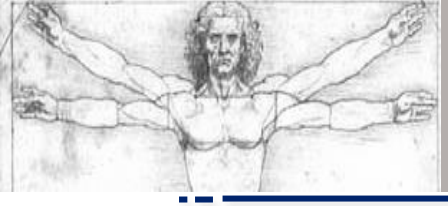
Speech Emotion Recognition

Pattern Recognition & Machine Learning Laboratory
Ha-Na Jo
Oct. 04, 2022

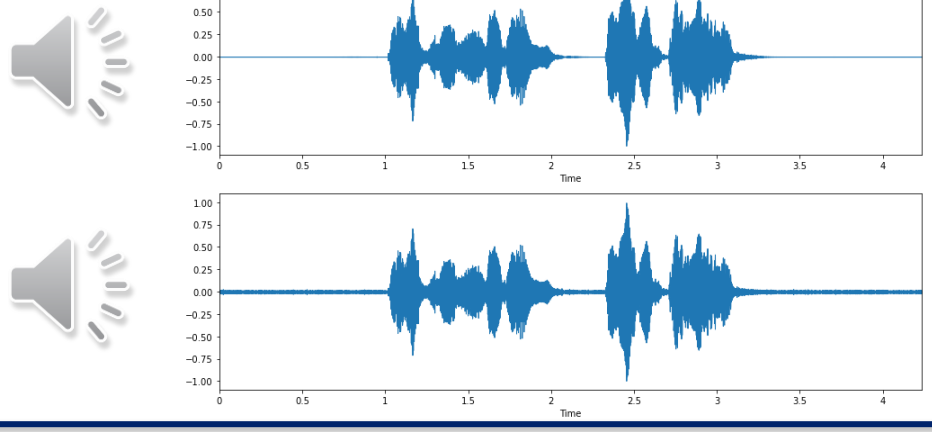
Project Description

- **Goal**
  - Attempting to recognize human emotion and affective states from speech
    - This is capitalizing on the fact that voice often reflects underlying emotion through tone and pitch.
    - This is also the phenomenon that animals like dogs and horses employ to be able to understand human emotion.
- **Datasets used in this project**
  - Crowd-sourced Emotional Multimodal Actors Dataset (Crema-D)
    - 7,442 original clips from 91 actors
    - Six different emotions (anger, disgust, fear, happy, neutral, and sad)
  - Ryerson Audio-Visual Database of Emotional Speech and Song (Ravdess)
    - 1,440 audio-only files from 24 actors
    - Seven different emotions (calm, happy, sad, angry, fear, surprise, and disgust)
  - Surrey Audio-Visual Expressed Emotion (Savee)
    - 480 utterances from four speaker
    - Seven different emotions (anger, disgust, fear, happy, sad, surprise, and neutral)
  - Toronto emotional speech set (Tess)
    - 2800 data points in total
    - Seven different emotions (anger, disgust, fear, happy, surprise, sad, and neutral)



Preprocessing

- **Data information**
  - Total 12,162 audio files
    - 80% for train and 20% for test set
- **Data preprocessing**
  - Data labeling
    - For eight different emotions
      - Neutral, calm, happy, sad, angry, fear, disgust, and surprise
  - Data augmentation
    - Noise injection (prevention overfitting)
      - Add uniform distribution noise in the audio

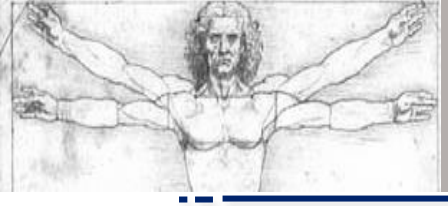


Count of emotions (train set)

Sample of origin audio

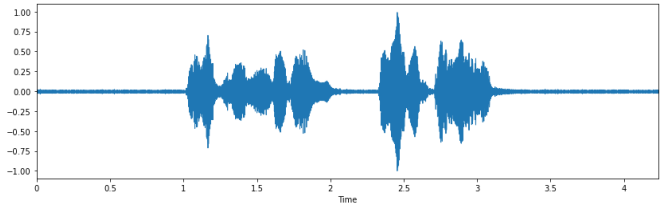
Sample of noise audio

Count of emotions (test set)



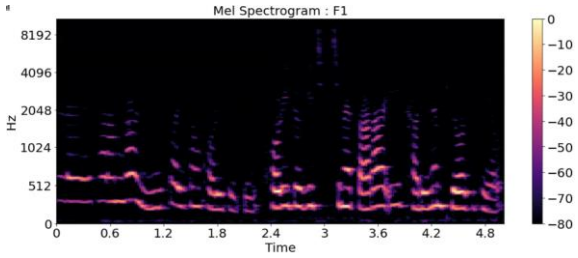
Preprocessing (Cont.)

- Speed modifying and pitch changing

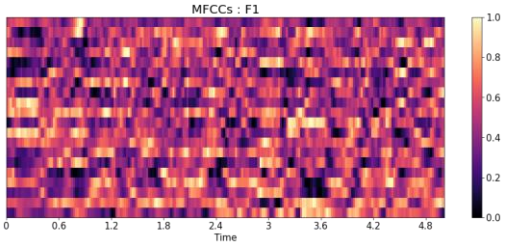


Sample of pitch changed audio

- **Feature extraction**
  - Zero crossing rate
    - The rate of sign-changes of the signal during the duration of a particular frame
  - Mel spectrogram
    - Analyzing audio with time-varying frequency characteristics
  - Mel-frequency cepstral coefficients (MFCC)
    - The inverse Fourier transform of the logarithmic value of the spectral signal
  - Chroma short-time Fourier transform (STFT)
    - The features of chroma contain harmonies or pitches information
  - Root mean square value

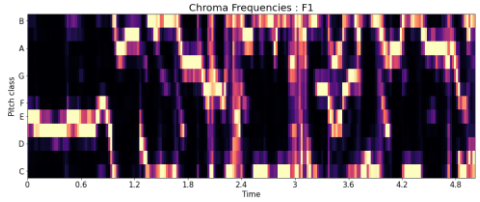


Example of Mel spectrogram



Example of MFCC

- **Preprocessing results**
  - 29,154 data for a train set and 7,332 data for a test set (tripled from original data)
    - By data augmentation
  - 162 features per 1 data are extracted



Example of chroma

Baseline Model

- **Resnet**
  - **Goal**
    - Prevented gradient vanishing and exploding problems in deep neural networks
  - **Methods**
    - Skip connection in the residual network
      - Solved gradient vanishing problem
    - Bottleneck design
      - Solved time-consuming problem
  - **Result**
    - A lower error rate in deep neural networks (34-layer) compared to lighter neural networks (18-layer)

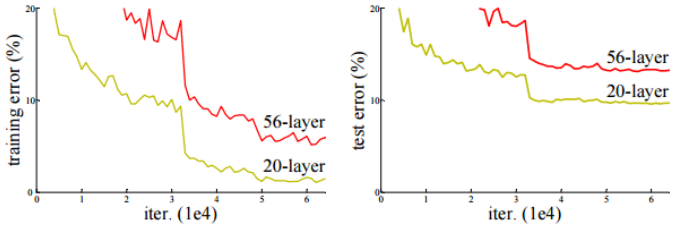


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

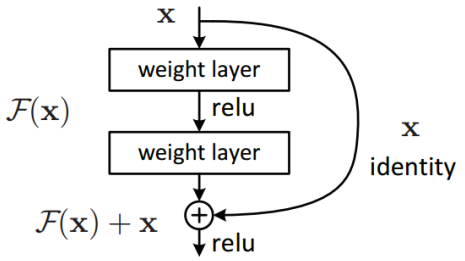


Figure 2. Residual learning: a building block.

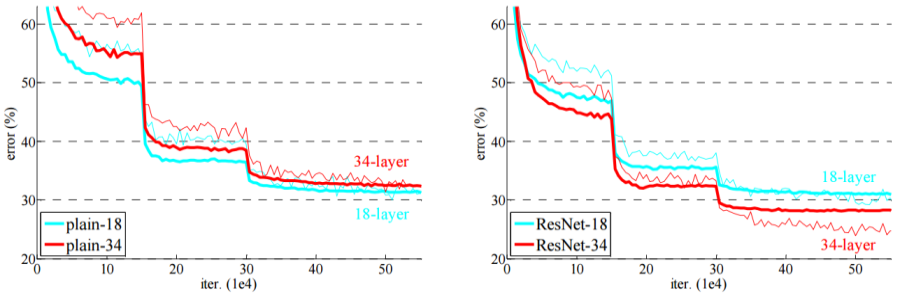


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

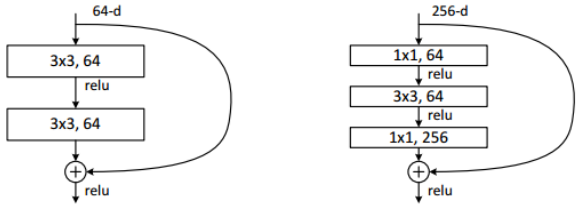


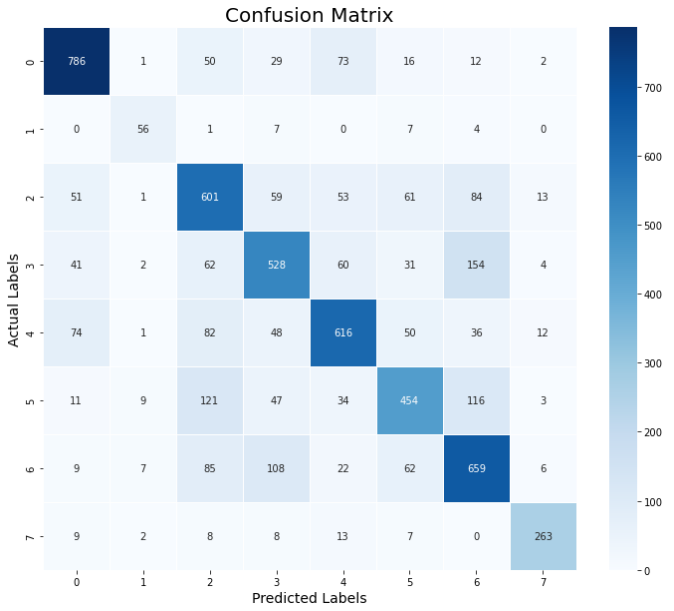
Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

Baseline Model (Cont.)

- **Training data**
  - Features data after data augmentation
    - 29,154(80 % of data) for train and 7,332(20 % of data) for test
      - 23,323(80 % of train set) for train and 5,831(20 % of train set) for validation
- **Training setting**
  - Resnet 34-layer, basic block
    - Batch size: 100, epoch: 30, learning rate: 0.0001
- **Result**
  - 88.21 % accuracy for the training set
  - 67.96 % accuracy for the validation set

Loss of train and validation

Accuracy of train and validation



Confusion matrix of validation

Additional Information

- **Git hub**
  - https://github.com/HanaJo-ku/NNAP
    - **Data part**
      - Feature data with label file for a train (.csv)
        - » You should train using this data, NOT RAW DATA
      - Feature data without label file for a test (.csv)
    - **Run part**
      - Preprocessing and feature extraction code (.ipynb)
      - Training and validation run code (.ipynb)
    - **Reference**
      - Train set, test set raw files (google drive link)
- **E-mail**
  - hn_jo@korea.ac.kr