Text-Independent Speaker Verification Using 3D Convolutional Neural Networks

by

Royston Rodrigues (16307R004)

Kanhaiya Kumar (13D070046)

Indian Institute of Technology Bombay

27th Nov. 2017

Outline

1. Task Definition

2. Implementation Details

    1. Data Representation
    2. Architecture Details of Development Phase
    3. Enrollment Phase
    4. Evaluation Phase

3. Dataset Details

4. Results

Task Definition

The Speaker verification problem refers to verifying the claimed identity of a speaker by using their voice characteristics. Our speaker verification system consists of three phases:

- Development Phase
  - Training 3D CNN
  - Discriminative Feature Extraction
- Enrollment Phase
  - Codebook formation
- Evaluation Phase
  - Speaker Identification
  - Speaker Verification

Implementation Details

Data Representation:

Typical MFCC Pipeline:

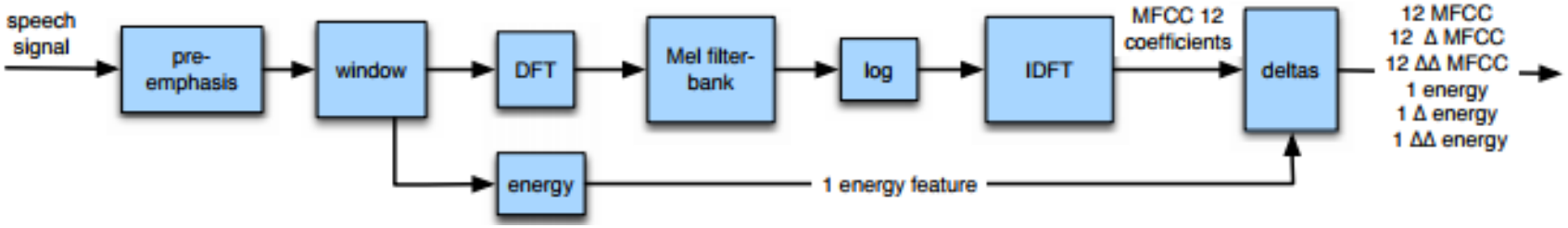


Image Source : D. Jurafsky, J. H. Martin, Speech and Language Processing

Implementation Details

Data Representation:

Modified MFEC Pipeline:

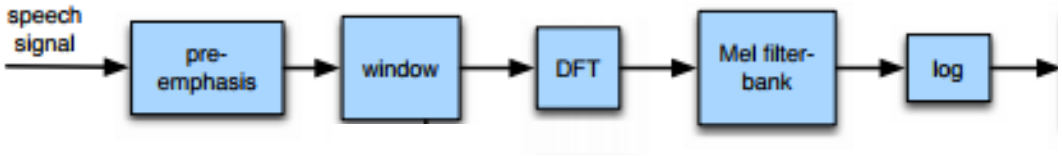
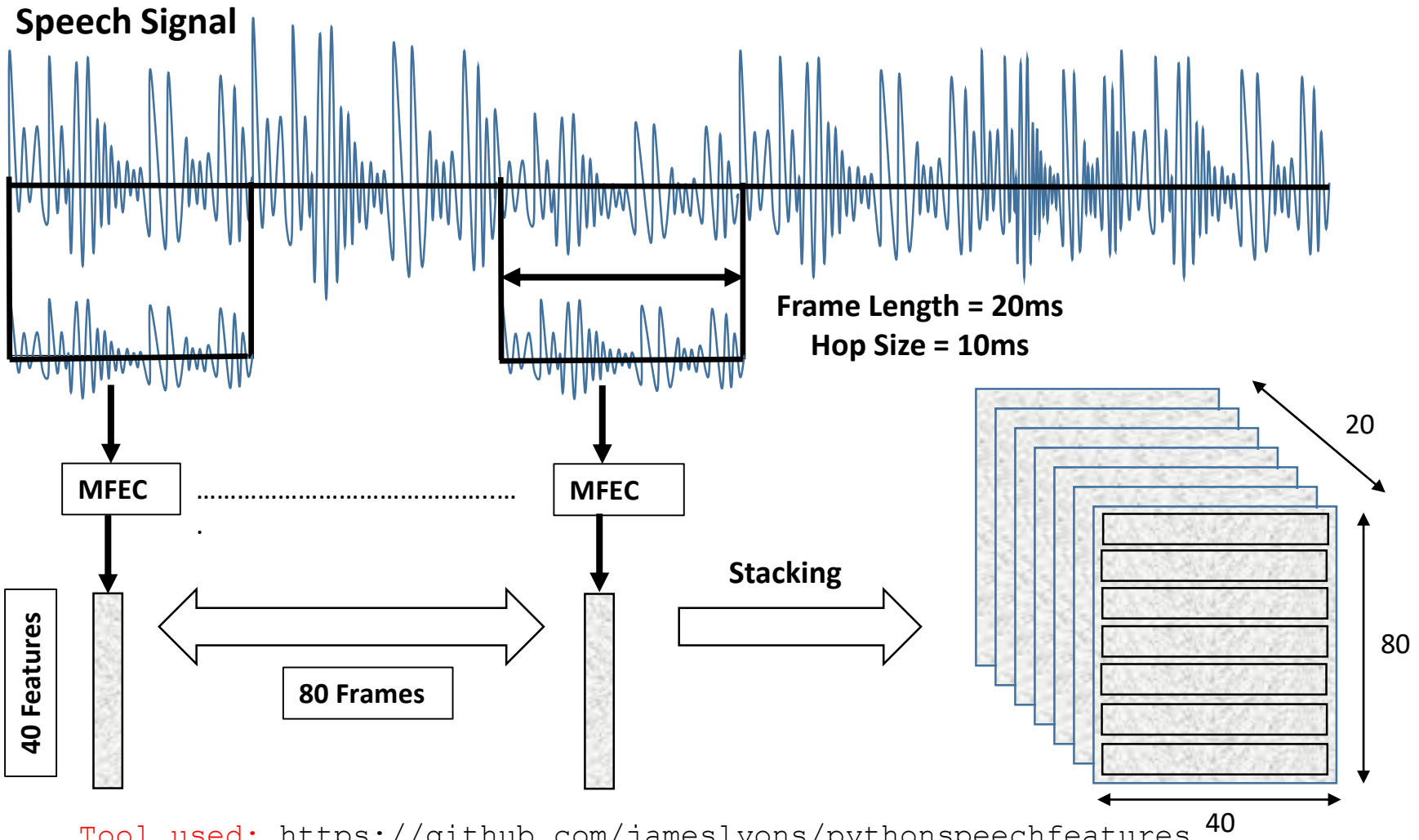


Image Source : D. Jurafsky, J. H. Martin, Speech and Language Processing

Implementation Details

Data Representation:



Tool used: https://github.com/jameslyons/pythonspeechfeatures

Implementation Details

Architecture Details of Development Phase :

Layer Name	Input Size	Output Size	Kernel	Stride
Conv1-1	20 x 80 x 40 x 1	18 x 80 x 36 x 16	3 x 1 x 5	1 x 1 x 1
Conv1-2	18 x 80 x 36 x 16	16 x 36 x 36 x 16	3 x 9 x 1	1 x 2 x 1
Pool1	16 x 36 x 36 x 16	16 x 36 x 18 x 16	1 x 1 x 2	1 x 1 x 2
Conv2-1	16 x 36 x 18 x 16	14 x 36 x 15 x 32	3 x 1 x 4	1 x 1 x 1
Conv2-2	14 x 36 x 15 x 32	12 x 15 x 15 x 32	3 x 8 x 1	1 x 2 x 1
Pool2	12 x 15 x 15 x 32	12 x 15 x 7 x 32	1 x 1 x 2	1 x 1 x 2
Conv3-1	12 x 15 x 7 x 32	10 x 15 x 5 x 64	3 x 1 x 3	1 x 1 x 1
Conv3-2	10 x 15 x 5 x 64	8 x 9 x 5 x 64	3 x 7 x 1	1 x 1 x 1
FC4	8 x 9 x 5 x 64	64	-	-
FC5	64	200	-	-

Table 1: Architecture Details

PYTORCH

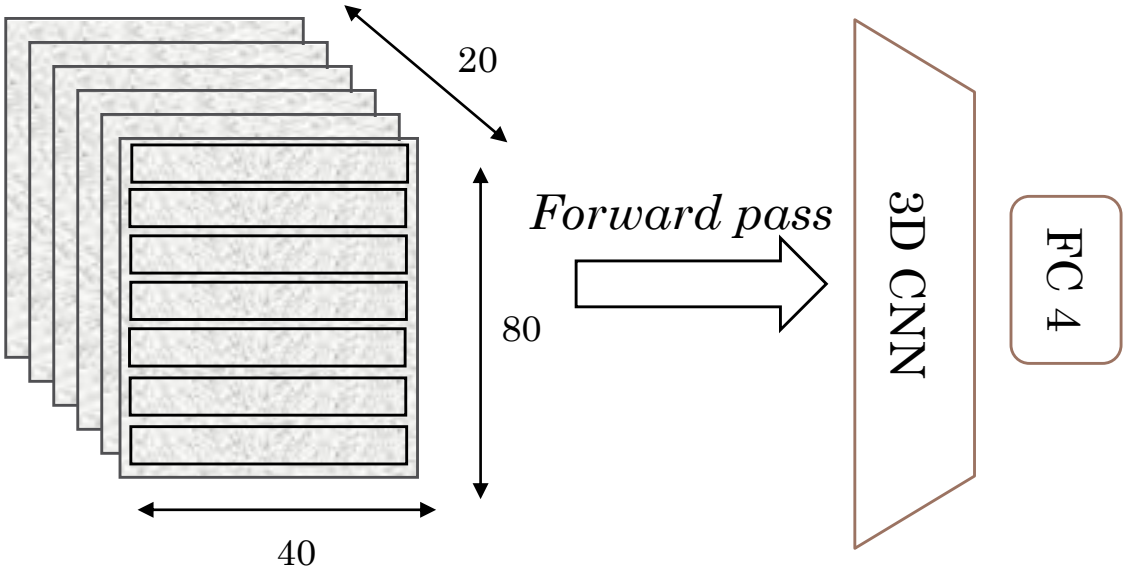
Implementation Details

Architecture Details of Development Phase :

- Activation Function – PReLU
- Loss Function – Cross Entropy, Classify 200 Speakers
- Pooling – Careful Temporal Pooling
- Dropout – Huge Training Time, Settled for Weight Decay L_2
- Batch Norm – Important, Otherwise Training is Stuck, No learning
- Batch Size – As much as possible in the GPU, Batch size = 128

Implementation Details

Enrollment Phase :



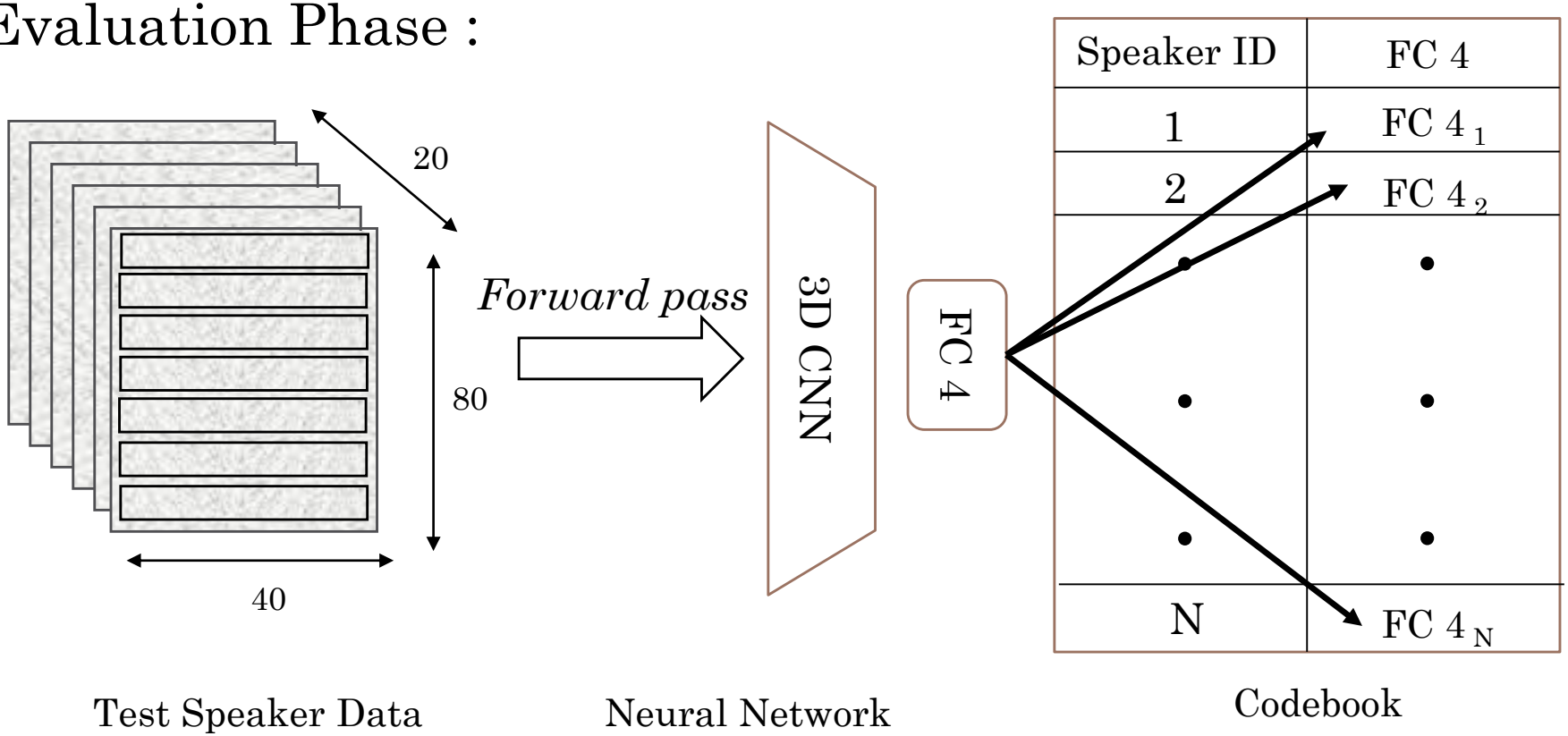
Speaker ID	FC 4
1	FC 4 $_1$
2	FC 4 $_2$
•	•
•	•
•	•
N	FC 4 $_N$

Speaker Data | Neural Network | Codebook

Important : *Set network training flag to false !!!*

Implementation Details

Evaluation Phase :



- *Minimum distance of test feature to existing enrolled features*
- *Distance threshold needed to avoid unenrolled speakers getting verified*

Implementation Details

Data Set Details :

VoxCeleb : –

- 100,000 utterances for 1,251 celebrities.
- Size of this dataset is 160 GB.
- 3 days to download.
- Parallel on four machines.

For Training Development stage :-

- Used 200 speakers data.
- Each speaker had atleast 80 examples.
- Validation Set : Randomly Sampled 10% Subset

For Enrollment stage :-

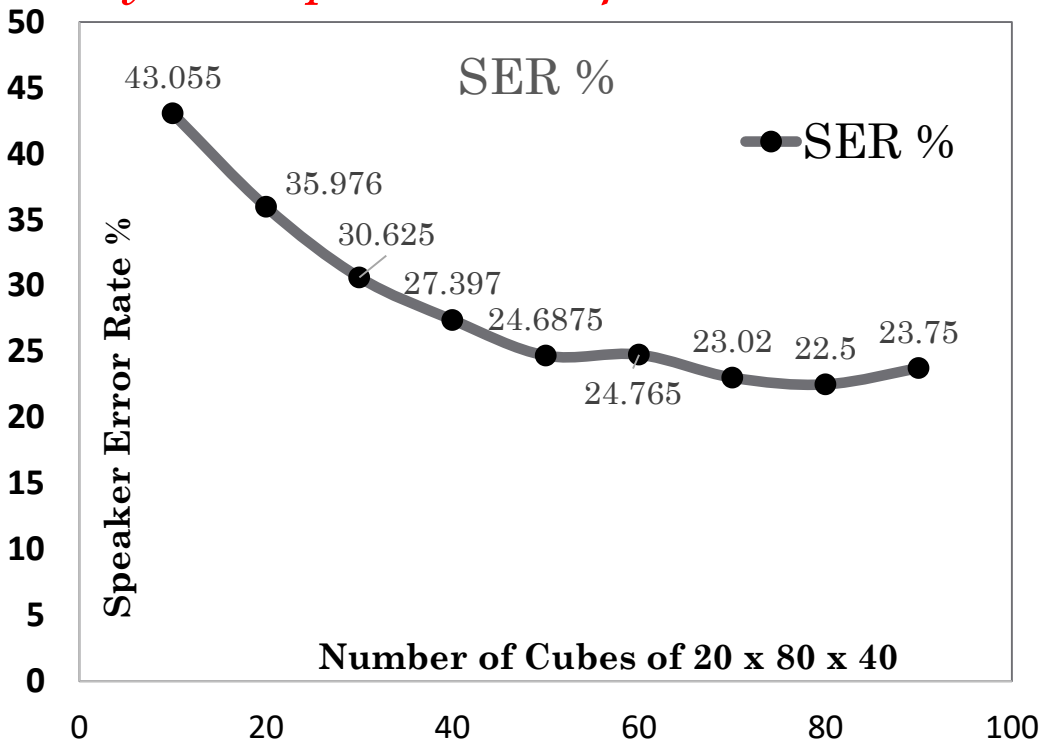
- *Codebook Formation*
- *Case 1* : Using speakers from training set (32x80)
- *Case 2* : From test set (32x80)
- On an average 16 mins of speech required for enrollment.

For Evaluation stage :-

- Speaker Identification
- Speaker Verification
  - Positive and negative data.

Result for speaker Identification task

How Many Examples needed for enrollment ?

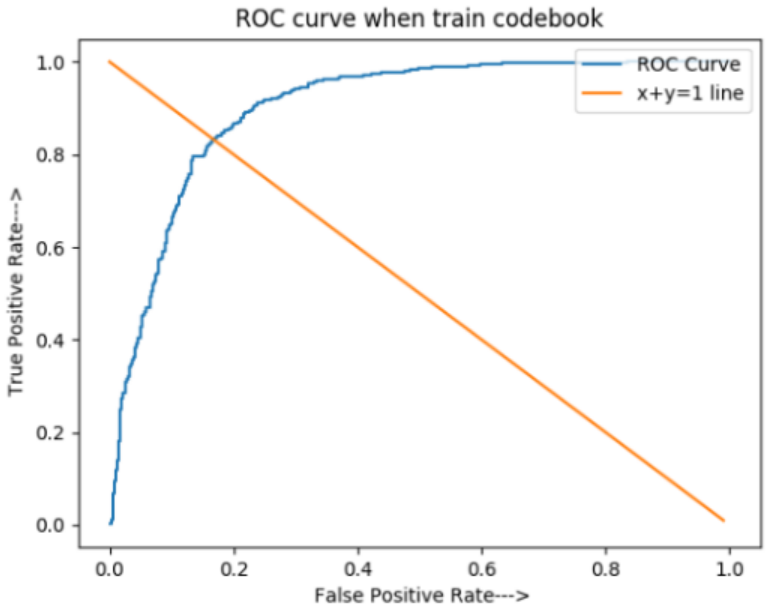


Why So many example during enrollment ?

Data	Accuracy
Train	92.491 %
Validation	68.213 %

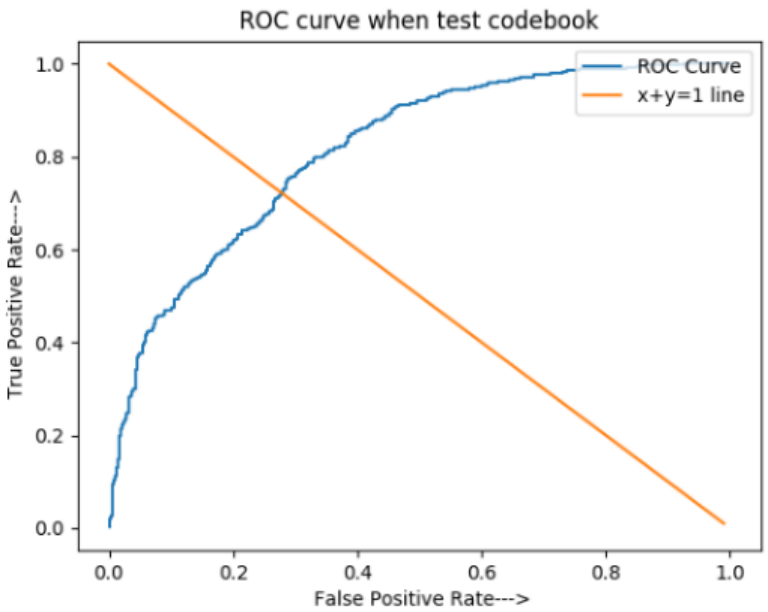
Development phase training results

Results for speaker verification task



Error	Percentage
EER	16.875 %
SER	6.09375 %

Error rate for users from training dataset



Error	Percentage
EER	27.8125 %
SER	22.5 %

Error rate for users only from test dataset

Thank You