A COMPARISON OF DEEP LEARNING METHODS FOR ENVIRONMENTAL SOUND DETECTION

PRESENTER

JUNCHENG (BILLY) LI

AUTHORS

Juncheng Li, Wei Dai, Florian Metze, Shuhui Qu, and Samarjit Das

BOSCH

Carnegie Mellon University

OVERVIEW

1) INTRODUCTION

- Environmental Sounds
- Dataset
- Feature Extraction

2) TRADITIONAL METHOD

- Gaussian Mixture Model
- Identity Vector

4) CONCLUSION

- Discussion
- Conclusion

DCASE CHALLENGE

(Detection and Classification of Acoustic Scenes and Events 2016)

3) DEEP LEARNING METHOD

- Deep Neural Network
- Recurrent Neural Network
- Convolutional Neural Network
- Model Ensembling

ENVIRONMENTAL SOUNDS

15 types of locations

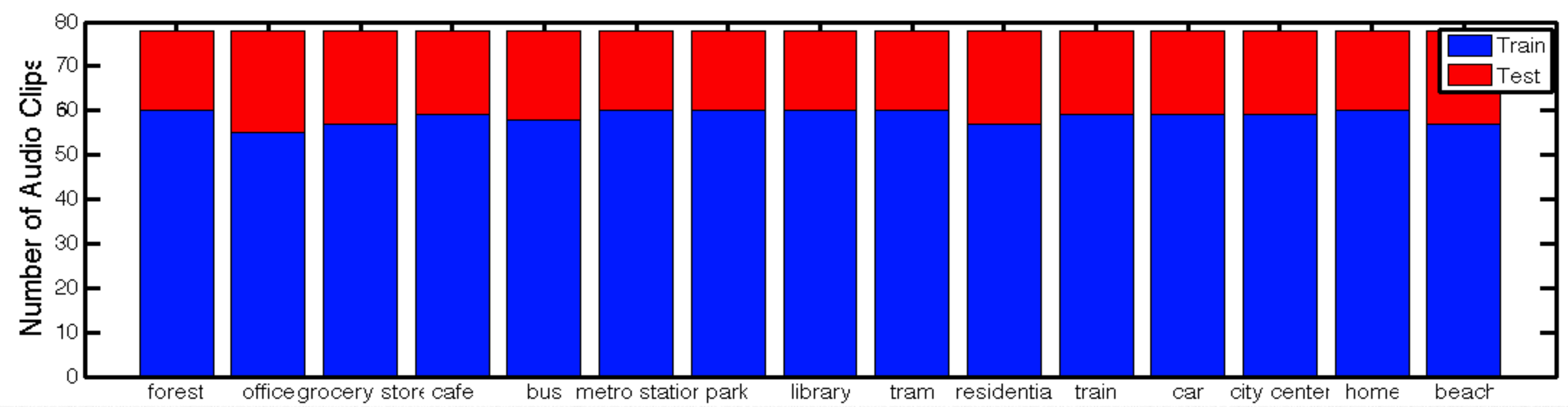
- BUS
- CAFÉ / RESTAURANT
- CAR
- CITY CENTER
- FOREST PATH
- GROCERY STORE
- HOME
- LAKESIDE BEACH
- LIBRARY
- METRO STATION
- OFFICE
- RESIDENTIAL AREA
- TRAIN
- TRAM
- URBAN PARK

DATASET

13 hours

of recording in total

- **1170** clips development set：
  - **4**-fold cross validation
  - **880** for **training**, **290** for **testing**
  - **30** seconds / clip, **~59** clips training per class
- **390** clips evaluation set
- **24**-bit audio, **2** channels, sampling rate **44100Hz**

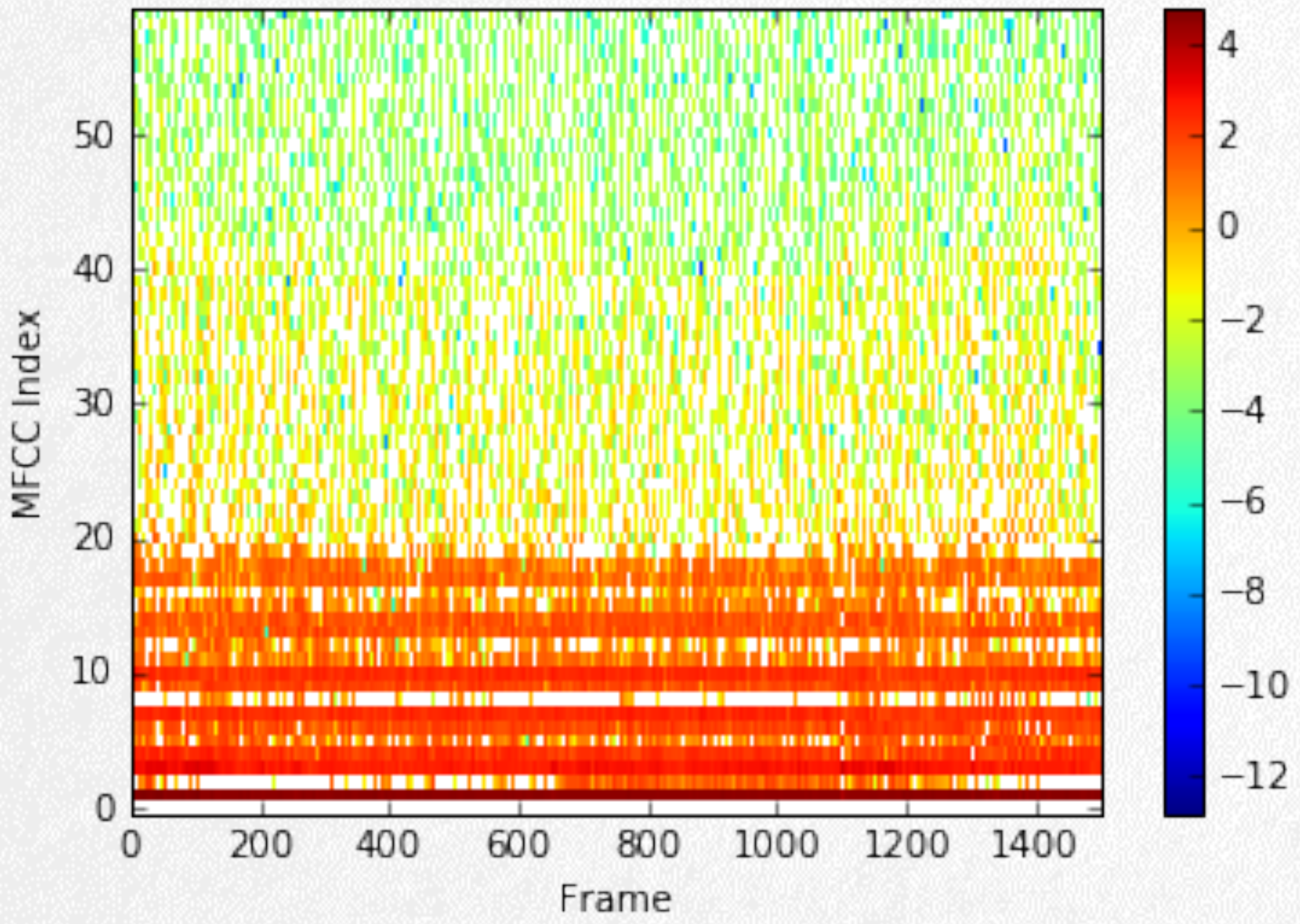


SIGNAL PROCESSING FEATURE EXTRACTION

MFCC

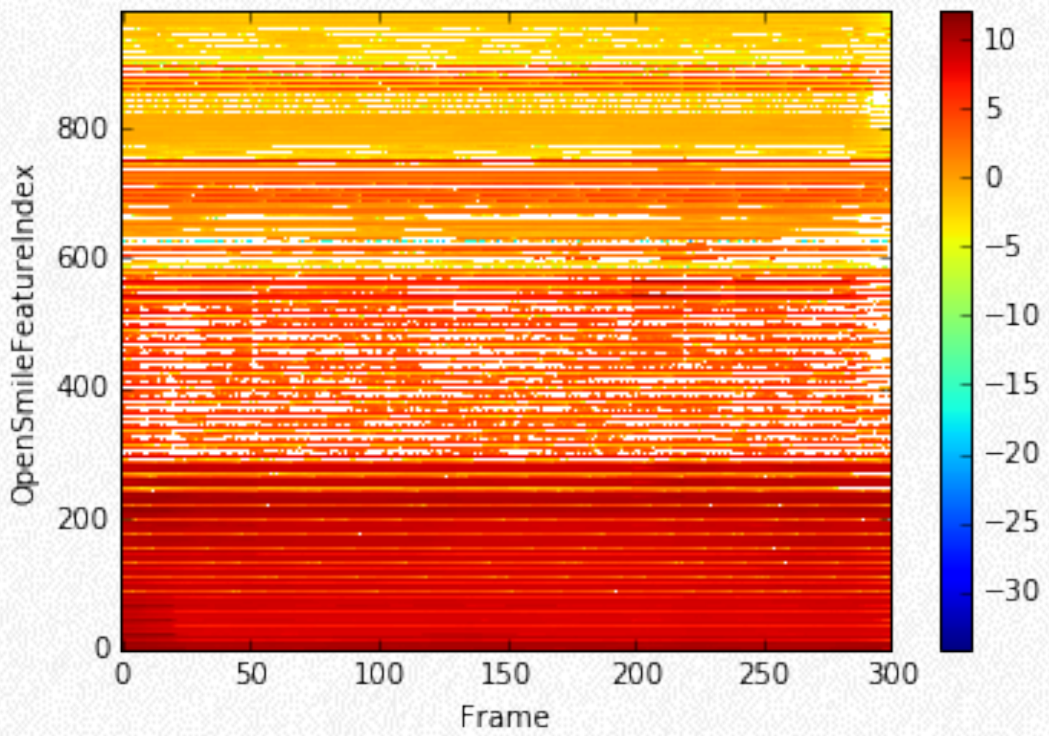
Mel-frequency cepstral coefficient (61-dim)

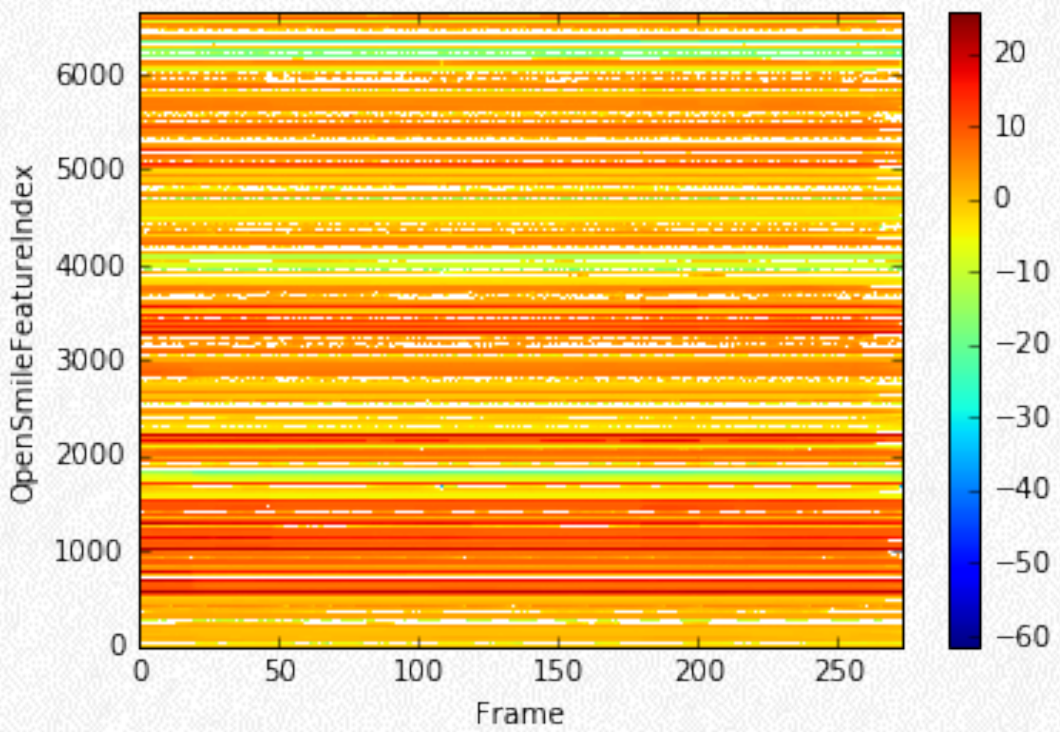
- Monaural MFCC : 23 window 20ms, excluding 0th, including 1st 2nd order difference
- Binaural MFCC **(BiMFCC)** : left, right, difference



OPENSMILE

[Eyben et al ,2010]
(983-dim, 6573-dim)

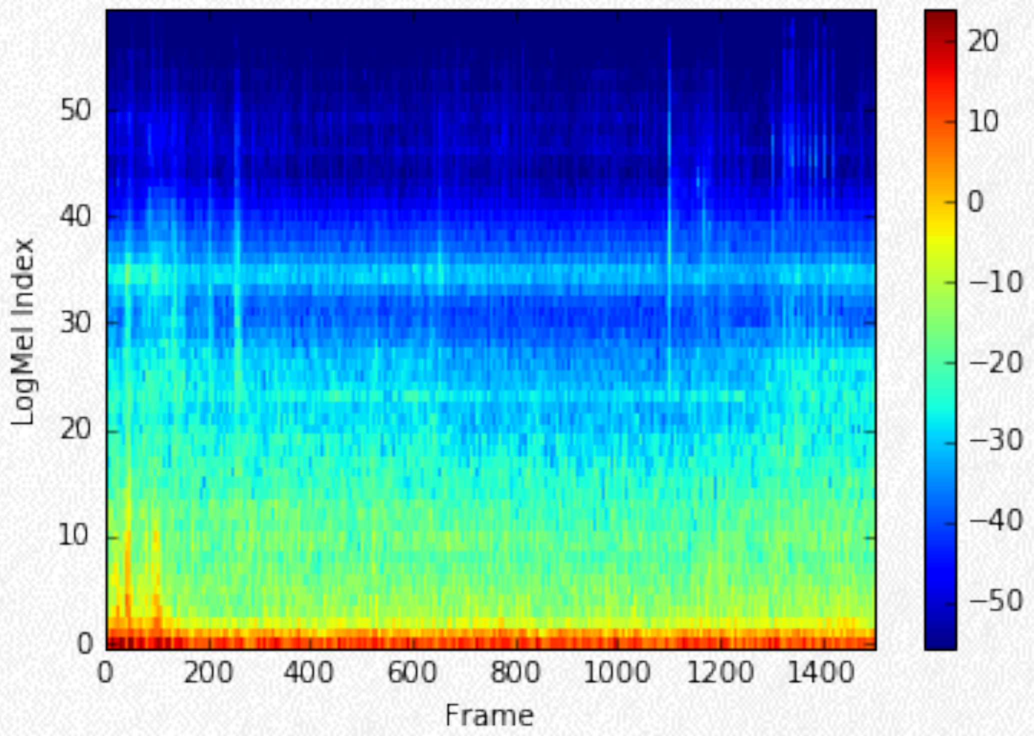


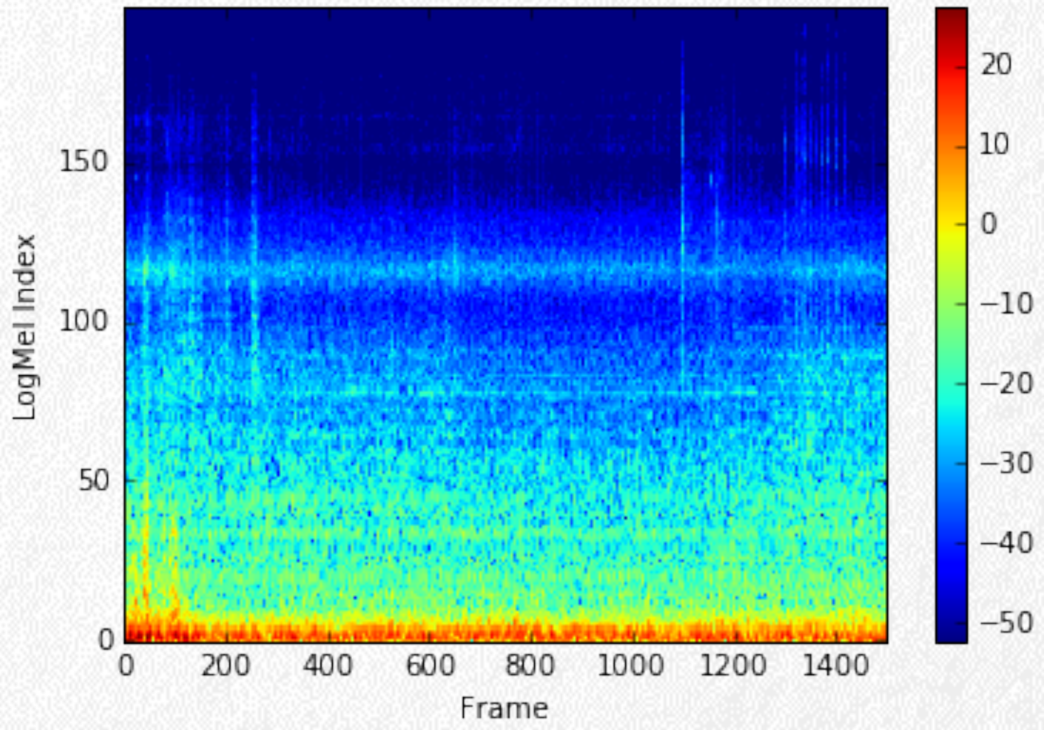


LOGMEL

(60-dim, 200-dim)

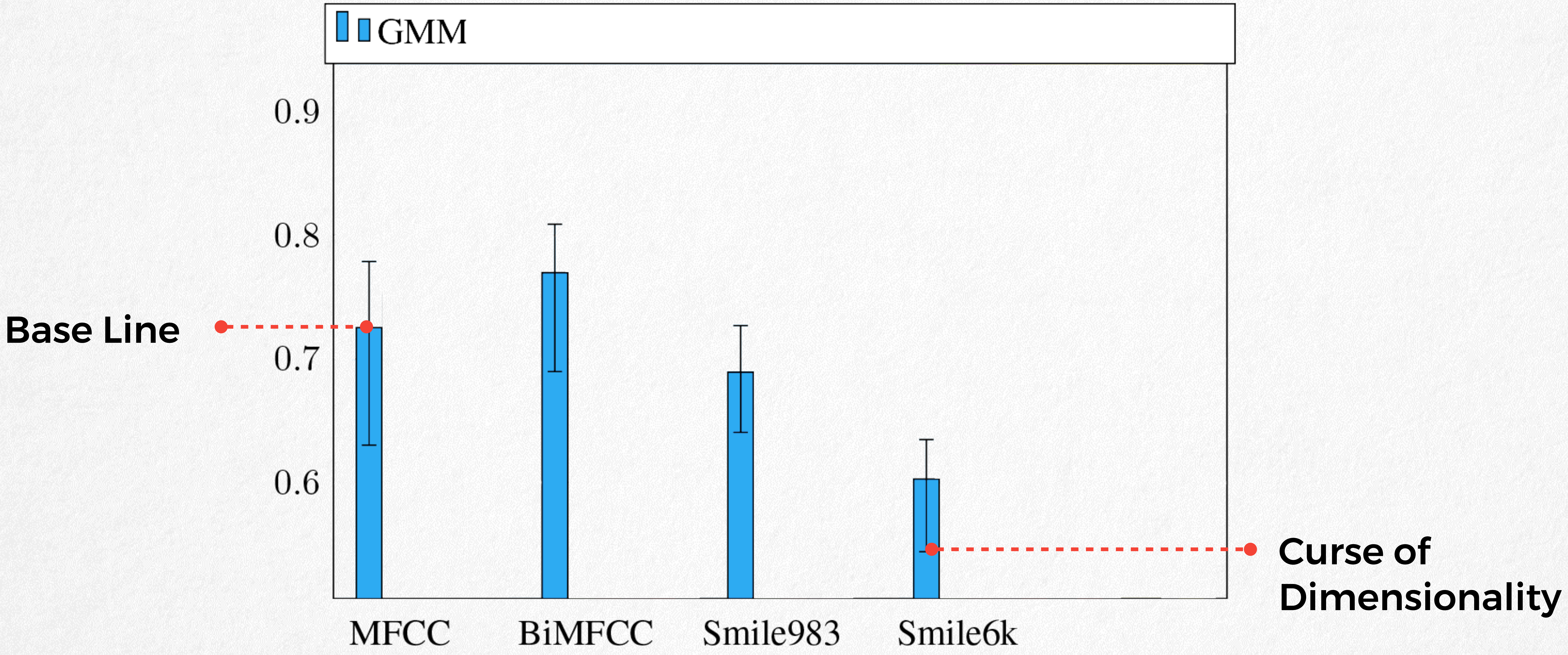
- Computed by LibROSA
- 60 and 200 mel filters





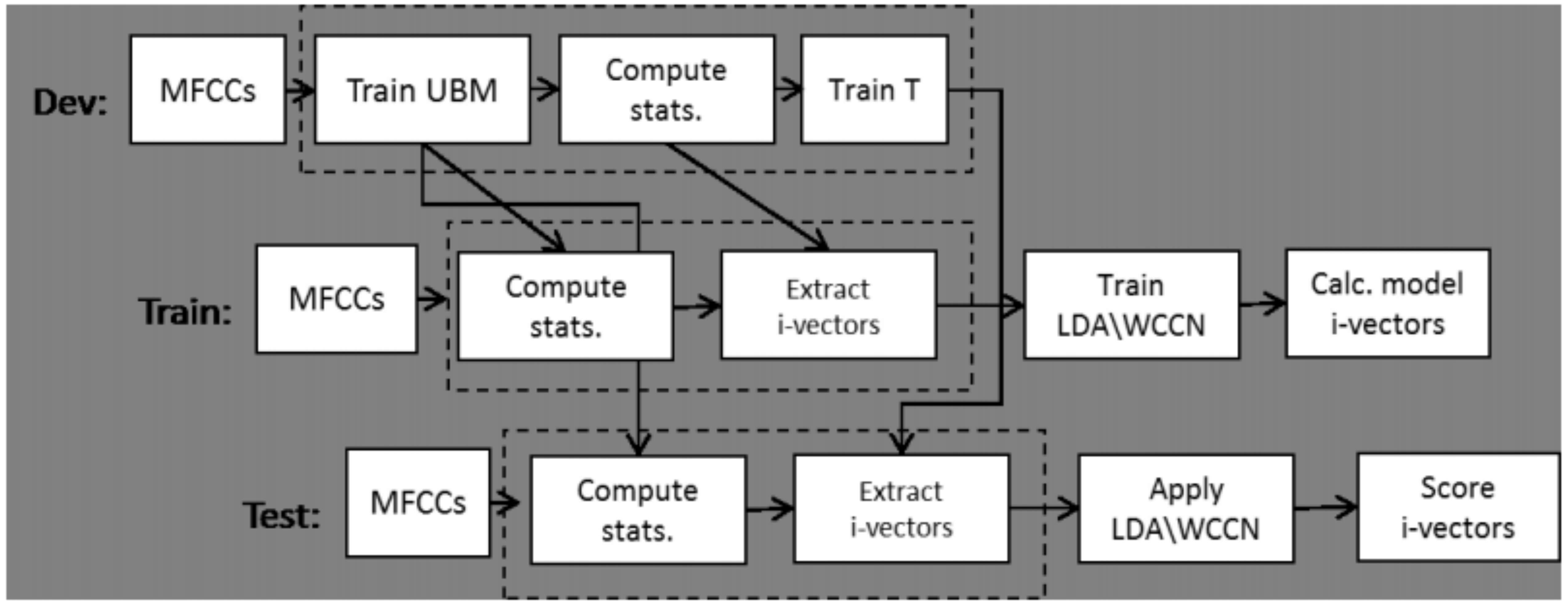
GAUSSIAN MIXTURE MODEL (GMM)

4- fold CV avg. accuracy



IDENTITY VECTOR (I-VECTOR)

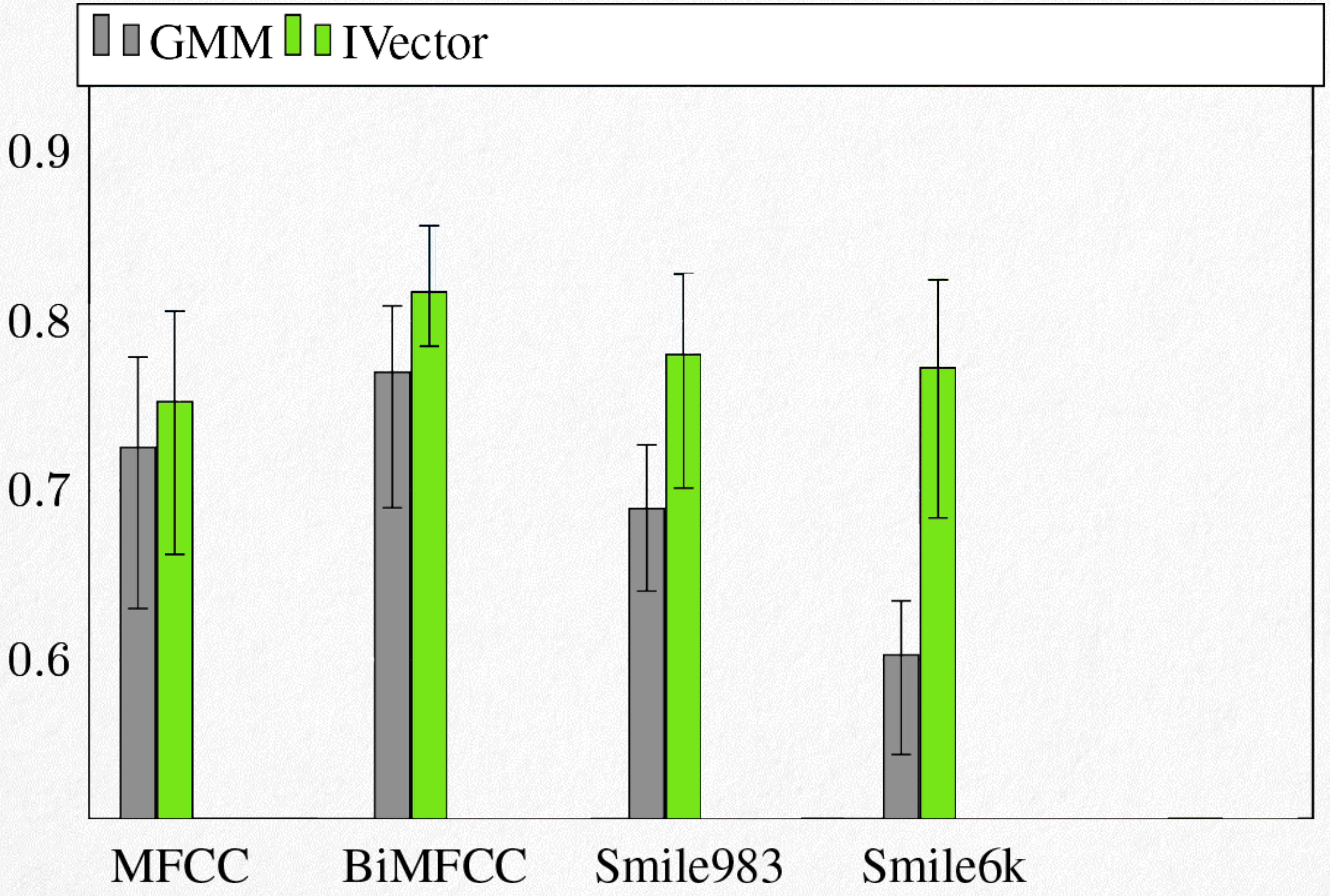
- State-of-the-art technique in the speaker verification field
- Universal background model (**UBM**), GMM with 256 components
- Mean Super Vector $M = m + T \cdot y,$
- Use **Kaldi** Toolkit and perform Linear Discriminant Analysis (**LDA**), and Within Class Covariance Normalization (**WCCN**)
- Each projected test i-vector is scored (**cosine similarity**) against all model i-vectors.



Block-diagram of Our I-vector Pipeline

IDENTITY VECTOR (I-VECTOR)

4- fold CV avg. accuracy



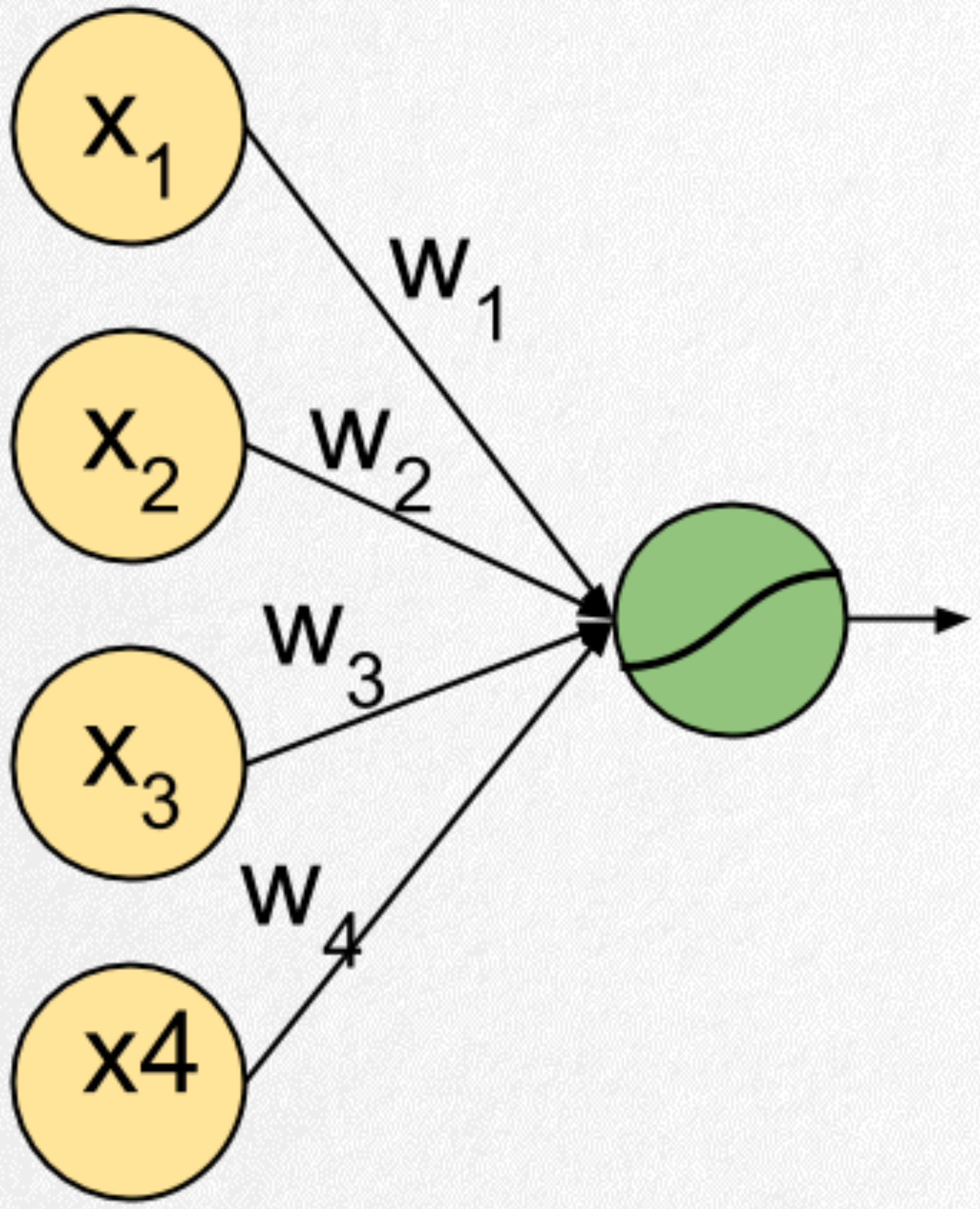
EXPERIMENT SETUP

- **Hyper parameter Tuning**
  - Tuned #layers, layer size, activation, optimizer, dropout, batch norm
  - Train >500 models
- **System Configuration**
  - 4 Titan X (single node)
  - 128GB, 16 cores (Intel i7)
- **Framework**
  - Tensorflow and Keras

MUILTI-LAYER PERCEPTRON

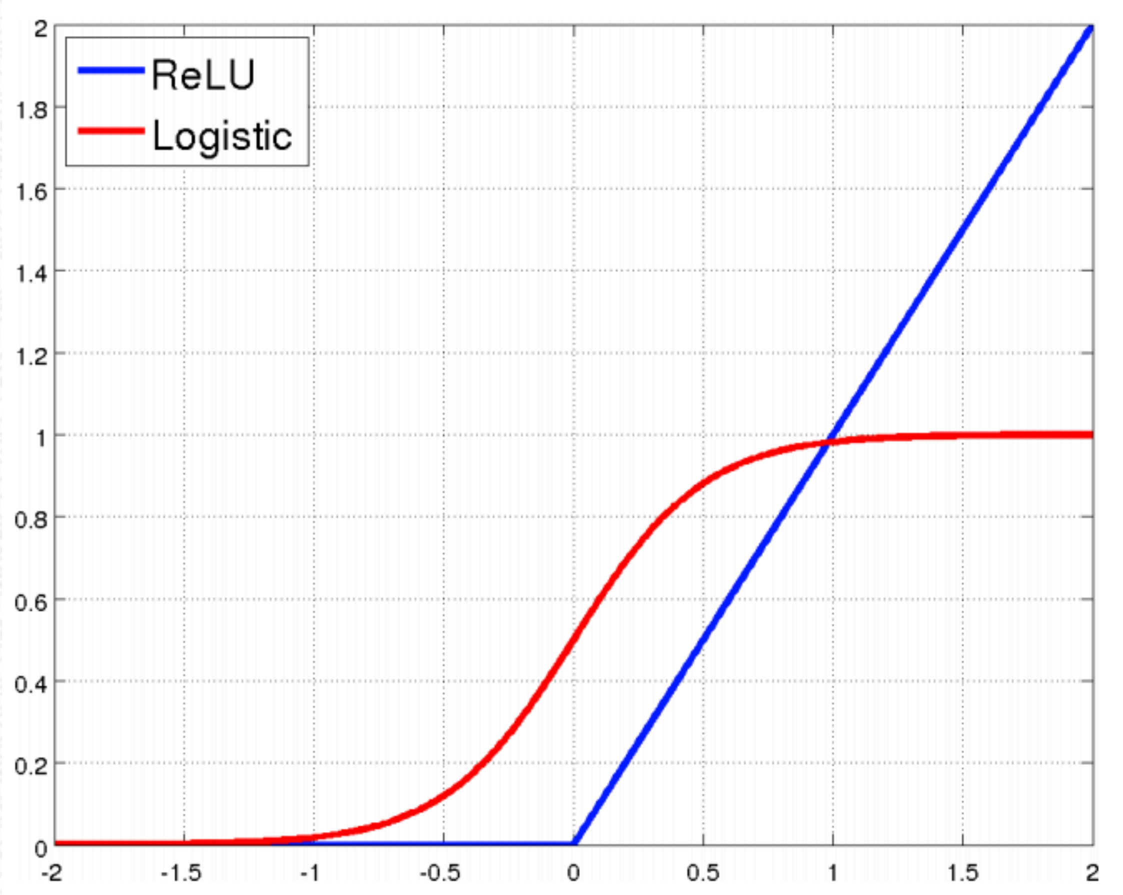
$$\boldsymbol{h} = \mathrm{step}(W_{xh}\boldsymbol{x} + \boldsymbol{b}_h)$$
$$y = \boldsymbol{w}_{hy}\boldsymbol{h} + b_y.$$

- DNN input:
  - Features over 2s windows
  - Window shift 1s
- DNN output:
  - Majority voting of window level decisions



$$\mathrm{step}(x) =$$

(Source of non-linearity)



Rectified Linear Unit (ReLU)
Sigmoid

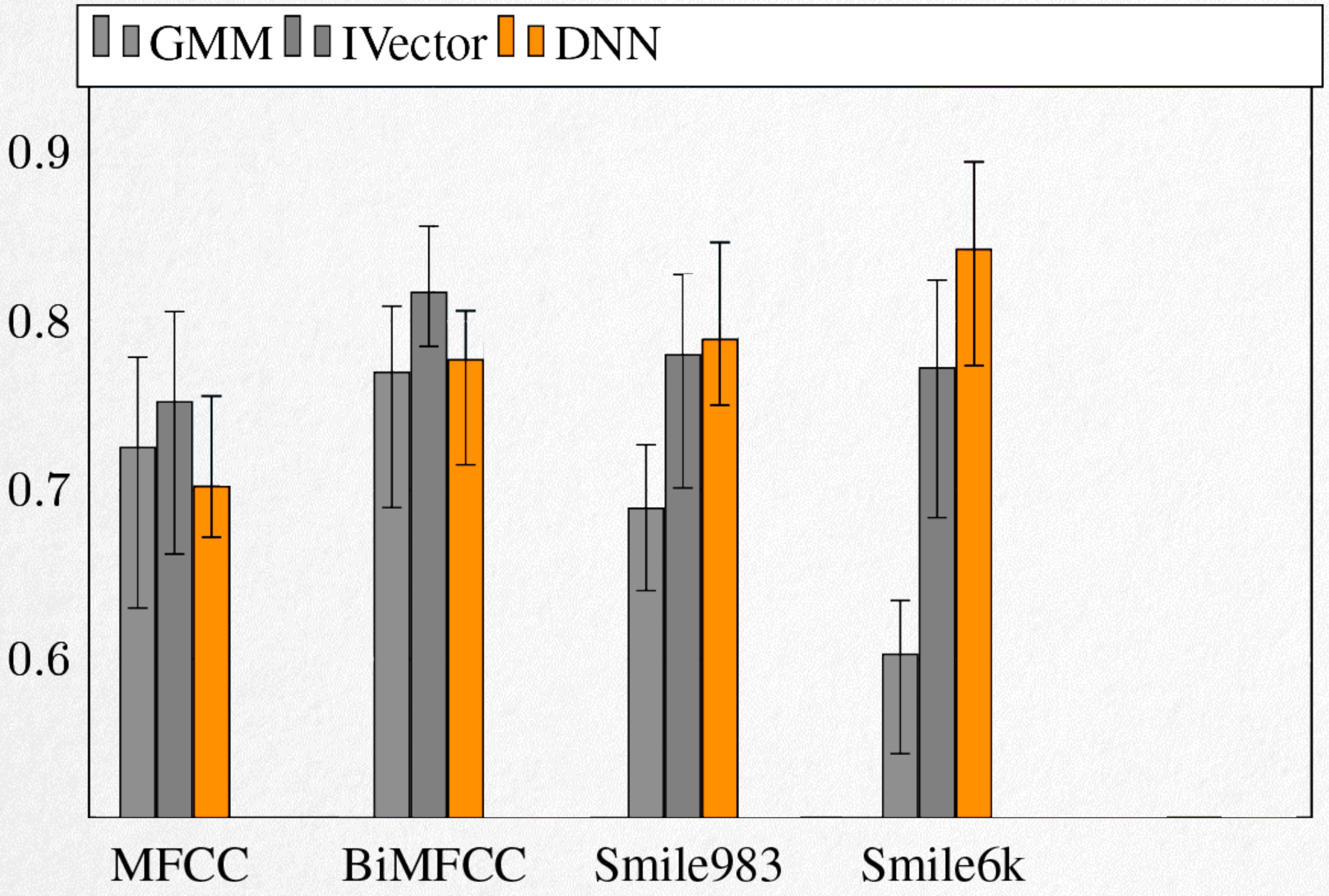
Model Specifications

DNN Input
Dense 256
BN + Dropout 0.2
Dense 256
BN + Dropout 0.2
Dense 256
BN + Dropout 0.2
Dense 256
BN + Dropout 0.2
Softmax

BN: Batch Normalization
ReLu: Rectified Linear Activation Function

DEEP NEURAL NETWORK (DNN)

4- fold CV avg. accuracy



Better Performance with Larger Features

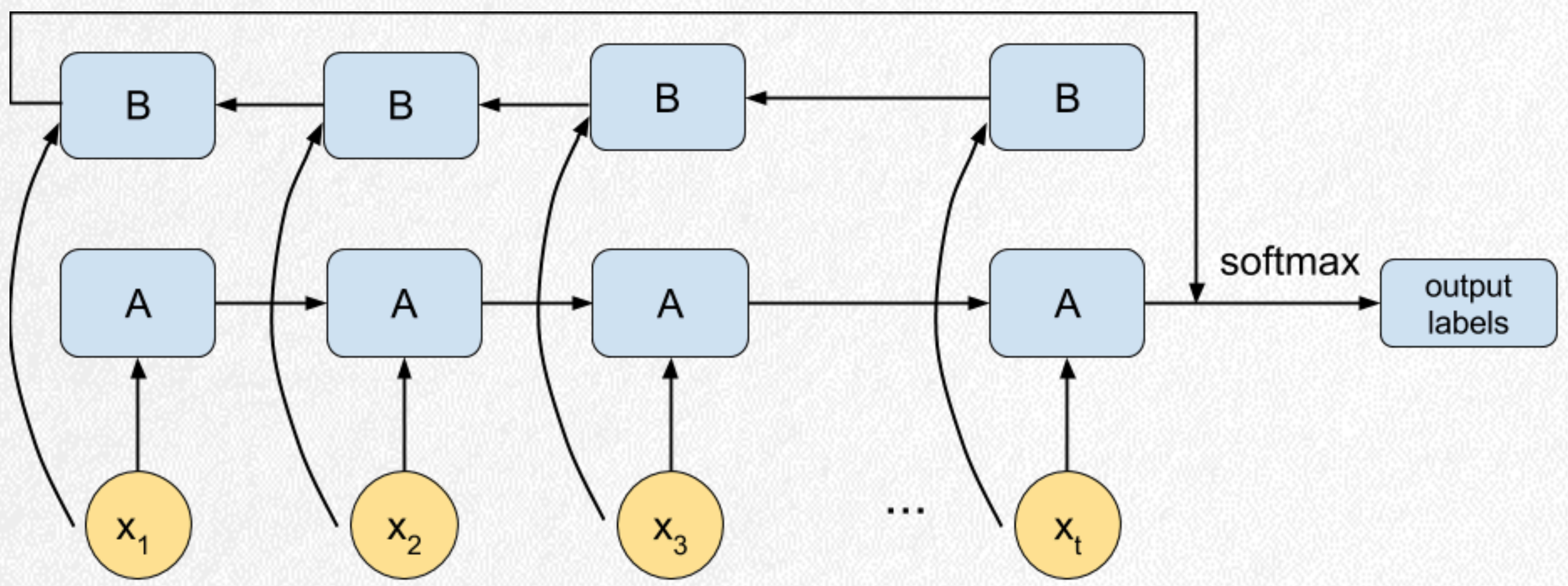
MFCC / BiMFCC:
12 layers / 1.1M params

Smile983:
10 layers / 1M params

Smile6k:
16 layers / 4.4M params

RECURRENT NEURAL NETWORK (RNN)

- Use Gated Recurrent Units (GRU)
  - Performs similarly to Long-Short Term Memory (LSTM) but faster
- Bi-directional RNN: Long-range context in both input directions



$$\boldsymbol{r}_t = \sigma(W_{xr}\boldsymbol{x}_t + W_{hr}h_{t-1} + \boldsymbol{b}_r)$$
$$\boldsymbol{z}_t = \sigma(W_{xz}\boldsymbol{x}_t + W_{hz}h_{t-1} + \boldsymbol{b}_z)$$
$$\tilde{\boldsymbol{h}}_t = \tanh(W_{xh}\boldsymbol{x}_t + W_{hh}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \boldsymbol{b}_h)$$
$$\boldsymbol{h}_t = (1 - \boldsymbol{z}_t)\boldsymbol{h}_{t-1} + \boldsymbol{z}_t\tilde{\boldsymbol{h}}_t.$$

RNN Pipeline

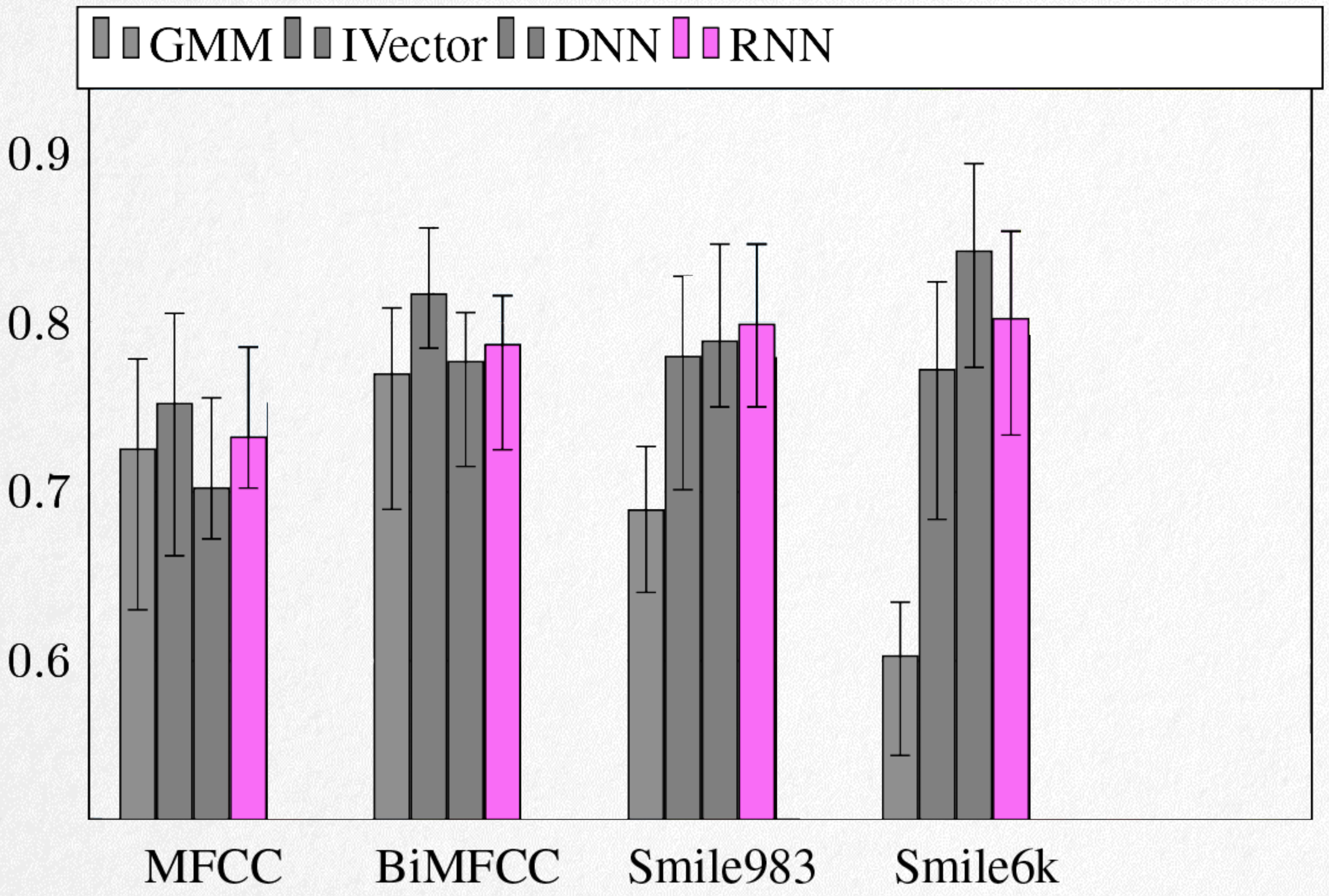
Model Specifications

RNN Input
GRU 512 forward
GRU 512 backward
Dropout 0.4
BN
Softmax

BN: Batch Normalization
ReLu: Rectified Linear Activation Function

RECURRENT NEURAL NETWORK (RNN)

4- fold CV avg. accuracy



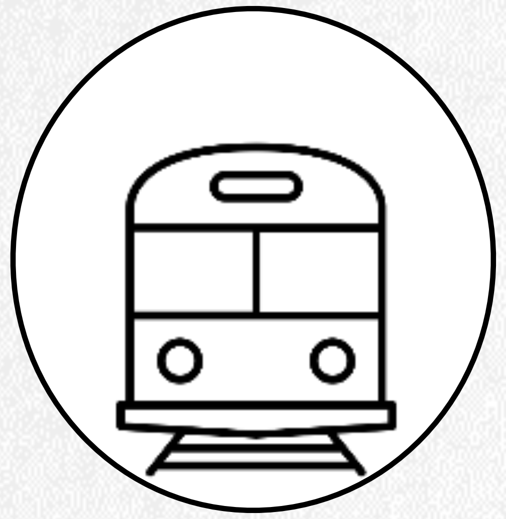
Better Performance with Larger Features

MFCC / BiMFCC:
4 layers / 50k params

Smile983:
4 layers / 4.6M params

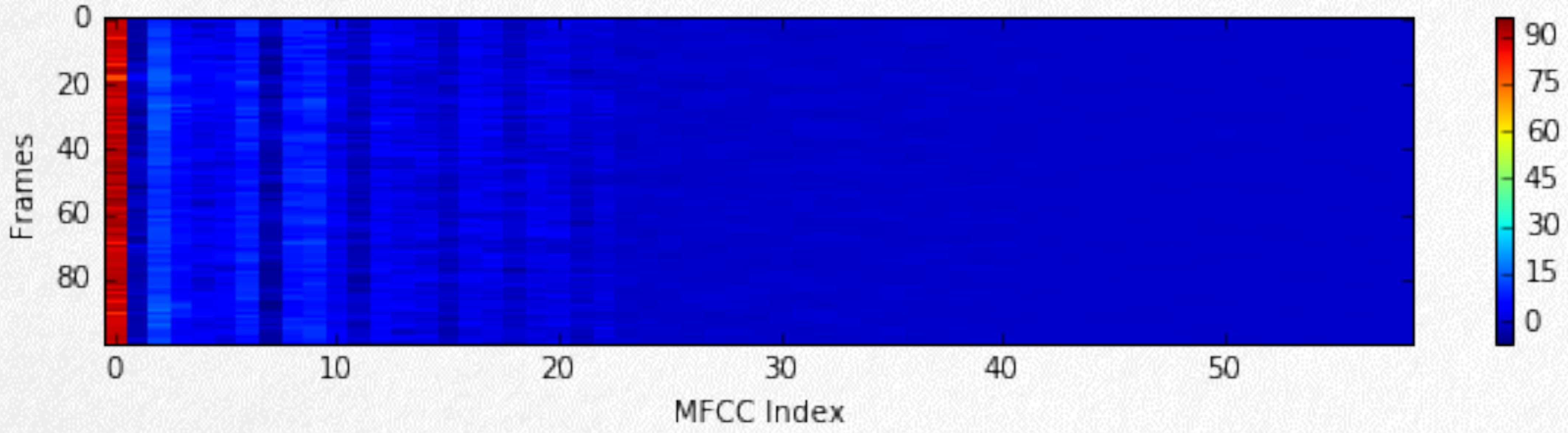
Smile6k:
4 layers / 26.8M params

RECURRENT NEURAL NETWORK (RNN)
SOME OBSERVATIONS

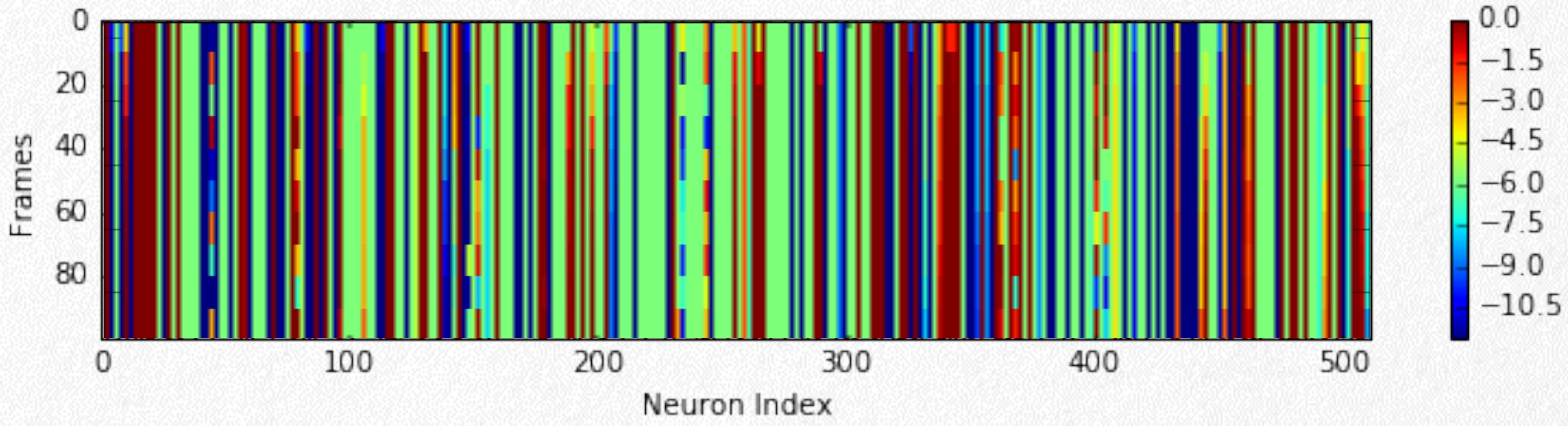


TRAIN

- Train Audio Example:
  - Not enough variation in the audio signal
- RNN may work better on event-rich audio scenes

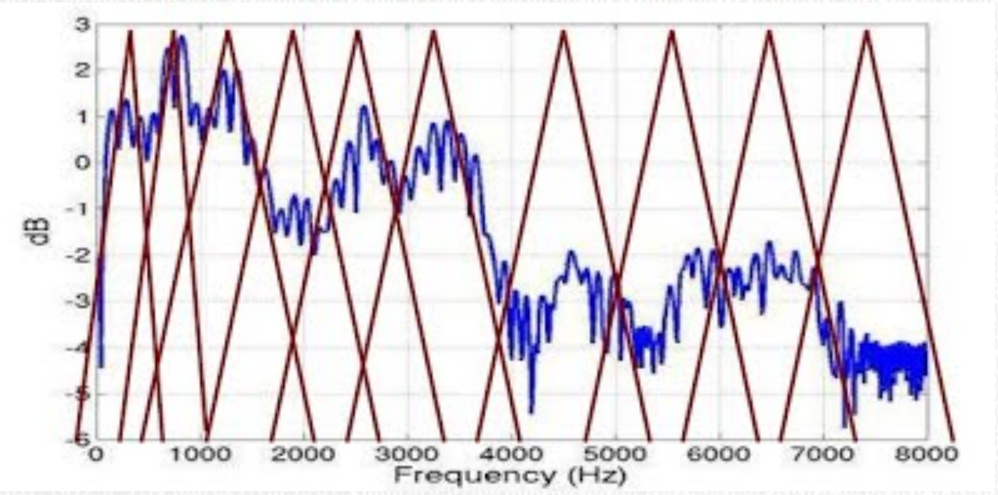


BiMFCC (61- dim) over 100 frames

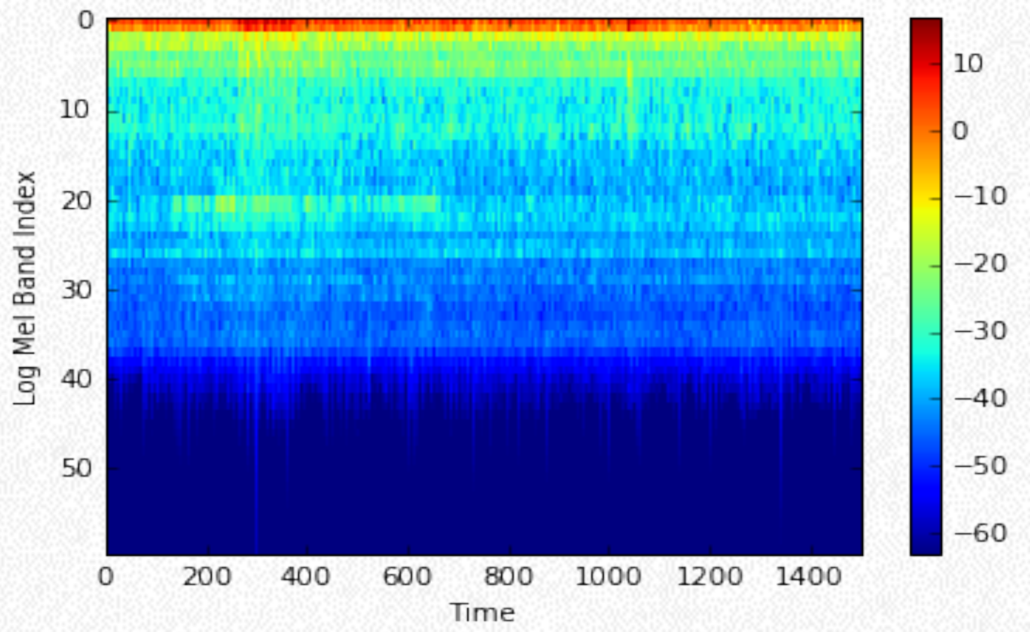


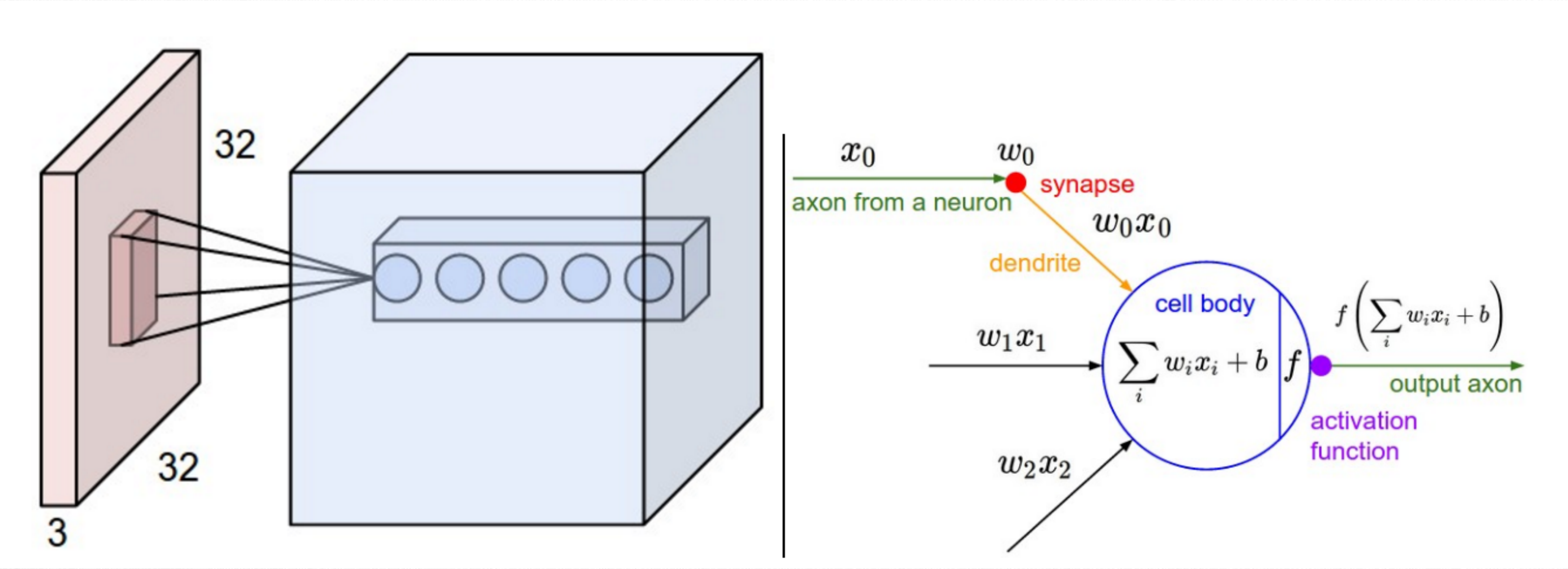
RNN Neuron (512- dim) Activation

CONVOLUTIONAL NEURAL NETWORK (CNN)



Log Mel-spectrum





CNN Pipeline

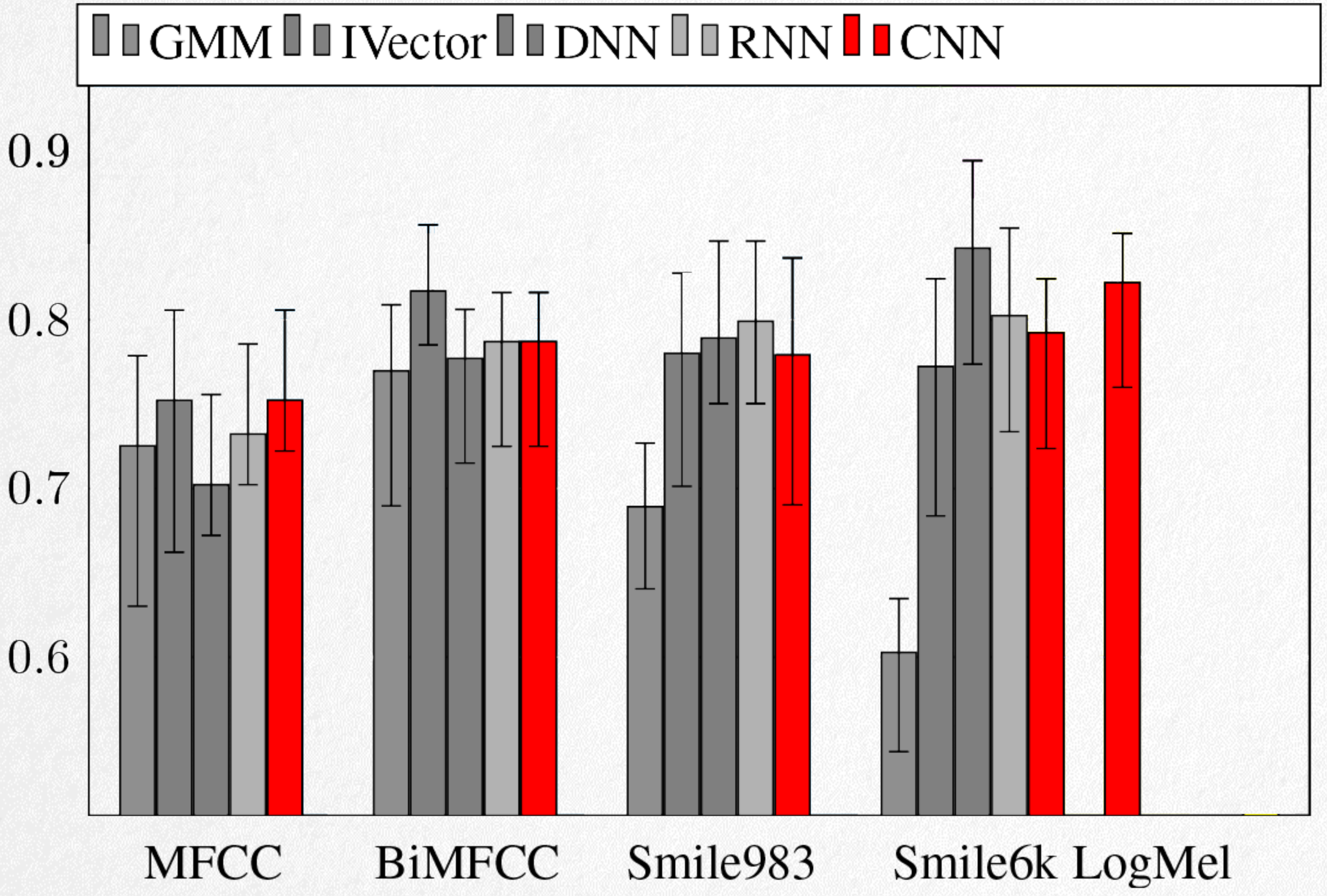
Model Specifications

CNN Input
32×3×3-BN-ReLu
32×3×3-BN-ReLu
MaxPool2×2+Dropout0.3
64×3×3-BN-ReLu
64×3×3-BN-ReLu
MaxPool2×2+Dropout0.3
128×3×3-BN-ReLu
128×3×3-BN-ReLu
MaxPool2×2+Dropout0.3
Softmax

BN: Batch Normalization
ReLu: Rectified Linear Activation Function

CONVOLUTIONAL NEURAL NETWORK (CNN)

4- fold CV avg. accuracy



Better Performance with Larger Features

MFCC / BiMFCC:
12 layers / 1.6M params

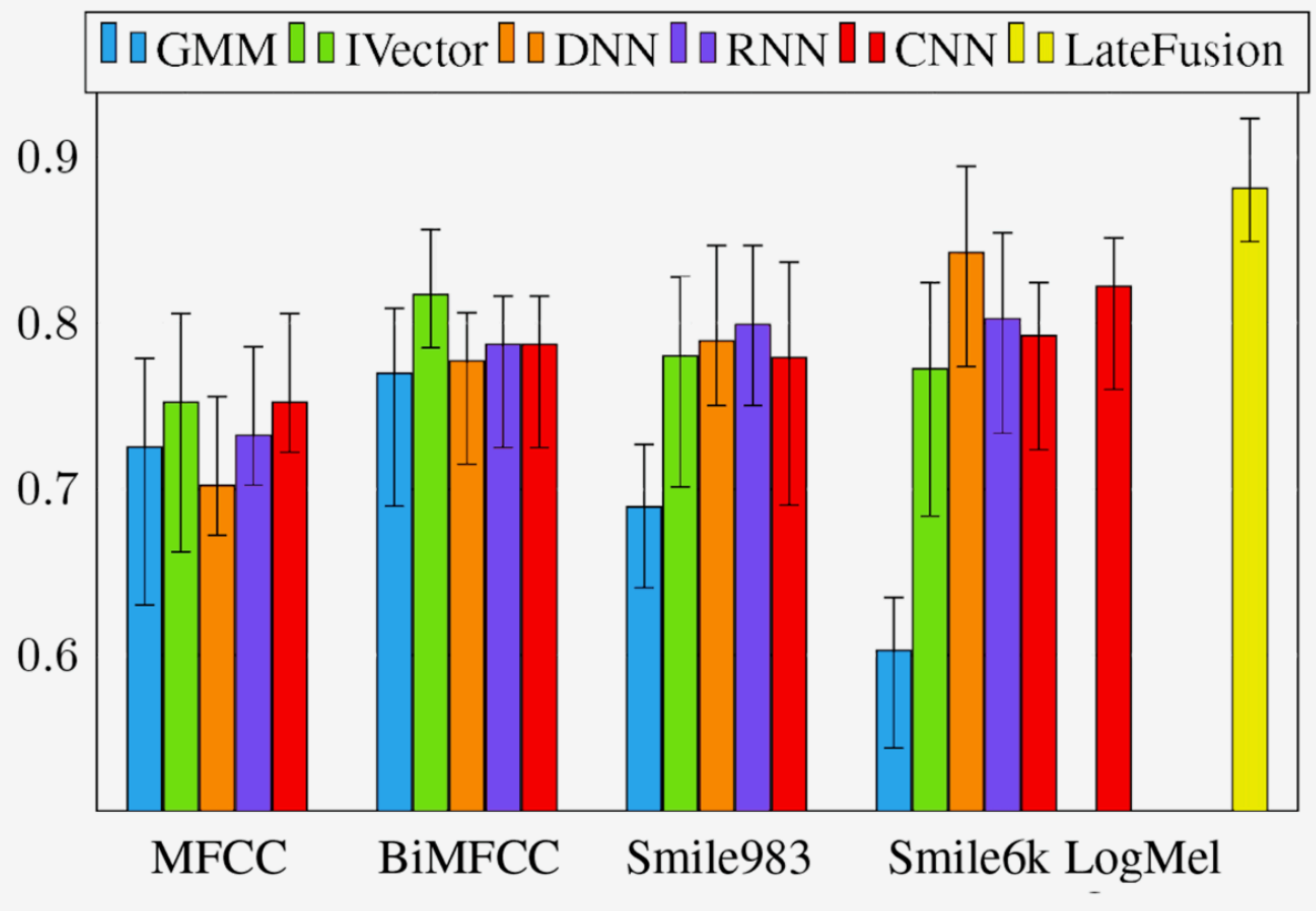
Smile983 / Smile6k:
12 layers / 2.6M params

LogMel:
12 layers / 3.6M params

MODEL ENSEMBLING

- Weighted averaging or voting of a **collection** of models
- Member models must be **accurate** and **diverse**
- Ensembling reaches **88.2%**

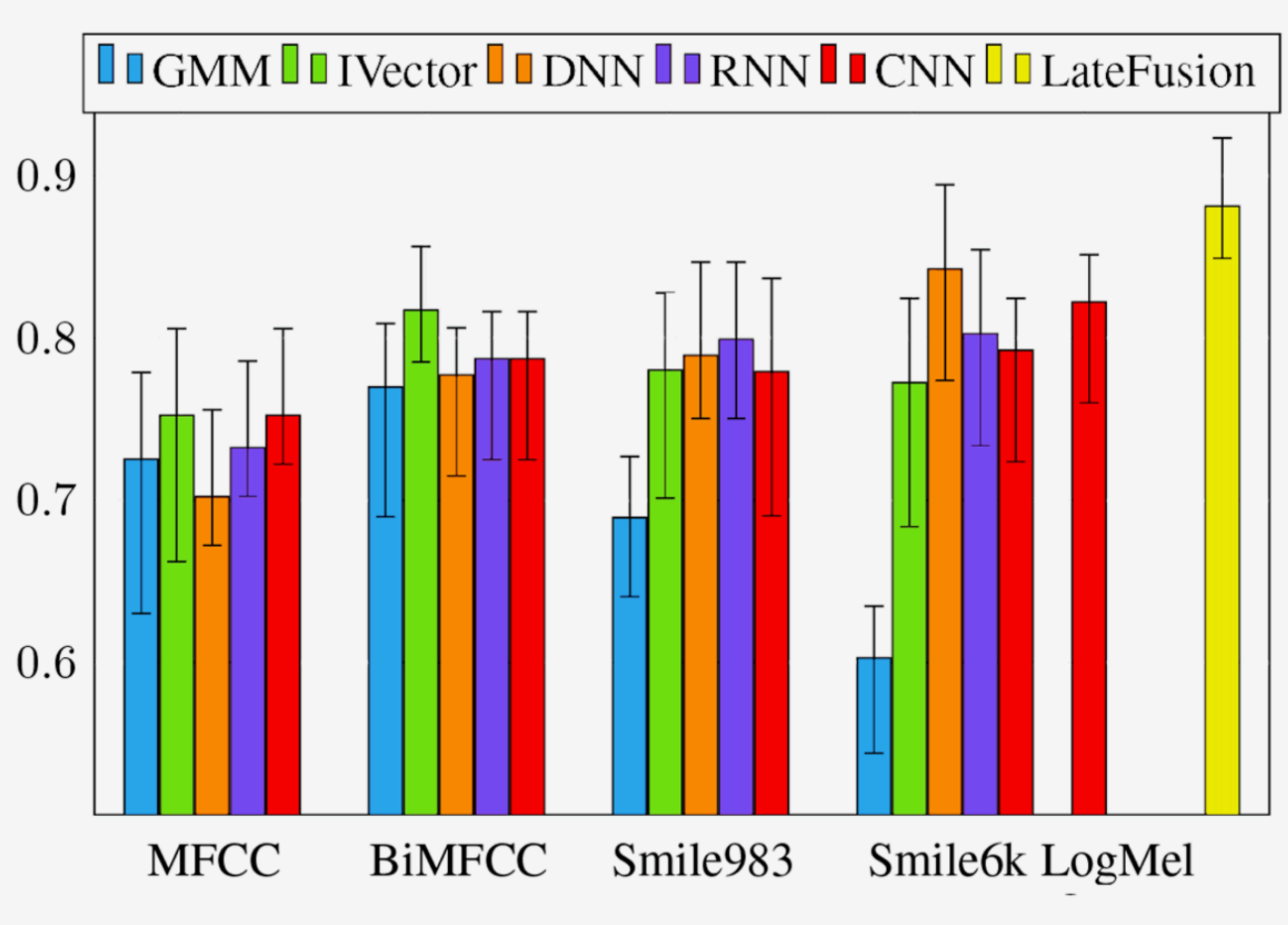
4- fold CV avg. accuracy



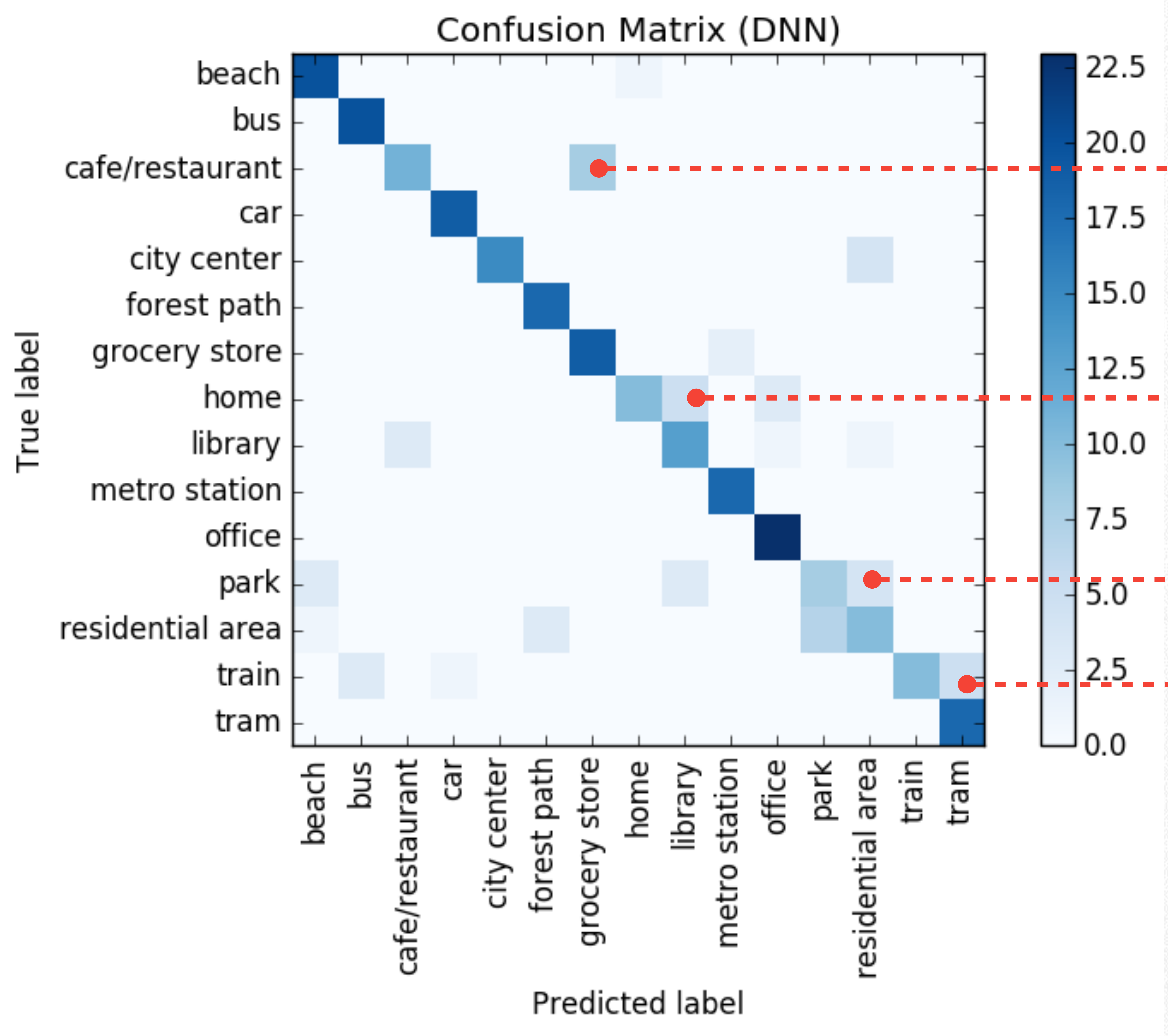
COMPARISON OF MODELS

- For neural network models (CNN, DNN, RNN), **larger** feature set produces **higher** accuracy
- RNN do not outperform DNN for Smile6k feature, showing that **temporal dynamics** is relatively weak
- RNN, CNN outperforms DNN on **smaller features** (MFCC, Smile983), as sequence input implicitly enhances feature complexity

4- fold CV avg. accuracy



DISCUSSION



café ~ grocery store

home ~ library

park ~ residential area

train ~ tram

CLASS WISE ACCURACY

	GMM	I-Vector	DNN	RNN	CNN	Fusion
Beach	69.3	80.7	89.8	80.3	78.7	92.3
Bus	79.6	82.4	95.3	88.6	72.1	95.3
Cafe/Rest.	83.2	70.0	69.9	64.7	66.4	79.9
Car	87.2	96.1	87.2	88.8	99.1	97.2
City	85.5	90.0	97.3	96.2	93.5	89.2
Forest	81.0	92.0	96.4	95.0	99.8	99.8
Grocery	65.0	93.8	79.3	75.5	85.3	96.2
Home	82.1	65.2	84.8	75.7	82.9	88.2
Library	50.4	76.1	81.2	81.6	72.7	86.2
Metro	94.7	83.5	97.3	93.7	98.7	92.3
Office	98.6	93.1	99.7	79.6	97.6	99.7
Park	13.9	78.6	49.4	45.8	45.7	71.2
Resident	77.7	66.5	76.9	68.7	81.6	77.0
Train	33.6	72.4	51.1	61.2	59.2	65.2
Tram	85.4	84.6	97.0	90.7	91.7	92.2
Average	72.5	81.7	84.2	80.2	82.2	88.1

Class-wise accuracy (%) of the best CV average models. Colored rows correspond to the most challenging classes in the confusion matrix from

CONCLUSION

- Feature extraction is **key**
- Deep learning models **>** traditional ones (GMM, i-vector)
- Environmental sound has **weak temporal** dynamics (DNN > recurrent networks)
- CNN, RNN don't do well (**not enough data** to learn better features than signal processing features)
- Ongoing work: Transfer Learning, Attention model, Raw Wave Input

Thank you for listening

A COMPARISON OF DEEP LEARNING METHODS
FOR ENVIRONMENTAL SOUND DETECTION

By **JUNCHENG (BILLY) LI**

BACK – UP SLIDES

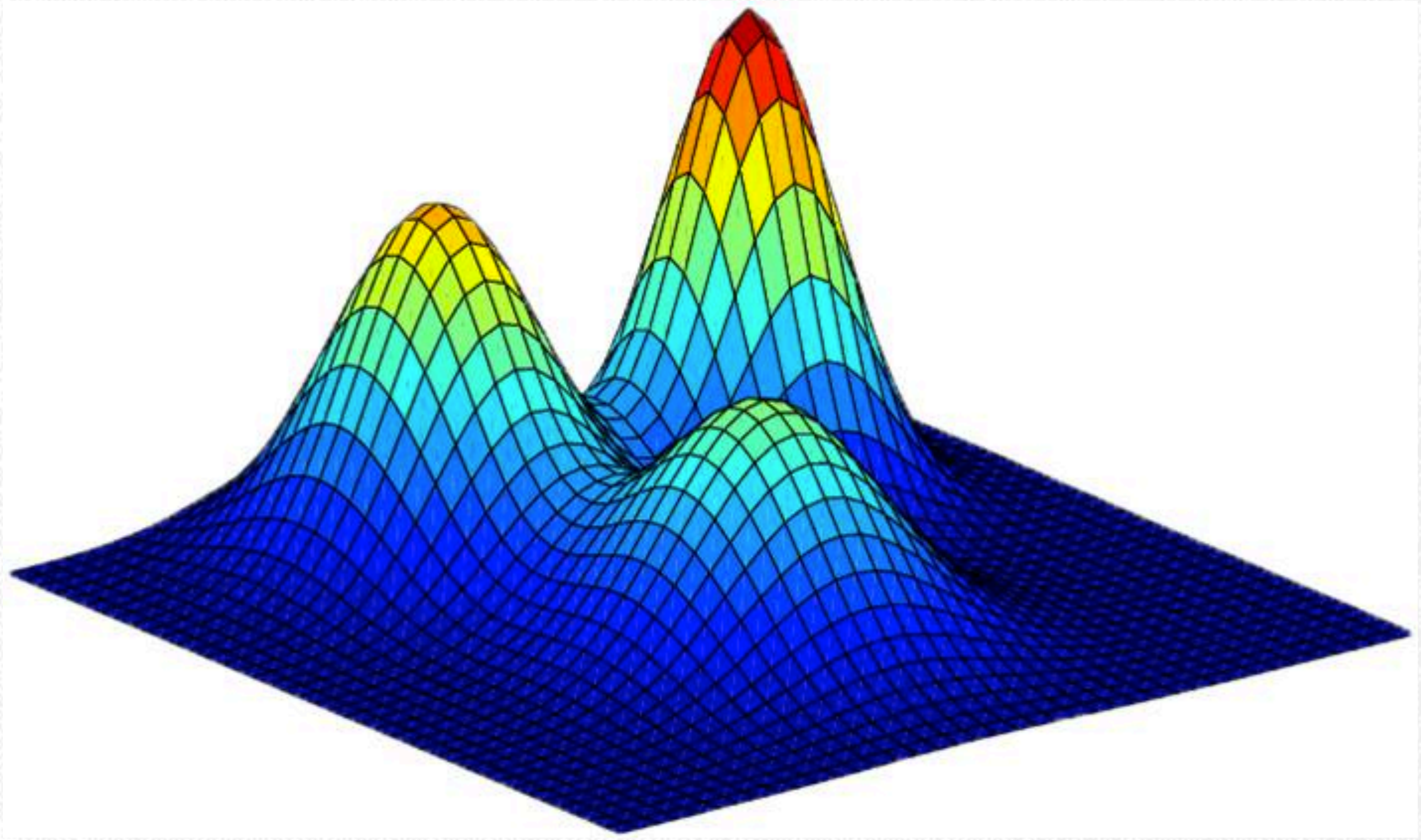
GAUSSIAN MIXTURE MODEL (GMM)

- Previous state-of-art speech & acoustic modeling
- Model each class with mixture of Gaussians. The probability for class j is

$$p^j(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k^j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k^j, \boldsymbol{\Sigma}_k^j)$$

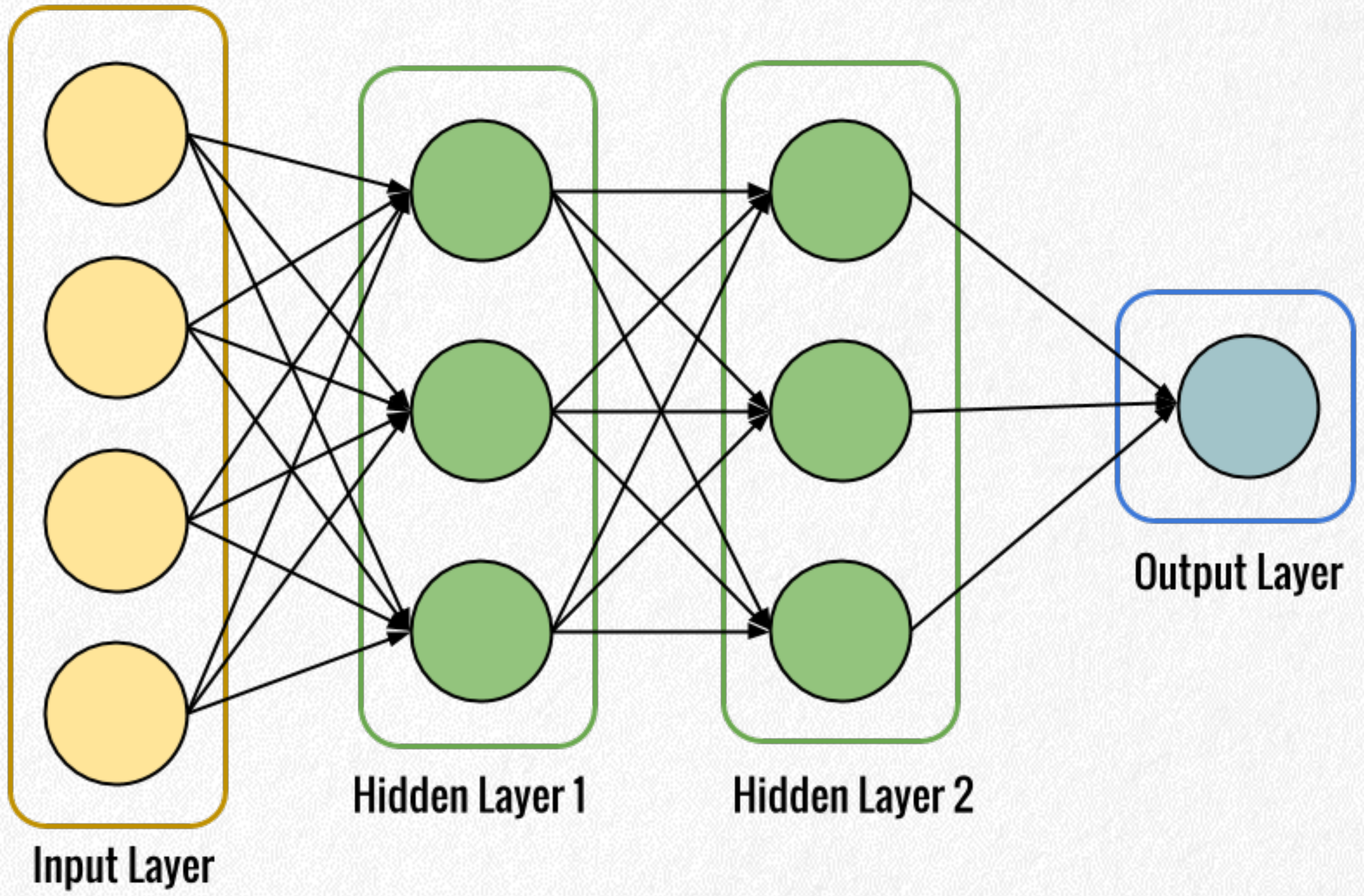
- Prediction sums over all audio segments, the pick the most likely class

$$\arg\max_j (\hat{p}^j = \sum_i p_i^j)$$



DEEP NEURAL NETWORK (DNN)

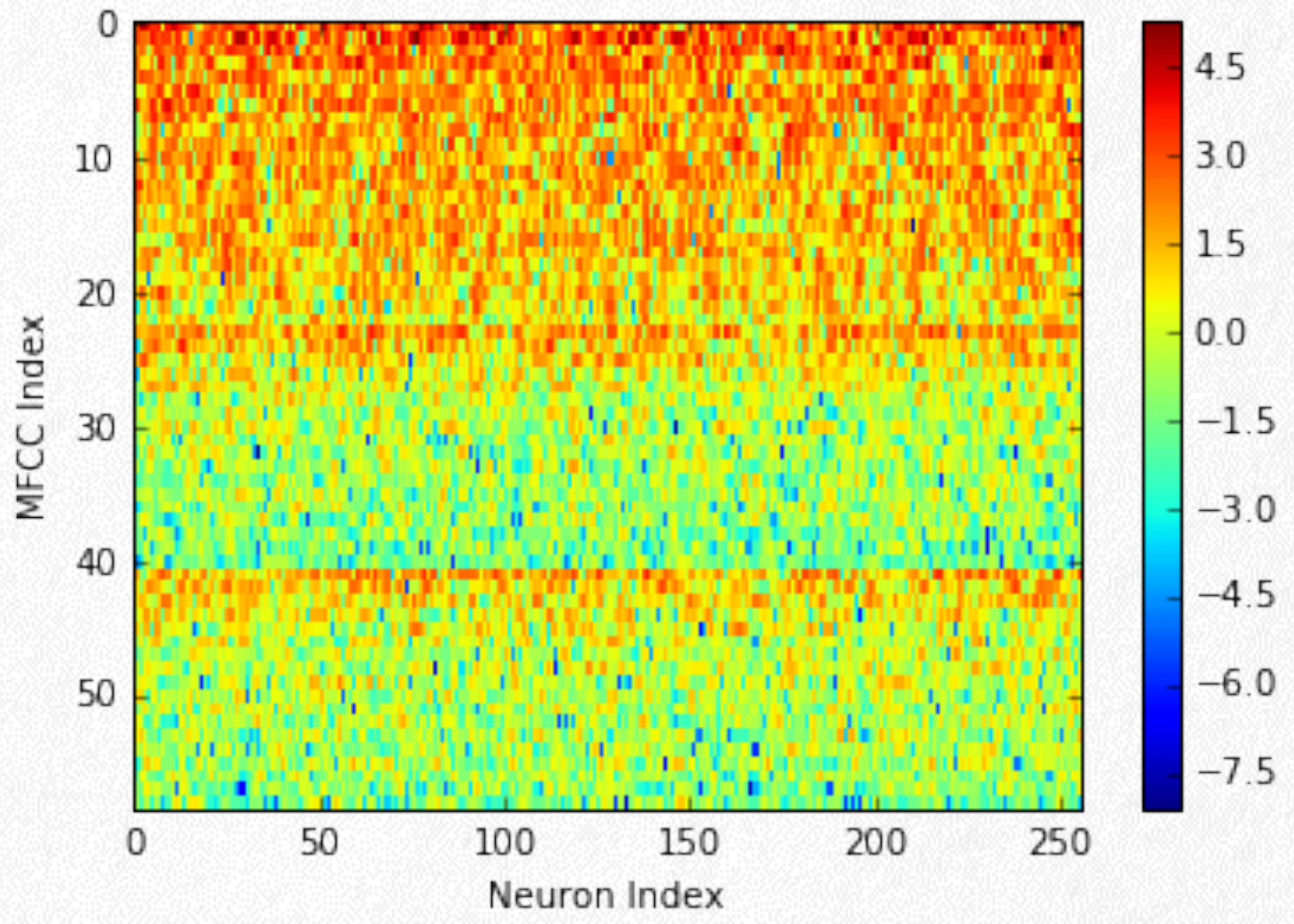
- **Each node ("neuron") introduces non-linearity**
- **Each layer introduces non-linearity**
- **Architectural choice:**
  - Types of neuron (which function to use)(relu, prelu...)
  - Number of layers (3,5,10, 12...)
  - Number of neurons (256, 512 ...)
  - Dropout (0-1)
  - Batch normalization
  - Optimizer (RMSprop, adadelta, SGD)



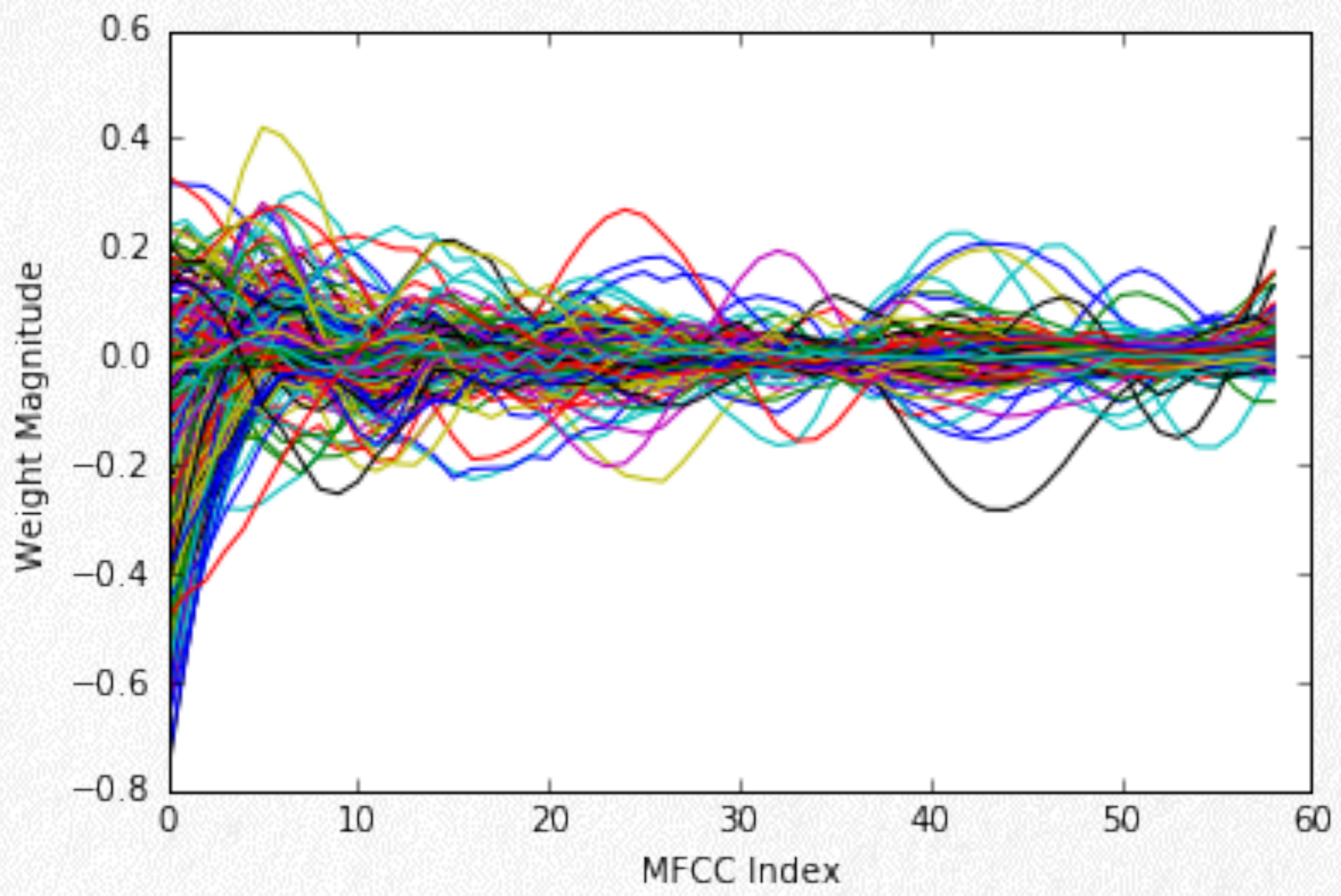
DEEP NEURAL NETWORK (DNN)
SOME OBSERVATIONS

- DNN's neurons are more **active** in the MFCC range (0-23) and are less active in the delta of MFCC (24-41) and double delta dimension (42-61).

- If we apply Savitzky-Golay **smoothing** function [24] which acts like a low-pass filter on each neuron's vector (61-dim). We get Figure2(b) which is the de-noised weight of layer (each colored line corresponds with one neuron vector), which looks like a filter bank.



DNN's 1st layer Weight after FFT

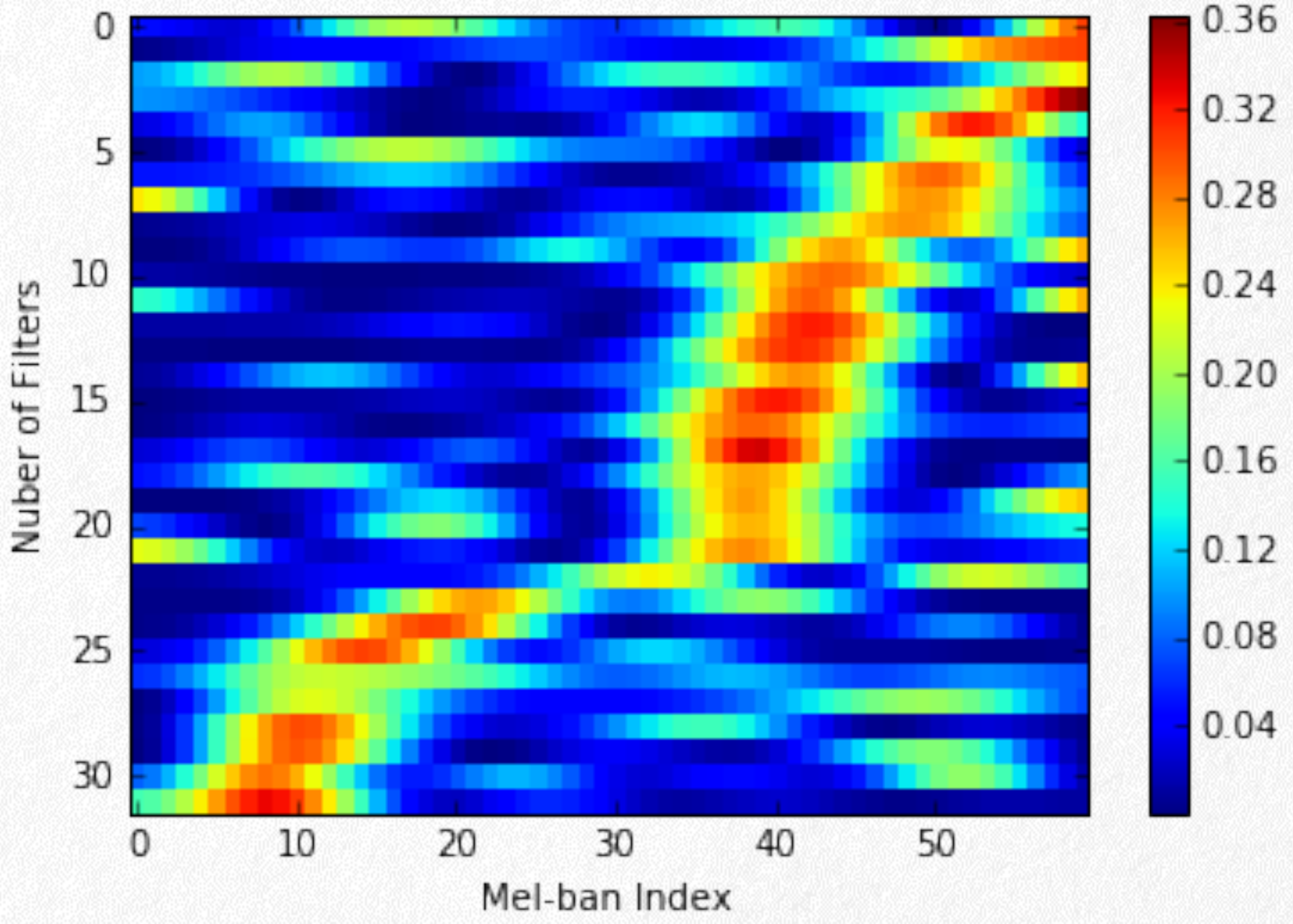


DNN's 1st layer Weight after Smoothing

CONVOLUTIONAL NEURAL NETWORK (CNN)

SOME OBSERVATIONS

This highly resembles a filter bank of bandpass filters. We notice there is **a sharp transition** in filters at around the **40th** Mel band. This is due to the **weak energy** beyond the 40th Mel band shown in Figure 5(a). Our finding is consistent with prior work on speech data [26]. The filter bank we learned are relatively **wider** compared with that is learned in speech.



CNN 1st Convolutional2D layer's Weight after FFT