CNN Architectures for Large-Scale Audio Classification and its Application

Authors: S. Hershey et al.

Presenter: Amber Ma
Facilitators: Abul Hassan Sheikh, Zak Semenov

September 23rd

aisc

Motivation

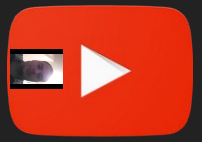
Good quality audio data is hard to come by.

- A 5s audio clip takes at least 5s for a human to recognize, where a picture of cat can be instantly recognized.
- Sound recognition is subjective and sometimes hard even for human.
- Scraping images is easy, but finding audios are hard.
- Specialized audio data (eg phone calls) don't cover enough variety of labels.

Motivation

Motivation






Motivation

Have to resort to

Extremely diverse data set and train from scratch

Or

Transfer learning where the model is trained on extremely abundant audios data set.

⇒ YouTube, which emcompasses almost all aspects of human activity.

Datasets -- YouTube-100M -- Overview

Google's internal data set.

Weakly labeled.

Consists of 100 million YouTube videos:

70M training videos,

20M videos that for validation,

10M evaluation videos.

Videos average 4.6 minutes each for a total of 5.4M training hours.

Abstract

Questions addressed using Google's YouTube-100M dataset:

- How popular Deep Neural Network such as AlexNet, VGG, Inception, and ResNet architectures compare on video soundtrack classification;
- How performance varies with different training set and label vocabulary sizes;
- How trained models can also be useful for Acoustic Event Detection (AED)

Question

Even if I had the 2 million raw audios from the public AudioSet dataset, is it feasible to train a model using those such that it performs as well as Google's VGG embedding model?

Datasets -- YouTube-100M -- Labelling

Each video is weakly labeled with 1 or more topic identifiers

Average of around 5 labels

Audio labels a combination of metadata (title, description, comments, etc.), context, and image model predictions 

The labels apply to the entire video and range from very generic (e.g. “Song”) to very specific (e.g. “Cormorant”).

not 100% accurate and of the 30K labels, some are clearly acoustically relevant (“Trumpet”) and others are less so (“Web Page”)

Datasets -- YouTube-100M -- Preprocessing

The audio is divided into non-overlapping ~1s (960 ms frames to be exact) (approximately 20 billion examples from the 70M videos)

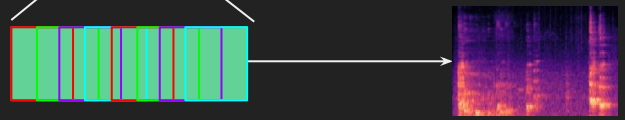
Decomposed with a short-time Fourier transform applying 25 ms windows every 10 ms

Log-mel spectrogram patches of 96 × 64 bins that form the input to all classifiers

Mini-batches of 128 input examples

Video cut into frames:

STFT on a frame:



Some Jargons

Fourier Transform: Decomposing a function as a sum of sin's and cos's at all frequencies

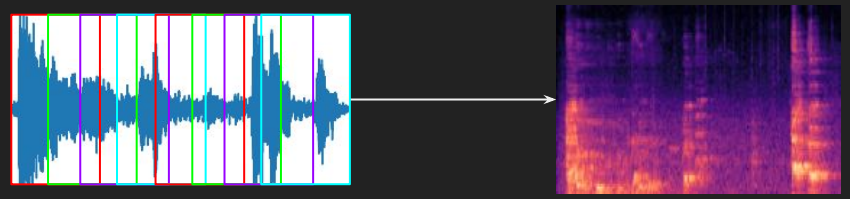
FFT (Fast Fourier Transform): Efficient algorithm for discrete Fourier Transform

STFT (Short time Fourier Transform): Doing FFT on short sliding windows of longer signals

Spectrogram: output "image" of STFT, where a pixel's value/colour signifies the intensity of that frequency within that short time window.

Mel-log Scale: Scaling of spectrogram frequencies so that it makes most sense for human ears

STFT:



Great visualization on Fourier Transform: https://youtu.be/spUNpyF58BY

Experiments -- Training (FC,AlexNet,VGG,ResNet, Inception)

TensorFlow, multiple GPUs, Adam optimizer.

Batch normalization was applied after all convolutional layers.

All models used a final sigmoid layer rather than a softmax layer since each example can have multiple labels.

Loss function: Cross-entropy

No dropout, weight decay, or other common regularization techniques.

For the models trained on 7M or more examples, no evidence of overfitting.

Monitoring: mean Average Precision (mAP) over a validation subset.

Experiments -- Evaluation

Data:
Three balanced evaluation sets, each with roughly 33 examples per class:
1M videos for the 30K labels
100K videos for the 3087 (henceforth 3K) most frequent labels
12K videos for the 400 most frequent labels

Metrics:
AUC (area under the Receiver Operating Characteristic (ROC) curve)
D-prime
Mean Average Precision (mAP)

Experiments -- Fully Connected CNN

Baseline model

Trained and evaluated using only the 10% most frequent labels of the original 30K (i.e, 3K labels).

RELU activations

Best performing model :

Had N = 3 layers, M = 1000 units,
Learning rate of 3×10^{-5} ,
10 GPUs and 5 parameter servers.
Approximately 11.2M weights and 11.2M multiplies.

Experiments -- AlexNet

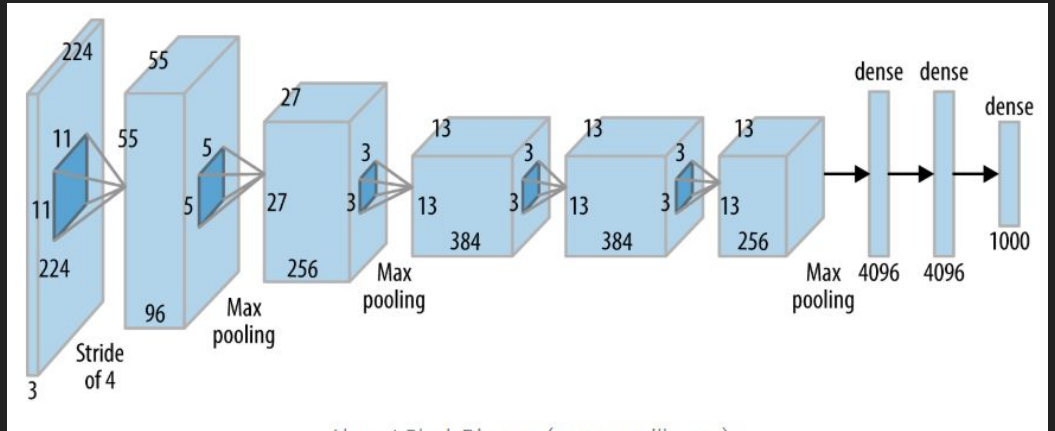
The original AlexNet architectures: a 224 × 224 × 3 input with an initial 11 × 11 convolutional layer with a stride of 4, but our inputs are 96 × 64 → stride of 2 × 1

Batch normalization after each convolutional layer instead of local response normalization (LRN) replace the final 1000 unit layer with a 3087 unit layer.

37.3M weights and 767M multiplies.

20 GPUs and 10 parameter servers.

Parameter server is a distributed training framework for large ML models where global parameters such as NN's weights are shared across servers but the training computations are distributed.



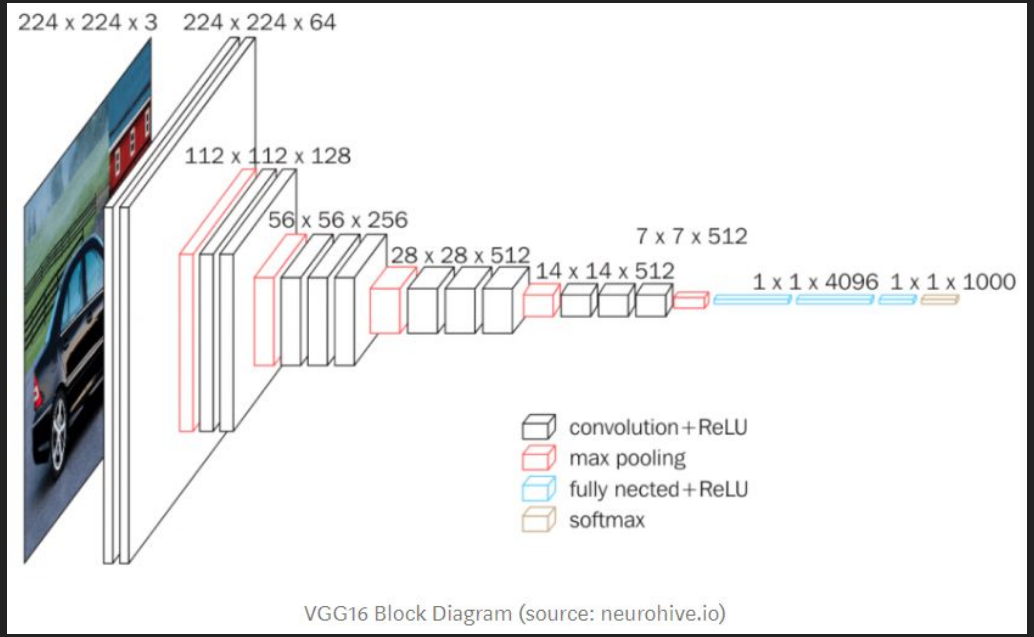
Alexnet Block Diagram (source:oreilly.com)

Experiments -- VGG

VGG marks the standardization of Convolution NNs after AlexNet where the network got deeper (from less than 10 layers to 11 - 19 conv layers)

more structured in terms of convolution kernel size, max pooling schemes, etc. It became a standard to double channels at each convolution layer.

the Local Response Normalization (LRN) -- normalizes local patches in the original paper



VGG16 Block Diagram (source: neurohive.io)

Experiments -- VGG (VGG 19)

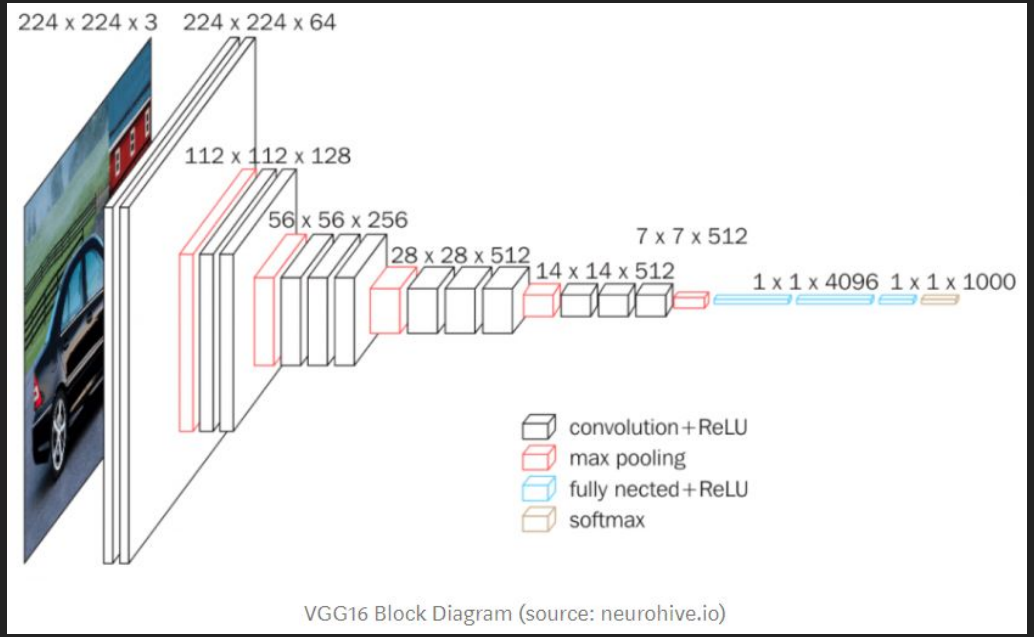
Changes to the original final layer size

Batch normalization instead of LRN (Local Response Normalization)

Audio variant uses 62M weights and 2.4B multiplies: a 57% reduction in weights and 90% reduction in multiplies

Trained with 10 GPUs and 5 parameter servers.

No change to convolution stride.



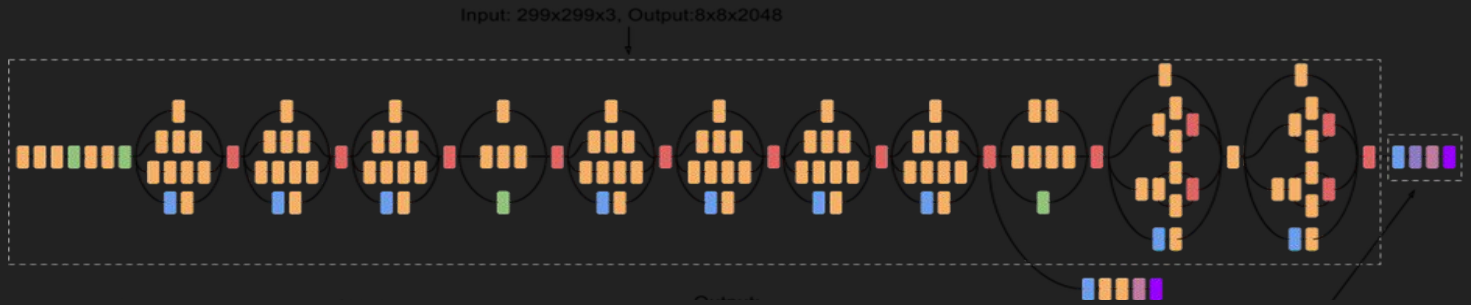
VGG16 Block Diagram (source: neurohive.io)

This is NOT the released VGGish (VGG11) model

Experiments -- Inception V3

Also known as GoogleNet consists of total 22 layers and was the winning model of 2014 image net challenge.

Inception modules are the fundamental block of InceptionNets. The key idea of inception module is to design good local network topology (network within a network)



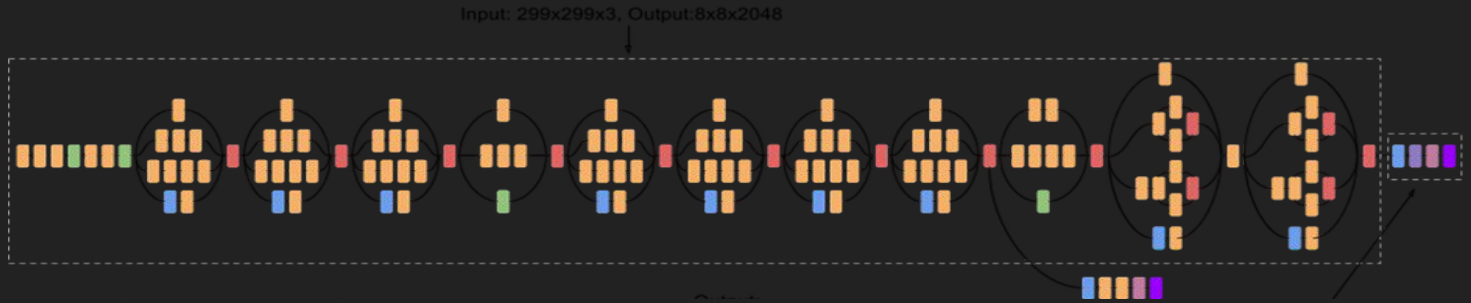
Experiments -- Inception V3

Removing the first four layers of the stem

Changed the Average Pool size to 10 × 6 to reflect the change in activations.

The audio variant has 28M weights and 4.7B multiplies. Similar in size as the original.

Trained with 40 GPUs and 20 parameter servers.

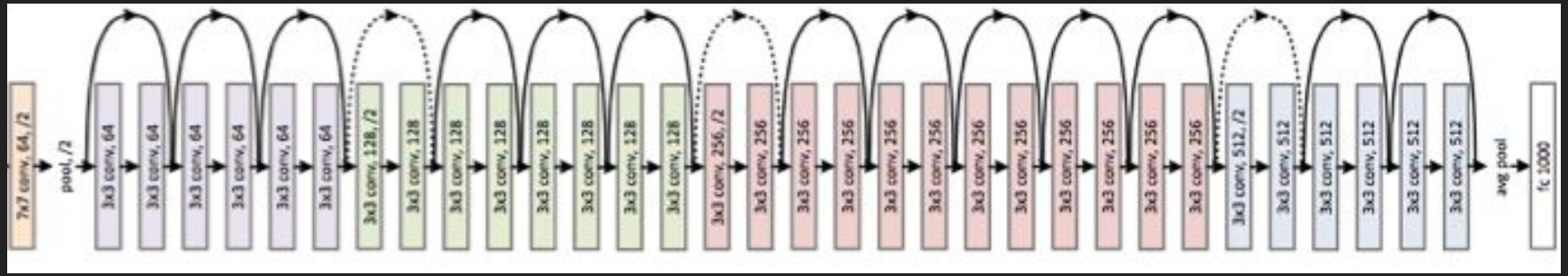


Experiments -- ResNet-50

All the previous models used deep neural networks in which they stacked many convolution layers one after the other. It was learnt that deeper networks are performing better. However:

- Network becomes difficult to optimize
- Vanishing / Exploding Gradients
- Degradation Problem (accuracy first saturates and then degrades)

ResNet introduces the concept of “skip-connections” where shortcuts are created to connect ith layer to (i+2)th layer if a identity block has two layers.



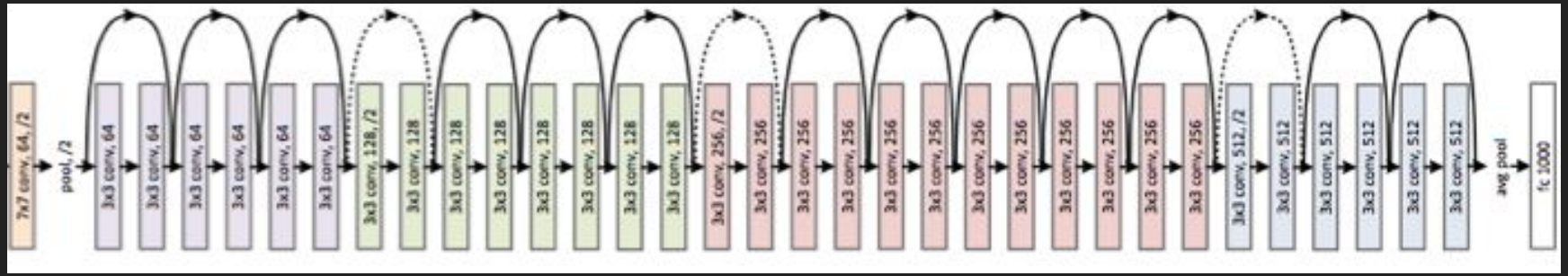
Experiments -- ResNet-50

Removing the stride of 2 from the first 7×7 convolution so that the number of activations was not too different in the audio version.

Changed the Average Pool size to 6 × 4 to reflect the change in activations.

The audio variant has 30M weights and 1.9B multiplies: size ⬆ but multiplies ⬇

Trained with 20 GPUs and 10 parameter servers.

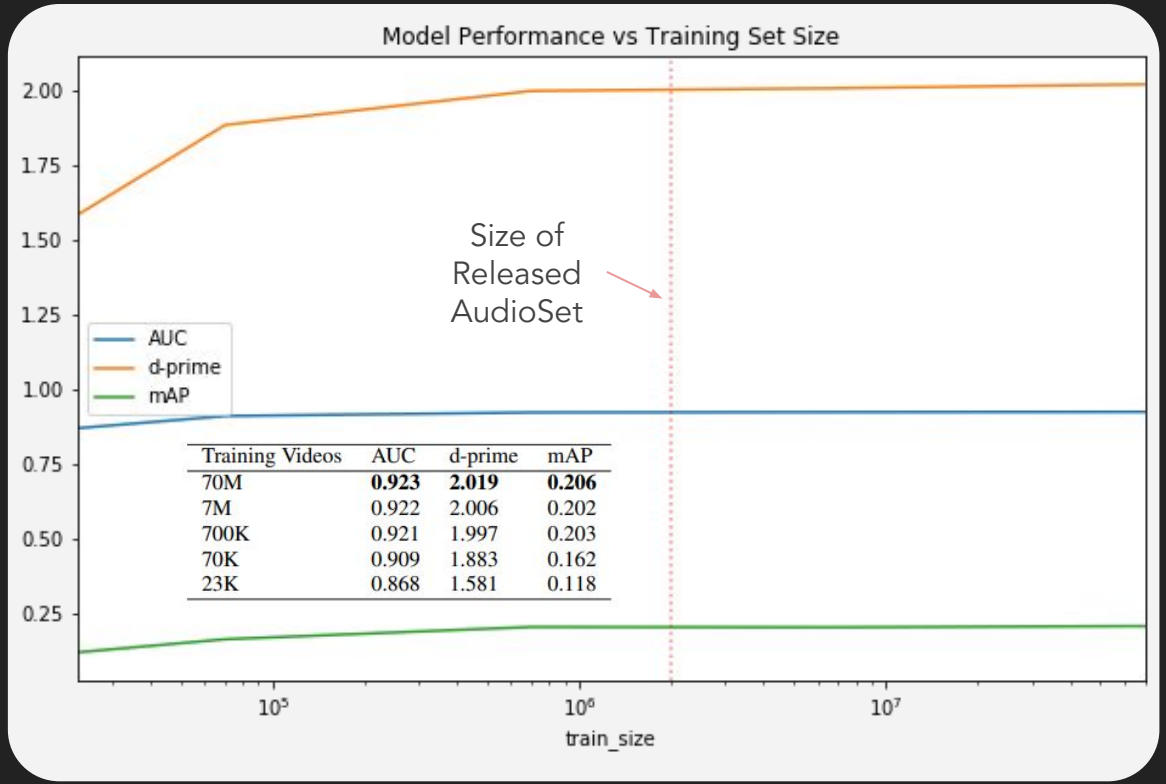


Results

Training size is important → Diversity is good

But "the validation plots (not included) showed that they **likely suffered from overfitting**.

Regularization techniques (or data augmentation) might have boosted the numbers on these smaller training sets."



Training Videos	AUC	d-prime	mAP
70M	**0.923**	**2.019**	**0.206**
7M	0.922	2.006	0.202
700K	0.921	1.997	0.203
70K	0.909	1.883	0.162
23K	0.868	1.581	0.118

* Using ResNet

Results

NN architecture is somewhat important, but not as important as training time.

Table 2: Comparison of performance of several DNN architectures trained on 70M videos, each tagged with labels from a set of 3K. The last row contains results for a model that was trained much longer than the others, with a reduction in learning rate after 13 million steps.

Architectures	Steps	Time	AUC	d-prime	mAP
Fully Connected	5M	35h	0.851	1.471	0.058
AlexNet	5M	82h	0.894	1.764	0.115
VGG	5M	184h	0.911	1.909	0.161
Inception V3	5M	137h	**0.918**	**1.969**	0.181
ResNet-50	5M	119h	0.916	1.952	**0.182**
ResNet-50	17M	356h	**0.926**	**2.041**	**0.212**

Results -- Label Set Size

"**weak support** to the notion that training with a broader set of categories can help to regularize even the 400 class subset."

Table 3: Results of varying label set size, evaluated over 400 labels. All models are variants of ResNet-50 trained on 70M videos. The bottleneck, if present, is 128 dimensions.

Bneck	Labels	AUC	d-prime	mAP
no	30K	—	—	—
no	3K	**0.930**	**2.087**	**0.381**
no	400	0.928	2.067	0.376
yes	30K	0.925	2.035	0.369
yes	3K	0.919	1.982	0.347
yes	400	0.924	2.026	0.365

What to Do If I Want to Build an AED Model?

Assuming I have scraped 2M labeled audio clips from AudioSet, is this enough to train a good AED model from scratch?

-- Not without spending lots of effort in regularization, data augmentation, time, and computing resources.

⇒ Transfer learning it is.

What's Made Public by Google

AudioSet (~2M 10s clips):
- Ontology (i.e. how sounds are classified)
- Data in embedding format (128 features per second)
- Video IDs and labels

VGGish Embedding Model (**!=** the VGG model in the paper)
- Configuration A (11 layers) instead of E (19 layers)
- BatchNorm not used
- The last group of convolutional and maxpool layers are dropped.

Why water down VGG? Why not ResNet (which is much smaller than VGG)**?**

AED with the AudioSet Data Set

Break

Caption4Sound

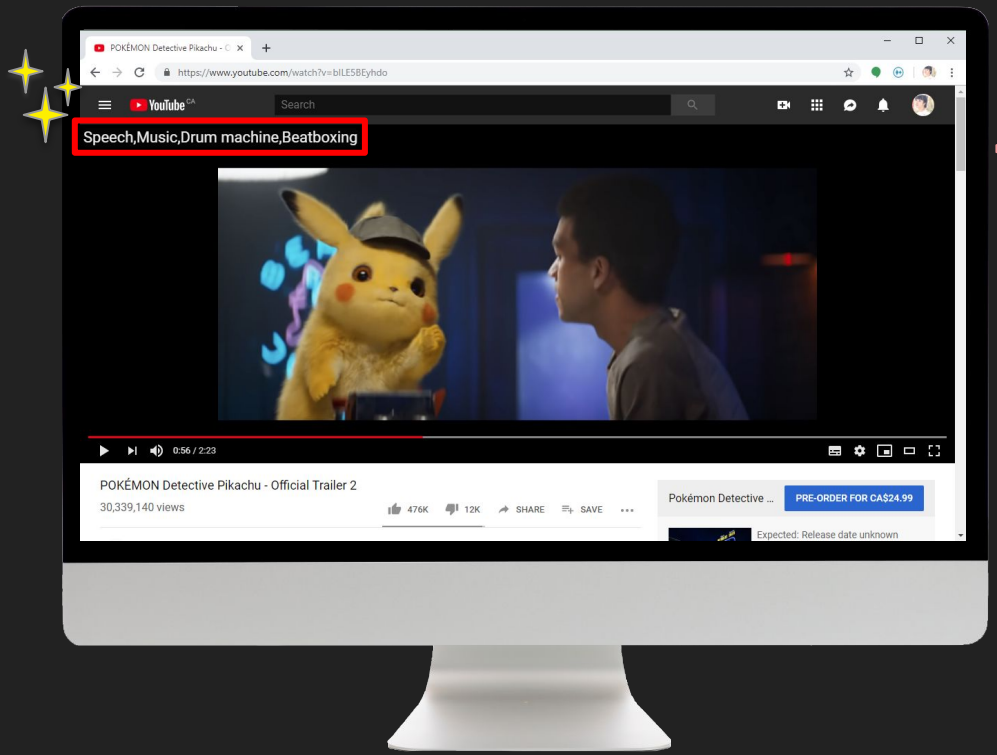
Closed Captioning for Video Sound Description

Motivation

- 1.3 billion videos
- Provides auto-captioning only for speech
- Understanding audio context helps hearing impaired people to fully enjoy videos

Pipeline for Caption4Sound

Chrome Extension



URL

Captions

Backend

Amazon EC2

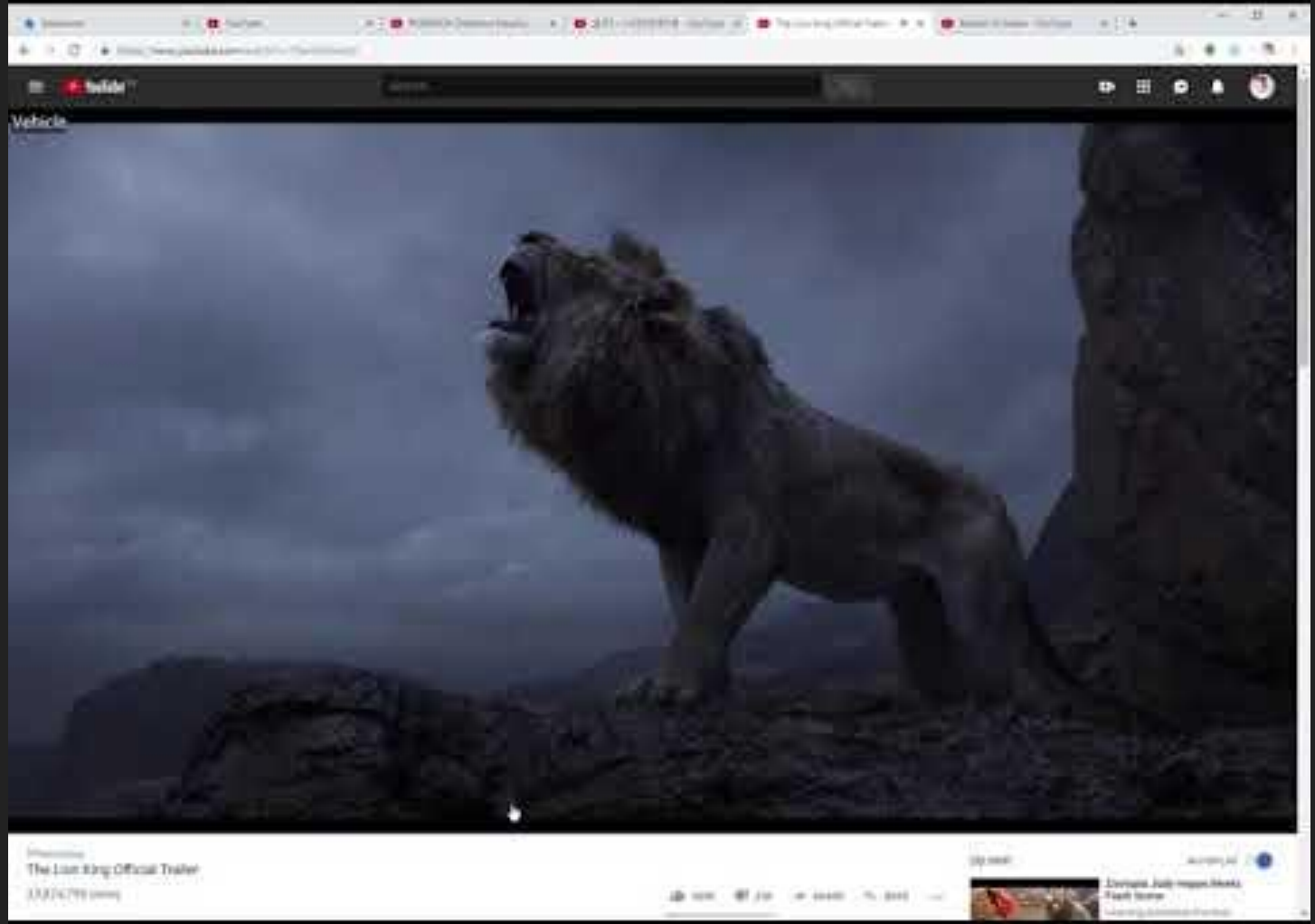
FastAPI

TensorFlow

uvicorn

Video Demo

Caption in top-left corner



Installation

Branch: master ▾ caption4sounds / chrome_extension / Create new file | Upload files | Find file | History

hellokikicat Update README.md — Latest commit 0c23405 4 days ago

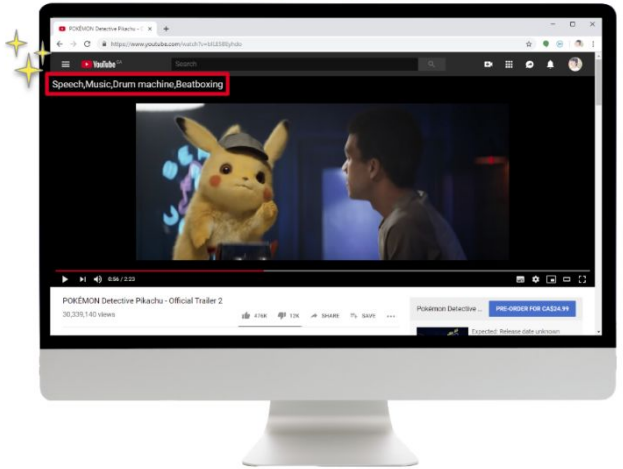
..		
caption4sounds	reblock window 5 hop 2	6 days ago
README.md	Update README.md	4 days ago

README.md

Caption4sound Chrome Extension

The front-end of Caption4sounds as a Chrome extension.

Chrome Extension

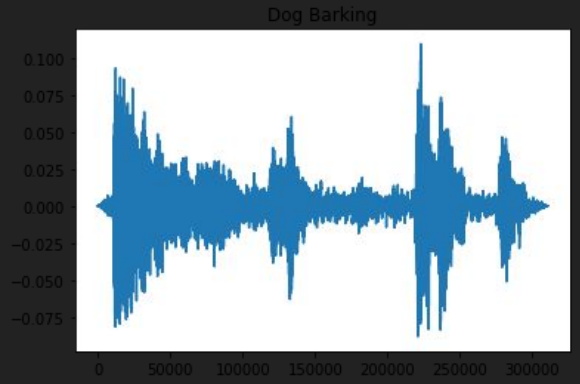


Installation

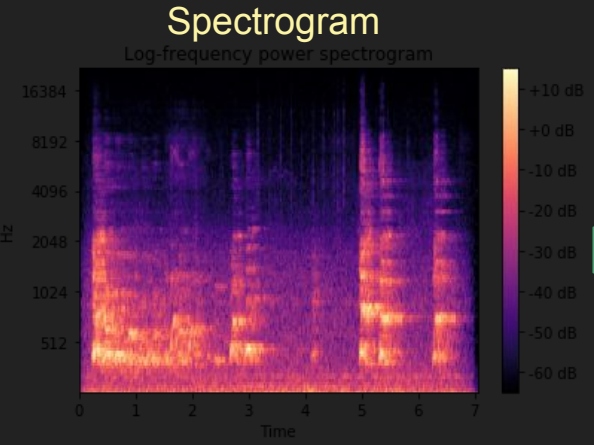
1. Download folder `caption4sounds` in this folder
2. Go to "chrome://extensions/" in your browser
3. Turn on "Developer Mode"
4. Click on "Load Unpacked" and select the downloaded "caption4sounds" folder
5. To use, go to a YouTube video page, reload, and push the extension button besides the address bar
6. Caption will appear once processing is finished on the server side

The Idea

Sound Event Recognition ➔ Image Recognition



Short Time Fourier Transform



CNN

.....
“Speech”
“Music”
“Vehicle”
“Key Jangling”
“Cat Purring”
“Fire Alarm”
“Rain”
.....

Youtube Audio clip

2D histogram of sound intensity

Larger Data Sets from Google

Public Domain

YouTube-100M

- Internal use only, BIG!!
- ~100M videos with video-level labels (~5M hours, 600 years)
- ~30K weak labels (not all obviously acoustically relevant)
- ~3 labels per videos

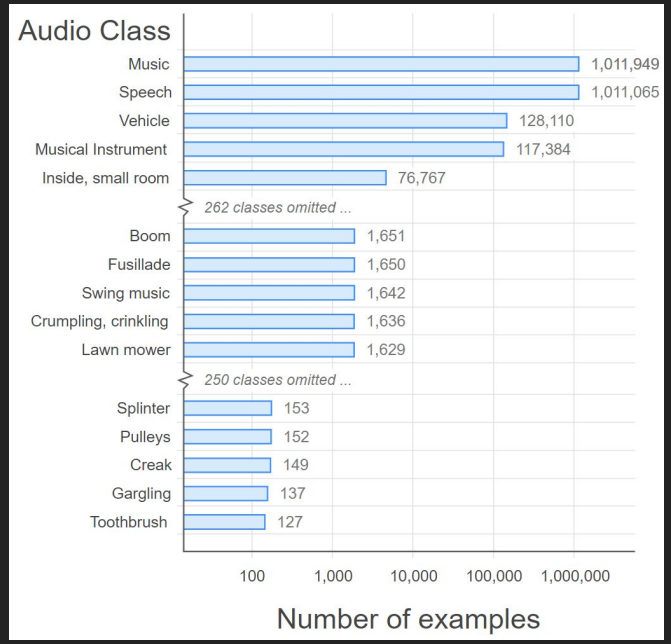
YouTube-8M

"ImageNet for videos"

AudioSet-2M

With strong audio labels

AudioSet

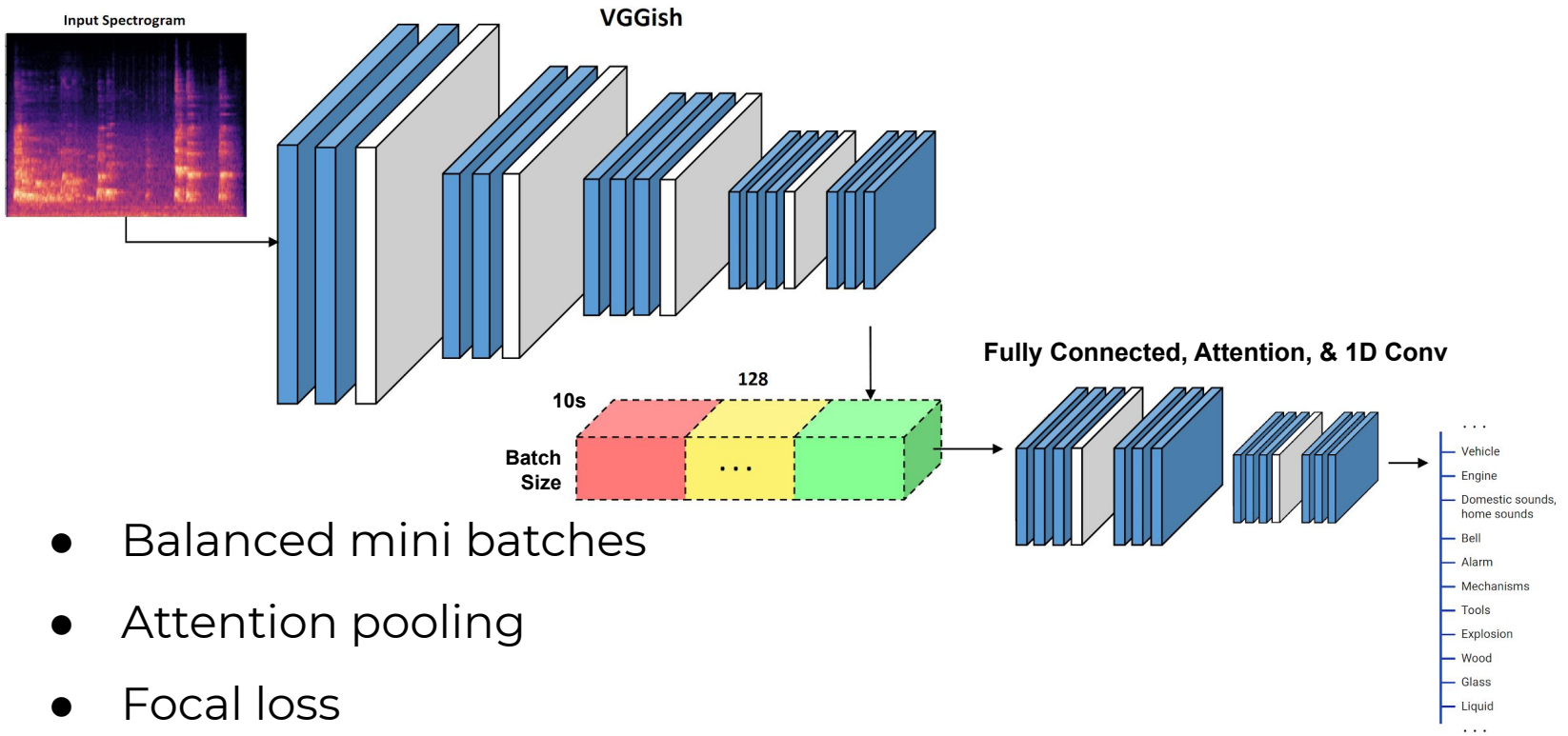


Histogram of label counts over the entire 1,789,621 segment dataset.

Embedding format, a.k.a. “VGGish”

- 2 million 10-second sound clips
- 527 audio event classes
- **Low quality labels**
  - Positive label implies event occurs in 10-sec segment, but we do not know extent
- **Extreme Class Imbalance**
  - Results in poor score calibration across classes
  - High-prior classes like speech and music are always strongest detections

Final CNN Model with Audiosets

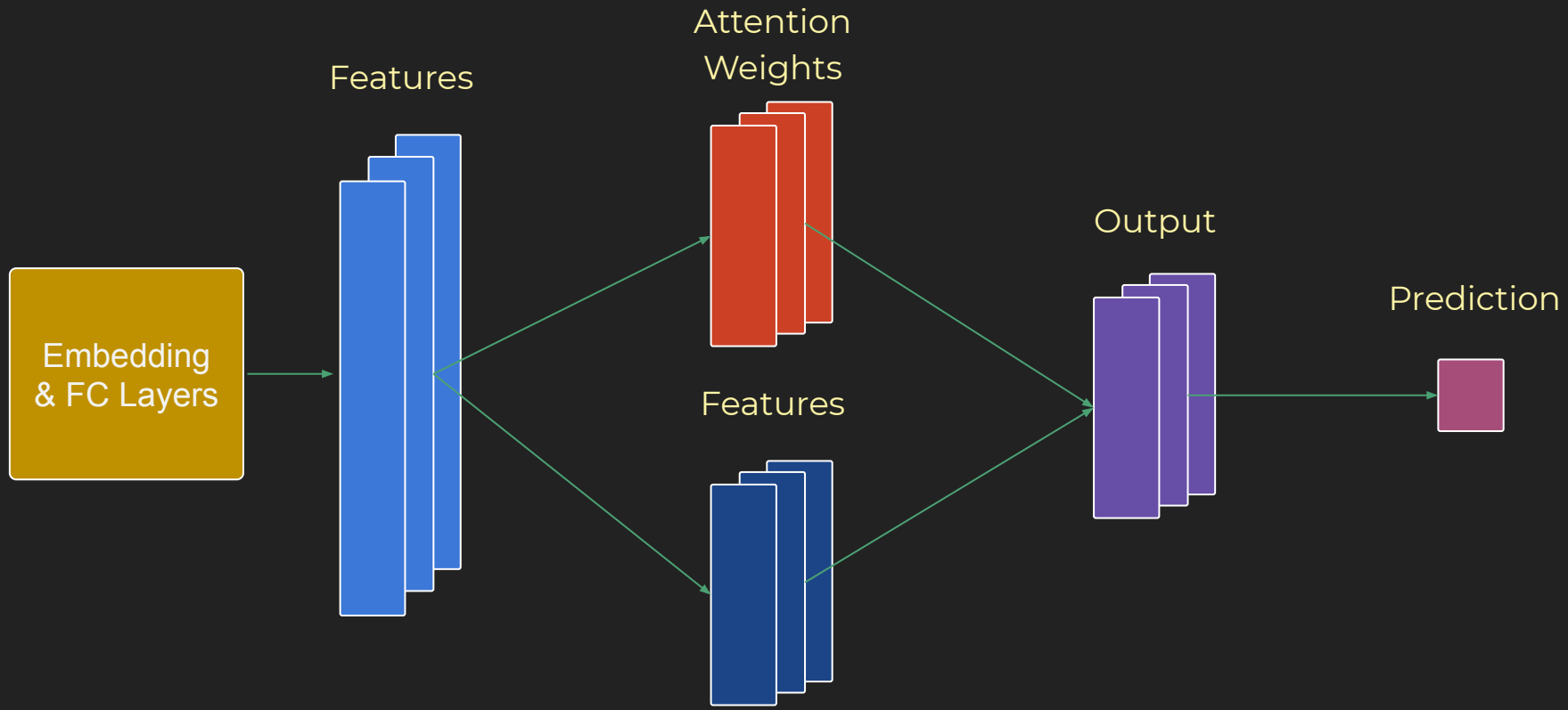


- Balanced mini batches
- Attention pooling
- Focal loss

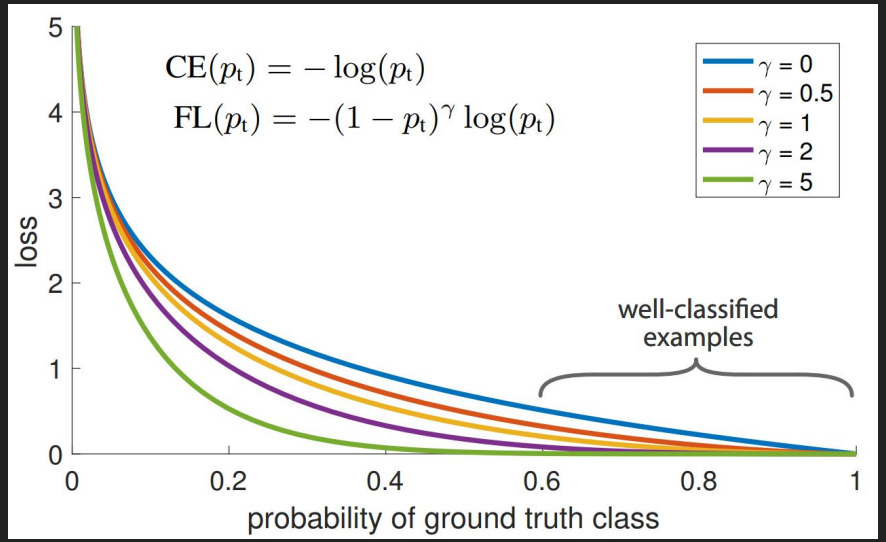
Balanced Mini-Batches

- Oversample “small” classes
- Try to include at least 1 sample from each class in each mini-batch (80% realistically)
- Did NOT undersample “huge” classes

(Second Order) Attention Pooling



Focal Loss



- Shifts training effort towards classes that are difficult to classify:
  - Easy classes → lower gradient
  - Hard classes → higher gradient
- Only helps imbalance between easy and difficult classes
- Does **NOT** help imbalance among classes

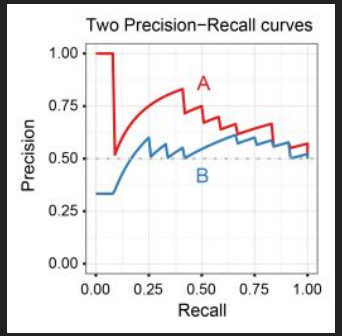
Model Validation

Model	Mean Average Precision	mAUC	d-prime
Google's Baseline [1]	0.314	0.959	2.452
Caption4Sound with attention pooling layers	0.343	0.962	2.516

[1]: CNN Architectures for Large-Scale Audio Classification
https://ai.google/research/pubs/pub45611

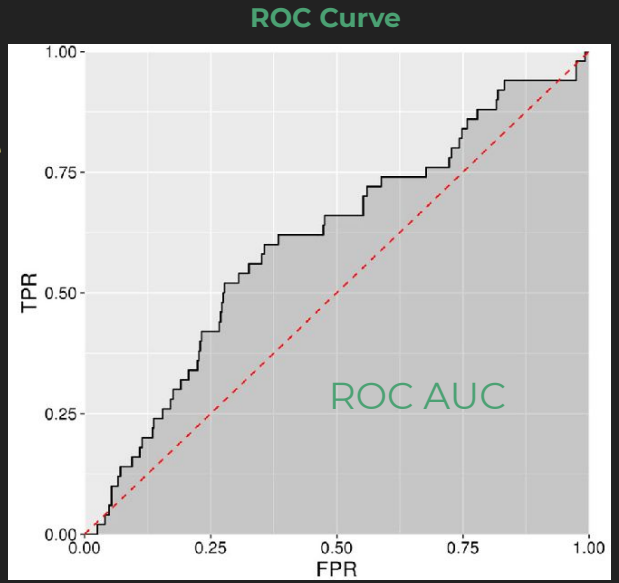
Metrics

- Recall
  - Recall = Correct Prediction / All Positive Ground Truth
  - e.g. of all clips labeled as “dog barking”, how many does the model predicts as “dog barking”?
- Precision
  - Precision = Correct Prediction / All Positive Prediction
  - e.g. of all clips which the model predicts as “dog barking”, how many is actually “dog barking”?
- Precision-Recall Curve
  - Vary classification threshold from 0 to 1
  - Plot Precision and Recall for each threshold
  - Describes trade off between precision and recall

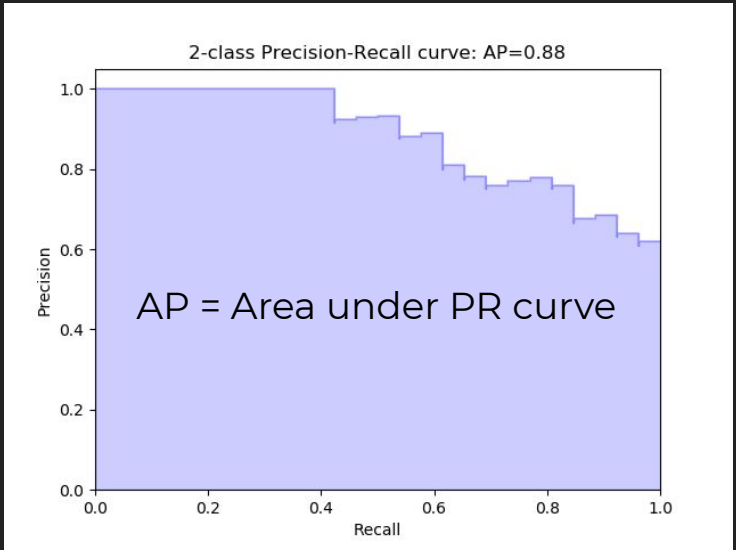


Metrics

- ROC Curve
  - Plots Recall vs False Positive Rates
  - Describes tradeoff between true positives and false positive
- ROC AUC
  - Area under the ROC curve
  - Measures the discrimination power of the model
- mAUC
  - Mean of AUCs across all classes
- d-prime
  - d-prime = sqrt(2) * Z(AUC) where Z() is the percentile function of the standard Normal Distribution

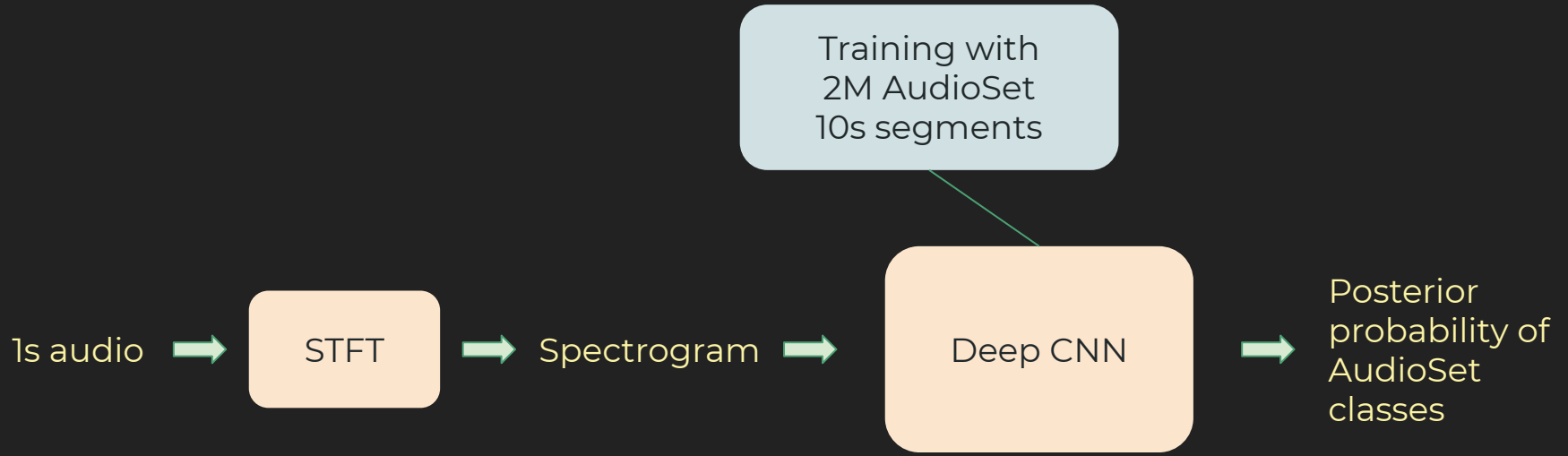


Metrics -- Mean Average Precision (mAP)

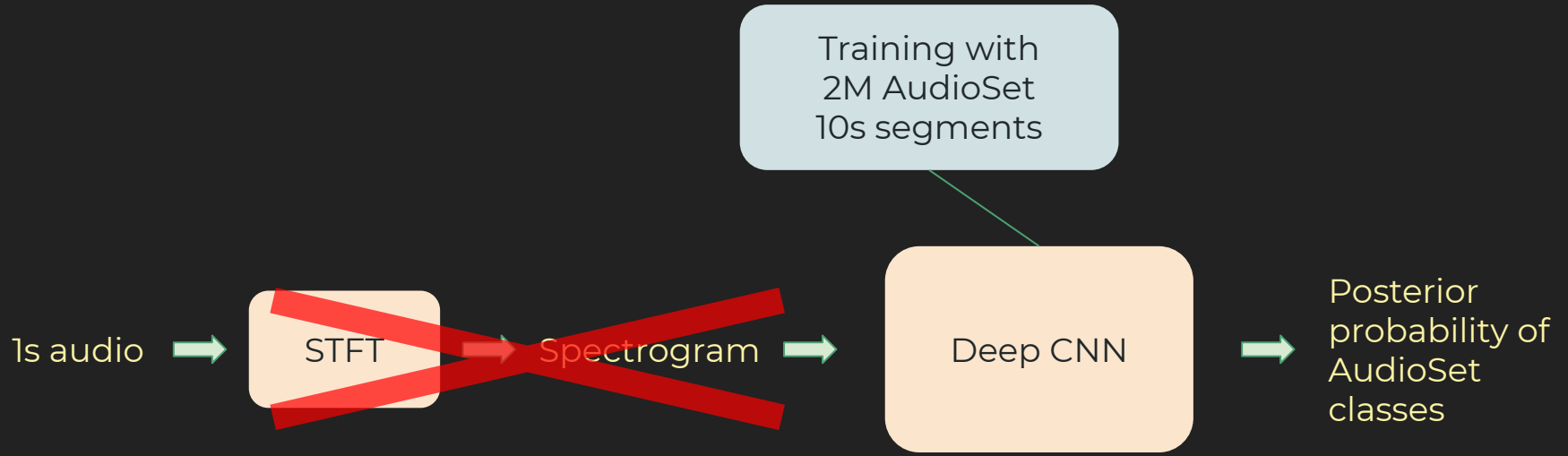


- Average Precision
  - Area under the Precision-Recall curve
  - “Average precision across all recalls as threshold increases”
- Mean Average Precision (mAP)
  - For multi-class classification
  - Calculate AP for each class
  - mAP = Mean of APs across all classes
- mAP is a better measure than ROC AUC for imbalanced data sets

Training from Scratch?

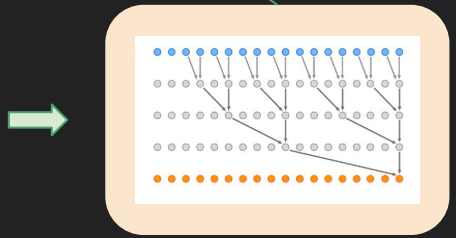


Training from Scratch?



With WaveNet?

Training with
2M AudioSet
10s segments



Posterior probability of AudioSet classes

Hoping for:

- Speed up
- Reduced size
- Simpler pipeline

Key takeaways

Image recognition CNNs are capable of excellent results on audio classification.

Training on more diverse/noisy dataset can improve performance.

Training on larger label set vocabularies can improve performance.

Longer training time helps models continue to improve.

Embedding model capable of excellent results for AED on the Audio Set dataset.

Discussion points

1. Do you think Google should release the full VGG or even the ResNet Model to the public instead of the “watered-down” version?
2. How to speed up the embedding VGGish model so that it can be used on CPU or edge computing devices (surrogate model?)
3. How well can STFT extend to other applications of time series modeling (e.g. financial market signals, sensor signals).
4. Other approaches for solving the problem of short time events when data is only available in 10s clips.

Discussion points

1. Do you think Google should release the full VGG or even the ResNet Model to the public instead of the “watered-down” version?
2. How to speed up the embedding VGGish model so that it can be used on CPU or edge computing devices (surrogate model?)
3. How well can STFT extend to other applications of time series modeling (e.g. financial market signals, sensor signals).
4. Other approaches for solving the problem of short time events when data is only available in 10s clips.

Discussion points (continued)

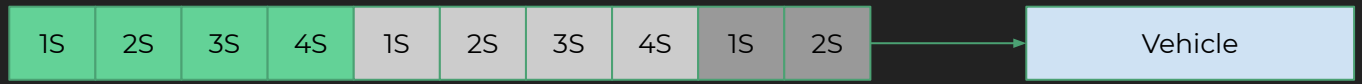
5. With a raise of hands, who thought that this study did a good job representing the potential of audio classification?
6. Are any of you surprised by the performance of one of the models ability to classify audio? Why?
7. Can any of you think of other sources of audio data that could be used for classification other than youtube? what advantages do you think that data source would have over youtube?

Short sound event

The classifier is trained on 10s of embeddings:

But want to predict on shorter time intervals for precise timing of events

- ✗ Option 1: train on shorter intervals → but training label doesn't specify timing
- ✗ Option 2: always predict on 10s → timing of captions is extremely off
- Option 3: fill the 10s window by repeating → fairly good results



Amber

B.Sc. in Finance & Information Science

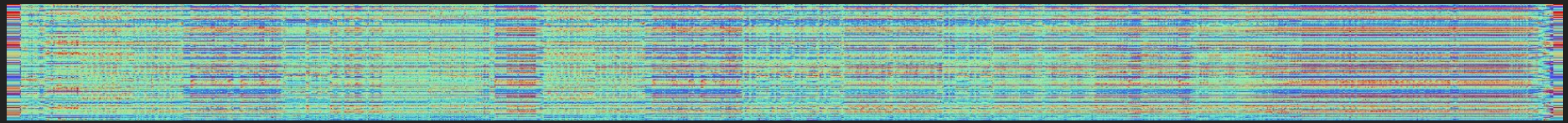
M.Sc. in Information Science

Research Assistant on Transportation Optimization

Machine Learning Specialist

Artificial Intelligence Fellow

Personal Project: Music Interpretation



Swan Lake Suite, Op. 20a, TH 219: Act I: Waltz -- Tchaikovsky