ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA RIADENIA A INFORMATIKY

KLASIFIKÁCIA ZVUKOV PROSTREDIA S VYUŽITÍM METÓD STROJOVÉHO UČENIA

Dizertačná práca

28360020223004

Študijný program:	Aplikovaná informatika
Študijný odbor:	Informatika
Pracovisko:	Katedra technickej kybernetiky
	Fakulta riadenia a informatiky, Žilinská univerzita v Žiline
Školiteľ:	doc. Ing. Peter Ševčík, PhD.

Žilina, apríl 2022 **Ing. Miroslav Chochul**

Čestné prehlásenie

Čestne prehlasujem, že som túto prácu vypracoval samostatne s využitím dostupnej literatúry a vlastných vedomostí. Všetky zdroje použité v tejto dizertačnej práci som uviedol v súlade s predpismi.

V Žilina, dňa 22. 04. 2022 Ing. Miroslav Chochul

Poďakovanie

Touto cestou by som sa chcel poďakovať vedúcemu práce doc. Ing. Petrovi Ševčíkovi, PhD. za jeho odbornú pomoc, pripomienky a usmerňovanie pri tvorbe tejto práce. Zároveň by som sa chcel poďakovať aj kolegom z Katedry technickej kybernetiky za pomoc a konzultácie.

Veľké poďakovanie patrí mojim rodičom za pomoc a podporu počas celého doktorandského štúdia. Špeciálne by som sa chcel poďakovať mojej sestre Natali, ktorej podpora v najťažších chvíľach nikdy nebude zabudnutá.

Abstrakt

CHOCHUL, Miroslav: *Klasifikácia zvukov prostredia s využitím metód strojového učenia.* [Dizertačná práca]. – Žilinská univerzita v Žiline. Fakulta riadenia a informatiky. Katedra technickej kybernetiky. - Vedúci: doc. Ing. Peter Ševčík, PhD. - Stupeň odbornej kvalifikácie: Doktor filozofie v študijnom odbore informatika. – Žilina: FRI UNIZA, 2022. Počet strán 100.

Kľúčové slová: klasifikácia environmentálnych zvukov, strojové učenie, konvolučná neurónová sieť, nízko-parametrická architektúra.

Dizertačná práca sa zaoberá klasifikáciou environmentálnych zvukov, teda zvukov prostredia za pomoci metód strojového učenia. Klasifikačný model teda na základe akustického signálu predikuje druh zvuku. Teoretická časť prace je venovaná rozboru environmentálnych zvukov, ich pôvodu a spôsobu klasifikácie. Taktiež popisuje metódy strojového učenia, ich rozdelenie a využitie pre klasifikačné problémy. Ďalej sú tu popísané metódy extrakcie príznakov a druhy transformácie akustického signálu. Experimentálna časť práce je venovaná výberu a vývoju vhodnej architektúry klasifikačného modelu. Popisuje použité metódy predspracovania dát, ich augmentáciu a následnú extrakciu príznakov. Taktiež sa venuje popisu vývoju stratégie trénovania a vyhodnocovania klasifikačného modelu. Hlavným cieľom tejto práce bol návrh architektúry klasifikačného modelu, ktorý by mal nízku veľkosť, z čoho vyplýva nízky počet parametrov, aby bolo možné takýto model implementovať na zariadenia s obmedzenou výpočtovou silou. Pre porovnanie bol zvolený referenčný model, ktorým bol nami navrhnutý klasifikačný model porovnávaný. Z tohto porovnania vyplýva, že s využitím 0.65% veľkosti referenčného modelu, je možné dosiahnuť takmer rovnakú presnosť klasifikácie.

Abstract

CHOCHUL, Miroslav: *Environmental sounds classification using machine-learning methods.* [Dissertation thesis]. – University of Žilina. Faculty of Management Science and Informatics; Department of Technical Cybernetics. – Supervisor: doc. Ing. Peter Ševčík, PhD. - Qualification level: Philosophiae doctor in the study field informatics. Žilina, 2022. Page count 100.

Key words: environmental sound classification, machine learning, convolution neural network, low-parametric architecture.

The topic of this thesis is a classification of environmental sounds, i.e. non-human sounds, using machine-learning methods. The classification model, based on an acoustic signal, predicts a source of a sound. The theoretical part of the thesis is dedicated to the analysis of environmental sounds, their origin, and classification approaches. In addition, machine-learning methods, their taxonomy and their usage in classification tasks are described in this part as well. Next described are the feature extraction methods and types of acoustic signal transformations. The experimental part of the thesis is dedicated to the choice and development of the suitable architecture of the classification model. Next, are the description of data pre-processing methods, data augmentation and feature extraction. Furthermore, the development of training and evaluation strategies of the classification model are detailed. The main goal of this thesis was the development of a classification model architecture with a small size, which means low parameter count, to make it possible to implement this kind of model on devices with limited computational power. For evaluation, a reference model was chosen, against which our classification model was compared. From this comparison results that by using a 0.65% size of the reference model it is possible to achieve nearly similar classification accuracy.

Obsah

Zoznam obrázkov 9

Zoznam tabuliek 11

Zoznam skratiek 12

Úvod 13

1 Environmentálne zvuky 15

1.1 Organizácia environmentálnych zvukov 16

1.2 Klasifikácia environmentálnych zvukov 17

2 Strojové učenie 20

2.1 Metódy klasifikácie 21

2.1.1 Rozhodovacie stromy 23

2.1.2 Náhodný les 23

2.1.3 Metóda podporných vektorov 23

2.1.4 k-Najbližších susedov 24

2.1.5 Umelé neurónové siete 24

2.1.6 Hlboké učenie 25

2.1.7 Naivný Bayes 25

2.2 Neurónové siete 25

2.2.1 Historický vývoj a použitie neurónových sietí 26

2.2.2 Neurón 27

2.2.3 Architektúra neurónovej siete 31

2.2.4 Rosenblattov perceptrón 33

2.2.5 Proces učenia neurónovej siete 34

2.3 Konvolučné neurónové siete 35

2.3.1 Konvolúcia 36

2.3.2 Konvolučná vrstva 37

2.3.3 Podvzorkovanie 42
3 Extrakcia príznakov 44
3.1 Fourierová transformácia 46
3.1.1 Diskrétna Fourierová transformácia 46
3.1.2 Rýchla Fourierová transformácia 47
3.1.3 Váhové funkcie 49
3.1.4 Krátkodobá Fourierová transformácia 51
3.1.5 KĺzavÁ diskrétna Fourierová transformácia 52
3.1.6 Gaborová transformácia 52
3.2 Biológiou inšpirované metódy transformácie 53
3.2.1 Mel spektrogram 53
3.2.2 Gammatonová banka filtrov 54
3.3 Ďalšie metódy pre analýzu signálu 54
3.3.1 Diskrétna kosínusová transformácia 54
3.3.2 Hilbert-Huangová transformácia 56
4 Experimentálna časť 58
4.1 Datasety 58
4.1.1 ESC-50 59
4.2 Metódy strojového učenia 60
4.3 Predspracovanie dát 64
4.4 Extrakcia príznakov 66
4.4.1 Metóda reformácie spektrogramu 67
4.4.2 Experiment – Veľkosť analyzovaného rámca 69
4.4.3 Experiment – Veľkosť prekrytia 72
4.4.4 Augmentácia dát 73
4.5 Zmeny trénovacieho procesu 76
4.6 Úpravy extrakcie príznakov 79

4.7 Zmeny v architektúre EffNet 80
4.8 Krížová validácia 83
4.9 Prenášané učenie 84
Záver 89
Referencie 91
Zoznam publikácií 98
Zoznam citácií 100

Zoznam obrázkov

Obrázok 1 Taxonómia strojového učenia 21
Obrázok 2 Model umelého neurónu [37] 27
Obrázok 3 Označenie neurónov a synaptických váh 28
Obrázok 4 Afinna transformácia spôsobená prítomnosťou biasu 29
Obrázok 5 Ukážky aktivačných funkcií neurónu 30
Obrázok 6 Viacvrstvová architektúra doprednej neurónovej siete 31
Obrázok 7 Architektúra rekurentnej neurónovej siete so skrytými neurónmi 32
Obrázok 8 Rozdelenie rozhodovacieho priestoru pre logické funkcie AND, OR, NOT a XOR 34
Obrázok 9 Aplikácia dvojrozmernej konvolúcie 38
Obrázok 10 Operácia konvolúcie pre vstup s dvomi kanálmi a korešpondujúcimi filtrami, tie vynásobia každý kanál samostatne a sú sčítané na konci čím vytvoria výstupnú mapu s jedným kanálom 39
Obrázok 11 Operácia konvolúcie s použitím dvoch konvolučných filtrov. Tie sú aplikované samostatne na vstupný tenzor a ukladané na seba čím vytvoria dvojkanálovú výstupnú mapu 39
Obrázok 12 Konvolučný filter o veľkosti 2x2 aplikovaný na vstupnú mapu príznakov, s nastavením kroku konvolúcie (2,2), čo má za následok redukciu výstupnej mapy príznakov 40
Obrázok 13 Operácia konvolúcie, kedy bola vstupná mapa príznakov rozšítená o jednu hodnotu 41
Obrázok 14 Operácia konvolúcie, kedy bola hodnota roztiahnutia filtra nastavená na 2 41
Obrázok 15 Operácia konvolúcie s rozdelením 2 kanálového vstupu a 4 kanálového výstupu do dvoch skupín 42
Obrázok 16 Aplikácia vrstvy združovania podľa maximálnej hodnoty v červenej farbe. Pre ilustráciu sme pridali aj združovanie podľa priemernej hodnoty, ktorá bola zaokrúhlená v žltej farbe 43
Obrázok 17 Prevod zvuku na číslicový signál 45
Obrázok 18 Príklady spektrogramov nahrávok v datasete ESC-50 60
Obrázok 19 Architektúra konvolučnej siete K. Piczak [67] 62

Obrázok 20 Architektúra EffNet s detailným EffNet blokom. „dw“ znamená hĺbková konvolúcia (angl. depthwise convolution) a „mp“ znamená združovanie podľa maxima (z angl. max-pool) 63
Obrázok 21 Zvuk rozbitia skla s rôznymi nastaveniami orezania ticha 66
Obrázok 22 Príklady mapovania spektrogramov ako RGB zobrazenie 68
Obrázok 23 Vyhodnotenie experimentu vplyvu veľkosti analyzačného rámca 71
Obrázok 24 Vyhodnotenie experimentu úrovne prekrytia 72
Obrázok 25 Demonštrácia techník augmentácie dát na nahrávke zvuku otvárania plechovky. Obrázok ukazuje spektrogramy pred a po aplikovaní jednotlivých techník. Parametre sú prehnané, aby bol efekt zobrazený jasne 74
Obrázok 26 Architektúra našej konvolučnej neurónovej siete s detailom konvolučného bloku. „dw“ znamená hĺbková konvolúcia, „mp“ znamená združovanie podľa maxima a „ap“ združovanie podľa priemeru. 82
Obrázok 27 Ilustrácia *k*-násobnej krížovej validácie, pre $k = 5$ 84
Obrázok 28 Ukážka prístupov k prenášanému učeniu. Červené bloky predstavujú časti, ktoré sú nanovo inicializované; modré bloky predstavujú časti, ktorých parametre boli zmrazené. 86

Zoznam tabuliek

Tabuľka 1 Všeobecné porovnanie datasetov 59
Tabuľka 2 Rozdelenie tried datasetu ESC-50 59
Tabuľka 3 Výsledné presnosti rozpoznávania pri použití 5-násobnej krížovej validácie ... 84
Tabuľka 4 Výsledné presnosti rozpoznávania. Trénovanie s využitím prenášaného učenia – prístup dotrénovanie 87
Tabuľka 5 Porovnanie klasifikačného modelu s referenčným 88

Zoznam skratiek

Hz	Hertz
Pa	Pascal
dB	decibel
W	Watt
DFT	Diskrétna Fourierová transformácia
FFT	Rýchla Fourierová transformácia
STFT	Krátkodobá Fourierová transformácia
SDFT	KĺzavÁ diskrétna Fourierová transformácia
DCT	Diskrétna kosínusová transformácia
HHT	Hilbert- Huangová transformácia
EMD	Empirická modálna dekompozícia
IMF	Vlastná modálna funkcia
PCM	Pulzná kódová modulácia
RGB	Farebný formát červená-zelená-modrá
SGD	Stochastický gradientový zostup
NLLoss	Negatívna logaritmická vierohodnosť

Úvod

Predstavme si, že stojíme na ulici v meste. Zavrieme oči, čo počujeme? Pravdepodobne okoloidúce autá a autobusy, kroky ľudí, ktorí prechádzajú okolo, možno smiech alebo plač dieťaťa. Na základe nášho sluchu vieme získať množstvo informácií o našom prostredí. Pre väčšinu ľudí je schopnosť počúvať samozrejmá a prirodzená, avšak v prípade výpočtovej techniky sa jedná o náročnú úlohu a algoritmy strojového počúvania, ktoré automaticky rozpoznajú zvukové udalosti, dodnes zostávajú otvorený problém.

Klasifikácia environmentálnych zvukov, pomocou ktorej by bolo možné analyzovať a kategorizovať akustické emisie okolia, má viacero možných využití. Ako príklad sa ponúka monitorovanie hlukového znečistenia v mestách, nakoľko zvuk je dôležitým zdrojom informácií o mestskom živote. Ďalšie zaujímavé využitie je v oblasti bioakustiky, kde sú pomocou akustických emisií skúmané rôzne živočíchy či celé biodiverzity. Využitie je možne aj pre takzvaný akustický bezpečnostný systém, nakoľko mikrofóny sú všeobecne menšie a lacnejšie než kamery a sú odolné voči environmentálnym podmienkam ako sú hmla či zmena denného svetla a vďaka faktu, že zvuk prechádza cez prekážky, je možné implementovať takýto systém aj na monitorovanie väčšej oblasti ako sú lesy alebo polia. Zároveň je zaznamenávanie zvuku zvyčajne menej energeticky náročné.

Smerovanie dizertačnej práce je orientované do oblasti monitorovania chránenej oblasti za účelom signalizácie alebo v skratke akustický bezpečnostný systém. Našou motiváciou je systém pre ochranu lesov, pred nelegálnou ťažbou alebo nepovoleným vstupom motorových vozidiel do lesných oblastí, ktorý by mohol v budúcnosti vzniknúť. Nelegálna ťažba dreva je pretrvávajúci problém, v policajných štatistikách, bolo v prípade trestného činu Nelegálnej (pytliackej) ťažby dreva zistených 489 prípadov v roku 2020, v roku 2019 to bolo 618 prípadov krádeže dreva, či už v štátnych alebo v súkromných lesoch, ktoré riešila polícia Slovenskej republiky. Z hľadiska nelegálneho vstupu motorového vozidla, tieto prípadu sú posudzované ako trestný čin Porušovanie ochrany živočíchov a rastlín, takýchto prípadov bolo v roku 2020 zistených 96 a v roku 2019 bolo týchto prípadov 70. Preto usudzujeme, že takéto monitorovanie by malo zmysel a zvuk ako informačné médium je vhodnou voľbou. Šírenie zvuku je zväčša odolné voči prekážkam, má vysokú informačnú hodnotu a jeho zaznamenávanie je energeticky výhodnejšie ako v prípade obrazu. Z tohto dôvodu sme sa rozhodli analyzovať úlohu klasifikácie environmentálnych zvukov.

Väčšina prístupov riešenia úlohy klasifikácie environmentálnych zvukov, ktoré využívajú metódy strojového učenia, sú založené na hlbokom učení, z čoho vyplývajú veľmi vysoké výpočtové požiadavky. Tieto prístupy dosahujú dobré výsledky, pokiaľ sa jedná o presnosť rozpoznávania, avšak ich implementácia na zariadenia s nižšou výpočtovou silou je pre ich veľkosť zvyčajne problematická. V našej práci sme sa preto rozhodli venovať návrhu klasifikátora s nízkou veľkosťou, ktorý by bolo možné implementovať aj na zariadenia s nižšiu výpočtovou silou.

Na základe tohto sme si určili nasledujúce ciele dizertačnej práce:

- Analýza metód strojového učenia a metód spracovávania akustického signálu.
- Návrh klasifikačného modelu pre klasifikáciu environmentálnych zvukov s využitím poznatkov z vykonanej analýzy.
  Tento klasifikačný model by mal byť navrhnutý s dôrazom na nízku veľkosť, preto boli určené dva sekundárne ciele:
  - Architektúra klasifikačného modelu by mala pracovať s čo najnižším počtom parametrov.
  - Klasifikačný model by mal pracovať s čo najmenšou vzorkou akustického signálu, ideálne do jednej sekundy.
- Porovnanie navrhnutého klasifikačného modelu so zvoleným referenčným modelom.

Prvá kapitola popisuje environmentálne zvuky, ich kategorizáciu a oblasti využívajúce klasifikáciu environmentálnych zvukov.

Druhá kapitola je venovaná strojovému učeniu a metódam klasifikácie. Zároveň sa tu uvádza rozbor problematiky umelých neurónových sietí a konvolučných neurónových sietí.

V tretej kapitole sú rozobraté metódy extrakcie príznakov, resp. transformácie akustického signálu.

Vo štvrtej kapitole sa venujeme experimentálnej časti práce. Je tu popísaný výber množiny vstupných dát a metód strojového učenia. Nachádza sa tu popis predspracovania dát, ich augmentácia a extrakcia príznakov. Ďalej je tu popísaný vývoj architektúry a stratégie trénovacieho procesu.

1 Environmentálne zvuky

Vo všeobecnosti môžeme rozdeliť environmentálne zvuky na zvukové, resp. akustické udalosti, pri ktorých je zvuk produkovaný separátnymi fyzickými zdrojmi hluku, ako napríklad prechádzajúce auto, spev vtákov alebo kostolný zvon. Zvukové udalosti majú len jeden zdroj, avšak definícia, čo sa ráta za jeden zdroj zvuku je subjektívna, príkladom môže byť prechádzajúce auto, zvuky kolies na vozovke a hukot motoru, čiže abstraktnejší zdroj alebo jeho parciálne časti. Zvukové udalosti sú zvyčajne vhodne definované v krátkom časovom úseku. Oproti tomu zvukové, resp. akustické scény odkazujú na komplexný zvuk, ktorý je tvorený spojenými zvukmi viacerých zdrojov, zvyčajne z reálneho prostredia, ako napríklad zvuková scéna ulice môže obsahovať zvuky prechádzajúceho auta, zvuk krokov, komunikáciu ľudí a iné. Zvuková scéna dom môže byť zložená zo zvukov práčky, hudby z rádia a detského smiechu [1].

Úloha extrakcie informácií o akustickej udalosti a scény z audio signálu v prípade použitia techník strojového učenia spadá do kategórie strojového vnímania, konkrétne strojové počúvanie (z angl. machine hearing), ktoré je podľa Bello et al. [2] akustickým ekvivalentom k strojovému videniu (z angl. machine vision), teda tiež kombinuje techniky spracovávania signálov so strojovým učením a vytvára tak systém, ktorý je schopný extrahovať užitočné informácie zo zvukov. Zjednodušene môžeme strojové počúvanie popísať ako schopnosť identifikovať a rozlišovať zvuky prítomné v audio signáli, s cieľom dosiahnuť rozpoznávanie zvuku na ľudskej úrovni [3]. Medzi typické úlohy strojového počúvania patrí klasifikácia, ktorej cieľom je kategorizovať akustické nahrávky do preddefinovaných kategórií. Klasifikovať môže jednotlivé akustické udalosti alebo celé akustické scény. Ďalšia úloha je detekcia, pri ktorej je cieľ určiť čas, kedy je špecifikovaný zvuk alebo zvuky aktívny. Zo špecifickejších úloh je jedna napríklad o odhadovaní, či dve audio nahrávky pochádzajú z jednej akustickej scény.

Z ohľadom na pôvod zvukovej informácie, rozpoznávame niekoľko oblastí výskumu. Tou najrozšírenejšou oblasťou je výskum rozpoznávania ľudskej reči, ktorej kľúčové úlohy zahŕňajú rozpoznávanie kľúčových slov, rozpoznávanie sekvencie slov vo vete alebo rozpoznávanie identity osoby, ktorá rozpráva. Druhá oblasť výskumu pracuje s hudbou, ktorá sa nazýva „získavanie hudobných informácií“ (z angl. Music information retrieval). Medzi úlohy tejto oblasti patrí rozpoznávanie sekvencií nôt, ktoré sú hrané, identifikovanie

hudobného žánru alebo identifikácia hraných hudobných nástrojov. Ďalšia oblasť výskumu je analýza každodenných zvukov, ktoré nepatria do predchádzajúcich oblastí, čiže zvuky okrem ľudskej reči a hudby. Nájdeme však paralely medzi jednotlivými úlohami týchto oblastí, napríklad klasifikácia akustickej scény, kedy chceme priradiť jedno označenie ako „reštaurácia“ alebo „park“ je príbuzná s úlohou rozpoznávania rečníka a rozpoznávaním hudobného žánru. Obdobne, úloha označovania zvukov, ktorej cieľom je priradiť množinu označení k nahrávke, napríklad pomenovanie počuteľných objektov, je príbuzná rozpoznávaniu hudobných nástrojov v nahrávke. Úloha detekcie akustických udalostí, ktorej cieľ je identifikovanie zvukových udalostí v dobe ich vzniku, v rámci audio signálu, je úloha príbuzná automatickému rozpoznávaniu reči alebo automatického prepisu hudby [1]. Aj vďaka týmto podobnostiam sú techniky, ktoré boli vyvinuté pre jednu úlohu, prenášané do iných oblastí. Je však dôležite si uvedomovať rozdiely v signáloch jednotlivých oblastí ako napríklad, že hudobný signál je zložený zo zvukov hudobných nástrojov, ktoré boli navrhnuté, aby mali harmonickú štruktúru, každodenné zvuky túto vlastnosť nezdieľajú.

1.1 Organizácia environmentálnych zvukov

Na základe fyzikálnych charakteristík (trvanie, frekvencia, fluktuácia, dynamika), estetických kvalít a referenčných aspektov navrhol M. Schafer tri rozličné klasifikačné schémy [4]. Tieto referenčné aspekty odkazujú na kategórie zdrojov a funkcií zvuku, popísaných ako:

- zvuky prírody – napríklad zvuky produkované vodou, zvieratami či ohňom
- ľudské zvuky – vytvárané priamo ľuďmi, napríklad kroky, kašeľ či tlkot srdca
- zvuky spoločnosti – odkazujú na zvuky ľudských aktivít, napríklad oslavy alebo rôznych typov prostredia, napríklad mesto, prístav či kuchyňa
- mechanické zvuky – zvuky strojov, napríklad traktor, vrtuľník či vŕtačka
- ticho
- zvuky indikátorov – zvuky, ktoré plnia určitú informačnú funkciu, napríklad siréna, zvon alebo požiarny hlásič

Delage vo svojej štúdii [5] navrhol klasifikáciu environmentálnych zvukov na základe stupňa ľudskej aktivity do troch kategórií:

- zvuky neprodukované človekom, napríklad zvuky prírody
- reflektujúce ľudskú aktivitu nepriamo – napríklad doprava, zvuky stavby
- reflektujúce ľudskú aktivitu priamo – napríklad chôdza, smiech

V kontexte zvukovej ekológie v prírodných ekosystémoch ako sú veľké parky a rezervácie, Pijanowski et al. navrhli kategorizáciu zvukov do kategórií [6]:

- geofónia (angl. geophony) – zvuky geofyzikálneho prostredia, ako napríklad dážď alebo vietor
- biofónia (angl. biophony) – zvuky produkované biologickými organizmami, napríklad vytie vlkov či spev vtákov
- antropofónia (angl. anthrophony) – zvuky produkované priamo alebo nepriamo ľudskou činnosťou, čiže motorové vozidlo, smiech či kašeľ

Vyššie popísané schémy predstavujú kategorizáciu zdrojov zvuku, ktoré ako také síce neboli využité v klasifikačných štúdiách s poslucháčmi, avšak ich princípy využité pre analýzu izolovaných zvukov boli. Avšak tieto klasifikačné schémy neberú v úvahu štruktúru akustických udalostí s ohľadom na rôznu mieru abstraktnosti [7]. Rôzne princípy kategorizácie môžu koexistovať, zvlášť s ohľadom na zdroj zvuku a činnosť produkujúcu zvuk. Salamon et al. navrhli taxonómiu mestských zvukov, ktorá do určitej miery zahŕňa činnosti produkujúce zvuk [8]. Na najvyššej úrovni sú štyri kategórie - človek, príroda, mechanické a hudba. Na nižších úrovniach sú potom kategórie zdrojov zvukov, ktoré sú dostatočne rozdelené, aby boli jednoznačné, čiže napríklad brzdy auta, motor auta alebo klaksón namiesto jednoducho auto. Tieto parciálne zvuky boli získané analýzou sťažností na hluk, ktoré boli podané v New Yorku v rokoch 2010-2014. Zároveň sú niektoré zdroje zvukov asociované z rôznymi činnosťami, napríklad podkategória auta je motor, ktorý môže zrýchľovať alebo bežať naprázdno. Takýto popis kategórií je v súlade so štúdiou, ktorú vykonali Morel et al., v ktorej vytvorili typológiu zvukov vozidla na základe voľnej verbálnej kategorizácie [9]. Táto štúdia bola zložená z dvoch experimentov, v prvom predstavili zvukové emisie prechádzajúceho vozidla skupine poslucháčov, ktorý ich verbálne kategorizovali bez vopred stanovených tried. V druhom experimente predstavili rovnaké zvukové emisie inej skupine poslucháčov, aby vytvorili párové porovnanie. Na základe týchto experimentov potom navrhli typológiu, ktorej štruktúru riadili dve hlavné kritéria, typ vozidla a stav jazdy.

1.2 Klasifikácia environmentálnych zvukov

Výskum systematickej klasifikácie zvukov reálneho sveta má svoje začiatky už v deväťdesiatych rokoch. Jeden z prvých systémov bol SoundFisher [10] navrhnutý Wold et al. v 1996. Tento fungoval na princípe poskytovania prístupu k databáze izolovaných

zvukových efektov na základe podobnosti. Každá nahrávka bola reprezentovaná vektorom príznakov, ktorý obsahoval príznaky ako hlasitosť, výška či jas a mal fixnú veľkosť. Využitie metód strojového učenia pri klasifikácii environmentálnych zvukov, môžeme nájsť už v roku 2006 kedy Wang et al. [11] navrhli hybridný klasifikátor, na základe k-najbližších susedov (k-NN) a metódy podporných vektorov (SVM). V tom istom roku Chu et al. [12] použili rozpoznávanie okolitého zvuku pre lokalizáciu robota.

S ohľadom na pôvod akustických emisií poznáme niekoľko oblastí výskumu klasifikácie environmentálnych zvukov. Jednou z týchto oblastí je snímanie a analýza zvukov, ktoré je možne bežne počuť v meste, v tzn. mestskej zvukovej oblasti. Napriek tomu, že charakteristiky týchto zvukových oblastí sa líšia vzhľadom na mesto, resp. susedstvo pôvodu, stále zdieľajú niektoré vlastnosti, ktoré ich odlišujú od iných zvukových oblastí. Tou asi najdôležitejšou je skladba zvukov, zatiaľ čo vo vidieckej zvukovej oblasti dominujú zvuky geofónie a biofónie v mestských zvukových oblastiach sú zvuky primárne z kategórie antropofónie. Monitorovanie mestskej zvukovej oblasti sťažuje fakt, že sa jedná o akusticky veľmi bohaté prostredie – množstvo rôznych zvukov, ktoré sú husto premiešané. Zároveň sú charakteristiky týchto zvukov veľmi rôznorodé, od impulzných zvukov ako výstrel po konštantné bzučanie motora, od zvuku vzduchotechniky, ktorý je bližší šumu, po harmonický zvuk ako je hlas [13]. Interakcie medzi viacerými zdrojmi zvuku a vybudovaným prostredím, ktoré je často husté a vysoko reflektívne, vytvárajú rôzne úrovne okolitého hukotu. Hluk ako taký je rastúcim problémom mestských častí, a so zvyšujúcou sa urbanizáciou rastie aj zvukové znečistenie. Preto vzniklo viacero projektov skúmajúcich túto problematiku, napríklad Dublin City Noise [14], The Sounds of New York (SONYC)[15][16] alebo Barcelona Noise Monitoring System [17].

Ďalšia oblasť, ktorá sa zaoberá rozpoznávaním zvuku prostredia je bioakustika. Táto oblasť skúma zvuk v biologickom kontexte, čo zahŕňa disciplíny ako mechanické šírenie zvuku prostredím či mechanizmy produkovania zvukov zvierat. Bioakustika je čoraz viacej dôležitá pre biodiverzitu [18]. Mnohé zvieratá a ekosystémy sú ohrozené ľudskou činnosťou, preto vznikajú mnohé projekty pre automatické alebo poloautomatické monitorovanie, ako napríklad Automatická klasifikácia druhov vtáctva [19] alebo Akustická identifikácia európskych netopierov[20]. Bioakustická analýza zároveň pomáha porozumieť problémom ako komunikácia zvierat, speciácia alebo kultúrna evolúcia. Charakteristika zvukov v prípade bioakustiky je rozmanitá. Mnohé cicavce vokalizujú zhruba podobne ako zvuky ľudských samohlások, čo má za následok harmonické zhluky rezonancií podobné

slovotvornému formantu, príklad môže byť vytie psov alebo vlkov. Niektoré vtáky taktiež produkujú harmonické zvuky, ale mnohé taktiež produkujú čisté tonové zvuky alebo zvuky veľmi podobné šumu; spevavce majú dedikované svalstvo, ktoré im dovoľuje vydávať komplexné zvuky s veľmi rýchlou frekvenčnou moduláciou. Obojživelníky a hmyz produkujú jednoduchšie, stereotypnejšie hlasové jednotky, viaceré z nich sa však zapájajú do koordinovaných skupinových volaní, z ktorých je ťažké identifikovať zvuky jednotlivcov. Z toho vyplýva, že v rámci analýzy bioakustiky je nutné sa vysporiadať so širokou škálou zvukových typov - harmonické alebo iné, stereotypné alebo vysoko rozmanité. Túto škálu je však možné zjednodušiť v jednodruhových štúdii [21].

Ďalšou oblasťou, ktorá sa dostáva do popredia, je rozpoznávanie akustickej udalosti v rámci domova. V rámci rozpoznávania audio signálu v domovoch sú v popredí tzn. Inteligentní domáci asistenti, ako napríklad Amazon Echo alebo Google Home, ktorí fungujú na princípoch automatického rozpoznávania reči, syntéze reči a systéme umelého dialógu. Aj rozpoznávanie akustických udalostí našlo svoje uplatnenie. Na najnižšej úrovni môžu poskytovať sofistikovaný spôsob snímania, napríklad označovaním prítomnosti vybraných zvukov alebo rozpoznávaním výskytu určitej scény nad rámec obyčajných meraných úrovní hlasitosti. Na vyššej úrovni môžu poskytovať sémantické interpretácie toho, čo sa deje v dome, napríklad zvuk požiarneho alarmu indikuje možnú prítomnosť ohňa. Príkladom takéhoto asistenta je ai3 od Audio Analytic [22], ktorého možné použitie zahŕňa detekciu rozbitia okna v prázdnom dome, detekcia zvukov hlásiča dymu alebo CO v prázdnom dome alebo napríklad detekcia anomálie [23].

V neposlednom rade je to výskum všeobecných klasifikátorov environmentálnych zvukov, ktorý nemá hlbší informačný presah. Táto klasifikačná úloha obsahuje podmnožinu zvukov z vyššie spomenutých oblastí. Hlavné zameranie v tejto oblasti je vývoj nových klasifikačných metód, nových architektúr a metód extrakcie príznakov.

2 Strojové učenie

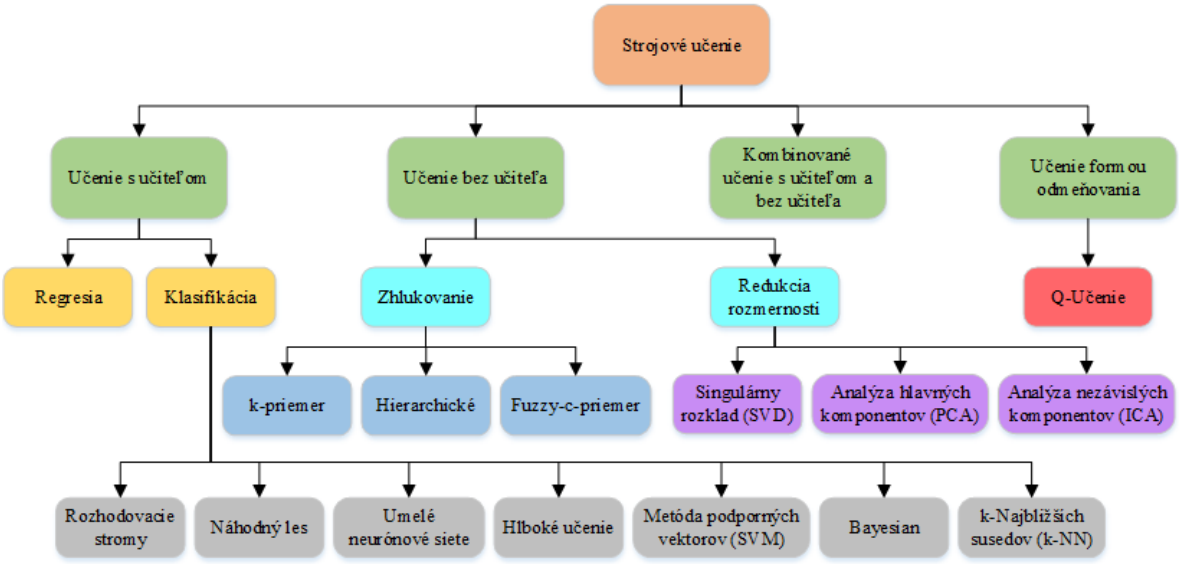
Metódy strojového učenia (ML, z angl. Machine Learning) implementujú algoritmy a štatistické modely, pomocou ktorých efektívne vykonávajú úlohy, bez nutnosti explicitne naprogramovať inštrukcie, pomocou ktorých sa majú tieto úlohy vykonávať. Namiesto toho sa pomocou algoritmu učenia naučí, ako danú úlohu vykonávať na základe poskytnutých dát. Sila strojového učenia je v jeho schopnosti poskytovať generalizované riešenie prostredníctvom architektúry, ktorá reprezentuje komplexné vzťahy v dátach. Využitím týchto metód môžeme docieliť, že výpočtové procesy budú efektívnejšie, spoľahlivejšie a cenovo výhodnejšie. ML sa konvenčne rozdeľuje do kategórií na základe procesu učenia, tieto kategórie sú:

- Učenie s učiteľom (angl. Supervised learning) – V prípade tohto typu učenia, poskytneme modelu množinu vstupných a výstupných dát (dataset z označeniami). Počas procesu učenia model upravuje svoje parametre na základe priameho porovnania výstupu modelu a požadovaného výstupu. Výsledný model potom reprezentuje vzťahy a závislosti medzi vstupnými dátami a predpokladanými výstupmi. Po skončení procesu učenia môžeme nájsť funkciu zo vstupu x s najlepšie odhadnutým výstupom y ($f: x \rightarrow y$). Učenie s učiteľom sa využíva v úlohách regresie a klasifikácie.
- Učenie bez učiteľa (angl. Unsupervised learning) - Pri tomto type učenia nemá model k dispozícií výstupné dáta. Učebný proces je založený len na vstupných dátach, z ktorých si model sám extrahuje informácie, len na základe korelácie vstupných dát, s cieľom nájsť významný vzor alebo príznak vo vstupných dátach bez pomoci učiteľa. Proces učenia je ukončený na základe vopred nastaveného kritéria, bez ktorého by tento proces pokračoval aj keby bol modelu poskytnutý vzor, ktorý nie je súčasťou trénovacej množiny, model by sa adaptoval na konštantne sa meniace prostredie. Využitie tohto druhu učenia nájdeme v úlohách redukcie rozmernosti alebo zhlukovania.
- Kombinované učenie s učiteľom a bez učiteľa (angl. Semi-supervised learning) – Vo viacerých ML aplikáciách, ako napríklad bioinformatika, kategorizácia textu, rozpoznávanie tváre a iné, je možné zozbierať veľké množstvo neoznačených dát relatívne ľahko, resp. za pomoci automatizácie. Avšak proces manuálneho

označovania dát je zvyčajne pomalý, drahý a náchylný na chyby. V prípade, že je k dispozícii len malé množstvo označených dát, neoznačené dáta môžu byť použité k zabráneniu degradácie výkonu z hľadiska preučenia. Pri kombinovanom učení rozpoznávame dva základné ciele, predikciu označenia na neoznačených dátach v tréningovom sete a predikcia označenia na budúcej testovacej množine. Použitie nájdeme v klasifikácii čiastočne označených dát, zhlukovaní označených aj neoznačených dát, či redukcii rozmernosti označených dát.

- Učenie formou odmeňovania (angl. Reinforcement learning) – Tato kategória algoritmov špecifikuje ako sa inteligentný agent, napríklad skutočný či simulovaný robot, môže učiť z poskytnutých akcií tak, aby maximalizoval odmenu. Je to špeciálny prípad učenia s učiteľom, pri ktorom presný výstup nie je známy. Učiteľ poskytuje iba spätnú väzbu ohľadom úspešnosti alebo neúspešnosti vykonanej akcie. V prípade tohto typu učenia je model odmenený za dobrý výsledok a penalizovaný za zlý výsledok. Tento druh učenia sa využíva napríklad v úlohách autonómneho riadenia vozidla či v hernom priemysle pre tvorbu pokročilých simulovaných hráčov.

Detailnejšia taxonómia strojového učenia [24] je zobrazená na obrázku 1.



Obrázok 1 Taxonómia strojového učenia

2.1 Metódy klasifikácie

V obore strojového učenia je klasifikácia druh problému, pri ktorom je cieľom rozdeliť dáta do skupín na základe logického rozhodovania, čo najbližšie skutočnému rozdeleniu. Klasifikácia je teda úloha rozpoznávania vzorov. Ako je uvedené v taxonómii na obrázku 1, patrí táto metóda do kategórie učenia s učiteľom. Klasifikátor je potom

algoritmus, ktorý implementuje klasifikáciu. Hlavná úloha klasifikátora je identifikovať triedu, do ktorej patria nové pozorované vzorky. Tento algoritmus je vytváraný na základe trénovacej množiny dát, ktorá obsahuje vstupné vzorky aj ich výstupné triedy. Následne po tejto fáze trénovania by mal byť klasifikátor schopný kategorizovať do naučených tried aj vopred neznáme dáta. Typickým príkladom môže byť kategorizovanie obrázkov psov na základe ich rasy, lokalizácia objektov vo fotografii, určovanie, či je daný text pozitívny alebo negatívny v danej téme/kontexte alebo rozhodnutie z akustickej nahrávky, aký zvuk je prítomný. Rozpoznávame tri základné typy klasifikačných úloh:

- Binárna klasifikácia – V prípade tejto úlohy sú dáta rozdelené do dvoch kategórií. Zvyčajne, pri použití binárnej klasifikácie je jedna trieda delegovaná ako normálny stav a druhá ako abnormálny stav. Ako príklad môže byť určovanie, či daný email je spam alebo nie. V tomto príklade je „nie je to spam“ normálny stav a „je to spam“ je abnormálny stav. Populárne algoritmy pre binárnu klasifikáciu sú: Neurónové siete, k-Najbližších susedov, Rozhodovacie stromy a Metóda podporných vektorov.
- Klasifikácia do viacerých tried s jedným označením – Táto klasifikačná úloha zahŕňa dáta, ktoré sú rozdelené do viacerých tried, avšak každá vzorka môže patriť len do jednej kategórie, ktoré sú si vzájomne exkluzívne. Príkladom takejto úlohy môže byť rozpoznávanie tváre, kde model klasifikuje fotografiu tváre do jednej z mnohých kategórií rozpoznávajúceho systému. Niektoré prístupy adaptujú algoritmy binárnej klasifikácie, kde stratégia rozhodovania je buď binárny klasifikačný model pre každú triedu, ktoré vyhodnocujú jednu triedu proti ostatným (nazývaná jeden-proti-zvyšku), alebo jeden model pre každú kombináciu tried (nazývaná jeden-proti-jednému). Populárne algoritmy pre klasifikáciu do viacerých tried z jedným označením sú: Neurónové siete, Náhodný les alebo k-Najbližších susedov. V prípade stratégie použitia viacerých binárnych klasifikátorov sa ponúka algoritmus Metóda podporných vektorov.
- Klasifikácia do viacerých tried z viacerými označeniami – Tento typ klasifikačnej úlohy využíva dáta, ktoré sú rozdelené do viacerých tried, avšak tieto triedy nie sú exkluzívne, teda jedna vzorka môže patriť do viacerých tried súčasne. Typický príklad tejto úlohy je lokalizácia objektov na fotografii, kde sa na jednej fotografii nachádza viacero známych objektov, napríklad „Osoba“, „Pes“, „Banán“ a tak ďalej. Algoritmus, ktorý natívne podporuje tento druh klasifikácie je Neurónová sieť a Hlboké učenie. Ďalej, niektoré verzie klasifikačných algoritmov boli adaptované tak,

aby podporovali tento druh klasifikácie, sú napríklad Rozhodovacie stromy pre viaceré označenia či k-Najbližších susedov pre viaceré označenia.

V nasledujúcej časti budú stručne popísane jednotlivé metódy klasifikácie.

2.1.1 Rozhodovacie stromy

Algoritmus Rozhodovacie stromy pozostáva zo série otázok typu ak-potom, ktoré tvoria stromovú štruktúru, pomocou ktorej je predikovaná výstupná trieda. Pri budovaní rozhodovacieho stromu používame viacero typov uzlov: koncové uzly alebo listy (výstupná trieda), ktoré reprezentujú triedu; rozhodovacie alebo vnútorné uzly, ktoré testujú príznaky vstupnej vzorky a pre výsledok testu je vytvorená vetva; najvyšší uzol predstavuje vstupný alebo koreňový uzol. Každá nová vzorka je klasifikovaná nasledovaním cesty od koreňového uzla smerom k listovému, kde rozhodovacie uzly rozhodujú o ceste. Výsledný listový uzol je považovaný za výsledok klasifikácie [25]. Hlavnými výhodami rozhodovacieho stromu je transparentnosť, znižuje nejednoznačnosť rozhodovacieho procesu a umožňuje obsiahlu analýzu.

2.1.2 Náhodný les

Algoritmus náhodný les využíva skupinu rozhodovacích stromov, kde každý strom v lese poskytuje klasifikáciu. Vytvorenie viacerých rozhodovacích stromov na podmnožine trénovacích dát a náhodností príznakov dát má za následok zníženie odchýlky. A teda hlavná myšlienka za náhodným lesom je vytvorenie lesa rozhodovacích stromov, z ktorých každý je natrénovaný na inom stĺpci z množiny dát pre trénovanie. Ak má každý strom šancu korektnej klasifikácie vyššiu ako 50%, jeho predikcia je zahrnutá do finálneho rozhodnutia. Vo svojej podstate vytvárame porotu a mal by byť pridaný každý, koho šanca na určenie správneho verdiktu je vyššia ako 50%, a teda zakaždým, keď je takýto člen pridaný, je výsledok presnejší za predpokladu, že sa každý člen rozhoduje samostatne. Náhodný les efektívne pracuje s väčšími množinami dát a heterogénnymi dátami. Avšak v posledných rokoch je táto technika používaná čoraz menej, z dôvodu vzostupu Rozhodovacích stromov s posilnením gradientu, ktoré väčšinou prevyšujú náhodné lesy s podobnými vlastnosťami. Z oblasti klasifikácie environmentálnych zvukov, dosiahol tento druh klasifikátora presnosť 44.3% [26] v prípade klasifikácie 5 sekundových nahrávok zvukov do 50 tried.

2.1.3 Metóda podporných vektorov

Metóda podporných vektorov je algoritmus, ktorý sa pokúša určiť optimálnu nadrovinu medzi rozličnými triedami, ktorá maximalizuje vzdialenosť medzi nadrovinou

a bodmi triedy. Týmto spôsobom sa pokúša nájsť významnú separáciu medzi triedami. Podporné vektory sú body na hrane rozdeľujúcej nadrovinu. Algoritmus poskytuje najlepšiu klasifikáciu z poskytnutej množiny dát. Z toho dôvodu nie je komplexnosť modelu ovplyvnená množstvom príznakov v trénovacích dátach. Preto je metóda podporných vektorov využívaná v úlohách, kde počet príznakov je vysoký z ohľadom na množstvo trénovacích vzoriek. Základný algoritmus Metódy podporných vektorov je použiteľný len pre binárnu klasifikáciu, avšak boli navrhnuté rozšírenia, ktoré umožňujú aj klasifikáciu do viacerých tried. Aj tento algoritmus bol použitý pre klasifikáciu environmentálnych zvukov a jeho presnosť dosiahla 39.6% na rovnakej množine dát.

2.1.4 k-Najbližších susedov

Tento algoritmus je považovaný za jeden z najstarších bez-parametrických klasifikačných algoritmov. Patrí to kategórie algoritmov, ktoré sa učia na základe príkladu, kde model predpokladá, že vzdialenosť je dostatočná pre dedukciu. K-Najbližších susedov klasifikuje na základe vzdialenosti neznámej vzorky od všetkých ostatných vzoriek, *k* najmenších vzdialeností je identifikovaných, a tá trieda z celkového množstva *k* tried, ktorá je najviac reprezentovaná, je považovaná za výstupnú. Pre výpočet vzdialenosti sa používajú rôzne funkcie, napríklad Euklidovská vzdialenosť, Hammigová vzdialenosť, Manhattanská vzdialenosť a iné. Komplexnosť modelu k-Najbližších susedov je závislá od veľkosti vstupnej množiny dát a optimálny výkon dosahuje, ak sú dáta rovnako škálované. Tento prístup dosiahol v oblasti rozpoznávania environmentálnych zvukov presnosť rozpoznávania 32.2% v rovnakej množine dát ako predchádzajúce dva algoritmy.

2.1.5 Umelé neurónové siete

Známe tiež ako Neurónové siete je metóda určená na štruktúru dát pre distribuované a paralelné spracovávanie dát. Táto metóda patrí do kategórie metód strojového učenia, ktoré boli voľne inšpirované fungovaním biologického neurónu v ľudskom mozgu, avšak je potrebné chápať, že neurónové siete nie sú modelom mozgu. Klasifikátor založený na neurónovej sieti sa skladá z jednotiek/neurónov, usporiadaných do vrstiev, ktoré prevádzajú vstupné vektory na výstup. Tieto vrstvy sú typicky prepojené uzlami, ktoré sú asociované s niektorou z aktivačných funkcií. Každá neurónová sieť pozostáva z troch typov vrstiev: vstupná vrstva, jedna a viac skrytých vrstiev a nakoniec výstupná vrstva. Neurónové siete sú schopné klasifikovať komplexné, nelineárne dátové množiny relatívne jednoducho a ich vstupy nie sú obmedzené ako pri iných klasifikačných metódach.

2.1.6 Hlboké učenie

Algoritmus hlbokého učenia môžeme chápať ako podkategóriu Umelých neurónových sietí, ale vo svojej podstate môže odkazovať aj na iné techniky strojového učenia, založené na vrstvách. Hĺbka v hlbokom učení odkazuje na počet vrstiev, ktoré sa nachádzajú medzi vstupnou a výstupnou vrstvou, ktoré sú v moderných hlbokých modeloch v rámcoch desiatok alebo až tisícok a všetky sa súčasne učia vo fáze trénovania. Takáto neurónová sieť je zostavená z jednoduchých nelineárnych modulov, ktoré postupne extrahujú vysokoúrovňové príznaky zo vstupu, ako postupne prechádzajú z nižších vrstiev do vyšších. Napríklad v rozpoznávaní obrazu nižšie vrstvy rozpoznávajú hrany a čím je vrstva vyššie, tým abstraktnejšie príznaky sú extrahované. Veľkou výhodou hlbokého učenia je jeho škálovateľnosť. Tým, ako sa veľkosť neurónovej siete zväčšuje, je možné ich trénovať na čoraz väčšom množstve dát a ich výkon sa neprestáva zvyšovať na rozdiel od iných techník strojového učenia.

2.1.7 Naivný Bayes

Naivný Bayes alebo tiež Naivný Bayesov klasifikátor patri do kategórie štatistických klasifikátorov. Klasifikačný model v prípade štatistického prístupu využíva ako podklad pravdepodobnosti. Výstupom modelu je pravdepodobnosť, v ktorej vzorka patrí do každej triedy na rozdiel od iných, ktorých výstup je trieda vzorky. Algoritmus klasifikácie je založený na Bayesovom teoréme s predpokladom, že príznaky v dátach sú vzájomne nezávisle. Predpokladá, že prítomnosť konkrétneho príznaku v triede nesúvisí s prítomnosťou akéhokoľvek iného príznaku. V prípade, že tieto príznaky sú na sebe závislé, alebo na existencii inej vzorky, všetky tieto príznaky nezávisle prispievajú k výslednej pravdepodobnosti. Výhoda tohto algoritmu je, že nepotrebuje veľkú množinu dát pri vytváraní modelu, ako aj to, že svojou jednoduchosťou prekonáva aj iné sofistikované klasifikačne algoritmy. Tento algoritmus našiel svoje využitie v oblasti spracovávania prirodzeného jazyka.

2.2 Neurónové siete

Neurónové siete, alebo tiež známe aj ako umelé neurónové siete alebo simulované neurónové siete, je technika strojového učenia, ktorej štruktúra bola inšpirovaná ľudským mozgom. Napodobňuje spôsob, ktorým biologické neuróny signalizujú medzi sebou. Vďaka tomuto pozadiu sú neurónové siete vysvetľované na vyššej úrovni za pomoci neurobiologických termínov, ako neurón, axón a synapsie, ktoré ich spájajú [27], avšak ako

už bolo spomenuté, napriek tomu, že neurónové siete boli inšpirované biologickým fungovaním mozgu, nie sú modelom mozgu.

2.2.1 Historický vývoj a použitie neurónových sietí

Historický vývoj umelých neurónových sietí má počiatok v prvej polovici 20. storočia, presnejšie v roku 1943 kedy W. S. McCulloch a W. Pitts [28] vytvorili jednoduchý matematický model neurónu a D. Hebb [29] navrhol pravidlá učenia pre synapsie. Ďalším dôležitým míľnikom bol návrh perceptrónu F. Rossenblattom [30] v roku 1957. Tento objav začal teoretický aj praktický vývoj neurónových sietí. Avšak z dôvodu, že táto sieť nevedela klasifikovať lineárne neseparabilné problémy, výskum v tejto oblasti upadol. Novú éru neurónových sietí začal v roku 1982, J. Hopfield [31]. Za jeden z najprominentnejších míľnikov vo výskume neurónových sietí je algoritmus učenia spätnej propagácie pre viacvrstvový perceptrón, ktorý publikovali D.E. Rumelhart et al. v roku 1986 [32], taktiež ho implementoval Y. LeCun [33] v roku 1987. Tretia vlna výskumu neurónových sietí začína objavom v roku 2006, kedy G. Hinton et al. [34] ukázal typ neurónovej siete, ktorý natrénoval pomocou stratégie lakomého predtrénovania po vrstvách. Okolo roku 2010, začali neurónové siete a hlboké učenie prekonávať ostatné metódy strojového učenia. Ako dôkaz môže slúžiť, že v pravidelnej súťaži rozpoznávania obrazu ImageNet [35] sa v roku 2012 umiestnil na prvom mieste AlexNet, model založený na konvolučnej neurónovej sieti s veľkým predstihom.

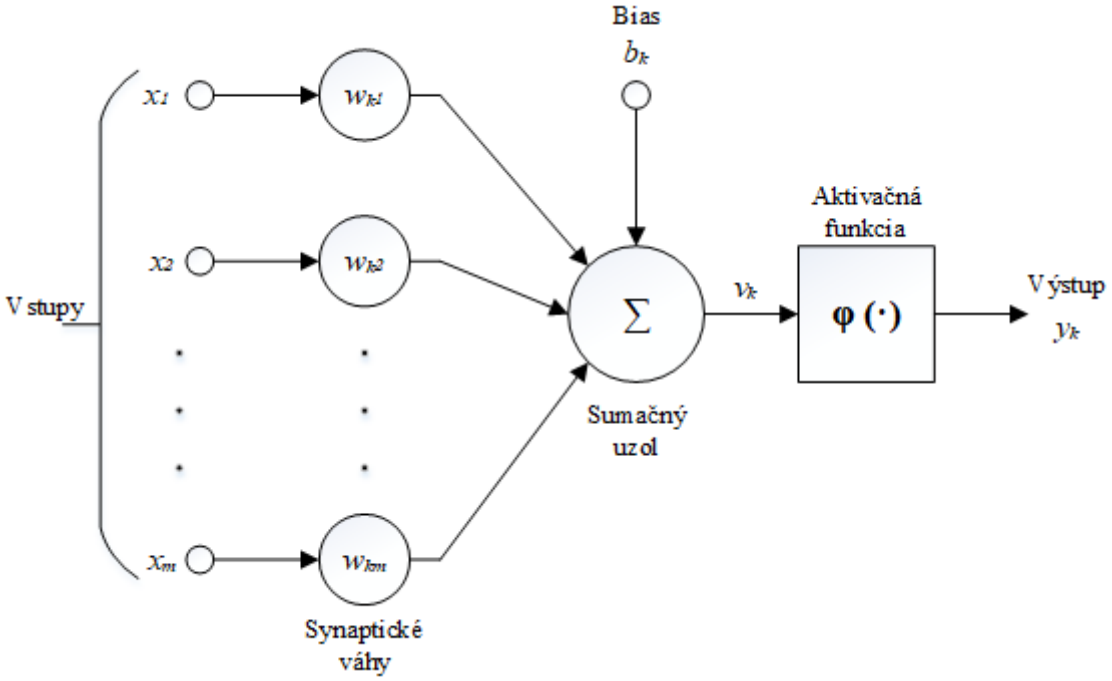
Neurónové siete našli svoje využitie v mnohých vedeckých a inžinierskych disciplínach a ich využitie pomáha riešiť rôzne komplexné problémy. Najme vďaka vývoju výpočtovej techniky vieme implementovať tieto techniky nielen v experimentálnych úlohách ale napomáhajú aj riešeniu rôznych problémov reálneho sveta. Svoje využitie našli v oblastiach strojového videnia, ako napríklad rozpoznávanie vzorov či segmentácia obrazu, strojové počúvanie, rozpoznávanie prirodzeného jazyka, v rôznych úlohách klasifikácie, úlohy optimalizácie procesu, strojový preklad jazyka alebo syntéza nových dát. V priemyselnej oblasti sú pomocou nich riešené detekcie rôznych chybových udalostí, prediktívna údržba strojov či kontrola kvality. Z finančníctva sa jedná o úlohy detekcie podvodu alebo predikcie rôznych hodnôt, od cien akcií po pravdepodobnosť bankrotu. V medicíne napomáhajú doktorom v diagnostike chorôb. Vo všeobecnosti sú neurónové siete využiteľné pre nasledujúce funkcie [36]:

- Aproximácia funkcie

- Klasifikácia
- Zhlukovanie a vektorová kvantizácia
- Optimalizácia
- Asociatívna pamäť
- Extrakcia príznakov

2.2.2 Neurón

Neurón je jednotka spracovávania informácií, ktorá je fundamentálna pre fungovanie neurónovej siete [35]. Prvý model neurónu (McCulloch-Pitts neurón) môžeme rozdeliť na dve časti, prvá vstupná časť urobí agregáciu a podľa tejto agregovanej hodnoty druhá časť urobí rozhodnutie. Aj vstupné aj výstupné hodnoty sú typu boolean. Na vstupe taktiež rozlišujeme dva druhy hodnôt, inhibičné a excitačné. Inhibičný vstup má hlavný účinok rozhodovania bez ohľadu na iné vstupy, teda ovplyvňujú aktiváciu neurónu. Excitačný vstup sám o sebe neovplyvní aktiváciu neurónu, aktivácia je možné len v kombinácii viacerých. Tento model využíval Booleovské funkcie ako AND, OR, NOT pre rozhodnutie. Tento model mal viacero nedostatkov a v dnešnej dobe sa skôr používa všeobecný základný model umelého neurónu. Štruktúra tohto neurónu je ilustrovaná na obrázku 2.



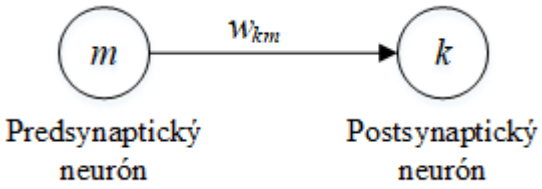
Obrázok 2 Model umelého neurónu [37]

Základné časti neurónu sú nasledovné [37][38][39]:

- Vstupy do neurónu $x_1, x_2, \dots, x_m$, ktoré modelujú dentrity

- Synaptické váhy, ktoré sú charakterizované hodnotou váh $w_{k1}, w_{k2}, \dots, w_{km}$ a určujú ich priepustnosť, zároveň tvoria prepojenia, ktoré spájajú jednotlivé neuróny do siete
- Sumačný uzol, ktorého výstup v_k, je vážená suma vstupných hodnôt a predstavuje vnútorný potenciál neurónu
- Prah neurónu, bias, ktorého hodnota b_k, posilňuje, resp. utlmuje vstup do aktivačnej funkcie
- Aktivačná funkcia neurónu φ, ktorá definuje výstup z neurónu s ohľadom na vstup v_k
- Výstup z neurónu y_k, ktorý modeluje elektrický impulz axónu

Je dôležité poznamenať, že označenie v prípade synaptických váh, vychádza z informačného toku, podľa ktorého rozdeľujeme neuróny na predsynaptický (zdrojové - pred synapsiou) a postsynaptický (cieľové - po synapsii). Ako je možne vidieť na obrázku 3, označenie synaptickej váhy w_{km} ukazuje, že sa jedná o prepojenie medzi postsynaptickým neurónom *k* a predsynaptickým neurónom *m*.



Obrázok 3 Označenie neurónov a synaptických váh

Z matematického hľadiska môžeme neurón *k* na obrázku 3 popísať párom rovníc [37]:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{1}$$

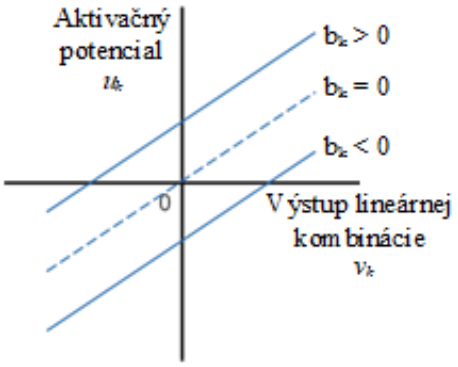
a

$$y_k = \varphi(u_k + b_k) \tag{2}$$

kde $x_1, x_2, \dots, x_m$ sú vstupné signály; $w_{k1}, w_{k2}, \dots, w_{km}$ sú jednotlivé synaptické váhy neurónu *k*; u_k (nezobrazený na obrázku 3) je výstup lineárnej kombinácie s ohľadom na vstupné signály; b_k je bias alebo prah neurónu; $\varphi(\cdot)$ je aktivačná funkcia; a y_k je výstupný signál z neurónu *k*. Použitie biasu b_k má za efekt aplikovanie afinnej transformácie na výstup lineárnej kombinácie u_k, čo môžeme zapísať ako:

$$v_k = u_k + b_k \tag{3}$$

V závislosti od toho či je bias b_k pozitívny alebo negatívny, je modifikovaný vzťah medzi aktivačným potenciálom v_k a výstupu lineárnej kombinácie u_k spôsobom, ktorý je zobrazený na obrázku 4. Ako je možné vidieť, vzhľadom na afinnu transformáciu, graf v_k verzus u_k už neprechádza počiatkom, zároveň je možné si povšimnúť, že $v_k = b_k$ v prípade keď $u_k = 0$ [37].



Obrázok 4 Afinna transformácia spôsobená prítomnosťou biasu

Bias b_k je externý parameter neurónu k, nevstupuje z iných neurónov. Môže ho zakomponovať pomocou rovnice (2). Alternatívne môžeme rovnice preformulovať nasledujúcim spôsobom:

$$v_k = \sum_{j=0}^{m} w_{kj} x_j \tag{4}$$

$$y_k = \varphi(v_k) \tag{5}$$

kde sme pridali novú synapsiu, ktorej vstup je $x_0 = +1$ a jej synaptická váha je $w_{k0} = b_k$.

Funkcia $\varphi(v)$ je nazývaná aktivačnou funkciou neurónu. Navrhnutých bolo viacero tvarov aktivačnej funkcie, z ktorých budú predstavené tie najdôležitejšie. Ich grafické zobrazenie je na obrázku 5. Tieto aktivačné funkcie sú závislé len na vstupe [37][38].

- Lineárna funkcia

$$v_k = \varphi(x_k) = x_k \tag{6}$$

- Funkcia signum, známa aj ako Heavisidova funkcia[1]

$$\varphi(v_k) = \begin{cases} 1 \; ak \; v_k \geq 0 \\ 0 \; ak \; v_k < 0 \end{cases} \tag{7}$$

[1] Neuróny s touto aktivačnou funkciou sú označované ako McCulloch-Pitts neuróny na počesť McCullocha a Pittsa (1943) a ich priekopnícku prácu.

- Funkcia sigmoid a hyperbolický tangens

$$\varphi(v_k) = \frac{1}{1 + e^{-av_k}} \tag{8}$$

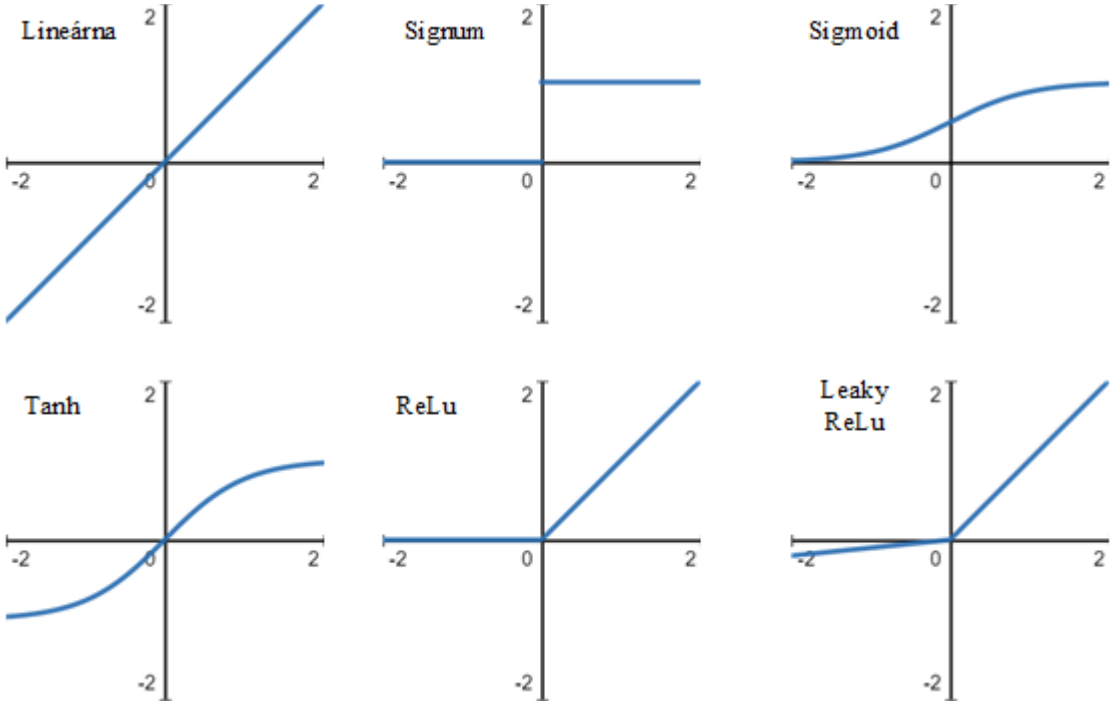
kde a je parameter strmosti. Táto funkcia je definovaná na základe rovnice (8) v rozmedzí 0 až +1. V prípade, že je nutné aby aktivačná funkcia bola v rozmedzí -1 až +1 a zároveň mala korešpondujúci tvar, môže byť použitý hyperbolický tangens $\varphi(v_k) = \tanh(v_k)$.

- Funkcia ReLU (9) a Leaky ReLu (10)

$$\varphi(v_k) = \begin{cases} v_k & ak\ v_k > 0 \\ 0 & ak\ v_k \leq 0 \end{cases} \tag{9}$$

$$\varphi(v_k) = \begin{cases} v_k & ak\ v_k > 0 \\ \alpha v_k & ak\ v_k \leq 0 \end{cases} \tag{10}$$

V dobe písania tejto práce je funkcia ReLU jednou z najpoužívanejších aktivačných funkcií v oblasti hlbokého učenia. Funkcia Leaky ReLU bola odvodená z ReLU, aby vyriešila problém takzvanej „mŕtvej“ ReLU, kde niektoré parametre neurónovej siete nebudú nikdy aktualizované, nakoľko pre záporné hodnoty nadobúda hodnotu 0. V rámci Leaky ReLU, v prípade zápornej hodnoty, je táto hodnota vynásobená parametrom α, ktorý má zvyčajne nízku hodnotu.



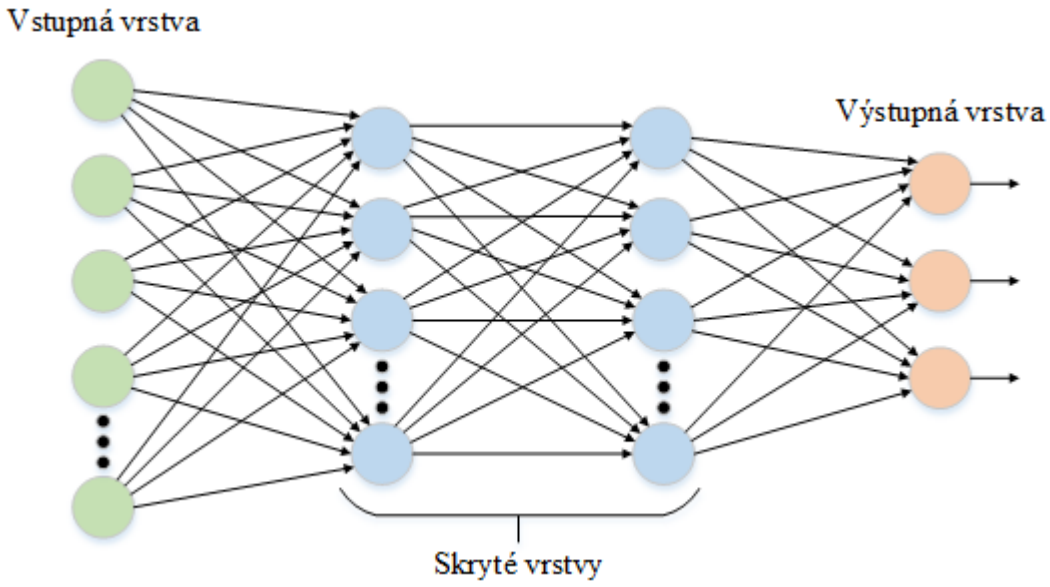
Obrázok 5 Ukážky aktivačných funkcií neurónu

Ako už popísané synaptické váhy majú svoju orientáciu. Tieto váhy ovplyvňujú vstupy do neurónov a tým aj ich stavy a v konečnom dôsledku celú neurónovú sieť. Je to práve moment zmeny váh Δw_{km}, ktorý je najdôležitejší pre činnosť neurónovej siete. Vo všeobecnosti rozdeľujeme synaptické váhy, na základe ich hodnoty, na [38]:

- kladné alebo excitačné
- záporné alebo inhibičné

2.2.3 Architektúra neurónovej siete

Architektúru neurónovej siete môžeme vo všeobecnosti popísať pomocou orientovaného grafu, kde vrcholy predstavujú neuróny a orientované hrany predstavujú synaptické prepojenia. Jednou z tradičných a pomerne dostatočne preskúmaných štruktúr je viacvrstvová štruktúra zobrazená na obrázku 6.



Obrázok 6 Viacvrstvová architektúra doprednej neurónovej siete

Ako je možné vidieť, vrstvy tejto štruktúry sú pomenované.

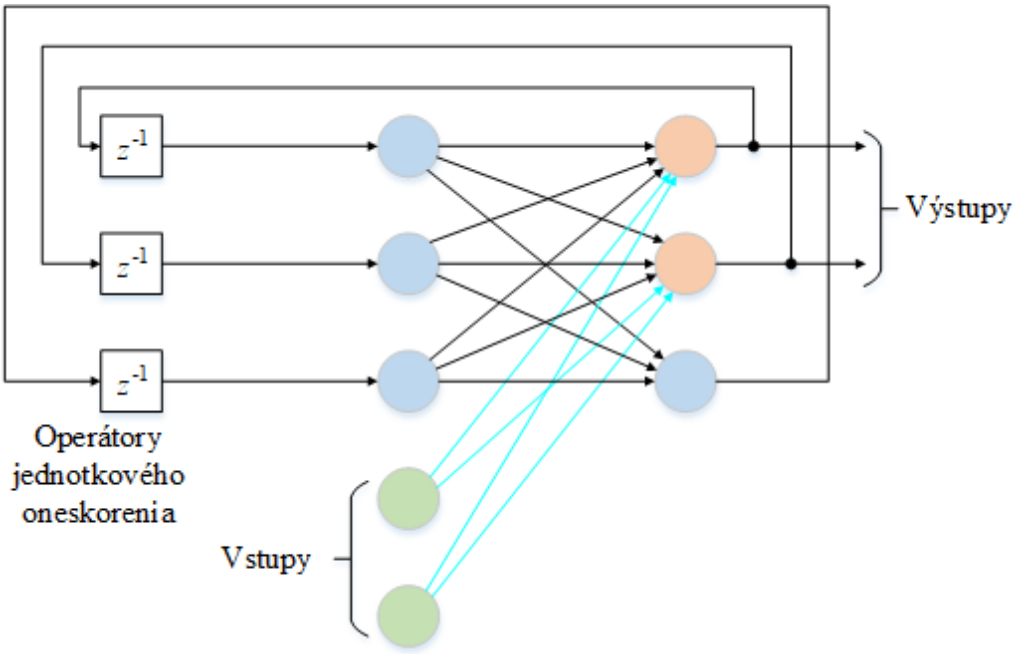
Rozpoznávame tri základné typy vrstiev[38]:

- Vstupná vrstva – neuróny tejto vrstvy prijímajú vstup z externého sveta a ich výstup je spracovávaný ďalšími neurónmi neurónovej siete.
- Skryté vrstvy – jedna alebo viacej vrstiev, ktoré sa nachádzajú medzi vstupnou a výstupnou vrstvou, ktorých neuróny prijímajú vstup z ostatných neurónov, alebo aj na základe prahového prepojenia z externého sveta. Ich výstup spracovávaný ďalšími neurónmi neurónovej siete.

- Výstupná vrstva – neuróny tejto vrstvy majú obdobnú funkciu ako u skrytých vrstiev, avšak ich výstup už nie je ďalej spracovávaný neurónovou sieťou a teda predstavuje odozvu neurónovej siete na daný vstup z externého sveta.

V tej najjednoduchšej forme má táto architektúra len vstupnú vrstvu, ktorá je priamo prepojená na výstupnú, nie však naopak. Takáto neurónová sieť je potom označovaná ako jednovrstvová sieť. Pridaním jednej alebo viacerých skrytých vrstiev neurónová sieť bude schopná zo vstupu extrahovať štatistiky vyššieho rádu [37].

Architektúra, ktorá bola predstavená na obrázku 6 patrí medzi takzvané acyklické alebo dopredné neurónové siete [39]. Informácia sa v prípade týchto neurónových sietí šíri po orientovaných synaptických prepojeniach len jedným smerom a to dopredu. Druhý základný typ architektúry je cyklická alebo rekurentná neurónová sieť. Hlavý rozdiel medzi týmito architektúrami je to, že rekurentná neurónová sieť implementuje minimálne jednu spätnú väzbu. Pri tomto druhu neurónových sietí môže byť dosť ťažké rozdelenie vrstiev na vstupné, resp. výstupné. Príklad takejto architektúry je možné vidieť na obrázku 7. Spätná väzba v prípade zobrazenej architektúry začína nielen v skrytých neurónoch, ale aj výstupných. Zároveň táto spätná väzba obsahuje jednotkové oneskorenie (označené z^{-1}), ktoré spôsobuje dynamické nelineárne chovanie, za predpokladu, že neurónová sieť obsahuje nelineárne jednotky [37]. Môžeme taktiež pozorovať, že výstup neurónovej siete nie je závislý len od aktuálneho vstupu, ale aj na základe predchádzajúcich informácií.



Obrázok 7 Architektúra rekurentnej neurónovej siete so skrytými neurónmi

Z hľadiska šírenia signálu v neurónovej sieti rozpoznávame viacero spôsobov, napríklad [28][39]:

- synchrónne, pri ktorom všetky neuróny menia svoj stav do taktu (na základe synchronizačných hodín),
- sekvenčné, pri ktorom neuróny svoj stav menia postupne pri šírení signálu,
- blok-sekvenčné, podľa vopred zvolenej stratégie, sú aktivizované len skupiny neurónov,
- asynchrónne, a teda neuróny menia svoje stavy nezávisle jeden od druhého.

2.2.4 Rosenblattov perceptrón

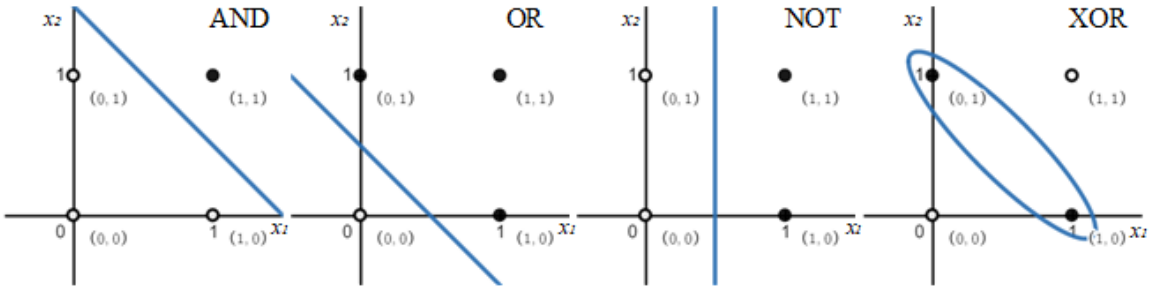
Perceptrón predstavuje najjednoduchšiu neurónovú sieť, ktorá je založená na základe jedného neurónu. Funkčnosť tohto perceptrónu je limitovaná na vykonávanie klasifikácie vzorov do dvoch tried (binárna klasifikácia). Zároveň sa predpokladá, že tieto triedy sú lineárne separovateľné. Rosenblattov perceptrón je postavený na základe nelineárneho neurónu, konkrétne McCulloch-Pitts neuróne [37].

Nech sú vstupy do perceptrónu označené $x_1, x_2, \dots, x_m$, k nim korešpondujúce synaptické váhy sú označené $w_{k1}, w_{k2}, \dots, w_{km}$, a bias b. Vnútorný potenciál tohto perceptrónu je potom definovaný pomocou rovnice:

$$v = \sum_{i=1}^{m} w_i x_i + b \tag{11}$$

Cieľom perceptrónu je správne klasifikovať externé stimuly $x_1, x_2, \dots, x_m$ do jednej z dvoch tried T_1 a T_2. Ako bolo spomínané McCulloch-Pitts neurón používa ako aktivačnú funkciu signum, teda môžeme vytvoriť rozhodovacie pravidlo, na základe ktorého môžeme určiť, že ak výstup z perceptrónu je 0, tak patria do T_1 a ak má výstup hodnotu 1, patria do T_2. Najjednoduchší spôsob ako rozdeliť m-dimenzionálny priestor tvorený m vstupnými hodnotami $x_1, x_2, \dots, x_m$, na dva rozhodovacie regióny, je pomocou nadroviny, ktorú definujeme ako:

$$\sum_{i=1}^{m} w_i x_i + b = 0 \tag{12}$$



Obrázok 8 Rozdelenie rozhodovacieho priestoru pre logické funkcie AND, OR, NOT a XOR

V prípade, že máme dve vstupné hodnoty, a teda máme dvojrozmerný rozhodovací priestor, je hranica klasifikácie priamka, ktorej rovnica je $w_1x_1 + w_2x_2 + b = 0$. Takto vieme natrénovať perceptrón na jednoduché problémy ako sú logické brány AND, OR alebo NOT, ktorých rozhodovací priestor vieme lineárne oddeliť. Príklady takýchto rozdelení je možné vidieť na obrázku 8. Zároveň môžeme vidieť príklad nelineárne separovateľnej funkcie XOR. Za pomoci Rosenblattovho perceptrónu nevieme v žiadnom prípade zabezpečiť separáciu výsledkov funkcie XOR a vyvstáva nutnosť implementácie zložitejšej architektúry, napríklad pridaním skrytého neurónu.

2.2.5 Proces učenia neurónovej siete

Proces učenia neurónovej siete môžeme rozdeliť na dve cyklicky striedajúce sa fázy, aktivačnú (rozhodovaciu) a adaptačnú (učiacu sa). V prípade využitia učenia s učiteľom, pozostáva trénovacia množina z dvojice vektorov vstup-výstup. V prípade, že je neurónová sieť natrénovaná, ostáva v aktivačnej fáze. Počas tejto fázy je na vstup neurónovej siete privedený vektor vstupnej informácie, ktorý postupuje naprieč celou neurónovou sieťou. Počas prechodu sú synaptické váhy fixované, čo znamená, že zmena je limitovaná na aktivačný potenciál a výstupy neurónov. Na výstupe neurónovej siete je výstupný vektor, ktorý je odozvou neurónovej siete na vstupný vektor. Následne v adaptačnej fáze je vypočítaná chyba alebo odchýlka medzi výstupným vektorom neurónovej siete a požadovaným výstupným vektorom. Táto chyba je posielaná cez neurónovú sieť v opačnom smere, teda výstup-vstup a na základe tejto chyby sú potom vykonávané úpravy synaptických váh, aby došlo k jej minimalizácii. Potom cyklus pokračuje a opäť nastane aktivačná fáza, novozískanú odchýlku pripočítame k predchádzajúcej a cyklus pokračuje. Prechodom celej trénovacej množiny, teda po dokončení jednej epochy, je získaný súčet odchýlok tiež nazývaný aj globálna odchýlka. V prípade, že získaná odchýlka je menšia ako nami požadovaná chyba je proces učenia neurónovej siete ukončený [37][39]. V dnešnej dobe najpoužívanejší algoritmus pre učenie neurónovej siete je, takzvaný Algoritmus spätnej

propagácie (z angl. Back-Propagation algorithm), ktorý môžeme stručne popísať v troch krokoch:

- Prechod vstupného signálu neurónovou sieťou dopredu (dopredná propagácia) a výpočet odchýlky výstupu siete y_k, od požadovaného výstupu z_k, označme $L(y_k, z_k)$.
- Spätná propagácia chyby aby sme získali jej gradient s ohľadom na každú synaptickú váhu zvlášť (viď. parciálna derivácia vo vzorci (13)).
- Využitie získaného gradientu k aktualizácii synaptických váh neurónovej siete, na základe vzorca:

$$\Delta w_{km} = -\eta \frac{\partial L(y_k, z_k)}{\partial w_{km}} \tag{13}$$

kde Δw_{km} predstavuje zmenu váh a η je parameter učenia. Táto aktualizácia môže byť vykonávaná, s ohľadnom na vstupnú množinu, jeden krát za epochu, po každej vzorke alebo, v prípade rozdelenia vstupnej množiny na dávky, po dávkach.

2.3 Konvolučné neurónové siete

Konvolučné neurónové siete, alebo tiež známe aj ako konvolučné siete [33], sú špecializovaný typ neurónovej siete pre spracovávanie dát, ktoré majú známu mriežkovitú topológiu, napríklad časovú postupnosť dát môžeme chápať ako jednorozmernú mriežku, kedy berieme vzorky v pravidelných časových intervaloch alebo obrazové dáta, ktoré si môžeme predstaviť ako dvojrozmernú mriežku pixelov. Jedna z prvých sietí, ktoré používali konvolúciu, bola navrhnutá v roku 1989 [33] a rozpoznávala ručne písané číslice PSČ, ale jednou z prvých konvolučných sietí, o ktorej sa písalo ako o konvolučnej sieti je LeNet5 [40] z roku 1998. Do popredia sa dostali až po roku 2012 po už spomínanej výhre AlexNet. Rovnako ako iné neurónové siete sa konvolučné siete skladajú z neurónov, ktoré majú svoje trénovateľné váhy a bias a typicky svoj výstup aktivujú na základe nelineárnej funkcie [41]. Ako názov konvolučná sieť napovedá, je v tejto neurónovej sieti implementovaná matematická operácia konvolúcie. V jednoduchosti je možne napísať, že konvolučné siete sú jednoducho neurónové siete, ktoré používajú konvolúciu namiesto skalárneho súčinu, aspoň v jednej zo svojich vrstiev [42]. V prípade tradičnej neurónovej siete je každý neurón v prvej skrytej vrstve prepojený na každú hodnotu vstupu. Tento prístup ale spôsobuje prudký nárast parametrov v prípade vysokodimenzionálneho vstupu, ak máme napríklad vstup obrázok s rozmermi 100x100x3 (100 pixelov široký, 100 pixelov vysoký a 3 farebné

kanály), každý neurón prvej skrytej vrstvy, by mal 30000 trénovateľných parametrov. Naproti tomu konvolučné siete zavádzajú princíp, kedy je každý neurón prepojený len s malou časťou vstupu (časť susedných položiek). Tento jav označujeme ako lokálna konektivita. Táto sa v konvolučných sieťach používa nielen vo vrstve prepojenej na vstupné dáta, ale aj v skrytých vrstvách a je propagovaná do celej siete. Typ neurónových sietí, ktoré tento princíp využívajú sú potom označované ako lokálne prepojené vrstvy [43]. Táto charakteristika poskytuje konvolučnej sieti dve zaujímavé vlastnosti [44]:

- Vzory, ktoré sa naučí sú invariantné – teda potom čo sa naučí určitý vzor napríklad v pravom dolnom rohu, môže ho konvolučné neurónová sieť rozpoznať kdekoľvek, napríklad v ľavom hornom rohu.
- Môžu sa naučiť priestorovú hierarchiu vzorov – Prvá konvolučná vrstva sa naučí malé lokálne vzory ako napríklad hrany. Druhá vrstva sa bude učiť zložitejšie vzory z prvej vrstvy a tak ďalej. Toto dovoľuje konvolučnej sieti sa efektívne naučiť stále komplexnejšie a abstraktnejšie vizuálne pojmy.

Druhým typickým rysom všeobecnej neurónovej siete je fakt, že každý neurón môže obsahovať unikátne synaptické váhy. Konvolučná sieť používa techniku zdieľania parametrov, čo znamená, že neuróny jednej vrstvy zdieľajú rovnaké hodnoty parametrov. Toto prináša výhodu v pamäťovej náročnosti, keďže počet parametrov, ktoré je nutné uchovať, je výrazne znížený. Zdieľanie parametrov vychádza z predpokladu, že každá vzorka vstupu obsahuje príznaky, ktoré sa v rámci nej opakujú. Príznak, ktorý rozpoznávame, je reprezentovaný práve množinou váh, ktorá je zdieľaná. Zároveň je zachovaná pozícia, kde bol daný príznak rozpoznaný, keďže konvolučné vrstvy majú na výstupe tzn. mapu príznakov. Intuitívne, jedna mapa príznakov rozpoznáva jeden príznak a mapuje ho na pozície vstupu. Je možné rozpoznávať viacero príznakov alebo vzorov v jednej vrstve. Vrstva potom obsahuje niekoľko množín váh a na výstupe je niekoľko máp príznakov, každá pre jeden vzor.

2.3.1 Konvolúcia

Ako už bolo spomínané, konvolučné neurónové siete, využívajú konvolučné vrstvy, teda vrstvy neurónov, ktoré pracujú na základe operácie konvolúcie. Konvolúcia je lineárna matematická operácia definovaná ako $y = x * k$, kde znak „*“ označuje operáciu konvolúcie, v terminológii konvolučných sietí je x vstup do konvolučnej vrstvy, k predstavuje filter alebo jadro (z angl. kernel). Vstup do konvolúcie je typicky viacrozmerné

pole hodnôt, nazývané tiež tenzor. Filter alebo jadro konvolúcie je zvyčajne rovnako dimenzionálne ako je vstup, čiže napríklad v prípade spracovávania obrazu sú to tri dimenzie (šírky, výška a farba). Výsledok konvolúcie *y* sa označuje ako mapa príznakov (z angl. feature map). V kontexte konvolučných neurónových sietí často používame konvolúciu vo viacerých osiach súčasne. Napríklad, ak použijeme dvojrozmerný vstup *X*, použijeme taktiež dvojrozmerný filter *K* a rovnica konvolúcie bude mať nasledovný tvar:

$$Y(i,j) = (X * K)(i,j) = \sum_m \sum_n X(m,n)K(i-m, j-n) \tag{14}$$

Operácia konvolúcie je komutatívna, čo znamená že môžeme rovnako napísať:

$$Y(i,j) = (K * X)(i,j) = \sum_m \sum_n X(i-m, j-n)K(m,n) \tag{15}$$

Zvyčajne je druhý vzorec jednoduchšie implementovať v knižnici strojového učenia, pretože existuje menšia odchýlka v rozsahu platných hodnôt *m* a *n* [42].

Komutatívna vlastnosť konvolúcie vzniká, pretože sme otočili filter vzhľadom na vstup v tom zmysle, že ako sa *m* zvyšuje, index vstupu rastie, ale index filtra klesá. Jediný dôvod, prečo otáčame filter, je, aby sme získali komutatívnu vlastnosť. Avšak z pohľadu konvolučných neurónových sietí zvyčajne túto vlastnosť nepotrebujeme, preto mnohé knižnice neurónových sietí využívajú podobnú funkciu, vzájomnú koreláciu (angl. cross-correlation), ktorá funguje obdobne, ale neotáča filter :

$$Y(i,j) = (K * X)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n) \tag{16}$$

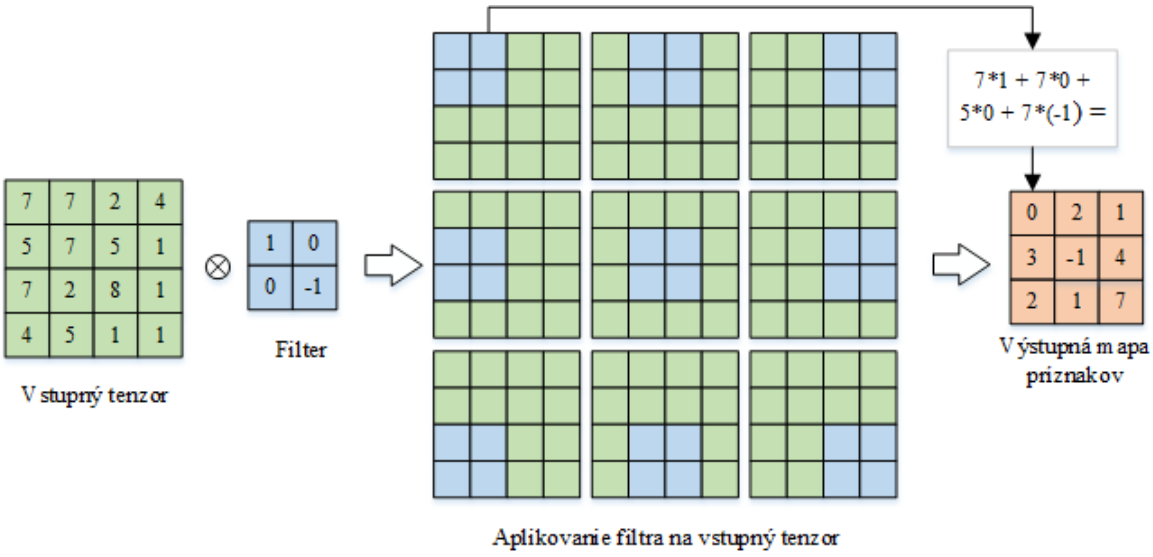
Viaceré knižnice strojového učenia implementujú vzájomnú koreláciu, ale nazývajú to konvolúciou [42].

2.3.2 Konvolučná vrstva

Kľúčovým stavebným blokom konvolučných neurónových sietí sú konvolučné vrstvy. Synaptické váhy neurónov sa v kontexte konvolučných vrstiev nazývajú jadro alebo filter. Pre upresnenie jedna skupina váh sa označuje ako filter a jedna vrstva môže mať viacero takýchto filtrov. Počas prechodu vrstvou je realizovaných niekoľko konvolúcií, podľa počtu filtrov v danej vrstve. Hodnoty tohto filtra predstavujú, spolu s biasom, trénovateľné parametre konvolučnej vrstvy. Z ohľadom na rozmernosť vstupu rozdeľujeme konvolučné vrstvy na:

- 1D konvolučná vrstva – najjednoduchší typ, zvyčajne používaný pre sekvenčné množiny dát,
- 2D konvolučná vrstva – najčastejšie používaný typ v konvolučných sieťach, zvyčajne využívaný pre obrazové dáta,
- 3D konvolučná vrstva – tento typ vrstiev sa využíva pri detekcii udalosti vo videu alebo pri medicínskych 3D obrazoch.

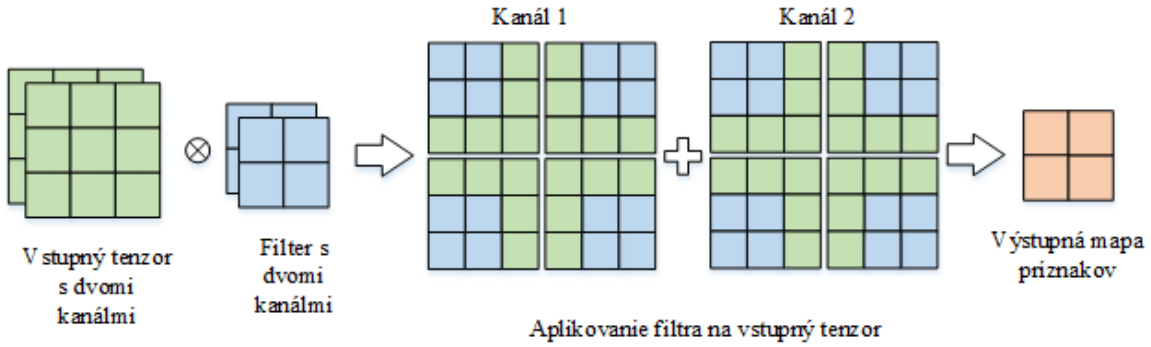
Filter v týchto vrstvách má rovnakú dimenziu ako vstup, ale niekoľkonásobne menšiu veľkosť. Tento filter je posúvaný naprieč vstupným kanálom, zľava doprava a zhora dole. V každej pozícii sú hodnoty filtra vynásobené s aktuálne prekrytými hodnotami vstupu a následne sčítaná do jednej hodnoty. Kolektívny výsledok potom predstavuje mapu príznakov. Ilustráciu tohto procesu môžeme vidieť na obrázku 9, kde je vyobrazená 2-D konvolúcia, avšak tento postup je možné generalizovať pre N-D konvolúciu. Napríklad v prípade 3-D konvolúcie má filter tvar kvádra a posúval by sa postupne po výške, šírke a hĺbke vstupnej mapy príznakov [45].



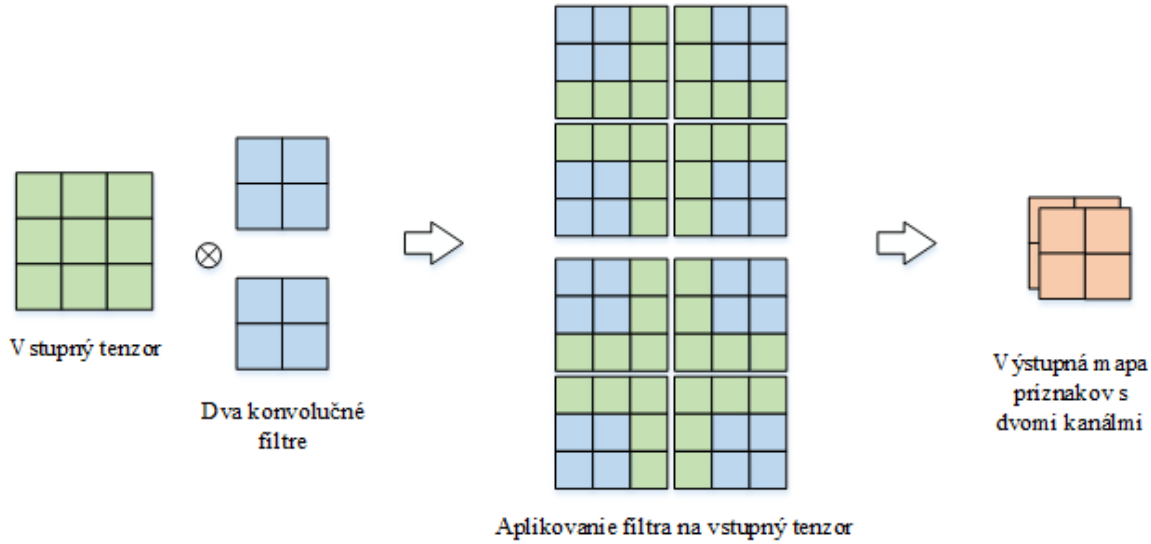
Obrázok 9 Aplikácia dvojrozmernej konvolúcie

Pre zjednodušenie budeme uvažovať v nasledujúcich vysvetleniach o 2D konvolučnej vrstve, čiže vstupné dáta sú dvojdimenzionálne, avšak pridávaná je tretia dimenzia, označovaná ako kanál. Ten môže predstavovať rôzne druhy dát, zvyčajne však v prípade prvej vrstvy predstavuje informáciu o farbe, teda 1 kanál v prípade čiernobieleho vstupu a 3 kanály v prípade farebného (informácia o farbe je v RGB formáte). Pomocou operácie konvolúcie môžeme produkovať viac než jeden kanál vo výstupnej mape príznakov, dá sa

to chápať tak, že aplikovaním konvolúcie mapujeme vstupný priestor príznakov na výstupný priestor príznakov. Obrázky 10 a 11 ilustrujú tento koncept.



Obrázok 10 Operácia konvolúcie pre vstup s dvomi kanálmi a korešpondujúcimi filtrami, tie vynásobia každý kanál samostatne a sú sčítané na konci čím vytvoria výstupnú mapu s jedným kanálom



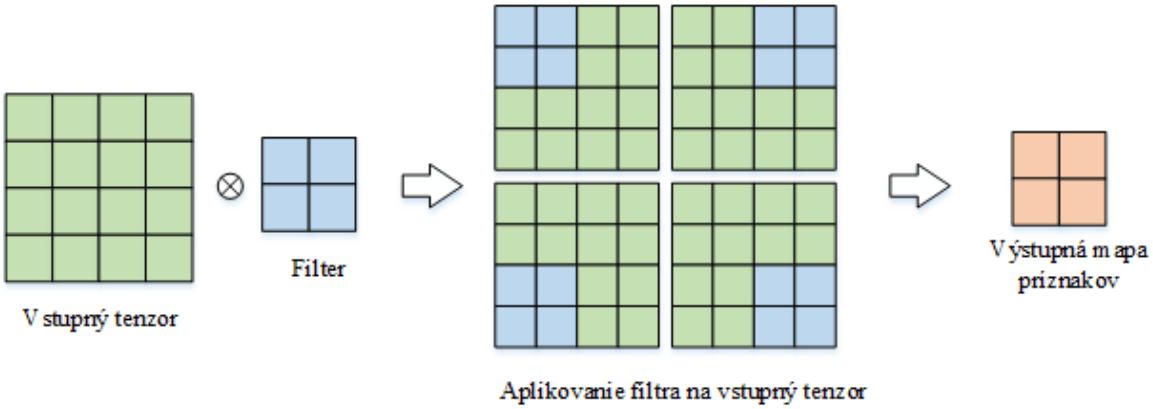
Obrázok 11 Operácia konvolúcie s použitím dvoch konvolučných filtrov. Tie sú aplikované samostatne na vstupný tenzor a ukladané na seba čím vytvoria dvojkanálovú výstupnú mapu

Výstupnú mapu príznakov ovplyvňuje viacero nastavení konvolučnej siete. Tieto sú [34]:

- Veľkosť filtra (angl. kernel size) – pomocou veľkosti filtra môžeme určiť množstvo lokálnych informácii, avšak so zväčšovaním veľkosti filtra sa zmenšuje výstupná mapa príznakov. Zároveň je veľkosť filtra jedným z parametrov, ktoré ovplyvňujú počet trénovateľných parametrov, čiže ak je veľkosť filtra 3x3, potom má 9 trénovateľných parametrov, plus jeden pre bias takže 10 trénovateľných parametrov celkovo. Na obrázkoch 9 až 15 môžeme vidieť, že bol použitý filter veľkosti 2x2, avšak nie je nutné aby bol filter v tvare štvorca, v niektorých prípadoch je vhodnejšie použitie filtra s rozmerom napríklad 1x5. V prípade použitia malej veľkosti filtra sú mapované malé,

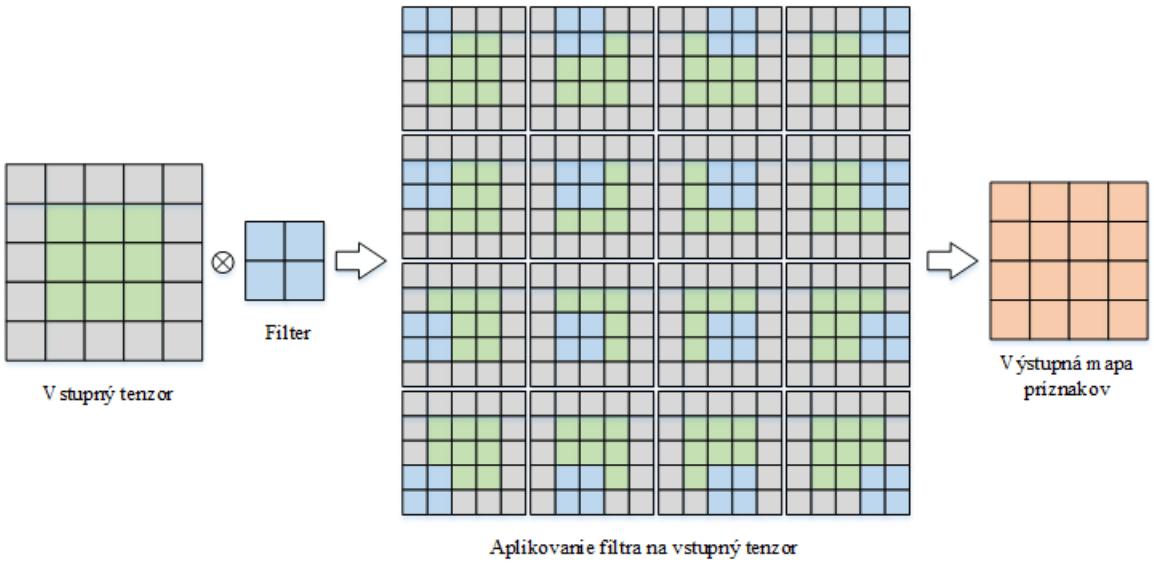
častejšie sa opakujúce vzory a naopak použitie väčšej veľkosti vedie k väčším vzorom, ktoré môžu mať veľkú výpovednú hodnotu, avšak nenastávajú až tak často.

- Krok konvolúcie (angl. stride) – tento parameter ovplyvňuje veľkosť skoku medzi jednotlivými konvolúciami. Je rovnako rozmerný ako vstup a teda v prípade kroku (1,3) je filter posúvaný o tri pozície horizontálne a o jednu vertikálne. Pomocou tejto hodnoty môžeme nastaviť prekrytie jednotlivých výpočtov, teda v prípade, ak je hodnota kroku rovnaká ako veľkosť filtra, sa konvolúcie neprekrývajú a naopak, ak je hodnota kroku 1, tak sú filtre maximálne prekrývané. Výstupnú mapu môžeme úmyselne zmenšovať, aby sme sumarizovali informáciu tým že zvýšime krok, ako je ukázané na obrázku 12.



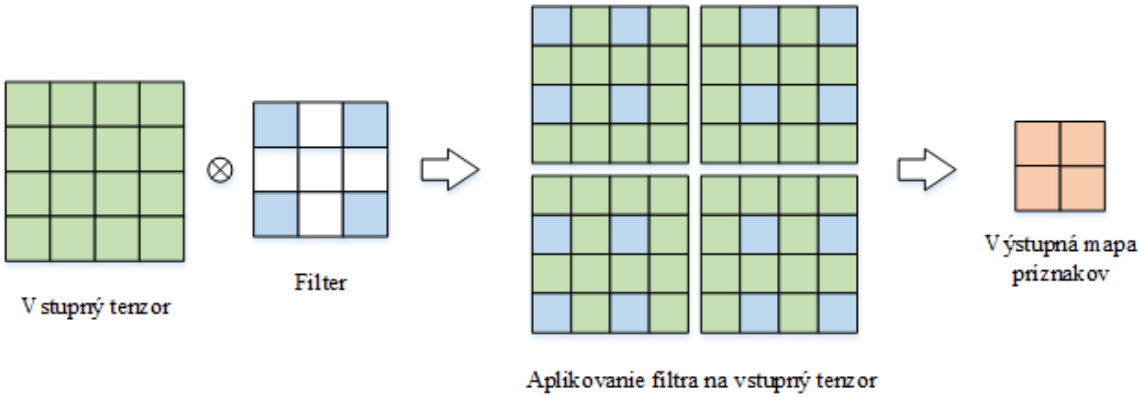
Obrázok 12 Konvolučný filter o veľkosti 2x2 aplikovaný na vstupnú mapu príznakov, s nastavením kroku konvolúcie (2,2), čo má za následok redukciu výstupnej mapy príznakov

- Rozšírenie (angl. padding) – Aj keď veľkosť filtra a krok konvolúcie dovoľujú kontrolovať rozsah každého vypočítaného príznaku, ich vedľajší efekt, ktorý nie je vždy žiadúci, je zmenšovanie výstupnej mapy príznakov. Zároveň to spôsobuje, že okrajové hodnoty sú menej reprezentované, nakoľko vstupujú do výpočtu konvolúcie menej krát. Aby sme tomuto efektu zabránili, môžeme zväčšiť veľkosť vstupnej mapy príznakov v každom smere, s ohľadom na jej dimenzie, pridaním nulových hodnôt. To má za následok vyšší počet konvolučných operácií, avšak veľkosť výstupnej mapy príznakov je kontrolovaná bez zmeny veľkosti, či kroku filtra. Nevýhodou je, že doplnenie hodnôt na okraje môže spôsobiť detekciu neexistujúcich vlastností v tejto lokalite, napriek tomu je toto nastavenie často používané. Príklad využitia tohto nastavenia je na obrázku 13. Ako môžeme vidieť, veľkosť výstupnej mapy je väčšia ako vstupnej. Typicky sa rozšírenie o jednu hodnotu požíva v kombinácii s veľkosťou filtra 3x3, kedy výstupná mapa príznakov je rovnakých rozmerov ako vstupná.



Obrázok 13 Operácia konvolúcie, kedy bola vstupná mapa príznakov rozšítená o jednu hodnotu

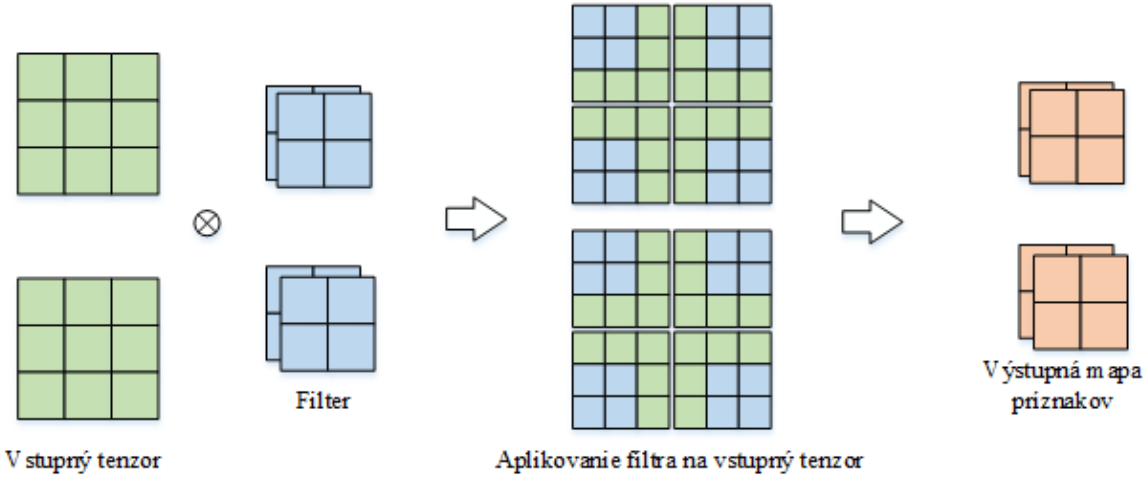
- Roztiahnutie (angl. Dilation) – Pomocou nastavenia roztiahnutia filtra nastavujeme, akým spôsobom bude aplikovaná konvolúcia na vstupnú mapu príznakov. Ako môžeme vidieť na obrázku 14, zvýšenie roztiahnutia zo základnej hodnoty 1 na 2 znamená, že elementy filtra sú aplikované na vstupný tenzor s rozstupom dvoch hodnôt. Toto môže byť užitočné v prípade, že chceme sumarizovať väčšie oblasti vstupnej mapy príznakov bez zvýšenia počtu parametrov. Tento druh konvolúcie sa ukázal ako užitočný v prípadoch, keď sú viaceré konvolučné vrstvy za sebou. Viacero po sebe idúcich roztiahnutých konvolúcií, exponenciálne zväčší veľkosť poľa vnímania, teda veľkosť vstupného priestoru, ktorý neurónová sieť vidí predtým než vykoná predikciu.



Obrázok 14 Operácia konvolúcie, kedy bola hodnota roztiahnutia filtra nastavená na 2

- Skupiny (angl. Groups) – Skupiny sú veľmi užitočné v špecifických prípadoch. Napríklad, ak máme niekoľko zreťazených zdrojov údajov. Keď nie je potrebné riešiť

ich závislosť jeden na druhom. Vstupné kanály je možné zoskupovať a spracovávať nezávisle. Nakoniec sú výstupné kanály zreťazené. Ako je možné vidieť na obrázku 15, ak máme dvojkanálovú vstupnú mapu a chceme získať štvorkanálovú výstupnú mapu, môžeme vstupnú mapu príznakov rozdeliť do dvoch skupín (jeden kanál v každej skupine), prechodom cez konvolučnú vrstvu dostaneme výstupnú mapu s polovicou požadovaných kanálov. Tieto potom zlúčime pozdĺž kanálovej osi. Je dôležité poznamenať, že počet vstupných aj výstupných kanálov musí byť deliteľný počtom skupín.



Obrázok 15 Operácia konvolúcie s rozdelením 2 kanálového vstupu a 4 kanálového výstupu do dvoch skupín

S ohľadom na tieto nastavenia môžeme veľkosť výstupnej mapy príznakov, pre každú jej dimenziu, vypočítať ako:

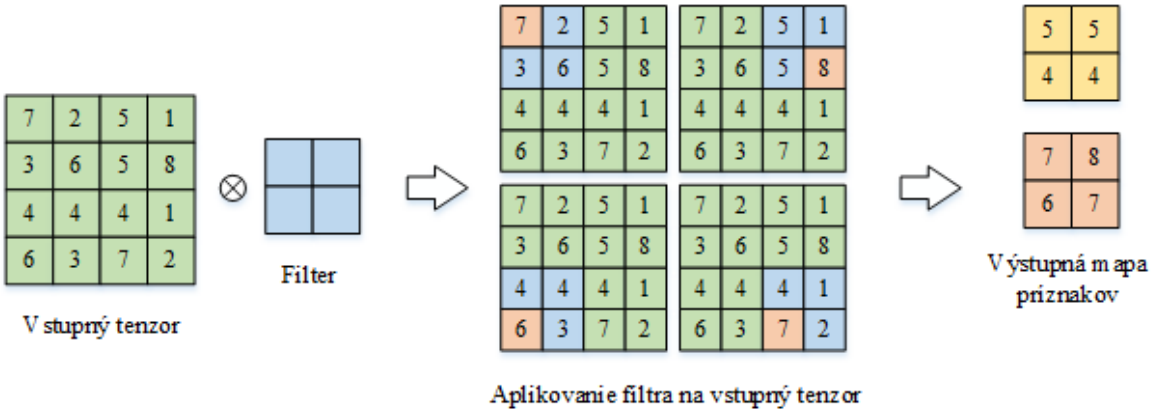
$$Y = \frac{X + 2P - D(K-1) - 1}{S} \tag{17}$$

kde Y je veľkosť výstupnej mapy príznakov, X je veľkosť vstupnej mapy príznakov, P je rozšírenie vstupnej mapy, D je roztiahnutie konvolučného filtra, K je veľkosť filtra a S je krok konvolúcie.

2.3.3 Podvzorkovanie

Okrem konvolučnej vrstvy ďalší prvok konvolučných neurónových sietí, ktorý ich robí efektívne, je postupné podvzorkovanie dát počas prechodu konvolučnými vrstvami. To núti model, aby sa naučil väčšie (s ohľadom na pôvodný vstupný priestor) a viac komplexnejšie príznaky (vzory vzorov) v neskorších vrstvách. Toto podvzorkovanie je možné vykonať tak, že použijeme konvolučnú vrstvu a nastavíme veľkosť filtra a veľkosť kroku konvolúcie na rovnakú hodnotu [47]. Ako to už bolo demonštrované na obrázku 12.

Druhým spôsobom, ktorý sa využíva, je aplikácie tzv. vrstvy združovania (z angl. pooling layer). Tento druh vrstvy bol použitý už v LeNet5. Vrstva združovania má podobné nastavenia ako konvolučná vrstva, teda nastavenia veľkosti filtra a kroku, ktorý je vo väčšine prípadov volený tak, aby sa oblasti aplikácie neprekrývali. Každý kanál vstupnej mapy príznakov je spracovávaný samostatne. Z každej oblasti aplikácie je potom vybratá jedna hodnota, v závislosti od typu vrstvy, buď je to priemerná hodnota alebo častejšie využívaná maximálna hodnota. Príklad tejto operácie môžeme vidieť na obrázku 16, kedy sme aplikovali združovanie podľa maximálnej hodnoty s veľkosťou filtra 2x2.



Obrázok 16 Aplikácia vrstvy združovania podľa maximálnej hodnoty v červenej farbe. Pre ilustráciu sme pridali aj združovanie podľa priemernej hodnoty, ktorá bola zaokrúhlená v žltej farbe

3 Extrakcia príznakov

V prvej kapitole bol zvuk popísaný s ohľadom na jeho informačnú hodnotu. Z fyzikálneho hľadiska môžeme zvuk, resp. akustický signál definovať, ako usporiadaný kmitavý pohyb častíc prostredia, v ktorom sa zvuk šíri. Kmitanie častíc zdroja zvuku sa pomocou vzájomného pôsobenia prenáša na častice v okolí, ktoré sa tiež rozkmitajú, nedochádza však k presunu hmoty, len k presunu energie. Keďže dochádza pri prenose k určitému oneskoreniu, vzniká postupná vlna, ktorá sa šíri smerom od zdroja zvuku. Celý proces je v podstate mechanickým kmitaním pružného prostredia. Na základe frekvencie kmitania sa potom zvuky delia do troch pásiem: infrazvuk - pásmo 0,7 – 16 Hz, sú to zvuky pod hranicou počuteľnosti; počuteľné pásmo - pásmo 16 – 20 000 Hz, toto pásmo predstavuje zvuky, ktoré sú schopné vyvolať zvukový vnem; ultrazvuk - pásmo 20 – 50 kHz, to sú zvuky nad hranicou počuteľnosti. Skutočný rozsah počuteľného zvuku je subjektívny, avšak najhlasnejšie sú vnímané signály v oblasti 500 – 5 000 Hz [48], čiže v tejto oblasti je ľudské ucho najcitlivejšie, zároveň s narastajúcim vekom sa častokrát stráca citlivosť vo vyšších frekvenciách.

Akustický signál môžeme charakterizovať nasledujúcimi vlastnosťami:

- Akustický tlak a hladina akustického tlaku – pri postupe akustickej vlny dochádza k zhlukovaniu väčšieho množstva kmitajúcich bodov na jednom mieste a zároveň k rednutiu množstva na inom mieste. Vznikajú tak tlakové vlny, ktoré vyvolávajú zvukový vnem. Najslabší zvuk, ktorý môže sluch človeka zaznamenať, je charakterizovaný akustickým tlakom 20 μPa, táto hodnota je označovaná ako prah počutia. Keďže sluch človeka je schopný zachytiť akustické tlaky viac než miliónkrát väčšie, bol zavedený logaritmus týchto hodnôt, ktorý sa označuje ako hladina akustického tlaku, ktorý prevádza rozsah akustického tlaku 20 – 1 000 000 000 μPa na rozsah 0 – 140 dB. Jednotka hladiny akustického tlaku je decibel [dB], a teda 1 dB potom predstavuje vyjadrenie najmenšej zmeny, ktorú je človek schopný zaznamenať.
- Frekvencia – Akustické vlnenie sa šíri od zdroja vo vlnoplochách. Všetky body vlnoplochy majú v daný časový okamžik rovnakú fázu. Smer šírenia v danom bode určuje kolmica na vlnoplochu, ktorú nazývame akustický lúč, ktorý má tú vlastnosť, že sa môže odrážať, prípadne lomiť na hranici dvoch prostredí. Vlnová dĺžka λ [m] je potom najmenšia vzdialenosť medzi dvoma susednými bodmi akustického lúča, ktoré majú rovnakú fázu, čo znamená vzdialenosť, ktorú zvuková vlna prejde za dobu

jedného kmitu T [s]. Frekvencia f [Hz], predstavuje počet kmitov za sekundu, ktoré vykoná kmitajúci bod v prostredí, v ktorom sa šíri zvuková vlna [49].

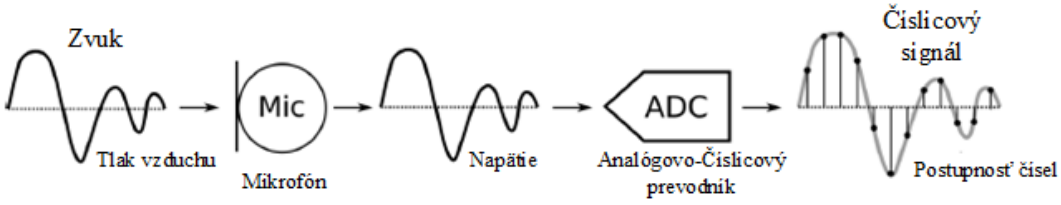
- Akustický výkon – Predstavuje energiu zvukových vĺn vyprodukovaných zdrojom za jednotku času. Rozsah akustického výkonu je značný, napríklad akustický výkon šepotu je $1*10^{-9}$ W, výkrik $1*10^{-3}$ W a prúdové lietadlo $1*10^{5}$ W.
- Rýchlosť šírenia akustických vĺn - Rýchlosť zvuku je závislá od prostredia, v ktorom sa zvuk šíri. Zároveň je ovplyvnená fyzikálnym stavom tohto prostredia, ako je jeho teplota, tlak a tak ďalej. Napríklad rýchlosť šírenia zvuku vo vzduchu je 340 m/s, vo vode dosahuje rýchlosť 1480 m/s.

Keďže zvuk, ktorý môžeme premeniť z akustickej podoby na elektronický signál pomocou mikrofónu, ako mnoho iných signálov v prírode je spojitý v čase i úrovni a technické prostriedky pracujú s diskrétnymi hodnotami v diskrétnom čase, je nutné využiť procesu diskretizácie. Tento proces sa skladá z dvoch krokov [50][51]:

- Diskretizácia v čase – vzorkovanie – proces vzorkovania transformuje signál na časovo diskrétny signál. Prostredníctvom vzorkovania sa získavajú hodnoty časovo spojitého signálu v presne definovaných časových okamihoch. V prípade periodického vzorkovania je spojitý signál $x(t)$ nahradený postupnosťou vzoriek $x(nT)$, kde T predstavuje periódu vzorkovania.
- Diskretizácia v úrovni – kvantovanie – proces kvantovania transformuje signál spojitý v úrovni na signál diskrétny v úrovni. Každá hodnota signálu je nahradená hodnotou vybranou z konečnej množiny prípustných hodnôt. Počas tohto procesu vzniká tzv. kvantizačná chyba, čo je rozdiel medzi skutočnou a priradenou hodnotou.

Po procese vzorkovania a kvantovania dostávame signál diskrétny v čase i úrovni. Potom nasleduje proces kódovania, kde je každej diskrétnej hodnote číslicového signálu priradený istý kód, najčastejšie b-bitová binárna postupnosť.

Na obrázku 17 je možné vidieť zjednodušenú ilustráciu prevodu zvuku na číslicový signál.



Obrázok 17 Prevod zvuku na číslicový signál

3.1 Fourierová transformácia

Fourierová transformácia je jedným zo základných pilierov spracovávania signálov. S využitím tejto transformácie môžeme rozložiť signál s ohľadom na jeho harmonické (sínusové, resp. komplexne exponenciálne) komponenty, efektívne tak prevádza signál z časovej oblasti do frekvenčnej oblasti. Predstavuje tak efektívny nástroj na frekvenčnú analýzu v oblasti spracovania signálov. Pomocou tejto transformácie vieme spracovávať signál, ktorý je aperiodický a môže byť spojitý alebo diskrétny v čase [51].

Pre výpočet Fourierovej transformácie bolo navrhnutých viacero algoritmov, napríklad diskrétna Fourierová transformácia (DFT), rýchla Fourierová transformácia (FFT), krátkodobá Fourierová transformácia (STFT) a ďalšie.

3.1.1 Diskrétna Fourierová transformácia

Keďže zvukový signál, prevedený do číslicovej podoby, pozostáva zo sekvencie diskrétnych vzoriek, na prevod do frekvenčnej oblasti môže byť použitá diskrétna Fourierová transformácia. Vo všeobecnosti je vstupom do transformácie vektor N komplexných čísel $x_k,\ k = 0 \dots N-1$, ktorý je transformovaný na vektor N komplexných čísel $y_k,\ k = 0 \dots N-1$. Na základe normalizácie potom rozlišujeme tri bežné definície DFT [52]:

$$y_m^{(1)} = \sum_{k=0}^{N-1} x_k e^{-2\pi i \frac{mk}{N}} \tag{18}$$

$$y_m^{(2)} = \frac{1}{\sqrt{N}} y_m^{(1)} \tag{19}$$

$$y_m^{(3)} = \frac{1}{N} y_m^{(1)} \tag{20}$$

kde $m = 0, 1, 2 \dots N-1$. Tato transformácia má aj inverznú podobu, ktorá všeobecne používa definíciu (20), ale líši sa v znamienku exponentu ($+2\pi i \frac{mk}{N}$ namiesto $-2\pi i \frac{mk}{N}$). Z praktického hľadiska implementácie DFT v počítačoch sa takmer vždy používa algoritmus rýchlej Fourierovej transformácie ako napríklad FFTW [53] a používa definíciu (18). Mathematica používa definíciu (19), ktorá je ako jediný symetrický spôsob ako definovať DFT a inverznú DFT, čo znamená, že ako aplikujeme DFT a následne inverznú DFT, reprodukujeme pôvodné dáta. Ak zvýšime N, (18) musí spracovať čoraz viacej signálu, v dôsledku čoho výsledok v adekvátnom frekvenčnom kroku je výšený proporčne k N,

zatiaľ čo požadovaný výsledok, amplitúda, by od N závisieť nemala. Čiže aj keď N vstupných vzoriek obsahuje viacej informácií, tieto sú rovnakým faktorom rozriedené, teda aj keď dlhšia DFT ponúka lepšie frekvenčné rozlíšenie, je rovnako zašumené ako krátka. Odpoveď na tento problém je využitie tzv. váhových funkcií. Algoritmus DFT potom spracováva signál, ako postupnosť parciálnych signálov, resp. segmentov vstupného signálu, z fixnou veľkosťou N, pričom delenie vstupného signálu na tieto segmenty je vykonávané pomocou váhových funkcií. Výsledné frekvenčné spektrá sú potom spriemerované, aby vytvorili $N/2 + 1$ bodové frekvenčné spektrum [54]. Výberom adekvátneho tvaru váhovej funkcie je potom možné ovplyvniť výsledné frekvenčné spektrum. Váhové funkcie sú rozobraté v sekcii 3.1.3.

Aj keď x_k môže vo všeobecnej definícii DFT nadobúdať komplexných hodnôt, časová postupnosť digitalizovaného vstupného signálu je vždy reálna. Z toho vyplýva, že pre výstupnú postupnosť y_k platí:

$$y_{N-m} = y_m^* \tag{21}$$

kde * značí komplexne združené číslo. Zvlášť ak je N párne číslo, y_0 a $y_{N/2}$ sú reálne čísla. Ak teda budeme uvažovať, že N je párne číslo, dolná polovica výslednej postupnosti $y_0 \dots y_{N/2}$ korešponduje s reálnou častou DFT spektra, horná polovica obsahuje negatívne hodnoty frekvencie a býva väčšinou ignorovaná.

3.1.2 Rýchla Fourierová transformácia

Výpočet diskrétnej Fourierovej transformácie je časovo náročné. Na výpočet každého bodu DFT pozostáva z N operácií komplexného násobenia a $N-1$ operácií komplexného sčítania. Na výpočet všetkých N bodov DFT je potom potrebné vykonať N^2 komplexných násobení a $N(N-1)$ komplexných sčítaní [50]. Preto, ako už bolo spomínané, sa v praktickej implementácií takmer vždy používa algoritmus FFT, ktorý bol odvodený z komplexnej DFT. Pomocou tohto algoritmu je možné zmenšiť počet komplexných násobení znížený na $N \log_2 N$ operácií. Existencia FFT algoritmu sa stala všeobecne známa v 1965 vďaka práci J.W. Cooley a J.W. Tukey, ktorým je dávaný kredit za odvodenie FFT. Retrospektívne je však možné zistiť, že efektívne metódy počítania DFT boli nezávisle objavené a v niektorých prípadoch implementované viacerými jednotlivcami, začínajúc K.F. Gauss v roku 1805. Jedno „znovuobjavenie“ FFT, to ktoré odvodili G.C. Danielson a C. Lanczos v roku 1942, poskytuje jedno z najjasnejších odvodení algoritmu. Danielson a Lanczos ukázali, že diskrétna Fourierová transformácia s dĺžkou N môže byť

prepísná ako súčet dvoch diskrétnych Fourierových transformácií, každej s dĺžkou $N/2$. Jedna z týchto dvoch je tvorená hodnotami s párnym indexom z pôvodných N a druhá je tvorená nepárnym [55]. Ak definujeme komplexné číslo

$$W = e^{-\frac{2\pi i}{N}} \tag{22}$$

môžeme (18) prepísať ako:

$$y_m = \sum_{k=0}^{N-1} W^{mk} x_k \tag{23}$$

potom môžeme algoritmus FFT podľa Danielson a Lanczos vyjadriť podľa vzorca

$$y_m = \sum_{k=0}^{\frac{N}{2}-1} W^{2mk} x_{2k} + W^k \sum_{k=0}^{\frac{N}{2}-1} W^{2mk} x_{2k+1} \tag{24}$$

$$y_m = y_m^p + W^k y_m^n \tag{25}$$

kde $m = 0, 1, 2, \dots N-1$. V (25) y_m^p označuje m-tý komponent Fourierovej transformácie dĺžky $N/2$ tvorený vzorkami s párnym indexom, y_m^n potom korešponduje s tými nepárnymi. Výhoda tohto algoritmu je, že je možné ho použiť rekurzívne, teda ak kalkulácia y_m môže byť rozdelená na kalkulácie y_m^p a y_m^n, potom aj tie môžu byť ďalej rozdelené. Teda y_m^p môže byť rozdelená na transformácie dĺžky $N/4$ ich parnými a nepárnymi indexami y_m^{pp} a y_m^{pn}. Takto môžeme pokračovať, až kým nedostaneme transformáciu s dĺžkou 1. Obmedzením je, že počet pôvodných vzoriek N musí byť párny a vo väčšine publikácií je odporúčané, aby sa $N = 2^n, n > 0$. V prípade, že dĺžka vstupných dát nespĺňa túto podmienku, sa odporúča doplniť vstupné dáta o nulové hodnoty tak, aby dĺžka bola mocninou dvojky [55][56]. V mnohých situáciách je to však zbytočne prísne obmedzenie. Balík FFTW [53] napríklad počíta FFT pre kladné celé čísla N, ktoré sú vo forme

$$N = 2^a 3^b 5^c 7^d 11^e 13^f \tag{26}$$

kde $e + f$ sa rovná buď 0 alebo 1, ostatné exponenty sú ľubovoľné. Toto vyjadrenie ponúka väčšiu voľnosť pri určení frekvenčného rozlíšenia.

Ak budeme uvažovať, že vzorkovacia frekvencia f_s je nemenná, z Nyquistovho teorému vyplýva, že maximálna užitočná frekvencia je $f_{Ny} = f_s/2$. Potom z hľadiska spracovania

dát zmena dĺžky dát N vstupujúcich do FFT je jediný spôsob ako ovplyvniť frekvenčné rozlíšenie Δf, resp. frekvenčný krok, ktorý môžeme vyjadriť ako

$$\Delta f = \frac{f_s}{N} \tag{27}$$

3.1.3 Váhové funkcie

Algoritmus FFT predpokladá, že vstupný signál je periodický, teda že časová postupnosť s dĺžkou N sa cyklicky opakuje donekonečna. Ak frekvencia sínusového vstupného signálu nie je násobkom frekvenčného rozlíšenia f_r, tento predpoklad nie je pravdivý a FFT zaznamená diskontinuitu medzi poslednou a prvou vzorkou z dôvodu cyklického opakovania. Tieto umelé diskontinuity sa potom prejavia vo FFT ako vysokofrekvenčné komponenty, ktoré neboli prítomné v pôvodnom signáli. Výsledné spektrum teda nebude spektrum pôvodného signálu, ale jeho rozmazaná verzia, teda akoby energia jednej frekvencie presakovala to ostatných frekvencií. Tento fenomén je nazývaný presakovanie spektra (z angl. spectral leakage) [57].

Tento efekt je možné minimalizovať použitím váhovej funkcie, resp. váhového okna. Teda časová postupnosť je vynásobená váhovou funkciou pred aplikáciou FFT. Všetky váhové funkcie sa zhodujú v troch vlastnostiach:

- mimo oblasti ich definície nadobúdajú nulovú hodnotu,
- sú symetrické a na hranici symetrie, teda v strede nadobúdajú maximum,
- na začiatku aj na konci, nadobúdajú hodnoty blízke alebo rovné nule.

Na základe týchto vlastností je diskontinuita odstránená. Bolo definovaných niekoľko váhových funkcií, ktorých tvar väčšinou odráža kompromis medzi šírkou výsledného vrcholu vo frekvenčnej oblasti, presnosťou amplitúdy a pomerom zníženia presakovania spektra [52]. Inými slovami, široký hlavný lalok, zapríčiňujúci nepriaznivé frekvenčné rozlíšenie, je spojený s malou amplitúdou postranných lalokov, pri ktorých sa presakovanie znižuje, a naopak úzky hlavný lalok, umožňujúci presnejšie odčítanie frekvencie signálu, je spojený s väčšou amplitúdou postranných lalokov, keď je presakovanie spektra väčšie [58].

Vybrané tvary váhových funkcií a ich charakteristiky [52][59]:

- Obdĺžnikové okno – patrí medzi najjednoduchšie a vyjadruje všeobecnú váhovú funkciu, ktorá je určená rovnicou

$$w_m = 1 \tag{28}$$

a je ekvivalentom nepoužitia váhovej funkcie. Pre transformáciu je vhodná len v prípade, ak je zaručená rovnaká dĺžka segmentu ako je celočíselný násobok periód analyzovaných frekvencií [60]. Vlastnosti obdĺžnikového okna:

- prvá nula je lokalizovaná v $\pm 1{,}00\ \Delta f$,
- zvýšením N sa zužuje hlavný lalok, čiže sa zlepšuje frekvenčné rozlíšenie,
- najvyšší bočný lalok je -13,3 dB voči hlavnému laloku, lokalizovaný v $\pm 1{,}43\ \Delta f$,
- pokles bočných lalokov je -6 dB na oktávu,
- ekvivalentná šírka šumu je $1{,}00\ \Delta f$,
- maximálna amplitúdová chyba $e_{max} = -3{,}9224\ dB = -36{,}3380\%$.

- Hanningová váhová funkcia – toto váhové okno má primerane nízke presakovanie spektra a šírku pásma, preto býva používané ako štandardné váhové okno mnohých komerčných spektrálnych analyzátorov, ak nie je nutná amplitúdová presnosť pre sínusové signály, napríklad v meraní hluku. Je definovaná ako:

$$z = \frac{2\pi \cdot m}{N}, \qquad m = 0 \ldots N - 1 \tag{29}$$

$$w_m = \frac{1 - \cos(z)}{2} = \cos^2\left(\frac{z - \pi}{2}\right) \tag{30}$$

Optimálne prekrytie medzi jednotlivými oknami je 50%. Vlastnosti Hanningovho okna:

- prvá nula je lokalizovaná v $\pm 2{,}00\ \Delta f$,
- najvyšší bočný lalok je -31,5 dB voči hlavnému laloku a nachádza sa v $\pm 2{,}36\ \Delta f$,
- pokles bočných lalokov je -18 dB na oktávu,
- ekvivalentná šírka šumu je $1{,}50\ \Delta f$,
- maximálna amplitúdová chyba $e_{max} = -1{,}4236\ dB = -15{,}1174\%$.

- Blackman-Harrisová váhová funkcia – je jednou z rodiny váhových funkcií, tvorených súčtom kosínusov. Zmenou počtu a hodnôt koeficientov je možné optimalizovať rozličné charakteristiky. Typickým predstaviteľom je funkcia nazývaná „ minimálne štvorčlenné Blackman-Harrisovo okno“ a bolo navrhnuté, aby malo malé bočné laloky pri hlavnom laloku v prechodovej funkcii. Táto váhová funkcia má veľmi nízke presakovania spektra v kombinácií s primeranou chybou amplitúdy a šírky pásma. Vďaka malým bočným lalokom je vhodná pre detekciu malých sínusových signálov,

ktoré sú vo frekvencii blízko veľkým signálom. Taktiež je vhodné, ako všeobecné okno pre aplikácie s veľkou dynamickou škálou, v prípade, že amplitúdová presnosť pre sínusové signály nie je príliš dôležitá. Táto váhová funkcia je definovaná ako:

$$z = \frac{2\pi \cdot m}{N}, \qquad m = 0 \dots N-1 \tag{31}$$

$$w_m = 0{,}35875 - 0{,}48829\cos(z) + 0{,}14128\cos(2z) - 0{,}01168\cos(3z) \tag{32}$$

Optimálne prekrytie medzi váhovými oknami je 66,1%. Vlastnosti Blackman-Harrisovho okna:

- prvá nula je lokalizovaná na $\pm 4{,}00\ \Delta f$,
- najvyšší bočný lalok je -92,0 dB a nachádza sa v $\pm 4{,}52\ \Delta f$, preto sa niekedy toto okno označuje ako „92 dB Blackman-Harrisovo okno“,
- pokles bočných lalokov je -6 dB na oktávu,
- ekvivalentná šírka šumu $2{,}0044\ \Delta f$,
- maximálna amplitúdová chyba $e_{max} = -0{,}8256\ dB = -9{,}067\ \%$.

3.1.4 Krátkodobá Fourierová transformácia

S využitím krátkodobej Fourierovej transformácie môžeme sledovať spektrálne zmeny v čase. Princípom STFT, ako už bolo popísané, je segmentácia signálu do kratších úsekov pomocou váhovej funkcie a následne je na každý úsek aplikovaná FFT. Použité váhové okno a jeho veľkosť ovplyvňujú výsledné zobrazenie. STFT môžeme definovať ako [61]:

$$X(m,k) = \sum_{n=0}^{N-1} x(hm+n)w(n)e^{-2\pi i\frac{mn}{N}} \tag{33}$$

kde $m = 0, 1, 2, \dots N-1$, N označuje dĺžku váhového okna, k definuje poradové číslo aktuálneho segmentu a h je jeho posun. Dĺžka okna predstavuje kompromis medzi časovým a frekvenčným rozlíšením, nakoľko dĺžka okna ovplyvňuje frekvenčné rozlíšenie priamo úmerne (27) a časové rozlíšenia nepriamo úmerne $\Delta t = N$. Výsledkom STFT sú komplexné čísla vyjadrujúce informáciu o fáze a amplitúde každého frekvenčného kroku. Amplitúdové spektrum potom môžeme vyjadriť ako absolútnu hodnotu výsledku (34), čím odstránime informáciu o fáze. Výkonové spektrum potom ako jeho druhú mocninu (35).

$$A(m) = |X(m)| \tag{34}$$

$$P(m) = |X(m)|^2 \tag{35}$$

3.1.5 Kĺzavá diskrétna Fourierová transformácia

Kĺzavá diskrétna Fourierová transformácia, SDFT (z angl. Sliding Discrete Fourier Transform) [79] vychádza z predpokladu, že po dobu dvoch po sebe idúcich časových úsekov, označme n a $n-1$,obsahujú oknové sekvencie $x(n-1)$ a $x(n)$ takmer rovnaké prvky. Princípom SDFT [80] je vykonávanie *N*-bodovej DFT na časových vzorkách v posuvnom okne, teda najskôr je vypočítaná DFT z *N* časových vzoriek, následne je časové okno posunuté o jednu vzorku a je vypočítaná nová *N*-bodová DFT, teda každá nová DFT je vypočítaná priamo z výsledkov predchádzajúcej DFT. Využíva sa tu skutočnosť, že DFT oknovej sekvencie s konečnou dĺžkou $X(k)$ priamo závisí od DFT tejto sekvencie kruhovo posunutej o jednu vzorku $X(k) \cdot e^{\frac{i2\pi k}{N}}$, teda spektrálne komponenty v čase posunutej postupnosti sú tvorené pôvodnými neposunutými spektrálnymi komponentami, ktoré sú násobené $e^{\frac{i2\pi k}{N}}$, kde k je index frekvenčnej domény záujmu [81]. Tento proces je možné vyjadriť pomocou rovnice:

$$X_k(n) = [X_k(n-1) - x(n-N) + x(n)] \cdot e^{\frac{i2\pi k}{N}} \tag{36}$$

kde $X_k(n)$ je nový spektrálny komponent a $X_k(n-1)$ je predchádzajúci spektrálny komponent.

Na výpočet každého ďalšieho spektrálneho komponentu je potrebné použitie konštantného počtu operácií, konkrétne dve operácie reálneho sčítania a jedna operácia komplexného násobenia. Výpočtová zložitosť každého následného *N*-bodového výstupu je v prípade SDFT $O(N)$.

3.1.6 Gaborová transformácia

Prvé časovo-frekvenčné zobrazenie je prisudzované D. Gabor v roku 1946 v jeho článku [82], ktorý sumarizoval svoju motiváciu vo svojom citáte:

„Hithertová teória komunikácie je založená na dvoch alternatívnych metódach analýzy signálu. Jednou je vyjadrenie signálu, ako funkciu času, druhou je Fourierová analýza. Obe sú idealizácie; prvá metóda pracuje s presne definovanými časovými okamihmi a druhá s nekonečnými vlnami prísne definovaných frekvencií. Ale naše každodenné skúsenosti, najmä naše sluchové vnemy však vyžadujú popis v oboch, aj v čase, aj vo frekvencii.

V tomto článku popísal prvé časovo-frekvenčné zobrazenie, ktoré je dnes nazývané Gaborová transformácia. V prípade Gaborovej transformácie je signál vynásobený

Gaussovou váhovou funkciou, následne je na výsledok aplikovaná Fourierová transformácia, čím dostaneme časovo-frekvenčné zobrazenie.

Neskôr bolo navrhnutých viacero spôsobov získavania časovo-frekvenčného zobrazenia. Jednou z nich je napríklad Krátkodobá Fourierová Transformácia, ktorú môžeme chápať ako zovšeobecnenie Gaborovej transformácie, keďže Gaborová transformácia je v podstate STFT s použití Gaussovhej váhovej funkcie.

3.2 Biológiou inšpirované metódy transformácie

Spracovávanie akustického signálu človekom inšpirovalo niekoľko metód transformácie akustického signálu.

3.2.1 Mel spektrogram

Ľudské ucho nevníma frekvencie lineárne naprieč frekvenčným spektrom, je pre neho ťažšie rozpoznávanie medzi vyššími frekvenciami a naopak ľahšie medzi nižšími. Na základe tohto faktu bola navrhnutá melová mierka. Konverzia frekvencie v Hertzoch do melovej mierky je uskutočnená podľa vzorca:

$$f_{mel} = 2595 \log_{10}\left(1 + \frac{f_{Hz}}{700}\right) \tag{37}$$

Pre vytvorenie Mel spektrogramu [62] je signál najskôr transformovaný pomocou STFT do frekvenčnej oblasti, následne je aplikovaná Melová banka filtrov, ktorá pozostáva z trojuholníkových filtrov. Tieto filtre majú maximálnu odozvu, teda 1, v centrálnej frekvencii a následne lineárne klesajú k nule. Keď dosiahnu centrálne frekvencie dvoch susedných filtrov, ich odozva je nula. Počet filtrov určuje výsledné rozlíšenie mel spektrogramu. Melovú banku filtrov môžeme modelovať pomocou systému rovníc:

$$H_m(k) = \begin{cases} 0, & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)}, & f(m-1) \leq k < f(m) \\ 1, & k = f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)}, & f(m) < k \leq f(m+1) \\ 0, & k > f(m+1) \end{cases} \tag{38}$$

kde m predstavuje počet požadovaných filtrov a $f(\cdot)$ je zoznam $m+2$ frekvencii v melovom rozložení.

3.2.2 Gammatonová banka filtrov

Formovanie Gammatonovej banky filtrom bolo inšpirované odozvou membrány slimáka vo vnútornom uchu ľudského sluchového systému. Impulzná odozva $g(t)$ Gammatónového filtra je súčin Gamma rozloženia a sínusového tónu v určitej frekvencii f_c, ktorá je počítaná ako [63]:

$$g(t) = Kt^{n-1}e^{-2\pi Bt}\cos(2\pi f_c t + \varphi), \;\; t > 0 \tag{39}$$

kde K je amplitúdový faktor, n je rád filtra, f_c je centrálna frekvencia v Hertzoch, φ je posun vo fáze a B predstavuje trvanie impulznej odozvy.

3.3 Ďalšie metódy pre analýzu signálu

Medzi populárne metódy pre analýzu signálu patria:

- Diskrétna kosínusová transformácia,
- Hilbert-Huangová transformácia,
- Waveletová transformácia,
- Wigner-Villeová distribúcia, taktiež známa aj ako Heisenbergov wavelet.

Prvé dve si bližšie popíšeme.

3.3.1 Diskrétna kosínusová transformácia

Diskrétna kosínusová transformácia (DCT, z angl. Discrete Cosine Transform) bola navrhnutá v roku 1974 Ahmed et al. v článku [83]. DCT je transformácia príbuzná Fourierovej transformácii, naproti nej však produkuje len reálne koeficienty. Dôležitou vlastnosťou DCT, ktorá ju robí užitočnou pre dátovú kompresiu je, že zoberie korelované vstupné dáta a koncentruje ich energiu v niekoľkých prvých transformačných koeficientoch. Ak vstupné dáta pozostávajú z korelovaných veličín, potom väčšina z N transformačných koeficientov, vyprodukovaných pomocou DCT, sú rovné nule alebo malému číslu a len niekoľko z nich sú veľké čísla (zvyčajne tých prvých). Prvotné koeficienty obsahujú informácie o nižších frekvenciách a neskoršie o vyšších frekvenciách. Ak teda budeme vychádzať z predpokladu, že informačný obsah je väčšinou obsiahnutý v nižších frekvenciách, môžeme dosiahnuť kompresie dát s DCT pomocou kvantovania koeficientov. Malé hodnoty sú kvantované hrubo, ideálne až na nulu a veľké hodnoty sú kvantované jemne, na najbližšiu celočíselnú hodnotu. Dekompresia je potom vykonávaná aplikovaním inverznej DCT kvantované koeficienty. Výsledné dáta nebudú zhodné s pôvodnými, nakoľko dochádza k strate, ale nelíšia sa príliš.

Formálne je DCT lineárne invertovateľná funkcia $F: \mathbb{R}^N \rightarrow \mathbb{R}^N$; transformujeme teda N reálnych čísel $x_0, x_1, \ldots, x_N$ na N reálnych čísel $X_0, X_1, \ldots, X_N$. Jednorozmerná DCT je popísaná ako [84]:

$$X_k = \sqrt{\frac{2}{N}} C_k \sum_{n=0}^{N-1} x_n \cos\left[\frac{\left(n+\frac{1}{2}\right) k\pi}{N}\right] \tag{40}$$

kde hodnoty škálovacieho faktoru C_k sú definované ako:

$$C_k = \begin{cases} \frac{1}{\sqrt{2}}, pre\ k = 0 \\ 1, pre\ 1 \leq k \leq N-1 \end{cases} \tag{41}$$

Hodnota N označuje maximálnu dĺžku signálu, tzn. rád transformácie, ktorú je schopná transformácia spracovať. V prípade že je vstupný signál dlhší, bude tento signál rozdelený na časti s maximálnou dĺžkou N, ktoré sú samostatne transformované. Prvý koeficient X_0 sa tiež nazýva DC koeficient a zvyšok je označovaný ako AC koeficienty.

Dvojrozmernú DCT môžeme vypočítať aplikovaním jednorozmernej DCT na každý riadok dátového bloku a následne na každý stĺpec výsledku. Popísať to môžeme ako [84]:

$$X_{kl} = \frac{2}{\sqrt{MN}} C_k C_l \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x_{kl} \cos\left[\frac{\left(m+\frac{1}{2}\right) l\pi}{M}\right] \cos\left[\frac{\left(n+\frac{1}{2}\right) k\pi}{N}\right] \tag{42}$$

pre $0 \leq k \leq N-1$ a $0 \leq l \leq M-1$ a C_k a C_l definované rovnako podľa vzorca (41). Je vhodné poznamenať, že aj keď je možné vykonať DCT na maticu rozmeru $M \times N$, väčšinou sa pre zjednodušenie výpočtov používa $M = N$. Prvý koeficient X_{00} je znovu označovaný ako DC, ostatné sú AC koeficienty.

Existujú štyri spôsoby ako vybrať N rovnomerne rozmiestnených uhlov, ktoré generujú ortogonálne vektory kosínusov. Tieto spôsoby korešpondujú (potom čo sú vektory normalizované pomocou škálovacieho faktoru) so štyrmi diskrétnymi kosínusovými transformáciami označovanými ako DCT-1 až DCT-4. Najpoužívanejšia je DCT-2, ktorá je všeobecne označovaná ako DCT a tento typ je použitý aj vo vzorcoch (40) a (42).

3.3.2 Hilbert-Huangová transformácia

Pomocou Hilbert-Huangovej transformácie (skrátene HHT) môžeme získať tzv. Hilbertovo spektrum, čo je forma časovo-frekvenčné amplitúdové spektrum. Algoritmus HHT pozostáva z dvoch procesov [85].

Prvý proces sa nazýva Empirická modálna dekompozícia (EMD). Pri použití EMD je signál, ktorý analyzujeme, adaptívne dekomponovaný na jednoduchšie úzkopásmové zložky, ktoré sa nazývajú vlastné modálne funkcie (IMF z angl. Intrisic Mode Functions) [86]. Vlastná modálna funkcia musí spĺňať dve podmienky: počet lokálnych extrémov a počet prechodov nulou sú zhodné alebo sa líšia maximálne o jedna od pôvodnej funkcie; priemerná hodnota obálky definovanej pomocou lokálneho maxima (horná obálka) a obálky definovanej pomocou lokálneho minima (dolná obálka) je konštantne rovná nule. Pre extrakciu všetkých IMF je použitý špeciálny algoritmus „preosievania“.

Algoritmus preosievania je nasledovný. Najprv sú vypočítane horná a dolná obálka signálu $x(t)$, spolu s ich priemerom $m_1(t)$. Prvým krokom algoritmu je výpočet rozdielu $h_1(t) = x(t) - m_1(t)$. Avšak $h_1(t)$ málokedy spĺňa vyššie spomínané podmienky pre IMF a je hneď zobraná ako prvá IMF. Preto je algoritmus preosievania vykonávaný viackrát a je rekurzívny, teda rozdiel získaný z predchádzajúceho preosievania je vstupný signál do ďalšieho. Ak po $k+1$ opakovaniach, korešpondujúci rozdiel $h_{1k}(t) = h_{1(k-1)}(t) - m_{1k}(t)$, spĺňa podmienky pre IMF, potom je tento rozdiel braný ako prvý IMF komponent a označíme ho ako $c_1(t)$, teda $c_1(t) = h_{1k}(t)$. V praxi, pre učenie či $h_{1k}(t)$ spĺňa podmienky pre IMF, je využívané kritérium smerodajnej odchýlky, ktoré kontroluje či je splnená nasledujúca nerovnosť [85]:

$$SD(k) = \sum_{t=0}^{T} \left[\frac{\left| h_{1(k-1)}(t) - h_{1k}(t) \right|^2}{h_{1(k-1)}^2(t)} \right] \leq 0.2 - 0.3 \tag{43}$$

kde T je dĺžka signálu. Následne zoberieme zvyšok signálu $r_1(t) = x(t) - c_1(t)$ ako „nový“ signál a aplikujeme naň algoritmus preosievania, aby sme získali druhý komponent IMF $c_2(t)$. Tento proces môžeme opakovať n krát, až kým posledné rezíduum $r_n(t)$ nie je monotónna funkcia. Keď je tento proces dekompozície dokončený, môžeme vyjadriť signál ako:

$$x(t) = \sum_{k=1}^{n} c_k + r_n \tag{44}$$

kde $c_1(t), c_2(t), \dots, c_n(t)$ sú všetky IMF komponenty signálu a $r_n(t)$ je zanedbateľné rezíduum.

V rámci druhého procesu HHT hľadáme amplitúdovo-časovo-frekvenčnú distribúciu, ktorú nazývame Hilbertovo spektrum, na základe získaných IMF. Najprv použijeme $c_k(t)$, ako reálnu časť a Hilbertovú transformáciu $c_k(t)$, $d(t) = \frac{1}{\pi}\int_{-\infty}^{\infty}\left(\frac{c_k(t')}{t-t'}\right)dt$, ako imaginárnu časť, aby sme vytvorili komplexnú analytickú funkciu $z_k(t)$:

$$z_k(t) = c_k(t) + id_k(t) = a_k(t)e^{i\theta_k(t)} \tag{45}$$

kde $a_k(t)$ je okamžitá amplitúda a $\theta_k(t)$ je okamžitá fáza IMF $c_k(t)$. Okamžitá frekvencia $\omega_k(t)$ IMF $c_k(t)$ je získaná ako $\omega_k(t) = \frac{d\theta_k(t)}{dt}$. To znamená, že môžeme signál $x(t)$ vyjadriť ako:

$$x(t) = Re\sum_{k=1}^{n} a_k(t)e^{i\int \omega_k(t)dt} \tag{46}$$

kedy rezíduum $r_n(t)$ je ignorované. Ďalším krokom je vykreslenie okamžitých amplitúd všetkých IMF do časovo-frekvenčnej roviny. Tento 3D obraz predstavuje špeciálne časovo-frekvenčnú rozloženie amplitúdy (alebo energie) signálu, ktorý je označovaný Hilbertovo spektrum a zvyčajne sa značí $H(\omega, t)$.

4 Experimentálna časť

4.1 Datasety

Pre vytvorenie kvalifikátora zvukov prostredia je potrebná množina vstupných dát – dataset, ktorý pozostáva z akustických nahrávok rôznych zdrojov zvuku, resp. činností produkujúcich zvuk. Keďže naším cieľom je klasifikácia samostatných akustických udalostí, nie akustických scén, mali by tieto nahrávky obsahovať každá len jeden anotovaný zdroj zvuku. V rámci výskumu klasifikácie environmentálnych zvukov bolo zostavených niekoľko datasetov, ktoré túto podmienku spĺňajú, preto nebolo potrebné zostavovať pre potreby nášho výskumu vlastný. Z hľadiska množstva dostupných dát sa ponuka AudioSet [64], ktorý bol zostavený z 10 sekundových zostrihov zvuku z YouTube videí, pozostáva z 527 kategórií zvuku. Avšak v rámci týchto kategórií sú aj zvuky hudobných nástrojov a ľudskej reči, preto je pre naše potreby nevhodný. Ďalšia nevýhoda je, že nie sú k dispozícii čisté nahrávky, ale len vopred extrahované príznaky. Z ohľadom na veľkosť množiny nahrávok a kategórií sa ako ďalší ponuka FSD50K [65], ktorý inšpiráciu čerpal v AudioSet-e a prevzal z neho 200 kategórií zvuku, pomocou ktorých boli označené nahrávky prevzaté z projektu FreeSound [66]. Avšak aj tento dataset obsahuje nahrávky hudobných nástrojov a ľudskej reči. Tento dataset je už zostavený zo samotných nahrávok, avšak tieto môžu obsahovať viacero anotácií. Dataset, ktorý neobsahuje zvuky hudobných nástrojov, ani ľudskú reč, je UrbanSound8K [8], ktorý bol zostavený na základe sťažností na hluk v New Yorku medzi rokmi 2010 a 2014, z ktorých bolo určených 10 rozličných tried zvuku. Nahrávky boli potom taktiež ako FSD50K získané z projektu FreeSound. Dataset pozostáva z 8 732 audio nahrávok, každá s maximálnou dĺžkou štyri sekundy. Tento dataset je úzko spätý s výskumom zvukového znečistenia v mestskej časti. Pre potreby tejto práce, sme sa nakoniec rozhodli využiť dataset ESC-50 [26], ktorý bol zostavený K. Piczakom. Tento dataset je z ohľadom na veľkosť menší ako predchádzajúce, pozostáva z 2 000 nahrávok rozdelených do 50 tried, ktoré môžeme zhrnúť do piatich kategórií pôvodu. Každá nahrávka je dlhá päť sekúnd a má pôvod v projekte FreeSound. Rozmanitosť pôvodov zdrojov zvuku, ktoré nie sú viazané na konkrétnu oblasť, je dôvod prečo je tento dataset vhodný pre návrh všeobecného klasifikátora. V tabuľke 1 je možne vidieť stručné porovnanie vyššie spomenutých datasetov.

dataset	Počet nahrávok	Dĺžka nahrávky	Celková dĺžka	Počet tried	Zdroj nahrávok
AudioSet	2,1 M	10 s	5 731 h	527	Youtube
FSD50K	51 197	0,3 – 30 s	108 k	200	FreeSound
UrbanSound8K	8 732	≤ 4 s	8,8 h	10	FreeSound
ESC-50	2 000	5 s	2,8 h	50	FreeSound

Tabuľka 1 Všeobecné porovnanie datasetov

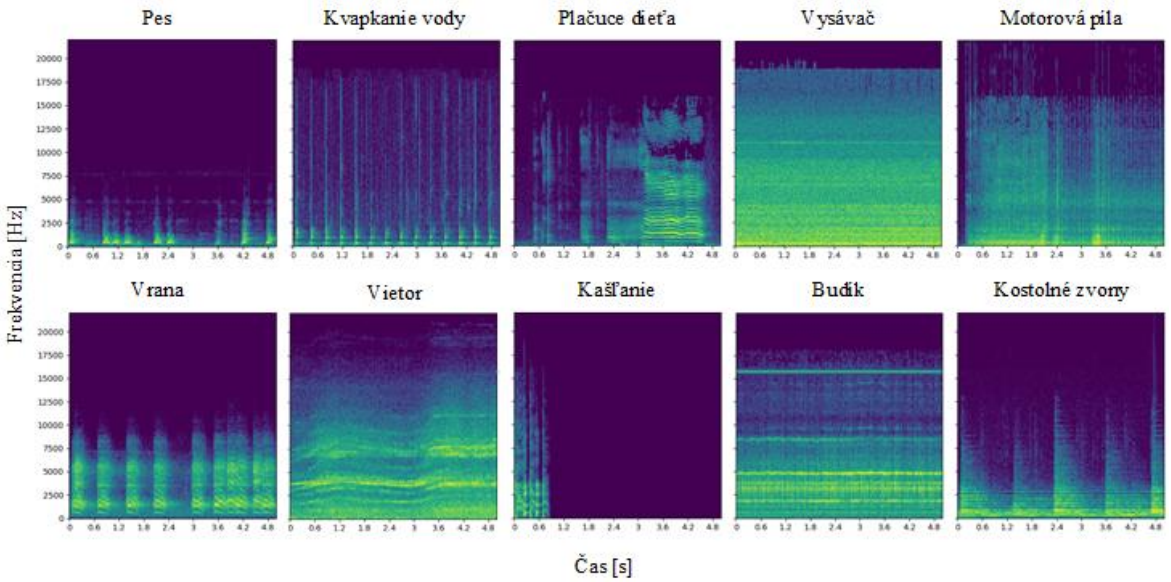
4.1.1 ESC-50

Ako už bolo spomenuté, dataset ESC-50 pozostáva z 2 000 anotovaných environmentálnych nahrávok samostatných akustických udalostí. Tieto nahrávky sú rovnomerne rozložené do päťdesiatich kategórií, 40 nahrávok na každú triedu. Tieto triedy sú rozložené do piatich voľne definovaných oblastí pôvodu, detailné zobrazenie tried je v tabuľke 2.

Zvieratá	Zvuky prírody a zvuky vody	Ľudské nerečové zvuky	Interiérové/ domáce zvuky	Exteriérové/ mestské zvuky
Pes	Dážď	Plačúce dieťa	Klopanie na dvere	Helikoptéra
Kohút	Morské vlny	Kýchnutie	Klikanie myšou	Motorová píla
Prasa	Praskanie ohňa	Tlieskanie	Písanie na klávesnici	Siréna
Krava	Cvrčky	Dýchanie	Dvere, vŕzganie dreva	Klaksón auta
Žaba	Čvirikajúce vtáky	Kašľanie	Otváranie konzervy	Motor
Mačka	Kvapkanie vody	Kroky	Práčka	Vlak
Sliepka	Vietor	Smiech	Vysávač	Kostolné zvony
Hmyz (lietajúci)	Nalievanie vody	Čistenie zubov	Budík	Lietadlo
Ovca	Spláchnutie toalety	Chrápanie	Tikanie hodín	Ohňostroj
Vrana	Búrka	Pitie, popíjanie	Rozbíjanie skla	Ručná píla

Tabuľka 2 Rozdelenie tried datasetu ESC-50

Ako je možné vidieť tento dataset obsahuje zvuky, ktoré sú veľmi bežné (smiech, štekot psa), niektoré celkom zreteľné (rozbíjanie skla, čistenie zubov) a niektoré, kde sú rozdiely jemnejšie (zvuk lietadla a helikoptéry). Uvedené zdroje zvuku sú veľmi heterogénne, čo sa týka dĺžky trvania či intenzity. Zároveň, ako môžeme vidieť na obrázku 18, zvuky majú vo frekvenčnej oblasti priebeh podobný šumu, ako napríklad vysávač a niektoré, ako napríklad budík, majú výrazné nosné signály, ktorých frekvencie môžeme odčítať. Toto môže spôsobovať ťažkosti pre model strojového učenia, ktorý sa snaží naučiť zvuky, ktoré môžu byť odlišné nahrávku od nahrávky.



Obrázok 18 Príklady spektrogramov nahrávok v datasete ESC-50

Zároveň môžeme vidieť, že v niektorých prípadoch, aj keď dĺžka nahrávky je 5 sekúnd, užitočný zvuk je prítomný len určité percento tohto času. Keďže väčšina prístupov klasifikácie delí nahrávky na menšie rámce, ktoré potom klasifikuje podľa vopred volenej schémy, vyvstáva otázka slabého anotovania. Teda v prípade, že človek vychádza z predpokladu, že užitočný zvuk je prítomný po celé trvanie nahrávky a tá je rozdelená na rámce, ktoré zdedia anotáciu celej nahrávky, môžu niektoré rámce zdediť anotáciu, aj keď neobsahujú užitočný zvuk, čo môže mať za následok pokles presnosti rozpoznávania.

4.2 Metódy strojového učenia

Ako už bolo spomenuté v predchádzajúcej časti, vybrali sme pre základ našej práce dataset ESC-50. Autor Karol Piczak, ktorý zostavil tento dataset environmentálnych zvukov, na ňom spravil niekoľko štúdií, ktoré boli zamerané na efektivitu klasifikačných metód strojového učenia [26][67]. Tieto metódy zahŕňali k-Najbližších susedov, ktorá dosahovala rozpoznávanie 32,2%, model najlepšie rozpoznával „kýchnutie“, „búrku“ a „otváranie

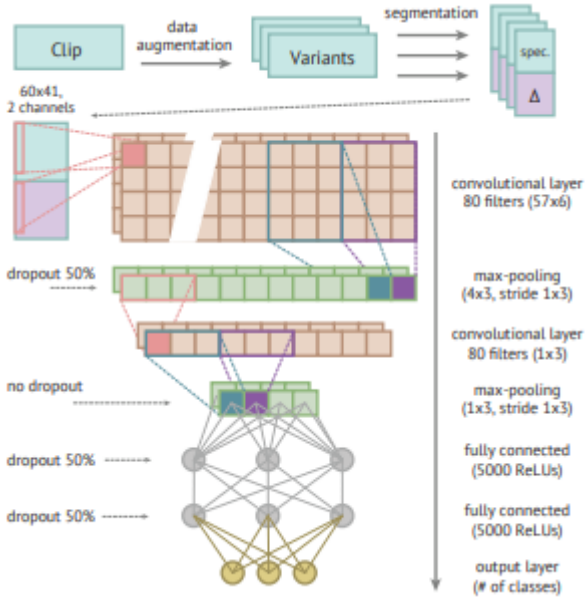
plechovky“ a naopak problémové triedy zvuku boli „tikanie hodín“, „klaksón auta“ a „motor“, kde bolo rozpoznávanie takmer nulové. Ďalšia bola metóda podporných vektorov dosahovala rozpoznávanie 39,6%, model založený na tejto metóde najlepšie rozpoznával „búrku“ a „klopanie“ a problematické triedy boli „dvere, vŕzganie dreva“ „helikoptéra“ a „motor“. Poslednou z klasických metód klasifikácie, bola metóda náhodný les, ktorá dosiahla rozpoznávanie 44,3%, model rozpoznával najlepšie triedy „klopanie“ a „búrka“, naopak problém s rozpoznávaním mal pri triedach „tikanie hodín“, „klaksón auta“ a „kvapkanie vody“. V článku [67] potom popísal model, založený na umelých neurónových sieťach, konkrétne na konvolučných neurónových sieťach. Najlepší model potom dosiahol rozpoznávanie 64,5%. To nám ukazuje, že metóda strojového učenia založená na umelých neurónových sieťach dosahuje lepších výsledkov ako klasické klasifikátory, čo je ďalej podporené, že na GitHub-ovej[2] stránke tohto datasetu je uvedený rebríček modelov a presnosť ich rozpoznávania a väčšina týchto modelov je založená práve na umelých neurónových sieťach. Z toho dôvodu sme sa aj my rozhodli pre architektúru modelu založenú na neurónových sieťach, konkrétne na konvolučných neurónových sieťach. Model, ktorý navrhol K. Piczak, bol zvolený ako referenčný. Ako už bolo spomínané väčšina doterajších prístupov používa veľmi hlboké modely, ktorých presnosť rozpoznávania je vyššia ako v prípade Piczakovho modelu, rovnako je však vyššia aj ich veľkosť, v niektorých prípadoch dosahuje až 87 miliónov. Nakoľko naším cieľom práce je návrh architektúry s malou veľkosťou modelu, resp. s nízkym počtom parametrov, a tento typ prístupu nemá takú popularitu, rozhodli sme sa preto použiť Piczakov model ako referenčný, nakoľko sa jedná o prvotné riešenie použitého datasetu ESC-50. Tento model využíval schému spracovávania environmentálnych zvukov na báze podrámcov. Teda schéma, kedy je každá nahrávka rozdelená do menších podrámcov, zvyčajne s nejakou úrovňou prekrytia a príznaky sú extrahované z každého rámca samostatne. Aby sa klasifikátor mohol trénovať, príznaky z jednotlivých podrámcov sú buď spojené do jedného veľkého vektora príznakov, alebo spriemerované tak, aby predstavovali jeden rámec. Druhá možnosť je nechať klasifikátor trénovať na každom podrámci a po spracovaní všetkých vykonať kolektívne zvolenie výslednej triedy pre celý rámec, na základe tried zvolených zo všetkých podrámcov, napríklad volenie podľa majority alebo volenie na základe pravdepodobnosti [68]. Mel spektrogramy, ktoré tento model požíva na extrakciu príznakov, boli rozdelené na 41 podrámcov s prekrytím 50% v prípade krátkeho variantu, alebo na 101 podrámcov

[2] https://github.com/karolpiczak/ESC-50

s prekrytím 90% v prípade dlhého variantu. K tým boli pridané ich delta príznaky, ktoré sú počítane pomocou Savitsky-Golay filtrovania a vytvorili tak dvojkanálový vstup. Predikcia triedy pre celú nahrávku, resp. celý rámec bola vykonaná buď na základe majority alebo na základe pravdepodobnosti predikovanej triedy z každého segmentu, čo v kombinácii s krátkym variantom poskytovalo najlepšie výsledky. Na obrázku 19 je možné vidieť architektúru modelu, ktorý navrhol K. Piczak. Táto sieť má v krátkom variante nasledujúce parametre:

- Trénovateľných parametrov: 26 534 130
- Veľkosť parametrov: 106,14 MB
- Celkový počet násobenie-sčítanie operácií: 34,54 *10^6

Tieto údaje boli vygenerované pomocou nástroja torchinfo[3].

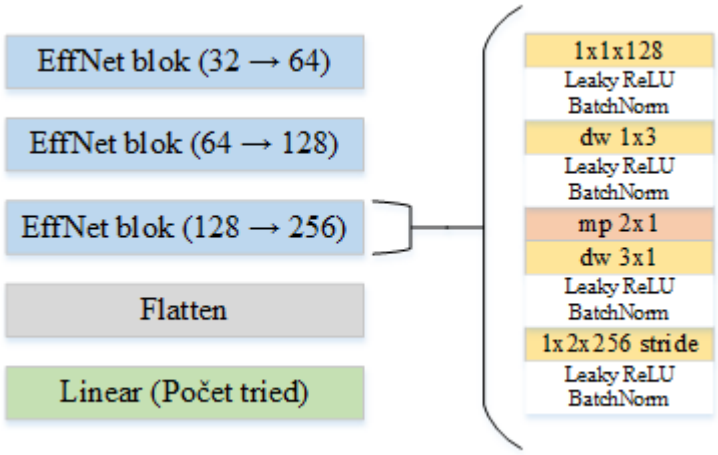


Obrázok 19 Architektúra konvolučnej siete K. Piczak [67]

Keďže výsledky konvolučných neurónových sietí, dosahujú v oblasti rozpoznávania obrazu vynikajúce výsledky, rozhodli sme sa čerpať inšpiráciu práve v týchto architektúrach. Konkrétne sa teda jedná o architektúry, ktoré boli navrhnuté s ohľadom na výpočtovú efektivitu, určené pre mobilné zariadenia. Navrhnutých bolo viacero takýchto konvolučných sietí. My sme v rámci rozboru zvažovali troch kandidátov ShuffleNet [69], MobileNet [70] a EffNet [71]. V rámci úvahy sme zhodnotili, že architektúry ShuffleNetu a MobileNetu

[3] https://github.com/TylerYep/torchinfo

obsahujú komplexné bloky, v prípade ShuffleNetu je to miešanie kanálov, v prípade MobileNetu je to invertované residuálne prepojenie „bottleneck“ bloku, a tým sťažujú ich možnú úpravu pre potreby našej práce. Preto sme sa rozhodli adaptovať EffNet, nakoľko jeho pomerne jednoduchá architektúra ponúka priestor pre ďalšie úpravy a je to teda vhodná štartovacia sieť. Na obrázku 20 môžeme vidieť zobrazenie architektúry EffNet. Ako môžeme vidieť, táto sieť pozostáva z troch tzv. EffNet blokov. Jeden z týchto blokov je zobrazený detailne. Na začiatku tohto bloku je vykonaná bodová konvolúcia (angl. pointwise convolution), následne je použitý špeciálny druh konvolúcie, ktorý sa nazýva priestorovo oddeliteľná konvolúcia (angl. spatial separable convolution), ktorej princíp je nahradenie jednej operácie konvolúcie s veľkosťou jadra VxŠ, dvomi operáciami konvolúcie s veľkosťami jadra, najprv Vx1 a následne 1xŠ, pomocou čoho zníži počet parametrov. V prípade EffNet bloku bola medzi tieto dve konvolučné operácie vložená operácia združovania podľa maximálnej hodnoty, max-pool. Zároveň je za každú konvolučnú vrstvu pridaná kombinácia vrstva nelineárnej aktivačnej funkcie Leaky ReLU a vrstva dávkovej normalizácie, ktorá normalizuje vstup do vrstvy a dovolí tak využitie vyšších hodnôt parametrov učenia a zároveň zníži počet potrebných epoch pre trénovanie. Posledná vrstva tohto bloku je konvolučná vrstva s krokom konvolúcie rovnakým ako je veľkosť jadra, aby došlo k ďalšej redukcii dát. Po týchto troch blokoch nasleduje vrstva „Flatten“, ktorá transformuje vstup na jednorozmerný výstupný vektor, ktorý je možné spracovať za pomoci plne-prepojenej vrstvy (Linear), ktorá vykonáva finálnu klasifikáciu.



Obrázok 20 Architektúra EffNet s detailným EffNet blokom. „dw“ znamená hĺbková konvolúcia (angl. depthwise convolution) a „mp“ znamená združovanie podľa maxima (z angl. max-pool)

4.3 Predspracovanie dát

Pri tvorbe tohto datasetu K. Pizcak rekonvertoval všetkých 2 000 nahrávok na jednotný formát:

- vzorkovacia frekvencia: 44 100 Hz,
- kanály: Mono,
- kodek: 16 bit PCM (S16 LE).

V rámci predspracovania dát sme sa rozhodli podvzorkovať tento akustický signál na vzorkovaciu frekvenciu 16 000 Hz. Hlavný dôvod, prečo sme sa rozhodli podvzorkovať tieto akustické signály, bola snaha o redukciu rozmernosti dát. Poznáme viacero metód, ako podvzorkovať signál, avšak keďže náš konverzný pomer nie je celočíselný, niektoré z nich nemôžeme použiť, ako napríklad proces decimácie. V rámci našej práce využívame metódu pásmovo-obmedzenej sinc interpolácie, ktorá je implementovaná aj v knižnici librosa, aj PyTorch audio, ktorá je ideálna pre digitálny audio signál [89]. Medzi nastavenia tejto metódy prevzorkovania patria [90]:

- Šírka dolnopriepustného filtra - taktiež označovaný ako počet prechodov nulou, pomocou ktorého kontrolujeme veľkosť filtra, ktorým ohraničíme interpoláciu. Použitím väčšej šírky poskytne ostrejší, presnejší filter, ale je výpočtovo viac náročný.
- Rollof – reprezentuje zlomok Nyquistovej frekvencie. Tento parameter určuje hranicu dolnopriepustného filtra a kontroluje úroveň aliasingu, ktorý nastáva keď frekvencie vyššie ako Niquistová frekvencia sú mapované do nižších frekvencií. Nižšia hodnota znamená redukciu aliasingu, avšak redukuje aj niektoré vyššie frekvencie.
- Váhová funkcia, resp. váhové okno – predstavuje druh váhového okna, ktoré bude aplikované na signál. Kaiserovo okno je takmer optimálne váhové okno, ktoré poskytuje ďalší parameter, pomocou ktorého môžeme ďalej ovplyvniť hladkosť filtra a šírku impulzu.

Nastavenie, ktoré dosahuje vysokej kvality, sa nazýva „kaiser_best“[4] a toto nastavenie využívame v našej práci pre podvzorkovanie dát.

Z hľadiska procesu klasifikácie sme sa rozhodli pre schému používajúcu rámce, teda nahrávka je vopred rozdelená na rámce. Z týchto rámcov sú extrahované príznaky a tieto sú následne použité pre trénovanie alebo testovanie. Rozhodovanie klasifikátora o triede zvuku

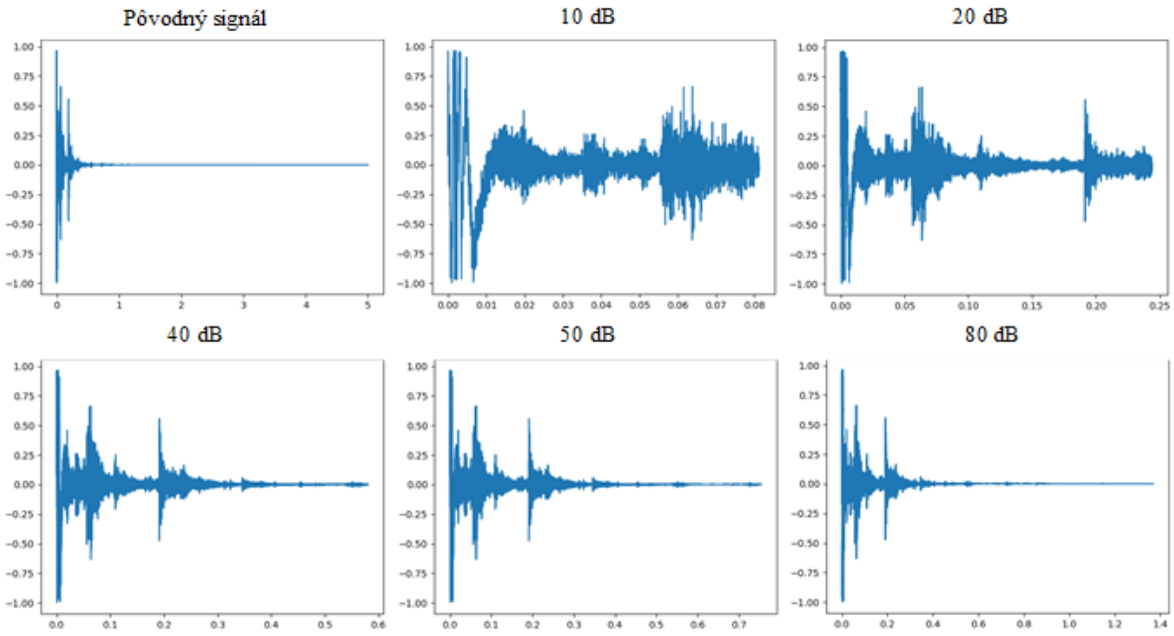
[4] https://resampy.readthedocs.io/en/master/api.html#module-resampy.filters

je vykonávané na každý rámec zvlášť, preto po sebe idúce rámce môžu patriť do rozličných tried. Nevýhoda tejto procesnej schémy je v tom, že niektoré zvuky sú krátkodobé, napríklad rozbitie skla a niektoré sú dlhodobé, napríklad búrka; preto je potrebné sa pri voľbe veľkosti rámca riadiť kompromisom. Ak je rámec príliš krátky, potom dlhodobé zmeny signálu nebudú zahrnuté počas extrakcie príznakov, naopak ak je rámec príliš dlhý, krátkodobé zvuky sa môžu stratiť [68].

Ďalší problém vyvstával, ako bolo spomenuté, s tým, že užitočný zvuk sa v nahrávke môže nachádzať len určité percento času. Prvá možnosť, ako tento problém vyriešiť je ignorovať ho, ale to by malo za následok slabo anotované dáta, resp. zle anotované dáta, nakoľko by rámec neobsahoval zvuk, ktorému je priradený, čo by malo za následok pokles presnosti rozpoznávania. Ďalšia metóda je orezať ticho alebo „prázdno“ a následne vytapetovať odrezaný čas užitočným zvukom tak, aby bola zachovaná dĺžka nahrávky 5 sekúnd, zároveň tak dataset zostane balancovaný . Túto metódu sme spočiatku využívali, ale neskôr bola vypustená, pretože neposkytovala signifikantné zlepšenie presnosti rozpoznávania; druhý dôvod bolo to, že táto metóda zavádza periodicitu tam, kde prirodzene nie je, napríklad rozbitie skla sa, požitím tejto metódy, stalo rozbitím niekoľkých skiel. Preto sme neskôr implementovali metódu, ktorá oreže ticho a ponechá len užitočný zvuk, čo má síce za následok, že dataset nie je plne balancovaný, ale inak nevytvára žiadnu nevýhodu.

Odstraňovanie ticha bolo vykonávané pomocou knižnice librosa[5]. Jedným z bežných problémov pri použití orezania ticha, ako metódy predspracovania pre celý dataset, je nastavenie hraničnej hodnoty v decibeloch, a teda hodnoty pod hranicou budú vyhodnocované ako ticho. Analyzované boli úrovne 10 dB, 20 dB, 40 dB, 50 dB a 80 dB s tým, že referenčná hodnota je maximálna hodnota výkonu signálu. Na obrázku 21 je možné vidieť príklad orezávaného akustického signálu, konkrétne zvuk z triedy „rozbitie skla“.

[5] https://librosa.org/doc/latest/index.html



Obrázok 21 Zvuk rozbitia skla s rôznymi nastaveniami orezania ticha

Hranica orezávania ticha na úrovni 40 dB sa ukázala ako najvhodnejšia, nakoľko nedochádza k možnej strate užitočnej informácie ako pri 10 dB a 20 dB ale orezáva ticho dostatočne skoro, aby užitočný zvuk zostal dominantný.

4.4 Extrakcia príznakov

Na rozdiel od úloh klasifikácie obrazu, klasifikácia environmentálnych zvukov predpokladá využitie lokálne korelovaných jednorozmerných signálov, čiže vstup je natiahnutý pozdĺž jednej osi. Reprezentácia akustického signálu je odlišná od vizuálnych signálov, ako sú fotografie, ktoré majú lokálne korelácie v oboch priestorových dimenziách. Z toho dôvodu bolo navrhnutých niekoľko metód špecificky pre oblasť audio signálu. Jednou z oblastí týchto metód sú Vopred vypočítané časovo-frekvenčné reprezentácie a ich následne spracovanie pomocou 2D konvolučnej siete. Do tejto kategórie patrí aj Piczaková sieť, ktorá operuje so vstupom vo forme Mel spektrogramov. Základom väčšiny týchto metód je Krátkodobá Fourierová transformácia, a následne môže byť spektrum ďalej spravované, napríklad do podoby Mel spektrogramov, Mel-Frekvenčných kepstrálnych koeficientov, spektrálny kontrast, chromagram a ďalšie.

Modely strojového učenia potom môžu využívať jednu z týchto metód, tzn. jedno-príznakový vstup alebo viacero týchto metód v kombinácii a vytvoriť tak viac-príznakový vstup, ako je napríklad Piczakov model, ktorý používa mel spektrogramy v kombinácii s delta príznakmi.

4.4.1 Metóda reformácie spektrogramu

Ako je možné si povšimnúť, veľa z vyššie spomenutých metód má svoj pôvod v rozpoznávaní reči alebo hudby. My sme sa v našej práci rozhodli nepoužiť žiadnu z týchto metód. Tieto metódy síce ponúkajú dodatočnú redukciu rozmernosti vstupných dát, avšak prístupov založených na týchto metódach, ako tých čo využívajú Mel spektrogramy je už dostatok alebo ako v prípade Mel-Frekvenčných kepstrálnych koeficientov sa ukázalo, že neprodukujú príliš silné riešenia.

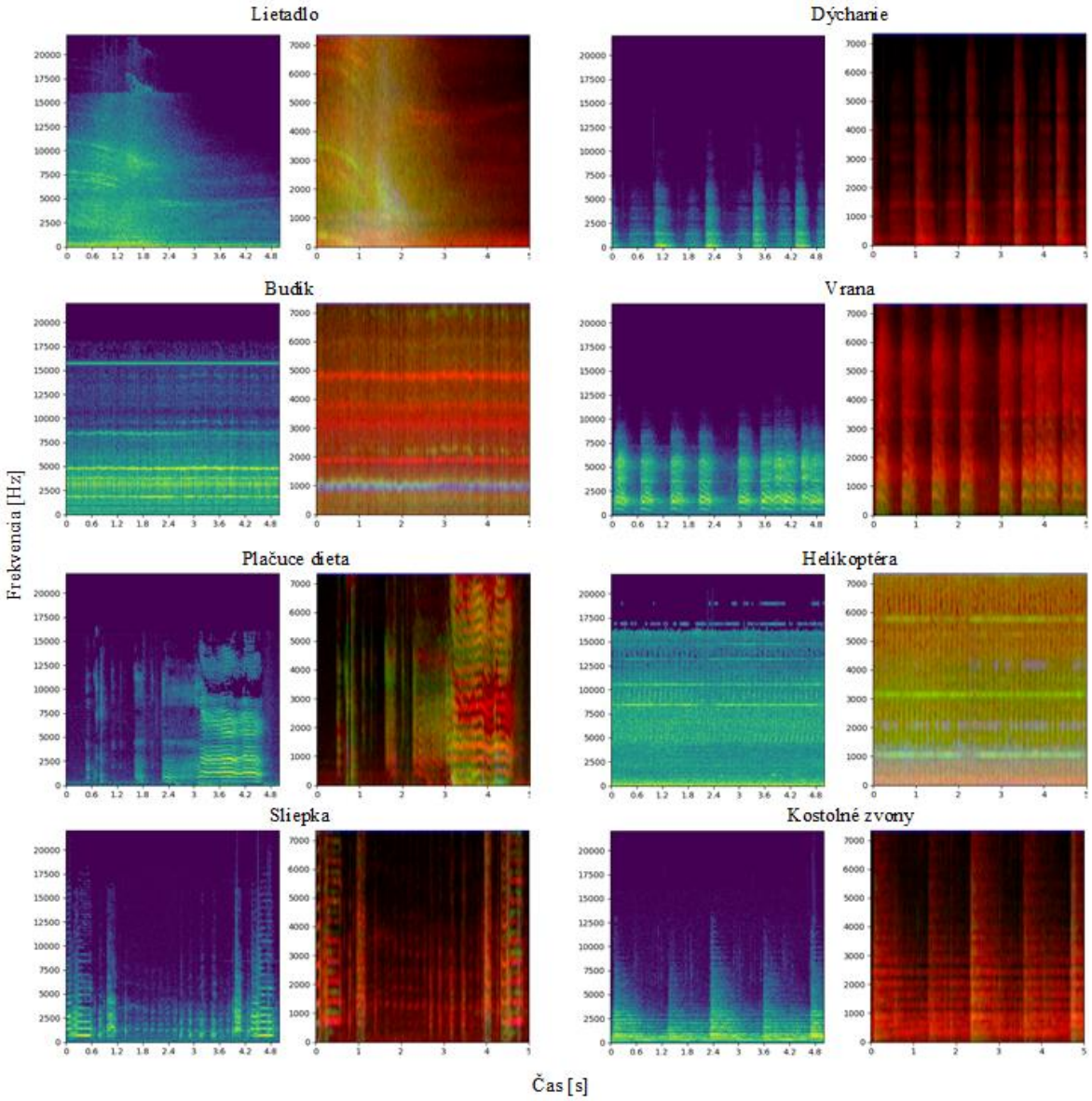
V rámci našej práce sme sa rozhodli využívať transformáciu vstupného signálu pomocou STFT, na základe čoho môžeme získať výkonový spektrogram, ktorý prevedieme do logaritmickej mierky.

Je vhodné poznamenať, že veľkosť konvolučnej neurónovej siete rastie spolu s veľkosťou vstupných dát. Zvyšuje sa počet jej parametrov, z toho vyplývajúca celková veľkosť, ako aj počet operácií násobenie-sčítanie. Tento nárast je zvlášť signifikantný v prípade plne-prepojenej vrstvy, pomocou ktorej je vykonávaná klasifikácia.

Ako metódu redukcie rozmernosti vstupných dát sme si zobrali za príklad rozpoznávanie farebného obrazu. V tomto prípade majú vstupné dáta tri kanály, teda jeden kanál pre každú farebná zložku formátu RGB. V rámci tejto metódy je vstupný jednokanálový spektrogram rozdelený pozdĺž frekvenčnej osi na tri frekvenčné pásma, ktoré sú potom mapované do troch kanálov. Spodné frekvenčné pásmo ako červená zložka farby, stredné frekvenčné pásmo ako zelená zložka farby a vrchné frekvenčné pásmo ako modrá zložka farby. Výsledné RGB zobrazenie je následne spracovávané konvolučnou neurónovou sieťou.

Takto mapované vstupné dáta ponúkajú redukciu počtu parametrov konvolučnej neurónovej siete, jej celkovej veľkosti a počtu operácii násobenie-sčítanie na približne 35 % oproti prípadu, kedy sú ako vstupné dáta použité jednokanálové spektrogramy. Je nutné poznamenať, že v prípade tejto metódy nedochádza k zníženiu počtu hodnôt vo vstupných dátach; tato metóda je využitá pre redukciu rozmernosti konvolučnej siete ako takej.

Na obrázku 22 je možné vidieť ukážky aplikácie tejto metódy- reformácie spektrogramu do RGB zobrazenia.



Obrázok 22 Príklady mapovania spektrogramov ako RGB zobrazenie

Následne bolo nutné tieto dáta pred vstupom do konvolučnej siete normalizovať, nakoľko tieto majú tendenciu dosahovať lepších výsledkov, ak sú vstupné dáta normalizované. Všeobecne sa využívajú dva spôsoby normalizácie:

Min-max normalizácia, pri ktorej je výstup škálovaný do požadovaného rozsahu, bežne sa škáluje na rozsah <0,1> alebo <-1,1>, toto je vykonávané podľa vzorca:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}(b - a) + a \tag{47}$$

kde x_{norm} je normalizovaný výstup, x je vstup, x_{min} je minimálna hodnota zo vstupu, x_{max} je maximálna hodnota vstupu a $\langle a, b \rangle$ sú hraničné hodnoty požadovaného rozsahu.

Druhou všeobecné používanou metódou normalizácie je tzv. z-skóre normalizácia, pomocou ktorej dosiahneme, že všetky vstupné dáta budú mať priemer rovný nule a smerodajnú odchýlku rovnú jednej. Táto normalizácia sa vykonáva podľa vzorca:

$$x_{norm} = \frac{x - \mu}{\sigma} \tag{48}$$

kde μ je priemerná hodnota a σ je smerodajná odchýlka.

V rámci úvodných testov sa ukázalo, že konvolučná sieť dosahuje lepšie výsledky v prípade z-skóre normalizácie ako min-max a preto bola táto metóda implementovaná do ďalších experimentov.

4.4.2 Experiment – Veľkosť analyzovaného rámca

Ako už bolo spomínané, konvolučná sieť rastie s veľkosťou vstupu, ktorý je priamo ovplyvnený veľkosťou analyzovaného rámca. Jedným z našich cieľov bolo, aby bol zvuk klasifikovaný na základe nahrávky do jednej sekundy. Konečná veľkosť rámca by však mala brať do úvahy krátkodobé aj dlhodobé zvuky. Preto sme sa rozhodli vykonať experimentálne meranie vplyvu veľkosti analyzačného rámca na presnosť EffNetu, ako nami adaptovanej siete.

V rámci tohto experimentu sme trénovali EffNet na rámcoch s čoraz väčšou veľkosťou od 250 ms po 1 sekundu s inkrementom po 10 ms a zaznamenávali sme presnosť rozpoznávania po natrénovaní. Pre každé nastavenie analyzačného rámca bol EffNet natrénovaný 10 krát. Je vhodné poznamenať, že čím menší je analyzačný rámec, na tým viacej rámcov je rozdelená jedna nahrávka a z toho vyplýva viacej dát pre trénovanie, čo môže mať za následok lepšie rozpoznávanie. Pre prácu s audio signálom bola využitá knižnica librosa.

Parametre extrakcie príznakov:

- podvzorkovanie na 16 000 Hz,
- prekrytie jednotlivých rámcov: 50%,
- dĺžka STFT: 512 vzoriek s prekrytím 50%,
- váhová funkcia: Hanning,
- ticho bolo vyplnené užitočným zvukom.

Pre implementáciu konvolučnej siete EffNet bol použitý framework PyTorch[6] a na trénovanie bola použitá knižnica Ignite[7]. Ako bolo spomenuté vyššie architektúra EffNetu používa pre finálnu klasifikáciu plne-prepojenú vrstvu, ktorej veľkosť sa mení s veľkosťou vstupu. To predstavovalo problém, nakoľko rôzne veľkosti analyzačného rámca vyžadovali rôzne nastavenia vstupu plne-prepojenej vrstvy. Riešenie pozostávalo z orezania EffNetu po treťom bloku, následne bol vykonaný jeden prechod sieťou, s použitím požadovaného rámca, a na základe výsledku bola vypočítaná potrebná veľkosť vstupu do plne-prepojenej vrstvy. Toto bolo vykonané po každej zmene veľkosti rámca.

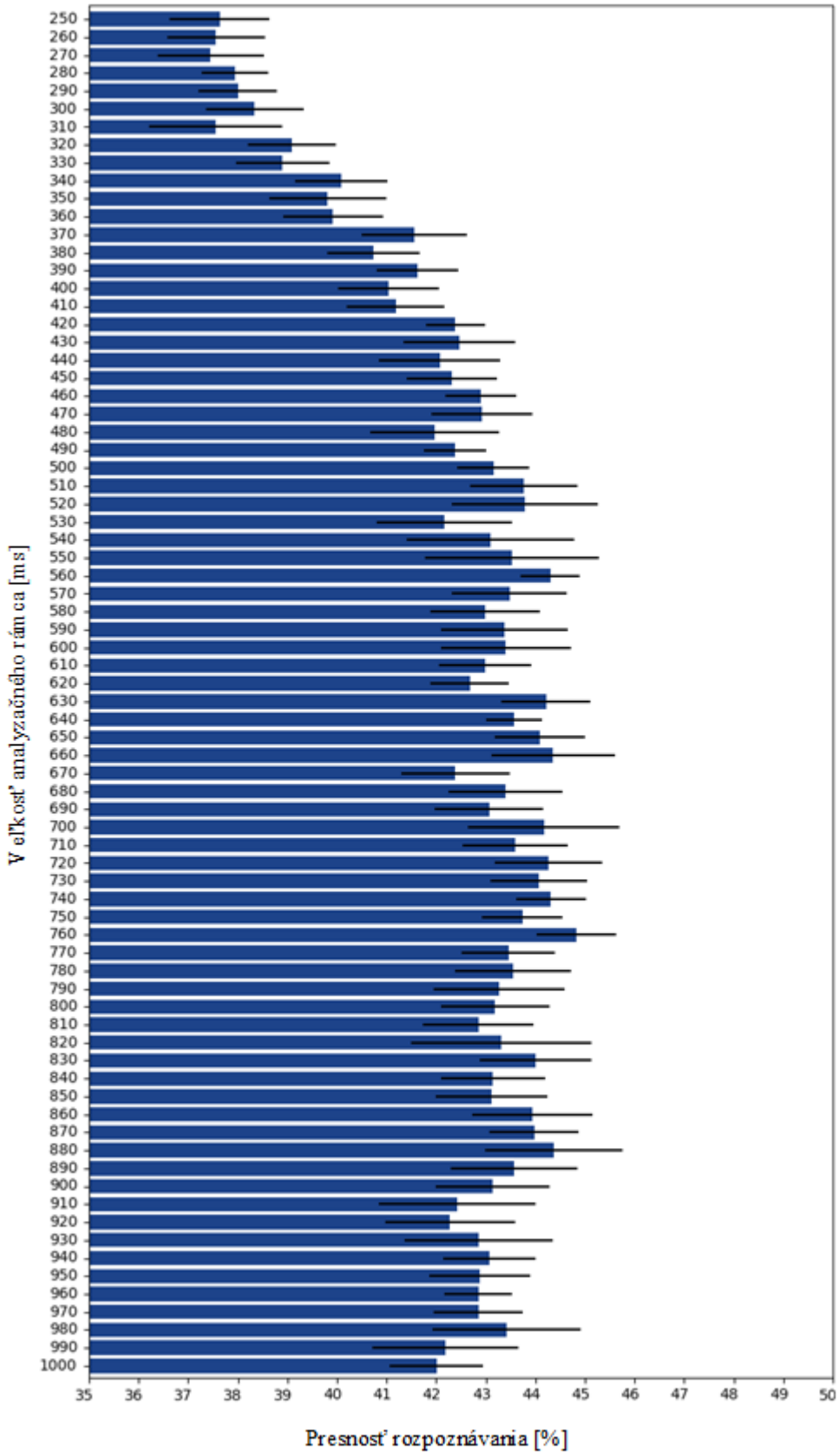
Nastavenie parametrov trénovania:

- počet epoch: 15,
- veľkosť dávky: 16,
- optimalizátor: Stochastický gradientový zostup (z angl. Stochastic Gradient descent, SGD) s použitím Nestrovho momentu 0.9,
- parameter učenia (angl. learning rate): 0,005,
- stratová funkcia: Negatívna logaritmická vierohodnosť (z angl. Negative log likelihood loss, NLLLoss),
- zoslabovanie váh: 0,001.

Na obrázku 23 je možné vidieť graf predstavujúci vyhodnotenie tohto experimentu. Pre každú veľkosť analyzačného rámca bola vypočítaná priemerná presnosť rozpoznávania po 15 epochách trénovania, čierna čiara potom predstavuje smerodajnú odchýlku.

Môžeme pozorovať, že aj keď rozdelenie nahrávok na menšie analyzačné rámce (250 – 310 ms) poskytuje viacej rámcov pre trénovanie, ich priemerná presnosť bola nízka. Najhoršiu priemernú presnosť rozpoznávania 37,43% dosahoval model pri použití veľkosti analyzačného rámca 270 ms. Najlepšia priemerná presnosť rozpoznávania bola dosiahnutá pri veľkosti okna 760 ms s priemernou presnosťou rozpoznávania 44,81%. Zároveň však toto nastavenie vykazovalo relatívne malú smerodajnú odchýlku, oproti nastaveniu, ktoré boli v presnosti za ním (550, 660, 880). Z týchto dôvodov sme sa rozhodli implementovať práve veľkosť analyzačného rámca 760 ms pre budúce trénovanie.

[6] https://pytorch.org/
[7] https://pytorch.org/ignite/index.html



Obrázok 23 Vyhodnotenie experimentu vplyvu veľkosti analyzačného rámca

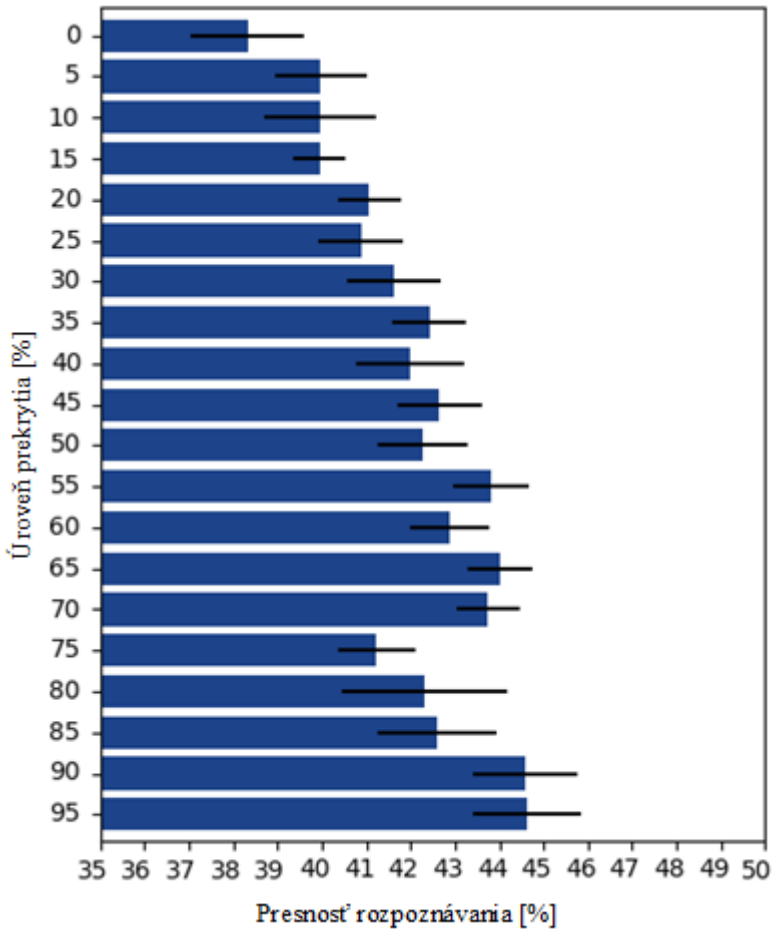
4.4.3 Experiment – Veľkosť prekrytia

V rámci tohto experimentu sme zisťovali, ako úroveň prekrytia ovplyvňuje presnosť rozpoznávania. Bola nastavená fixná veľkosť analyzačného rámca, následne bol EffNet natrénovaný 10 krát pre každé nastavenie úrovne prekrytia od 0 – 95% s inkrementom po 5%.

Parametre extrakcie príznakov:

- podvzorkovanie na 16 000 Hz,
- veľkosť analyzačného rámca: 500 ms,
- dĺžka STFT: 512 vzoriek s prekrytím 50%,
- váhová funkcia: Hanning,
- ticho bolo vyplnené užitočným zvukom.

Nastavenie parametrov učenia konvolučnej siete bolo rovnaké ako v predchádzajúcom pokuse. Na obrázku 24 je možné vidieť graf predstavujúci vyhodnotenie tohto experimentu.



Obrázok 24 Vyhodnotenie experimentu úrovne prekrytia

Pre každé nastavenie bola vypočítaná priemerná presnosť rozpoznávania a smerodajná odchýlka, ako môžeme vidieť v grafe. Je vhodné poznamenať, že čím vyššie je prekrytie, tým viacej analyzačných rámcov je možné generovať z jednej nahrávky, avšak v prípade príliš vysokého prekrytia tieto rámce budú značne korelované. Keďže my klasifikujeme jednotlivé rámce samostatne a nie sekvencie menších rámcov, je vhodné, aby použitý dataset vstupných dát nebol príliš korelovaný, preto sme vyradili nastavenia prekrytia 90% a 95%. V prípade, že by sme použili klasifikačnú schému rozdelenia na rámce, bolo by vhodné použitie týchto vyšších nastavení prekrytia. Rozhodli sme sa použiť nastavenie prekrytia 65% , nakoľko pri tomto nastavení dosahoval EffNet dobré výsledky a zároveň vzniknutý dataset analyzačných rámcov nebol príliš korelovaný.

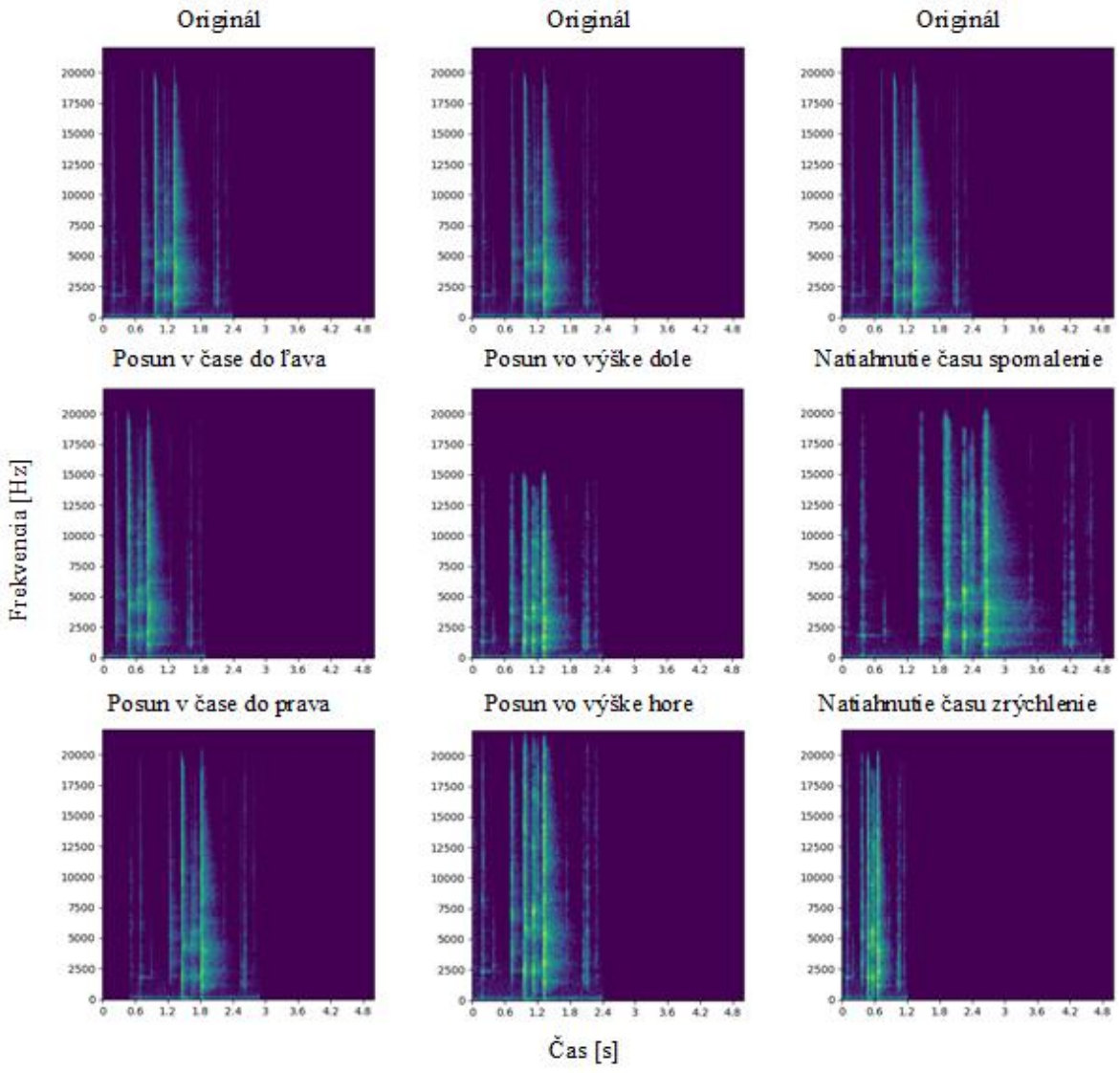
4.4.4 Augmentácia dát

Počas analýzy postupov klasifikácie sme zistili, že implementovanie jednoduchých techník augmentácie dát môže mať za následok zlepšenie presnosti klasifikácie, resp. fungovať ako regulátor zabraňujúci preučeniu konvolučnej siete. Pomocou augmentácie dát je možné synteticky vygenerovať nové anotované dáta z už existujúcich, a tým tak efektívne rozšíriť trénovaciu množinu dát. Jednoduchú formu augmentácie dát je možné vykonať pomocou miernej modifikácie vstupných dát. Existuje viacero kategórií augmentácie dát, zväčša s ohľadom na signál, na ktorý sú tieto techniky aplikované, čiže na akustický signál, na vytvorené spektrogramy, miešanie viacerých nahrávok a ďalšie. My sme aplikovali modifikácie na čistý akustický signál pred transformáciou na vstupné reprezentácie dát, v našom prípade pred aplikovaním STFT.

Bežné techniky, ktoré sú aplikované priamo na akustický signál:

- Natiahnutie v čase – pomocou tejto techniky upravujeme temporálnu charakteristiku akustického signálu (zrýchlenie alebo spomalenie) bez úpravy jeho spektrálnej charakteristiky.
- Posun po časovej osi – táto technika spôsobí, že užitočný zvuk je posunutý po časovej osi bez zmien jeho temporálnej alebo spektrálnej charakteristiky. Táto technika je efektívne aplikovaná použitím prekrytia analyzačných rámcov.
- Posun výšky tónu – použitím tejto techniky zachovávame temporálnu charakteristiku signálu a manipulujeme jeho spektrálnu charakteristiku (zvýšenie alebo zníženie), je to v podstate opak natiahnutia v čase.

Na obrázku 25 je možné vidieť ukážky týchto techník augmentácie dát. V rámci nastavenia jednotlivých koeficientov, v prípade „natiahnutia v čase“, je to faktor zrýchlenia, resp. spomalenia; v prípade „posun výšky tónu“ je to hodnota v poltónoch, o ktorú má byť akustický signál posunutý, využili sme prácu J. Salamon a J.P. Bello, ktorých štúdia [72] analyzovala vplyv augmentácie dát na rozpoznávanie environmentálnych zvukov. A teda pre techniku „natiahnutie v čase“ sú to faktory {0.81, 0.93, 1.07, 1.23} a pre techniku „posun výšky tónu“ je to posun o hodnoty v poltónoch {-3.5, -2.5, -2, -1, 1, 2, 2.5, 3.5}, čiže, ako je možné vidieť, z každej originálnej nahrávky je vytvorených 12 syntetických.



Obrázok 25 Demonštrácia techník augmentácie dát na nahrávke zvuku otvárania plechovky. Obrázok ukazuje spektrogramy pred a po aplikovaní jednotlivých techník. Parametre sú prehnané, aby bol efekt zobrazený jasne

Uvažovali sme o implementovaní augmentácie dát SpecAugment [73], ktorá modifikuje priamo na spektrogram a pozostáva z aplikácie frekvenčných a časových masiek, avšak

keďže my mapujeme spektrogram do RGB zobrazenia, spôsobovalo by to „výpadky vo farbe“, kedy by maskovaná frekvencia bola prekrytá frekvenciami zo zvyšných frekvenčných pásiem.

Pomocou experimentálneho merania sme zisťovali, akú priemernú presnosť rozpoznávania dosiahne EffNet, keď aplikujeme augmentáciu dát.

Parametre extrakcie príznakov:

- podvzorkovanie na 16 000 Hz,
- veľkosť analyzačného okna: 760 ms,
- prekrytie analyzačných okien: 65%,
- dĺžka STFT: 512 vzoriek s prekrytím 50%,
- váhová funkcia: Hanning,
- ticho bolo vyplnené užitočným zvukom.

Boli vykonané prvé zmeny v architektúre EffNetu. Pred plne-prepojenú vrstvu bola pridaná vrstva výpadku (angl. Dropout) s hodnotou $p = 0{,}5$, ktorá počas trénovania náhodne, s pravdepodobnosťou p , nastaví niektoré elementy vstupného vektoru na nulu. Zároveň boli pridané na koniec každého EffNet bloku vrstvy dvojrozmerného výpadku (angl. Dropout2D), ktoré fungujú obdobne ako jej jednorozmerná verzia, akurát je vynulovaný kompletný kanál (celá mapa príznakov) [61]. Tieto vrstvy boli taktiež nastavené na hodnotu $p = 0{,}5$. Táto regularizačná technika sa využíva, aby sa obmedzilo preučenie siete. Hlavnou myšlienkou je, že zavedením šumu do výstupných hodnôt vrstvy môže rozdeliť vzory náhodných udalostí, ktoré nie sú významné a ktoré si sieť začne pamätať, ak nie je prítomný žiaden šum [44].

Nastavenie parametrov trénovania:

- počet epoch: 15,
- veľkosť dávky: 32,
- optimalizátor: Stochastický gradientový zostup (z angl. Stochastic Gradient descent, SGD) s použitím Nestrovho momentu 0.9,
- parameter učenia (angl. learning rate): 0,005,
- stratová funkcia: Negatívna logaritmická vierohodnosť (z angl. Negative log likelihood loss, NLLLoss),
- zoslabovanie váh: 0,001.

Takto pozmenený EffNet bol natrénovaný 5-krát, využili sme všetky techniky augmentácie dát a výsledná priemerná presnosť rozpoznávania bola 50,44%.

Z hľadiska parametrov, má EffNet v tejto konfigurácii:

- veľkosť vstupného tenzora: 3x86x48,
- trénovateľných parametrov: 878 226,
- veľkosť parametrov: 3,51 MB,
- celkový počet násobenie-sčítanie operácií: 22,84 $*10^6$.

Ako môžeme vidieť parametre v aktuálnej konfigurácii sú nižšie ako tie v Piczakovom modeli, zároveň však nedosahujeme jeho presnosť rozpoznávania.

4.5 Zmeny trénovacieho procesu

Z rámci optimalizácie trénovania procesu sme experimentovali s viacerými nastaveniami.

Začiatočné hodnoty synaptických váh majú signifikantný efekt na proces trénovania. Tieto hodnoty by mali byť volené náhodne, ale zároveň podľa určených pravidiel, aby sa zabránilo stavom ako sú miznúci gradient alebo explodujúci gradient. Ak nastane niektorý z týchto stavov, gradient stratovej funkcie bude príliš malý alebo naopak veľký, aby spätná propagácia bola užitočná a konvergencia umelej neurónovej siete bude trvať dlho, resp. vôbec nenastane. Z tohto hľadiska bolo vytvorených niekoľko stratégií náhodného výberu hodnôt. Ako aktivačná funkcia je používaná Leaky ReLU, pri ktorej problém miznúceho gradientu nenastane, rovnako ako pri použití ReLU. Pôvodne sme pre všetky vrstvy využívali stratégiu, ktorú navrhol LeCun et al. [77], ktorá je v knižnici PyTorch predvolená. Pri nej sú náhodné hodnoty vyberané z rozloženia, ktoré má priemer nula a smerodajnú odchýlku

$$\sigma_w = \frac{1}{\sqrt{m}} \tag{49}$$

kde m je počet synaptických prepojení, ktoré vstupujú do uzla. Neskôr sme pre konvolučné vrstvy implementovali stratégiu inicializácie váh, ktorú navrhol He et al. [78], ktorá bola navrhnutá s ohľadom na použitie aktivačných funkcií ReLU, resp. Leaky ReLU, ktorá je vo svojej podstate modifikáciou stratégie LeCun. Rovnako ako pri LeCun, sú náhodné hodnoty vyberané z rozloženia, ktoré má priemer nula, avšak pre výpočet smerodajnej odchýlky je použitý nasledujúci vzorec:

$$\sigma_w = \frac{a}{\sqrt{m}} \tag{50}$$

kde m je počet synaptických prepojení, ktoré vstupujú, resp. vystupujú z uzla, v závislosti od použitého módu. Hodnota a je závislá od aktivačnej funkcie pre ReLU je to $\sqrt{2}$, v prípade nami používanej Leaky ReLU je to $\sqrt{\frac{2}{1+negatívny\_sklon^2}}$. Bias je inicializovaný na nulu. V prípade Normalizačných vrstiev sme použili inicializáciu na konštantnú jednotku a bias na nulu, nakoľko pri tejto inicializácií mala stratová funkcia tendenciu klesať rýchlejšie.

Aby sme zabránili tomu, že hodnoty synaptických váh nadobudnú príliš vysokých hodnôt a taktiež, aby sme pomohli regulovať preučenie siete, využili sme techniku zoslabovania váh. V rámci tejto techniky pridávame malú penalizáciu, zvyčajne L2 norma (Eukleidivská norma) synaptických váh k stratovej funkcii. Hodnota zoslabovania váh bola nastavená na hodnotu $5 * 10^{-4}$.

Ďalším dôležitým parametrom, ktorý ma veľký vplyv na proces trénovania, je parameter učenia. Zistili sme že naša konvolučná sieť sa najlepšie trénuje s parametrom učenia $2{,}5 * 10^{-4}$. To však bolo predtým, čo sme odhalili chybu, ktorá vznikla potom, čo sme zmenili framework TensorFlow za PyTorch. Ako bolo spomenuté v popise siete využívame Normalizačný vrstvu v jednotlivých konvolučných blokoch. Funkčnosť tejto vrstvy je možné ovplyvniť pomocou parametra momentum, ktorý ovplyvňuje zotrvačnosť kĺzavého priemeru. V oboch frameworkoch je tento parameter označený rovnako, avšak výpočet sa líši, ako môžeme vidieť v nasledujúcich definíciách, prvá je pre PyTorch [87] a druhá pre TensorFlow [88].

$$\begin{aligned} \hat{x}_n &= (1 - momentum) \times \hat{x} + momentum \times x_n \\ \hat{x}_n &= momentum \times \hat{x} + (1 - momentum) \times x_n \end{aligned} \tag{51}$$

kde $\hat{x}_n$ je nový kĺzavý priemer, $\hat{x}$ je kĺzavý priemer a x_n je nová pozorovaná hodnota. Ako si môžeme povšimnúť parameter momentum ovplyvňuje, akým dielom nová hodnota prispieva ku kĺzavému priemeru, avšak jednotlivé frameworky k tomu pristupujú naopak. Keďže sme preniesli naše pôvodné nastavenie z TensorFlow do Pytorch, toto spôsobilo, že nová hodnota kĺzavého priemeru bola počítaná z 0.1 krát kĺzavého priemeru a 0.9 krát nová pozorovaná hodnota, čo je presne naopak, ako sme to zamýšľali. To malo za následok, že sme museli nastaviť nižší parameter učenia a zvýšiť počet epoch trénovania na 100. Po napravení tejto chyby sme mohli určiť vyšší parameter učenia a znížiť počet epoch

trénovania na 40 tým, že v rámci procesu trénovania bol uchovávaný model, ktorý vykazoval najvyššiu presnosť rozpoznávania. Zároveň sme zvýšili veľkosťou dávky 64. V rámci určenia nového parametru učenia sme využili nastroj z fastai, pomocou ktorého si môžeme vykresliť graf vzťahu medzi parametrom určenia a straty, ktorý nám ukázal, že ideálny štart hľadania je v oblasti 10^{-2} . V rámci ďalšieho empirického testovania sme hodnotu parametru učenia určili ako $5 * 10^{-3}$.

Zároveň sme sa rozhodli aplikovať dve techniky plánovania zmeny parametru učenia, ktoré napomáhajú trénovaniu a to je exponenciálne zoslabovanie parametru učenia. Pri tejto technike je po každej epoche trénovania parameter učenia vynásobený parametrom γ, v našom prípade sme zvolili hodnotu zoslabovania $\gamma = 0{,}985$.

Druhá implementovaná technika je tzv. „ohrievanie“ (z angl. warm up) parametru učenia. Princíp „ohrievania“ je v tom, že je zvolená nízka hodnota parametru učenia a tá je následne zvyšovaná pokiaľ nedosiahne požadovanú hodnotu. Benefity tejto techniky boli demonštrované vo viacerých aplikáciách strojového učenia. Naša stratégia zmeny parametru učenia je teda nasledovná:

- parameter učenia je nastavený na $5 * 10^{-4}$,
- následne je lineárne „zohrievaný“ po 5 epoch, až kým nedosiahne hodnotu $5 * 10^{-3}$,
- po zvyšok trénovania je hodnota parametru učenia exponenciálne oslabovaná hodnotou $\gamma = 0{,}985$.

V rámci hľadania optimálneho nastavenia procesu trénovania sme otestovali nahradenie aktivačnej funkcie Leaky ReLu za ReLU, samozrejme s adekvátnou zmenou inicializácie váh korešpondujúcou s touto aktivačnou funkciou. Táto zmena však mala za následok pokles presnosti rozpoznávania, preto sme sa rozhodli ponechať aktivačnú funkciu Leaky ReLU. Ďalej sme testovali rôzne nastavenia úrovne negatívneho sklonu. Toto nastavenie nemalo signifikantný vplyv na presnosť rozpoznávania, preto sme úroveň negatívneho sklonu ponechali na hodnote 0,3.

Rozhodli sme sa otestovať optimalizátor Adam v základnom nastavení, ako ho definuje PyTorch, teda $\beta_1 = 0.9$ a $\beta_2 = 0.999$. Výsledná presnosť rozpoznávania sa príliš signifikantne nelíšila, preto sme sa aj naďalej rozhodli používať optimalizátor SGD s Nestrovím momentom.

4.6 Úpravy extrakcie príznakov

V rámci týchto zmien boli mierne upravené aj všeobecne nastavenia extrakcie príznakov. Bola implementovaná technika samotného orezávania ticha, bez tapetovania užitočným zvukom z dôvodov vyššie spomenutých.

Pri aplikácii STFT sme začali využívať váhovú funkciu Blackman-Harris, nakoľko poskytuje primeranú šírku pásma a veľmi nízke presakovanie spektra. V rámci zisťovania efektu dĺžky STFT, resp. zvoleného temporálneho rozlíšenia štúdie ukazujú, že tento parameter je viazaný na architektúru siete. Napríklad v štúdii [74] je ukázané, že AlexNet dosahuje lepšie výsledky pri temporálnom rozlíšení 30 ms a GoogLeNet zase pri temporálnom rozlíšení 40 ms. Na základe nášho testovania sme zistili, že naša konvolučná sieť dosahuje najlepších výsledkov pri temporálnom rozlíšení 45 ms, pri podvzorkovaní na 16 000 Hz. Zároveň sme dĺžku STFT začali počítať podľa vzorca (26). Uvedomujeme si, že funkcia STFT v knižnici librosa nie je optimalizovaná podľa tohto vzorca, avšak nepozorovali sme negatíva, naopak keďže podľa tohto vzorca je veľkosť váhového okna rovná dĺžke STFT, nemusíme dopĺňať váhové okno o nulové hodnoty a výsledok STFT je tak nižších rozmerov. Dĺžka STFT tak je 720 vzoriek namiesto 1024 ako by mala byť. Keďže sme použili váhovú funkciu Blackman-Harris, prekrytie váhových okien bolo nastavené na 66,1%. Na základe týchto nastavení sa nám veľkosť vstupného tenzora zmenila na 3x120x51.

Ďalej sme sa rozhodli otestovať online augmentáciu dát. Pre implementáciu online augmentácie dát bola nutná zmena stratégie predspracovania dát a extrakciu príznakov nasledovným spôsobom: fáza prípravy dát pozostávala len z podvzorkovania, orezania ticha a rozdelenia audio signálu na príslušné analyzačné okná. Počítanie spektrogramov a ich následná reformácia do RGB zobrazenia, bola vykonávaná priamo počas procesu učenia, nie vopred, ako tomu bolo pri offline augmentácii dát, kedy sú všetky tieto úkony vykonané ešte pred začiatkom procesu učenia. Vďaka tomu sme mohli implementovať aj online augmentáciu dát, ktorá bola vykonávaná s určitou pravdepodobnosťou. Teda každá vzorka mala určitú pravdepodobnosť, že na ňu bude aplikovaná niektorá z augmentácií. Myšlienka za týmto postupom je, aby sa dáta neustále menili, resp. aby sme model počas trénovania vystavili čo najväčšej rozmanitosti dát. Rozhodli sme sa aplikovať štyri druhy augmentácie dát, z ktorých bude použitá najviac jedna:

- posun výšky tónu,

- obrátenie časovej osi,
- zmena hlasitosti,
- pridanie farebného šumu.

Z hľadiska pravdepodobnosti bola určená pravdepodobnosť 50%, že bude daná vzorka augmentovaná; v prípade, že áno, pravdepodobnosť jednotlivých augmentácií bola rozložená rovnomerne. Je teda možné, že jedna vzorka bude počas jednej epochy augmentovaná a počas druhej už nie, čo by malo prispieť k robustnosti nášho modelu a následnému zlepšeniu presnosti rozpoznávania.

Naša hypotéza sa však nepotvrdila. Takto natrénovaná sieť dosahovala obdobných výsledkov ako v prípade využitia offline augmentácie dát. Zároveň určovanie pravdepodobnosti augmentácie dát pre každú vzorku samostatne predĺžilo proces trénovania. Z tohto dôvodu sme sa rozhodli ponechať offline augmentáciu dát a extrakciu príznakov.

4.7 Zmeny v architektúre EffNet

Ako už bolo vyššie spomínané pridali sme do architektúry EffNet vrstvy výpadku. V tejto časti budú popísané ďalšie zmeny vykonané na tejto architektúre, aby sme docielili vyššieho rozpoznávania a ďalej znížili počet parametrov.

Pôvodný EffNet patril medzi tzv. konvenčné konvolučné siete, teda konvolučné siete, ktoré vykonávajú konvolúciu v nižších vrstvách a pre klasifikáciu sú výstupné mapy príznakov poslednej konvolučnej vrstvy vektorizované a následne spracované plne-prepojenou vrstvou. Takáto štruktúra premosťuje konvolučnú štruktúru s tradičnými klasifikátormi na báze neurónovej siete. Avšak plne-prepojené vrstvy majú tendenciu sa preučiť, a zároveň sú „drahé“ z pohľadu parametrov, keďže jej veľkosť rastie spolu so vstupom. Preto sme sa rozhodli implementovať za poslednou konvolučnou vrstvou vrstvu globálneho združovania podľa priemeru (z angl. Global Average Pooling). Jednou z výhod tejto vrstvy je, že vynucuje zhody medzi mapami príznakov a kategóriami, čiže mapy príznakov môžu byť ľahšie interpretované ako mapy istoty kategórie [75]. Ďalšou výhodou je, že rapídne znižujú veľkosť plne-prepojenej vrstvy a zároveň túto veľkosť fixujú na jednu hodnotu, čím prestáva byť táto vrstva závislá od veľkosti vstupu.

Uvažovali sme o nahradení priestorovo oddeliteľnej konvolúcie iným typom bloku, napríklad hĺbkovo oddeliteľnou konvolúciou, nakoľko pri použití priestorovo oddeliteľnej

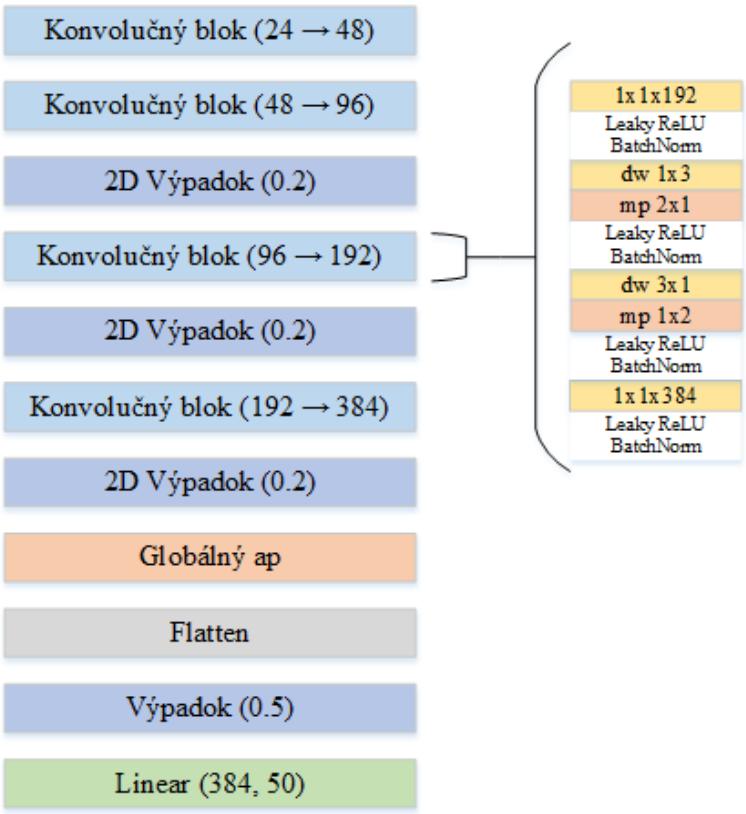
konvolúcie je nutná špecifická symetria jadra, aby ho bolo možné rozdeliť, čo má za následok obmedzené množstvo jadier, ktoré je možné použiť. V prípade hĺbkovo oddeliteľnej konvolúcie takáto požiadavka nevzniká. Avšak priestorovo oddeliteľná konvolúcia dovoľuje využiť združovanie podľa maxima medzi jednotlivými vrstvami. To má za následok zníženie celkového počtu parametrov, čo je v súlade s našimi cieľmi, preto sme sa rozhodli ponechať tento typ konvolučného bloku.

Následne sme nahradili poslednú konvolučnú vrstvu v bloku za bodovú konvolučnú vrstvu a pridali sme vrstvu združovania podľa maxima 1x2 s príslušným krokom, čím sme ďalej redukovali počet trénovateľných parametrov. Avšak to malo za následok, že sieť stratila schopnosť sa doučovať. Preto sme sa rozhodli znovu reštrukturalizovať výpadové vrstvy, nakoľko ich hodnota pravdepodobnosti výpadu bola príliš vysoká pre tak malú sieť. Na základe experimentov sme určili, že po prvom bloku nie je výpadová vrstva vhodná vôbec. Dvojrozmerné výpadové vrstvy, ktoré nasledovali zvyšné bloky, mali zníženú hodnotu pravdepodobnosti výpadu $p = 0{,}2$. Vo výpadovej vrstve pred plne-prepojenou vrstvou zostalo ponechané nastavenie pravdepodobnosti výpadu $p = 0{,}5$. Taktiež bol v rámci trénovania znížený parameter zoslabovania váh na hodnotu $1*10^{-4}$. Po týchto zmenách bola sieť znovu schopná sa doučiť.

Skúšali sme nahradiť vrstvu normalizácie dávky (angl. BatchNorm) za vrstvu renormalizácie dávky (angl. BatchRenorm) podľa článku [76], avšak neprinieslo to poznateľné zlepšenie presnosti rozpoznávania, preto sme sa rozhodli ponechať klasickú vrstvu normalizácie dávky, zároveň však využitie tejto vrstvy spôsobilo predĺženie procesu trénovania.

Ďalej sme sa rozhodli zmeniť počty kanálov jednotlivých blokov. Prvotne sme len pridali blok s počtom kanálov 512, resp. 1024, avšak tieto nastavenia neposkytli tak významné zlepšenie presnosti rozpoznávania, aby to ospravedlnilo zvýšenie počtu parametrov. Rozhodli sme sa preto reštrukturalizovať počty kanálov v celej konvolučnej sieti. Konečná architektúra našej siete je zobrazená na obrázku 26.

Vykonané zmeny považujeme za natoľko významné, že túto sieť už nepovažujeme za EffNet, ale za novú konvolučnú neurónovú sieť.



Obrázok 26 Architektúra našej konvolučnej neurónovej siete s detailom konvolučného bloku. „dw“ znamená hĺbková konvolúcia, „mp“ znamená združovanie podľa maxima a „ap“ združovanie podľa priemeru.

V tejto konfigurácii má naša konvolučná sieť nasledujúce parametre:

- veľkosť vstupného tenzora: 3x120x51,
- trénovateľných parametrov: 171 386,
- veľkosť parametrov: 0,69 MB,
- celkový počet násobenie-sčítanie operácií: 18,37 *10^6.

Pri porovnaní s Piczakovým modelom zistíme, že naša konvolučná sieť operuje len s 0,65% trénovateľných parametrov oproti Piczakovmu modelu. Vo vyššie zobrazenej konfigurácii má táto konvolučná sieť nasledujúce výsledky:

- Priemerná presnosť (angl. accuracy): 57,96%
- Precision: 59,49%
- Recall: 57,78%

Najkorektnejšie klasifikovaná trieda bola „budík“ a za ňou nasledovali „kostolné zvony“, „rozbitie skla“ a „ručná píla“. Naopak, najväčší problém pre sieť predstavovala trieda „vietor“ a blízko pri nej boli „kýchnutie“, „pitie, popíjanie“ a „ovca“.

4.8 Krížová validácia

Avšak tieto výsledky boli dosiahnuté pri rozdelení datasetu ESC-50 nekorektným spôsobom. Všetky doterajšie merania presnosti boli vykonané pri rozdelení datasetu na trénovacie, validačné a testovacie dáta. To však spôsobilo redukciu množiny dát pre trénovanie. Samotný autor tejto množiny dát zoradil nahrávky do piatich rovnomerných množín pre krížovú validáciu tak, aby nahrávky, ktoré pochádzajú z jedného zdrojového súboru boli vždy obsiahnuté v jednej množine [26]. Z tohto hľadiska sme sa rozhodli množinu dát nemiešať pred rozdelením na jednotlivé diely.

Preto sme sa aj mi rozhodli implementovať krížovú validáciu, nakoľko nám to ponúkne korektné porovnanie presnosti rozpoznávania s referenčným modelom. Zároveň je táto technika validácie vhodná pre menšie množiny dát, nakoľko táto je rozdelená v jednu dobu len na dve časti: trénovaciu a validačnú. Ďalšou výhodou je pomerne presný odhad klasifikačnej presnosti, ale za cenu časovej náročnosti, nakoľko treba model natrénovať viackrát.

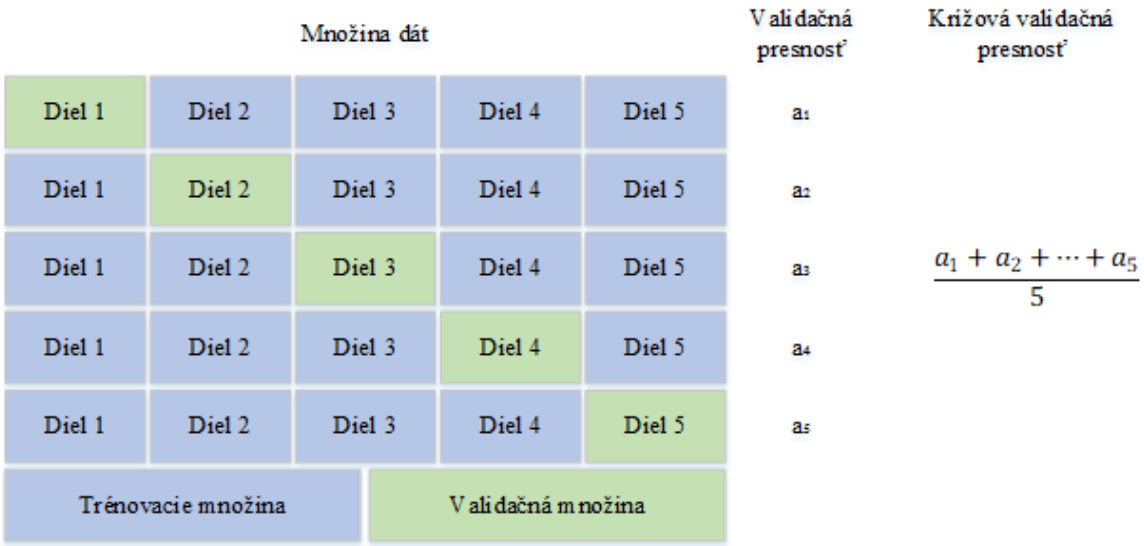
Konkrétny typ krížovej validácie, ktorý využívame sa nazýva *k*-násobná krížová validácia, kde k je celé číslo, ktoré predstavuje počet dielov, na ktoré bude množina dát rozdelená. Typické hodnoty k sú päť a desať. Teda množina dát je rozdelená na k dielov, jeden z nich je delegovaný ako validačný a zvyšných $k - 1$ sú delegované ako množina trénovacích dát, na ktorých je následne model natrénovaný. Výsledná presnosť rozpoznávania, resp. validačná presnosť je zaznamenaná a proces krížovej validácie pokračuje. Ďalší diel v poradí je delegovaný ako validačný a proces trénovania je znovu zahájený. Je vhodné poznamenať, že pred každým procesom trénovania je model nanovo inicializovaný. Tento proces je zopakovaný k krát, takže každý diel je delegovaný ako validačný práve jeden krát. Výsledkom *k*-násobnej krížovej validácie, označovaný tiež ako krížovo-validačná presnosť, je priemerná hodnota presnosti rozpoznávania, vypočítaná z k validačných presností. Ilustráciu tohto procesu je možné vidieť na obrázku 27, ktorý ukazuje *k*-násobnú krížovú validáciu pre $k = 5$.

Pre určenie presnosti rozpoznávania našej siete, sme náš model natrénovali päťkrát pre každý diel a výslednú priemernú hodnotu pre každý diel sme použili pre výpočet výslednej krížovo-validačnej presnosti rozpoznávania. Výsledné priemerné presnosti rozpoznávania pre jednotlivé diely ako aj výslednú krížovo-validačnú presnosť je možné visieť v tabuľke 3.

Pridali sme taktiež metriku Top-5, ktorá predstavuje presnosť toho, koľkokrát je cieľové označenie v piatich najvyšších pravdepodobnostiach predikcie označenia našej siete. Nami doteraz používaná presnosť rozpoznávania je v podstate Top-1 metrika, ktorá predstavuje presnosť toho, koľkokrát je cieľové označenie najvyššia pravdepodobnosť predikcie označenia.

Validačný diel	Validačná presnosť Top-1	Smerodajná odchýlka Top-1	Validačná presnosť Top-5	Smerodajná odchýlka Top-5
Diel 1	59,87 %	0,66 %	84,83 %	0,26 %
Diel 2	59,46 %	0,75 %	85,36 %	0,69 %
Diel 3	62,56 %	0,81 %	85,62 %	0,36 %
Diel 4	66,06 %	0,67 %	88,70 %	0,24 %
Diel 5	59,46 %	0,71 %	85,52 %	0,82 %
Krížovo-validačná	61,48 %	2,66 %	86,01 %	1,47 %

Tabuľka 3 Výsledné presnosti rozpoznávania pri použití 5-násobnej krížovej validácie



Obrázok 27 Ilustrácia *k*-násobnej krížovej validácie, pre $k = 5$

4.9 Prenášané učenie

Keďže veľkosť nami používaného datasetu je relatívne malá, rozhodli sme sa využiť princíp tzv. prenášaného učenia (z angl. transfer learning). Hlavnou motiváciou pre implementáciu tejto metódy však bolo to, že vo viacerých štúdiách tento prístup napomáhal zvýšeniu presnosti rozpoznávania klasifikačného modelu.

Na základe prieskumu, ktorý vykonal S.J. Pan a Q. Yang [91], môžeme definovať prenášané učenie ako: Vylepšenie učenia cieľovej prediktívnej funkcie f_T pre cieľovú úlohu T_T v cieľovej oblasti D_T, s použitím vedomostí zo zdrojovej oblasti D_S a zdrojovej úlohy T_S, kde platí $D_S \neq D_T$ alebo $T_S \neq T_T$. Na základe [91] prenášané učenie môže byť rozdelené na tri kategórie:

- Induktívne prenášané učenie – používa sa, keď cieľová úloha T_T je odlišná od zdrojovej úlohy T_S, anotované dáta cieľovej oblasti sú dostupné bez ohľadu na to, či sú dostupné anotované dáta v zdrojovej oblasti.
- Transduktívne prenášané učenie – používa sa, keď cieľová úloha T_T je zhodná so zdrojovou úlohou T_S, anotované dáta v cieľovej oblasti nie sú dostupné a zdrojová oblasť D_S a cieľová oblasť D_T sú odlišné.
- Prenášané učenie bez učiteľa – používa sa, keď cieľová úloha T_T je odlišná od zdrojovej úlohy T_S, a nie sú dostupné anotované dáta v cieľovej ani zdrojovej oblasti.

V tomto ohľade budeme využívať v rámci našej práce induktívne prenášané učenie, keďže sme sa rozhodli využiť rozsiahlu množinu dát ImageNet, konkrétne ImageNet z roku 2012, ktorý sa sústreďoval na rozpoznávanie objektov. Táto obrovská množina dát pozostáva z viac než milióna trénovacích vzoriek rozdelených do 1 000 tried.

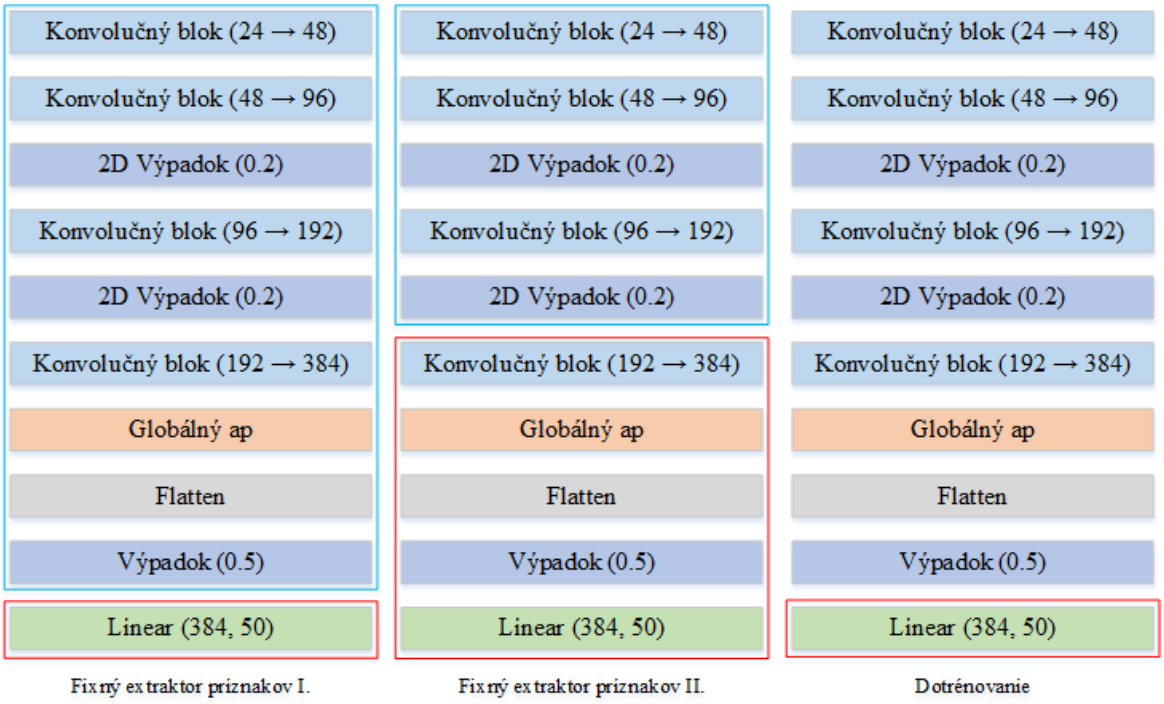
V rámci fázy predtrénovania sme trénovali náš klasifikačný model na tejto množine dát dvakrát a pre ďalšie spracovávanie sme použili ten najlepší model z nich. Toto obmedzené množstvo bolo preto, že jedno takého predtrénovanie trvalo vyše týždňa a to z dôvodu veľkosti množiny dát ako aj obmedzenej dostupnej výpočtovej sily. Zároveň sa však výsledné presnosti rozpoznávania príliš nelíšili a dosiahli približne 30% na testovacej množine. Uvedomujeme si, že táto hodnota nie je vysoká, ale ak vezmeme v úvahu nízky počet parametrov našej siete a fakt, že sa jedná o predtrénovanie, je tento výsledok dostatočný.

V rámci analýzy sme zistili, že existuje viacero prístupov k prenášanému učeniu. Tieto druhy zväčša závisia od veľkosti použitej množiny dát, ako aj od podobnosti úloh. Na obrázku 28 môžeme vidieť nami skúmané prístupy. Konkrétne sú to tri prístupy.

Fixný extraktor príznakov I., kedy je posledná plne-prepojená vrstva inicializovaná nanovo, nakoľko sa nám líši počet cieľových tried a zvyšok siete má zmrazené parametre, resp. majú zastavenú schopnosť učenia. Počas procesu učenia je teda trénovaná len finálna vrstva. Tento

prístup je vhodný pre podobné úlohy a v našom prípade takto natrénovaná sieť dosahovala len obmedzené výsledky, čo sme však predpokladali, nakoľko tento prístup je vhodný pre veľmi podobné úlohy, resp. rovnaké úlohy.

Druhý prístup je voľnejší fixný extraktor príznakov II. V rámci tohto prístupu sú zmrazené parametre začiatočných vrstiev, ktoré sú zodpovedné za extrakciu jednoduchých tvarov ako sú napríklad hrany. Nanovo inicializované sú plne prepojená vrstva a posledný konvolučný blok, ktorý zvyčajne extrahuje abstraktné tvary špecifické pre danú úlohu. Tento prístup je vhodný pre stredne veľké množiny dát. V rámci testovania tohto prístupu sme natrénovali náš klasifikačný model pomocou 5 násobnej krížovej validácie päť krát. Výsledná krížovo-validačná presnosť rozpoznávania bola 55,21 % so smerodajnou odchýlkou 1,63 %, metrika Top-5 pre tento prístup predstavovala 82,93 % so smerodajnou odchýlkou 0,64 %. Ako si môžeme povšimnúť táto presnosť rozpoznávania je významne nižšia ako pri čistom trénovaní, teda ak je celý model inicializovaný nanovo. Tento prístup sa teda v prípade našej práce ukázal ako nevhodný.



Obrázok 28 Ukážka prístupov k prenášanému učeniu. Červené bloky predstavujú časti, ktoré sú nanovo inicializované; modré bloky predstavujú časti, ktorých parametre boli zmrazené.

Posledný prístup, ktorý sme skúmali je tzn. dotrénovanie (z angl. fine-tune). Pri využití tohto prístupu je nanovo inicializovaná plne-prepojená vrstva a zvyšok modelu je inicializovaný pomocou predtrénovaného modelu, avšak žiadne parametre nie sú zmrazené, teda počas procesu trénovania si ponechajú schopnosť učiť sa. Takto inicializovaný model je následne

trénovaný podľa zvolenej stratégie. Takto nastavený klasifikačný model sme znova trénovali pomocou využití 5 násobnej krížovej validácie 5 krát. Zistili sme, že klasifikačný model, ktorý bol inicializovaný týmto spôsobom dosahoval lepších výsledkov presnosti rozpoznávania ako v prípade náhodnej inicializácie, ktorú sme používali v predchádzajúcich testovaniach. Výslednú krížovo-validačnú presnosť rozpoznávania, výsledky pre jednotlivé validačné diely ako aj Top-5 presnosť je možné vidieť v tabuľke 4.

Validačný diel	Validačná presnosť Top-1	Smerodajná odchýlka Top-1	Validačná presnosť Top-5	Smerodajná odchýlka Top-5
Diel 1	59,47 %	0,15 %	86,10 %	0,20 %
Diel 2	61,22 %	0,51 %	85,94 %	0,73 %
Diel 3	63,18 %	0,68 %	86,65 %	0,17 %
Diel 4	67,96 %	0,34 %	88,97 %	0,22 %
Diel 5	59,20 %	0,52 %	84,70 %	0,32 %
Krížovo-validačná	62,21 %	3,25 %	86,47 %	1,45 %

Tabuľka 4 Výsledné presnosti rozpoznávania. Trénovanie s využitím prenášaného učenia – prístup dotrénovanie

Ako je možné si povšimnúť krížovo-validačná presnosť rozpoznávania modelu, ktorý bol inicializovaný pomocou hodnôt synaptických váh získaných z predtrénovania na množine dát ImageNet dosahuje vyšších hodnôt ako v prípade náhodnej inicializácie.

Ak teda porovnáme našu konvolučnú neurónovú sieť so zvolenou referenčnou sieťou, z hľadiska parametrov sa situácia nezmenila, teda naša sieť stále klasifikuje s použitím 0,65% veľkosti referenčného modelu, čo sa týka počtu trénovateľných parametrov a z toho vyplývajúca veľkosť modelu. Náš klasifikačný model dosahuje krízovo-validačnej presnosti rozpoznávania 62,21%, referenčný model dosahuje presnosť rozpoznávania 64,5%, ako teda môžeme vidieť náš klasifikačný model stráca 2,29% presnosti rozpoznávania oproti referenčnému. Avšak náš klasifikačný model je schopný klasifikovať na základe 760 milisekúnd akustického signálu, naproti tomu referenčný model vyžaduje 5 sekúnd.

Detailné porovnanie nášho klasifikačného modelu s referenčnou model K. Piczaka je možné vidieť v tabuľke 5.

	Referenčný (Piczakov) model	Náš klasifikačný model
Potrebná dĺžka akustického signálu	5 s	0.73 s
Veľkosť vstupného tenzora	$2 \times 60 \times 41$	$3 \times 120 \times 51$
Trénovateľných parametrov	26 534 130	171 386
Veľkosť parametrov	106,14 MB	0,69 MB
Celkový počet násobenie-sčítanie operácií	$34{,}54 \cdot 10^6$	$18{,}37 \cdot 10^6$
Presnosť rozpoznávania	64,5 %	62,21 %
Top-5 presnosť rozpoznávania	-	86,47 %

Tabuľka 5 Porovnanie klasifikačného modelu s referenčným

Záver

Táto práca sa zaoberá návrhom klasifikačného modelu pre klasifikáciu environmentálnych zvukov, tento model by mal byť založený na metódach strojového učenia. Hlavnou motiváciou, za vývojom tohto klasifikačného modelu, bolo jeho možné budúce využitie pre systém ochrany lesov a to pred nelegálnou ťažbou alebo nepovoleným vstupom motorových vozidiel do lesných oblastí, resp. ako základ akustického bezpečnostného systému.

Bolo nutné vykonať analýzu týchto metód s ohľadom na ich využiteľnosť pri riešení klasifikačných problémov. Dôraz bol kladený na nízku veľkosť modelu, aby v budúcnosti bola možná implementácia na zariadenie s obmedzenou výpočtovou silou. Z tohto hľadiska sa ukázalo ako najvhodnejší prístup využitie konvolučných neurónových sietí, ktorých využitie sa osvedčilo vo viacerých prístupoch.

Ďalej bola vykonaná analýza metód extrakcie príznakov s ohľadom na ich využiteľnosť spolu s konvolučnou neurónovou sieťou, ktorá vo väčšine prístupov predpokladá dvojrozmerné vstupné dáta. Z tohto dôvodu sme sa venovali transformáciám akustického signálu do časovo-frekvenčnej oblasti.

Ako inšpiráciu počas návrhu sme zobrali efektívne rozpoznávanie obrazu a náš klasifikačný model sme postavili na obdobných princípoch. Rozpoznávanie obrazu nás taktiež inšpirovalo k návrhu metódy reformácie spektrogramu, pomocou ktorej je jednokanálový spektrogram reformovaný na trojkanálové RGB zobrazenie. V rámci experimentov sme testovali vplyv dĺžky vstupného akustického signálu na presnosť klasifikácie a taktiež úroveň prekrytia medzi jednotlivými rámcami. Na základe týchto experimentov sme určili, že vhodná dĺžka akustického signálu, ktorý vstupuje do klasifikačného procesu je 0,76 sekúnd s prekrytím medzi jednotlivými rámcami 65%. Ďalej sme experimentovali s rôznymi druhmi augmentácie dát, ako aj so spôsobmi ich aplikácie (online vs. offline augmentácia). Na základe experimentov sme určili vhodnú stratégiu trénovania nášho klasifikačného modelu, ako aj jeho korektnú evaluáciu.

Bol zvolený referenčný model, proti ktorému sme porovnávali náš klasifikačný model. Náš model operuje s 0,65% veľkosti referenčného modelu. Nami navrhnutý klasifikačný model je schopný klasifikovať akustický signál s dĺžkou 0,76 sekúnd, oproti 5 sekundám, ktoré

vyžaduje referenčný model. Avšak čo sa týka presnosti klasifikácie náš klasifikačný model stráca voči referenčnému modelu 2,29%.

Prínosom práce je najmä overenie vhodnosti použitia princípov efektívneho rozpoznávania obrazu pre návrh klasifikátora environmentálnych zvukov, samotný návrh klasifikačného modelu s nízkou veľkosťou, schopného klasifikácie na základe krátkeho akustického signálu. Ďalším prínosom je návrh metódy reformácie spektrokramu, pomocou ktorej sme redukovali veľkosť konvolučnej siete.

Aj keď je presnosť klasifikácie nášho modelu 62,21%, máme za to, že tento model je použiteľný ako základ akustického bezpečnostného systému, nakoľko aj keď priama presnosť rozpoznávania nie je vysoká, pri použití metriky Top-5 zistíme, že v najvyšších pravdepodobnostiach predikcie je cieľová skupina prítomna s presnosťou 86,47%, čo znamená, že je možná určitá kompenzácia presnosti.

Stručné zhrnutie, hlavným cieľom dizertačnej práce bol návrh klasifikačného modelu s nízkou veľkosťou, náš model má veľkosť 0,69 MB, tento bod teda rátame za splnený. Sekundárnym cieľom bolo, aby klasifikačný model pracoval s čo najmenšou vzorkou akustického signálu, ideálne do jednej sekundy; náš model klasifikuje na základe 0.76 sekundy, teda aj tento cieľ sme splnili. Záverečný cieľ bolo porovnanie nášho klasifikačného modelu s referenčným, výsledok tohto porovnania sa nachádza v tabuľke 5. Čím boli ciele dizertačnej práce splnené.

Referencie

[1] Cepoi, L., Donţu, N., Şalaru, V., & Şalaru, V. (2016). Removal of organic pollutants from wastewater by cyanobacteria. In *Cyanobacteria for bioremediation of wastewaters* (pp. 27-43). Springer, Cham.

[2] Bello, J. P., Silva, C., Nov, O., Dubois, R. L., Arora, A., Salamon, J., ... & Doraiswamy, H. (2019). Sonyc: A system for monitoring, analyzing, and mitigating urban noise pollution. *Communications of the ACM*, *62*(2), 68-77.

[3] Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., ... & Ritter, M. (2017, March). Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 776-780). IEEE.

[4] Schafer, R. M. (1977). Our Sonic Environment and the Tuning of the World: The Soundscape. *Vermont: Destiny Books Rochester*.

[5] Delage, B.: Paysage sonore urbain. Technical Report, Plan Construction, Paris (1979).

[6] Pijanowski, B. C., Farina, A., Gage, S. H., Dumyahn, S. L., & Krause, B. L. (2011). What is soundscape ecology? An introduction and overview of an emerging new science. *Landscape ecology*, *26*(9), 1213-1232.

[7] Guastavino, C. (2018). Everyday sound categorization. *Computational analysis of sound scenes and events*, 183-213.

[8] Salamon, J., Jacoby, C., & Bello, J. P. (2014, November). A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 1041-1044).

[9] Morel, J., Marquis-Favre, C., Dubois, D., & Pierrette, M. (2012). Road traffic in urban areas: A perceptual and cognitive typology of pass-by noises. *Acta acustica united with acustica*, *98*(1), 166-178.

[10] Wold, E., Blum, T., Keislar, D., & Wheaten, J. (1996). Content-based classification, search, and retrieval of audio. *IEEE multimedia*, *3*(3), 27-36.

[11] Wang, J. C., Wang, J. F., He, K. W., & Hsu, C. S. (2006, July). Environmental sound classification using hybrid SVM/KNN classifier and MPEG-7 audio low-level descriptor. In *The 2006 IEEE international joint conference on neural network proceedings* (pp. 1731-1735). IEEE.

[12] Chu, S., Narayanan, S., Kuo, C. C. J., & Mataric, M. J. (2006, July). Where am I? Scene recognition for mobile robots using audio features. In *2006 IEEE International conference on multimedia and expo* (pp. 885-888). IEEE.

[13] Bello, J. P., Mydlarz, C., & Salamon, J. (2018). Sound analysis in smart cities. In *Computational Analysis of Sound Scenes and Events* (pp. 373-397). Springer, Cham.

[14] "Dublin City Noise web." [Online]. Dostupné: http://www.dublincitynoise.com.

[15] "Sound of New York (SONYC) web." [Online]. Dostupné: http://wp.nyu.edu/sonyc.

[16] Bello, J. P., Silva, C., Nov, O., Dubois, R. L., Arora, A., Salamon, J., ... & Doraiswamy, H. (2019). Sonyc: A system for monitoring, analyzing, and mitigating urban noise pollution. *Communications of the ACM*, *62*(2), 68-77.

[17] Farrés, J. C. (2015, June). Barcelona noise monitoring network. In *Proceedings of the Euronoise* (pp. 218-220).

[18] Laiolo, P. (2010). The emerging significance of bioacoustics in animal species conservation. *Biological conservation*, *143*(7), 1635-1645.

[19] Mporas, I., Ganchev, T., Kocsis, O., Fakotakis, N., Jahn, O., Riede, K., & Schuchmann, K. L. (2012, November). Automated acoustic classification of bird species from real-field recordings. In *2012 IEEE 24th International Conference on Tools with Artificial Intelligence* (Vol. 1, pp. 778-781). IEEE.

[20] Walters, C. L., Freeman, R., Collen, A., Dietz, C., Brock Fenton, M., Jones, G., ... & Jones, K. E. (2012). A continental-scale tool for acoustic identification of E uropean bats. *Journal of Applied Ecology*, *49*(5), 1064-1074.

[21] Stowell, D. (2018). Computational bioacoustic scene analysis. In *Computational analysis of sound scenes and events* (pp. 303-333). Springer, Cham.

[22] "Audio Analytic web" [Online]. Dostupné: https://www.audioanalytic.com/

[23] Krstulović, S. (2018). Audio event recognition in the smart home. *Computational Analysis of Sound Scenes and Events*, 335-371.

[24] Kumar, D. P., Amgoth, T., & Annavarapu, C. S. R. (2019). Machine learning algorithms for wireless sensor networks: A survey. *Information Fusion*, *49*, 1-25.

[25] Aly, M. (2005). Survey on multiclass classification methods. *Neural Netw*, *19*(1-9), 2.

[26] Piczak, K. J. (2015, October). ESC: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia* (pp. 1015-1018).

[27] Warr, K. (2019). *Strengthening deep neural networks: making AI less susceptible to adversarial trickery*. O'Reilly Media.

[28] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115-133.

[29] Hebb, D. O. (1949). The first stage of perception: growth of the assembly. *The Organization of Behavior*, *4*, 60-78.

[30] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

[31] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, *79*(8), 2554-2558.

[32] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation*. California Univ San Diego La Jolla Inst for Cognitive Science.

[33] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541-551.

[34] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), 1527-1554.

[35] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.

[36] Du, K. L., & Swamy, M. N. (2006). *Neural networks in a softcomputing framework* (Vol. 501). London: Springer.

[37] Simon, H. (2009). Neural Networks and Learning Machines. Third Edition /Simon Haykin.–the USA.

[38] Sinčák, P., & Andrejková, G. (1996). Neurónové siete Inžiniersky prístup (1. diel). *Elfa: Kosice*.

[39] Šíma J., Neruda R. (1996). Teoretické otázky neuronových sítí, 1. vyd. Praha: MATFYZPRESS, Dostupne: http://www2.cs.cas.cz/~sima/kniha.pdf

[40] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

[41] Karpathy, A.: CS231n: Convolutional Neural Networks for Visual Recognition. 2017. Dostupné: http://cs231n.stanford.edu/

[42] Goodfellow, I.; Bengio, Y.; Courville, A.: Deep Learning. MIT Press, 2016, Dostupné: http://www.deeplearningbook.org.

[43] Le, Q. V. (2015). A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks. *Google Brain*, *20*, 1-20.

[44] Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster.

[45] Dumoulin, V., & Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.

[46] Rao, D., & McMahan, B. (2019). *Natural language processing with PyTorch: build intelligent language applications using deep learning*. " O'Reilly Media, Inc.".

[47] Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

[48] Ginsberg, J. H. (2018). *Acoustics: A Textbook for Engineers and Physicists* (Vol. 2, p. 698). New York: Springer.

[49] Nový. R. (2009). Hluk a Chveni. České vysoké učení technické v Praze Česká technika - nakladatelstvl ČVUT

[50] Miček, J., & Jurečka, M. (2013). Moderné prostriedky implementácie metód číslicového spracovania signálov 1. *Žilina: EDIS*.

[51] Proakis J.G., Manolakis D.G. (2007) Digital signal processing: principles, algorithms and applications, Prentice Hall, ISBN 0-13-187374-1.

[52] Heinzel, G., Rüdiger, A., & Schilling, R. (2002). Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new at-top windows.

[53] FFTW knižnica, Dostupná: http://www.fftw.org/

[54] Smith, S. W. (1999). The scientist and engineer's guide to digital signal processing. Second Edition

[55] Isaacson, E. (1989). Numerical Recipes in C: The Art of Scientific Computing (William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling); Numerical Recipes: Example Book (C)(William T. Vetterling, Saul A. Teukolsky, William H. Press, and Brian P. Flannery). *SIAM Review*, *31*(1), 142.

[56] Smith, J. O. (2007). *Mathematics of the discrete Fourier transform (DFT): with audio applications*. Julius Smith.

[57] Understanding FFTs and Windowing Dostupné: https://download.ni.com/evaluation/pxi/Understanding%20FFTs%20and%20Windowing.pdf

[58] Zimmermann, J. Spektrálna skladba segmentov rečového signálu.

[59] Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, *66*(1), 51-83.

[60] Smith, J.O. Spectral Audio Signal Processing, Dostupné: http://ccrma.stanford.edu/~jos/sasp/

[61] Dokumentácia Pytorch, Dostupné: https://pytorch.org/docs/

[62] Rabiner, L., & Schafer, R. (2010). *Theory and applications of digital speech processing*. Prentice Hall Press.

[63] Valero, X., & Alias, F. (2012). Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification. *IEEE Transactions on Multimedia*, *14*(6), 1684-1689.

[64] Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., ... & Ritter, M. (2017, March). Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 776-780). IEEE.

[65] Fonseca, E., Favory, X., Pons, J., Font, F., & Serra, X. (2020). FSD50k: an open dataset of human-labeled sound events. *arXiv preprint arXiv:2010.00475*.

[66] Font, F., Roma, G., & Serra, X. (2013, October). Freesound technical demo. In *Proceedings of the 21st ACM international conference on Multimedia* (pp. 411-412).

[67] Piczak, K. J. (2015, September). Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th international workshop on machine learning for signal processing (MLSP)* (pp. 1-6). IEEE.

[68] Chachada, S., & Kuo, C. C. J. (2014). Environmental sound recognition: A survey. *APSIPA Transactions on Signal and Information Processing*, *3*.

[69] Ma, N., Zhang, X., Zheng, H. T., & Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 116-131).

[70] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).

[71] Freeman, I., Roese-Koerner, L., & Kummert, A. (2018, October). Effnet: An efficient structure for convolutional neural networks. In *2018 25th IEEE International Conference on Image Processing (ICIP)* (pp. 6-10). IEEE.

[72] Salamon, J., & Bello, J. P. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal processing letters*, *24*(3), 279-283.

[73] Park, D. S., Chan, W., Zhang, Y., Chiu, C. C., Zoph, B., Cubuk, E. D., & Le, Q. V. (2019). Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*.

[74] Boddapati, V., Petef, A., Rasmusson, J., & Lundberg, L. (2017). Classifying environmental sounds using image recognition networks. *Procedia computer science*, *112*, 2048-2056.

[75] Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.

[76] Ioffe, S. (2017). Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. *Advances in neural information processing systems*, *30*.

[77] LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K. R. (2012). Efficient backprop. In Neural networks: Tricks of the trade (pp. 9-48). Springer, Berlin, Heidelberg.

[78] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision (pp. 1026-1034).

[79] Jacobsen, E., & Lyons, R. (2003). The sliding DFT. IEEE Signal Processing Magazine, 20(2), 74-80.

[80] Jacobsen, E., & Lyons, R. (2004). An update to the sliding DFT. IEEE Signal Processing Magazine, 21(1), 110-111.

[81] Šarafín, P. (2014). Modul pre digitalizáciu a predspracovanie akustického signálu: diplomová práca. Žilina: UNIZA, 68s.

[82] Gabor, D. (1946). Theory of communication. Part 1: The analysis of information. Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering, 93(26), 429-441.

[83] Ahmed, N., Natarajan, T., & Rao, K. R. (1974). Discrete cosine transform. IEEE transactions on Computers, 100(1), 90-93.

[84] Salomon, D. (2004). Data compression: the complete reference. Springer Science & Business Media.

[85] Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., ... & Liu, H. H. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear

and non-stationary time series analysis. Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences, 454(1971), 903-995.

[86] Karlovský V. (2016): Analýza variability slnečnej aktivity metódou EMD, Zborník referátov z 23. celoštátneho slnečného seminára, 1-8

[87] Dokumentácia PyTorch: normalizačná vrstva, dostupná https://pytorch.org/docs/stable/generated/torch.nn.BatchNorm2d.html

[88] Dokumentácia TensorFlow: normalizačná vrstva, dostupná https://www.tensorflow.org/api_docs/python/tf/keras/layers/BatchNormalization

[89] Smith, J.O. Digital Audio Resampling Home Page, http://www-ccrma.stanford.edu/~jos/resample/

[90] Dokumentácia PyTorch: prevzorkovania, dostupná https://pytorch.org/audio/stable/tutorials/audio_resampling_tutorial.html

[91] Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.

Zoznam publikácií

[1] *An overview of practices used in environmental sound classification*. M. Chochul and P. Ševčík. In: ICETA 2021 : 19th IEEE International Conference on Emerging eLearning Technologies and Applications : proceedings : 19th IEEE International Conference on Emerging eLearning Technologies and Applications : proceedings / zost. František Jakab. - 1. vyd. - Denver : Institute of Electrical and Electronics Engineers, 2021. - 441 s. [online, USB-key]. - ISBN 978-1-6654-2101-0. - s. 76-81

[2] *A survey of low power wide area network technologies*. M. Chochul and P. Ševčík. In: ICETA 2020 : 18th IEEE International conference on emerging elearning technologies and applications : Information and communication technologies in learning : proceedings : 18th IEEE International conference on emerging elearning technologies and applications : Information and communication technologies in learning : proceedings / zost. František Jakab. - 1. vyd. - Denver : Institute of Electrical and Electronics Engineers, 2020. - 789 s. [online]. - ISBN 978-0-7381-2366-0. - s. 1-5

[3] *Optical communication system for a robot in project Aeris*. M. Chochul and P. Ševčík. In: ICETA 2021 : 19th IEEE International Conference on Emerging eLearning Technologies and Applications : proceedings : 19th IEEE International Conference on Emerging eLearning Technologies and Applications : proceedings / zost. František Jakab. - 1. vyd. - Denver : Institute of Electrical and Electronics Engineers, 2021. - 441 s. [online, USB-key]. - ISBN 978-1-6654-2101-0. - s. [1-5]

[4] *Dynamic system parameter identification based on the acceleration data.* P. Šarafin, L. Formanek and M. Chochul. In: ICETA 2020 : 18th IEEE International conference on emerging elearning technologies and applications : Information and communication technologies in learning : proceedings : 18th IEEE International conference on emerging elearning technologies and applications : Information and communication technologies in learning : proceedings / zost. František Jakab. - 1. vyd. - Denver :

Institute of Electrical and Electronics Engineers, 2020. - 789 s. [online]. - ISBN 978-0-7381-2366-0. - s. [1-4]

[5] *Prediction of temperature in WSN using artificial intelligence*. L. Formanek, M. Chochul and O. Karpiš. In: Sensors and electronic instrumentation advances : proceedings of the 5th international conference on sensors and electronic instrumentation advances : proceedings of the 5th international conference on sensors and electronic instrumentation advances / S. Y. Yurish. - 1. vyd. - Barcelona : IFSA Publishing, 2019. - ISBN 978-84-09-14413-6. - s. 126-129.

[6] *Compressed Sensing and Acoustic Analysis for Use in Localization Tasks*. V. Olešnaníková, O. Karpiš, P. Šarafín, L. Formanek, M. Chochul. In: Sensors and electronic instrumentation advances [electronic] : proceedings of the 5th international conference on sensors and electronic instrumentation advances. - 1. vyd. - Barcelona: IFSA Publishing, 2019. - ISBN 978-84-09-14413-6. - s. 333-338.

[7] *Forest fire detection and localization within WSN*. M. Chochul. In: Mathematics in science and technologies : proceedings of the MIST conference 2019. - [S.l.] : [s.n.]. - ISBN 9781794002180. - s. 28-32

Zoznam citácií

[1] *A survey of low power wide area network technologies*. M. Chochul and P. Ševčík. In: ICETA 2020 : 18th IEEE International conference on emerging elearning technologies and applications : Information and communication technologies in learning : proceedings : 18th IEEE International conference on emerging elearning technologies and applications : Information and communication technologies in learning : proceedings / zost. František Jakab. - 1. vyd. - Denver : Institute of Electrical and Electronics Engineers, 2020. - 789 s. [online]. - ISBN 978-0-7381-2366-0. - s. 1-5

- 2021 [01] UGWUANYI, S., PAUL, G., IRVINE, J. *Survey of iot for developing countries : performance analysis of lorawan and cellular nb-iot networks.* In: Electronics. ISSN 2079-9292, 2021, vol. 10, iss. 18, art. no. 2224, s. 1-30. SCOPUS; WoS