Learning Local Feature Aggregation Functions with Backpropagation

Angelos Katharopoulos*, Despoina Paschalidou*,
Christos Diou, Anastasios Delopoulos

Multimedia Understanding Group, Electrical and Computer Engineering Department,
Aristotle University of Thessaloniki, Greece

Aristotle University of Thessaloniki

EUSIPCO2017

Motivation

Local Feature Aggregation Methods are used to **generate discriminative global representations** from local image/video features.

Existing local feature aggregation functions ignore the subsequent usage of the global feature representation!

Idea

- **Compose** local feature aggregation functions **with a classifier's cost function** and **back-propagate the gradient** to learn the parameters
- The resulting representations **outperform BOW, VLAD and Fisher Vectors**, with respect to classification accuracy, **by a large margin**

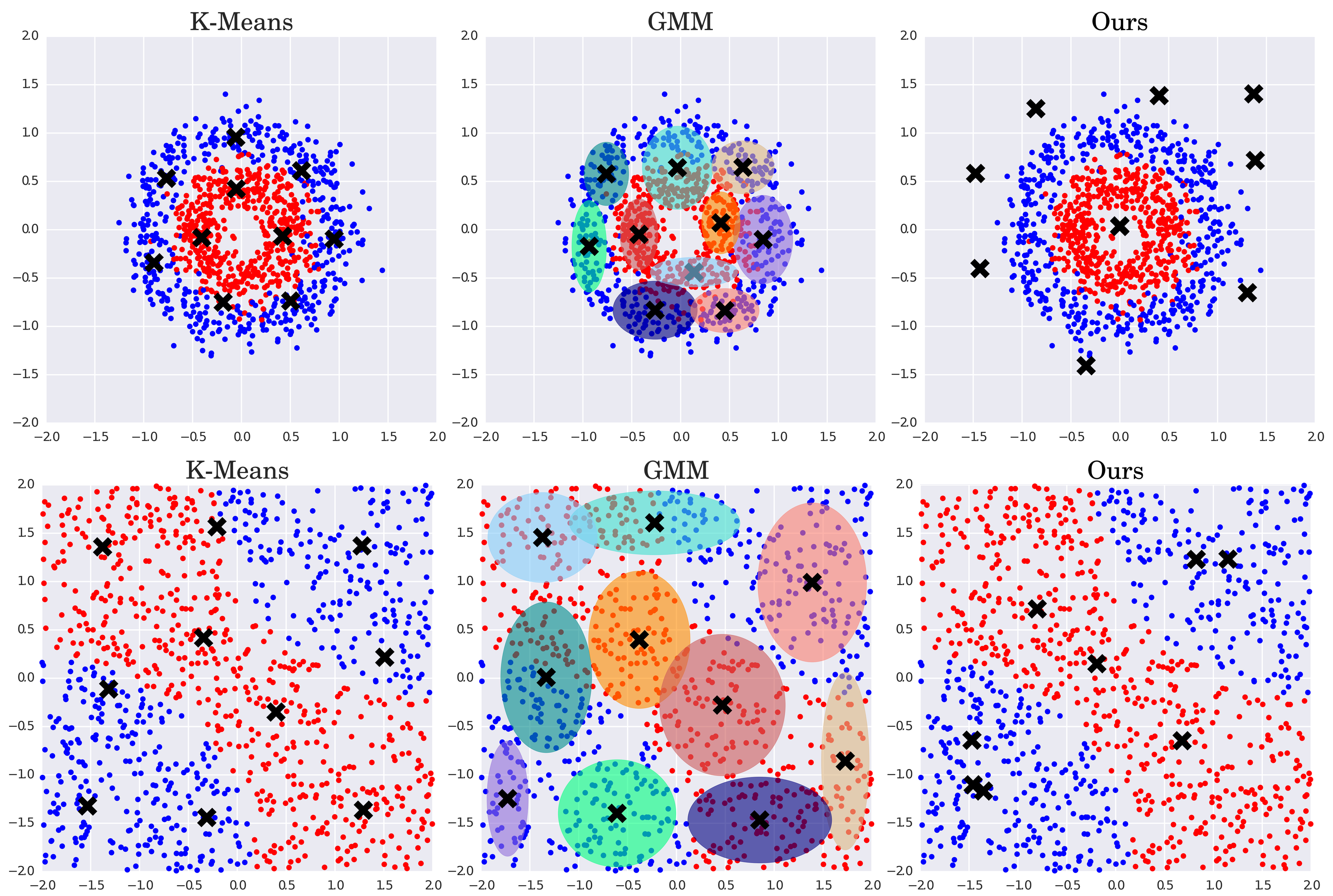

Figure 1: Codebooks learned by K-Means, GMM and our method (from left to right). Our method takes into account the classes and results into **separable global representations**

Aggregation Functions

We introduce a **family of local feature aggregation functions** that can be expressed as follows

$$R(F\,;\,\Theta) = \frac{1}{N_F}\sum_{n=1}^{N_F} T(f_n\,;\,\Theta)$$

where $T(\cdot\,;\,\Theta) : \mathbb{R}^D \to \mathbb{R}^K$ is a differentiable function with respect to the parameters Θ and $F = \{f_1, f_2, \dots, f_{N_F}\}$ is the set of N_F local descriptors extracted from an image or video.

We experiment with two local feature aggregation functions:

- **A generalized Soft-assignment Bag of Words (BOW)**

$$T_1(f_n\,;\,C,\Sigma) = \frac{1}{Z(f_n,C,\Sigma)}\begin{bmatrix} D(f_n,C_1,\Sigma_1) \\ \vdots \\ D(f_n,C_K,\Sigma_K)\end{bmatrix}$$

- **A Soft-assignment Vector of Locally Aggregated Descriptors (VLAD)**

$$T_2(f_n\,;\,C,\Sigma) = \frac{1}{Z(f_n,C,\Sigma)}\begin{bmatrix} D(f_n,C_1,\Sigma_1)(f_n - C_1) \\ \vdots \\ D(f_n,C_K,\Sigma_K)(f_n - C_K)\end{bmatrix}$$

where

$$D(f_n,C_k,\Sigma_k) = \exp\left(-\gamma\,(f_n - C_k)^T \Sigma_k^{-1}(f_n - C_k)\right)$$

and

$$Z(f_n,C,\Sigma) = \sum_{k=1}^{K} D(f_n,C_k,\Sigma_k)$$

C_k is the codebook and Σ_k is the diagonal covariance matrix used to compute the Mahalanobis distance between the n-th local feature and the k-th codeword.

Parameter Estimation

We jointly learn a classifier and a feature aggregation function by solving the following optimization problem, where $J(\cdot\,;\,W)$ is the cost function of **any classifier**.

$$W^*, \Theta^* = \underset{W,\Theta}{\arg\min} \sum_{i=1}^{N} J\left(R(F^{(i)}\,;\,\Theta), y^{(i)}\,;\,W\right)$$

Algorithm 1 Procedure to learn the parameters of a local feature aggregation function

procedure TRAINAGGFUN(F, y)
  if initialize with K-Means **then** // Parameter initialization
    $C_0 \leftarrow KMeans(F)$
    $\Sigma_0 \leftarrow \mathbf{I}$
  else
    $C_0, \Sigma_0 \leftarrow GMM(F)$
  end if
  $W_0 \leftarrow \arg\min_W \sum_{i=1}^{N} J\left(R(F^{(i)}\,;\,C_0,\Sigma_0), y^{(i)}\,;\,W\right)$

  $t \leftarrow 0$ // Core training
  repeat
    $i \sim \text{DiscreteUniform}(1, N)$
    Sample $\hat{F}^{(i)}$ from $F^{(i)}$
    $W_{t+1} \leftarrow \text{SGD}(\nabla_{W_t} J(R(\hat{F}^{(i)}\,;\,C_t,\Sigma_t), y^{(i)}\,;\,W_t))$
    $C_{t+1} \leftarrow \text{SGD}(\nabla_{C_t} J(R(\hat{F}^{(i)}\,;\,C_t,\Sigma_t), y^{(i)}\,;\,W_t))$
    $\Sigma_{t+1} \leftarrow \text{SGD}(\nabla_{\Sigma_t} J(R(\hat{F}^{(i)}\,;\,C_t,\Sigma_t), y^{(i)}\,;\,W_t))$
    $t \leftarrow t + 1$
  until $t \geq$ specific number of mini-batches

  $C^* \leftarrow C_t$ // Classifier fine tuning
  $\Sigma^* \leftarrow \Sigma_t$
  $W^* \leftarrow \arg\min_W \sum_{i=1}^{N} J\left(R(F^{(i)}\,;\,C^*,\Sigma^*), y^{(i)}\,;\,W_t\right)$
end procedure

Any classifier can be used, but we use **Logistic Regression** and an additional χ^2 feature map in the case of $T_1(\cdot)$.

Experimental Results

We conduct experiments on image and video datasets, **CIFAR10 and UCF-11** respectively, using state-of-the-art local features such as **Deep Convolutional Neural Networks** and **Improved Dense Trajectories** (IDT).

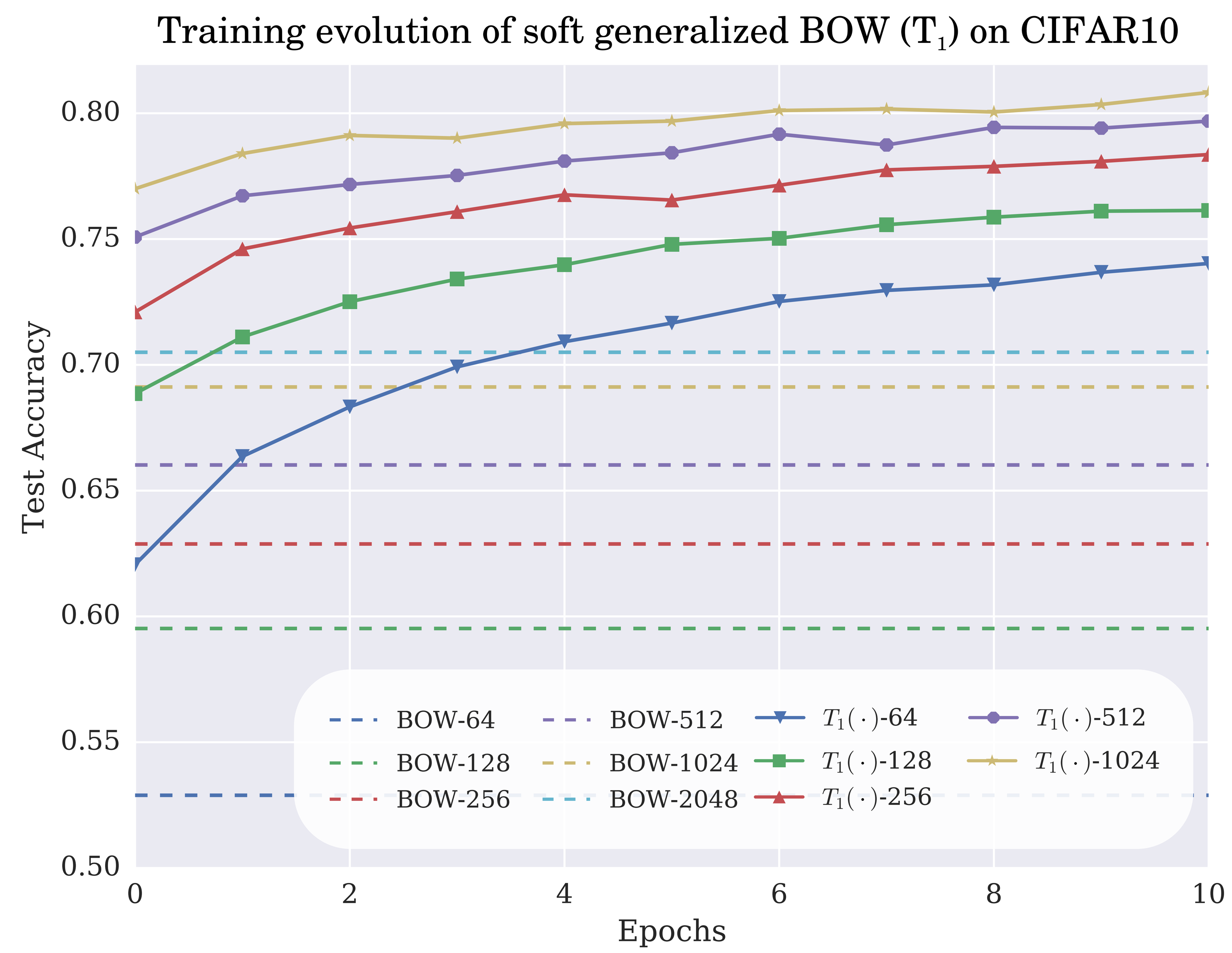

Figure 2: Classification accuracy on the test set with respect to the training epochs for various representation sizes on CIFAR10 with local features extracted from a conv net.

Learning a codebook using our method achieves 3.66% better classification accuracy **with 64 dimensions** than the codebook found with **K-Means with 2048 dimensions** in the CIFAR image classification benchmark.

Our method improves upon the state-of-the-art feature aggregation methods in **all** cases usually requiring a much smaller final representation.

Method-Codebook	*CIFAR10-DCNN*	*UCF11-IDT_HOF*	*UCF11-IDT_TRAJ*
BOW-1024	69.12%	89.72% $\pm$ 0.50	83.88% $\pm$ 0.39
BOW-2048	70.50%	91.03% $\pm$ 0.35	85.65% $\pm$ 0.53
T1-1024	80.87%	92.23% $\pm$ 0.37	86.90% $\pm$ 0.63
T1-2048	**81.12%**	**93.00%** $\pm$ 0.30	**87.01%** $\pm$ 0.48
VLAD-64	-	90.25% $\pm$ 0.33	78.71% $\pm$ 0.94
FV-64	-	90.55% $\pm$ 0.26	78.92% $\pm$ 0.21
T2-64	-	**91.08%** $\pm$ 0.26	**83.82%** $\pm$ 0.34

Table 1: Classification accuracy of Bag of Words (BOW), VLAD, Fisher Vectors (FV) and the two proposed aggregation methods $T_1(\cdot)$ and $T_2(\cdot)$