VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2021 Bc. Denis Kramář

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

ANALÝZA ZVUKOVÝCH NAHRÁVEK POMOCÍ HLUBOKÉHO UČENÍ
DEEP LEARNING BASED SOUND RECORDS ANALYSIS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR
Bc. Denis Kramář

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. Jiří Přinosil, Ph.D.

BRNO 2021

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Audio inženýrství**
specializace Zvuková produkce a nahrávání
Ústav telekomunikací

Student: Bc. Denis Kramář ***ID:*** 186622
Ročník: 2 ***Akademický rok:*** 2020/21

NÁZEV TÉMATU:

Analýza zvukových nahrávek pomocí hlubokého učení

POKYNY PRO VYPRACOVÁNÍ:

V rámci práce se seznamte s principem analýzy dat pomocí strojového učení. Na základě získaných znalostí navrhněte a implementujte algoritmus pro analýzu zvukových nahrávek pomocí konvolučních neuronových sítí. Ověřte realizovaný algoritmus na databázi zvukových nahrávek z lesního prostředí se zaměřením na klasifikaci zvuku motorové pily. Proveďte diskuzi nad dosaženými výsledky.

DOPORUČENÁ LITERATURA:

[1] HERSHEY, Shawn, et al. CNN architectures for large-scale audio classification. In: 2017 ieee international conference on acoustics, speech and signal processing (icassp). IEEE, 2017. p. 131-135.

[2] HOMBURG, Helge, et al. A Benchmark Dataset for Audio Classification and Clustering. In: ISMIR. 2005. p. 528-31.

Termín zadání: 1.2.2021 ***Termín odevzdání:*** 24.5.2021

Vedoucí práce: Ing. Jiří Přinosil, Ph.D.

doc. Ing. Jiří Schimmel, Ph.D.
předseda rady studijního programu

ABSTRAKT

Tato diplomová práce se zabývá řešením problému audio-klasifikace zvuku těžby motorové pily v přirozeném prostředí s využitím převážně konvolučních neuronových sítí. Nejprve je probrána teorie týkající se grafické reprezentace zvukového signálu. Další část je věnována oblasti strojového učení. Ve třetí kapitole jsou prezentovány některé současné práce zabývající se touto problematikou. V rámci praktické části je představen použitý dataset a testované neuronové sítě. Dosažené výsledky testování jsou porovnány na základě dosažené úspěšnosti a pomocí křivek ROC. Robustnost představených řešení je ověřena pomocí navrženého detekčního programu a zhodnocena pomocí objektivních kritérií.

KLÍČOVÁ SLOVA

audiosignál, klasifikace, detekce, konvoluční neuronové sítě, CNN, LSTM, MFCC, zpracování signálu, strojové učení, neuronová síť, nelegální kácení

ABSTRACT

This master thesis deals with the problem of audio-classification of the chainsaw logging sound in natural environment using mainly convolutional neural networks. First, a theory of grafical representation of audio signal is discussed. Following part is devoted to the machine learning area. In third chapter, some of present works dealing with this problematics are given. Within the practical part, used dataset and tested neural networks are presented. Final resultes are compared by achieved accuracy and by ROC curves. The robustness of the presented solutions was tested by proposed detection program and evaluated using objective criteria.

KEYWORDS

audio signal, classification, detection, convolutional neural network, CNN, LSTM, MFCC, signal processing, machine learning, neural network, illegal logging

KRAMÁŘ, Denis. *Analýza zvukových nahrávek pomocí hlubokého učení*. Brno, 2021, 64 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Jiří Přinosil, Ph.D.

PROHLÁŠENÍ

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

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

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Jiřímu Přinosilovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Seznam obrázků

Úvod

Nelegální těžba dřeva je jeden z největších světových problémů, který má nedozírný dopad na životní prostředí. Ze studií vyplývá, že každý rok se nelegálně pokácí přes 100 milionů m^3 dřeva [13]. Aktivní monitoring lesního prostředí hraje velmi významnou roli v boji s tímto problémem. Dále se ukazuje, že nejlépe realizovatelným typem monitoringu jsou audio-klasifikační systémy.

V této práci je představeno řešení otázky detekce zvuku těžby dřeva pomocí algoritmů *strojového učení*. Pro klasifikaci zvukového signálu jsou zde využity *Konvoluční neuronové sítě* a jejich kombinace s *Rekurentní neuronovou sítí*. Využití neuronových sítí pro klasifikaci obrazových dat je v současné době hojně rozšířené a tyto sítě vykazují nejlepší výsledky ve srovnání s jinými technikami. V poslední době se však ukazuje, že mohou být velice užitečné i pro analýzu zvukových dat.

První kapitola obsahuje rozebrání problematiky grafické reprezentace zvukového signálu. Jsou zde popsány nejpoužívanější metody transformací zvukového signálu a extrakce příznaků využívané pro diskrétní analýzu zvukových dat.

Druhá kapitola je věnována rozboru problematiky týkající se *strojového učení*. Je zde popsána základní teorie včetně historického kontextu jejich vzniku. Dále je zde popsán princip *neuronových sítí* a také *konvolučních a rekurentních neuronových sítí*, které jsou posléze využity v samotném návrhu řešení.

Ve třetí části jsou uvedeny současné práce zabývající se zmíněnou problematikou. Jsou zde rozebrány tři články, které se zabývají přímo detekcí zvuku těžby dřeva. Další dvě uvedené práce řeší taktéž problematiku klasifikace audio signálů v jiných oblastech aplikace.

Čtvrtou kapitolou je samotný popis navrženého řešení. Je zde popsán způsob tvorby použitého datasetu a jeho následné zpracování. Dále jsou zde detailně popsány architektury použitých sítí a jejich nastavení. Zbylá část této kapitoly je věnována funkci detekčního programu, navrženého pro účely testování účinnosti sítí.

Poslední pátá kapitola obsahuje diskuzi dosažených výsledků představených sítí. Nejprve jsou prezentovány výsledky prvotního testování dosažené v rámci semestrální práce. Dále jsou zde již porovnány jednotlivé průběhy učení navržených sítí a úspěšnost jejich následné klasifikace testovací sady datasetu. Poslední část je věnována porovnání výsledků klasifikace testovacích nahrávek vytvořených za účelem ověření funkčnosti navržených řešení v reálném použití.

1 Reprezentace audio signálu

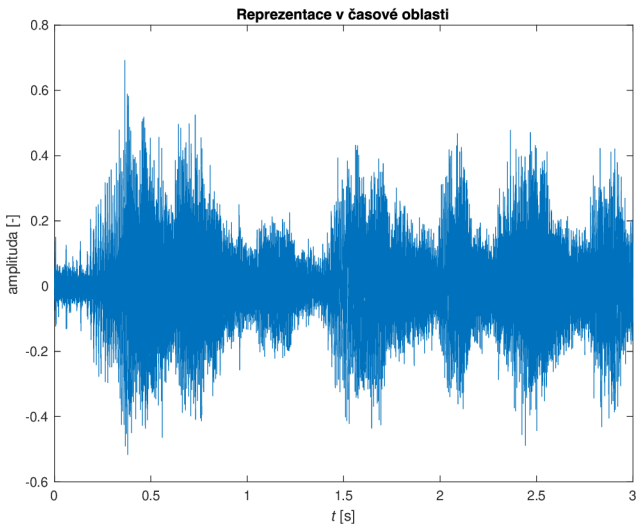
Zvukový signál po převedení do diskrétní oblasti představuje posloupnost jednotlivých zvukových vzorků. Aby bylo možné tato data následně analyzovat či využít v oblasti strojového učení, je žádoucí jejich úprava pomocí vhodných transformací.

V této části jsou přiblíženy nejpoužívanější metody reprezentace zvukového signálu a parametry používané pro následnou analýzu. Nejprve je zde popsána časová reprezentace signálu a poté jednotlivé časově-frekvenční reprezentace používané pro analýzu zvukových dat, a to *spektrogram*, spektrogram s využitím banky *melovských filtrů*, *Constant-Q transformace* a *kepstrální analýza*. Ukázky grafických zobrazení jsou prezentovány na reálném zvukovém vzorku motorové pily použitém pro pozdější trénování neuronové sítě.

1.1 Reprezentace signálu v časové oblasti

Zvukový signál lze jednoduše zobrazit jakožto aktuální hodnotu amplitudy v určitém časovém momentě. Tato reprezentace je vhodná pro zobrazení delších časových úseků, nicméně samotný časový průběh obsahuje příliš mnoho informací a není tak zcela vhodný pro analýzu, jejímž cílem je popsat charakter obsahu zvukové nahrávky. Z tohoto důvodu se využívají parametrické popisy v časově-frekvenční oblasti pro zjednodušení a popsání signálu menším množstvím parametrů [1].

Na obrázku 1.1 je zobrazen časový průběh nahrávky motorové pily ze vzdálenosti 5 m o délce trvání 3 s pro vzorkovací kmitočet 16000 Hz.



Obr. 1.1: Časový průběh nahrávky motorové pily.

1.2 Spektrogram

Nejpoužívanějším nástrojem pro zobrazení signálu v časově-frekvenční oblasti je spektrogram. Jedná se o zobrazení změny frekvenčního spektra v čase, kde je hodnota spektrální amplitudy reprezentována barevnou škálou. Pro transformaci signálu z časové do časově-frekvenční domény se zpravidla používá diskrétní krátkodobá Fourierova transformace (*Short-Time Fourier Transform* – STFT).

Nejprve dochází k segmentaci signálu na krátké časové úseky o jednotné délce. Během tohoto procesu vlivem periodicity signálu však dochází k prosakování spektrálních složek. Jednotlivé segmenty se tedy zpravidla násobí takzvaných *oknem*, aby se tento jev minimalizoval. Nejpoužívanějšími typy okna jsou okno pravoúhlé a Hammingovo, často se však setkáváme i s Hannovým nebo Blackmanovým oknem. Výběr okna se odvíjí od typu informace, která má být analyzována.

Pravoúhlé okno poskytuje vyšší rozlišení kmitočtového spektra, nicméně dochází k většímu prosakování než u okna Hammingova. To však poskytuje nižší frekvenční rozlišení. Velikost jednotlivých segmentů (oken) opět závisí na povaze vstupního signálu. Zde platí Heisingův princip neurčitosti, krátké okno způsobuje lepší časové rozlišení na úkor frekvenčního a naopak. Ideální nastavení je tedy dáno kompromisem obou domén. Nicméně minimální limit pro časové a frekvenční rozlišení zvoleného okna je dán vztahem,

$$\Delta_t \Delta_f \geq \frac{1}{4\pi}, \tag{1.1}$$

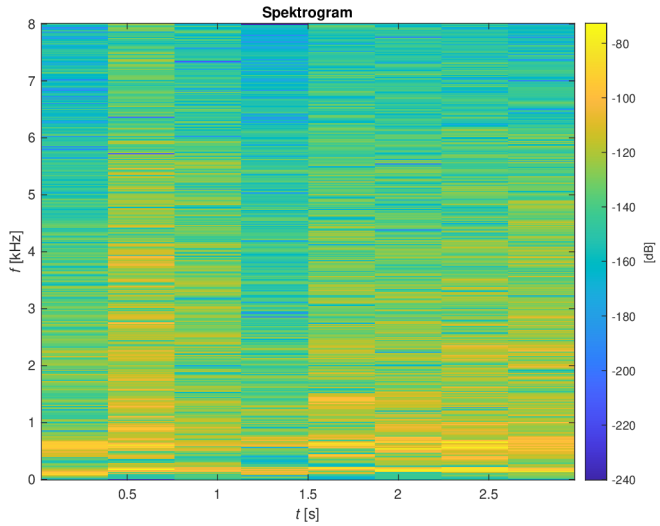
kde Δ_t představuje změnu v časové doméně a Δ_f ve frekvenční. Při váhování oknem se také zpravidla nastavuje překryv jednotlivých segmentů. Při překryvu 50 % začíná následující segment v polovině segmentu předchozího. Při použití jiného než pravoúhlého okna kompenzuje překryv úbytek energie v okrajových oblastech okna [3].

Vztah pro výpočet STFT můžeme pro vstupní diskrétní signál $s\,[n]$ a časové okno $w\,[n-m]$ s posuvem m zapsat jako:

$$S_{\mathrm{STFT}}\left(e^{\mathrm{j}\omega}, m\right) = \sum_{n=-\infty}^{\infty} s\,[n]\, w\,[n-m]\, e^{-\mathrm{j}2\pi f n}, \tag{1.2}$$

kde f je aktuální hodnota kmitočtu. Dostáváme tedy reprezentaci se spojitou kmitočtovou osou a diskrétní časovou osou [1].

Spektrogram signálu, jehož reprezentace v časové oblasti je zobrazena na obrázku 1.1, je možno vidět na obrázku 1.2. Signál byl váhován Hammingovým oknem o délce 1600 vzorků s překryvem 50 %.



Obr. 1.2: Spektrogram nahrávky motorové pily.

1.3 Mel spektrogram

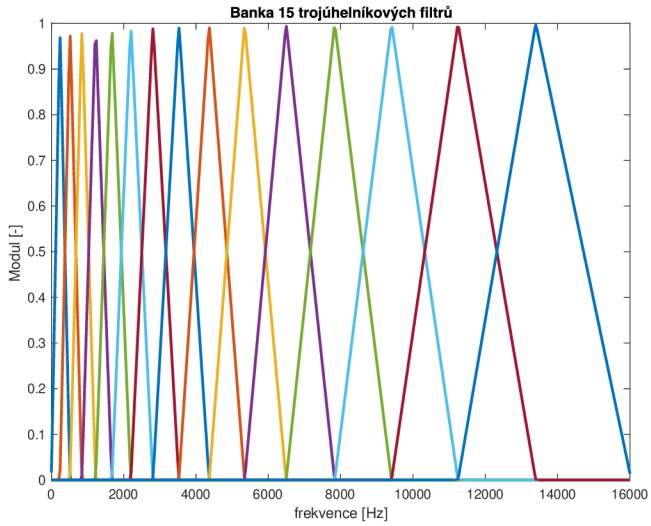
Jednou z mnoha modifikací klasického spektrogramu je takzvaný *Mel spektrogram*. Jednotlivé segmenty signálu jsou po transformaci pomocí STFT váhovány bankou *trojúhelníkových mel filtrů*. Výsledné hodnoty jsou poté pro jednotlivé filtry sečteny. Tento postup se poté aplikuje na další časové rámce a tím dostáváme podobu spektrogramu, jehož rozměry jsou dány počtem *mel filtrů* m a délkou váhovacího okna N, jako $m \times N$ [2].

Použití *trojúhelníkových filtrů* má za účel napodobit vjemové vlastnosti lidského ucha. Ze stejného důvodu jsou jednotlivé filtry rozmístěny na logaritmické *mel škále*, která má napodobovat logaritmické vnímání zvuku bazilární membrány lidského ucha. Rozložení 15 mel filtrů škály je možné vidět na obrázku 1.3. Pro přepočet mezi frekvenční a melovskou škálou se používá vztah:

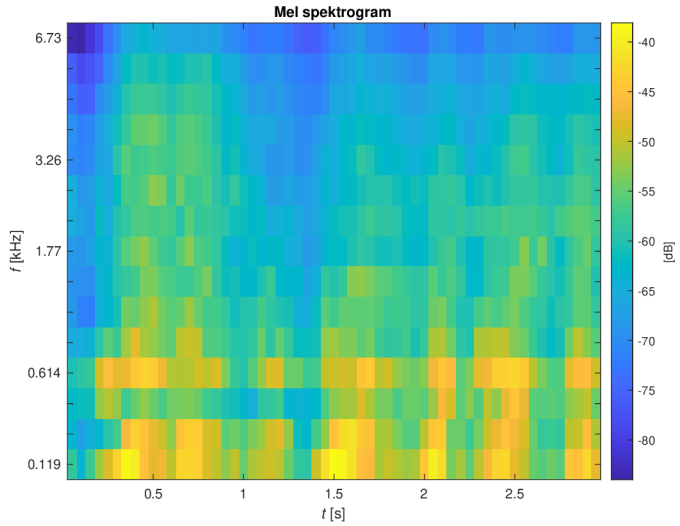
$$mel\,(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right), \tag{1.3}$$

kde f představuje hodnotu frekvence ve frekvenční škále [1]. Vliv banky filtrů zobrazené na obrázku 1.3 na spektrogram motorové pily je možné vidět na obrázku 1.4.

Využití logaritmické škály a váhování filtry je vhodné právě pro analýzu zvukových dat. Lineární rozložení při využití klasické STFT způsobuje příliš velkou koncentraci informací v oblasti vysokých kmitočtů, které nejsou z hlediska zpracování hudebního signálu příliš důležité.



Obr. 1.3: Modulová frekvenční charakteristika banky 15 mel filtrů.



Obr. 1.4: Mel spektrogram nahrávky motorové pily.

1.4 Constant-Q

Stejně jako v případě Mel spektrogramu, je *Constant-Q transformace* založena na logaritmickém rozložení škály za pomoci filtrace bankou logaritmicky rozložených filtrů. Jednou z hlavních předností Constant-Q je reprezentace harmonických složek v pravidelných intervalech, což napomáhá identifikaci zvuku.

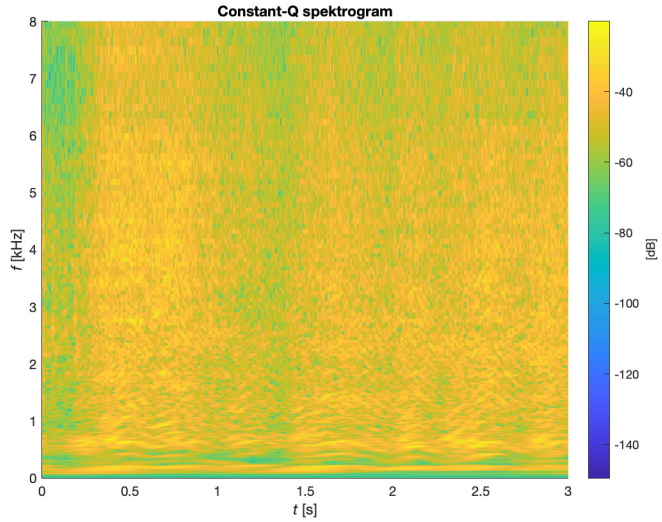
Zavádíme tedy parametr Q, který je v případě čtvrttónového rozlišení dán jako:

$$Q = \frac{f_k}{(\delta f)_{\mathrm{CQT}}} = \frac{f_k}{(2^{1/24} - 1)\, f_k} \approx 34, \tag{1.4}$$

kde δf představuje frekvenční rozlišení vypočtené jako podíl vzorkovacího kmitočtu a počtu vzorků transformace a f_k je reprezentací jednotlivých frekvencí. Potom je možné samotnou diskrétní Fourierovu transformaci s využitím Constant-Q pro spojitý signál zapsat jako:

$$X_{\mathrm{CQT}}(k) = \frac{1}{N_k} \sum_{n=0}^{N_k-1} W(n,k)\, x(n)\, e^{-\mathrm{j}\frac{2\pi}{N_k}Qn}, \tag{1.5}$$

pro $k = 0, 1, ..., (N-1)$, kde $W(n,k)$ představuje váhovací funkci použitého okna [4]. Výsledek Constant-Q transformace testovacího signálu, při stejném nastavení jako v případě spektrogramu a 34 pásmech, je zobrazen na obrázku 1.5.



Obr. 1.5: Constant-Q spektrogram nahrávky motorové pily.

1.5 Kepstrální analýza

Při zpracování řeči se velmi často používá *komplexní a reálné kepstrum*. Jedná se o nelineární metodu transformace signálu, jejíž výsledek si lze zjednodušeně představit jako nelineární spektrum spektra vstupního signálu. Kepstrum nám udává rychlost změn v čase pro různé frekvenční oblasti. Jelikož je řečový signál dán konvolucí impulsní charakteristiky modelu hlasového traktu a vstupního budícího signálu, je pomocí kepstrální analýzy možné oddělit parametry hlasového traktu od reálného budícího signálu a zjistit tak základní periodu řeči. Odezva hlasového traktu se v kepstru projeví jako pomalé změny blízko počátku, rychlé změny se projeví jako špička v čase, který odpovídá základní hlasivkové periodě. V praxi se setkáváme

s komplexním a reálným kepstrem. Výpočet komplexního kepstra je však poměrně komplikovaný, proto se v oblasti zpracování řeči využívá reálné kepstrum [1].

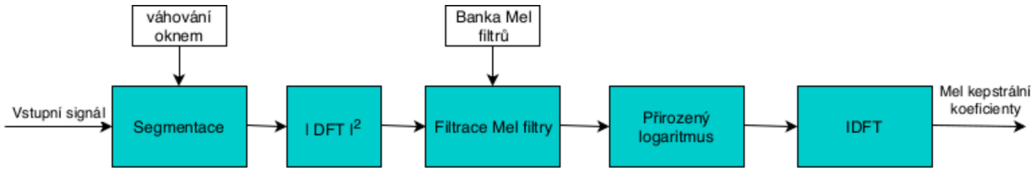
Pro spektrum $S(k)$ diskrétního signálu vyjádřené pomocí vztahu 1.2 je možné reálné kepstrum definovat jako:

$$c(n) = \mathrm{Re}\left\{\frac{1}{N}\sum_{k=0}^{N-1} \ln|S(k)|e^{\mathrm{j}\frac{2\pi}{N}kn}\right\}, \tag{1.6}$$

pro $k = 0, 1, ..., (N)$, kde N představuje počet hodnot FFT. Celý postup určení reálného kepstra je tedy následovný. Nejprve je signál segmentován a váhován časovým oknem a pomocí FFT převeden na spektrum. Následně je na absolutní hodnotu spektra aplikován přirozený logaritmus a zpětná diskrétní transformace pomocí zpětné diskrétní kosinové transformace (*Inverse Discrete Cosine Transform* – IDCT), jejíž reálná část odpovídá reálnému kepstru [2].

1.5.1 Mel frekvenční kepstrální koeficienty

Jedním z oblíbených druhů parametrů pro zpracování řeči jsou také *melovské kepstrální koeficienty*. Opět je zde využito nelineární rozložení, které bere v potaz maskovací vlastnosti lidského sluchového systému. Jedná se tedy o výpočet reálného kepstra signálu váhovaného bankou melovských filtrů (viz 1.3). Nejprve je signál segmentován a váhován časovým oknem, poté je pomocí DFT transformován do podoby výkonnostního frekvenčního spektra. Poté je spektrum filtrováno bankou trojúhelníkových filtrů. Hodnoty pro jednotlivé filtry všech segmentů jsou následně sečteny, aby každému filtru banky odpovídala jedna hodnota. Poté následuje výpočet reálného kepstra podle postupu uvedeného výše. Celý tento proces je znázorněn na blokovém schématu 1.6.



Obr. 1.6: Blokové schéma výpočtu Mel frekvenčních kepstrálních koeficientů.

2 Strojové učení

Strojové učení (*Machine learning*) je jednou z oblastí oboru *Umělá inteligence*. Jedná se o algoritmy schopné se samostatně zdokonalovat v plnění zadaného úkolu na základě rostoucích zkušeností. Toto učení probíhá na základě změny vnitřních stavů systému. Algoritmy strojového učení vytvářejí matematický model na základě databáze vstupních dat, díky kterému jsou schopné poskytnout aproximaci možného řešení zadaného úkolu [5]. Oblastí využití algoritmů strojového učení je nepřeberné množství takřka ve všech odvětvích. Základními druhy řešených úloh systémů jsou *klasifikace* – rozdělování vstupních dat do výstupních tříd podle zadaných parametrů, *regrese* – odhad číselné hodnoty výstupu na základě vstupních dat a také *shlukování* – zařazování objektů do skupin na základě společných znaků.

Rozdělujeme základní dva typy algoritmů na základě způsobu učení na algoritmy *učení s učitelem* a *učení bez učitele*. Pro vstupní data algoritmů učení s učitelem (*supervised learning*) existuje předem daný kýžený výstup (třída v případě klasifikačních algoritmů a hodnota v případě regresivních), naproti tomu u učení bez učitele (*unsupervised learning*), jsou výstupy neznámé a algoritmus sám hledá podobnosti ve vstupních datech pro jejich separaci bez možnosti posouzení správnosti třídění. Pro učení s učitelem jsou tedy na vstup sítě kromě vstupních dat přivedena žádaná výstupní data. Tato data se poté člení na trénovací, validační a testovací data. Algoritmus během učení na sadě trénovacích dat kontroluje správnost postupu na sadě validačních dat. Po dosažení zadané přesnosti nebo doby trvání se úspěšnost sítě ověřuje na sadě testovacích dat.

Ve zbytku kapitoly jsou popsány základní a pro tuto práci důležité oblasti strojového učení. Největší prostor je věnován *Konvolučním neuronovým sítím*, jejich základnímu principu, využití a architektuře.

2.1 Umělé neuronové sítě

Umělé neuronové sítě (*Artificial Neural Networks* – ANN nebo pouze NN) jsou podoblastí strojového učení, potažmo Umělé inteligence, jejíž popularita je v současné době čím dál větší. Jak už název napovídá, inspirací pro vznik tohoto odvětví byla studie rozpoznávacích schopností lidského mozku. Ten je tvořen velmi komplexní sítí přibližně 10^{11} mezi sebou propojených neuronů. Každý neuron je spojen průměrně s 10^4 dalšími neurony [6]. Algoritmy neuronových sítí jsou tedy založeny na principu sítě mezi sebou propojených jednoduchých článků.

S neuronovými sítěmi se setkáváme už od roku 1943, kdy neurofyziolog W. Culloch a matematik W. Pitt na základě zkoumání lidského mozku sestrojili první jednoduchou neuronovou síť, na které popsali jejich představu funkce nervové aktivity [7].

Jednotlivé umělé neurony zde byly reprezentovány lineární funkcí schopnou zpracovat několik vstupů s jediným binárním výstupem nabývající hodnotu 0 v případě neaktivity a 1, pokud součet vstupů překročil zadanou hodnotu.

V roce 1958 představil F. Rosenblatt model *Perceptronu*, nejjednodušší neuronové sítě, sestávající z jednoho jediného neuronu [8]. Jednalo se o převratný objev, později však bylo zjištěno, že lze použít pouze na lineárně separovatelné množiny dat.

K největšímu rozmachu na poli neuronových sítí však dochází až od 80. let 20. století s příchodem výkonnější výpočetní techniky. V roce 1986 se také poprvé setkáváme s pojmem *Hluboké učení* (*Deep Learning*), jež formulovala profesorka Rina Dechter [9]. Pojem „hluboké" odkazoval na použití skrytých vrstev v architektuře sítě.

2.1.1 Model umělého neuronu

Základní stavební buňkou neuronových sítí je umělý neuron (*perceptron*). Jedná se o nejjednodušší podobu neuronové sítě s jedním binárním výstupem. Na vstupu je vektor $x = (x_1, x_2, x_3, ..., x_n)$ o n prvcích, přičemž každému vstupu je přiřazena hodnota váhovacího vektoru $w = (w_1, w_2, w_3, ..., w_n)$. Tato váha je reálná hodnota odpovídající míře důležitosti daného vstupu na žádaný výstup. Ten může odpovídat hodnotě 0 nebo 1 v závislosti na váženém součtu:

$$\sum_j w_j x_j. \tag{2.1}$$

Pokud je tato suma větší než zadaná prahová úroveň b (bias), bude na výstupu 1, v opačném případě 0. Pro výstup perceptronu y je možné toto pravidlo zapsat jako:

$$y = \begin{cases} 0, & \text{pro } w \cdot x + b \leq 0, \\ 1, & \text{pro } w \cdot x + b > 0. \end{cases} \tag{2.2}$$

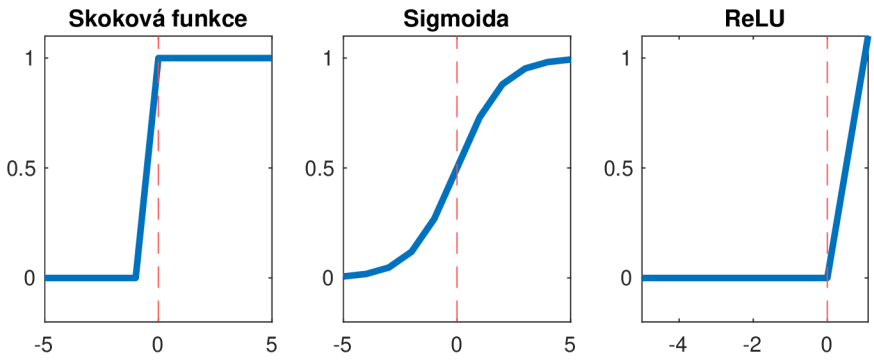
Výstup sumační funkce 2.1 zpravidla bývá přiveden na vstup nelineární aktivační funkce f_x, která ovlivňuje, jakým způsobem bude vnitřní potenciál neuronu převeden na výstup [2]. Mezi nejčastěji používané funkce patří:

- Skoková funkce
$$f(x) = \begin{cases} 0, & \text{pro } x < 0, \\ 1, & \text{pro } x \geq 0, \end{cases} \tag{2.3}$$
- Sigmoida
$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}, \tag{2.4}$$

- ReLU

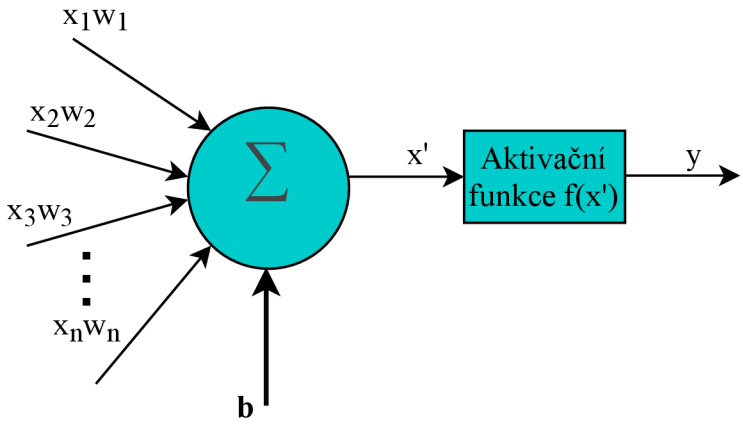
$$f(x) = \begin{cases} 0, & \text{pro } x < 0, \\ x, & \text{pro } x \geq 0. \end{cases} \tag{2.5}$$

Jejich grafická podoba je znázorněna na obrázku 2.1. Struktura samotného perceptronu je na obrázku 2.2.



Obr. 2.1: Přehled aktivačních funkcí.

Na vstup neuronu jsou postupně přiváděna vstupní data trénovací množiny v podobě vstupního souboru $D = \{x, d\}$, kde x představuje testovací vzorek a d informaci o žádané výstupní klasifikaci. Po předložení všech vzorků testovací množiny se upravují váhy neuronu a celý proces se iterativně opakuje, dokud nedojde ke splnění ukončující podmínky. Algoritmus hledá co nejlepší konfiguraci vah neuronu, aby byl schopný separovat vstupní množinu s co nejnižší chybou. Tento proces se nazývá *učení neuronu* a je základní myšlenkou neuronových sítí [10].

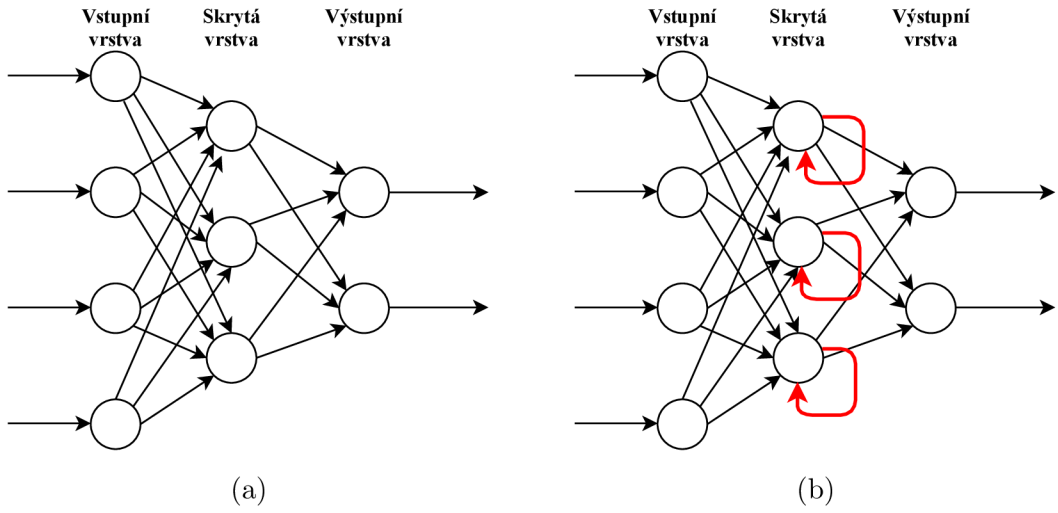


Obr. 2.2: Model umělého neuronu.

2.1.2 Vrstvy neuronové sítě

Jak již bylo řečeno, neuron není sám o sobě schopen řešit složitější úlohy než binární separaci lineárně separovatelné množiny dat. Aby bylo možné jeho funkci využít na složitější zadání, je nutné použít neuron, jakožto stavební buňku, a uspořádat jich více do takzvaných *vrstev*. Vznikají tak *vícevrstvé perceptronové sítě* (*Multi-layred Perceptron* – MLP). V zásadě rozlišujeme jednotlivé vrstvy těchto sítí na *vstupní, skryté* a *výstupní*. Jako skryté vrstvy se označují všechny vrstvy nacházející se mezi vstupní a výstupní vrstvou. Jednotlivé neurony jedné vrstvy mezi sebou nejsou nijak propojené, propojení vrstev však probíhá stylem „každý s každým". Výstup jednoho neuronu je tedy přiveden na vstup všech neuronů další vrstvy. Každé propojení je opatřené samostatnou váhou. Tento případ propojení sítě, kdy jsou výstupy neuronů přivedeny pouze na vstup následující vrstvy, se označuje jako *dopředná síť*. V případě *rekurentní* sítě se objevují zpětné vazby a výstupy neuronů jsou přivedeny zpět na jejich vstup či na vstup neuronu v předcházející vrstvě. Na obrázku 2.3 můžeme vidět příklad dopředné a rekurentní neuronové sítě s jednou skrytou vrstvou.

Počet neuronů vstupní vrstvy závisí na podobě vstupních dat, počet neuronů výstupní vrstvy zase určuje podobu výsledné klasifikace a zpravidla odpovídá počtu klasifikačních tříd. Počet skrytých vrstev, jejich velikost a nastavení, stejně jako volba architektury, se odvíjí od charakteru řešeného problému a podoby datasetu.



(a) (b)

Obr. 2.3: Příklad dopředné (a) a rekurentní (b) neuronové sítě.

2.2 Konvoluční neuronové sítě

Konvoluční neuronové sítě (*Convolutional Neural Network* – CNN) jsou rozšířenou variantou dopředných neuronových sítí, speciálně uzpůsobenou pro úlohu klasifikace obrazových dat. Základním principem je extrakce příznaků obrazu pomocí aplikace několika konvolučních filtrů a jejich následný výběr podvzorkovací vrstvou. Čím hlubší vrstva, tím vyšší jsou extrahované příznaky, pomocí kterých je následně objekt na obraze rozpoznán. Na výstupu konvoluční sítě je vektor hodnot, který je přiveden na vstup plně propojené neuronové vrstvy [10]. Jako aktivační funkce konvoluční vrstvy se zpravidla používá funkce ReLU, popsaná vztahem 2.5.

Konvoluční neuronové sítě se používají převážně pro klasifikaci obrazů a jejich částí či jejich shlukování. Nicméně po vhodné úpravě vstupních dat jsou i velmi užitečné pro práci se zvukovými daty.

V následujících částech budou blíže popsané jednotlivé druhy vrstev konvolučních sítí a jejich použití.

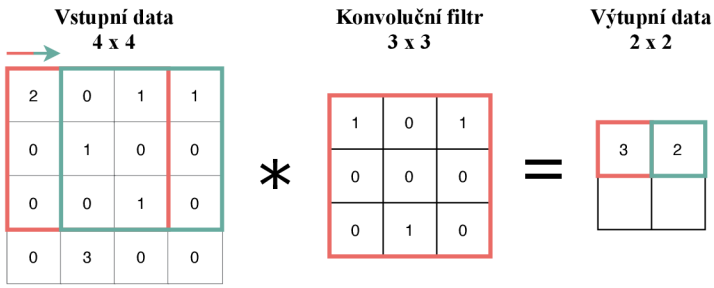
2.2.1 Konvoluční vrstva

Jeden ze dvou základních druhů vrstev konvolučních sítí jsou samotné vrstvy provádějící konvoluci vstupních dat s jednotlivými filtry (*jádry* či *kernely*). Na vstupní obraz o rozměrech $m \times n \times r$, kde m, n představují výšku a šířku obrazu a r počet barevných kanálů, je aplikován konvoluční filtr o rozměrech $n \times n \times r$. Pro konvoluční jádro platí, že jeho velikost n je menší než velikost vstupního obrazu a počet kanálů r stejný nebo menší než u vstupních dat. Z matematického hlediska se jedná o diskrétní 2D konvoluci, kterou lze pro vstupní matici x o dvou rozměrech a dvourozměrné konvoluční jádro w zapsat jako:

$$y\left[m,n\right] = x\left[m,n\right] * w\left[m,n\right] = \sum_{k}\sum_{l} x\left[k,l\right] w\left[m-k,n-l\right]. \tag{2.6}$$

Aby bylo zajištěno, že se do výsledku konvoluce započítají i krajové hodnoty vstupního obrazu, bývá vstupní matice obohacena o nulové okraje. Tomuto procesu se říká *zero padding*.

Rozměr výsledného příznakového prostoru je dán nejen rozměry vstupu a jádra, ale i krokem posunu jádra po vstupní matici. Názorný grafický příklad konvoluční vrstvy pro krok velikosti 1 je zobrazen na obrázku 2.4. Jednotlivé konvoluční filtry představují matice vah. Na začátku učení jsou zvoleny náhodné hodnoty vah a jejich hodnota se následně v průběhu trénování mění pomocí zpětného šíření chyby. Úlohou každé vrstvy je redukce příznakového prostoru prostřednictvím zvýraznění nejdůležitějších oblastí obrazu změnou obrazového prostoru [10].

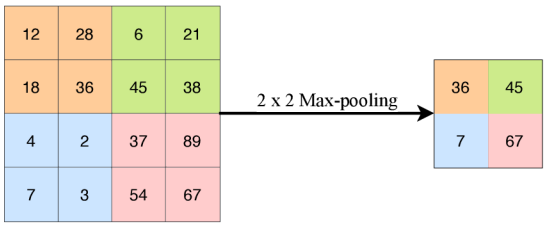


Obr. 2.4: Příklad funkce konvoluční vrstvy.

2.2.2 Podvzorkovací vrstvy

Druhým typem vrstev CNN jsou vrstvy sdružovací nebo také podvzorkovací vrstvy. Jejich úlohou je snížení počtu výstupních prvků konvoluční vrstvy pomocí funkcí *mean* a *max*, což tak umožní další vrstvě extrahovat vyšší příznaky obrazu při zachování stejného rozměru jádra konvoluční vrstvy. Tento proces i značně snižuje výpočetní náročnost algoritmu a tím zrychluje jeho průběh. Nejčastěji používaným typem sdružovací vrstvy je takzvaná *max-pooling* vrstva, která na základě její velikosti vybírá prvek s nejvyšší hodnotou daného regionu. Ukázku jednoduché max-pooling vrstvy je možné vidět na obrázku 2.5.

Dalším příkladem podvzorkovací vrstvy může být *flatten* vrstva, jejíž úlohou je



Obr. 2.5: Princip funkce max-pooling.

převod matice příznakové mapy na vektor, který může být přiveden na plně propojenou neuronovou vrstvu.

2.2.3 Batch normalization

Speciálním druhem vrstvy konvoluční neuronové sítě je *Batch normalization* vrstva. Její úlohou je normalizace dat vstupní dávky před následující konvoluční vrstvou. Vrstva nejprve vypočítá průměr a rozptyl výstupu předcházející vrstvy a na základě těchto hodnot je provedena normalizace dat, aby jejich střední hodnota byla nulová. Vzorek x_i dávky $\boldsymbol{b} = (x_1, x_2, ..., x_n)$ je normalizován podle:

$$\widehat{x}_i = \frac{x_i - \mu_{\boldsymbol{b}}}{\sigma_{\boldsymbol{b}}^2 + \epsilon}, \tag{2.7}$$

kde $\mu_{\boldsymbol{b}}$ je průměr, $\sigma_{\boldsymbol{b}}$ rozptyl a ϵ přidaná konstanta zaručující numerickou stabilitu [11]. Normalizace zaručuje stabilnější rozdělení hodnot a menší citlivost na nastavení počátečních parametrů, což může vézt k rychlejší konvergenci a tím i rychlejšímu chodu algoritmu [10].

2.2.4 Dropout vrstva

Úlohou *dropout* vrstvy je deaktivace zadaného procenta náhodně vybraných spojů mezi vrstvami neuronové sítě. Tato vrstva se zavádí jako obrana proti *přeučení* sítě, což je nežádoucí jev, při kterém vzniká přílišná závislost nastavení sítě na trénovací množině, což může mít za následek špatné výsledky při testování sítě na neznámých datech. Zavedením *dropout* vrstvy se snižuje vzájemná závislost jednotlivých neuronů.

Tato metoda je jednou z takzvaných *regularizačních* metod, které jsou používány z důvodu snížení počtu parametrů a mohou napomoci i k rychlejšímu učení. Jako další regularizační (*regression*) metody stojí za zmínku *L1* a *L2*, které pracují na principu regulace velikosti vah pomocí zavedení váhovacího kritéria. Dochází tak k omezení vysokých aktivačních hodnot a malé změny na vstupu sítě nevyvolávají velké skoky na výstupu. Nesprávnou volbou nastavení však může docházet ke špatnému rozlišování výstupních tříd [10].

2.2.5 Softmax vrstva

Poslední vrstvou před výstupní klasifikací konvoluční sítě velmi často bývá *softmax vrstva*. Jejím principem je transformace výstupních dat předchozí vrstvy na pravděpodobnostní množinu hodnot v intervalu $\langle 0, 1\rangle$, jejíž počet prvků odpovídá počtu výstupních tříd a součet se rovná 1. Transformace probíhá prostřednictví funkce *softmax*. Pravděpodobnostní hodnota prvku x_i vstupního vektoru x pro N výstupních tříd je dána vztahem

$$f(x)_i = \frac{e^{x_i}}{\sum_{n=1}^{N} e^{x_n}}. \tag{2.8}$$

Tato transformace je vhodná pouze v případě, že každému vstupu odpovídá pouze jedna výstupní třída.

2.3 Rekurentní neuronové sítě

Druhým typem architektury neuronových sítí jsou sítě zpětnovazební neboli rekurentní (*Recurrent Neural Network* – RNN). Jak již název napovídá, jedná se o sítě opatřené zpětnou vazbou, kde výstup neuronu je přiveden opět na jeho vstup nebo

na vstup jiného neuronu v předcházející vrstvě. Ukázka architektury takové sítě je zobrazená na obrázku 2.3b.

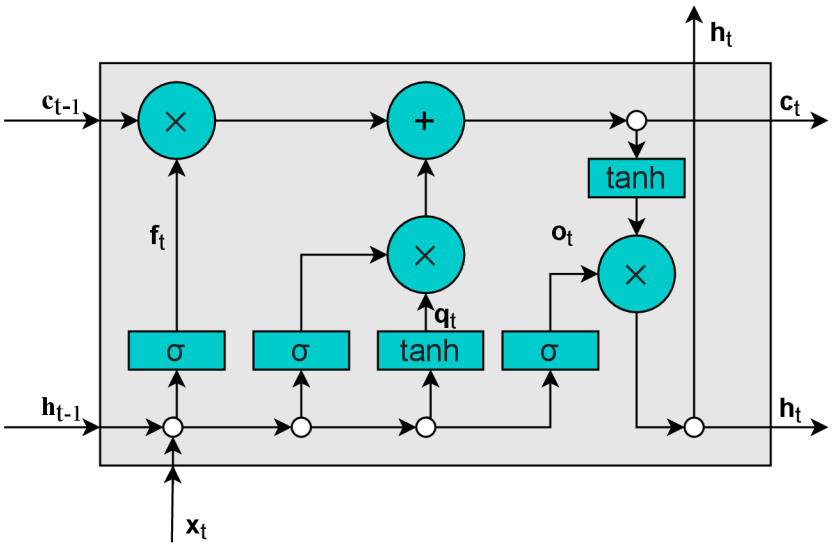
Zavedením zpětné vazby tak vzniká jakási paměť, díky které je síť schopná zpracovávat libovolně dlouhé, časově proměnné posloupnosti. Hlavním důvodem využívání těchto sítí je jejich schopnost nacházet souvislosti u delších časových úseků vstupního signálu, díky jejich dynamickému zpracovávání dat [19].

2.3.1 LSTM sítě

Jednou z konkrétních realizací rekurentních sítí jsou LSTM sítě (*Long Short-Term Memory*). Jejich hlavní výhodou oproti základní architektuře RNN sítě je schopnost uchovat v paměti až tisíce vzorků, přičemž jednoduché RNN jsou schopny pracovat jen v řádu desítek vzorků [19]. Základními prvky těchto sítí jsou LSTM buňky, které jsou řetězeny za sebe. Každá buňka uchovává dva stavy. *Vnitřní stav* $\boldsymbol{c}_t$ uchovává informaci o vstupní posloupnosti a prochází celým řetězcem beze změn. Díky tomu je síť schopná nacházet souvislosti i u časově velmi vzdálených úseků. Druhým stavem je *skrytý stav* $\boldsymbol{h}_t$, který je zprostředkovaný pomocí krátkodobé paměti.

LSTM buňky mohou upravovat své vnitřní stavy pomocí takzvaných bran. Tyto brány jsou realizovány pomocí operace násobení a nelineární sigmoidní funkce. Schéma popisované LSTM buňky je zobrazené na obrázku 2.6.

První z bran je brána *zapomínací*. Tato brána rozhoduje o tom, zda má být od-



Obr. 2.6: Schéma LSTM buňky.

straněna informace z vnitřního stavu buňky. Na základě rovnice 2.9 nabývají složky vektoru $\boldsymbol{f}_t$ hodnotu 0 v případě, že má být daná složka vektoru předchozího stavu zahozena a 1 v případě zachování nezměněné hodnoty složky předchozího stavu.

Využitím funkce sigmoida, jakožto aktivační funkce, tento vektor nabývá hodnot v intervalu $\langle 0, 1 \rangle$, kterými jsou následně násobeny prvky vnitřního stavu předchozího kroku řetězce. Tento vztah lze vyjádřit jako

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_\mathrm{f}\boldsymbol{x}_t + \boldsymbol{U}_\mathrm{f}\boldsymbol{h}_{t-1} + \boldsymbol{b}_\mathrm{f}\right), \tag{2.9}$$

kde vektor $\boldsymbol{x}_t$ je vektorem vstupních hodnot buňky, $\boldsymbol{h}_{t-1}$ reprezentuje skrytý stav předchozí buňky. Matice $\boldsymbol{U}_\mathrm{f}$ a $\boldsymbol{W}_\mathrm{f}$ představují matice vah zapomínací brány. Vektor $\boldsymbol{b}_\mathrm{f}$ je vektorem prahových hodnot zapomínací brány.

Druhou branou LSTM buňky je brána *vstupní*. Funkcí této brány je rozhodovat, které nové informace budou přidány do vnitřního stavu. O výběru prvků vektoru $\boldsymbol{q}_t$, které budou přidány, rozhoduje rovnice 2.10. Následně je vytvořen samotný vektor pomocí rovnice 2.11. Pro vstupní bránu tedy platí vztahy

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_\mathrm{i}\boldsymbol{x}_t + \boldsymbol{U}_\mathrm{i}\boldsymbol{h}_{t-1} + \boldsymbol{b}_\mathrm{i}\right), \tag{2.10}$$

$$\boldsymbol{q}_t = \tanh\left(\boldsymbol{W}_\mathrm{q}\boldsymbol{x}_t + \boldsymbol{U}_\mathrm{q}\boldsymbol{h}_{t-1} + \boldsymbol{b}_\mathrm{q}\right), \tag{2.11}$$

kde váhy $\boldsymbol{W}_\mathrm{i}$, $\boldsymbol{U}_\mathrm{i}$, $\boldsymbol{W}_\mathrm{q}$ a $\boldsymbol{U}_\mathrm{q}$ jsou opět vztaženy na základě indexů na jednotlivé prvky. Zbylé prvky rovnice jsou definovány stejně jako v případě zapomínací brány.

Třetím typem brány je *výstupní brána*. Tato brána rozhoduje, jakým způsobem bude upraven skrytý stav buňky. Nejprve je pomocí rovnice 2.12 na základě vektorů určených předchozími branami upravena aktuální hodnota vnitřního stavu buňky. Poté jsou pomocí rovnice 2.13 vybrány indexy prvků vektoru vnitřních stavů, které mají být přivedeny na výstup. Výstupní hodnota je potom dána vektorovým součinem aktuálního vnitřního stavu s výstupem výstupní brány, což je popsáno rovnicí 2.14. Všechny tyto vztahy jsou popsány následujícími rovnicemi

$$\boldsymbol{c}_t = \boldsymbol{f}_t \times \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \times \boldsymbol{q}_t, \tag{2.12}$$

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_\mathrm{o}\boldsymbol{x}_t + \boldsymbol{U}_\mathrm{o}\boldsymbol{h}_{t-1} + \boldsymbol{b}_\mathrm{o}\right), \tag{2.13}$$

$$\boldsymbol{h}_t = \tanh\left(\boldsymbol{c}_t\right) \times \boldsymbol{o}_t. \tag{2.14}$$

3 Dosavadní práce

Tato kapitola se zabývá dosavadními pracemi v oboru analýzy zvukových dat pomocí metod hlubokého učení. Pro každý uvedený článek je zde popsán použitý dataset, metoda reprezentace signálu a zvolená metoda klasifikace. Dále jsou zhodnoceny dosažené výsledky daných řešení.

Nejprve jsou uvedeny práce týkající se přímo problematiky detekce zvuku motorové pily za účelem zamezení nelegální těžby dřeva. Práce jsou popsány především s ohledem na jednotlivá softwarová řešení.

Další část obsahuje výběr prací zabývajících se podobnou problematikou klasifikace zvukové nahrávky, jejichž výsledky a řešení jsou zajímavé pro účely této práce.

3.1 Audio detekce nelegální těžby dřeva

V otázce prevence nelegálního kácení hraje dohled a aktivní monitoring lesa významnou roli. V tomto ohledu bylo vytvořeno několik systémů založených na rozdílných technologických přístupech. Za zmínku například stojí satelitní snímání obrazu, místní video monitoring, síť bezdrátových senzorů a v neposlední řadě audio detekce zvuku motorové pily. Z posledních studií zároveň vyplývá, že audio monitoring se ukazuje jako nejefektivnější ze zmíněných přístupů [13]. V této části budou popsány některé práce zabývající se touto problematikou, jejich navržená řešení a dosažené výsledky.

3.1.1 TreeSpirit

V článku *TreeSpirit: Illegal Logging Detection and Alerting System using Audio Identification over an IoT Network* [13] z roku 2017 byl představen detekční systém využívající hluboké neuronové sítě, který byl zároveň doplněn o lokalizační algoritmus schopný určit směr přicházejícího zvuku. Pro trénování bylo použito více než 100 reálných nahrávek těžby motorové pily v deštném pralese z rozdílných vzdáleností. Tato data byla transformována pomocí FFT do frekvenčně-časové oblasti. Takto vzniklé spektrogramy byly v podobě jednokanálových obrazových dat o velikosti 128×128 použity pro trénování neuronové sítě *TensorFlow*. Síť obsahovala tři konvoluční vrstvy s ReLU aktivační vrstvou, každou následovanou vrstvou pooling. Výstup posledního konvolučního bloku byl přiveden na plně propojenou vrstvu s 50% dropout funkcí. Jako aktivační funkce před klasifikací byla použita funkce softmax.

Algoritmus dosahoval úspěšnosti přibližně 96 % na testovacích datech. Ztrátová

funkce po 300 epochách vykazovala hodnotu 0, 98. Pro následnou lokalizaci příchozího signálu byly použity algoritmy založené na výpočtu fázového rozdílu dvou zvukových signálů. Pro zvýšení účinnosti byly zařazeny i metody pro odstranění nízkofrekvenčního šumu.

3.1.2 Lightweight Acoustic Detection

Roku 2014 přichází autoři článku *Lightweight Acoustic Detection of Logging in Wireless Sensor Networks* [14] s řešením implementace detekčního algoritmu v rámci výpočetně omezeného systému. Pro extrakci příznaků testovacích dat byla využita autokorelační funkce, a to z důvodu redukce výpočetní náročnosti FFT s ohledem na omezený výkon jednotlivých hardwarových částí návrhu. Použitou funkci pro vstupní signál s_n o délce $N = 256$ vzorků lze vyjádřit jako:

$$R(m) = \sum_{n=0}^{|s|} (s_n - \overline{s}) \cdot \left(s_{((n+m) mod |s|)} - \overline{s}\right). \tag{3.1}$$

Z této funkce byla dále zjištěna lokální maxima, průměrná vzdálenost od lokálních maxim a byla vypočtena krátkodobá energie vstupního signálu. Tyto a další parametry byly použity jako vstupní data algoritmů *rozhodovacích stromů* – (*Decision trees*). V tomto případě byly použity Decision Tree (ADT), Best-First decision tree, Decision Stump, J48 (C4.5), J48Graft (grafted C4.5), LogitBoost Alternating Decision Tree (LADTree), Random Forest, Random Tree, Reduced-Error Pruning Tree a Simple Classification and Regression Tree (CART) [14].

Použitý dataset tvořilo 570 reálných nahrávek motorové pily ze vzdáleností 5 – 60 m a nahrávky okolního prostředí v poměru 60 : 40. Testovací data byla pro potřebu algoritmů nejprve podvzorkována na hodnotu vzorkovací frekvence 8333 Hz s bitovou hloubkou 10 bit. Následně byla velikost dat opětovně snížena odfiltrováním hluboko-frekvenčních složek.

Algoritmy byly porovnány z hlediska úspěšnosti na testovací sadě a z hlediska falešně pozitivních a falešně negativních klasifikací. Nejlepších průměrných výsledků dosahoval algoritmus ADT. Klasifikace zaznamenala 84,9% úspěšnost s 2 % falešně pozitivních a 13 % falešně negativních detekcí. K ověření samotné implementace byl využit částečně syntetický testovací soubor a nejlepších výsledků vykazovala varianta s třemi zapojenými senzory.

3.1.3 Chainsaw Detection Using One-Class Kernel

Autoři práce *A Framework for Chainsaw Detection Using One-Class Kernel and Wireless Acoustic Sensor Networks into the Amazon Rainforest* [15] z roku 2016 využívají pro reprezentaci signálu koeficienty MFCC. Databáze je tvořená nahrávkami

motorové pily se vzorkovacím kmitočtem 44,1 kHz o délce 1 s, nahrávkami pozadí a ostatními průmyslovými zvuky pro lepší zamezení falešných detekcí. Nahrávek pozitivní třídy (motorové pily) bylo pořízeno 570 a dataset negativních nahrávek čítal přes 10000 vzorků. V rámci předzpracování signálu byl použit pre-emfázový filtr a váhování Hammingovým oknem, dále byly vypočítány melovské koeficienty s použitím různě velké banky filtrů.

Pro klasifikaci byla využita jednoduchá třívrstvá neuronová síť obsahující *radiálně bazickou funkci* (RBF) ve skryté vrstvě. Klasifikace výstupní vrstvy byla spočtena vztahem:

$$Y(t) = \frac{1}{n\sigma} \sum_{i=1}^{n} exp\left(\frac{-\|z - x_i\|^2}{2\sigma^2}\right), \tag{3.2}$$

kde z představuje neznámou matici MFCC, x_i je matice MFCC vzorku třídy obsahující zvuky motorové pily a σ šířka pásma filtru bazické funkce.

Výsledky byly hodnocené pomocí křivky *AUC-ROC*, jež je ukazatelem poměru falešně negativních a pozitivních detekcí. Práce se v rámci testování zabývá ideálním nastavením šířky kernelu σ a počtu melovských filtrů k. Nejlepších výsledků bylo dosaženo pro nastavení $\sigma = (0, 1), k = 14$, a to s úspěšností klasifikace 98 %.

3.2 Další práce v oblasti zvukové klasifikace

V této části je rozebráno několik článků, jež jsou zaměřeny na obdobný problém audio klasifikace. Ne vždy se jedná o řešení klasifikace jednoho určitého zvuku, jako v případě detekce motorové pily, nicméně rešerše použitých technik a jejich výsledků je pro tuto práci taktéž přínosná. Přístup řešení zde popsaných úloh je ve většině případů aplikovatelný i v případě tématu této práce.

3.2.1 Gunshot Detection Using Convolutional Neural Networks

Prvním článkem je práce *Gunshot Detection Using Convolutional Neural Networks* [16] z roku 2020. Jejím cílem je představení algoritmu schopného detekovat přicházející zvuk výstřelu z pistole v rušném prostředí. Jedná se tedy o shodný problém, jako v případě detekce motorové pily, pouze s využitím rozdílného datasetu.

V tomto případě byla jako databáze pozitivní třídy použita veřejná knihovna *The Free Firearm Sound Library - Expanded Edition* obsahující více než 1000 nahrávek výstřelů. Pro negativní třídu pozadí byl použit dataset *UrbanSound8K*, jež obsahuje přes 8000 nahrávek zvuků města, jako jsou zvuky ulice, štěkot psa, dětský pláč a podobně. Celkový dataset byl tvořen 7000 audio vizualizacemi a byl rozdělen v poměru 60 : 20 : 20 na tréninková, validační a testovací data. V této práci byly porovnávány výsledky při použití samotného spektrogramu a kombinace

spektrogramu, MFCC a matice sebe-podobnosti, jakožto reprezentace audio signálu pro trénování sítě. V rámci předzpracování signálu pro jeho transformaci byl signál nejprve segmentován na úseky o délce 256 vzorků a poté váhován Hammingovým oknem. Pro výpočet MFCC byla použita FFT o délce 2048 vzorků a 20 filtrů mel banky.

Pro účely klasifikace byly testovány tři různé architektury neuronových sítí, předtrénované sítě *VGG16* a *InceptionV3* a síť *ResNet18* natrénovaná zmíněným datasetem, všechny implementované pomocí knihoven *TensorFlow* a *Keras* jazyka *Python*. Rozměr vstupních dat se lišil v závislosti na použité síti 224×224 a 299×299. Velikost dávky byla 10 vzorků a délka učení 5 epoch. Síť *VGG16* obsahovala konvoluční vrstvy s jádry o velikosti 3×3, síť *InceptionV3* 1×1, 3×3 a 5×5. Síť *ResNet18* je síť tvořená reziduálními spojeními bloků, za účelem minimalizace problému označovaného jako *mizející gradient*, který způsobuje pomalé trénování sítě [17].

Z testování vyplývá, že nejlepších výsledků v kombinaci s nejkratší výpočetní dobou dosahovala síť *ResNet* za použití kombinace spektrogramu, MFCC a matice sebe-podobnosti, a to 99,14 % úspěšnosti. V případě podvzorkování vstupního signálu na hodnotu vzorkovacího kmitočtu 8 kHz klesla úspěšnost pouze na 99,07 %, ovšem za značně stížené výpočetní náročnost, a tím i zkrácené doby potřebné pro klasifikaci. Zvolená metoda, popsaná v tomto článku, vykazuje oproti srovnávaným metodám především mnohem nižší procentuální zastoupení falešně pozitivních klasifikací, což byl jeden z hlavních cílů, které si autoři vytyčili.

Testování optimalizovaných sítí probíhalo pomocí reálných nahrávek prostředí obsahujících několik výstřelů. Algoritmus tyto nahrávky nejprve rozdělil na úseky o délce 1 s a ty pak postupně podrobil klasifikaci. Výstupem tohoto programu byl zobrazený signál v časové oblasti s vyznačenými úseky, určenými jako výstřel. Dále bylo řešení testováno i na obdobných zvucích impulsivního charakteru (buben, otevření dveří nebo rozbití skla). Zde z výsledků vyplývá, že účinnost algoritmu závisí na individuálním nastavení rozhodovací úrovně klasifikátoru a jeho citlivosti.

3.2.2 Audio Event Classification Using Deep Neural Networks

Článek z roku 2013 *Audio Event Classification Using Deep Neural Networks* [18] se zabývá návrhem rozpoznávacího systému pro klasifikaci ruchů města. Dataset byl vytvořen z nahrávek získaných z veřejně přístupné knihovny *FreeSound.org*, a poté manuálně opatřen „labely" patřičné třídy nebo více tříd. Pro účely této práce byly zkoumány následující čtyři typy ruchů: dav lidí, doprava, potlesk (jásot nebo křik) a hudba zaznamenaná v rušném prostředí. Použité nahrávky byly ve formátu jednokanálových audio souborů podvzorkovaných na hodnotu 22 kHz. Celková použitá délka datasetu byla 228 minut.

Nahrávky byly nejprve rozděleny na úseky o délce 5 s a 50% překryvu. Z těchto segmentů bylo následně extrahováno 192 příznaků. Segment byl rozdělen pomocí rámců o délce 64 ms s 50% překryvem a extrakce příznaků probíhala třemi různými způsoby. Prvních 64 hodnot tvořily průměrné hodnoty pro jednotlivé MFCC celého segmentu. Dalších 64 příznaků představovalo jejich směrodatné odchylky. Pro poslední část byla nejprve vypočtena směrodatná odchylka logaritmického spektra rámce, a až poté byla váhována bankou mel filtrů. Tato kombinace vykazovala dle autorů nejlepší výsledky v rámci testování.

Jako klasifikátor byla v této práci použita dopředná vícevrstvá neuronová síť. V rámci zjednodušení učení byla síť vrstvu po vrstvě předtrénována pomocí modelu *Boltzmannova stroje*, což je speciální model stochastické Hopfieldovy sítě se skrytou vrstvou. Po fázi předtrénování jsou upraveny váhy sítě pomocí procesu zpětného šíření chyby – *back-propagation*. Výstupní vrstva neuronové sítě byla tvořena počtem neuronů odpovídajícím počtu klasifikačních tříd. Z důvodu nevyváženosti datasetu bylo použito rozdílné váhování pro jednotlivé třídy. Učení probíhalo po dobu 200 epoch a nejlepší výsledky vykazovala architektura tvořená třemi skrytými vrstvami o 200 neuronech.

Klasifikace pomocí výše popsané DNN byla porovnána s výsledky klasifikace pomocí metody *podpůrných vektorů* (*Support Vector Machines* – SVM) s použitím *radiální bazické funkce* jádra (RBF) a pomocí metody *gaussovských smíšených modelů* (GMM).

Testování probíhalo pomocí výše zmíněného datasetu, který byl vždy náhodně rozdělen na trénovací a testovací data v poměru 80 : 20. Jednotlivé metody klasifikace byly testovány padesátkrát pomocí *křížové validace*. Výsledky klasifikací byly porovnány pomocí ukazatele *EER* (*Equal Error Rate*), který byl určen průměrnou hodnotou pro všechna testování pro každou třídu zvlášť. Dále byla tato hodnota doplněna o hodnotu směrodatné odchylky.

Nejhorších výsledků oproti zbylým metodám bylo dosaženo pomocí klasifikátoru GMM. Představená metoda založená na DNN vykazovala ve srovnání se SVM lepších výsledků klasifikace pro všechny testované třídy s výjimkou třídy „hudba", kde byl nejúspěšnější právě model SVM. Nejlepších výsledků bylo dosaženo pomocí kombinace obou zmíněných modelů. Průměrná chyba pro nejúčinnější představené řešení byla rovna $14,76 \pm 0,54$ a nejlepších výsledků bylo dosaženo v případě klasifikace zvuků třídy „potlesk" s výslednou odchylkou $7,89 \pm 0,79$.

4 Návrh řešení

V této kapitole je představen návrh systému detekce zvuku motorové pily v přirozeném prostředí. Systém byl realizován v prostředí Matlab ve verzi R2019b s využitím toolboxů *Signal Processing Toolbox* a *Deep Learning Toolbox*. Druhý zmíněný toolbox je nástrojem pro tvorbu neuronových sítí v prostředí Matlab a pro fungování programu je nezbytná jeho instalace.

Nejprve je v této kapitole popsána tvorba datasetu a předzpracování signálu s tím spojené. V další části jsou popsány navržené architektury neuronových sítí spolu s jejich nastavením. Nejprve je popsáno řešení za pomoci konvoluční neuronové sítě, poté druhé řešení využívající kombinaci CNN a rekurentní neuronové sítě LSTM. Závěr této kapitoly je věnován popisu detekčního programu, který byl vytvořen pro potřeby testování.

4.1 Dataset

Pro tvorbu datasetu byly použity reálné nahrávky těžby motorovou pilou dodané ve formátu WAVE (*.wav). Nahrávky byly pořízeny pomocí rekordéru *TASCAM DR-05X*, jakožto mono nahrávky se vzorkovací frekvencí 44,1 kHz a bitovou hloubkou 16 bit. Jednalo se o několik nahrávek zvuků těžby jedné nebo dvou motorových pil z různých vzdáleností o přibližné celkové délce 120 minut. Tyto nahrávky byly nejprve pomocí DAW Logic Pro X sestříhány do podoby obsahující čistě zvuk pily.

Pro zvuky pozadí, obsažené v druhé části datasetu, byly použity veřejně dostupné audio nahrávky zvuků lesa „*SlowRadio: zvuky jihočeské přírody / sounds of south bohemia nature – 11 hodin / 11 hrs*" a „*SlowRadio: jarní déšť v jižních Čechách / south Bohemia rain in spring*", kterých bylo použito přibližně 8 hodin záběru. Nahrávky byly pořízeny za použití mikrofonu *Sennheiser MKE 600* a externí zvukové karty *Steinberg UR242* a následně stažené ve formátu WAVE(*.wav), 48 kHz, 16 bit. V rámci vylepšování řešení navrženého v semestrální práci byl dataset doplněn o další nahrávky motorové pily a dvě hodiny záznamu zvuků lesa z veřejně dostupných nahrávek. A dále pak o dvě hodiny záznamu rušné ulice pořízené mobilním telefonem. Tato část datasetu byla doplněna z v důvodu eliminace falešně pozitivních detekcí způsobených motorovými stroji, hovorem a dalšími zvuky.

4.1.1 Předzpracování datasetu

Veškerá zvuková data byla nejprve podvzorkována na hodnotu 16 kHz a případně upravena do jednokanálové podoby. Poté byly nahrávky rozděleny na úseky o délce 3 s a následně normalizovány. V případě nahrávek pozadí byl použit i algoritmus

detekující maximální hodnotu amplitudy segmentu, aby nebyl dataset naplněn nahrávkami obsahujícími pouze ticho.

V rámci testování byly pro reprezentaci signálu použity koeficienty MFCC, jejichž výpočet probíhal pomocí vlastní funkce `mfcc.m`. Zde je nejprve volána funkce `segmentace.m`, která rozdělí vstupní signál na segmenty odpovídající zadané velikosti okna a překryvu. Poté je vypočteno modulové spektrum, banka mel filtrů pomocí funkcí `melbank.m` a `mel2Hz.m` a následně je dokončen výpočet samotných koeficientů.

Vstupní parametry transformační funkce byly zvoleny následovně:

- *velikost časového okna*: 1600 vzorků,
- *překryv okna*: 50 %,
- *zvolené časové okno*: Hamming,
- *počet mel filtrů*: 50,
- *počet vzorků FFT*: 800.

Výstupem funkce **mfcc.m** je poté matice koeficientů o rozměru 50×59. Tyto koeficienty jsou následně zobrazeny a uloženy jako obrázek ve formátu PNG (*.png). Příklad grafického zobrazení MFCC pro nahrávku pily a pozadí je vidět na obrázku 4.1.

Výsledný dataset použitý pro trénování a testování neuronové sítě tvořilo 672 příznakových map MFCC motorové pily a 7680 pozadí. Ty byly poté pomocí funkce toolboxu *Deep Learning* náhodně rozděleny na trénovací, validační a testovací data v poměru 60 : 20 : 20.

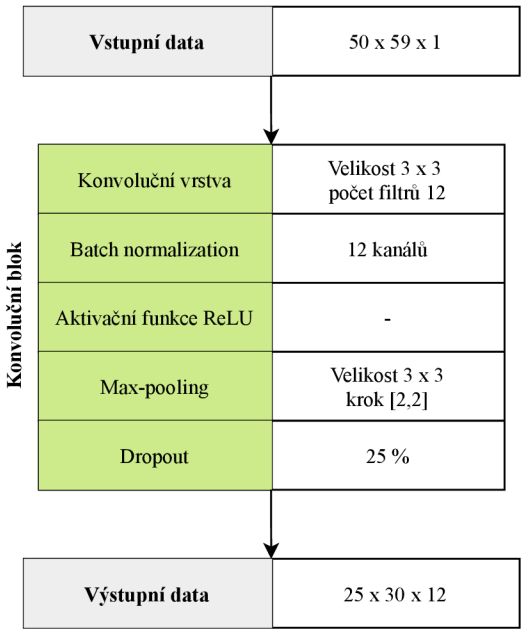
(a) (b)

Obr. 4.1: Příklad MFCC zobrazení nahrávky motorové pily (a) a pozadí (b).

4.2 Konvoluční neuronová síť

Jak již bylo zmíněno výše, konvoluční neuronové sítě dosahují v oblasti rozpoznávání zvuku velmi dobrých výsledků, proto byly využity i v případě této práce. Síť byla složena z konvolučních bloků obsahujících *konvoluční vrstvu* s určitým množstvím filtrů, poté vrstvu *batch normalization*, aktivační vrstvu *ReLU*, *max-pooling vrstvu* a *dropout* vrstvu. Ze vstupních dat takovéhoto bloku jsou nejprve extrahovány příznaky pomocí několika konvolučních jader konvoluční vrstvy, čímž dochází k navyšování třetího rozměru dat. Výstupní data konvoluce jsou poté normalizována před vstupem do aktivační funkce *ReLU* vrstvou *batch normalization*. Následně je provedena redukce rozměru dat pomocí *pooling funkce* metodou *max-pooling* a na závěr je snížen počet aktivních vazeb pomocí *dropout* vrstvy za účelem ochrany sítě proti přeučení. Ukázka prvního konvolučního bloku sítě a jeho vliv na vstupní data je zobrazena na obrázku 4.2.

Nejlepších výsledků síť dosahovala v případě architektury obsahující pět kon-



Obr. 4.2: Ukázka použitého konvolučního bloku.

volučních bloků, přičemž čtvrtý blok neobsahoval *pooling vrstvu* z důvodu kompatibility rozměru dat a z posledního bloku byla během testování odstraněna vrstva *dropout*, jež způsobovala velké výkyvy ztrátové funkce. Tento blok byl přiveden na plně propojenou neuronovou vrstvu o dvou neuronech. Na výstup této vrstvy byla aplikována funkce *softmax* a byla provedena klasifikace pomocí váhované klasifikační vrstvy. Váhování výstupů bylo v tomto případě zavedeno z důvodu nevyrovnaného obsažení datasetu. Třída obsahující méně vzorků je tedy zvýhodněna. Toto váhování

probíhá za pomoci křížové ztrátové funkce definované pro vypočítanou hodnotu predikce Y a žádaný výstup T vztahem:

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K} w_i T_{ni} \log\left(Y_{ni}\right), \tag{4.1}$$

kde N je počet vzorků, K počet tříd a w vektor vah každé třídy [12].

Ukázka změn velikostí vstupních dat po průchodu jednotlivými vrstvami sítě je pro vstupní vzorek o velikost $50 \times 59 \times 1$ zobrazena na obrázku 4.3.

Blok	Výstupní rozměr	Nastavení vrstev	
Conv1	25 x 30 x 12	Konvoluční vrstva	Velikost 3 x 3 počet filtrů 12
		Max-pooling	Velikost 3 x 3 krok [2,2]
		Dropout	25 %
Conv2	13 x 15 x 24	Konvoluční vrstva	Velikost 3 x 3 počet filtrů 24
		Max-pooling	Velikost 3 x 3 krok [2,2]
		Dropout	25 %
Conv3	7 x 8 x 48	Konvoluční vrstva	Velikost 3 x 3 počet filtrů 48
		Max-pooling	Velikost 3 x 3 krok [2,2]
		Dropout	25 %
Conv4	7 x 8 x 48	Konvoluční vrstva	Velikost 3 x 3 počet filtrů 48
		Dropout	25 %
Conv5	1 x 8 x 48	Konvoluční vrstva	Velikost 3 x 3 počet filtrů 48
		Max-pooling	Velikost 7 x 1 krok [1,1]
Fully Connected	**1 x 1 x 2**	Softmax	
		WeightedClassification	

Obr. 4.3: Přehled rozměrů výstupních dat jednotlivých vrstev.

4.2.1 Nastavení parametrů sítě CNN

Nastavení parametrů sítě probíhalo převážně experimentálně v závislosti na dosažené úspěšnosti algoritmu. Jako optimalizační algoritmus byl zvolen model *Adam*. Parametr ovlivňující rychlost učení *learning rate* byl testováním nastaven na hodnotu $0,0003$. V případě použití příliš vysoké hodnoty tohoto parametru často dochází k náhodným validačním hodnotám a ztrátová funkce má tendenci divergovat. Příliš nízká hodnota tohoto parametru naopak způsobuje pomalé učení sítě [2].

Dalším parametrem, jehož nastavení byla věnována zvýšená pozornost, byl parametr *batch-size*, který udává počet vzorků, které algoritmus zpracovává současně

během jedné iterace. Nejlepších výsledků bylo dosaženo pro velikost dávky 30 vzorků, přičemž k validaci aktuálního stavu sítě docházelo po 50 iteracích programu. Tímto nastavením bylo dosaženo rovnoměrnějšího poklesu validační funkce a postupného zvyšování úspěšnosti. Zároveň trénování po menších dávkách je méně výpočetně náročné. Při použití větší velikosti dávky docházelo ke skokovým změnám validační funkce a dosažení 100% úspěšnosti již po první nebo druhé epoše, nicméně takto natrénovaná síť vykazovala větší množství falešně pozitivních detekcí.

Epochou je míněn úsek učení, při němž jsou algoritmu předloženy všechny vzorky trénovací množiny. Maximální počet epoch byl pro výše zmíněné parametry nastaven na 8. Dále byla také použita funkce *shuffle*, díky které byla trénovací data po každé epoše náhodně promíchána. Stejný proces byl aplikován i na validační data před každou validací.

4.3 Konvoluční neuronová síť + LSTM síť

Druhým testovaným návrhem řešení byla kombinace výše představené architektury konvoluční neuronové sítě a rekurentní LSTM sítě. Toto řešení bylo zvoleno za účelem zkombinování schopnosti sítí CNN nacházet souvislosti na krátkých časových úsecích a paměti sítí LSTM schopné zpracovávat delší časové posloupnosti vstupních dat [20].

Níže popsané řešení tedy spočívá v kombinaci dvou samostatných neuronových sítí. Konvoluční neuronové sítě o pěti blocích a jednoduché sítě LSTM. Konvoluční síť zde slouží pro převod dvourozměrných vstupních dat v podobě MFCC koeficientů na jednorozměrný vektor příznaků, který je následně použit jako vstupní prvek sekvenční vstupní vrstvy LSTM sítě. Aby byla síť schopna kombinovat krátkodobé příznaky s těmi globálními, byl pro účely této architektury upraven použitý dataset následujícím způsobem. Opět byly pro grafickou reprezentaci audio signálu použity melovské kepstrální koeficienty. Na rozdíl od samostatné CNN však byly nahrávky rozděleny na úseky o délce 1 s s překryvem 50 %. Takto bylo možné popsat úsek o délce 3 s pěti spektrogramy. Pro dosažení vyššího frekvenčního rozlišení vzhledem k třetinové délce úseku oproti předchozímu návrhu byly parametry transformační funkce zvoleny následovně:

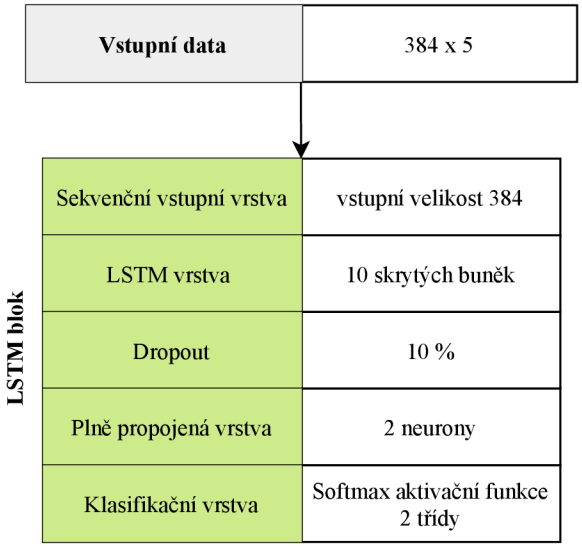
- *velikost časového okna*: 500 vzorků,
- *překryv okna*: 50 %,
- *zvolené časové okno*: Hamming,
- *počet mel filtrů*: 50,
- *počet vzorků FFT*: 1600.

Výstupem této funkce pak byly matice koeficientů o rozměrech 50×63. Po transformaci nahrávek pak vznikl dataset čítající 34000 vzorků, z toho 8800 vzorků motorové

pily. Tato data byla opět rozdělena na trénovací a validační a použita pro natrénování konvoluční neuronové sítě, popsané v 4.2, upravené pro novou velikost vstupních dat.

Následně bylo nutné transformovat dataset pro potřeby LSTM sítě. K tomu byla využita funkce `activations.m`, díky které je možné získat výstupní data zvolené vrstvy neuronové sítě. Pomocí již předtrénované konvoluční sítě pomocí 1 s nahrávek byl pro každý vzorek datasetu získán výstup poslední max-poolingové vrstvy v podobě vektoru příznaků o rozměru 384×1. Výstupy pro pět po sobě jdoucích vzorků, představujících úsek o délce 3 s, pak byly uloženy do matice o rozměru 384×5, kde 384 představuje počet příznaků a 5 délku sekvence. Tato matice byla opatřena jedním *labelem* (informací o žádaném výstupu) a uložena jako jeden prvek datasetu pro trénování LSTM sítě. Takto transformovaný dataset poté tvořilo 6700 sekvencí. Tato data byla opět náhodně rozdělena na trénovací, validační a testovací v poměru $60 : 20 : 20$.

Testovaná architektura byla tvořena jednou LSTM vrstvou s 10 skrytými paměťovými buňkami. Za LSTM vrstvou byla použita *dropout* vrstva s hodnotou 10 %, jako prevence proti přeučení sítě. Na závěr byla použita plně propojená neuronová vrstva o dvou neuronech, aktivační funkce *softmax* a klasifikační vrstva. Ukázka celé architektury testované LSTM sítě je zobrazena na obrázku 4.4.



Obr. 4.4: Architektura LSTM sítě.

4.3.1 Nastavení parametrů sítě LSTM

Parametry učení pro předtrénovanou CNN, použitou pro transformaci datasetu byly zvoleny obdobné jako v případě parametrů konvoluční sítě, popsaných v části 4.2.1.

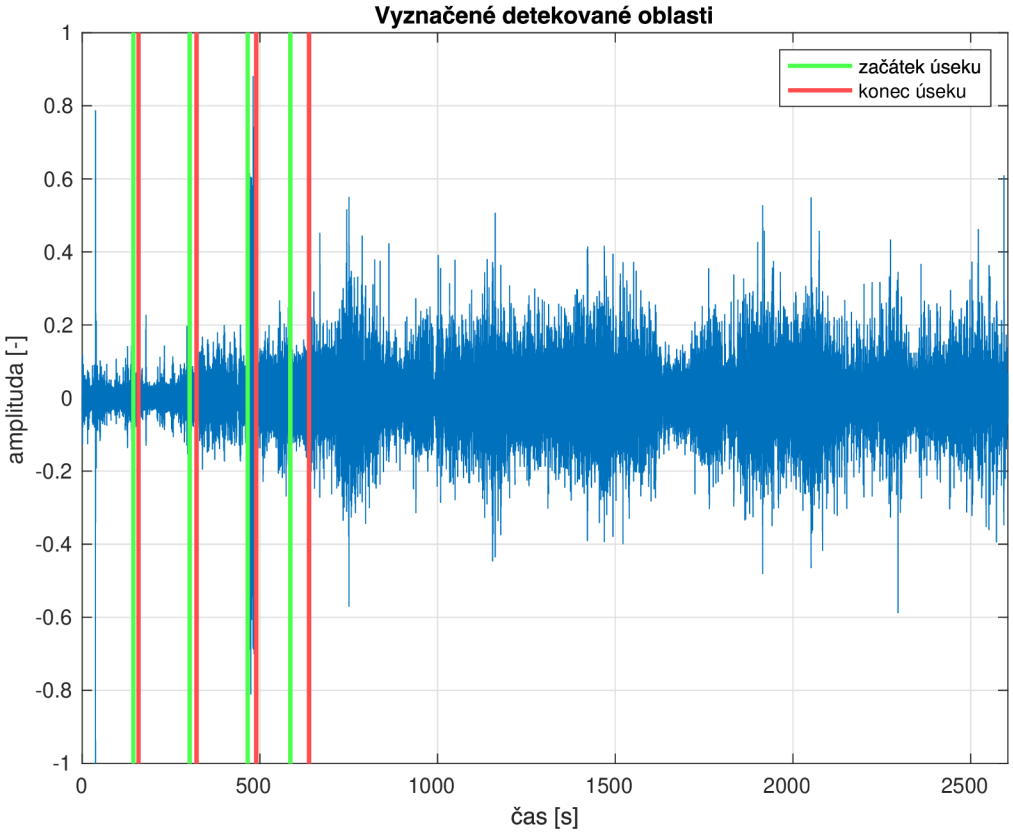
Pro LSTM síť byly použity následující parametry. Parametr *learning rate* byl 0,003 a jako optimalizační algoritmus byl opět zvolen *Adam*. Nejlepší výsledky vykazovala síť při nastavení parametru *batch-size* na hodnotu 30 vzorků s validací po 50 iteracích. Maximální počet epoch byl zvolen 150, kdy už nedocházelo ke zlepšování hodnot ztrátové funkce. V případě LSTM sítě se pomalé učení ukázalo jako nejúčinnější. Pro dosažení co nejnižších hodnot ztrátové funkce byla po 130 epochách automaticky snížena hodnota *learning rate*, a to na desetinu. Opět byla použita funkce *shuffle* pro náhodné promíchání trénovacích a validačních dat po každé epoše.

4.4 Detekční program

Pro účely testování byl navržen detekční program **detection.m**, schopný analyzovat zvukové nahrávky o velké délce. Vstupní parametry funkce jsou struktura obsahující natrénovanou neuronovou síť `net` a případný vstupní signál `data`. Pokud není vstupní signál zadán, je zobrazeno okno pro načtení analyzovaného signálu a jeho uložení do proměnné. Výstupními parametry funkce jsou proměnná `PILA` a hodnota vzorkovací frekvence uložená v proměnné `fs`.

Analyzovaný signál je nejprve upraven do podoby mono a podvzorkován na hodnotu 16 kHz. V další části funkce jsou nastaveny parametry pro výpočet MFCC, které jsou shodné jako v případě přípravy datasetu popsané v části 4.1.1. Dále následuje smyčka, ve které je signál analyzován po segmentech o délce 3 s. Jsou postupně vypočteny MFCC a provedena klasifikace segmentů pomocí natrénované sítě.

Detekované úseky jsou uloženy do proměnné `PILA` pro pozdější přehrání a index začátku a konce úseku je uložen do proměnné `IND`. V konzole je vypsána časová informace o detekovaném úseku. Na základě výsledků klasifikace je následně vykreslen graf zobrazující vstupní signál v časové oblasti s vyznačenými detekovanými úseky. Pokud program žádné detekce nezaznamená, je pouze vypsána informační hláška. Příklad grafického výstupu programu je zobrazený na obrázku 4.5.



Obr. 4.5: Ukázka výstupu detekčního programu s vyznačenými oblastmi detekce.

5 Výsledky testování

V této kapitole budou nejprve představeny prvotní výsledky trénování a testování navrženého řešení pro různá nastavení parametrů neuronové sítě. Algoritmus byl nejprve trénován a testován pomocí zmenšené verze datasetu popsaného v části 4.1. Výsledky byly poté ověřeny na testovacích nahrávkách pomocí detekčního programu představeného v sekci 4.4.

Pro porovnání jednotlivých nastavení byly použity křivky průběhů úspěšnosti a ztrátové funkce trénování a validace po dobu učení. Dalším použitým parametrem byly křivky *ROC*, které udávají, s jakou jistotou algoritmus přiřadil vzorek dané třídě. Čím více se křivka blíží bodu $[0, 1]$, tím lze klasifikaci považovat za důvěryhodnější [2].

Podobnou evaluační úlohu zastává i *matice záměn* (*Confusion Matrix*). V tomto případě se jedná o matici zobrazující rozložení jednotlivých typů detekcí dle pozitivity a pravdivosti. Další část je věnována testování funkčnosti navrženého detekčního programu a úspěšnosti první verze sítě.

V následující části jsou představeny průběhy trénování tří testovaných neuronových sítí. Nejprve vylepšená CNN3s, navržená v předchozí práci, a poté řešení využívající kombinaci CNN1s a LSTM sítě. Pro srovnání byla zařazena i samotná CNN1s využitá pro transformaci dat pro účely LSTM sítě.

Poslední část je věnována srovnání výsledků těchto sítí při klasifikaci testovacích nahrávek pomocí detekčního programu.

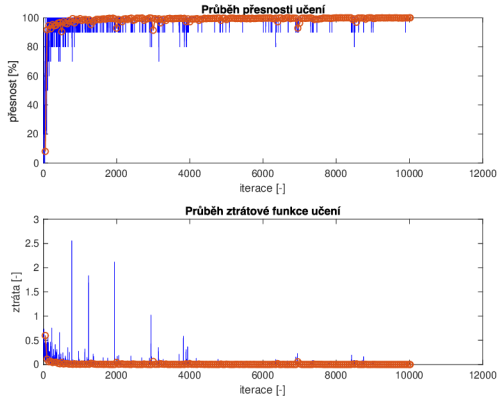
5.1 Trénování

Po empirickém nastavení vstupních parametrů a architektury sítě byla provedena měření pro různá nastavení. Jako jeden z parametrů nejvíce ovlivňující výslednou úspěšnost programu se ukázala velikost vstupní tréninkové dávky.

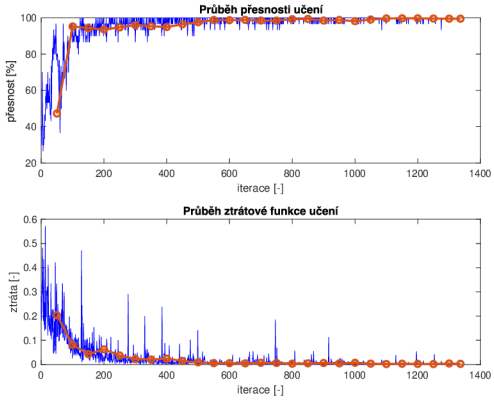
Na obrázku 5.1 jsou zobrazeny průběhy učení pro tři různá nastavení parametru *batch-size*, udávajícího počet současně analyzovaných vstupních vzorků datasetu. Jak je patrné z jednotlivých grafů, ve všech případech dosáhl algoritmus celkové úspěšnosti přes 90 %, nicméně bylo nutné kompenzovat velikost dávky délkou doby trvání učení. V případě dávky o velikosti 10 vzorků bylo k dosažení úspěšnosti převyšující 95 % nutno provést 20 tréninkových epoch, zatímco v případě dávky o 30 vzorcích algoritmu stačilo pouhých 8 epoch, jak je patrné z grafu 5.1a a 5.1b. Při zvolení větší dávky, jako v případě 5.1c, docházelo k poměrně rychlému nárůstu úspěšnosti již během první epochy, nicméně k dosažené stejné úspěšnosti a hodnotě ztrátové funkce jako v případě 5.1b bylo taktéž nutné delší učení.

Vysoké skokové hodnoty ztrátové funkce učení přisuzuji efektu funkce *mini-batch*.

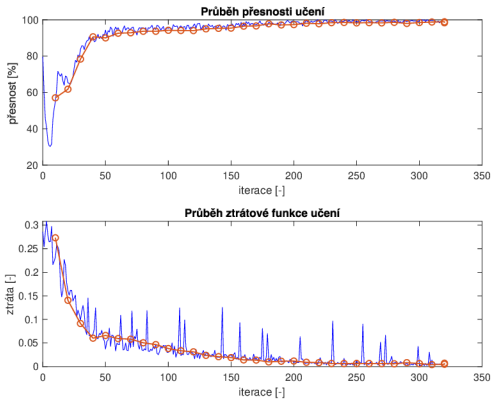
V případě posledního „batche" dochází k učení s neúplnou dávkou vzhledem k velikosti datasetu, a to může způsobovat nárůst ztráty.



(a) Batch-size 10 vzorků.



(b) Batch-size 30 vzorků.



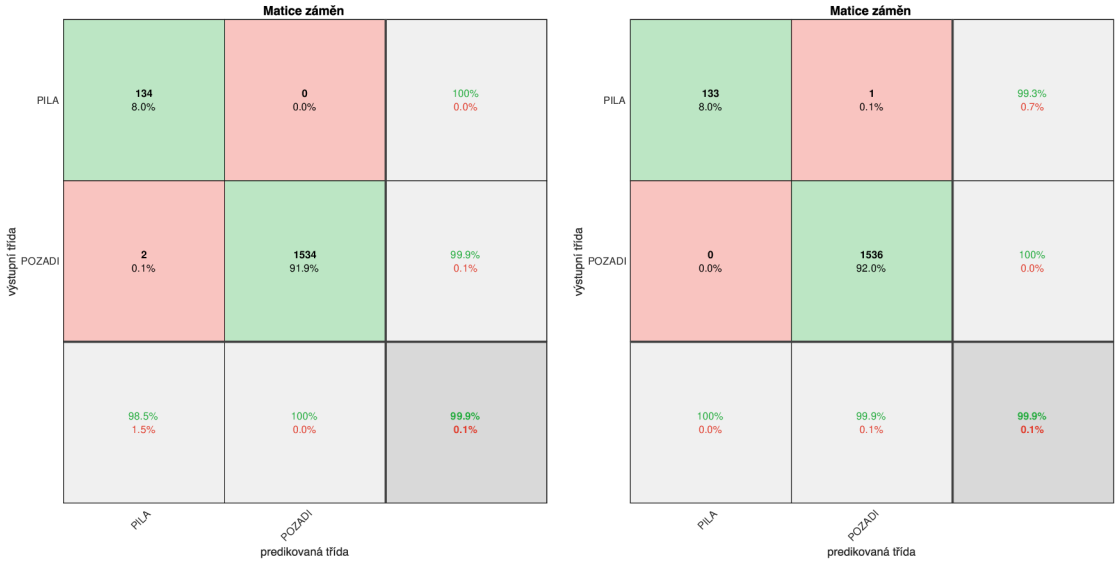
(c) Batch-size 300 vzorků.

Obr. 5.1: Porovnání průběhů trénování pro rozdílná nastavení parametru *batch-size*.

5.2 Srovnání křivek ROC a Matic záměn

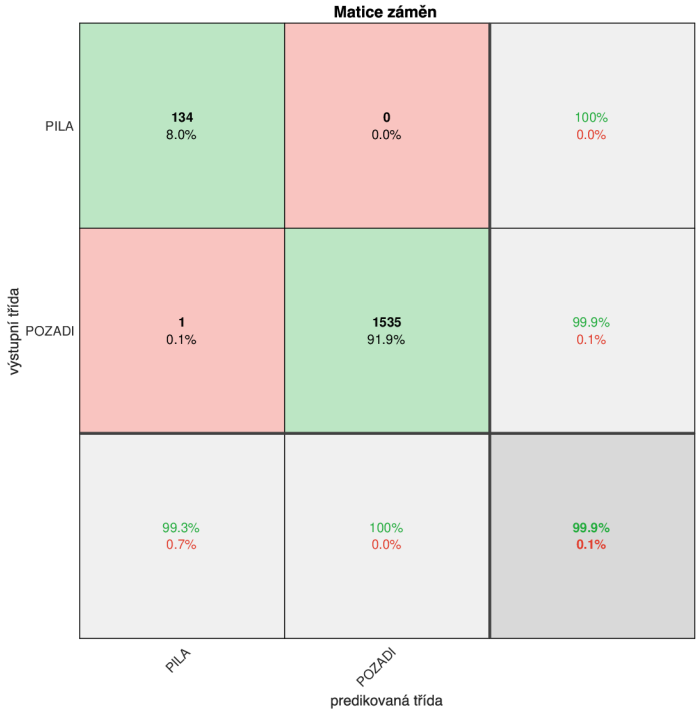
Po úspěšném natrénování sítě byla provedena klasifikace testovací množiny, kterou tvořilo 20 % celkové velikosti datasetu. Úspěšnost klasifikace testovací sady se ve většině případů takřka shodovala s validační přesností.

Výsledky této klasifikace pro nastavení, jejichž průběhy učení jsou zobrazeny v grafech 5.1, je možné porovnat pomocí matic záměn na obrázku 5.2. Zde je patrné, že algoritmus pro všechna nastavení vykazoval takřka totožné, velmi dobré výsledky.



(a) Batch-size 10 vzorků.

(b) Batch-size 30 vzorků.



(c) Batch-size 300 vzorků.

Obr. 5.2: Porovnání výsledků klasifikace testovací sady pomocí matic záměn pro rozdílná nastavení parametru *batch-size*.

5.2.1 Evaluace klasifikace

Kromě grafického zobrazení úspěšnosti klasifikace pomocí matic záměn se obvykle využívají evaluační nástroje pro hodnocení klasifikace, jakými jsou *přesnost*, *spolehlivost* nebo *senzitivita* klasifikátoru. Výpočet těchto ukazatelů je prováděn z výsledků klasifikace testovací sady na základě počtu vzorků jednotlivých typů klasifikace. Výsledek klasifikace může s ohledem na požadovaný výstup nabývat čtyř hodnot, které jsou zobrazeny tabulkou 5.1 [2].

Pomocí již výše zmíněné matice záměn je poté možné graficky zobrazit výsledek

Tab. 5.1: Typy výsledků klasifikace pro výpočet dalších parametrů.

Zkratka	Název	Popis
TP	*True Positive*	Klasifikovaný vzorek zařazen správné třídě.
TN	*True Negative*	Klasifikovaný vzorek nezařazen nesprávné třídě.
FP	*False Positive*	Klasifikovaný vzorek zařazen nesprávné třídě.
FN	*False Negative*	Klasifikovaný vzorek nezařazen správné třídě.

klasifikace testovací množiny v závislosti na počtu prvků spadajících do těchto typů výstupní klasifikace.

Za parametr *senzitivita* (*sensitivity*) bývá označován takzvaná *míra pravdivé pozitivity* (*True Positive Rate* – TPR). Jeho výpočet je dát poměrem pravdivě pozitivních predikcí a součtu pravdivě pozitivních a falešně negativních predikcí. Tento vztah lze zapsat jako:

$$TPR = \frac{TP}{TP + FN}. \tag{5.1}$$

Dalším používaným ukazatelem je *spolehlivost* (*precision*). Za tu je považován poměr pravdivě pozitivních predikcí a součtu pravdivě pozitivních a pravdivě negativních predikcí. Vyjadřuje spolehlivost klasifikace pozitivní třídy a její odolnost vůči falešných detekcím. Výpočet *spolehlivosti* lze tedy zapsat vztahem

$$Spolehlivost = \frac{TP}{TP + FP}. \tag{5.2}$$

Pro úplnost je možné ještě dodat parametr udávající *míru falešné pozitivity* (*False Positive Rate* – FPR), jehož výpočet je dán rovnicí

$$FPR = \frac{FP}{FP + TN}. \tag{5.3}$$

Nejpoužívanějším parametrem k evaluaci klasifikace je však *přesnost* (*Accuracy* – ACC). Vyjadřuje míru správných predikcí s ohledem na jejich celkový počet. Vztah pro výpočet ACC je následující:

$$ACC = \frac{\text{počet správných predikcí}}{\text{celkový počet predikcí}} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{5.4}$$

V případě testovací množiny čítající 100 vzorků by přesnost klasifikace při 80 správně zařazených vzorcích byla 80 %.

5.2.2 Křivka ROC

Často používaným prostředkem hodnocené kvality klasifikace je křivka ROC (*Receive Operating Characteristic*). Je dána jako závislost parametrů *TPR* a *FPR* pro různé klasifikační prahy [21]. Klasifikační prahy udávají pro binární klasifikátor hodnoty, na základě kterých je rozhodováno o zařazení do jedné ze tříd. Čím více se hodnota vykreslené křivky ROC blíží bodu $[0, 1]$, tím je klasifikátor spolehlivější a rozlišitelnost hustot pravděpodobností obou tříd vyšší. Tento bod tedy odpovídá ideálnímu klasifikačnímu prahu [2].

Parametrem, který udává spolehlivost klasifikátoru pro všechny uvažované klasifikační prahy, je parametr *AUC-ROC* (*Area Under the ROC Curve*). Ten je dán jako integrál pod samotnou křivkou ROC. Výsledkem je tedy jedna hodnota, která udává kvalitu klasifikace daného řešení. Vztah pro výpočet AUC-ROC může vypadat následovně:

$$ROC\text{-}AUC = \int_0^1 ROC\,(p)\,\mathrm{d}p. \tag{5.5}$$

Pro testovaná nastavení konvoluční sítě je podoba křivek ROC možná pozorovat na obrázku 5.3. Zde je patrné, že rozhodování probíhalo s vysokou rozlišitelností a hodnotou AUC-ROC v intervalu od 0,9 do 1, což odpovídá velmi dobré úspěšnosti predikce.

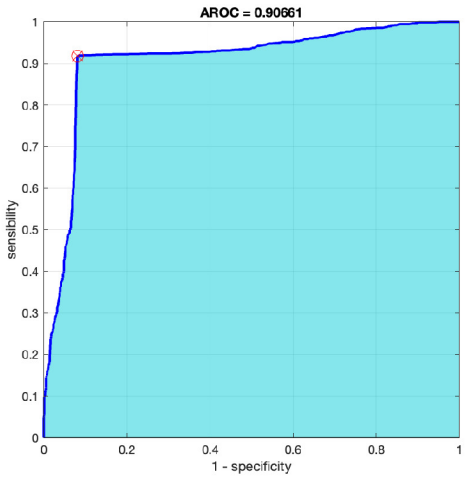
5.3 Prvotní testování

Další testování probíhalo pomocí výše popsaného detekčního programu. Jako testovací zvuky byly použity některé nesestříhané nahrávky těžby použité v datasetu a také nahrávky pozadí s uměle vloženými zvuky pily pomocí DAW.

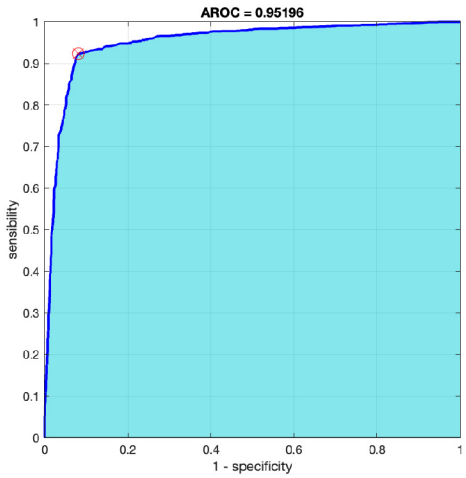
Z testování vyplývá, že i přes vysokou úspěšnost na testovací a validační sadě dochází pro některá nastavení sítě k nepřesné detekci. Především se objevuje značné množství falešných detekcí způsobených například zvuky jako: silný déšť, lidský hlas nebo hučení motoru.

Nejspolehlivějších výsledků dosahovala síť natrénovaná s použitím dávky 30 vzorků po dobu 8 epoch. Porovnání výstupů detektoru pro variantu batch-size = 30 a batch-size = 300 je možné vidět na obrázcích 5.4. Zde jsou zobrazeny detekce testovacího souboru `pozadi-rain.wav`, obsahujícího pouze zvuk lesa za deště a souboru `test1.wav` s uměle vloženými zvuky pily.

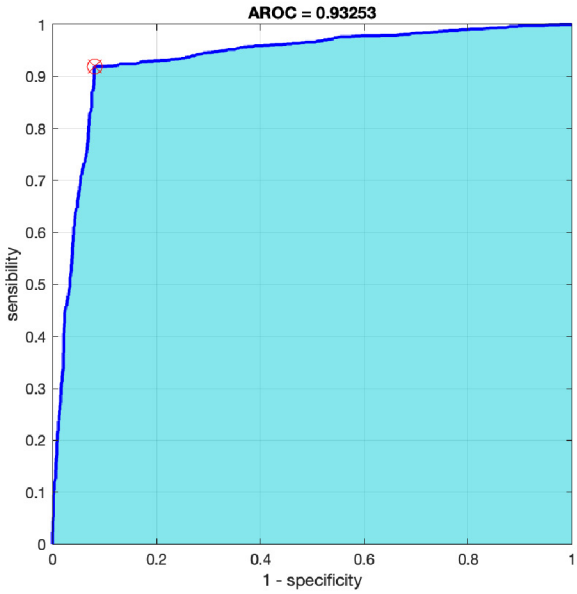
Je patrné, že v případě trénování s větší dávkou docházelo k většímu počtu falešných detekcí u obou vzorků. Oproti tomu druhé nastavení dosahuje velmi dobrých



(a) Batch-size 10 vzorků.



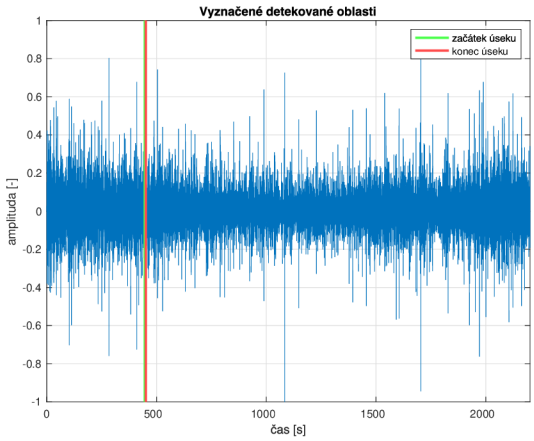
(b) Batch-size 30 vzorků.



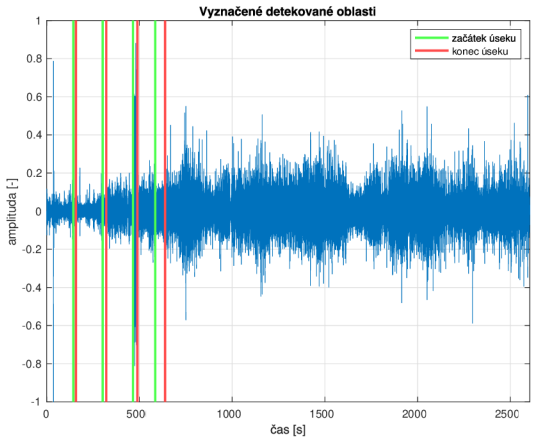
(c) Batch-size 300 vzorků.

Obr. 5.3: Porovnání křivek *ROC* pro rozdílná nastavení parametru *batch-size*.

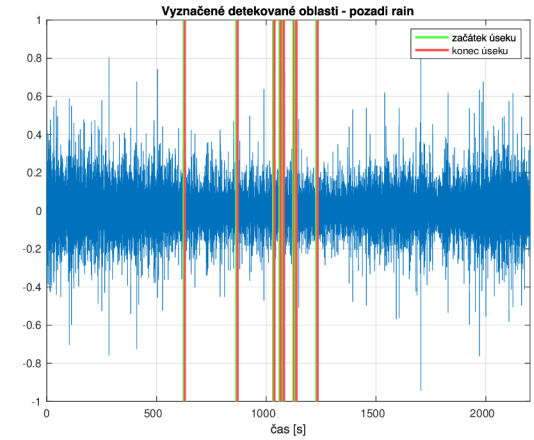
výsledků u obou nahrávek, kde v případě `pozadi-rain` falešně detekuje pouze jeden 3 s úsek na celkové délce nahrávky 23 minut. U nahrávky `test1` jsou správně detekované všechny uměle vytvořené pozitivní úseky.



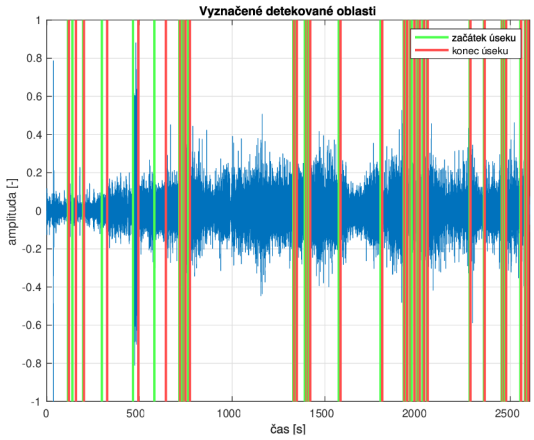
(a) `pozadi-rain.wav` pro batch-size 30.



(b) `test1.wav` pro batch-size 30.



(c) `pozadi-rain.wav` pro batch-size 300.



(d) `test1.wav` pro batch-size 300.

Obr. 5.4: Porovnání výstupu detekčního programu pro dvě testovací nahrávky a různá nastavení parametru *batch-size*.

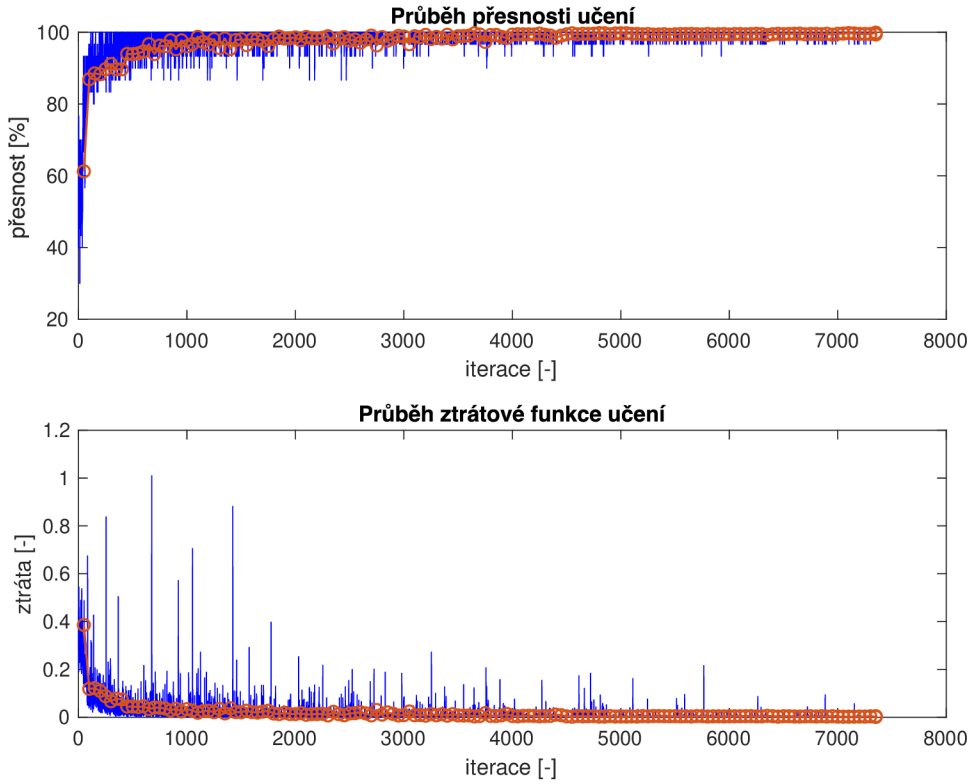
5.4 Další testování

V rámci navazující práce byl rozšířen použitý dataset o další nahrávky těžby motorové pily a zvuky pozadí, což v rámci testování přineslo výrazné zlepšení. Do části datasetu obsahující zvuky pozadí byly zařazeny i nahrávky zvuků, u kterých bylo během prvotního testování zjištěno, že způsobují falešné detekce klasifikačního programu.

Kromě architektury navržené v rámci semestrální práce byly do testování zařazeny další dvě řešení, jejichž průběh trénování je také představen v této části práce. Jedná se o konvoluční neuronovou síť trénovanou pomocí nahrávek o délce 1 s a kombinaci této sítě s rekurentní LSTM sítí.

5.4.1 Trénování rozšířené CNN3s

Po rozšíření datasetu na velikost čítající přes 12000 vzorků bylo nutné prodloužit dobu učení až na 30 epoch pro dosažení žádané hodnoty ztrátové funkce. Výsledkem trénování však byla síť s úspěšností klasifikace validační množiny převyšující 99 %. Průběh změn úspěšnosti učení a ztrátové funkce je vidět na obrázku 5.5. Stejně jako v případě trénování v rámci semestrální práce se opět v průběhu objevovaly skokové nárůsty ztrátové funkce, které měly za následek lokální propad úspěšnosti validace během učení. Nicméně s delší dobou učení se tento jen postupně vytrácel a na výsledek učení neměl vliv. I přes značný nárůst potřebných iterací zabíralo trénování sítě s použitím jednoho CPU pouze několik desítek minut. Při klasifikaci



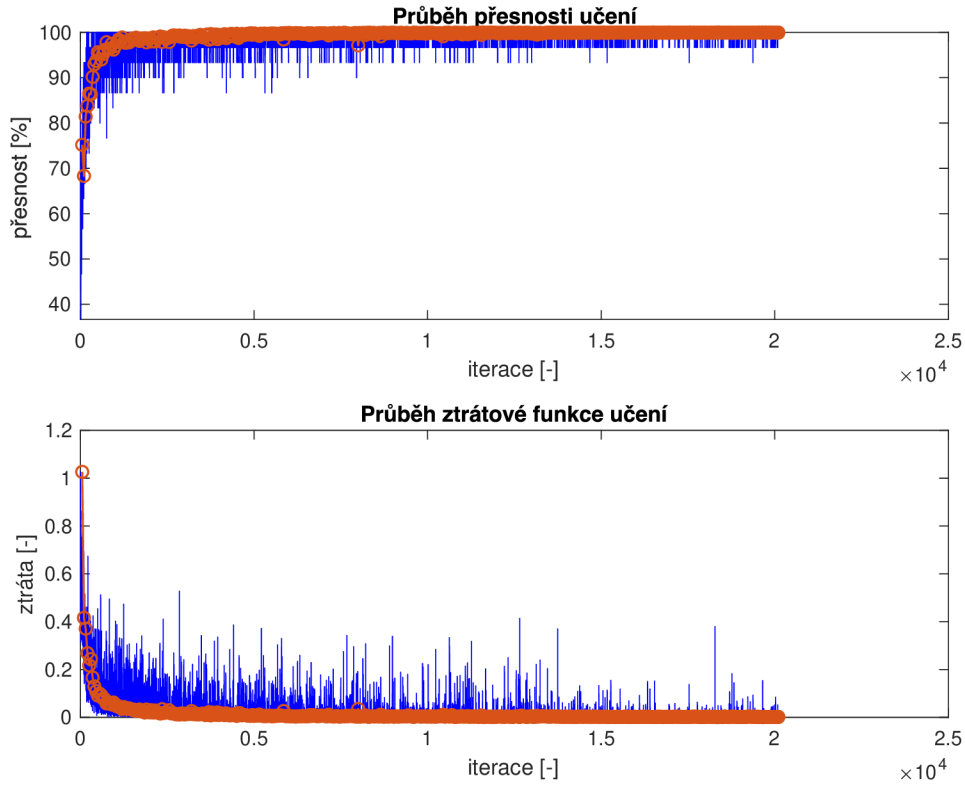
Obr. 5.5: Průběh učení rozšířené CNN3s.

testovací sady síť opět vykazovala velmi dobré výsledky, a to jak z hlediska úspěšnosti klasifikace, tak i její přesnosti, jak dokazují matice záměn a graf ROC 5.8. Přesnost klasifikace byla v tomto případě 99,87 % a spolehlivost 99 %. Z celé testovací sady, čítající okolo 2500 vzorků síť chybně klasifikovala pouze 3 z nich, a to jako falešně pozitivní.

5.4.2 Trénování CNN1s

Pro účely LSTM sítě byl vytvořen samostatný dataset nahrávek o délce 1 s a překryvu 50 %. Tímto datasetem byla natrénována navržená architektura CNN použitá pro klasifikaci 3 s vzorků. Tato síť nevykazovala v průběhu testování příliš dobré výsledky, nicméně byla pro srovnávací účely zařazena mezi testované sítě. Průběh jejího učení je na obrázku 5.6. Opět byla nastavena doba učení na 30 epoch. Jak je z grafů patrné, průběh učení nebyl zdaleka tak hladký jako v případě výše zmíněné sítě a ke konvergenci docházelo pomaleji. Z důvodu velkého nárůstu datasetu, takřka na pětinásobek, z důvodu kratších analyzovaných úseků, bylo učení více časově náročné. Stále však doba trénování nepřevyšovala jednu hodinu.

V případě klasifikace testovací množiny datasetu docházelo k vysoké úspěšnosti klasifikace – 99,82 % se spolehlivostí 100 %, nicméně s nižší rozlišitelností klasifikace, jak je vidět na rozhodovacích prazích křivky ROC na obrázku 5.9b. Klasifikační prahy, podílející se na rozhodování, byly na poměrně nízkých hodnotách oproti zbylým testovaným sítím.

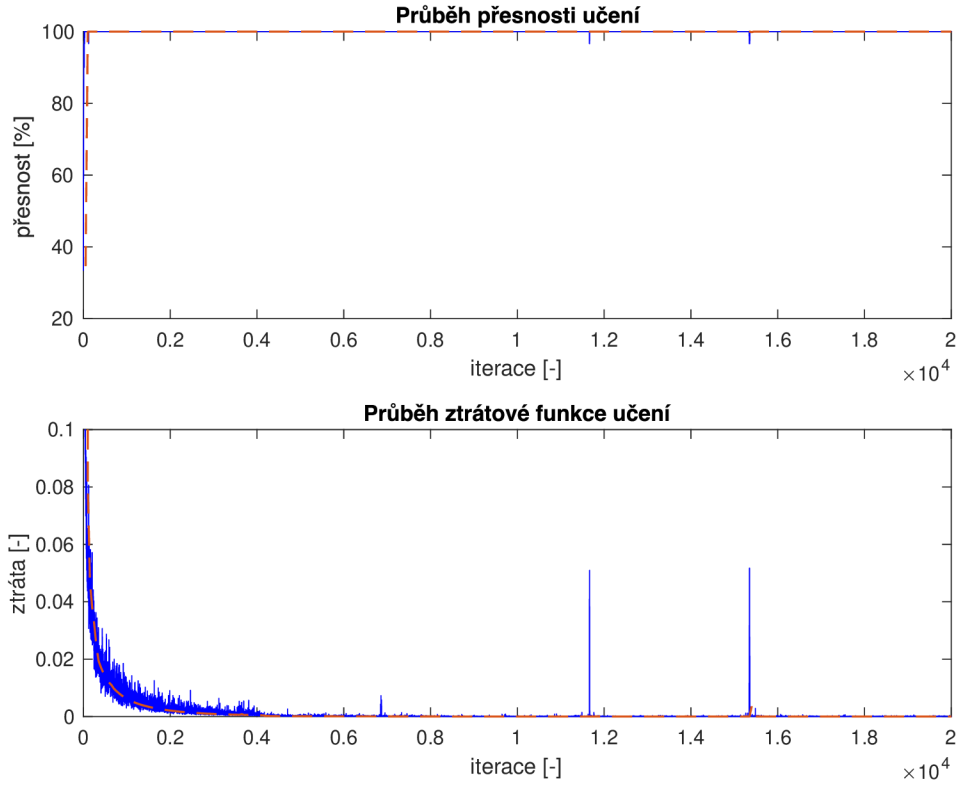


Obr. 5.6: Průběh učení CNN1s.

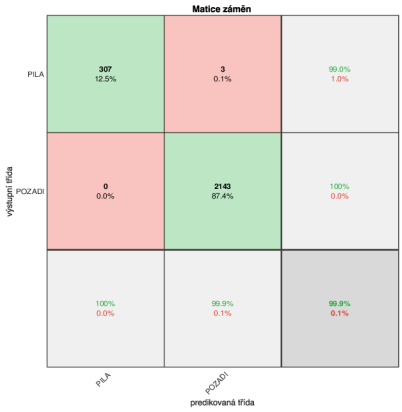
5.4.3 Trénování CNN1s + LSTM

Trénování navržené LSTM sítě bylo výpočetně mnohem méně náročné než v případě CNN. Byla testována různá nastavení parametrů sítě pro dosažení co nejlepších výsledků. V případě počtu skrytých buněk LSTM sítě se s vyšším počtem zvyšovala výpočetní náročnost bez znatelného zvýšení úspěšnosti klasifikace. Stejný efekt přineslo i zvýšení počtu LSTM vrstev. Jako nejúčinnější se tedy ukázalo použití jediné LSTM vrstvy s pouhými 10 buňkami a prodloužení doby učení až na 150 epoch. To v případě navržené rekurentní sítě probíhalo pouze v řádu jednotek minut. Nicméně při zohlednění doby nutné k předtrénování použité konvoluční sítě a následné transformaci dat pro účely LSTM se jedná rozhodně o časově nejvíce náročné řešení. Celý průběh učení a validace, spolu s průběhem ztrátové funkce je zobrazen grafem 5.7.

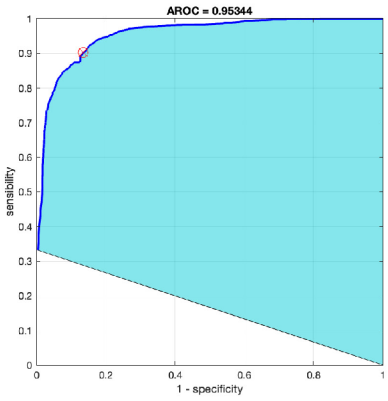
Při klasifikaci testovací sady datasetu, kterou tvořilo 20 % náhodně vybraných vzorků datasetu, byla dosažena 100% úspěšnost se 100% spolehlivostí klasifikace a dobrou rozlišitelností tříd, což je opět možné vidět na matici záměn 5.10a a průběhu křivky ROC 5.10b. Zde je vidět, že výsledky jsou srovnatelné s navrženou konvoluční sítí pro klasifikaci 3 s zvukových vzorků.



Obr. 5.7: Průběh učení CNN1s + LSTM.

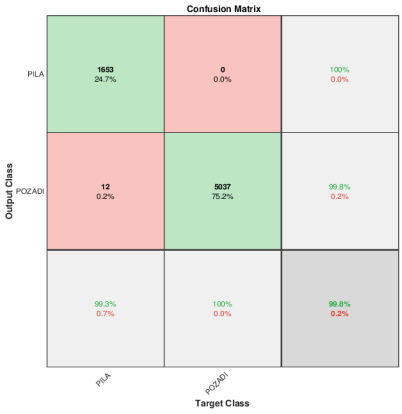


(a) Matice záměn.

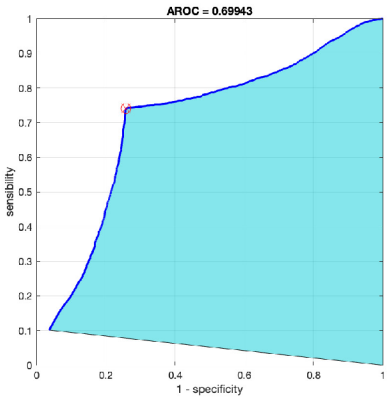


(b) Křivka ROC.

Obr. 5.8: Úspěšnost klasifikace testovací sady sítí CNN3s.

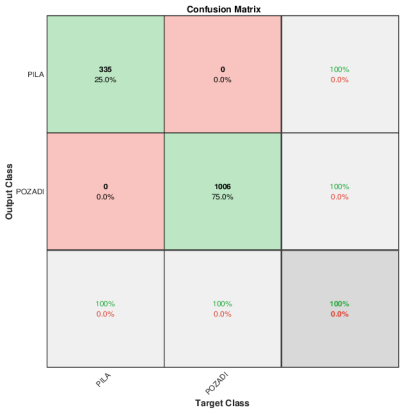


(a) Matice záměn.

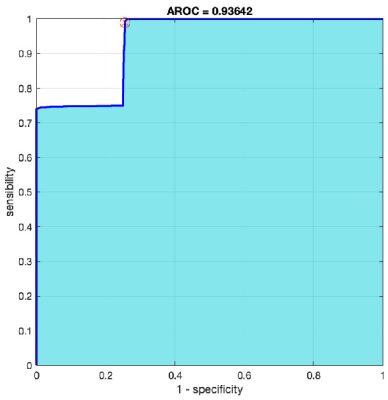


(b) Křivka ROC.

Obr. 5.9: Úspěšnost klasifikace testovací sady sítí CNN1s.



(a) Matice záměn.



(b) Křivka ROC.

Obr. 5.10: Úspěšnost klasifikace testovací sady sítí CNN1s + LSTM.

5.4.4 Testování a srovnání

V této části jsou porovnány výsledky klasifikace tří testovacích nahrávek pomocí detekčního programu, využívajícího představené sítě. Testované sítě jsou v této části značeny následovně:

- **CNN3s** – Konvoluční neuronová síť natrénovaná na 3 s úseky signálu,
- **CNN1s** – Konvoluční neuronová síť natrénovaná na 1 s úseky signálu,
- **LSTM** – Kombinace CNN1s a LSTM pro klasifikaci 3 s úseků signálu.

Detekované oblasti, určené programem, byly porovnány s reálným rozložením pozitivních a negativních oblastí. Kvalitativní parametry klasifikace pak byly vypočteny na základě správně a nesprávně určených vzorků nahrávky. Manuálně bylo stanoveno, jaká časová část vzorku odpovídá danému typu klasifikace (TP, TN, FP, FN). Z těchto hodnot byly následně vypočítány evaluační parametry, popsané v části 5.2.1.

Pro účely testování byly vytvořeny tři samostatné zvukové nahrávky, každá o délce 5 minut. Aby bylo dosaženo co nejefektivnějšího testování klasifikace z pohledu falešně pozitivních detekcí, byly do těchto nahrávek zasazeny i zvuky událostí, u kterých byla v průběhu testování sítí zaznamenána náchylnost na FP predikce. Jako pozadí testovaných nahrávek byly použity reálné nahrávky pořízené v lesním prostředí, obsahující pouze přirozené zvuky, případně úryvky konverzace. K tomuto pozadí byly poté přidány pořízené zvuky těžby motorové pily z různých vzdáleností a s různou hlasitostí. Záměrně byly vybrány nahrávky, které nebyly použity pro trénování sítí. Tyto nahrávky byly zasazeny do pozadí tak, aby částečně docházelo k jejich maskování a nižšímu odstupu signálu od hluku pozadí. Všechny tři testovací soubory obsahovaly několik úseků zvuků motorové pily o přibližné celkové délce 1 minuty pro každou z nahrávek.

Jak již bylo zmíněno, kromě schopnosti programu detekovat zvuk těžby dřeva, byla testována i jeho odolnost vůči falešně pozitivním detekcím. Z toho důvodu byly přidány nahrávky motokrosových motorek, projíždějících vozidel, silného deště a větru nebo hlasité lidské řeči.

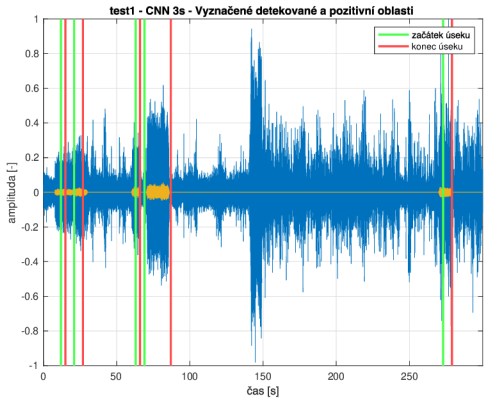
Každá z testovacích nahrávek byla zaměřena na určitý testovaný aspekt. První měla ověřit schopnost programu detekovat zvuk pily z větších vzdáleností v kombinaci s lidskou řečí v blízké vzdálenosti. Zároveň bylo testováno, zda zvuk těžkého stroje (v tomto případě traktoru) bude způsobovat falešné detekce programu.

Druhá nahrávka obsahovala opět zvuk těžby ve větší vzdálenosti, tentokrát byla však testována i schopnost zachytit krátké úseky těžby. Jako rušivý prvek byly použity nahrávky motokrosové motorky a projíždějících vozidel. U těchto zvuků testo-

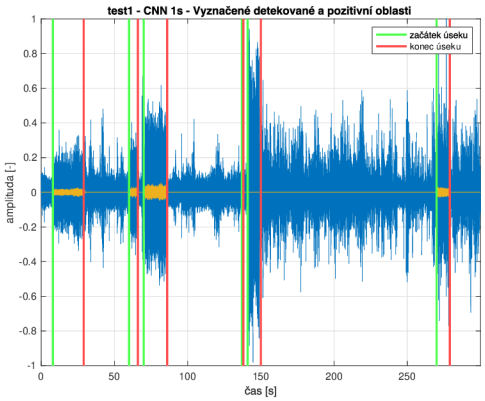
vána odolnost vůči hlasitým podnětům, ale i vzorkům s přidanou ambiencí.

Pro třetí soubor byla jako pozadí požita nahrávka poměrně hlučného prostředí, obsahujícího převážně zpěv ptáků. U této nahrávky bylo zkoumáno, zda je program schopen správně klasifikovat úryvky těžby s vysokým maskováním pozadí. Z tohoto důvodu byly také použity obtížně rozpoznatelné nahrávky těžby. Pro rušivé zvuky byly použity nahrávky projíždějících vozidel, motokrosu a silného deště s větrem. Všechny tyto nahrávky byly zakomponovány tak, aby působily co nejvíce přirozeně.

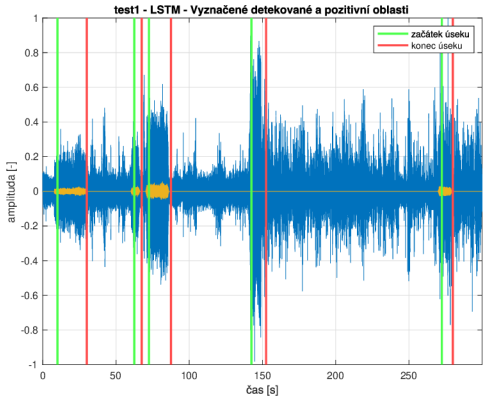
Na obrázcích 5.11 jsou porovnané výstupy detekčního programu jednotlivých sítí



(a) `test_track1.wav` – CNN3s.



(b) `test_track1.wav` – CNN1s.



(c) `test_track1.wav` – LSTM.

Obr. 5.11: Porovnání detekce navržených sítí pro soubor `test_track1.wav`.

pro první testovací nahrávku `test_track1.wav`. Zelené svislé čáry značí začátek detekované oblasti a červené její konec. Žlutě vyznačené oblasti nahrávky odpovídají přesným polohám zvuků těžby v nahrávce.

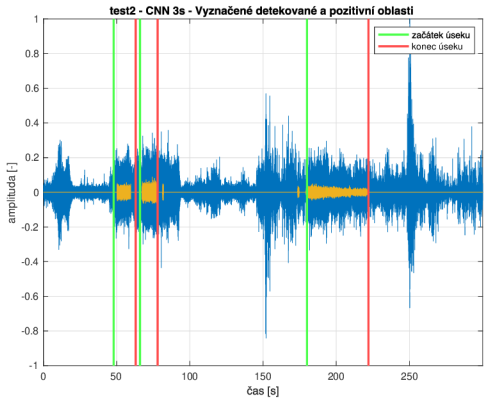
Jak je z grafů patrné, všechny tři testované sítě byly schopné zachytit všechny pozitivní úseky. V případě CNN3s nedocházelo k žádným falešně pozitivním detekcím, nicméně z důvodů většího klasifikačního okna se ne vždy síti podařilo přesně

zachytit začátek a konec události.

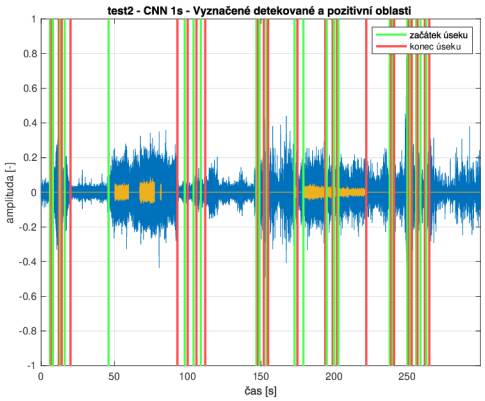
Pomocí sítě CNN1s se podařilo všechny požadované úseky zachytit téměř ze 100 %, a to především díky klasifikaci po 1 s úsecích signálu. Nicméně se zde objevovaly falešné detekce, způsobené zvukem traktoru, což je možné vidět v oblasti okolo 150. sekundy nahrávky zobrazené v grafu 5.11b.

Síť LSTM zachycovala začátky a konce pozitivních úseků o něco lépe než CNN3s, stejně jako v případě CNN1s však zvuk traktoru způsoboval falešné detekce.

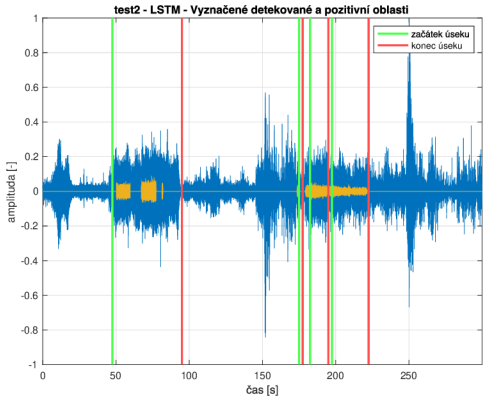
V případě druhé testovací nahrávky dosahovala nejlepších výsledků síť CNN3.



(a) `test_track2.wav` – CNN3s.



(b) `test_track2.wav` – CNN1s.



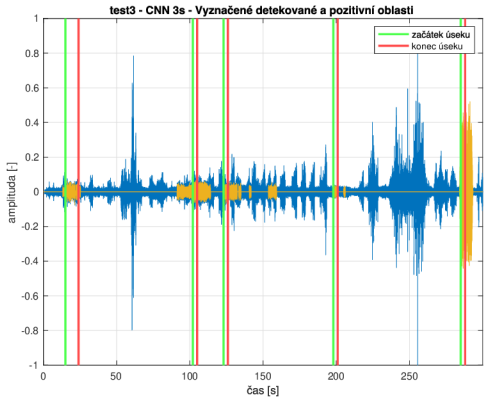
(c) `test_track2.wav` – LSTM.

Obr. 5.12: Porovnání detekce navržených sítí pro soubor `test_track2.wav`.

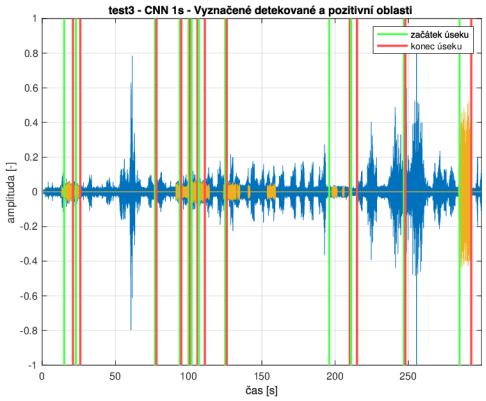
Opět vykazovala nejméně falešně pozitivních detekcí, nicméně krátké úseky těžby se jí nepodařilo detekovat, což lze pozorovat například v čase přibližně na 170. vteřině signálu v grafu 5.12a. Síť CNN1s stejně jako u předchozího testovacího vzorku detekovala takřka všechny zvuky motorové pily, opět však na úkor falešně pozitivních detekcí. Ty byly způsobeny zvukem motorky i projíždějícího auta, což je patrné v prvotní a závěrečné části signálu na obrázku 5.12b. Síť LSTM reagovala na falešné

podněty velmi dobře, stejně jako CNN3s. Problém jí však dělaly krátké pauzy mezi těžbou vyplněné například jiným zvukem motoru, což lze pozorovat na první detekované oblasti této sítě. Tento aspekt je možné přičíst faktu, že tato síť klasifikuje sekvence složené z pěti sekundových úseků s 50% překryvem, což odpovídá klasifikačnímu oknu o délce 3 s.

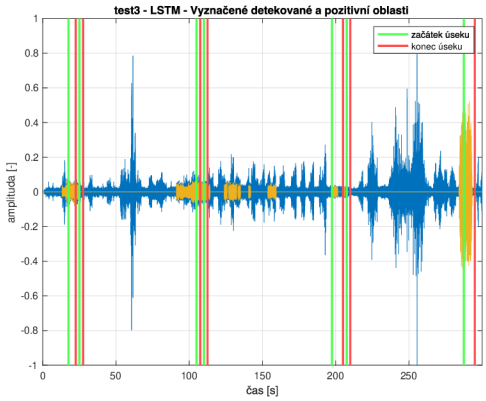
Třetí testovací nahrávka opět potvrdila úspěšnost sítě CNN3s. Tentokrát se však



(a) `test_track3.wav` – CNN3s.



(b) `test_track3.wav` – CNN1s.



(c) `test_track3.wav` – LSTM.

Obr. 5.13: Porovnání detekce navržených sítí pro soubor `test_track3.wav`.

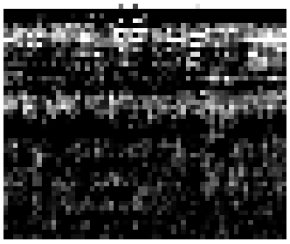
ukázalo, že vysoká úroveň maskování zvuků těžby je pro detekční program problematická na rozpoznání. Žádná z testovaných sítí nebyla schopna správně zařadit jako zvuk motorové pily oblast nahrávky v okolí 150. vteřiny. U ostatních vzorků však byly i přes hlasitý hluk pozadí víceméně úspěšné. Síť CNN3s opět nezaznamenala téměř žádné falešně pozitivní detekce a zachytila většinu vzorků těžby. Opět však nebyla schopna správně klasifikovat celou zvukovou událost, jak je možné pozorovat na grafu 5.13a.

U sítě CNN1s se objevilo malé množství falešně pozitivních detekcí způsobených

zvukem silného deště a projíždějícím vozidlem. Klasifikace delších úseků těžby byla pro tuto síť také problematická a správně určovala pouze krátké výseky v rozsáhlejších oblastech, což mělo za následek nárůst falešně negativních predikcí.

Výsledky dosažené pomocí sítě LSTM, zobrazené v grafu 5.13c, byly obdobné jako v případě sítě CNN3s. Opět se síti nepodařilo určit značně maskovanou část v půli nahrávky, ostatní vzorky těžby však určila správně. V případě třetí testovací nahrávky se u této sítě objevovalo pouze malé množství falešně pozitivních detekcí, které byly takřka výhradně způsobeny nesprávnou klasifikací vzorků bezprostředně za pozitivní oblastí.

Falešné detekce, které se během testování objevovaly, byly ve většině případů způsobeny jinými motorovými vozidly nebo v případě CNN1s krátkodobými rušivými prvky ve spektru. V případě této sítě by bylo možné zamezit těmto lokálním rušením například zavedením podmínky minimální délky klasifikovaného úseku, nutného k prohlášení oblasti za pozitivní. Nepřesnosti způsobené zvuky jiného charakteru se vyskytovaly z důvodu velmi podobného spektra se spektrem zvuku motorové pily. Podobnost příznakových map MFCC koeficientů pro zvuk pily, motorky a projíždějícího auta je znázorněna na obrázku 5.14.



(a) MFCC motorové pily.

(b) MFCC motokrosové motorky.

(c) MFCC projíždějícího auta.

Obr. 5.14: Porovnání MFCC příznaků.

Detailní srovnání výsledků klasifikace všech tří testovacích signálů je zobrazené tabulkou 5.2. Jako srovnávací parametry byly vybrány parametry *FPR*, *Sensitivity*, *Precision* a *Accuracy*.

Ukazatel *FPR* vypovídá o odolnosti sítě vůči falešným detekcím. Pro praktické

Tab. 5.2: Srovnání úspěšnosti klasifikace testovacích nahrávek.

test_track1	FPR	Sensitivity	Precision	Accuracy
CNN3s	**0,009**	0,629	**0,944**	0,921
CNN1s	0,041	**0,963**	0,839	**0,959**
LSTM	0,049	0,796	0,754	0,913
test_track2				
CNN3s	**0,030**	**0,954**	**0,899**	**0,966**
CNN1s	0,264	**0,954**	0,504	0,784
LSTM	0,126	0,692	0,625	0,832
test_track3				
CNN3s	**0,005**	0,306	**0,950**	**0,826**
CNN1s	0,08	**0,460**	0,644	0,809
LSTM	0,047	0,290	0,667	0,789

využití detekčního programu se jedná o jeden z nejdůležitějších parametrů. Aby reálný systém detekce nelegálního kácení byl efektivní, je žádoucí co nejnižší procento planých detekcí. V tomto ohledu se jako nejspolehlivější ukázala síť CNN3s, u které tato hodnota nepřekročila u žádné z testovaných nahrávek hodnotu 3 %. U zbylých dvou sítí byl ukazatel *FPR* ve všech případech několikanásobně vyšší.

Z pohledu ukazatele *Sensitivity*, který představuje schopnost sítě zachytit pravdivě pozitivní vzorek, nejlépe ze všech testovaných sítí vycházela síť CNN1s. U prvních dvou testovaných zvukových souborů byla tato síť schopná správně klasifikovat 96,3 % a 95,4 % pozitivních vzorků. V případě třetího souboru byla rozpoznávací schopnost všech sítí značně snížená vysokou úrovní hluku pozadí, což mělo za následek razantní propad hodnot *Sensitivity*. Nejvyšší hodnotu opět zaznamenala síť CNN1s, a to 46 %.

Parametr *Precision* ukazuje souhrnnou spolehlivost pravdivé klasifikace sítě, tedy jaká část vzorků označených jako pozitivní byla ve skutečnosti pozitivní. Vzhledem k podstatě řešeného problému se opět jedná o jeden z nejzásadnějších parametrů pro hodnocení kvality navržené sítě. Ve všech třech případech bylo nejlepších výsledků dosaženo pomocí sítě CNN3s, u níž hodnota ukazatele *Precision* dosahovala minimálně 90 %. Nejspolehlivější klasifikace bylo dosaženo u nahrávky `test_track3.wav`, a to 95 %, přičemž hodnoty zbylých dvou sítí odpovídaly přibližně 66 % správně určených pozitivních vzorků.

Posledním zkoumaným parametrem byla celková *Úspěšnost* sítě, tedy poměr všech správných detekcí vůči jejich celkovému počtu. V případě první nahrávky všechny zkoumané sítě vykazovaly vysoké hodnoty, převyšující 90 %. U nahrávky

druhé již síť CNN1s a LSTM nebyly tak úspěšné. Naopak síť CNN3s zaznamenala zcela nejvyšší hodnotu 96,6 %. U poslední testované nahrávky byla dle očekávání zaznamenána nižší úspěšnost klasifikace a výsledky všech sítí byly srovnatelné. Nejvyšší hodnotu však opět vykazovala síť CNN3s – 82,6 %.

Klasifikace pomocí sítě CNN3s byla nejméně výpočetně náročná a zpracování testovacích nahrávek o délce 5 minut trvalo na běžném CPU průměrně 1,8 s. V případě sítě CNN1s se doba klasifikace z důvodu třetinového klasifikačního okna prodloužila na 4 s. Nejvíce výpočetně náročná byla z hlediska doby potřebné ke klasifikaci nahrávky síť LSTM, u které bylo pro klasifikaci jednoho 3 s úseku vypočítáno pět MFCC příznakových map. Klasifikace testovacích nahrávek tedy této síti zabrala až 12 s.

Závěr

Tato diplomová práce se zabývala problémem klasifikace zvukových nahrávek za účelem detekce zvuku nelegální těžby dřeva. Pro řešení tohoto problému byla použita klasifikační metoda založená na využití konvolučních neuronových sítí a rekurentních neuronových sítí. Jako vstupní data neuronové sítě byly zvoleny koeficienty *MFFC*.

V první kapitole jsou popsány nejčastěji užívané grafické reprezentace zvukového signálu, používané pro účely číslicové analýzy. Jsou zde mimo jiné vysvětleny pojmy: *spektrogram*, *mel-spektrogram*, *kepstrum* a *Constant-Q*.

Druhá kapitola se zabývá oblastí Strojového učení. Nejprve je zde rozebrána základní teorie potřebná k pochopení samotné problematiky. Nejvíce prostoru je zde věnováno *neuronovým sítím* a především *konvolučním neuronovým sítím*, které jsou využity v praktické části práce. Jsou zde popsány jednotlivé použité vrstvy, jejich význam a funkce.

V následující třetí kapitole je rozebráno několik současných metod řešení problematiky zvukové klasifikace. Jsou zde představeny tři práce zaměřené právě na detekování zvuku těžby dřeva. U každé z nich jsou detailně popsány použité postupy a dosažené výsledky. V následující části jsou zmíněny další dva články, jejichž námět a použitá řešení jsou také blízká této práci.

Ve čtvrté kapitole jsou představeny návrhy neuronových sítí pro řešení v prostředí Matlab. Je zde popsána tvorba použitého datasetu a blíže popsány navržené architektury sítí. Závěr kapitoly je věnován popisu funkce testovacího detekčního programu.

Poslední kapitola obsahuje zhodnocení dosažených výsledků a srovnání testovaných sítí. První část této kapitoly je věnována diskuzi výsledků klasifikačního programu pro různá zvolená nastavení v rámci prvotního testování. Jsou zde srovnány jednotlivé průběhy učení, výsledné *ROC* křivky a matice záměn a uvedeny ukázky výsledků detekčního programu pro různé testovací nahrávky.

V druhé části jsou zobrazeny průběhy učení sítí představených v rámci navazující práce po rozšíření použitého datasetu. Závěr práce obsahuje srovnání výsledků představených sítí při klasifikaci testovacích nahrávek. Jednotlivá řešení jsou zde porovnána pomocí grafického výstupu detekčního programu a pomocí objektivních evaluačních parametrů.

V rámci diplomové práce byla ověřena schopnost konvolučních neuronových sítí úspěšně klasifikovat zvukové nahrávky za účelem detekce zvuku motorové pily. Nejlepších výsledků bylo dosaženo pomocí sítě CNN3s, u které úspěšnost klasifikace dosahovala až 96 % při klasifikaci testovacích nahrávek. Zároveň tato síť prokázala nejvyšší odolnost vůči falešně pozitivním detekcím, jejichž minimalizace byla jedním

z cílů této práce.

Klasifikace sítí CNN3s se také ukázala jako nejméně výpočetně náročná, což v reálném použití detekčního programu hraje také velkou roli. U všech sítí se však ukázala doba potřebná ke klasifikaci nahrávky jako dostatečně nízká pro případnou real-time aplikaci programu.

Výsledky dosažené představeným řešením se dají považovat za uspokojivé, nicméně testování ukázalo, že vysoká hlučnost okolí v kombinaci s velkou vzdáleností přicházejícího zvuku motorové pily způsobuje problémy se správnou klasifikací.

Klasifikační úspěšnost programu by bylo možné do budoucna vylepšit například využitím různých kombinací použitých příznaků či kombinací různých délek klasifikačních oken. Reálná aplikace programu v detekčním systému umístěném v lese by byla možná ještě obohatit o možnost určení směru příchozího zvuku s využitím dvou přijímačů na základě rozdílné fáze příchozího signálu.

Literatura

[1] SMÉKAL, Z. *Zpracování řeči.* Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací, 2013. ISBN 978-80-214-4896-4.

[2] SEMELA, René. *Automatické tagování hudebních děl pomocí metod strojového učení.* Brno, 2020. Diplomová práce. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací. Vedoucí práce Tomáš Kiska.

[3] SLÍVOVÁ, Martina. *Frekvenčně-časová analýza zvukových signálů.* Ostrava, 2017. Bakalářská práce. Technická univerzita Ostrava Fakulta elektrotechniky a informatiky. Katedra telekomunikační techniky. Vedoucí práce Ing. Jan Skapa Ph.D.

[4] GRAZIOSI, D.B., C.N. DOS SANTOS, S.L. NETTO a L.W.P. BISCAINHO. *A constant-Q spectral transformation with improved frequency response.* In: 2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512) [online]. IEEE, 2004, V-544-V-547 [cit. 2020-12-06]. ISBN 0-7803-8251-X. Dostupné z: doi:10.1109/ISCAS.2004.1329710

[5] PAVOL, Harár. *Audio classification with deep learning on limited data sets.* Brno, 2019. Doctoral thesis. Brno University of Technology, Faculty of Electrical En- gineering and Communication, Department of Telecommunications. Vedoucí práce Ing. Jiří Mekyska Ph.D.

[6] MITCHELL, Tom M. *Machine learning.* Maidenhead, U.K.: McGraw-Hill, 1997, 432 s. ISBN 0070428077.

[7] MCCULLOCH, Warren S. a Walter PITTS. textitA logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics [online]. 1943, 5(4), 115-133 [cit. 2020-12-06]. ISSN 0007-4985. Dostupné z: doi:10.1007/BF02478259

[8] ROSENBLATT, F. textitThe perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review [online]. 1958, 65(6), 386-408 [cit. 2020-12-06]. ISSN 1939-1471. Dostupné z: doi:10.1037/h0042519

[9] RINA, Dechter. *Learning while searching in constraint-satisfaction problems* [online]. In: . University of California: Computer Science Department, Cognitive Systems Laboratory, 1986/8/11, s. 178-185 [cit. 2020-12-06]. Dostupné z: https://www.aaai.org/Papers/AAAI/1986/AAAI86-029.pdf

[10] PETR, Hanzlík. *Metody umělé inteligence v rozpoznávání rostlin*. Praha, 2018. Dizertační práce. Česká zemědělská univerzita v Praze. Provozně ekonomická fakulta. Katedra informačního inženýrství. Vedoucí práce Ing. Arnošt Veselý, CSc.

[11] IOFFE, Sergey a Christian SZEGEDY. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. Dostupné z: https://arxiv.org/pdf/1502.03167v3.pdf

[12] HO, Yaoshiang a Samuel WOOKEY. *The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling*. IEEE Access [online]. 2020, 8, 4806-4813 [cit. 2020-12-06]. ISSN 2169-3536. Dostupné z: doi:10.1109/ACCESS.2019.2962617

[13] KALHARA, P. G., V.D. JAYASINGHEARACHCHD, A. H. A. T. DIAS, V. C. RATNAYAKE, C. JAYAWARDENA a N. KURUWITAARACHCHI. textitTreeSpirit: Illegal logging detection and alerting system using audio identification over an IoT network. In: 2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA) [online]. IEEE, 2017, 2017, s. 1-7 [cit. 2020-12-06]. ISBN 978-1-5386-4602-1. Dostupné z: doi:10.1109/SKIMA.2017.8294127

[14] GU, Lin, Tarek ABDELZAHER, Bruce H. KROGH, et al. Lightweight detection and classification for wireless sensor networks in realistic environments. In: *Proceedings of the 3rd international conference on Embedded networked sensor systems - SenSys '05* [online]. New York, New York, USA: ACM Press, 2005, 2005, s. 205- [cit. 2020-12-06]. ISBN 159593054X. Dostupné z: doi:10.1145/1098918.1098941

[15] COLONNA, Juan Gabriel, Bernardo GATTO, Eulanda MIRANDA DOS SANTOS a Eduardo Freire NAKAMURA. A Framework for Chainsaw Detection Using One-Class Kernel and Wireless Acoustic Sensor Networks into the Amazon Rainforest. In: *2016 17th IEEE International Conference on Mobile Data Management (MDM)* [online]. IEEE, 2016, 2016, s. 34-36 [cit. 2020-12-06]. ISBN 978-1-5090-0883-4. Dostupné z: doi:10.1109/MDM.2016.86

[16] BAJZIK, Jakub, Jiri PRINOSIL a Dusan KONIAR. Gunshot Detection Using Convolutional Neural Networks. In: *2020 24th International Conference Electronics* [online]. IEEE, 2020, 2020, s. 1-5 [cit. 2020-12-06]. ISBN 978-1-7281-5868-6. Dostupné z: doi:10.1109/IEEECONF49502.2020.9141621

[17] LAMPA, Ondřej. *Paralelní trénování hlubokých neuronových sítí*. Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Hradiš Michal.

[18] LIM, Minkyu, Donghyun LEE, Kwang-Ho KIM, Ji-Hwan KIM. Audio Event Classification Using Deep Neural Networks. *Phonetics and Speech Sciences* [online]. 2015, 7(4), 27-33 [cit. 2020-12-06]. ISSN 2005-8063. Dostupné z: doi:10.13064/KSSS.2015.7.4.027

[19] MYŠKA, Vojtěch. *Rekurentní neuronové sítě pro klasifikaci textů* [online]. Brno, 2018 [cit. 2021-5-21]. Dostupné z: http://hdl.handle.net/11012/80785. Diplomová práce. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací. Vedoucí práce Lukáš Povoda.

[20] ZHAO, Jianfeng, Xia MAO a Lijiang CHEN. Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomedical Signal Processing and Control* [online]. 2019, 47, 312-323 [cit. 2021-5-21]. ISSN 17468094. Dostupné z: doi:10.1016/j.bspc.2018.08.035

[21] Wen Zhu , Nancy Zeng , Ning Wang. Sensitivity, Specificity, Accuracy, Associated Confidence Interval and ROC Analysis with Practical SAS Implementations. *NESUG2010 - Health Care and Life Sciences*. Dostupné z: https://www.lexjansen.com/nesug/nesug10/hl/hl07.pdf

Seznam symbolů, veličin a zkratek

2D	Two-Dimensional – Dvourozměrný
ACC	Accuracy – Přesnost
ADT	Alternating Decision Tree
ANN	Artificial Neural Networks – Umělé neuronové sítě
AUC	Area Under the Curve – Plocha pod křivkou
CART	Classification and Regression Tree
CNN	Convolutional Neural Networks – Konvoluční neuronové sítě
CQT	Constant-Q Transform – Constant-Q transformace
CPU	Central Processing Unit – Centrální procesorová jednotka
DAW	Digital Audio Workstation
DCT	Discrete Cosine Transform – Diskrétní kosinová transformace
DFT	Discrete Fourier Transform – Diskrétní Fourierova transformace
DNN	Deep Neural Networks – Hluboké neuronové sítě
EER	Equal Error Rate
FFT	Fast Fourier Transform – Rychlá Fourierova transformace
FN	False Negative – Falešné negativní
FP	False Positive – Falešné pozitivní
FPR	False Positive Rate – Míra falešné pozitivity
GMM	Gaussian Mixture Models – Gaussovské smíšené modely
IDCT	Inverse Discrete Cosine Transform – Zpětná diskrétní kosinová transformace
IOT	The Internet Of Things
LAD	LogitBoost Alternating Decision Tree
LSTM	Long Short-Term Memory

MFCC Mel-frequency Cepstral Coefficients – Mel-frekvenční kepstrální koeficienty

MLP Multi-layred Perceptron – Vícevrstvá perceptronová síť

NN Neural Networks – Neuronové sítě

RBF Radial Basis Function – Radiálně bazická funkce

ReLU Rectified Linear Unit

RNN Recurrent Neural Network – Rekurentní neuronová síť

ROC Receiver Operating Characteristic – Operační charakteristika přijímače

STFT Short-time Fourier Transform – Krátkodobá Fourierova transformace

SVM Support Vector Machines – Metoda podpůrných vektorů

TN True Negative – Pravdivě negativní

TP True Positive – Pravdivě pozitivní

TPR True Positive Rate – Míra pravdivé pozitivity

A Dokumentace přiloženého detekčního programu

Tato část slouží jako uživatelský návod pro použití přiloženého detekčního programu na rozpoznávání zvuku těžby motorové pily ze zvukové nahrávky. Funkčnost programu je garantována na systému *Mac OS Catalina 64-bit* a *Matlab R2019b*. Pro správné fungování programu je třeba mít nainstalované základní toolboxy, spolu se *Signal Processing Toolbox* a *Deep Learning Toolbox*.

Program je spouštěn pomocí funkce `detection.m` a je možné jej spustit přímo z příkazového okna Matlabu. Při volání funkce bez vstupních a výstupních parametrů je uživatel vyzván k načtení zvukového souboru ve formátu `*.wav`. Výstupem funkce poté bude vykreslený graf s vyznačenými detekovanými oblastmi a časové informace o detekovaných oblastech vypsané do konzoly. V případě, že program žádný zvuk pily nezachytí, vypíše pouze hlášku.

Detekční program obsahuje kromě vstupního zvukového souboru druhou volitelnou vstupní proměnnou, která je při jejím nezadání vyplněna automaticky. Touto proměnnou je parametr `net_type`, kterým je možné měnit neuronovou síť, která se podílí na klasifikaci vzorků. Na výběr jsou tři možnosti – `CNN1s`, `CNN3s` a `LSTM`. Standardně je při vynechání této proměnné nastavena síť CNN3s. Detekční program poté nahraje jednu z uložených natrénovaných sítí, a ta je využita při klasifikaci. Tyto sítě jsou přiloženy v adresáři programu s příponou `*.mat`

Volitelné výstupní proměnné funkce jsou pole buněk `PILA` a informace o vzorkovacím kmitočtu `fs`. Do proměnné PILA jsou postupně jsou za sebe ukládány detekované oblasti, které je po zadání výstupní proměnné možné zpětně přehrát, například ve formátu `sound(PILA{1},fs)`.

Pro fungování programu jsou v adresáři také nezbytné funkce `mfcc.m`, `melbank.m`, `mel2Hz.m` a `segmentace.m`, které slouží jako podpůrné funkce pro výpočet MFCC ze vstupních dat. Pro potřeby predikce sítí CNN1s a CNN3s je také přiložen soubor `weightedClassificationLayer.m` obsahující váženou klasifikační vrstvu neuronové sítě.

Veškeré nahrávky použité pro tvorbu datasetu a samotné datasety vytvořené pro učení představených sítí, spolu s testovacími soubory, jsou k dispozici ve veřejně přístupné složce na *GoogleDrive*, dostupné z URL: `https://1url.cz/cKwJ6`, kde je ve formě samostatného dokumentu vložena i tato příloha.

B Obsah přiloženého CD

Přiložené CD obsahuje zdrojové soubory detekčního programu potřebné k jeho správné funkčnosti. Jsou zde také přiloženy skripty použité pro trénování testovaných neuronových sítí. V adresáři detekčního programu jsou také uloženy natrénované neuronové sítě, testované v této práci. Funkčnost všech skriptů je zaručena při použití *Matlab 2019b*.

Podrobná dokumentace k funkčnosti detekčního programu je uvedena v první části přílohy A. Obsah přiloženého CD je také přístupný v online verzi pod odkazem uvedeným v první části přílohy.

```
/ ..... kořenový adresář přiloženého CD
├── Detection_program ..... adresář souborů detekčního programu
│   ├── detection.m ..... skript detekčního programu
│   ├── mel2Hz.m ..... funkce přepočtu Mel škály na Hz
│   ├── melbank.m ..... funkce generující banku MEL filtrů
│   ├── segmentace.m ..... funkce segmentace signálu pro výpočet MFCC
│   ├── weightedClassificationLayer.m ..... funkce vážené klasifikační vrstvy CNN
│   └── manual.pdf ..... manuál k detekčnímu programu
├── CNN_train.m ..... skript pro trénování sítě CNN
├── LSTM_train.m ..... skript pro trénování sítě LSTM
├── plotting.m ..... funkce pro zobrazení výsledků trénování a klasifikace
├── 1s_cnn.m ..... natrénovaná síť CNN1s
├── 3s_cnn.m ..... natrénovaná síť CNN3s
└── 3s_lstm.m ..... natrénovaná síť LSTM
```