Západočeská univerzita v Plzni
Fakulta aplikovaných věd

BAKALÁŘSKÁ PRÁCE

Detekce akustických událostí

Vedoucí práce: Ing. Luboš Šmídl, Ph.D.
Katedra kybernetiky

Plzeň, 2023

Daniel Tauš

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Daniel TAUŠ**
Osobní číslo:	**A20B0391P**
Studijní program:	**B0714A150005 Kybernetika a řídicí technika**
Specializace:	**Umělá inteligence a automatizace**
Téma práce:	**Detekce akustických událostí**
Zadávající katedra:	**Katedra kybernetiky**

Zásady pro vypracování

i) Nastudujte problematiku automatického rozpoznávání řeči, zaměřte se na metody detekce klíčových slov a akustických událostí.
ii) Připravte vhodná trénovací/testovací data, využijte otevřených databází.
iii) Navrhněte a natrénujte model pro detekci akustických událostí pro offline a online detekci. Uvažujte možnost nasazení na zařízení s malým výpočetním výkonem.
iv) Modely otestujte a vyhodnoťte.

Rozsah bakalářské práce: **30-40 stránek A4**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

Dodá vedoucí práce

Vedoucí bakalářské práce: **Ing. Luboš Šmídl, Ph.D.**
Katedra kybernetiky

Datum zadání bakalářské práce: **17. října 2022**
Termín odevzdání bakalářské práce: **22. května 2023**

Západočeská univerzita v Plzni
Fakulta aplikovaných věd

Doc. Ing. Miloš Železný, Ph.D.
děkan

Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

V Plzni dne 17. října 2022

Prohlášení

Prohlašuji, že jsem bakalářskou práci na téma:

Detekce akustických událostí

vypracoval samostatně pod odborným dohledem vedoucího bakalářské práce za použití pramenů uvedených v přiložené bibliografii.

V Plzni den 22. května 2023 .....

Daniel Tauš

Poděkování

Tímto bych rád poděkoval Ing. Luboši Šmídlovi Ph.D. za vedení této bakalářské práce, velmi cenné a důležité rady a hlavně za ochotu a trpělivost, která mi byla po celou dobu věnována.

Abstrakt

Tato práce se zabývá neuronovými sítěmi pro detekci a klasifikaci audio signálu. V první kapitole se seznámíme se strojovým učením, druhy neuronových sítí, jak zpracováváme signál a jak se výsledky vyhodnocují. Druhá kapitola obsahuje experimenty na ozkoušeném datasetu Speech Commands Dataset a experimenty s tvorbou dat pro neuronové sítě. Třetí kapitola se věnuje detekci akustických událostí na vlastním datasetu, na kterém se testují již vyzkoušené postupy z druhé kapitoly.

Klíčová slova

Detekce akustické události, neuronové sítě, hlasové povely, vnitřní stav neuronové sítě

Abstract

This thesis is focused on neural networks for audio signal detection and classification. In the first chapter, we will learn about machine learning, types of neural networks, how we process the signal and how the results are evaluated. The second chapter contains experiments on the tried-and-tested Speech Commands Dataset and experiments with data generation for neural networks. The third chapter is devoted to the detection of acoustic events on its own dataset, on which already tested procedures from the second chapter are tested.

Key Words

Acoustic event detection, neural networks, voice commands, internal state of neural network

Obsah

1 Úvod

Akustické události jsou velmi rozsáhlým pojmem, který teoreticky může zahrnovat cokoliv spojeného se zvukem. Nicméně některé z těchto událostí jsou důležitější než jiné a jejich zachycení je žádoucí. Tyto události mohou zahrnovat například střelbu, křik, dopravní nehody a mnoho dalších. Firma JALUD Embedded s.r.o. již nabízí zařízení, která tuto detekci vykonávají v ulicích Plzně z hlediska bezpečnosti. Tyto zařízení mohou být využívány i pro jiné účely, jako například při kontrole výroby ve firmách, a nahradit tak tradiční kamery. Proto se pokusíme tento problém vyřešit na malém datasetu s akustickými událostmi s různými akustickými událostmi.

Jiná, ale zároveň velmi podobná oblast problémů je rozpoznávání klíčových slov, která slouží například k aktivaci chytrých asistentů na mobilních telefonech, ale může se vztahovat i na obecné hlasové povely pro ovládání chytrých zařízení. Řešení je pro tyto úlohy na stejném principu, tedy rozpoznat požadovanou akustickou událost, a proto se budeme zabývat i obsáhlým datasetem, který místo různých akustických událostí obsahuje pouze mluvené slova.

Výše zmíněné cíle se pokusíme splnit za využití metody strojového učení, konkrétně neuronových sítí, a jejich použití pro klasifikaci a detekci akustických událostí. Cílem této práce je vyzkoušet různé metody a přístupy k neuronovým sítím na dvou různých datasetech a zjistit, jaké možnosti pro další vývoj existují.

2 Strojové učení pro zpracování signálu

Strojové učení je metodou umělé inteligence. Jde o algoritmy se schopností učení a zdokonalení na základě předložených dat. Tyto algoritmy tvoří matematický model na základě databáze vstupních dat, díky kterému jsou schopné poskytnout aproximaci možného řešení zadaného úkolu [12]. Jejich využití je v mnoha odvětvích a stále roste. Existují jejich různé rozdělení podle učení nebo účelu. Z hlediska účelu jsou tři základní druhy a to klasifikace - rozdělení dat do tříd, regrese - na základě vstupních dat se odhaduje numerická hodnota jako číslo nebo přímka a shlukování - slučování objektů do skupin podle společných příznaků.

Druhý způsob dělení je na učení s učitelem, kdy ke vstupním datům předložíme i požadované výstupy a bez učitele, kdy se algoritmus snaží sám naleznout určité společné příznaky a rozdělit data bez možnosti kontroly správnosti jejich dělení. Metod strojového učení existuje více, například rozhodovací stromy, genetické algoritmy nebo statistické metody. V následující části je popsaná metoda neuronových sítí, která je pro tuto práci klíčová.

2.1 Neuronová síť

Základní stavební kámen neuronové sítě je neuron, který je inspirován reálnými neurony v mozku. Jako první se s tímto oborem setkal neurofyziolog W. Culloch a matematik W. Pitt v roce 1943, kdy při snaze o popsaní nervové aktivity sestrojili první, ačkoliv velmi jednoduchou, neuronovou síť, kde reprezentací neuronů byla lineární funkce s více vstupy a jedním, binárním výstupem. Tato síť však sloužila pro vysvětlení jejich představy fungování nervové soustavy a první síť, která byla použita pro strojové učení, je Perceptron. Tato síť byla modelována F. Rosenblattem v roce 1957 a je složena z jednoho neuronu [2]. Díky jednoduchosti je možné ji využít pouze na data, která jsou lineárně rozdělitelná, tedy například úloha rozdělení dat do dvou skupin přímkou.

V těchto letech nebylo možné nějaké složitější neuronové sítě implementovat z hlediska hardwaru, na kterém jsou velmi závislé. Proto se také k většímu rozmachu dostalo až okolo roku 1980 a v roce 1986 profesorka Rina Dechter zavedla pojem hluboké učení, který referuje na skryté vrstvy neuronových sítí [3].

Matematický model neuronu je ukázán na rovnici 1,

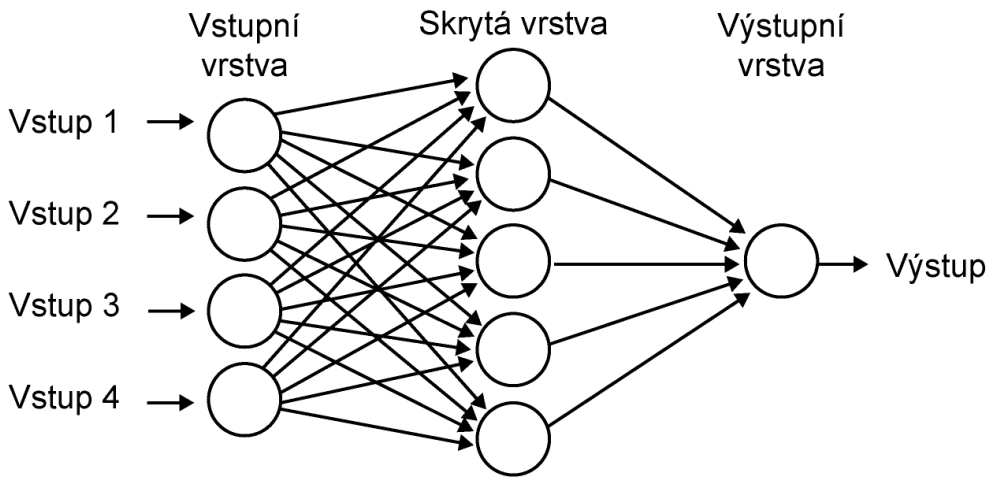
$$y = f(\sum_{i=1}^{n} w_i \times \ x_i - h), \tag{1}$$

kde f značí aktivační funkci, w_i značí jednotlivé váhy neuronu a x_i jsou vstupy. Nejprve se tedy sečtou všechny prvky vstupního vektoru vynásobené jejich váhami, od tohoto výsledku se poté odečte práh h a až tento výsledek je vstupem do aktivační funkce. Každý neuron není stejný a liší se samozřejmě jak nastavením vah, tak aktivační funkcí.

Nastavení vah probíhá při tréninku podle předložených dat a většinou není nutné do něj zasahovat. Aktivační funkce se však určuje předem a značí způsob zpracování signálů, které jsou do neuronu přivedeny. Mezi hojně používané funkce patří sigmoid a ReLu [18]. Základní princip ReLU spočívá v rozhodování, kde pokud je vstup do funkce záporný, vrátí nulu, jinak vrací hodnotu vstupu. Sigmoidová funkce a její výstup je počítán podle rovnice 2.

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

Obecně síť není tvořena pouze jedním neuronem, ale neurony jsou uspořádány do vrstev. Můžeme mít například 5 vrstev, kde v každé bude 5 neuronů. Vstup je poté zpracován od první vrstvy, která se nazývá vstupní, postupně až k poslední, výstupní vrstvě. Vrstvy mezi těmito dvěma se nazývají skryté. Toto schéma je znázorněno na obrázku 1.



Obrázek 1: Schéma vrstev neuronové sítě, převzato z [13]

2.2 Trénování neuronových sítí

V této práci je využito metody trénování s učitelem. To znamená, že síti je předkládaná k datům i informace o požadovaném výstupu. Takové trénování poté probíhá opakovanou úpravou vah.

V první řadě je nutné definovat, jak často se váhy mají upravovat a proto je zavedené značení batch size a pojem batch. Batch je označení pro několik dat, které jsou síti předloženy a poté se upraví váhy pomocí trénovacího algoritmu. Data jsou tedy rozdělena při trénování na batche, všechny mají stejnou velikost a tou je batch size. Tato velikost není omezená a její velikost může ovlivnit výsledky a také rychlost trénování, proto by to nemělo být jen náhodné číslo.

Data rozdělená na batche se předloží síti a proběhne úprava vah. Pokud jsou síti již předložena všechna data, je hotova jedna epocha trénování. Počet epoch je tedy dalším parametrem při trénování sítě a při vysokém počtu může být jednou z příčin přetrénování. Mezi pracemi se tento počet liší a velmi záleží na konkrétním modelu i datech.

Samotné upravování vah probíhá pomocí ztrátové funkce, kde se váhy upravují pomocí chyby, která je zpětně šířena od výstupu metodou backpropagation. Snaha je zde snížit ztrátovou funkci a proto úprava probíhá pomocí gradientu, kde se váhy a prahy mění v jeho záporném směru. Tuto část dělá optimalizátor. Výsledky a rychlost tréninku také velmi závisí na jeho výběru. Nejpoužívanější optimalizátory jsou Adam [19] a jeho varianty, Stochastic Gradient Descent neboli SGD a Root Mean Squared Propagation neboli RMSprop.

2.3 Vrstvy neuronových sítí

Metody pro rozpoznávání, ať už pro klasifikaci nebo detekci, se velmi liší v použitých modelech. Z hlediska hlubokých neuronových sítí jsou zde tři hlavní přístupy a mnohem více vrstev, které je možné použít. Tato práce se zabývá pouze sekvenčními modely, tedy modely kde je jedna vrstva spojena s maximálně dvěma okolními vrstvami.

2.3.1 Vstupní vrstva

Vrstva, která se nachází v každém modelu, je vrstva vstupní. Tato vrstva může sloužit k případnému předzpracování a augmentaci dat, není to však podmínkou. Při vytváření je zde velmi důležité upřesnit tvar vstupu, tedy například u obrázku upřesnit výšku, šířku a hloubku. Některé modely vyžadují také znát batch size.

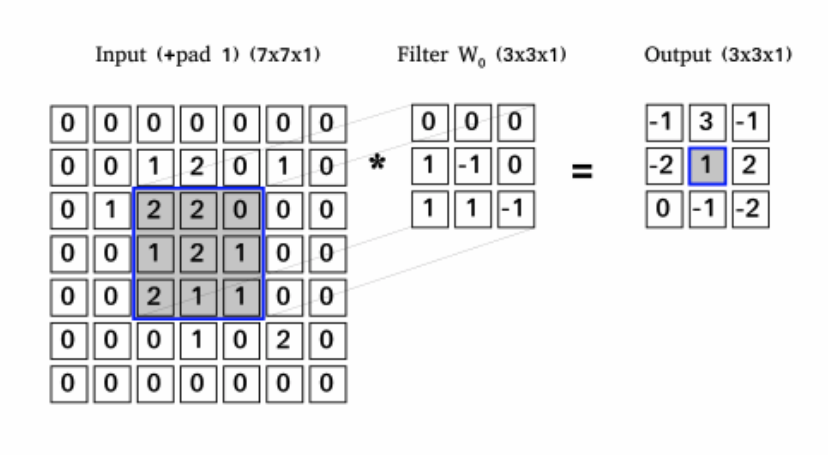
2.3.2 Plně propojené vrstvy

Nejjednodušší přístup spočívá v propojení všech výstupů jedné vrstvy se všemi vstupy následující vrstvy. Model může být upravován počtem vrstev a neuronů. Jejich porovnání lze najít v [8].

2.3.3 Konvoluční vrstva

První specifická vrstva je vrstva konvoluční. Tato vrstva je využívána hlavně pro obrazová data, které jsou ve 3D tvaru, tedy výška, šířka a hloubka. Pomocí konvolučních filtrů, tedy matic se stejnými nebo menšími dimenzemi je počítána konvoluce vstupních dat se všemi filtry. Výpočet konvoluce pro jeden filtr je ukázán na obrázku 2.

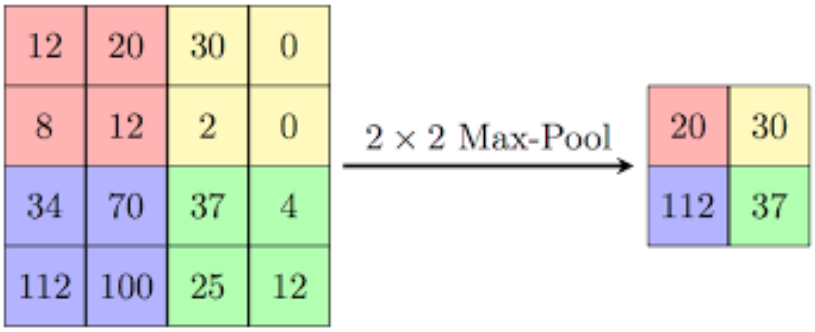
Kvůli započítání krajních čísel se matice může doplnit nulami po okrajích. Takový proces se nazývá zero padding.



Obrázek 2: Operace konvoluce se zero paddingem. Převzato z [15].

2.3.4 Sdružovací (Pooling) vrstva

Dalším krokem po konvoluční vrstvě bývá v modelu běžně vrstva sdružovací, a to hlavně z důvodu snížení počtu výstupních dat. Princip fungování je rozdělení vstupu na segmenty, ze kterých se poté vybere například maximum jako na obrázku 3 nebo střední hodnota.



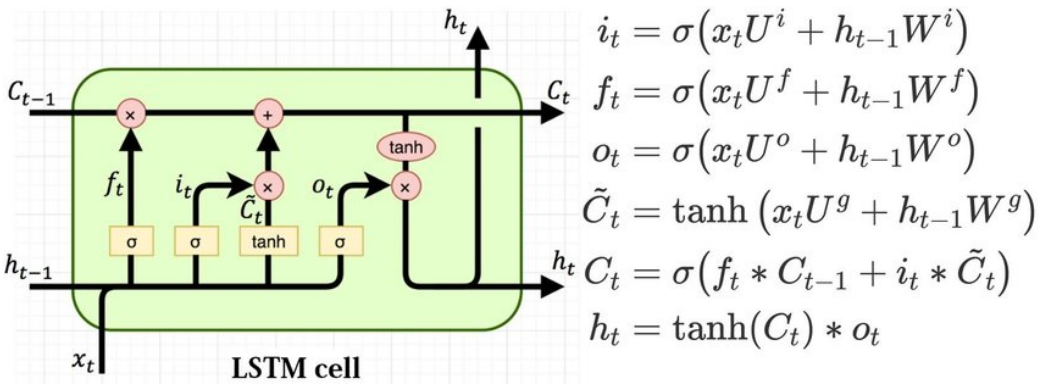
Obrázek 3: Operace sdružování podle maxima. Převzato z [14].

2.3.5 Rekurentní vrstva

Velmi zajímavý přístup představují rekurentní neuronové sítě, kde výstup jednoho uzlu může ovlivňovat vstup předchozího nebo toho samého. Rozdíl spočívá také ve zpracování dat, kde rekurentní sítě mohou zpracovávat sekvence dat se zachováním posloupnosti a časového kontextu.

Takovým příkladem je LSTM [20] neboli Long Short-Term Memory. Je to speciální verze rekurentní sítě s vnitřním stavem, který umožňuje síti vidět určitý kontext ve zpracovávané sekvenci. Je složena z LSTM buněk, které obsahují několik bran, konkrétně vstupní, dopřednou a zapomínací. Úkolem těchto bran je určit, jaké signály budou poslány dál. Tento princip je znázorněný na obrázku spolu se svými rovnicemi pro jednotlivé brány.

C reprezentuje vnitřní stav a vstup je označen jako x. V rovnicích je vidět váhy U a W, které zde reprezentují spojení k ostatním uzlům rekurentní sítě. Tyto sítě je možné trénovat právě pro detekci akustických událostí, například pro detekci slov, jako v práci [9].



Obrázek 4: Struktura LSTM buňky a rovnice jejích bran. Převzato z [16]

2.4 Úlohy pro rozpoznávání signálu

Rozpoznávání signálu může mít více podob, hlavně pro obrazová data, v této práci se setkáme ale pouze s detekcí a klasifikací. Hlavním rozdíl spočívá v přístupu k časové informaci.

Klasifikace tuto časovou informaci nezachovává a pouze rozhoduje, do jaké třídy daný vstup patří. Přístupy se zde liší pouze ve výstupu neuronové sítě, a to podle tříd. Při binární klasifikaci dává výstupní vrstva pouze binární výstup 0 nebo 1. Klasifikace ale může být i pro více tříd, a přesně s takovou úlohou se potýká tato práce.

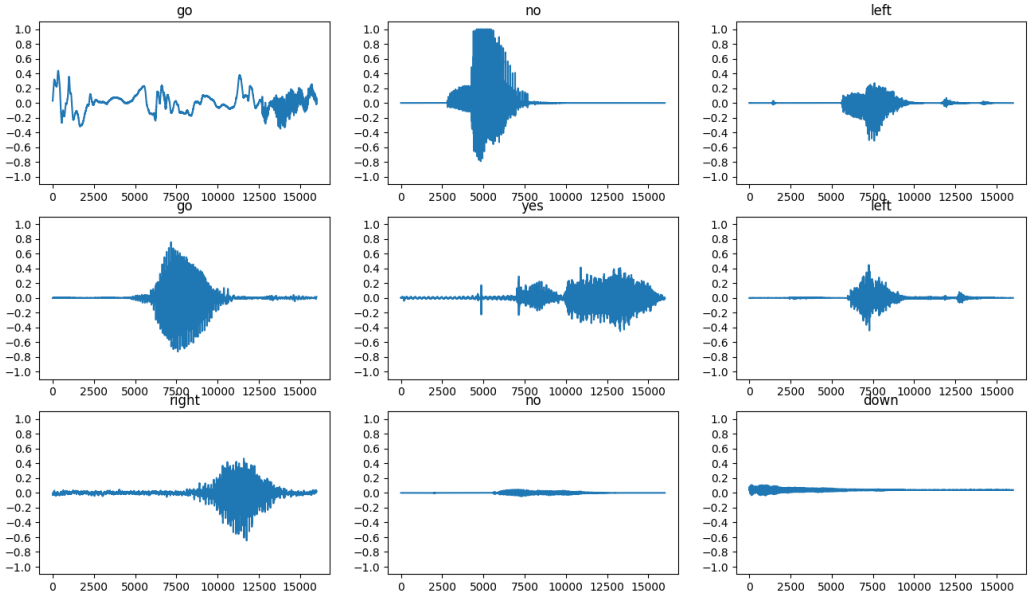
Při detekci je síti předkládán stream dat a pokud se v něm vyskytuje nějaká událost, síť podle toho zareaguje na výstupní vrstvě. Takový výstup může být opět buď binární ve smyslu vyskytuje/nevyskytuje a nebo přímo rozhodnutí, jaká třída události byla detekována. Díky takovému přístupu je ale také možné určit, kdy byla tato událost detekována.

2.5 Zpracování signálu

Velmi důležitá a různorodá část v trénování prakticky jakékoliv neuronové sítě je jak budou data zpracovány. Jak již bylo zmíněno, neuronové sítě mají určité požadavky na tvar jejich vstupu, ať už to je velikost jednotlivých batchů nebo tvar samotných prvků na vstupu. V oblasti audia je tato část ještě mnohem důležitější, jelikož možností existuje mnoho.

2.5.1 Signál z hlediska časového vývoje

Audio se dá v nejobyčejnější formě zobrazit a reprezentovat jako průběh hodnot v čase. Na obrázku 5 je možné vidět různé nahrávky pocházející z Speech Commands Datasetu [4] právě v této formě.



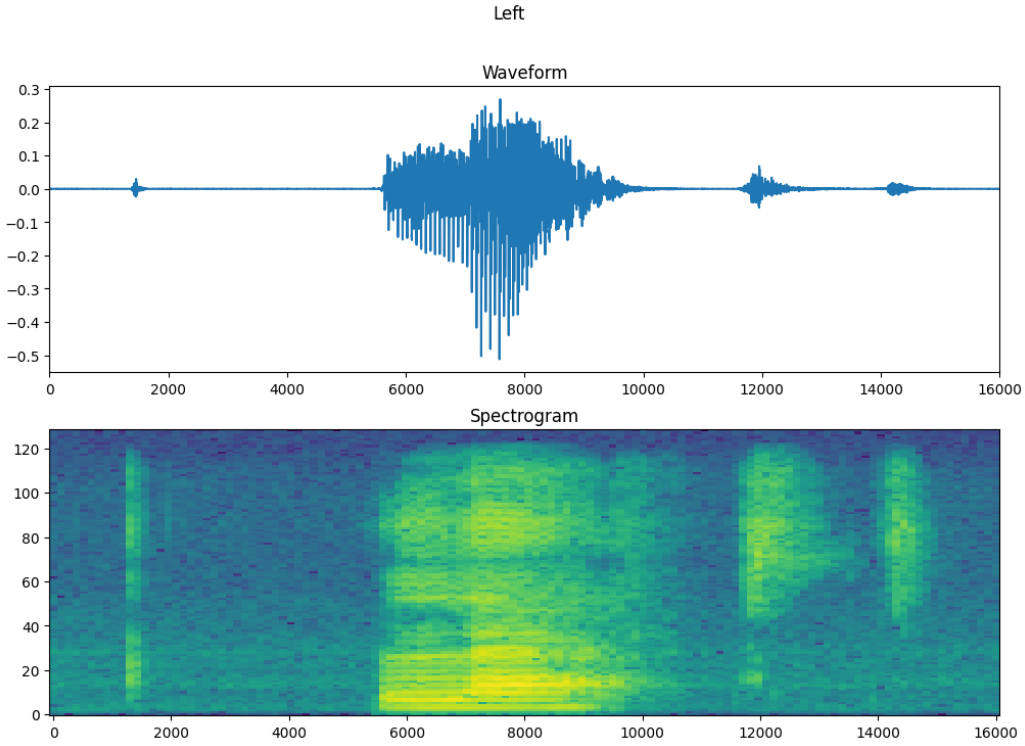
Obrázek 5: Zobrazení signálů pro různé nahrávky

Pro rozpoznávání audia se tato forma skoro nepoužívá a místo ní se využívá frekvencí obsažených v signálu.

2.5.2 Signál z hlediska frekvenčního vývoje

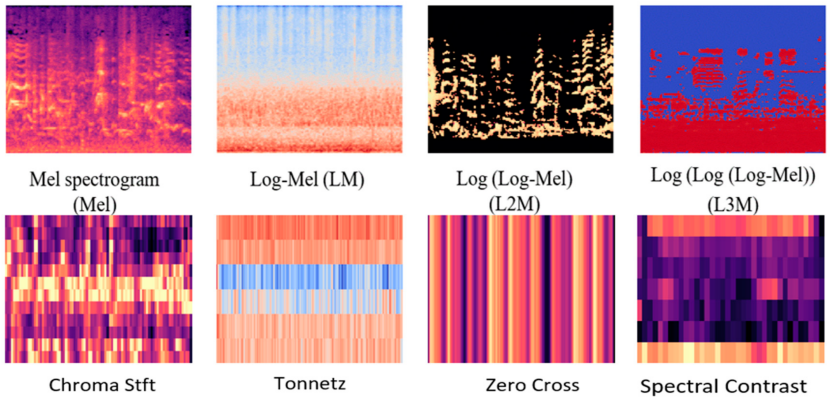
Neuronovým sítím se audio signál nejčastěji předkládá jako vývoj frekvencí v čase. Výhoda je zde hlavně velká možnost úpravy a augmentace samotných dat jako je využito například zde [6].

Využívá se Fourierovo transformace signálu po kouskách, které se nazývají okna, ty se následně průměrují. Po Fourierovo transformaci dostaneme spektrogram, který je zobrazen na obrázku 6.



Obrázek 6: Časový a frekvenční vývoj pro audio nahrávku slova "Left"

Tato forma nemusí být nutně konečná a může být rozvinuta pomocí dalších úprav do mnoha podob, které potom mohou být sloučeny do jednoho příznakového vektoru. Toho je využito v [7] způsobem ukázaným na obrázku 7.



Obrázek 7: Různé formy reprezentace audia ve frekvenční doméně [7]

Spektrogramy a jejich další formy se mohou lišit i ve způsobu jejich vytváření, kde se může měnit například délka jednotlivých oken, jejich překrývání a délka posunu. V této práci si ukážeme porovnání pro různé délky oken a posunů a jejich vliv na rozpoznání.

2.5.3 Augmentace dat

Augmentace dat slouží jako řešení nedostatku dat pro trénování a pomáhá k větší robustnosti modelu. Tato metoda spočívá v úpravě již existujících dat a rozšíření datasetu. Toho je docíleno u obrazových dat například otočením, rotací nebo zvýšením jasu. U audia je možné například nahrávky zrychlit nebo zpomalit, popřípadě přidat nějaké zvuky do pozadí a tím tak připravit model na budoucí data, která mohou být různorodá.

2.6 Přístup z hlediska délky zpracování

Tvar spektrogramu je matice $X \times Y$, kde osa Y reprezentuje časové měřítko relativně podle délky okna a posunu. Data se ovšem síti mohou předkládat dvěma způsoby. První možnost je předložit síti celý spektrogram, tato možnost je velmi používaná u konvolučních sítí a nejen zde pro klasifikaci, kde není podstatné časové zařazení. Druhá možnost je právě po framech. Jako frame se rozumí jeden sloupec spektrogramu a dá se na něj pohlížet jako na jeden příznakový vektor. Počet těchto vektorů nemusí být pevně daný a tak vstupní data mohou být například formou nekonečného streamu takových vektorů. Hlavní výhodou je zachování časové informace.

Právě pro LSTM sítě s vnitřním stavem je forma streamu velmi používaný způsob a díky tomu je možné pro LSTM buňky vidět souvislosti mezi framy.

2.7 Vyhodnocení výsledků

Pokud model již není ve fázi trénování (s učitelem), nemůže sám určit správnost svých výsledků. Proto je nutné jeho výstup při testování vyhodnocovat jinými způsoby.

Způsobů pro vyhodnocení výsledků strojového učení je opravdu mnoho a závisí na úloze, ať už například ROC křivka, míra falešných poplachů, absolutní chyba a nespočet dalších. V první řadě je dobré definovat si, z čeho výsledky vycházejí.

Data obsahují nebo neobsahují akustickou událost. Pokud ji data obsahují, označíme je jako P, positives. Pokud ji data neobsahují, označíme je jako N, negatives. Pokud jsou data P klasifikována jako P, značíme je jako True Positive (TP). Pokud jsou data P klasifikována jako N, značíme je False Negative (FN). Data N klasifikována jako P značíme False Positive (FP) a data N klasifikována jako N jsou True Negative (TN).

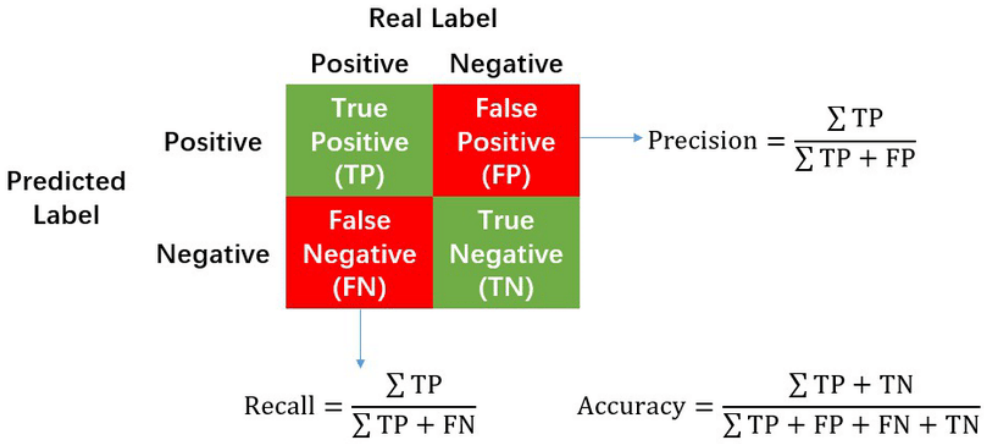
V této práci se vyskytují dva pojmy při vyhodnocení, které toto značení využívají, a to přesnost - podíl správných výsledků vůči všem a F1 míra, která je harmonickým průměrem dvou jiných vyhodnocovacích metod a to precisionu a recallu. Tyto vzorce jsou znázorněné na rovnicích 3, 4 a 5.

$$precision = \frac{TP}{TP + FP} \tag{3}$$

$$recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1míra = 2 \times \frac{precision \times recall}{precision + recall} \tag{5}$$

Vizualizace těchto metod vyhodnocení je na obrázku 8.



Obrázek 8: Vizualizace vyhodnocovacích metod. Převzato z [21].

Také je bráno v potaz, že model ukáže na výstupu správný výsledek z hlediska třídy ale nepřesný z hlediska času. Na to je ve výsledných metodách pro vyhodnocení parametr, který určuje časovou toleranci. To znamená, že pokud výstup modelu bude posunutý o 20 framů, což může znamenat například o 200 milisekund, stále se tento výsledek bude počítat jako správný.

3 Rozpoznávání hlasových povelů

V dnešní době se s rozpoznáváním hlasových povelů setkal již prakticky každý ve formě chytrých zařízení jako jsou například reproduktory a telefony. Jejich ovládání je možné pomocí zabudovaných hlasových asistentů jako například Siri [17] od Applu. Pro jejich aktivaci je nutné na začátku zavolat určitý příkaz jako třeba "Hey, Siri". Naše chytrá zařízení proto musí být schopná rozpoznat tyto hlasové povely a toho je docíleno právě pomocí strojového učení. Ačkoliv zde nejsou nutně využívány jen neuronové sítě ale například srovnávání se vzorem, je tato úloha skvělá na vyzkoušení modelů. Vhodný dataset pro takovou úlohu je Speech Commands Dataset.

3.1 Dataset a augmentace

Speech Commands Dataset [4], dále jen SCD, je populární dataset pro úlohy jako rozpoznávání klíčových slov, detekce a klasifikace audia. Má více verzí které se liší hlavně v počtu kategorií, v této práci je nicméně využita první verze tohoto datasetu.

Ta obsahuje dohromady přes 60 tisíc nahrávek anglicky mluvených slov, každá se vzorkováním 16kHz. Nahrávky jsou rozděleny do 30 tříd a do těch ve výsledku i klasifikujeme. Každá nahrávka patří pouze do jedné třídy.

Pro předzpracování dat byl zvolen způsob spektrogramů jako je ukázáno na obrázku 6. Pro vytvoření datasetu nebyl zvolen generátor, data se připraví jako jedna matice do paměti. Batch size, neboli počet spektrogramů ukázaných síti najednou, je roven 16. Data jsou dále rozdělena na trénovací, validační a testovací v poměru $0.8 : 0.1 : 0.1$.

Z důvodu velmi velkého rozsahu tohoto datasetu a modelů probíhá trénování na vzdáleném hardwaru a to za využití Google Colaboratory[1]. To je virutální prostředí podobné Jupyteru, ale je možné využít pomocné hardwarové prostředky. GPU použité pro toto trénování je pravděpodobně NVIDIA Tesla T4, grafická karta specializovaná na výpočty tohoto typu. Není samozřejmě možné využívat tyto prostředky neomezeně a časové omezení se mění úměrně s jeho využitím, tedy čím více se GPU používají, tím méně je

[1] https://colab.research.google.com/

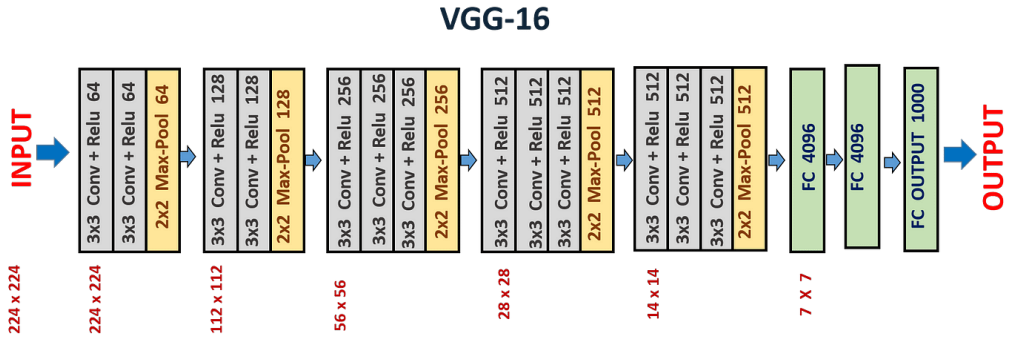
máte k dispozici.

3.2 Konvoluční sítě pro klasifikaci

První experimenty se zabývají využítím konvolučních sítí. Cílem bude klasifikace dat pomocí třech různých modelů a to velmi hluboký model VGG16 [24], residuální model ResNet34 [25] a vlastní vytvořený model, na které se blíže podíváme v následujících sekcích.

3.2.1 VGG16

Model VGG16 je velmi využívaný právě pro klasifikaci obrazových dat. Je to poměrně velký model ve smyslu počtu vrstev, kde obsahuje celkem 13 konvolučních vrtsev, 4 MaxPoollingové a tři plně propojené. Výsledný model je znázorněn na obrázku 9 a obsahuje přes 134 milionů parametrů, tudíž je potřeba decentního hardwaru pro jeho trénink. Ten je ale k dispozici na omezenou dobu díky službě Colaboratory. I tak ale pro snížení náročnosti zredukujeme velikost spektrogramů na 120×120. Výsledný model obsahuje stále přes 50 milionů parametrů.

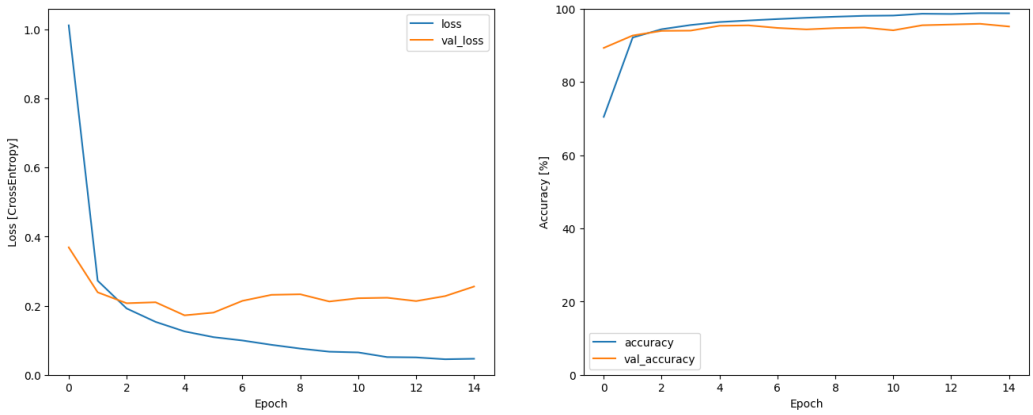


Obrázek 9: Architektura modelu VGG16. Převzato z [23].

Pro natrénování modelu byl použit optimalizátor Adam, který v porovnání [10] vykazuje nejlepší výsledky. Rychlost učení, také označovaná jako learning rate, byla nastavena na 0.0001. Cíl modelu je minimalizovat

ztrátovou funkci a tou je zde kategorická křížové entropie. Zjednodušeně je cíl maximalizovat přesnost klasifikace do tříd. Ta je měřena právě jako procentuální úspěšnost zaklasifikování.

Model byl nastaven pro trénování po dobu 40 epoch s automatickým zastavením, pokud validační ztrátová funkce nebude klesat 10 epoch po sobě. S tímto nastavením se model trénoval 14 epoch, kde přesnost na trénovacích datech dosáhla přes 98% a přes 95% na validačních. Více směrodatná je úspěšnost na testovacích datech, která je 94.89%. Průběh trénování je znázorněn na obrázku 10

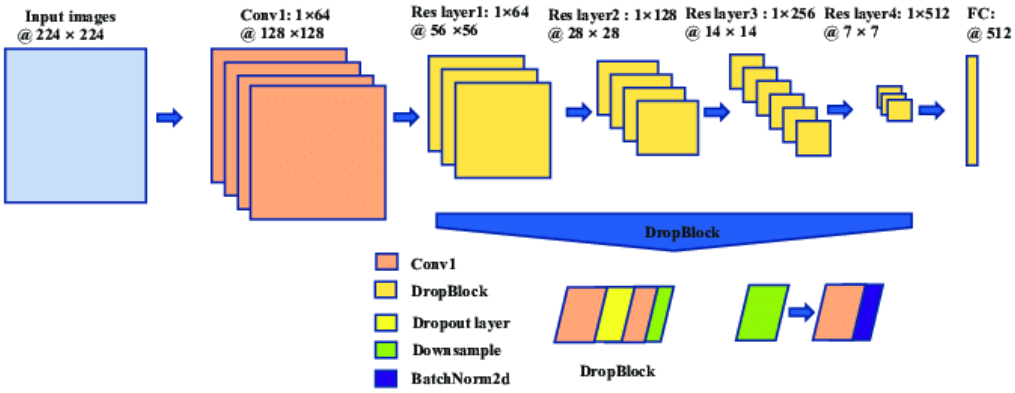


Obrázek 10: Průběh trénování pro model VGG16

Jako další úprava byla do modelu přidána jedna vrstva Dropout mezi plně propojené vrstvy, která nastaví určitý počet vah při každém průchodu na nulu a měla by pomoct hlavně s daty, které síť ještě neviděla. Testovací výsledky to zlepšilo o necelé jedno procento na 95.5%.

3.2.2 ResNet34

Druhý použitý model je model ResNet34, poměrně složitá neuronová konvoluční síť s obrovskou hloubkou, která se skládá z jednotlivých reziduálních bloků, které obsahují konvoluční a sdružující vrstvy. Tato implementace velkého množství vrstev předchází problému mizení gradientu, který jinak při velkém počtu vrstev a parametrů může nastat. Tato architektura je znázorněna na obrázku 11 a obsahuje přes 23 milionů parametrů a 34 vrstev.



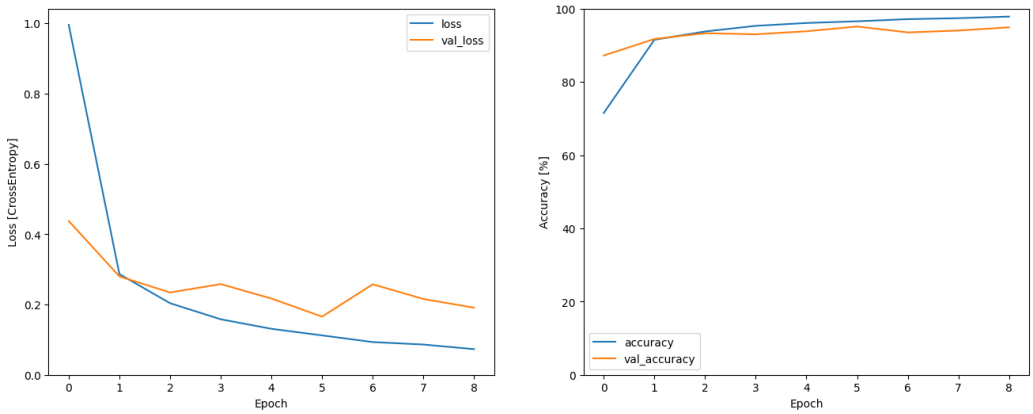
Obrázek 11: Architektura modelu ResNet34. Převzato z [22].

Modelů ResNet existuje více a číslo za nimi značí výsledný počet vrstev. Tento je obecně jeden z menších, jelikož existují i modely jako například ResNet152 využitý pro detekci Covidu z rentgenových snímků [11] a z hlediska implementace mohou být teoreticky i větší. Model byl trénován se stejným optimalizátorem jako předchozí a výsledky tohoto modelu jsou ukázané v tabulce 1 formou přesnosti.

Trénovací data	97.8
Validační data	94.9
Testovací data	94.3

Tabulka 1: Přesnost [%] modelu ResNet34

Na obrázku 12 můžeme ale vidět, že výsledné přesnosti nebyly nejlepší dosažené v rámci celého trénování a v jeho průběhu model dosáhl ještě nižší ztrátové funkce a o necelé procento lepších výsledků.



Obrázek 12: Průběh trénování modelu ResNet34

3.2.3 Vlastní model

Na předchozích modelech je vidět, že jsou velmi náročné na výpočetní výkon už jenom kvůli obrovskému počtu parametrů a velikosti. Proto je dalším cílem použít znalosti z předchozích modelů a vytvořit vlastní s ohledem na počet parametrů ale samozřejmě zachovat co nejvyšší přesnost.

Pro tuto úlohu fungují velmi dobře hluboké modely s více konvolučními vrstvami jak je možné vidět na ResNetu a VGG16. Konvoluční vrstvy sníží velikost obrazu, a tudíž poté plně propojená vrstva neobsahuje několik desítek milionů parametrů.

Díky výkonnému hardwaru od Google Colaboratory je možné vyzkoušet velmi hluboký model, který je znázorněný v tabulce 2. Počet parametrů přesahuje lehce 13 milionů. Parametry optimalizéru se opět neliší.

1x	Conv2D	512 filtrů, (5,5) jádro
4x	Conv2D	256 filtrů, (3,3) jádro
1x	MaxPooling	(2,2) jádro
1x	Dropout	0.1
1x	Conv2D	256 filtrů, (3,3) jádro
1x	MaxPooling	(2,2) jádro
1x	Conv2D	512 filtrů, (3,3) jádro
1x	MaxPooling	(2,2) jádro
1x	Conv2D	512 filtrů, (3,3) jádro
1x	MaxPooling	(2,2) jádro
1x	Flatten	
1x	Dropout	(0.1)
1x	Dense	1024, ReLU
1x	Dense	30, Sigmoid

Tabulka 2: Architektura vlastního modelu

Výsledky pro tento model jsou ukázané v tabulce 3 opět pomocí přesnosti.

Trénovací data	97.3
Validační data	94.3
Testovací data	94

Tabulka 3: Přesnost [%] vlastního modelu

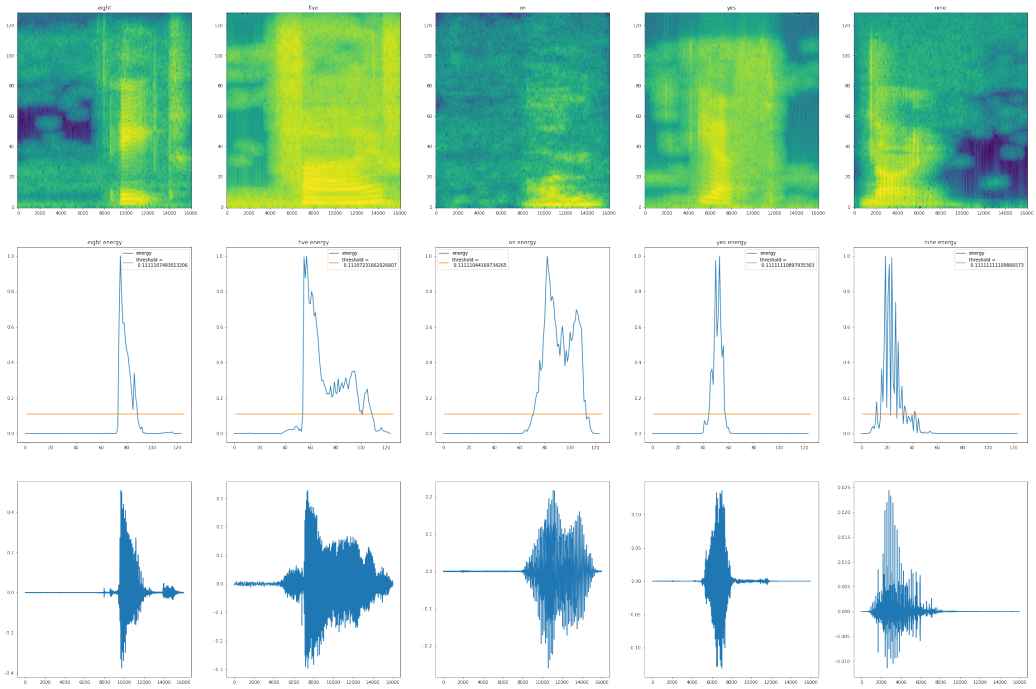
3.3 LSTM síť pro detekci

Druhá část experimentů na SCD se zabývá rekurentními neuronovými sítěmi. Pro detekci jednotlivých slov v nahrávce bude využita LSTM vrstva v kombinaci s plně propojenými vrstvami.

Předzpracování dat zde probíhá jinak, jelikož pro každý frame ve spektrogramu je nutné mít jednotlivý label. Toho je možné dosáhnout například za využití energie nahrávky. Z důvodu omezené paměti je také použita pouze třetina datasetu.

V první řadě se vytvoří opět spektrogram jako matice $X \times Y$ s posunem 10 milisekund a oknem 20 milisekund. Pro každý prvek osy X, která odpovídá časové ose, se vypočte následovně energie a to součtem mocnin všech Y vydělené délkou X. Na obrázku 13 můžeme vidět dole signál nahrávky, poté uprostřed jeho energii s vyznačeným prahem a nakonec spektrogram nahoře.

Pomocí také znázorněného prahu je určeno, na jakých framech se nachází dané slovo. Poté se vytvoří 1D pole, které reprezentuje labely pro daný spektrogram ve formě jedniček a nul. Jedničky jsou přiřazovány relativně k délce nahrávky pro budoucí využití a pokud událost je dlouhá 1 sekundu, labely se vytvoří pro posledních 0.4 sekundy.



Obrázek 13: Energie a spektrogramy pro nahrávky

3.3.1 LSTM model s ohledem na výkonnost

V prvním pokusu se budeme zabývat poměrně obsáhlým modelem se třemi LSTM vrstvy, kde každá obsahuje 1024 buněk a následně tři plně propojené vrstvy, kde první dvě mají 2048 neuronů a výstupní samozřejmě 30. Celkový počet parametrů modelu je přes 28 milionů.

Výstup modelu se prahuje a z prahovaného výstupu je poté vypočítána F1 míra. Z 800 testovacích spektrogramů je ukázaná jejich průměrná F1 míra v tabulce 4.

Tolerance	F1 míra
250 ms	73.76
150 ms	68.27
50 ms	53.99
10 ms	43.82

Tabulka 4: Výsledky LSTM modelu 1

3.3.2 LSTM model s ohledem na velikost

V druhém pokusu s rekurentními sítěmi je cílem vytvořit model, který je malý a rychlý, nicméně stále přesný. Ideální představa je pod milion parametrů a šest vrstev. Tato síť je tvořena třemi LSTM vrstvy se 128 buňkami, dvěmi plně propojenými se 512 neurony a výstupní plně propojenou vrstvou s 30 neurony. Opět je zde použita vrstva BatchNormalization. Počet parametrů se podařilo snížit na 800 tisíc.

Tento model splňuje velikostní požadavky ale výkonnostně velmi zaostává za předchozím, mnohonásobně složitějším modelem a tak můžeme tvrdit, že pro komplexní dataset jako je SCD, je potřeba obsáhlejší model. Výsledky můžeme vidět v tabulce 5

Tolerance	F1 míra
250 ms	5.4
150 ms	4.8
50 ms	3.4
10 ms	2.3

Tabulka 5: Výsledky zjednodušeného modelu

4 Detekce akustických událostí

Algoritmy ověřené a vyzkoušené v předchozí kapitole se nyní pokusíme aplikovat na novou oblast a nový dataset. Dále budeme zkoumat, zda jsou sítě s vnitřním stavem schopné zpracovávat data v reálném čase a tudíž vhodné pro nepřetržité používání na zařízeních s malým výpočetním výkonem, jako jsou například malá ESP zařízení s mikroprocesorem.

4.1 Dataset a augmentace

Z datasetu SCD se nyní přesuneme na vlastní dataset, který odpovídá reálným požadavkům na detekci akustických událostí. Při reálném použití může jít pouze o detekci jedné události, jako například rozbití skla ve sklárně, nebo i o detekci více událostí současně jako třeba z hlediska rozpoznávání bezpečnostních hrozeb. Použitý dataset obsahuje celkem 5 tříd: *Animal* - nahrávky psů a koček, *GlassBreak* - nahrávky tříštění skla, *Gunshot* - nahrávky výstřelů z malých zbraní, *Speech* - nahrávky ze Speech Commands Datasetu a *Scream* - nahrávky křiku. Kromě kategorie *Speech* pocházejí všechny nahrávky z internetové platformy Freesound[2]. Dataset obsahuje celkem 1209 nahrávek, ale některé z nich nebyly použité kvůli formátu, a tak po vyřazení jich je v datasetu 1160. Rozložení nahrávek v jednotlivých třídách není rovnoměrné. Údaje o datasetu jsou zobrazeny v tabulce 6.

Rozdělení a vlastnosti dat			
Trénovací split	Validační split	Testovací split	
936	104	120	
Třída	Počet nahrávek	Počet v testovacím splitu	Trvání
Animal	305	24	1 s - 3 s
Gunshot	396	55	$\leq 1s$
Glassbreak	199	17	0.5 s - 3.8 s
Speech	288	23	1 s
Scream	21	1	0.5 s - 4 s

Tabulka 6: Rozložení dat v datasetu

[2]https://freesound.org/

Z hlediska předzpracování se používají různé přístupy, které se liší v každém experimentu. Společné je vždy vytvoření 4sekundového pozadí pomocí bílého šumu, nahrávky ulice, nebo kombinace obou. Někdy se také používá pozadí bez šumu, tedy matice nul. Poté je do pozadí vložena nahrávka akustické události a vytvoří se spektrogram. Konkrétní parametry jsou uvedeny u každého experimentu.

4.2 Konvoluční sítě

Při použití konvolučních sítí máme za cíl otestovat dva různé přístupy v závislosti na délce segmentu nahrávky, který je vstupem do sítě. Prvním přístupem je předložení spektrogramu celé nahrávky, zatímco druhým přístupem je rozdělení nahrávky na menší segmenty, které jsou poté klasifikovány. Díky druhému přístupu je možné zachovat časovou informaci.

4.2.1 Zpracování celé nahrávky pomocí konvoluční sítě

Vstupem sítě je celý spektrogram 4sekundové nahrávky s posunem 10ms a oknem 20ms. Jako informace od učitele slouží jedno číslo označující požadovanou třídu. Architektura použitého modelu je ukázaná v tabulce 7.

1x	Resizing	128,128
1x	Conv2D	512 filtrů, (4,4) jádro
1x	Conv2D	512 filtrů, (3,3) jádro
1x	MaxPooling	(2,2) jádro
1x	Conv2D	256 filtrů, (3,3) jádro
1x	MaxPooling	(2,2) jádro
1x	Conv2D	256 filtrů, (3,3) jádro
1x	MaxPooling	(2,2) jádro
1x	Flatten	
1x	Dense	1024, ReLU
1x	Dense	5, Softmax

Tabulka 7: Architektura konvolučního modelu pro celé nahrávky

Celkový počet parametrů přesahuje 48 milionů, což z něj činí poměrně rozsáhlý model. Pro trénování byl použit optimalizátor Adam s učící konstantou 0,00001 a trénování probíhalo po dobu 18 epoch. Výsledky tohoto modelu jsou prezentovány v následujících tabulkách a to jak přesností celkovou, tak pro jednotlivé třídy.

	Animal	Glassbreak	Gunshot	Speech	Scream	Celková přesnost
Bíly šum	79.16	100	98.18	100	0	94.16
Pozadí z ulice	83.33	100	87.27	91.30	0	88.33
Kombinace	83.33	100	85.45	86.95	0	86.66
Data bez pozadí	79.16	100	98.18	100	0	94.16

Tabulka 8: Přesnost [%] konvolučního modelu pro celé nahrávky natrénovaného na nahrávkách bez pozadí

	Animal	Glassbreak	Gunshot	Speech	Scream	Celková přesnost
Bíly šum	75	94.12	92.73	100	0	90
Pozadí z ulice	79.16	94.12	98.18	95.65	0	92.50
Kombinace	79.16	94.12	98.18	95.65	0	92.50
Data bez pozadí	75	94.12	96.36	100	0	91.66

Tabulka 9: Přesnost [%] konvolučního modelu pro celé nahrávky natrénovaného na nahrávkách s pozadím ulice

	Animal	Glassbreak	Gunshot	Speech	Scream	Celková přesnost
Bíly šum	83.33	100	98.18	100	0	95
Pozadí z ulice	87.50	100	81.81	86.95	0	85.83
Kombinace	87.50	100	78.18	86.95	0	84.16
Data bez pozadí	83.33	100	98.18	100	0	95

Tabulka 10: Přesnost [%] konvolučního modelu pro celé nahrávky natrénovaného na nahrávkách s bílým šumem

	Animal	Glassbreak	Gunshot	Speech	Scream	Celková přesnost
Bíly šum	75	94.12	92.73	100	0	90
Pozadí z ulice	79.16	94.12	98.18	100	0	93.33
Kombinace	79.16	94.12	98.18	100	0	93.33
Data bez pozadí	75	94.12	96.36	100	0	91.66

Tabulka 11: Přesnost [%] konvolučního modelu pro celé nahrávky natrénovaného na nahrávkách s kombinací obou pozadí

4.2.2 Zpracování rozdělené nahrávky pomocí konvoluční sítě

Spektrogram pro 4 sekundovou nahrávku je zde rozdělený na 8 spektrogramů a to s posuvem 0.5s a délkou 1s. Toho je docíleno pomocí energie ve spektrogramu a labely jsou poté strojově přiřazeny pouze na ty spektrogramy, kterých energie přesahuje určitý práh.

Pro tento experiment byl zvolen jiný model ukázaný v tabulce 12.

1x	Conv2D	128 filtrů, (5,5) jádro
1x	MaxPooling	(2,12) jádro
1x	Conv2D	256 filtrů, (5,5) jádro
1x	MaxPooling	(5,10) jádro
1x	Flatten	
1x	Dense	256, ReLU
1x	Dense	5, Softmax

Tabulka 12: Architektura konvolučního modelu pro rozdělené nahrávky

I tento model byl při trénování nestabilní. Výstup modelu se poté prahuje aby se oddělily spektrogramy bez informace. Tedy pokud výstup žádného z 5 neuronů nepřesáhne mez 0.3, je výstup brán jako spektrogram bez informace.

Pro všech 8 spektrogramů jedné nahrávky je poté získaná přesnost, tedy tyto výsledky nejsou po framech ale po celých nahrávkách a slovy znamenají, zda a jak dobře síť v nahrávce, tedy ve všech 8 spektrogramech, detekovala událost. Po této úpravě můžeme vidět výsledky v tabulce 13. Přesnost po framech zde uvedená není, jelikož neodpovídala reálným výsledkům ani ztrátové funkci.

Data pro trénink	Testovací výsledky			
	Bíly šum	Pozadí z ulice	Kombinace	Bez pozadí
Bíly šum	73.22	32.51	32.51	73.55
Pozadí z ulice	38.40	35.47	35.47	38.48
Kombinace	57.23	47.01	46.32	57.25
Bez pozadí	74.59	46.24	46.24	75.22

Tabulka 13: Přesnost [%] konvolučního modelu pro rozdělené nahrávky

Není překvapivé, že nahrávky s pozadím tvořeného pouze nulami dosahují nejlepších výsledků, je zde totiž největší rozdíl mezi nahrávkami s událostí a bez ní a tedy pro síť jednodušší data na naučení. Při srovnání výsledků lze konstatovat, že konvoluční sítě dosahují větší přesnosti při použití celého spektrogramu nahrávky. Důvodem pro takové výsledky může být fakt, že síť se při celých nahrávkách soustředí pouze na daných 5 tříd a neexistuje zde možnost, že ve spektrogramu akustická událost není, tím se poté snižuje komplexita úlohy.

4.3 LSTM sítě

Pro LSTM sítě existuje více přístupů, které nás zajímají, a to z hlediska vnitřního stavu a z hlediska vstupních dat.

První dva provedené experimenty pracují s LSTM modelem, který obsahuje dvě LSTM vrstvy s 512 a 256 LSTM buňkami, po kterých následují dvě plně propojené vrstvy s 256 a 5 neurony. Jeho vnitřní stav zde není využívaný a takový model nevidí souvislosti mezi jednotlivými daty, jedná se tedy o stateless přístup. Tento model má 28.5 milionu parametrů a velikostně je tak srovnatelný s konvoluční sítí.

Na těchto dvou experimentech je cílem dostat jejich srovnání s konvolučními sítěmi a proto jim budou předloženy stejná data, a to jak trénovací tak testovací ve stejném pořadí.

4.3.1 Zpracování celých nahrávek pomocí LSTM modelu

Zde jsou stejně jako u konvoluční sítě použity celé spektrogramy nahrávek, ke kterým patří jeden label označující třídu. Optimalizátor Adam byl opět využitý, ale tentokrát s nižší učící konstantou, jelikož jinak model nekonvergoval k dobrým výsledkům. Trénování probíhalo po dobu 90 epoch, bylo tedy značně pomalejší. Výsledky tohoto modelu, které jsou ukázané v tabulce 23, velmi zaostávají za konvolučním modelem, hlavně z hlediska robustnosti modelu.

	Animal	Glassbreak	Gunshot	Speech	Scream	Celková přesnost
Bíly šum	83.33	88.23	90.90	100	0	90
Pozadí z ulice	79.16	88.23	100	86.95	0	90.83
Kombinace	79.16	88.23	100	95.65	0	92.50
Data bez pozadí	83.33	88.23	92.72	100	0	90.83

Tabulka 14: Přesnost [%] LSTM modelu pro celé nahrávky natrénovaného na nahrávkách s kombinací obou pozadí

	Animal	Glassbreak	Gunshot	Speech	Scream	Celková přesnost
Bíly šum	83.33	88.23	98.18	100	0	93.33
Pozadí z ulice	79.16	94.11	98.18	91.30	0	91.66
Kombinace	79.16	94.11	98.18	91.30	0	91.66
Data bez pozadí	83.33	88.23	94.54	100	0	91.66

Tabulka 15: Přesnost [%] LSTM modelu pro celé nahrávky natrénovaného na nahrávkách s pozadím ulice

	Animal	Glassbreak	Gunshot	Speech	Scream	Celková přesnost
Bíly šum	75	82.35	98.18	95.65	0	90
Pozadí z ulice	75	94.11	98.18	91.30	0	90.83
Kombinace	75	76.47	98.18	91.30	0	88.33
Data bez pozadí	79.16	94.11	98.18	91.30	0	91.66

Tabulka 16: Přesnost [%] LSTM modelu pro celé nahrávky natrénovaného na nahrávkách s bílým šumem

	Animal	Glassbreak	Gunshot	Speech	Scream	Celková přesnost
Bíly šum	79.16	94.11	98.18	82.60	0	90
Pozadí z ulice	75	94.11	98.18	91.30	0	90.83
Kombinace	75	94.11	96.36	82.60	0	88.33
Data bez pozadí	75	94.11	98.18	91.30	0	90.83

Tabulka 17: Přesnost [%] LSTM modelu pro celé nahrávky natrénovaného na nahrávkách bez pozadí

4.3.2 Zpracování rozdělené nahrávky pomocí LSTM modelu

V tomto experimentu bylo vyzkoušeno více modelů, regularizérů a vrstev. Trénování ale bylo vždy nestabilní a model se zaměřoval hlavně na spektrogramy bez informace, kterých je v datasetu většina. Oproti konvoluční síti v předchozím experimentu tedy nebylo možné tento model úspěšně otestovat.

4.3.3 Zpracování nahrávek za využití vnitřního stavu LSTM modelu

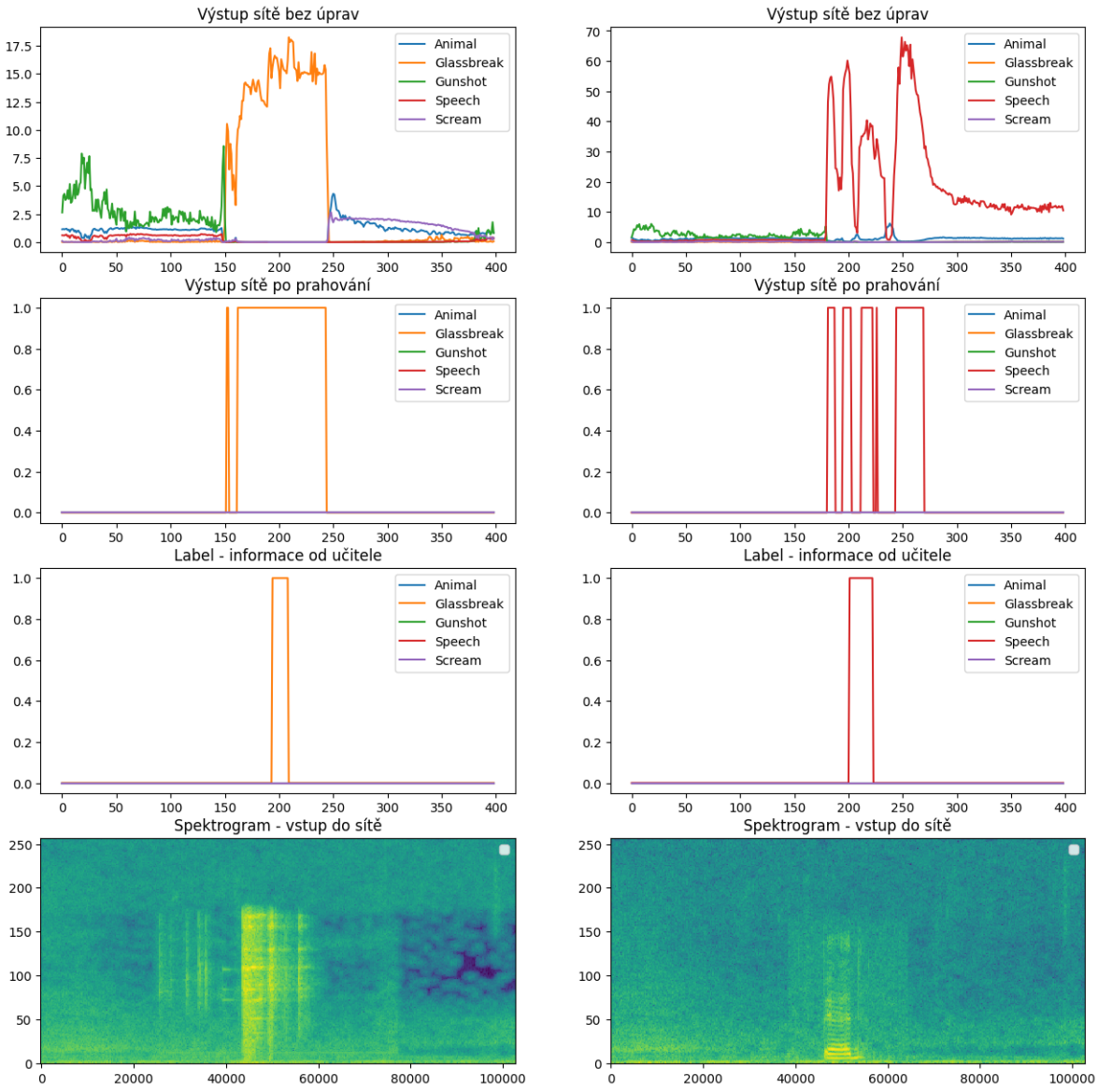
V této práci je využita výhoda LSTM sítě, která má vnitřní stav. Tento typ sítě je použit v experimentu k detekci akustických událostí. Z audio nahrávek jsou vytvořeny spektrogramy s různými parametry. Pro každý frame v spektrogramu jsou vytvořeny labely pomocí energie, kde je událost označena, pokud je energie tohoto framu větší než práh. Práh je určen z maxima energie v daném spektrogramu a je pro každou nahrávku odlišný.

Síť pracuje s daty po framech, kde pro každý sloupec spektrogramu, reprezentujících 20 milisekund zvukové nahrávky, jsou na výstupu data z 5 neuronů. Tento přístup umožňuje zachování časové informace a sítě tak dokážou vnímat souvislosti v signálu. Nicméně kvůli malému množství dat a jejich velké variaci, algoritmus pro vytváření informace od učitele, který fungoval na SCD, není tak přesný a některé labely jsou posunuty na časové ose.

Výstupní vrstva používá aktivační funkci softplus a výstup je poté prahován, aby byly odstraněny výstupy s nízkou pravděpodobností. Prahovací konstanta může být přizpůsobena a je mezi 0 a 1. Výstup sítě je normalizován od 0 do 1 a vše nad prahovací konstantou je označeno jako 1 a zbytek jako 0. Celý tento proces je ilustrován na obrázku 14, kde je prahovací konstanta nastavena na 0.9.

Cílem této práce je zhodnotit výkonnost LSTM sítě s vnitřním stavem při detekci akustických událostí a také zjistit, jaký vliv má princip neurčitosti při tvorbě spektrogramů. Tento princip se projevuje v tom, že výsledný spektrogram se může lišit v závislosti na velikosti okna pro provedení Fourierovy transformace.

Vzhledem k tomu, že dosud používané modely jsou velmi časově a výpočetně náročné, s mnoha vrstvami a desítkami milionů parametrů, zvažuje se nasazení na zařízení s malým výpočetním výkonem. Ideálním výsledkem je síť, která bude schopna zpracovat data v reálném čase.



Obrázek 14: Zpracování signálu LSTM sítí

Navržený LSTM model byl vytvořen s ohledem na jeho velikost a vychází pouze z doposud získaných znalostí.

Obsahuje dvě LSTM vrstvy s 512 a 256 buňkami, následované třemi plně propojenými vrstvami s 512, 256 a výstupními 5 neurony. Celkově model obsahuje 2.6 milionu parametrů pro data s oknem 20ms a 3.1 milionu pro data se 40ms oknem kvůli rozdílným dimenzím vstupních spektrogramů. Trénování probíhá pomocí optimalizátoru Adam a učící konstanta se v průběhu tréninku snižuje od 0.01 pro dosažení nejlepších výsledků.

Během testování bylo zjištěno, že model zanedbával třídu Gunshot nebo měl tendenci řadit všechna data do jedné třídy. Pro zlepšení výsledků byly vyzkoušeny regularizéry, konkrétně tři typy pro jádro sítě, vstup a výstup. Tyto regularizéry zlepšily průběh učení a výkonnost sítě zejména na datech, které síť ještě neviděla, a tím byla síť robustnější. Výsledný model implementuje L2 regularizátory a byl opět vyhodnocen pomocí F1 míry, Precisionu a Recallu s tolerancí časové odchylky 200 ms.

Výstup modelu se opět prahuje a to hodnotou 0.99, jelikož výstupní funkce softmax dává vždy v součtu 1 a modely měly tendenci u jedné třídy dávat všude poměrně vysoké pravděpodobnosti a tento práh spolehlivě odstranil nechtěné výsledky, ukážeme si zde ale také porovnání prahovaných výsledků s výsledky, na které je po prahování provedena opearace dilatace. Dilatace prahované výsledky sjednotí a vytvoří tedy souvislý časový úsek detekované nahrávky. Pro události, které jen jen na velmi malém počtu framů tím ale může vznikat nepřesnost, proto se na tyto výsledky podíváme a porovnáme tyto dvě metody. Výsledky jsou uvedeny v následujících tabulkách.

Spektrogramy - 20 ms posun, 30 ms okno				10 ms posun, 15 ms okno		
Testovací data	F1 míra	Precision	Recall	F1 míra	Precision	Recall
Bílý šum - dilatace	71.01	61.86	86.96	70.85	58.10.	93.30
Bílý šum - prahování	60.13	72.24	55.62	70.98	64.91	82.35
Pozadí z ulice - dilatace	51.01	40.06	76.33	34.76	22.92	88.37
Pozadí z ulice - prahování	51.66	66.30	45.69	35.55	26.78	73.37
Kombinace - dilatace	44.37	35.89	66.44	26.08	16.08	83.30
Kombinace - prahování	42.39	58.58	37.23	24.18	15.90	63.96
Data bez pozadí - dilatace	27.60	17.27	82.99	71.21	60.20	91.53
Data bez pozadí - prahování	53.12	53.89	55.08	72.83	71.98	78.76

Tabulka 18: Výsledky stateful modelu přes framy natrénovaného na nahrávkách s bílým šumem

Spektrogramy - 20 ms posun, 30 ms okno				10 ms posun, 15 ms okno		
Testovací data	F1 míra	Precision	Recall	F1 míra	Precision	Recall
Bílý šum - dilatace	13.61	10.07	20.99	24.70	14.56	88.16
Bílý šum - prahování	17.80	10.53	72.95	21.81	13.05	72.79
Pozadí z ulice - dilatace	64.13	49.87	93.45	65.78	55.75	82.22
Pozadí z ulice - prahování	59.53	53.02	76.87	58.97	55.31	66.00
Kombinace - dilatace	64.34	52.20	90.82	63.24	56.03	78.02
Kombinace - prahování	55.56	54.51	71.08	54.84	55.32	61.96
Data bez pozadí - dilatace	14.99	0.08	89.38	21.02	12.23	90.75
Data bez pozadí - prahování	19.47	11.76	60.61	41.23	30.26	72.97

Tabulka 19: Výsledky stateful modelu přes framy natrénovaného na nahrávkách s pozadím ulice

Spektrogramy - 20 ms posun, 30 ms okno				10 ms posun, 15 ms okno		
Testovací data	F1 míra	Precision	Recall	F1 míra	Precision	Recall
Bílý šum - dilatace	26.99	17.33	68.79	53.36	39.79	87.78
Bílý šum - prahování	33.86	28.46	44.77	56.60	51.41	69.50
Pozadí z ulice - dilatace	65.40	56.75	80.84	63.95	54.83	84.74
Pozadí z ulice - prahování	55.07	72.19	48.63	52.81	51.50	64.2
Kombinace - dilatace	69.91	60.82	84.35	63.24	54.11	80.58
Kombinace - prahování	54.42	72.71	46.91	60.55	62.62	64.28
Data bez pozadí - dilatace	51.25	38.11	85.62	22.78	13.59	82.18
Data bez pozadí - prahování	54.63	55.87	56.69	39.17	30.11	60.66

Tabulka 20: Výsledky stateful modelu přes framy natrénovaného na nahrávkách s kombinací obou pozadí

Spektrogram - 20 ms posun, 30 ms okno				10 ms posun, 15 ms okno		
Testovací data	F1 míra	Precision	Recall	F1 míra	Precision	Recall
Bílý šum - dilatace	18.46	10.68	91.04	65.84	58.26	77.93
Bílý šum - prahování	18.57	11.02	74.12	59.84	64.03	59.20
Pozadí z ulice - dilatace	20.99	14.41	79.51	43.64	36.28	63.02
Pozadí z ulice - prahování	25.00	16.21	65.34	36.45	40.27	42.67
Kombinace - dilatace	33.85	21.72	85.87	35.30	26.52	58.73
Kombinace - prahování	46.75	39.28	65.39	33.02	31.65	41.66
Data bez pozadí - dilatace	69.18	54.22	98.80	61.03	53.35	78.79
Data bez pozadí - prahování	70.69	61.34	88.30	50.05	51.59	57.66

Tabulka 21: Výsledky stateful modelu přes framy natrénovaného na nahrávkách bez pozadí

Tyto výsledky nejsou tolik vypovídající, jelikož LSTM model vždy událost posune, je zde také mnoho proměnných jako délka okna, velikost prahu či dilatace a při posunu prahu může například precision růst a recall klesat a naopak. To můžeme vidět v porovnání výsledků s dilatací a prahem, kde výsledky s dilatací mají skoro vždy větší precision ale menší recall.

Proto jsou výsledky vyhodnoceny ještě druhou metodou, která dává výsledek pro celou nahrávku a to zda událost byla nebo nebyla správně nalezena. Přesnost je zde 100% pokud je nalezena pouze správná třída, 0% pokud správná třída detekována není a pokud je nalezena správná třída a nějaké další, je přesnost počítána jako 1/počet nalezených tříd.

Tyto výsledky v následujících tabulkách jsou tedy porovnatelné s výsledky pro celé nahrávky v předchozí části práce.

	Animal	Glassbreak	Gunshot	Speech	Scream	Celková přesnost
Bíly šum	91.66	88.23	78.18	91.30	0	84.16
Pozadí z ulice	83.33	88.23	80.00	86.95	0	82.50
Kombinace	87.50	88.23	67.27	86.95	0	77.50
Data bez pozadí	85.41	88.23	83.63	100	0	87.08

Tabulka 22: Přesnost [%] stateful LSTM modelu přes celé nahrávky natrénovaného na nahrávkách bez pozadí

	Animal	Glassbreak	Gunshot	Speech	Scream	Celková přesnost
Bíly šum	43.75	97.05	0	39.13	0	30.00
Pozadí z ulice	79.16	91.17	61.81	95.65	0	75.41
Kombinace	79.16	91.17	60.90	86.95	0	73.33
Data bez pozadí	66.66	94.12	20.60	43.47	0	44.44

Tabulka 23: Přesnost [%] stateful LSTM modelu přes celé nahrávky natrénovaného na nahrávkách s pozadím ulice

	Animal	Glassbreak	Gunshot	Speech	Scream	Celková přesnost
Bíly šum	75.00	100	18.18	95.65	0	55.83
Pozadí z ulice	87.50	88.23	0	82.60	0	45.83
Kombinace	87.50	88.23	0	80.43	0	45.41
Data bez pozadí	75.00	100	14.54	97.82	0	54.58

Tabulka 24: Přesnost [%] stateful LSTM modelu přes celé nahrávky natrénovaného na nahrávkách s bílým šumem

	Animal	Glassbreak	Gunshot	Speech	Scream	Celková přesnost
Bíly šum	70.83	94.12	68.18	93.47	0	76.66
Pozadí z ulice	70.83	91.17	82.72	95.65	0	83.33
Kombinace	70.83	94.12	82.72	93.47	0	83.33
Data bez pozadí	70.83	94.12	71.81	95.65	0	75.75

Tabulka 25: Přesnost [%] stateful LSTM modelu přes celé nahrávky natrénovaného na nahrávkách s kombinací obou pozadí

Při pozorování výsledků bylo zjištěno, že největší problém této sítě byly třídy animal a gunshot, jelikož jejich průběh byl někdy velmi podobný a proto model většinou detekoval pro třídu gunshot obě třídy gunshot a animal, někdy je také zaměňoval.

Výpočetní doba vyhodnocení a prahování 8 spektrogramů je přibližně 3 sekundy, což znamená, že za tuto dobu lze zpracovat časový úsek délky 32 sekund. Tento proces probíhá na výkonném procesoru Intel Core i7-8700 s použitím jednoho jádra. Je důležité poznamenat, že tato doba nezahrnuje vytváření spektrogramu a zpracování nahrávky. Bohužel by na mikroprocesoru pravděpodobně nebylo možné zpracovávat data v reálném čase pomocí tohoto modelu. Nicméně by se to dalo řešit pomocí prahu pro energii signálu a rozpoznáváním pouze v případě, že energie překročí stanovenou úroveň.

Je očekávané, že stejné jako ostatní i tato síť měla potíže s klasifikací třídy scream, která měla v datasetu mnohem méně nahrávek než ostatní třídy. Je patrné, že třídy s větším počtem nahrávek v datasetu dosahují tedy lepších výsledků. Tento fakt je podpořen i výsledky konvoluční a stateless sítě, kde třída s největším počtem nahrávek dosahuje průměrně nejlepších výsledků.

Pro dosažení optimálních výsledků je tedy důležité vytvořit dataset s rovnoměrným počtem nahrávek pro každou třídu, ale existují techniky jako regularizéry, díky kterým je možné dosáhnout optimálních výsledků i s malým a nevyrovnaným datasetem.

5 Závěr

Tato práce se zaměřila na algoritmy strojového učení pro rozpoznávání událostí v akustickém signálu. V první kapitole jsou uvedeny teoretické informace, zejména o neuronových sítích a jejich vrstvách, které jsou v následujících kapitolách použity. Zbylé kapitoly tvoří praktická část, která ověřuje algoritmy na dvou různých datasetech, nejprve SCD a následně na vlastním.

Výsledky ukazují, že s nárůstem komplexnosti dat roste i nárok na větší složitost modelu a s tím se zvyšuje časová a výpočetní náročnost, což omezuje použití na mikroprocesorech s počítáním v reálném čase. Tento problém lze řešit klasifikací signálu pouze s dostatečnou energií, nebo s využitím dvou neuronových sítí. První by sloužila k detekci události v časovém úseku bez konkrétního zařazení a byla by jednodušší a zpracovatelná v reálném čase. Druhá by sloužila k následné klasifikaci označeného úseku. Pro takovou konfiguraci by první mohla být LSTM síť s vnitřním stavem pro detekci a následně konvoluční síť pro klasifikaci.

Pokud není brána v úvahu výpočetní náročnost, LSTM model s vyšším počtem parametrů by mohl být velmi dobrým způsobem pro detekci v reálném čase. Pokud by byl dostatek trénovacích dat pro dosažení robustnosti, mohl by tento model podávat velmi dobré výsledky.

Seznam obrázků

Seznam tabulek

Reference

[1] McCulloch W.S., W. Pitts, (1943), A logical calculus of the ideas immanent in nervous activity. Dostupné na: http://dx.doi.org/10.1007/BF02478259

[2] Rosenblatt F, (1958) The perceptron: A probabilistic model for information storage and organization in the brain. Dostupné na: https://doi.org/10.1037/h0042519

[3] Dechter R.(1986), Learning while searching in constraint-satisfaction problems. Dostupné na: https://cdn.aaai.org/AAAI/1986/AAAI86-029.pdf

[4] Warden P. (2018), Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. Dostupné na: http://arxiv.org/abs/1804.03209

[5] Huzaifah M. (2017), Comparison of Time-Frequency Representations for Environmental Sound Classification using Convolutional Neural Networks. Dostupné na: http://arXiv:1706.07156

[6] Coimbra de Andre D., Leo S., Viana M., Bernkopf Ch. (2018), A neural attention model for speech command recognition. Dostupné na: http://arXiv:1808.08929

[7] Mushtaq Z., Su S. (2020), Efficient Classification of Environmental Sounds through Multiple Features Aggregation and Data Enhancement Techniques for Spectrogram Images. Dostupné na: http://arXiv:1808.08929

[8] Zhang Y., Suda N., Lai L., Chandra V. (2017), Hello Edge: Keyword Spotting on Microcontrollers. Dostupné na: http://arXiv:1711.07128

[9] Bulín M., Šmídl L., Švec J. (2019), On Using Stateful LSTM Networks for Key-Phrase Detection. Dostupné na: https://www.researchgate.net/publication/335551026

[10] Dogo M. E., Afolabi J. O., Twala B. (2022), On the Relative Impact of Optimizers on Convolutional Neural Networks with Varying Depth and Width for Image Classification. Dostupné na: https://doi.org/10.3390/app122311976

[11] Pustokhin D., Puskothina I., Dinh N. P. (2020), OAn effective deep residual network based class attention layer with bidirectional LSTM for diagnosis and classification of COVID-19. Dostupné na: https://www.researchgate.net/publication/347170147

[12] Harár P. (2019), Audio classification with deep learning on limited data sets. Doctoral thesis. Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications. Vedoucí práce Ing. Jiří Mekyska Ph.D.

[13] Bláha M. (2019), Koncept umělé neuronové sítě, Portál matematické biologie. Dostupné na: https://portal.matematickabiologie.cz/

[14] Computer Science Wiki. Dostupné na: https://computersciencewiki.org/index.php/File:MaxpoolSample2.pngfile

[15] COSMOS: THE PORTAL FOR USERS OF ESA'S SCIENCE DIRECTORATE'S MISSIONS Dostupné na: https://www.cosmos.esa.int/web/machine-learning-group/convolutional-neural-networks-introduction

[16] Varsamopoulos S., Bertels K. (2018), Designing neural network based decoders for surface codes. Dostupné na: https://www.researchgate.net/publication/329362532

[17] Apple, Siri Dostupné na: https://www.apple.com/siri/

[18] Agarap A.F. (2018), Deep Learning using Rectified Linear Units (ReLU) Dostupné na: https://doi.org/10.48550/arXiv.1803.08375

[19] Kinga P.D., Ba J. (2014), Adam: A Method for Stochastic Optimization Dostupné na: https://doi.org/10.48550/arXiv.1412.6980

[20] Hochreiter S., Schmidhuber J. (1997), Long Short-term Memory Dostupné na: https://www.researchgate.net/publication/13853244$_{L}ong_{S}hort - term_{M}emory$

[21] Ma J., Ding Y., Cheng P.C.J., Tan Y. (2019), Analyzing the Leading Causes of Traffic Fatalities Using XGBoost and Grid-Based Analysis: A City Management Perspective Dostupné na: https://www.researchgate.net/publication/336402347

[22] Nie L., Xiong G., Shen Z., Pan Z. (2021), Face Image Based Automatic Diagnosis by Deep Neural Networks Dostupné na: https://www.researchgate.net/publication/354224999

[23] Khandelwal V. (2020), The Architecture and Implementation of VGG-16 Dostupné na: https://pub.towardsai.net/16-b050e5a5920b

[24] Simonyan K., Zisserman A. (2015), Very Deep Convolutional Networks for Large-Scale Image Recognition Dostupné na: https://arxiv.org/abs/1409.1556

[25] He K., Zhang X., Ren S., Sun J. (2015), Deep Residual Learning for Image Recognition Dostupné na: https://arxiv.org/abs/1512.03385v1