Audio Classification by RestNet50

周呈陽 F14081046 / 馬世常 F14086151 / 蘇　晃 F14086224

OUTLINE

I. **Introduction**

II. **Training Process**

  - Dataset Choosing
  - Model & Train

I. **Deep Learning Structure**

  - Architecture
  - Loss Function

I. **Experiment**

II. **Future Work**

OUTLINE

I. **Introduction**

II. **Training Process**

- Dataset Choosing
- Model & Train

I. **Deep Learning Structure**

- Architecture
- Loss Function

I. **Experiment**

II. **Future Work**

INTRODUCTION

I. Motivation

- Rencently, the audio speech recognition is widely utilized in many fields.
- Despite there being an improvement in the standard CNN architectures, there has been no work that has used these pre-trained ImageNet models for audio tasks.
- People have ignored a strong ImageNet pretrained model baseline to compare the customized models against.

I. Objective

- By simple pre-trained ImageNet models with a single set of input features for audio we can achieve good results

I. Contribution

- Successfully using the pretrained models to achieve state-of-the-art results
- For CNNs between different image tasks seem to hold for transfer learning between images and spectrograms.
- Using Integrated Gradients to understand CNN learns the entire shape of the spectrograms.

OUTLINE

I. Introduction

II. **Training Process**

  - Dataset Choosing
  - Model & Train

I. Deep Learning Structure

  - Architecture
  - Loss Function

I. Experiment

II. Future Work

TRAINING PROCESS

DATASET CHOOSING

I. **UrbanSound8k**

- This dataset contains 8732 labeled sound excerpts (<=4s) of urban sounds from 10 classes:

  air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music.

I. **GTZAN Dataset**

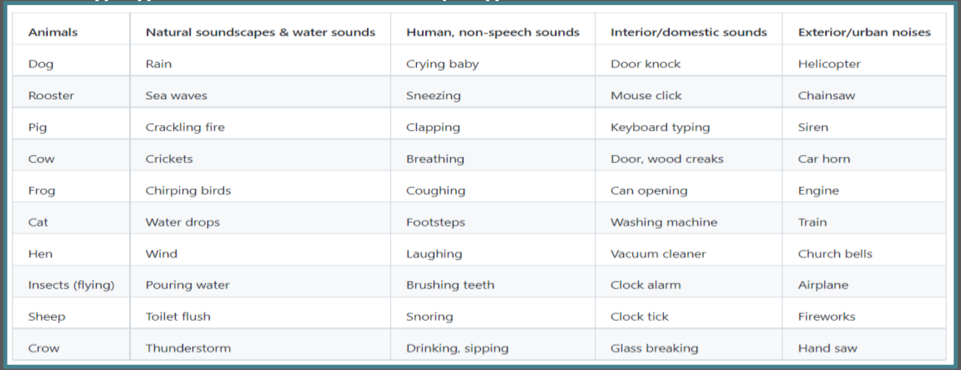
- The GTZAN audio dataset contains 1000 tracks of 30 second length
- There are 10 genres, each containing 100 tracks which are all 22050Hz Mono 16-bit audio files in wav format. The genres are:

  blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, and rock.

TRAINING PROCESS

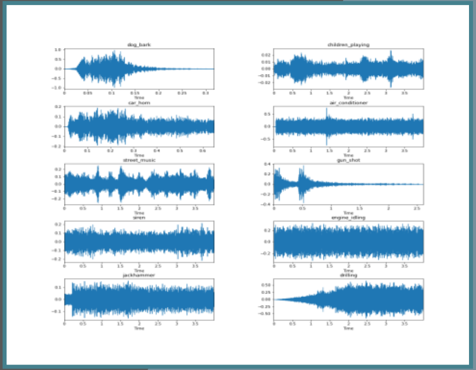
III. ESC-50 (the DATASETS which we choose to implement for this Final Project)

- The ESC-50 dataset is a labeled collection of 2000 environmental audio recordings suitable for benchmarking methods of environmental sound classification, and each of length 5s
- Besides,the dataset consists of 5-second-long recordings organized into 50 semantical classes (with 40 examples per class) loosely arranged into 5 major categories, sounds
- ranging from sounds of Chirping Birds to Car Horn Sounds.



Animals	Natural soundscapes & water sounds	Human, non-speech sounds	Interior/domestic sounds	Exterior/urban noises
Dog	Rain	Crying baby	Door knock	Helicopter
Rooster	Sea waves	Sneezing	Mouse click	Chainsaw
Pig	Crackling fire	Clapping	Keyboard typing	Siren
Cow	Crickets	Breathing	Door, wood creaks	Car horn
Frog	Chirping birds	Coughing	Can opening	Engine
Cat	Water drops	Footsteps	Washing machine	Train
Hen	Wind	Laughing	Vacuum cleaner	Church bells
Insects (flying)	Pouring water	Brushing teeth	Clock alarm	Airplane
Sheep	Toilet flush	Snoring	Clock tick	Fireworks
Crow	Thunderstorm	Drinking, sipping	Glass breaking	Hand saw

ESC-50



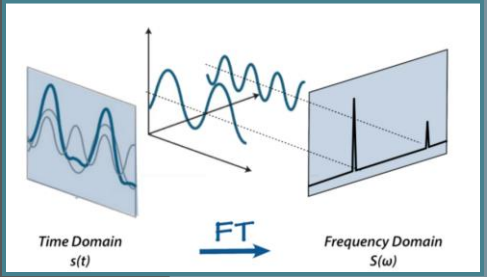
GTZAN

TRAINING PROCESS

DATA PREPROCESSING

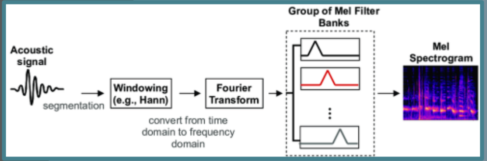
I. Log-Spectrograms

- The Fourier transform is a mathematical formula that allows us to decompose a signal into it’s individual frequencies and the frequency’s amplitude.
- It converts the signal from the time domain into the frequency domain.The result is called a spectrum.



I. Log-MelSpectrograms

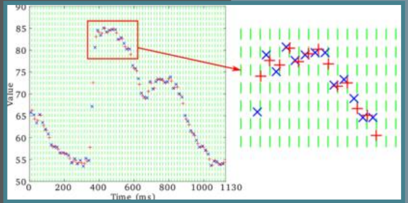
- Humen are better at detecting differences in lower frequencies than higher frequencies. For example, we can easily tell the difference between 500 and 1000 Hz, but we will hardly be able to tell a difference between 10,000 and 10,500 Hz.
- In 1937, someone proposed a unit of pitch such that equal distances in pitch sounded equally distant to the listener→ the mel scale.
- A mel spectrogram is a spectrogram where the frequencies are converted to the mel scale.



TRAINING PROCESS

III. MFCCs

- Mel-frequency cepstral, inverse Fourier transform of the logarithm of the estimated signal spectrum, coefficients are coefficients that collectively make up an MFC.
- In Music Information Retrieval it is often used to describe timbre or classify genre, since it represents short-duration musical textures.
- They are a small set of features (usually about 10–20) which concisely describe the overall shape of a spectral envelope.



III. Gammatone-Spectrogram

- Gammatone filterbank originally proposed by Roy Patterson and colleagues in 1992.
- Gammatone filters were conceived as a simple fit to experimental observations of the mammalian cochlea, and have a repeated pole structure leading to an impulse response.

TRAINING PROCESS

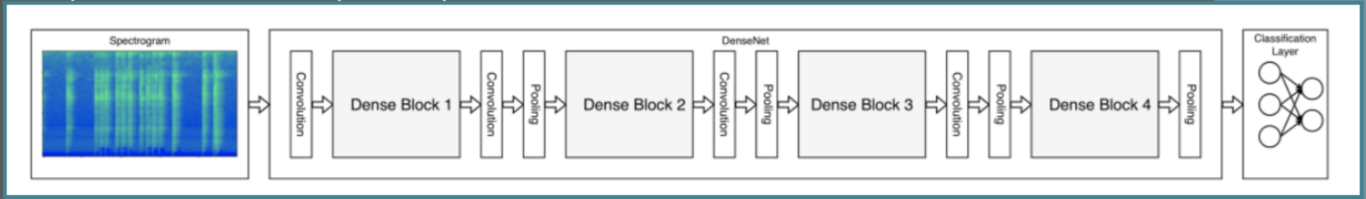
MODEL & TRAIN

I. Inception

- An Inception Layer is a combination of all the layers namely, 1x1 Convolutional layer, 3x3 Convolutional layer, 5x5 Convolutional layers with their output filter banks concatenated into a single output vector forming the input of the next stage.
- A typical Inception network consists of several Inception layers stacked upon each other, with occasional max-pooling layers with stride 2 to halve the resolution of the grid.

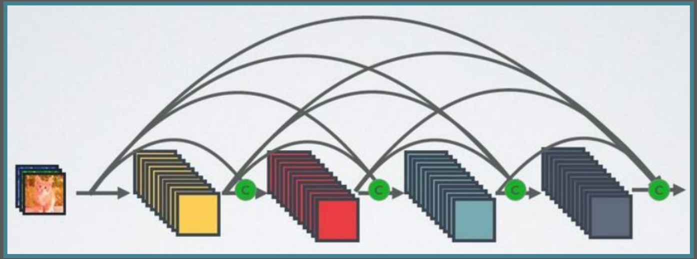
I. DenseNet

- Dense Convolutional Network (DenseNet), connects each layer to every other layer in a feed-forward fashion. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers.

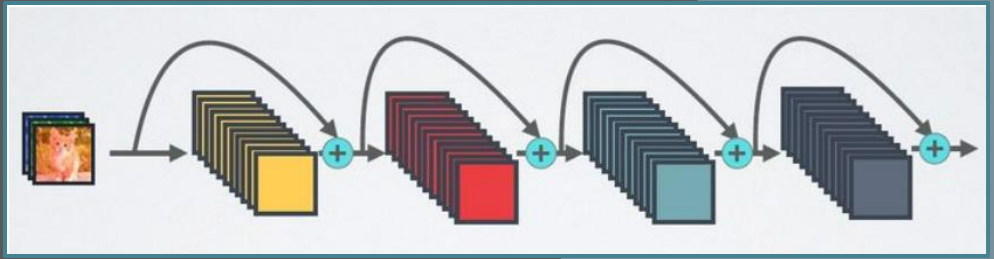


TRAINING PROCESS

- Traditional convolutional networks with L layers have L connections one between each layer and its subsequent layer a dense network has L(L+1) / 2 direct connections.
- DenseNet diminishes the vanishing gradient problem, and it requires fewer parameters to train the model. Dynamic feature propagation takes care of the seamless flow of information.



DenseNet

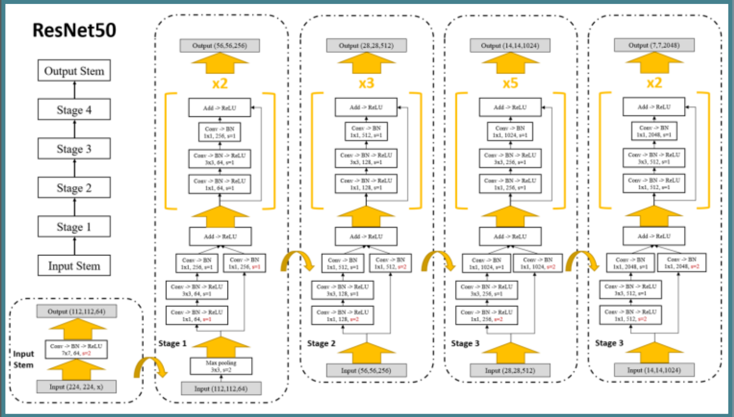


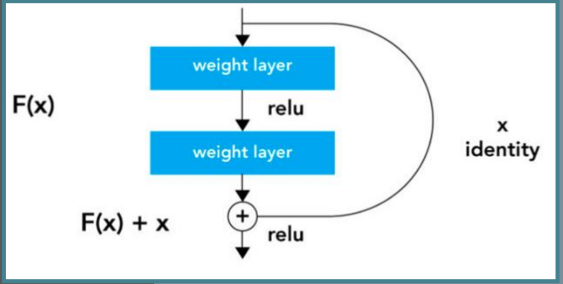
ResNet

TRAINING PROCESS

III. ResNet

- ResNet consists of several residual blocks stacked on top of each other. The residual block has two 3x3 convolutional layers with the same number of output channels. Each convolutional layer is followed by a batch normalization layer and a ReLU activation function. A skip connection is added which skips these two convolution operations and adds the input directly before the final ReLU activation function.
- With adding more layers on top of a network, its performance degrades. This could be blamed on the optimization function, initialization of the network and more importantly vanishing gradient problem.This pproblem of training very deep networks has been alleviated with the introduction of ResNet.


ResNet50
Output Stem
Stage 4
Stage 3
Stage 2
Stage 1
Input Stem
Input Stem
Output (112,112,64)
Conv -> BN -> ReLU 7x7, 64, s=2
Input (224, 224, x)
Output (56,56,256)
x2
Add -> ReLU
Conv -> BN 1x1, 256, s=1
Conv -> BN -> ReLU 3x3, 64, s=1
Conv -> BN -> ReLU 1x1, 64, s=1
Add -> ReLU
Conv -> BN 1x1, 256, s=1
Conv -> BN 1x1, 256, s=1
Conv -> BN -> ReLU 3x3, 64, s=1
Conv -> BN -> ReLU 1x1, 64, s=1
Stage 1
Max pooling 3x3, s=2
Input (112,112,64)
Output (28,28,512)
x3
Add -> ReLU
Conv -> BN 1x1, 512, s=1
Conv -> BN -> ReLU 3x3, 128, s=1
Conv -> BN -> ReLU 1x1, 128, s=1
Add -> ReLU
Conv -> BN 1x1, 512, s=1
Conv -> BN 1x1, 512, s=2
Conv -> BN -> ReLU 3x3, 128, s=1
Conv -> BN -> ReLU 1x1, 128, s=2
Stage 2
Input (56,56,256)
Output (14,14,1024)
x5
Add -> ReLU
Conv -> BN 1x1, 1024, s=1
Conv -> BN -> ReLU 3x3, 256, s=1
Conv -> BN -> ReLU 1x1, 256, s=1
Add -> ReLU
Conv -> BN 1x1, 1024, s=1
Conv -> BN 1x1, 1024, s=2
Conv -> BN -> ReLU 3x3, 256, s=1
Conv -> BN -> ReLU 1x1, 256, s=2
Stage 3
Input (28,28,512)
Output (7,7,2048)
x2
Add -> ReLU
Conv -> BN 1x1, 2048, s=1
Conv -> BN -> ReLU 3x3, 512, s=1
Conv -> BN -> ReLU 1x1, 512, s=1
Add -> ReLU
Conv -> BN 1x1, 2048, s=1
Conv -> BN 1x1, 2048, s=2
Conv -> BN -> ReLU 3x3, 512, s=1
Conv -> BN -> ReLU 1x1, 512, s=2
Stage 3
Input (14,14,1024)



OUTLINE

I. **Introduction**

II. **Training Process**

- Dataset Choosing
- Model & Train

I. **Deep Learning Structure**

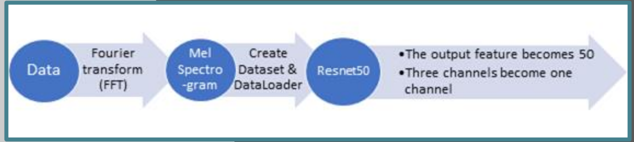
- Architecture
- Loss Function

I. **Experiment**

II. **Future Work**

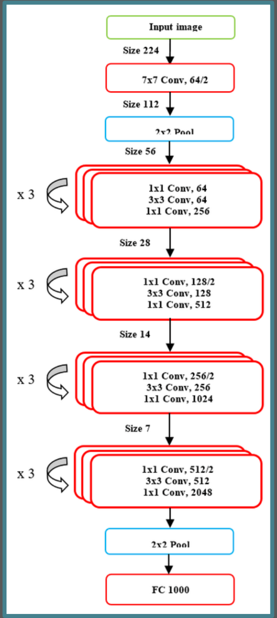
DEEP LEARNING STRUCTURE

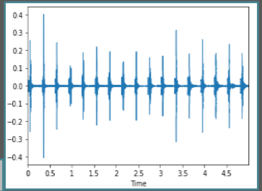
ARCHI TECTURE



Data → Fourier transform (FFT) → Mel Spectro -gram → Create Dataset & DataLoader → Resnet50 →

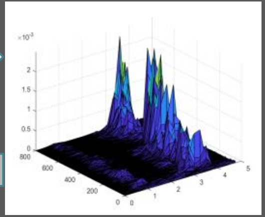
- The output feature becomes 50
- Three channels become one channel





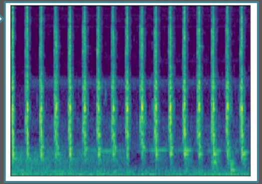
original audio data (ESC50)

Fourier Transform (FFT)



Spectrogram (frequency vs. time vs.amplitude)

Perspective rotation



MelSpectrogram (frequency vs. time)

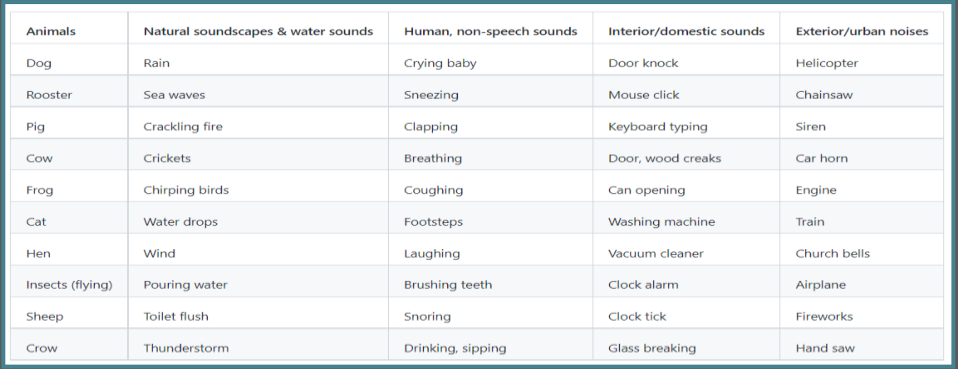
DATASET INTRODUCTION

ESC-50

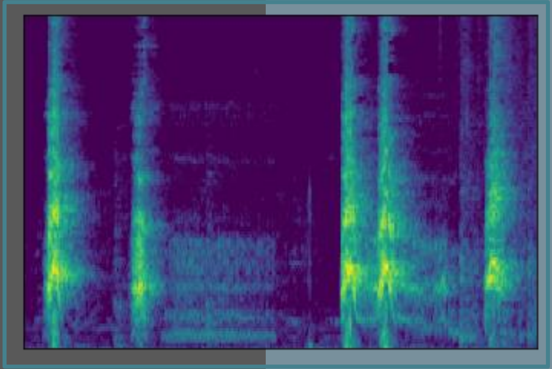
I. The ESC-50 dataset is a labeled collection of 2000 environmental audio recordings suitable for benchmarking methods of environmental sound classification, and each of length 5s

II. Besides,the dataset consists of 5-second-long recordings organized into 50 semantical classes (with 40 examples per class) loosely arranged into 5 major categories, sounds

III. ranging from sounds of Chirping Birds to Car Horn Sounds.

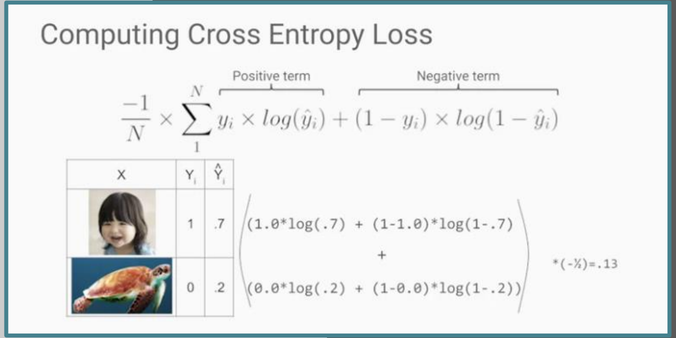


Animals	Natural soundscapes & water sounds	Human, non-speech sounds	Interior/domestic sounds	Exterior/urban noises
Dog	Rain	Crying baby	Door knock	Helicopter
Rooster	Sea waves	Sneezing	Mouse click	Chainsaw
Pig	Crackling fire	Clapping	Keyboard typing	Siren
Cow	Crickets	Breathing	Door, wood creaks	Car horn
Frog	Chirping birds	Coughing	Can opening	Engine
Cat	Water drops	Footsteps	Washing machine	Train
Hen	Wind	Laughing	Vacuum cleaner	Church bells
Insects (flying)	Pouring water	Brushing teeth	Clock alarm	Airplane
Sheep	Toilet flush	Snoring	Clock tick	Fireworks
Crow	Thunderstorm	Drinking, sipping	Glass breaking	Hand saw



DEEP LEARNING STRUCTURE

LOSS FUNCTION: CROSS ENTROPY

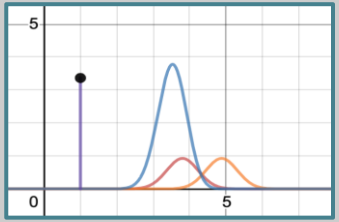


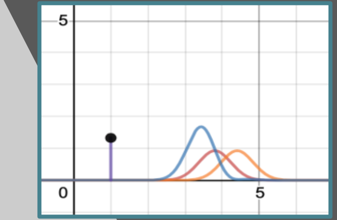
Computing Cross Entropy Loss

$$\frac{-1}{N} \times \sum_{1}^{N} \overbrace{y_i \times log(\hat{y}_i)}^{\text{Positive term}} + \overbrace{(1 - y_i) \times log(1 - \hat{y}_i)}^{\text{Negative term}}$$

X	Y_i	$\hat{Y}_i$
	1	.7
	0	.2

```
(1.0*log(.7) + (1-1.0)*log(1-.7)
              +
(0.0*log(.2) + (1-0.0)*log(1-.2))   *(-½)=.13
```





I. Advantages: Cross Entropy is the most commonly used loss function for classification problems. Entropy is the average amount of information contained in all messages received. The more uncertain events, that is, the more information events, the more information they will have. With high Entropy, the probability of predicting success will also increase.

I. The predicted probability distribution is the orange block, the real probability distribution is the red block, the blue part is the cross-entropy block, and the purple is the calculated value. The more the difference between our predicted value and the actual value, the greater the amount of information representing the connotation, the more uncertainty, and the higher the cross-entropy.

OUTLINE

I. Introduction

II. Training Process

- Dataset Choosing
- Model & Train

I. Deep Learning Structure

- Architecture
- Loss Function

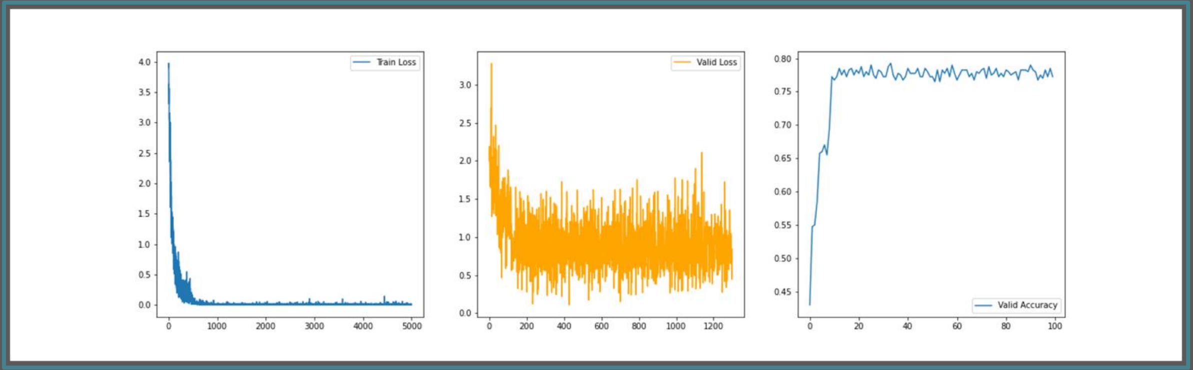
I. Experiment

II. Future Work

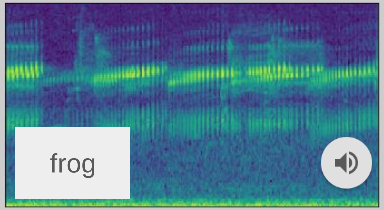
EXPERIMENT

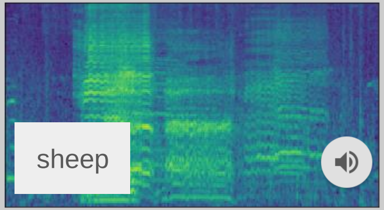
TRAINING RESULT

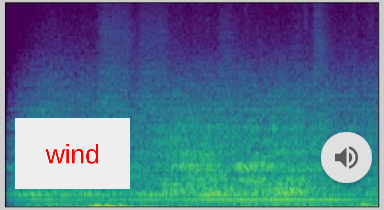
I. Doing training with 100 epoch, we then have the result:



I. With the trained model, we can predict a result by given data. Here we put some prediction sample onh the right hand side.







EXPERIMENT

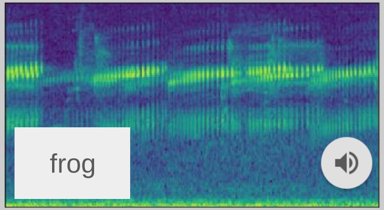
RESULT ANALIZE

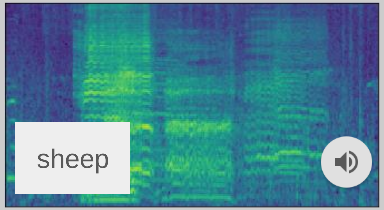
I. **Reason of success**

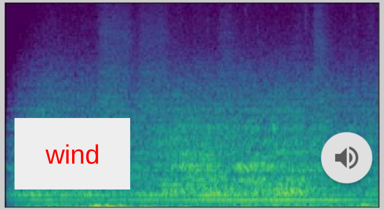
- By transforming the data from audio to frequency based image, each of the classes has a certain range of frequency, which can be classied into a certain class by CNN model.

I. **Reason of failing**

- For two similar audios, they might be composed of very similar frequrncy, which will leads to failed classification as there is no difference between two images without good preprocessing.
- On the other hand, without appropriate data preprocessing, there might be a lot of noise in the audio that caused a bad result.
- Using a ResNet50, a very deep and large model, might leads too many parameters for this audio classification task.



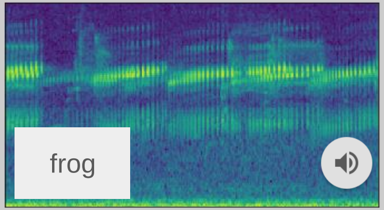


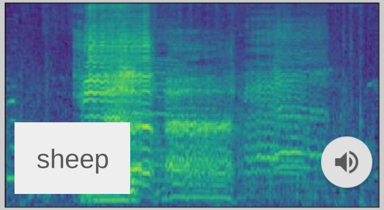


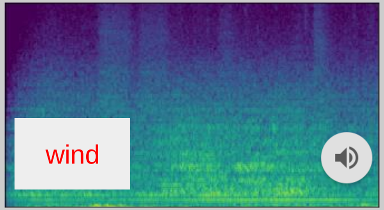
EXPERIMENT

III. Conclusion

- In order to increase the accuracy, augamentation is utilized. However, the results did not change with augamentation.
- We try to simplify the model by using ResNet18 to get the better results, and it works with increasing the accuracy to 80 %.
- From the above two results, we have a brief conclusion. First, the way we do data augamentation might be inappropriate for the classification task, which leads similar or worse results. Second, the parameter of ResNet50 might be too many for this case, which is improved by using ResNet18. Last but not the least, frequency based images tranformed from the audio datas can be used for a CNN model to do the classification task.







OUTLINE

I. **Introduction**

II. **Training Process**

  - Dataset Choosing
  - Model & Train

I. **Deep Learning Structure**

  - Architecture
  - Loss Function

I. **Experiment**

II. **Future Work**

FEUTURE WORK

I. **Data Preprocessing**

- From the previous slides, we know we might need to find an appropriate way for data augamentation, in order to have more data for training while it's hard to collect such a big number of datas.
- On the other hand, we may try to do data denoise to see if this can improve the training result.

I. **Real world audio classification**

- While getting better model, we are using the data from ESC-50 dataset. We may try to use the audio collected by ourselves to see if the model can work with the real world audio.

THANKS FOR LISTENING

Group 34