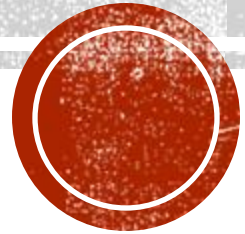
KLASIFIKACE ZVUKOVÉHO SIGNÁLU

Tvorba titulků pro neslyšící



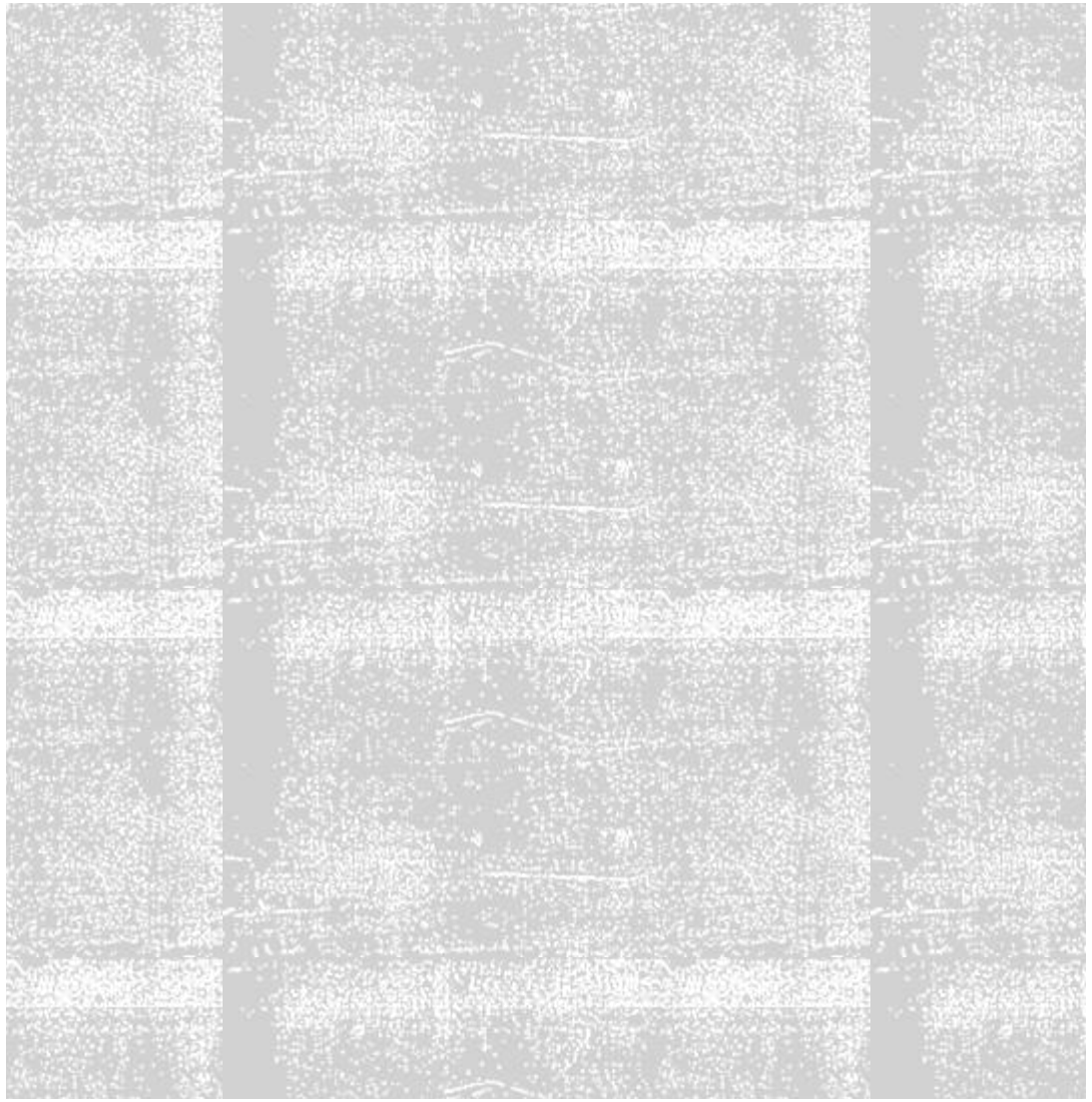
Filip Širc
Martin Kunz

CÍLE PROJEKTU

- Automatické využití dat z YouTube
- Efektivní klasifikace zvukových dat
- Klasifikace pro každou sekundu zvukové stopy

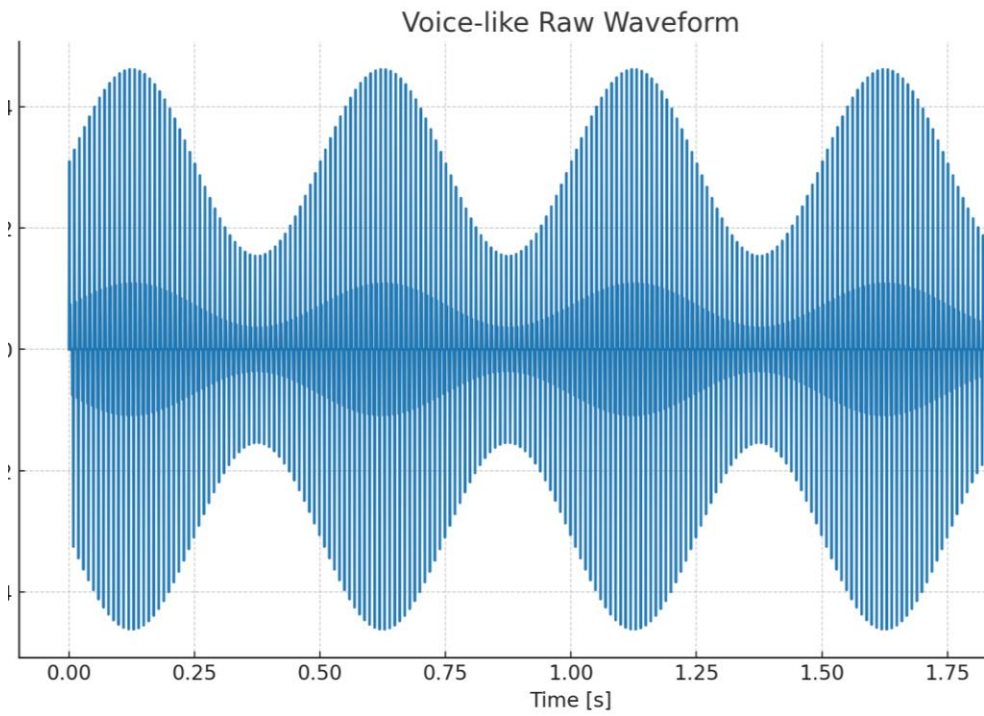
Dál už mi nic neříkejte! Dejte mi papír, tužku a kružidlo a nechte mne na pokoji. Však já na to přijdu sám!

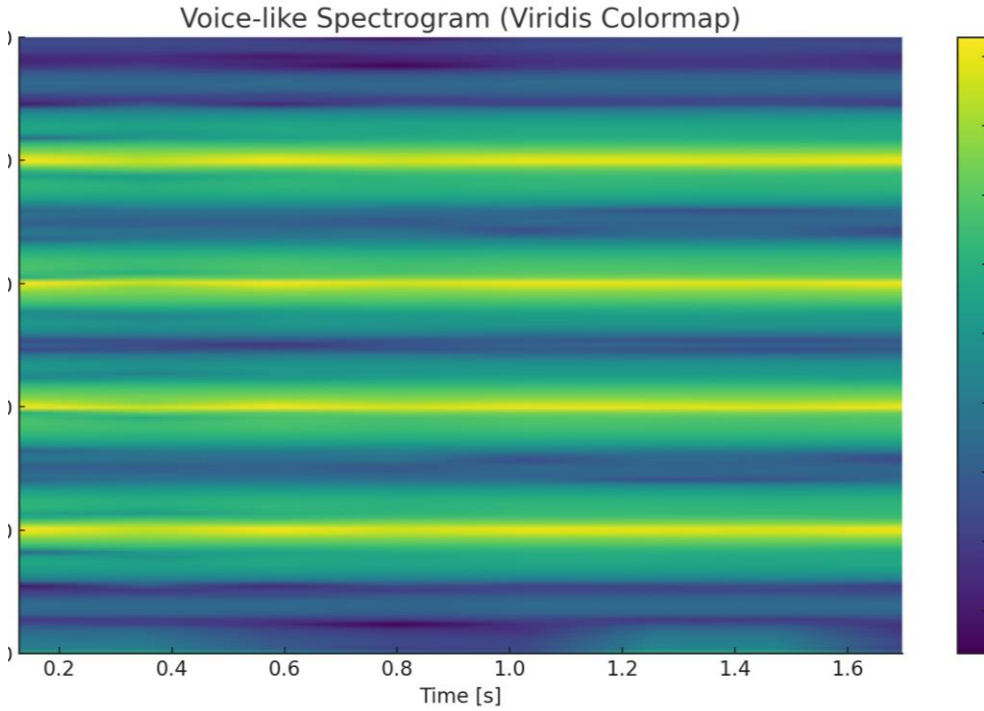
[Zdeněk Jirotka - Saturnin]



TEORETICKÝ ÚVOD

- **Zvuk** je mechanické vlnění, které se šíří látkovým prostředím
- Jednotlivé zvukové signály se mohou lišit
  - Amplitudou
  - Složením frekvencí
  - Vývojem frekvencí v čase
- Zvuk lze reprezentovat jako waveform či spektrogram
  - Waveform lépe reflektuje informace o amplitudě
  - Spektrogram o složení frekvencí

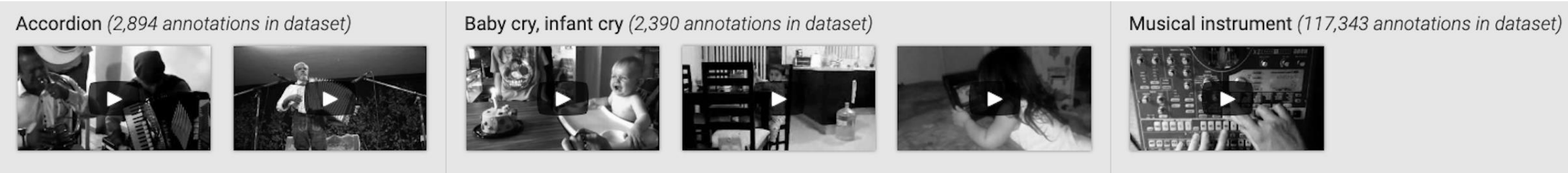




ZÍSKÁVÁNÍ DAT

- **Google Audioset** – 2 100 000 anotovaných videí v mnoha kategoriích
  - Uloženo formou odkazů na youtube videa se začátkem a koncem
  - Každé video může obsahuje více kategorií
  - Nejedná se o "čistý" zvuk - obsahuje velké množství šumu
  - Chybovost anotací cca 5 – 10% dle kategorie

- **Yt_dlp –** python knihovna pro těžení dat z youtube, pro stažení dat je nutné
  - Otevřít okno v chromiu a odscrollovat až na konec při zohlednění času na load
  - Stáhnou meta data o jednotlivých videích - id videa, label, čas začátku a konce zvukového záznamu
  - Stáhnout si vybraná videa podle metadat



PLÁN IMPLEMENTACE

"Po částech"

- Naučíme se klasifikovat "čisté" zvukové stopy
- Výslednou stopu rozsekáme na části, které budeme evaluovat

Výhody

- Rychlejší učení

Nevýhody

- Horší klasifikace "přechodů" mezi zvuky

"Dohromady"

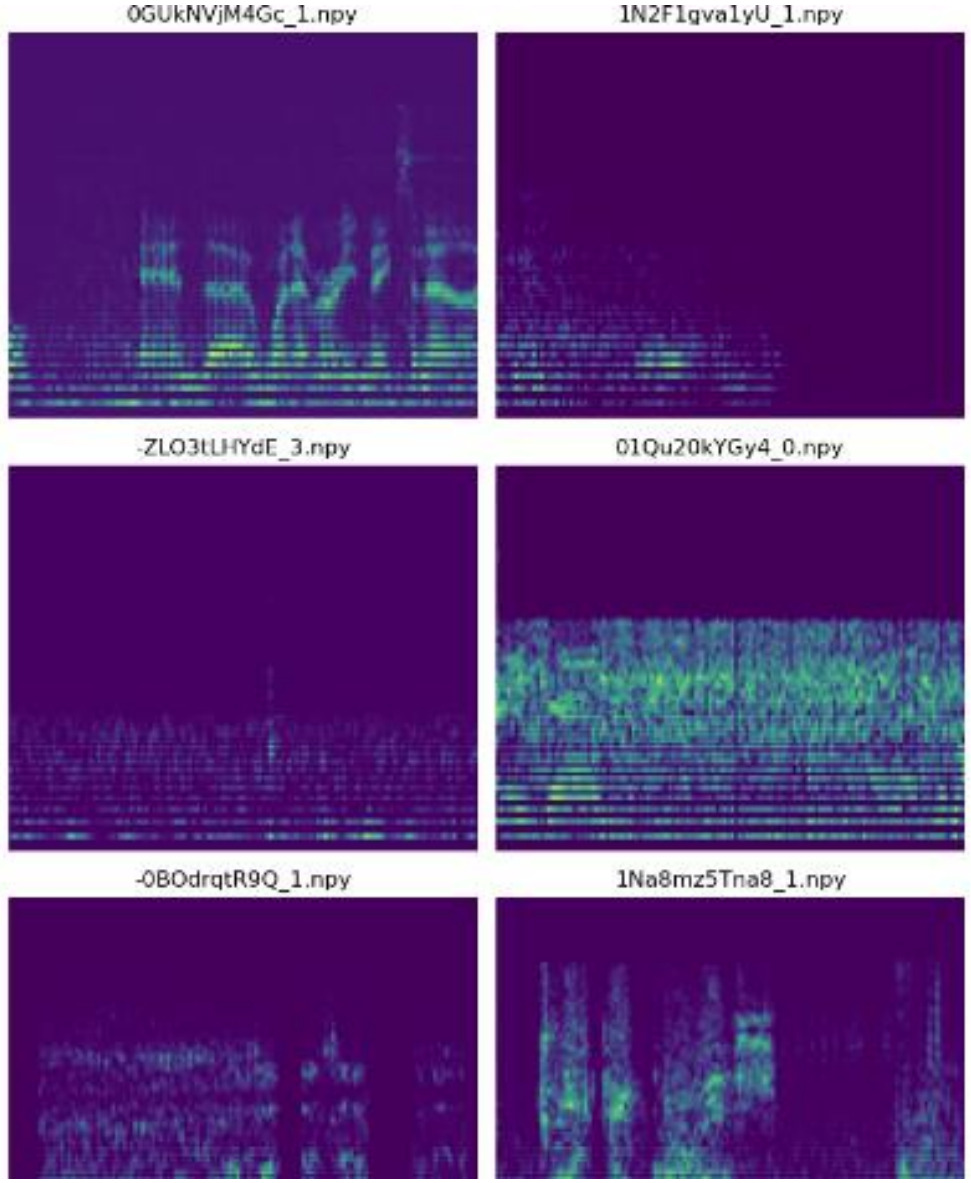
- Uměle si vytvoříme anotované zvukové stopy spojováním menších celků
- Síť budeme učit na takto anotovaných datech
- Výslednou stopu budeme evaluovat "as is"

Výhody

- Vyšší citlivost na změny ve zvuku

Nevýhody

- Větší výpočetní náročnost
- Větší nároky na čistotu dat



PŘÍPRAVA DAT - SPEKTROGRAMY

- Zvukový signál je důležité očistit a napočítat spektogramy
  - Sampling rate - uniformovat
  - Délka signálu
  - Normalizace
- Výpočet spektogramů
  - Rozlišení
    - 128x128x1
    - 256x256x3
  - Typ
    - Klasický
    - Melův

ZÁKLADNÍ MODEL

Konvoluční neuronová síť

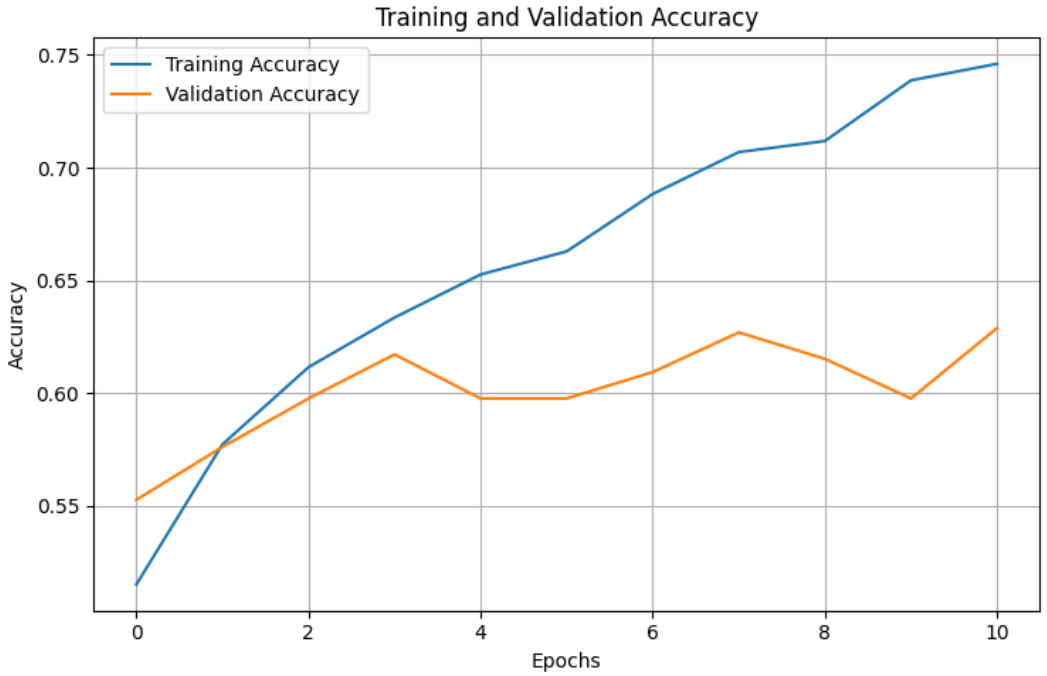
- 4 konvoluční vrstvy (32, 64, 128, 256)
- Fully connected klasifikační vrstva

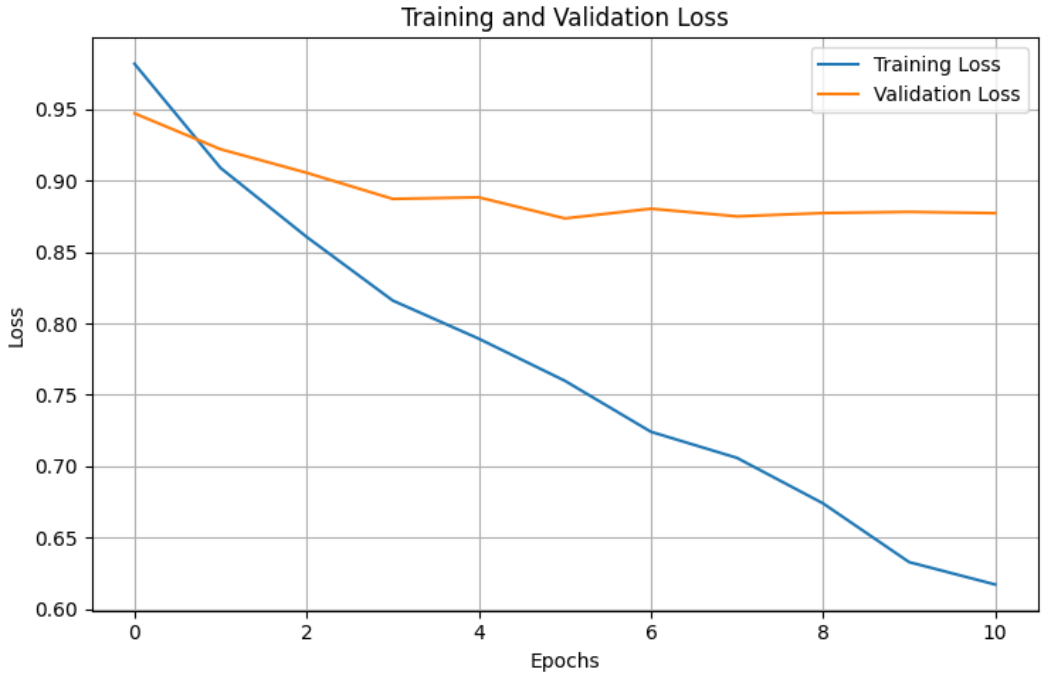
Dataset:

- 3000 zvukových záznamů ve 3 kategoriích
- 128x128x1
- Bez augmentace dat

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 128, 128, 32)	320
batch_normalization_10 (BatchNormalization)	(None, 128, 128, 32)	128
max_pooling2d_8 (MaxPooling2D)	(None, 64, 64, 32)	0
dropout_10 (Dropout)	(None, 64, 64, 32)	0
conv2d_9 (Conv2D)	(None, 64, 64, 64)	18,496
batch_normalization_11 (BatchNormalization)	(None, 64, 64, 64)	256
max_pooling2d_9 (MaxPooling2D)	(None, 32, 32, 64)	0
dropout_11 (Dropout)	(None, 32, 32, 64)	0
conv2d_10 (Conv2D)	(None, 32, 32, 128)	73,856
batch_normalization_12 (BatchNormalization)	(None, 32, 32, 128)	512
max_pooling2d_10 (MaxPooling2D)	(None, 16, 16, 128)	0
dropout_12 (Dropout)	(None, 16, 16, 128)	0
conv2d_11 (Conv2D)	(None, 16, 16, 256)	295,168
batch_normalization_13 (BatchNormalization)	(None, 16, 16, 256)	1,024
max_pooling2d_11 (MaxPooling2D)	(None, 8, 8, 256)	0
dropout_13 (Dropout)	(None, 8, 8, 256)	0
flatten_2 (Flatten)	(None, 16384)	0
dense_4 (Dense)	(None, 256)	4,194,560
batch_normalization_14 (BatchNormalization)	(None, 256)	1,024
dropout_14 (Dropout)	(None, 256)	0
dense_5 (Dense)	(None, 3)	771

Total params: 4,586,115 (17.49 MB)
Trainable params: 4,584,643 (17.49 MB)
Non-trainable params: 1,472 (5.75 KB)





Model má sklony k overfittingu

velké množství parametrů => malý dataset

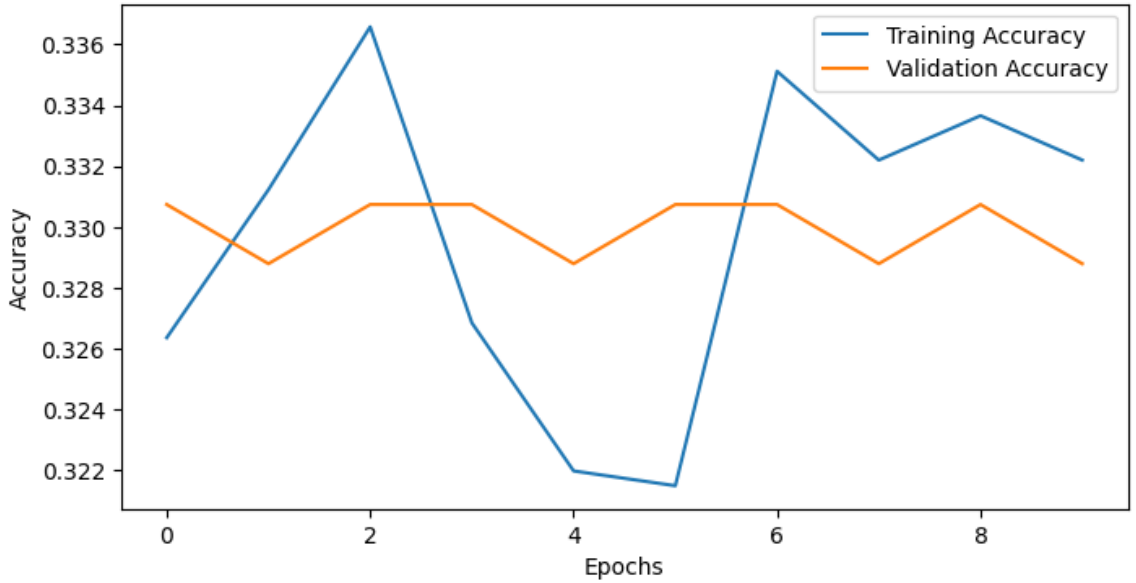
ZJEDNODUŠENÝ MODEL

Snížení počtu parametrů

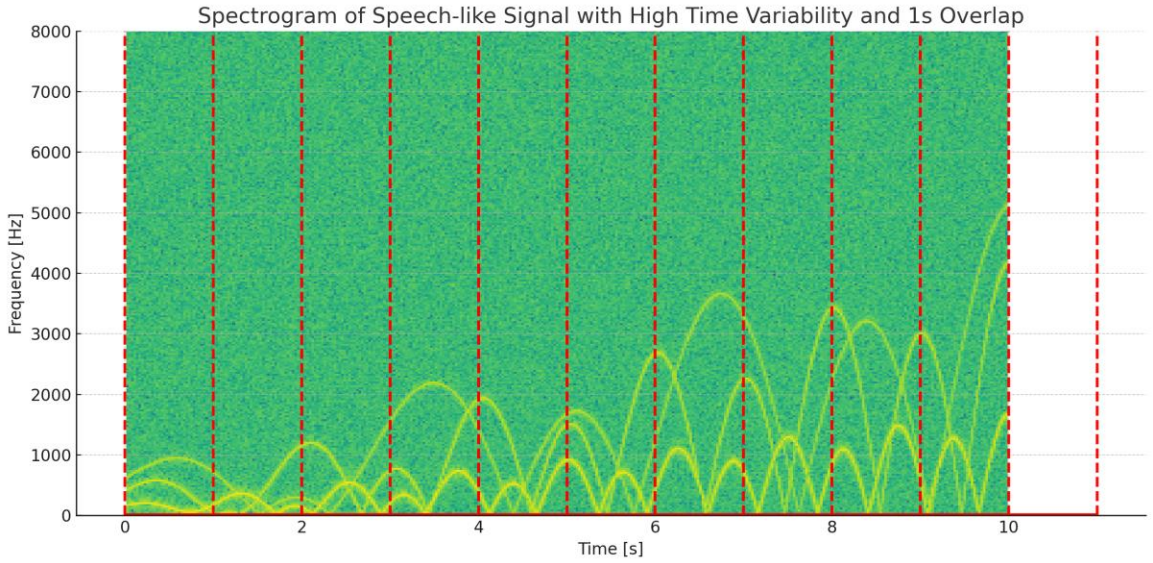
- Zmenšení počtu konvolučních filtrů
- Zmenšení / nahrazení fully connected vrstvy

Ztráta schopnosti učit se

- Složitá data (velké množství detailů)
- nutnost velkého množství parametrů pro interpretaci



AUGMENTACE DAT



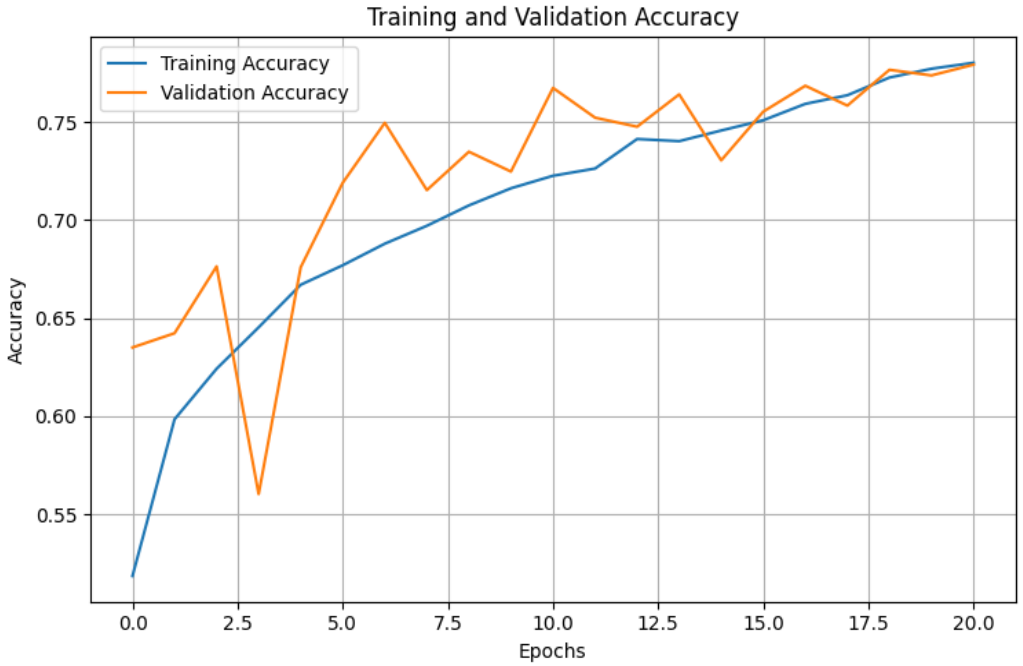
V případě zvukových dat lze množství dat zvýšit

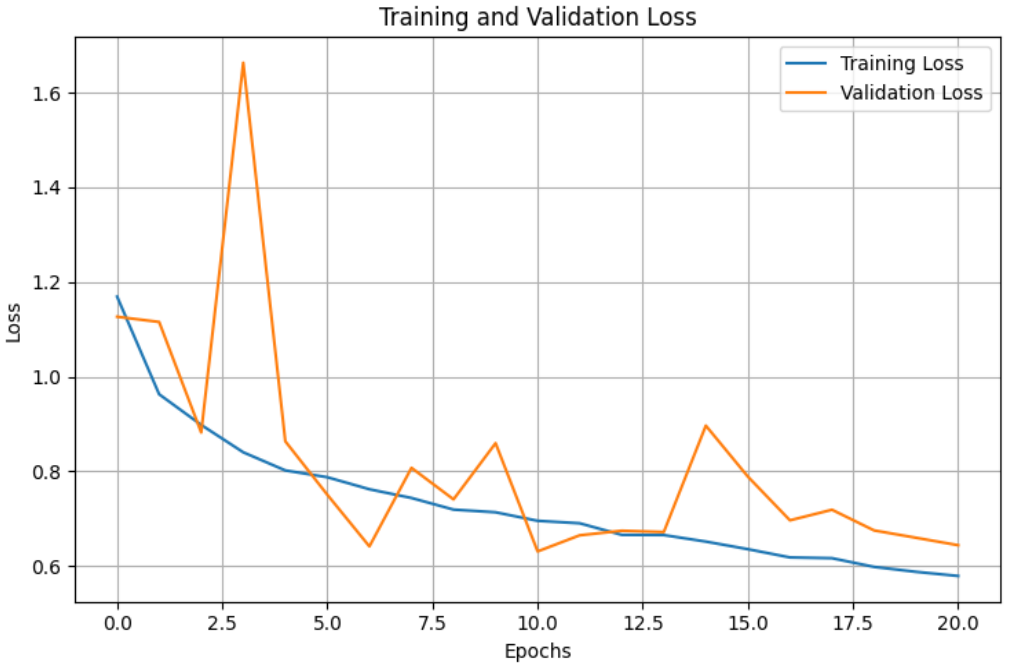
- Šumem
- Rozpadnutím signálu na menší části
- Přidáním překryvu

Rozpad signálu na menší části

- Větší detail při malém vzorkování
- Větší množství dat
- Ztráta časové souslednosti!

VÝSLEDKY PO AUGMENTACI



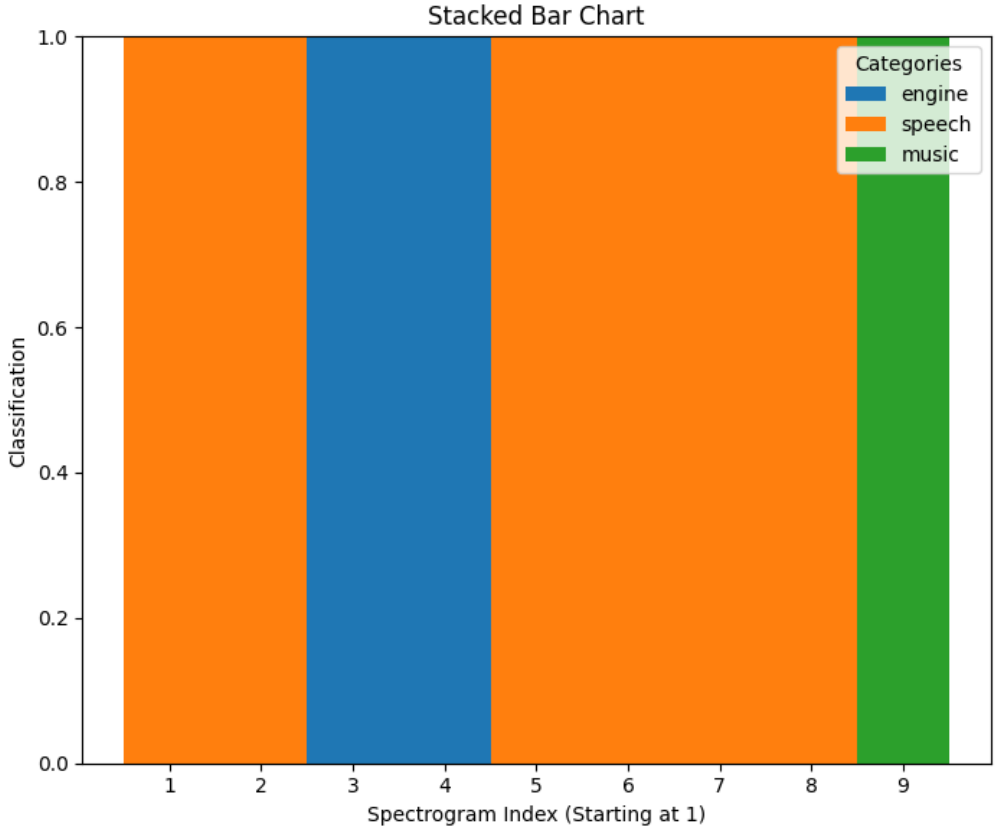


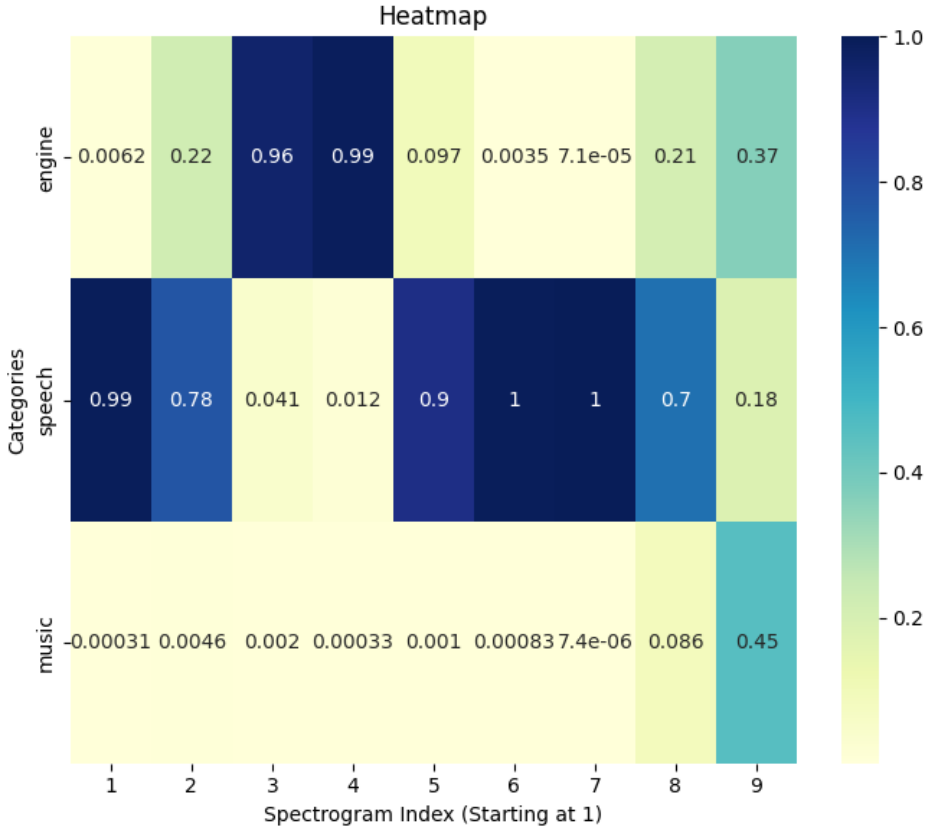
Množství dat: 3000 vzorků => 22000 vzorků

Závěry: Stabilizace učení o 10% vyšší přesnost

VÝSLEDNÁ KLASIFIKACE

ZVUKOVÁ STOPA





DALŠÍ MOŽNOSTI PRO ZLEPŠENÍ

Zvětšení datasetu

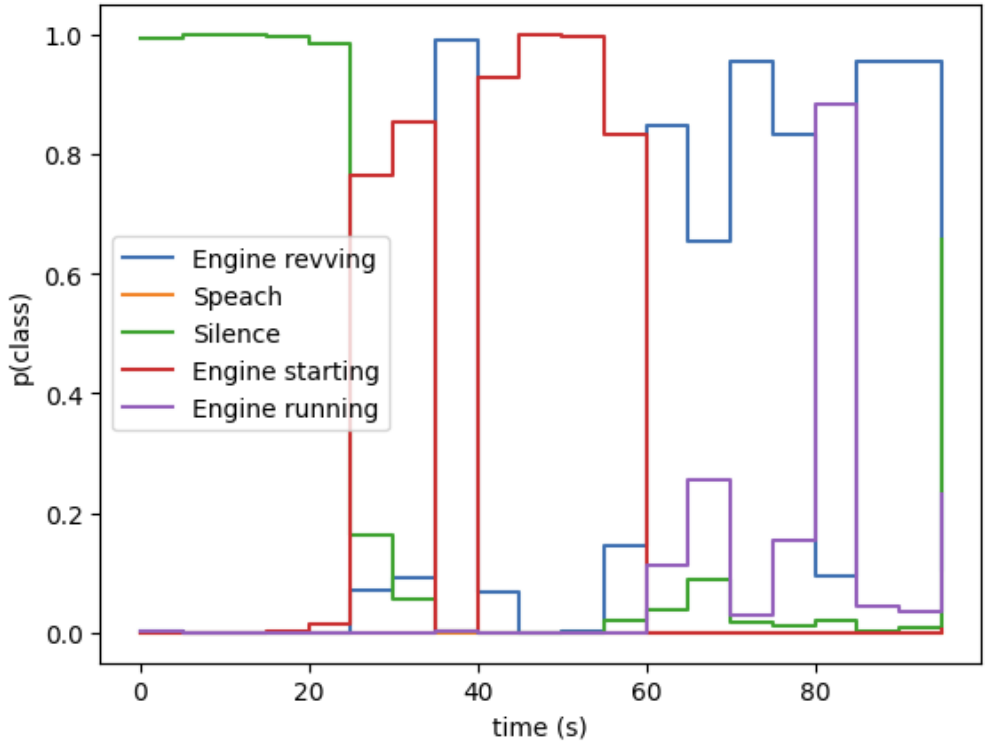
- Přidání dalších tříd
- Očištění dat s více třídami

Vyčištění datasetu

- Manuálně
- Zero-shot
- Clap Model

Transfer learning

- EfficientNet
- ResNet
- VGG



TIPY K IMPLEMENTACI

Vizuální kontrola dat :-)

- Bývá dobré si před samostatným učením ověřit, že data vypadají rozumně
- Jinak možná budete podobně nešťastní jako já :-D

Předpříprava dat v samostatném skriptu

- K učení využívat už předpřipravená data (spektrogramy)
- Výrazné zrychlení oproti výpočtům "on-the-fly"

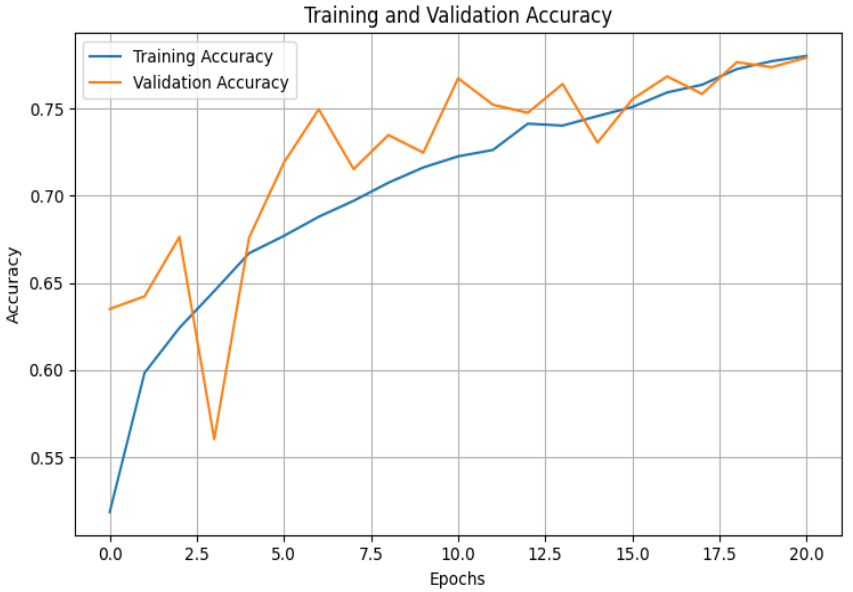
Využívat RAM

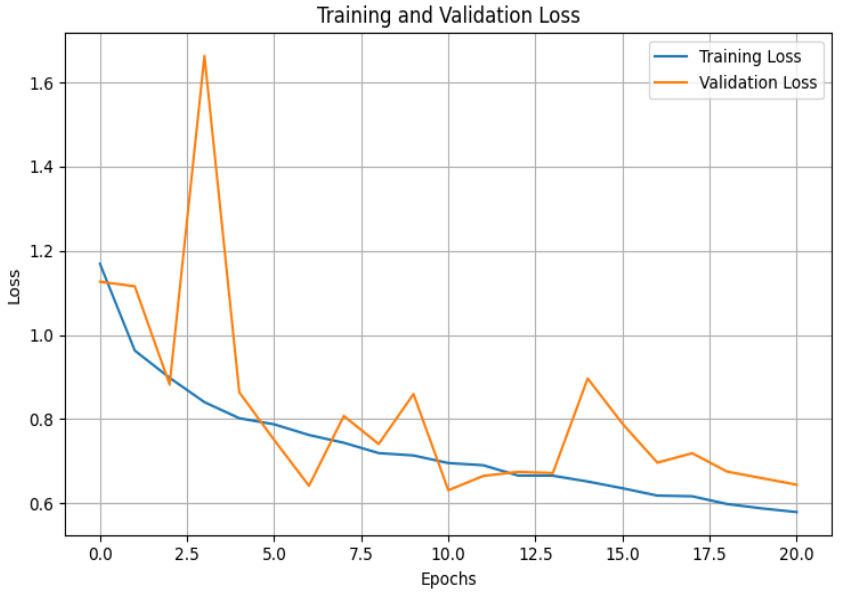
- I/O největší bottleneck
- Načtení dat do paměti zvýšilo výpočetní rychlost na GPU 100x
- 6 GB dat, 17 mil parametrů => 43s na epochu na T4

VLIV ROZLIŠENÍ SPEKTROGRAMU

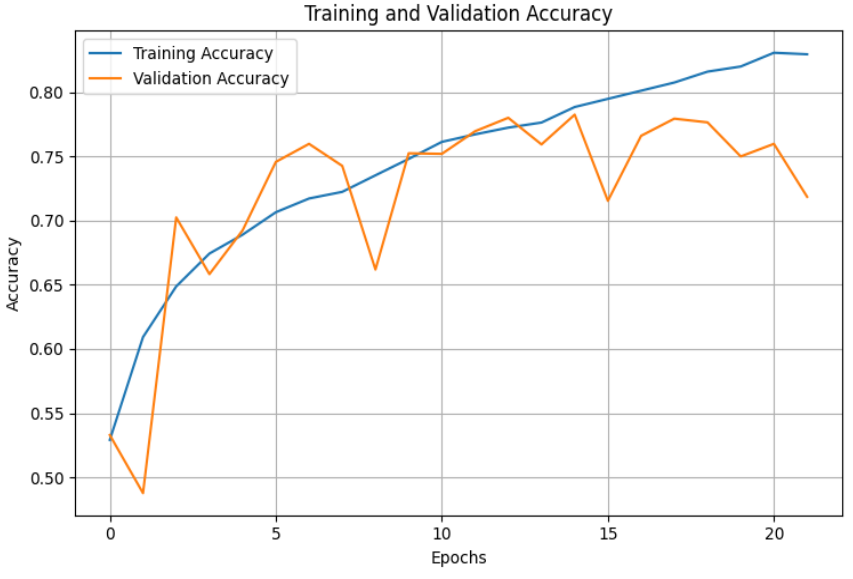
Zanedbatelný až záporný
Spektogram 256 x 256 neposkytuje výrazně vyšší detail

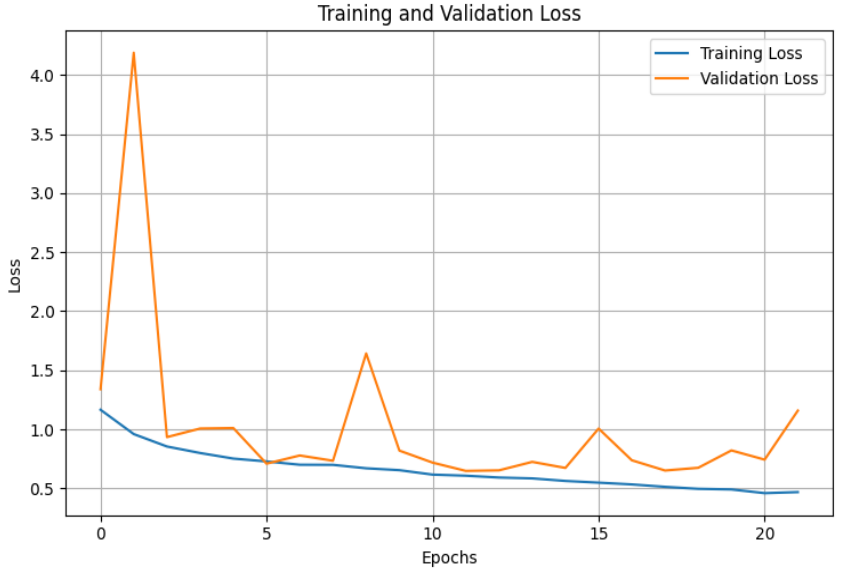
128 x 128





256 x 256





Děkujeme za pozornost!