# End-to-end Language Identification using NetFV and NetVLAD


*Jinkun Chen*[2], *Weicheng Cai*[1,2], *Danwei Cai*[1], *Zexin Cai*[1], *Haibin Zhong*[3], *Ming Li*[1]

[1]Data Science Research Center, Duke Kunshan University, Kunshan, China

[2]School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China

[3]Jiangsu Jinling Science and Technology Group Limited, Nanjing, China

ming.li369@dukekunshan.edu.cn


## Introduction

- Language identification (LID) is a kind of utterance-level paralinguistic speech attribute recognition task with variable-length speeches.
- It is important to find an effective and robust method to retrieve the utterance-level information and encode them into fixed dimensional vector representations.
- Conventional encoding methods:
  - Vector quantization (VQ)
  - Universal Background Model (UBM)
  - Pre-trained phoneme decoder (DNN)
  - Average pooling
- End-to-end learnable encoding methods:
  - Learnable dictionary encoding (LDE)
  - NetFV and NetVLAD (Our works)

## Methods

- End-to-end learnable encoding layers
  a. Variable-length input sequence $D\times L$ → fixed-length representation $D\times C$
  b. All parameters are learnable
  c. Proposed encoding methods for LID: NetFV and NetVLAD

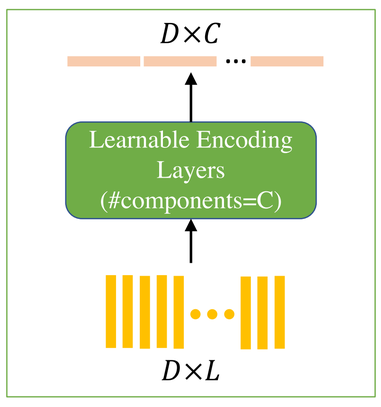


Fig. 1 Schematic diagram of encoding layer

- Implementation of the NetFV layer
  a. Derived from the standard Fisher Vector (FV) algorithm
  b. Suppose a $K$-components GMM $u_\lambda = \sum_{k=1}^{K} \alpha_k u_k(\boldsymbol{x}_i)$ is used in FV, the FV w.r.t. the mean and the standard deviation parameters:

$$\nabla\boldsymbol{\mu}_k \log u_\lambda(\boldsymbol{x}_i) = \frac{1}{\sqrt{\alpha_k}}\gamma_i(k)\left(\frac{\boldsymbol{x}_i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k}\right)$$

$$\nabla\boldsymbol{\sigma}_k \log u_\lambda(\boldsymbol{x}_i) = \frac{1}{\sqrt{2\alpha_k}}\gamma_i(k)\left[\frac{(\boldsymbol{x}_i-\boldsymbol{\mu}_k)^2 - 1}{\boldsymbol{\sigma}_k^2}\right]$$

  where the $\gamma_i(k)$ is the posterior probability of $\boldsymbol{x}_i$ on the $k^{th}$ GMM component.

  c. To make the FV definition equations differentiable, simplifications are introduced to the original FV:
     I. Assume all GMM components have equal weights.
     II. Simplify the Gaussian density $u_k(\boldsymbol{x}_i)$ to

$$u_k(\boldsymbol{x}_i) = \frac{1}{\sqrt{(2\pi)^D}}\exp\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)\}$$

  d. Let $\boldsymbol{w}_k = 1/\boldsymbol{\sigma}_k$ and $\boldsymbol{b}_k = -\boldsymbol{\mu}_k$, and assume that $\Sigma_k = diag(\boldsymbol{\sigma}_k^2)$. The final differential definition equations of the NetFV layer:

$$\nabla\boldsymbol{\mu}_k \log u_\lambda(\boldsymbol{x}_i) = \gamma_i(k)[\boldsymbol{w}_k \odot (\boldsymbol{x}_i + \boldsymbol{b}_k)]$$

$$\nabla\boldsymbol{\sigma}_k \log u_\lambda(\boldsymbol{x}_i) = \frac{1}{\sqrt{2}}\gamma_i(k)[(\boldsymbol{w}_k\odot (\boldsymbol{x}_i + \boldsymbol{b}_k))^2 - 1]$$

$$\gamma_i(k) = \frac{\exp\{-\frac{1}{2}[\boldsymbol{w}_k\odot(\boldsymbol{x}_i+\boldsymbol{b}_k)]^T[\boldsymbol{w}_k\odot(\boldsymbol{x}_i+\boldsymbol{b}_k)]\}}{\sum_{c=1}^{K}\exp\{-\frac{1}{2}[\boldsymbol{w}_c\odot(\boldsymbol{x}_i+\boldsymbol{b}_c)]^T[\boldsymbol{w}_c\odot(\boldsymbol{x}_i+\boldsymbol{b}_c)]\}}$$

  e. The parameters set of the NetFV layer is $\lambda = \{\boldsymbol{w}_k, \boldsymbol{b}_k\}$.

- Implementation of the NetVLAD layer
  a. VLAD is another strategy used to aggregate a set of feature descriptors into a fixed-size representation.
  b. The conventional definition of the VLAD is

$$\boldsymbol{V}(k) = \sum_{i=1}^{L}\beta_k(\boldsymbol{x}_i)\,(\boldsymbol{x}_i - \boldsymbol{\mu}_k)$$

  where $\boldsymbol{V} \in R^{K\times D}$, $\beta_k(\boldsymbol{x}_i)$ is the "hard" alignment of $\boldsymbol{x}_i$ to the cluster $\boldsymbol{\mu}_k$.

  c. Change the hard alignment into the soft alignment, i.e.

$$\beta_k(\boldsymbol{x}_i) = \frac{\exp(\boldsymbol{w}_k\boldsymbol{x}_i^T + b_k)}{\sum_{c=1}^{K}\exp(\boldsymbol{w}_c\boldsymbol{x}_i^T + b_c)}$$

  d. The final differential definition equation of the NetFV layer is

$$\boldsymbol{V}(k) = \sum_{i=1}^{L}\frac{\exp(\boldsymbol{w}_k\boldsymbol{x}_i^T + b_k)}{\sum_{c=1}^{K}\exp(\boldsymbol{w}_c\boldsymbol{x}_i^T + b_c)}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)$$

  e. The parameters set of the NetVLAD layer is $\lambda = \{\boldsymbol{\mu}_k, \boldsymbol{w}_k, b_k\}$.

## End-to-end LID Framework

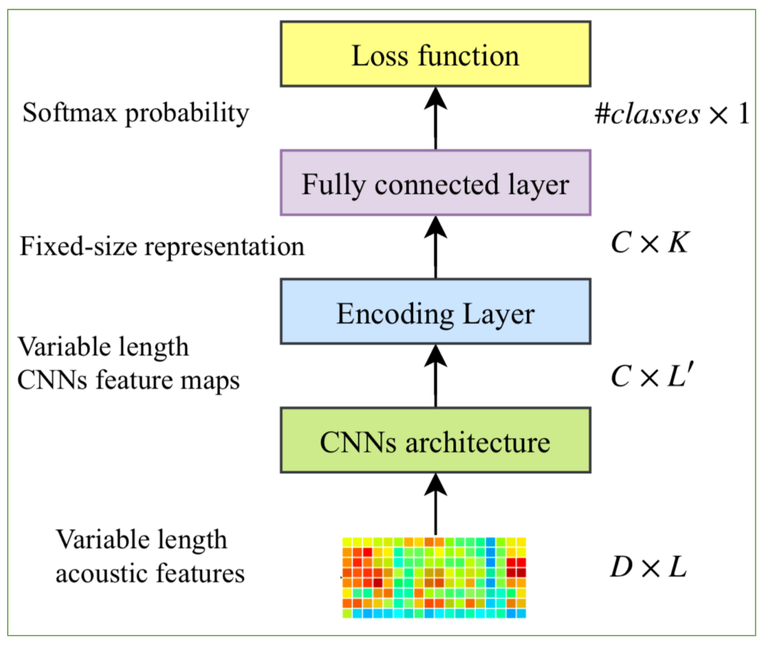


Fig. 2 Schematic diagram of end-to-end LRE framework

- The CNNs architecture acts as a feature extractor.
- The NetFV and NetVLAD are adopted in the encoding layer.
  I. Capture the zero-order and first-order statistics.
  II. Superior to the temporal average pooling (TAP) layer
- All the parameters are learnable. Train the LID system under the end-to-end principle.
- Support variable-length utterances as input during both training and testing phases.

## Experimental results

- The LRE07 closed-set language detection task:
  Totally 14 target languages.
  Training set: 39000 utterances.
  Test set: 7530 utterances.
  Three nominal durations: 3, 10 and 30 seconds.

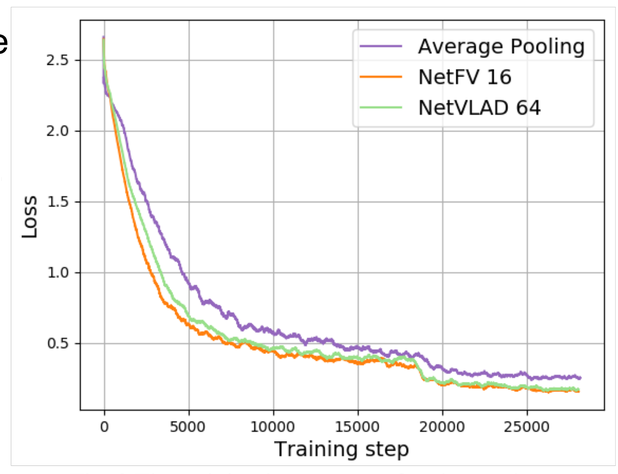


Fig. 3 The training loss curves of end-to-end systems

- Experimental setup
  a. ResNet-34 (3-4-6-3 stacked blocks, 128 output channels)
  b. K-clusters (K ranges from 16 to 128) in encoding layer
  c. SGD optimizer with (momentum 0.9 and weight decay $10^{-4}$)
  d. 90 epochs in training. Change the learning rate at the epoch 60 and 80.
  e. Train and test with the same model on the 3s, 10s and 30s utterances.

- Both the NetFV and NetVLAD based LID systems outperform the GMM i-vector or the TAP based systems.
- The TAP based baseline system achieves the accuracies of 75.49%, 89.71% and 93.56% on the 3s, 10s and 30s test set respectively, while the NetVLAD based system improves the accuracies to 76.14%, 91.43% and 96.85%.
- Overall, NetVLAD is slightly superior to NetFV in the test phase and achieves the best performance when the cluster size is 64.
- The fusion system improves the $C_{avg}$ and the EER metrics further.

Table 1: *Performances on 2007 NIST LRE task*

| System description | $C_{avg}(\%)/EER(\%)$ | | |
|---|---|---|---|
| | 3s | 10s | 30s |
| GMM i-vector | 20.46/17.71 | 8.29/7.00 | 3.02/2.27 |
| ResNet34 TAP | 9.24/10.91 | 3.39/5.58 | 1.83/3.64 |
| ResNet34 LDE 64 | **8.25/7.75** | **2.61/2.31** | **1.13/0.96** |
| ResNet34 NetFV 16 | 9.47/9.04 | 2.96/2.59 | 1.31/1.08 |
| ResNet34 NetFV 32 | 8.95/8.37 | 2.88/2.49 | 1.35/1.31 |
| ResNet34 NetFV 64 | 8.91/8.26 | 2.88/2.74 | 1.19/1.15 |
| ResNet34 NetFV 128 | 9.05/8.64 | 2.91/2.72 | 1.27/1.34 |
| ResNet34 NetVLAD 16 | **8.23/8.06** | 2.90/2.62 | 1.36/1.17 |
| ResNet34 NetVLAD 32 | 8.87/8.58 | 3.10/2.50 | 1.46/1.15 |
| ResNet34 NetVLAD 64 | 8.59/8.08 | **2.80/2.50** | **1.32/1.02** |
| ResNet34 NetVLAD 128 | 8.72/8.44 | 3.15/2.76 | 1.53/1.14 |
| Fusion system | **6.14/6.86** | **1.81/2.00** | **0.89/0.92** |