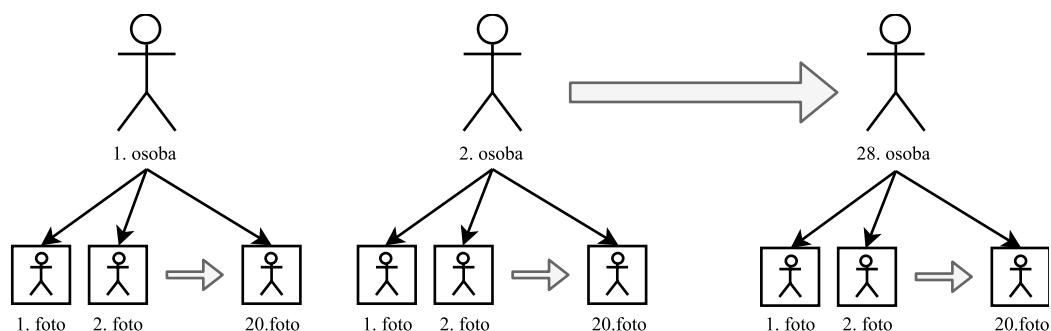
1. **Trénovací skupina** – 1. až 10. osoba
2. **Validační skupina** – 10. až 20. osoba
3. **Testovací skupina** – 20. až 28. osoba

Pro každou osobu ve skupině poté bylo vybráno prvních 20 vzorků snímaných fotografií. Struktura výsledného použitého datasetu je tedy složena ze tří skupin, kde každá skupina obsahuje určitý počet osob po 20 vzorcích, jak je znázorněno na schématu 6.3.

Omezení, kdy se bere pouze prvních 20 vzorků od každé osoby bylo zavedeno zejména pro omezení velikosti vstupního datasetu. Při trénování navrženého a vytvořeného modelu neuronové sítě jsou vytvořeny náhodné kombinace dvojic fotek, určených k porovnávání.

- **Trénovací data** - tvoří dvojice obrazových vstupů s hodnotou podobnosti, udanou číslem 0, či 1.





Obr. 6.3: Struktura použitého datasetu

- **Validační data** - páry obrazových dat s udanou číselnou hodnotou podobnosti 0, nebo 1, avšak zde dochází v průběhu tréninku modelu k testování a následnému vyhodnocení, zda funkce modelu odhadla správnou hodnotu.
- **Testovací data** - mají stejnou funkci, jako validační data (tzn. dvojice obrazových dat s udanou správnou hodnotu), avšak liší se původem - neobsahují data, shodná s daty ostatních (trénovací a validační) skupin. Aplikace testovacích dat probíhá až po ukončení trénování modelu.

Tvorbou kombinací dochází k vysokému nárůstu objemu dat ke zpracování při průběhu trénování modelu. Každé trénování je však omezeno výpočetním výkonem použitého stroje.

## 7 REALIZACE NEURONOVÉ SÍTĚ

Model neuronové sítě byl navržen na základě principu siamských neuronových sítí se sdílenými váhami. Dochází k identickému zpracování obou vstupních obrazových 2D matic. Výstup z větví je poté porovnáván ke zjištění shody výstupů jednotlivých větví.

### 7.1 Hlavní části a jejich proměnné parametry

Celý model se skládá ze dvou hlavních částí, dělicích se dále na menší stavební bloky. Hodnoty, udané symboly řecké abecedy představují proměnné, jejichž hodnota se mění v rámci testování ve snaze získat model s nejlepšími parametry. Části:

1. **Dvě oddělené větve**, představující samostatné konvoluční sítě. Tyto větve tvoří „siamskou“ část celého modelu, jelikož sdílejí stejné hodnoty jednotlivých vah. Struktura obou větví je identická, vyjmenovaná následně:

(a) **Konvoluční vrstva** (*Conv2D*) s parametry:

- Rozměry vstupních dat do první konvoluční vrstvy:
  - Výška = 224
  - Šířka = 224
  - Bitová hloubka = 24 ( $3 \times 8$ , tj. 3 barvy)
- Velikost jádra –  $\gamma$
- Počet filtrů –  $\mu$
- Typ aktivační funkce – *ReLU*

(b) **Vrstva sdružování** (typu MaxPooling), parametry vrstvy:

- Velikost bloku ke sdružení –  $\sigma$
- Velikost kroku –  $\kappa$

Po různě strukturované sérii těchto vrstev následuje vrstva typu **Flatten**, následovaná plně propojenou vrstvou (**Dense**)

- Dimenze plně propojené vrstvy – 500

2. **Následující část je společná** pro výstupní data z obou větví::

(a) **Vrstva pro provedení operace** „kosinová podobnost dvou vektorů“ nad oběma výstupními vektory ze siamských větví (*Lambda* vrstva).

(b) **Plně propojená vrstva** (*Dense*).

- Dimenze – 128
- Typ aktivační funkce – *ReLU*

(c) **Plně propojená vrstva** (*Dense*)

- Dimenze – 2
- Typ aktivační funkce – *Softmax*

Poslední vrstvou modelu je plně propojená vrstva s dimenzí **2** – tato dimenze reprezentuje dva možné výstupy modelu a to:

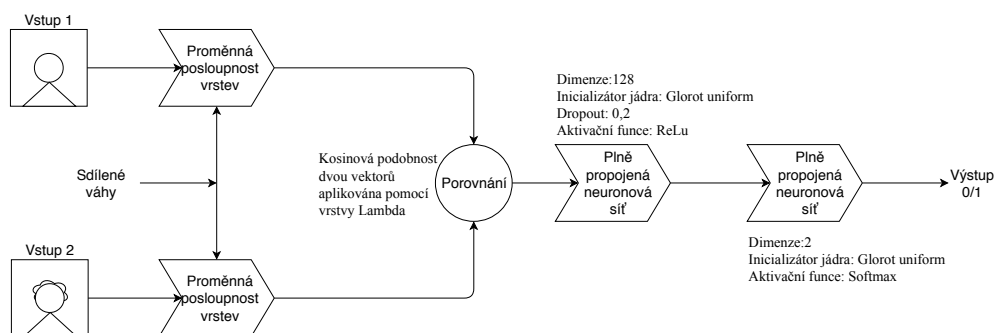
- 1 – vstupní data jsou vyhodnocena jako shodná.
- 0 – odhad zní, že vstupní data jsou rozdílná.

Kapitola 7.2 názorně zobrazuje jednotlivé návrhové modely.

## 7.2 Struktury modelu

Struktura modelů a nastavení parametrů vrstev může být libovolné. V tomto případě byly zvoleny následující struktury modelů z důvodu vhodnosti využití konvolučních neuronových vrstev ke zpracovávání obrazových dat.

Rozdílnost testovaných struktur je v posloupnostech a parametrech jednotlivých siamských větví. Následující (společná) část, zodpovědná za porovnání výstupních dat z větví je společná pro všechny návrhové modely.



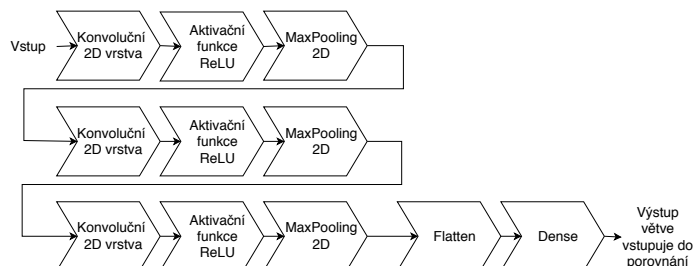
Obr. 7.1: Obecný model

Množství testovaných konfigurací jednotlivých struktur modelů se odvíjí od jejich dosažené přesnosti na testovacích i validačních datech.

- Pokud se přesnost modelu na testovacích a validačních datech pohybuje okolo 50%, je tento model vyhodnocen jako **nevyhovující** a další testy struktury nejsou prováděny
- Když se přesnost modelu na testovacích a validačních datech pohybuje výrazně nad 50%, je tento model následně natrénován a posléze testován s pozměněnými parametry, zda není možno dosáhnout lepší přesnosti. Nejlepší možnost z testovaných je následně brána jako **vyhovující** a bude porovnána s nejlepšími výsledky případných jiných úspěšných struktur sítí.

### 7.2.1 Struktura 1

První testovací struktura má následující rozložení vrstev:

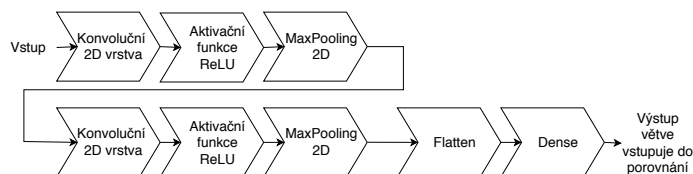


Obr. 7.2: Struktura 1

Přehled testovaných parametrů znázorněné struktury:

- Počet trénovacích epoch: 15, 40, 60, 100, 200
- Počet filtrů pro konvoluci: 20, 50, 70
- Velikost jádra konvoluce: 3, 4, 5, 6
- Velikost okna pooling: 2x2, 4x4, 6x6

### 7.2.2 Struktura 2



Obr. 7.3: Struktura 2

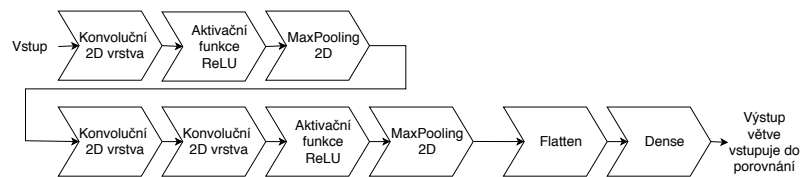
Přehled testovaných parametrů znázorněné struktury:

- Počet filtrů pro konvoluci: 20, 50, 90
- Velikost jádra konvoluce: 3, 5
- Velikost okna pooling: 4x4

### 7.2.3 Struktura 3

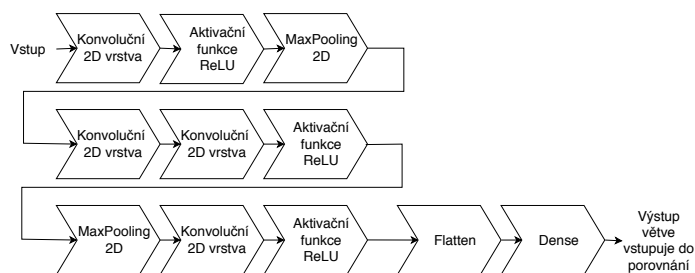
Přehled testovaných parametrů znázorněné struktury:

- Počet trénovacích epoch: 100, 200
- Počet filtrů pro konvoluci: 20, 50
- Velikost jádra konvoluce: 3, 5
- Velikost okna pooling: 2x2, 4x4



Obr. 7.4: Struktura 3

## 7.2.4 Struktura 4

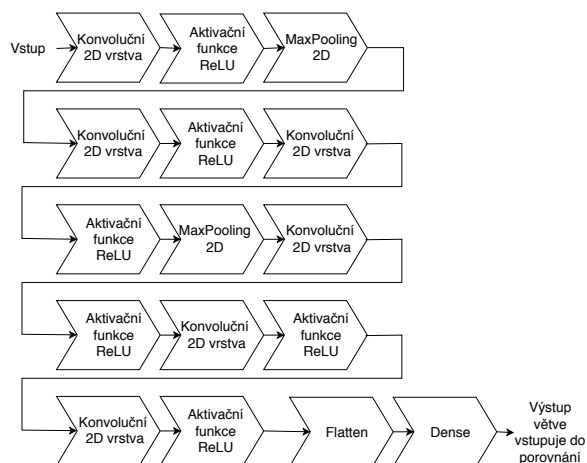


Obr. 7.5: Struktura 4

Přehled testovaných parametrů znázorněné struktury:

- Počet filtrů pro konvoluci: 20, 50, 70
- Velikost jádra konvoluce: 3, 5, 6, 8
- Velikost okna pooling: 3x3, 4x4

## 7.2.5 Struktura 5

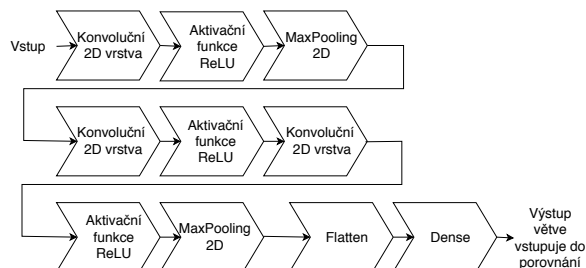


Obr. 7.6: Struktura 5

Přehled testovaných parametrů znázorněné struktury:

- Počet trénovacích epoch: 100, 200
- Počet filtrů pro konvoluci: 20, 50
- Velikost jádra konvoluce: 3
- Velikost okna pooling: 4x4

### 7.2.6 Struktura 6

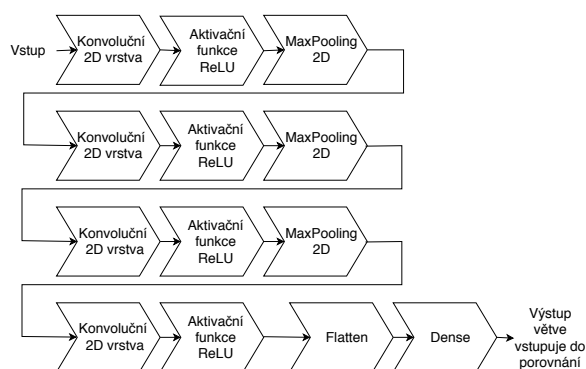


Obr. 7.7: Struktura 6

Přehled testovaných parametrů znázorněné struktury:

- Počet trénovacích epoch: 100, 300, 400, 500
- Počet filtrů pro konvoluci: 20, 50
- Velikost jádra konvoluce: 3
- Velikost okna pooling: 4x4

### 7.2.7 Struktura 7



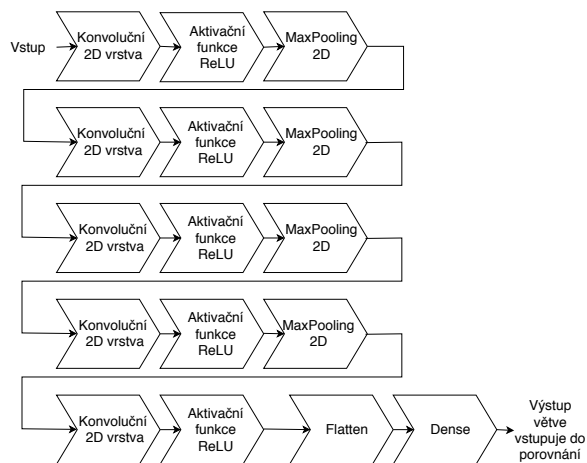
Obr. 7.8: Struktura 7

Přehled testovaných parametrů znázorněné struktury:

- Počet filtrů pro konvoluci: 20, 50
- Velikost jádra konvoluce: 5
- Velikost okna pooling: 2x2

### 7.2.8 Struktura 8

Osmá testovací struktura má následující rozložení vrstev:

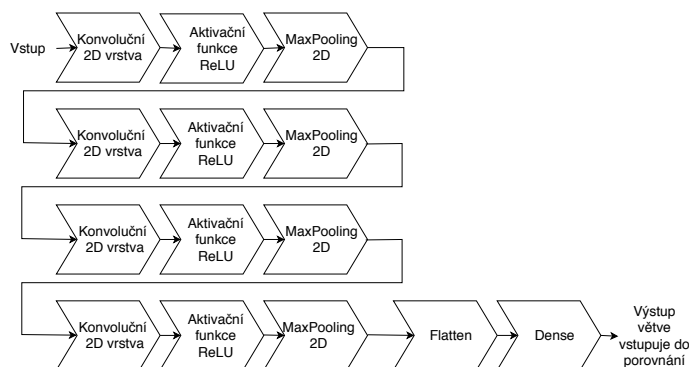


Obr. 7.9: Struktura 8

Přehled testovaných parametrů znázorněné struktury:

- Počet filtrů pro konvoluci: 20, 50
- Velikost jádra konvoluce: 5
- Velikost okna poolingů: 2x2

### 7.2.9 Struktura 9



Obr. 7.10: Struktura 9

Přehled testovaných parametrů znázorněné struktury:

- Počet filtrů pro konvoluci: 20, 50, 70, 90
- Velikost jádra konvoluce: 5
- Velikost okna poolingů: 2x2

## 7.3 Syntaxe definice modelu

Vrstvy navrženého modelu jsou skládány do tzv. *sekvence*. V prostředí nástroje Keras je možné definovat všechny zmíněné a využívané vrstvy následovně:

### 7.3.1 Část siamských větví

Konvoluční vrstva

```
sequence.add(Conv2D(20 , kernel_size=5, padding="same",  
input_shape="224, 224, 3'))
```

Aktivační funkce

```
sequence.add(Activation("relu"))
```

Vrstva MaxPooling

```
sequence.add(MaxPooling2D(pool_size=(2, 2), strides=None))
```

Vrstva *Flatten*

```
sequence.add(Flatten())
```

Plně propojená vrstva

```
sequence.add(Dense(500))
```

### 7.3.2 Společná část

Vrstva *Lambda*

```
distance = Lambda(cosine_distance)([vector_left, vector_right])
```

Plně propojená vrstva

```
fc1 = Dense(128, kernel_initializer="glorot_uniform")(distance)
```

Vrstva zahazování

```
fc1 = Dropout(0.2)(fc1)
```

Aktivační funkce - ReLu

```
fc1 = Activation("relu")(fc1)
```



### Plně propojená vrstva - výstupní

```
pred = Dense(2, kernel_initializer="glorot_uniform")(fc1)
```

### Aktivační funkce - výstupní

```
pred = Activation("softmax")(pred)
```

Jednotlivé parametry vrstev jsou samozřejmě proměnné podle testovaných hodnot, k nalezení nejlepšího nastavení. Cílem je získání nejlepších výsledků testování trénovaného modelu.

Mezi testované parametry patří následující hodnoty, zastoupené řeckými písmeny:

- $\gamma$  – Velikost jádra (rozměr filtru)
- $\mu$  – počet filtrů konvoluční vrstvy
- $\sigma$  – Velikost bloku ke sdružení ve vrstvě typu *MaxPooling*
- $\kappa$  – Velikost kroku, po kterém posouvat bloky ke sdružení. Ve všech případech testování ponechán parametr na hodnotě „None“, to znamená, že  $\sigma = \kappa$ .

## 8 VÝSLEDKY

Výsledky trénování a testování modelů jsou shromážděny pro následující proměnné parametry:

- **Počet trénovacích epoch**  $[-]$
- **Počet konvolučních filtrů**  $[-]$
- **Velikost jádra**(filtru, či váhy)  $[px]$
- **Pooling** (Velikost okna)  $[px]$

Přesnosti jednotlivých modelů a jejich konfigurací jsou zapsány v následujících tabulkách. Pro nejlepší průběhy jsou uvedeny grafy průběhů trénování. Hodnoty přesností jsou uvedeny jednak pro validační data (vyhodnocované v průběhu trénování modelu) a pak také pro testovací data (vyhodnocené po ukončení trénování modelu)

Tab. 8.1: Struktura 1 - Přehled výsledků

Epoch $[\alpha]$	Filtrů $[-]$	Rozměr jádra $[\gamma]$	Pooling $[\sigma]$	% validační	% testovací
15	20, 20, 20	5	2x2	68,86%	75,38%
			4x4	<b>83,95%</b>	<b>85,86%</b>
			6x6	68,40%	64,31%
40	20, 20, 20	5	2x2	64,78%	82,32%
			4x4	84,05%	73,19%
			6x6	61,83%	81,86%
60	20, 50, 50	5	2x2	76,72%	75,49%
			4x4	73,57%	84,29%
			6x6	76,03%	78,06%
100	20, 20, 20	4	2x2	50,08%	50,00%
		5	4x4	49,79%	50,00%
			4x4	72,36%	77,22%
		6	2x2	49,87%	50,00%
	20, 50, 50	3	4x4	55,11%	77,57%
	20, 50, 50	5	6x6	69,62%	66,45%
	20, 50, 50		2x2	49,96%	50,00%
	20, 50, 50		4x4	75,51%	76,17%
	20, 50, 50	6	6x6	65,83%	76,51%
	70, 70, 70	5	4x4	49,82%	50,00%
200	50, 50, 50	5	4x4	50,36%	50,00%

První uvedená struktura vykazuje nejlepší hodnoty na nejnižších počtech trénovacích epoch s parametry o hodnotách:

- Velikost jádra  $\gamma - 5$
- Počet filtrů  $\mu - 20, 20, 20$
- Velikost poolingu  $\sigma - 4x4$

Při těchto hodnotách parametrů model **vyhovuje** hledání.

Tab. 8.2: Struktura 2 - Přehled výsledků

Epoch [ $\alpha$ ]	Filtrů [-]	Rozměr jádra [ $\gamma$ ]	Pooling [ $\sigma$ ]	% validační	% testovací
100	20, 50	3	4x4	54,43%	51,71%
100	90, 90	5	4x4	50,07%	50,00%

Model, navržený podle struktury 2 nevykazuje zajímavé výsledky - vyhodnocení přesností odhadů na validačních i testovacích datech se pohybuje okolo 50%, model tedy **nevyhovuje** požadavkům.

Tab. 8.3: Struktura 3 - Přehled výsledků

Epoch [ $\alpha$ ]	Filtrů [-]	Rozměr jádra [ $\gamma$ ]	Pooling [ $\sigma$ ]	% validační	% testovací
100	20, 50, 50	3	4x4	80,32%	77,30%
		3, 5, 5	2x2	49,83%	50,00%
200	20, 50, 50	3	4x4	70,40%	72,94%

Třetí struktura modelu vykazuje **vyhovující** hodnoty při parametrech:

- Velikost jádra  $\gamma$  – 3
- Počet filtrů  $\mu$  – 20, 50, 50
- Velikost poolingů  $\sigma$  – 4x4

Při navýšení počtu trénovacích epoch však **nedochází ke zlepšování** přesnosti odhadu ani na validačních, ani na testovacích datech.

Při změně parametrů nevykazuje struktura lepší výsledky a je **nevyhovující**.

Tab. 8.4: Struktura 4 - Přehled výsledků

Epoch [ $\alpha$ ]	Filtrů [-]	Rozměr jádra [ $\gamma$ ]	Pooling [ $\sigma$ ]	% validační	% testovací
100	20, 50, 50, 50	3	4x4	68,90%	71,53%
		6, 8, 8, 8	4x4	49,39%	50,00%
	20, 50, 50, 70	3, 5, 5, 5	4x4	80,27%	76,09%
		6	3x3	49,84%	50,00%
200	20, 50, 50, 70	3, 5, 5, 5	3x3	49,75%	50,00%

Trénování modelu podle struktury 4 a jeho výsledky mají dobré výsledky, hlavně při konfiguraci parametrů:

- Velikost jádra  $\gamma$ 
  - První konvoluční vrstva – 3
  - Druhá a třetí konvoluční vrstva – 5
  - Čtvrtá konvoluční vrstva – 5
- Počet filtrů  $\mu$  – 20, 50, 50
- Velikost poolingů  $\sigma$  – 4x4

Tab. 8.5: Struktura 5 - Přehled výsledků

Epoch [ $\alpha$ ]	Filtrů [-]	Rozměr jádra [ $\gamma$ ]	Pooling [ $\sigma$ ]	% validační	% testovací
100	20, 50, ...	3	4x4	68,52%	70,59%
200	20, 50, ...	3	4x4	50,49%	50,00%

Tab. 8.6: Struktura 6 - Přehled výsledků

Epoch [ $\alpha$ ]	Filtrů [-]	Rozměr jádra [ $\gamma$ ]	Pooling [ $\sigma$ ]	% validační	% testovací
100	20, 50, 50	3	4x4	52,71%	68,39%
300	20, 50, 50	3	4x4	65,37%	69,98%
400	20, 50, 50	3	4x4	78,55%	77,85%
500	20, 50, 50	3	4x4	50,00%	50,00%

Model podle struktury 6 zpočátku vykazuje velmi nízkou přesnost na validačních datech. Přesnost na testovacích datech sice dosahuje lepších hodnot. Při trénování s vyšším počtem trénovacích epoch se obě hodnoty přesností téměř blíží 80%, ale vyšších hodnot již nedosáhnou a naopak, na počtu 400 trénovacích epoch je model již **přetrénován**.

Tab. 8.7: Struktura 7 - Přehled výsledků

Epoch [ $\alpha$ ]	Filtrů [-]	Rozměr jádra [ $\gamma$ ]	Pooling [ $\sigma$ ]	% validační	% testovací
100	20, 50, ...	5	2x2	49,85%	50,00%

Testování modelu dle struktury 7 vykazuje pouze **nevyhovující** výsledky.

Tab. 8.8: Struktura 8 - Přehled výsledků

Epoch [ $\alpha$ ]	Filtrů [-]	Rozměr jádra [ $\gamma$ ]	Pooling [ $\sigma$ ]	% validační	% testovací
100	20, 50, ...	5	2x2	50,16%	50,00%

Testování modelu dle struktury 8 vykazuje pouze **nevyhovující** výsledky.

Tab. 8.9: Struktura 9 - Přehled výsledků

Epoch [ $\alpha$ ]	Filtrů [-]	Rozměr jádra [ $\gamma$ ]	Pooling [ $\sigma$ ]	% validační	% testovací
100	20, 50, 70, 90	5	2x2	49,89%	50,00%

Testování modelu dle struktury 9 vykazuje pouze **nevyhovující** výsledky.

Následující grafy zobrazují průběhy nejlepších konfigurací jednotlivých struktur. Nejsou vyvedeny grafy struktur s neuspokojivými výsledky.

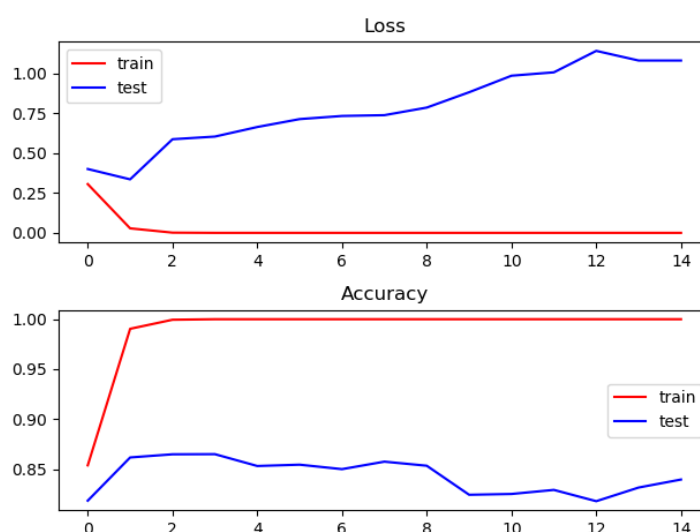
Vysvětlivky grafů:

- **Loss** – Zobrazení průběhů ztrát modelu s počtem trénovacích epoch.
- **Accuracy** – Zobrazení vývoje přesnosti hodnot s počtem trénovacích epoch.

Jednotlivé grafy následně obsahují barevně odlišené průběhy:

- **Červený průběh** – Hodnoty trénovacích dat.
- **Modrý průběh** – Hodnoty validačních dat.

### Nejlepší konfigurace struktury 1 (viz tab. 8.1 )



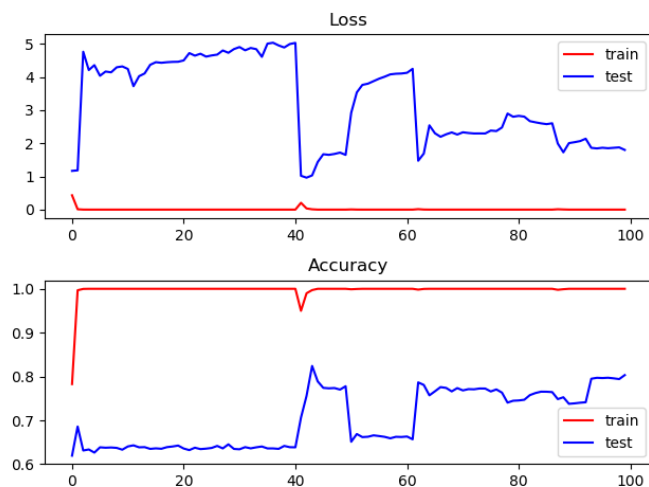
Obr. 8.1: Graf nejlepšího výsledku struktury 1

Jedná se o graf průběhu trénování modelu s nejlepšími konečnými výsledky přesnosti jak validačních, tak i testovacích dat.

**Ztráty modelu** – s vyšším počtem trénovacích epoch mají stoupavý charakter.

**Přesnost modelu** – vzhledem ke zvyšujícím se hodnotám ztrát modelu dochází s vyšším počtem trénovacích epoch k zneprůhledňování modelu. Návrh je nejpřesnější na nižším počtu trénovacích epoch.

### Nejlepší konfigurace struktury 3 (viz tab. 8.3 )

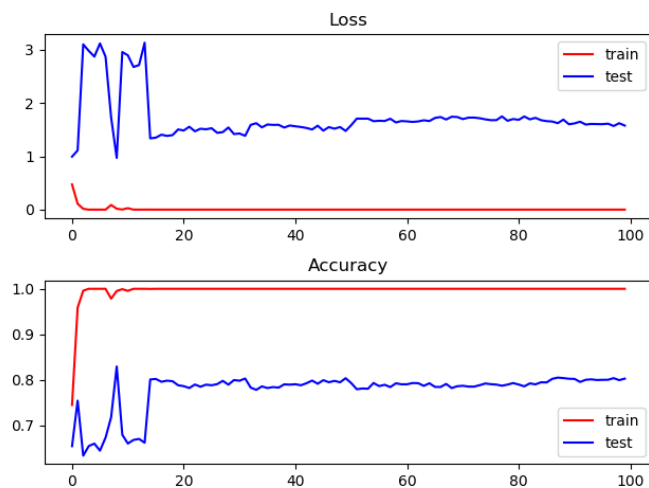


Obr. 8.2: Graf nejlepšího výsledku struktury 3

**Ztráty modelu** – s průběhem trénování dochází k ustálení hodnot ztrát modelu.

**Přesnost modelu** – dle ztrát modelu dochází k ustálení hodnoty.

### Nejlepší konfigurace struktury 4 (viz tab. 8.4 )

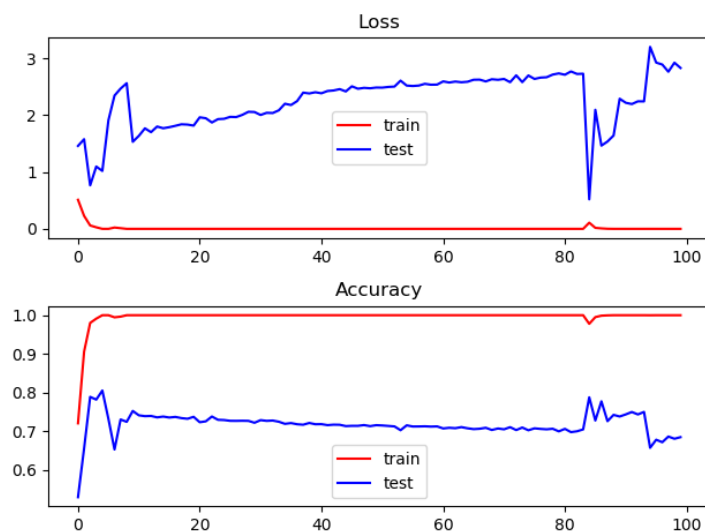


Obr. 8.3: Graf nejlepšího výsledku struktury 4

**Ztráty modelu** – s průběhem trénování dochází k ustálení hodnot ztrát modelu jako u předchozího průběhu.

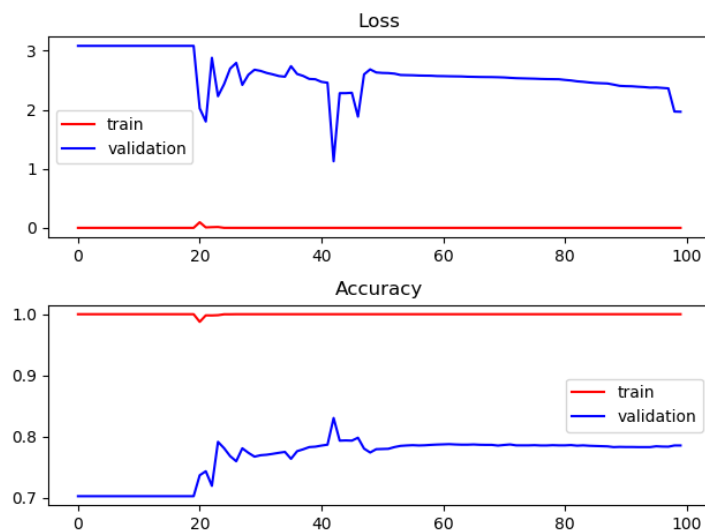
**Přesnost modelu** – s rostoucím počtem trénovacích epoch dochází k ustálení hodnoty.

### Nejlepší konfigurace struktury 5 (viz tab. 8.5 )



Obr. 8.4: Graf nejlepšího výsledku struktury 5

### Nejlepší konfigurace struktury 6 (viz tab. 8.6 )



Obr. 8.5: Graf nejlepšího výsledku struktury 6

Jedná se o graf průběhu trénování 300 až 400 epoch, tj. nejpresnější test modelu. Indexace osy X grafu je v rozmezí 0-100 z důvodu trénování modelu ve více krocích (po 100 epochách odděleně z důvodu lepšího sledování průběhu). Ačkoliv se zdá průběh dle grafu nadějný, další vývoj vede skokově k přetrénování modelu.

## 9 VYHODNOCENÍ VÝSLEDKŮ

Při vyhodnocení výsledků jednotlivých trénovaných modelů je brána v potaz přesnost, ověřovaná na validačních datech, především pak ale výsledná přesnost nezávislých testovacích dat, ověřovaných po celkovém natrénování modelu.

### 9.1 Nejlepší návrh

Nejlepšího návrhu, tj. nejvyšší přesnosti při odhadu podobnosti dvou obrazů na **testovacích** datech bylo dosaženo s využitím struktury 1, ilustrované v schématu 7.2. Neméně důležité jsou parametry jednotlivých vrstev modelu, se kterými bylo nejlepšího výsledku dosaženo. Jedná se o hodnoty:

- Počet trénovacích epoch  $\alpha - 15$
- Počet filtrů  $\mu - 20, 20, 20$
- Velikost jádra  $\gamma - 5$
- Velikost poolingů  $\sigma - 4 \times 4$

Graf průběhu trénování modelu lze sledovat na obr. 8.1. S těmito hodnotami konfigurace sítě bylo dosaženo přesnosti 83,95% při odhadu podobnosti na validačních datech. Při ověřování univerzálnosti modelu na nezávislých testovacích datech pak bylo dosaženo dokonce přesnosti **85,86%**, což činí z tohoto modelu na dané konfiguraci parametrů nejúspěšnější testovaný návrh.

Díky nejvyšší dosažené přesnosti byl tento model testován nejvíce ze všech uvažovaných struktur ve snaze získat ještě lepší hodnoty přesnosti, než při výše zmíněné konfiguraci. Všechny výsledky pro jednotlivé parametry lze sledovat v tabulce 8.1.

Jedna z následně testovaných konfigurací nakonec dosáhla ještě vyšší přesnosti na validačních datech - 84,05% - s parametry:

- Počet trénovacích epoch  $\alpha - 40$
- Počet filtrů  $\mu - 20, 20, 20$
- Velikost jádra  $\gamma - 5$
- Velikost poolingů  $\sigma - 4 \times 4$

Při ověřování modelu na nezávislých testovacích datech však bylo dosaženo přesnosti pouze 73,19%, což je znatelně méně, než při předchozí konfiguraci.

### 9.2 Ostatní návrhy

Mezi testy s úspěšnými přesnostmi se dále řadí modely podle struktur 7.4, 7.5, 7.6 a 7.7



U modelu podle struktury 3 (7.4) bylo dosaženo výsledné přesnosti na validačních datech 80,32% a na testovacích datech 77,30%. Z grafu 8.2 a tabulky výsledků 8.3 průběhu vývoje hodnot s počtem trénovacích epoch je patrné, že dochází k ustálení ztrát modelu a je zde předpoklad, že přesnost se již nezlepší a nedosáhne lepších hodnot, než nejúspěšnější model.

Model podle struktury 4 (7.5) dosáhl nejlepších hodnot přesnosti validačních dat 80,27% a přesnosti testovacích dat 76,09%. S rostoucím počtem trénovacích epoch lze na grafu 8.3 lze opět sledovat ustálení hodnot bez dalšího zlepšování výsledných přesností.

Další úspěšnější model podle struktury 5(7.6) skončil na hodnotách přesností validačních dat 68,52% a testovacích dat 70,59%. Z grafu průběhu trénování 8.4 lze usuzovat, že s nárůstem počtu trénovacích epoch se bude výsledná přesnost modelu již jen zhoršovat.

Poslední z modelů, které dosáhly výrazně vyšší hodnoty přesností odhadu je sestaven podle struktury 6 (7.7). Z tabulky jednotlivých výsledků testování modelu 8.6 je patrné, že s vyšším počtem trénovacích epoch dochází k určitému zpřesňování modelu. Nejvyšší hodnoty 77,85% přesnosti na testovacích datech bylo dosaženo pro 400 trénovacích epoch. Při dalším trénování je ale názorně vidět, že dojde k přetrénování modelu a skokové ztrátě přesnosti - model se stane příliš specializovaným na to, aby byl schopen řešit problém v obecném měřítku.

Při vyhodnocení nebereme v potaz struktury, které dosáhly výrazně dobré přesnosti. Modely s přesností blížící se 50% jsou považovány za **nevyhovující**.

## 10 ZÁVĚR PRÁCE

Cílem práce bylo prostudovat možnosti využití hlubokého učení a konvolučních vrstev při práci s obrazem. Konkrétní zkoumanou oblastí zpracovávání obrazu bylo zjištění a vyhodnocení podobnosti dvou obrazových dat.

V teoretické části práce byly stručně popsány možnosti struktury a reprezentace obrazových dat v digitálních technologiích. Následuje úvod do obecné problematiky neuronových sítí, popis jejich základních stavebních prvků – neuronů – a jejich následné shlukování, tvořící síť. Dále je popsána podoblast neuronových sítí nazvaná *Hluboké učení*. Stručná historie uvádí postupný vývoj této technologie, počínaje teoretickými úvahami a konče reálnými konstrukcemi a nasazením. Uvedeny jsou dva druhy hlubokých neuronových sítí, rekurentní a konvoluční. Další text práce je věnován právě konvolučním hlubokým neuronovým sítím, které se hodí pro práci s obrazem zejména díky jejich vlastnostem a charakteristikám. Přehled a popis funkčnosti jednotlivých stavebních vrstev tohoto typu sítí je popsán v rámci této kapitoly. Závěrem teoretické části je popis možnosti porovnání obrazových dat na základě jejich datové struktury a konečně úvaha využití hlubokého učení k vytvoření modelu, realizujícím porovnání obrazů na základě natrénovaných schopností.

Praktická část práce uvádí způsob a postupy řešení samotného problému porovnání obrazových dat s využitím hlubokého učení. S pomocí nástroje Keras a dalších vývojových nástrojů, (vyjmenovaných a popsáných v kapitole 6.1) bylo snahou docílit realizace vhodného návrhu neuronové sítě. Tím byl zvolen model tzv. *siamských* neuronových sítí, jejichž hlavní vlastnost spočívá v konstrukci dvou oddělených větví konvolučních(a dalších) vrstev, které však sdílejí totožné váhy. Výstup těchto oddělených větví byl poté vhodnou metodou porovnáván a výstupem modelu je hodnota, udávající, zda se jedná o obrazy podobné, či rozdílné.

Vytvořený model byl trénován za pomoci *trénovacího* a *validačního* datasetu. Po ukončení trénování byl testován na oddělených nezávislých *testovacích* datech. Důležitým aspektem při trénování modelu byl samotný návrh struktury modelu a poté parametry sítě, které byly upravovány. Různými kombinacemi konfigurací modelu a počtu trénovacích epoch bylo dosaženo mnoha výsledků, uvedených v kapitole 8.

Z výsledků kombinací těchto konfigurací bylo zjištěno, že celkově nejlepší skóre *testovací* množiny obrazových dat (**85,86%**) bylo dosaženo při využití struktury modelu 7.2 a parametrech, kde velikost jádra konvoluce  $\gamma = 5$ , rozměr poolingů  $\sigma = 4$  a počet filtrů konvoluce  $\mu = 20$  u všech konvolučních vrstev modelu. Počet trénovacích epoch nejúspěšnějšího modelu byl stanoven na  $\alpha = 15$ . Jakožto model

s nejlepším výsledkem celého testování byl tento návrh podroben nejvyššímu počtu kombinací hodnot parametrů sítě.

Při vyšším počtu trénovacích epoch a testování dalších struktur modelů již nebylo v žádném případě konfigurace dosaženo vyššího skóre *testovací* množiny jako v předchozím uvedeném případě. Jedním z důvodů je tzv. efekt *přetrénování* modelu neuronové sítě, kdy se síť stává moc specializovaná a ztrácí schopnost vyhodnocovat řešený problém na obecné množině vstupních dat.

Maximální dosažené hodnoty při testování, provedeném v rámci této práce nevyklučují možnost dosažení ještě vyšších přesností při zvolení výrazně jiných struktur modelů neuronových sítí k trénování a tvorbě modelu pro rozhodování.

Na základě zjištěných skutečností lze s jistotou potvrdit, že metody hlubokého učení a neuronových sítí lze k vyhodnocování podobnosti obrazových dat s jistotou využít. Při zpracovávání problému se též potvrdila komplexnost a síla metod neuronových sítí. Do budoucna se toto odvětví jistě bude dále široce rozvíjet a bude uplatňováno ve stále širším spektru oborů.

# LITERATURA

- [1] VALLADAS, H. CLOTTE J., GENESTE J.M., GARCIA M.A., ARNOLD M., CACHIER H., TISNÉRAT-LABORDE N. *Palaeolithic paintings: Evolution of prehistoric cave art*. Nature [online]. 2001, vol. 413, no. 6855, s. 479. ISSN 00280836.
- [2] BURGER, W., BURGE M.J. *Digital Image Processing: An Algorithmic Introduction Using Java*. 2, ilustrované vydání. London: Springer London, 2016. ISBN 9781447166849.  
Dostupné z: <https://books.google.cz/books?id=YpzWCwAAQBAJ>
- [3] ARNTSON, A.E. *Graphic design basics*. 2nd ed. Fort Worth: Harcourt Brace Jovanovich College Publishers, c1993. ISBN 0030554837.
- [4] GOODFELLOW, I., BENGIO, Y., COURVILLE, A. *Deep Learning*, 2016.  
Dostupné z: <http://www.deeplearningbook.org>
- [5] SAMUEL, A. L. *Some Studies in Machine Learning Using the Game of Checkers*. IBM Journal of Research and Development. Volume: 3, Issue: 3, July 1959. Armonk: International Business Machines Corporation. ISSN 0018-8646.
- [6] SCHMIDHUBER, J. *Deep learning in neural networks*.  
ScienceDirect 2014. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0893608014002135>
- [7] KVASNIČKA, V. *Úvod do teórie neurónových sietí*. Slovenská republika: IRIS, 1997. ISBN 80-88778-30-1.
- [8] GERNSHEIM, Helmut. *A concise history of photography*. 3rd rev. ed. New York: Dover Publications, 1986. ISBN 0486251284.
- [9] KOHOUT, P. *Prodej to: Úvod fotografů do světa mikrofotobank*. KKnihy.cz 2014. ISBN 9788088061007.  
Dostupné z: <https://books.google.cz/books?id=NbgyBQAAQBAJ>
- [10] RICHTER, J. *Inventing the camera*. New York: Crabtree Pub. Co., c2006. ISBN 0778728145.
- [11] ELMANSY, R. *Illustrator foundations the art of vector graphics and design in Illustrator*. New York: Focal Press, 2013. ISBN 1136146059.
- [12] ROJAS, R., FOREWORD BY JEROME FELDMAN. *Neural networks: a systematic introduction*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. ISBN 3642610684.

- [13] GURNEY, K. *An Introduction to Neural Networks*. London: Routledge, 1997. ISBN 0203451511.
- [14] SABIN-WILSON, L. *WordPress Web Design For Dummies* 3. vydání, 2015 ISBN 9781119088608.  
Dostupné z: <https://books.google.cz/books?id=2FezCgAAQBAJ>
- [15] RICHARDSON, I.E.G. *H.264 and MPEG-4 Video Compression Video Coding for Next-generation Multimedia*. Chichester: John Wiley, 2003. ISBN 0470869607.
- [16] GRAUPE, D. *Deep Learning Neural Networks: Design and Case Studies*. 2016. ISBN 9789813146471.  
Dostupné z: <https://books.google.cz/books?id=e5hIDQAAQBAJ>
- [17] EBERHART, R.C. *Neural Network PC Tools: A Practical Guide*, Elsevier Science, 2014. ISBN 9781483297002
- [18] GOODFELLOW, I., BENGIO, Y., COURVILLE, A. *Deep Learning*. MIT Press, 2016. ISBN 9780262035613
- [19] LEE, H., GROSSE, R., RANGANATH R., and Y. Ng., A. *Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations* In Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09), pages: 609-616. ACM, New York, NY, USA 2009. DOI: <http://dx.doi.org/10.1145/1553374.1553453>
- [20] GULCEHRE, C., MOCZULSKI, M., DENIL, M., BENGIO, Y. *Noisy Activation Functions* Proceedings of Machine Learning Research, str. 3059–3068, 2016
- [21] FERGUS, R. *Neural Networks: Facebook AI Research* Machine Learning Summer School 2015 [cit. 2017-12-03]. Dostupné z: [http://mlss.tuebingen.mpg.de/2015/slides/fergus/Fergus\\_1.pdf](http://mlss.tuebingen.mpg.de/2015/slides/fergus/Fergus_1.pdf)
- [22] ARIK, S., HUANG, T. LAI, W.K., LIU, Q. *Neural Information Processing: 22nd International Conference ICONIP 2015*, Istanbul, Turkey, November 9-12, 2015. Proceedings, Díl 1. ISBN 9783319265322
- [23] RUDER, S. *An overview of gradient descent optimization algorithms*, Journal Article, arXiv preprint arXiv:1609.04747. 2016

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

PNG	Portable Network Graphics
API	Application Programming Interface
CPU	Central Processing Unit
GPU	Graphics Processing Unit
RNN	Recurrent neural network
CNN	Convolutional neural network
ReLU	Rectified Linear Units
ADAM	Adaptive Moment Estimation
RGB	Red-green-blue
RGBA	Red-green-blue-alpha