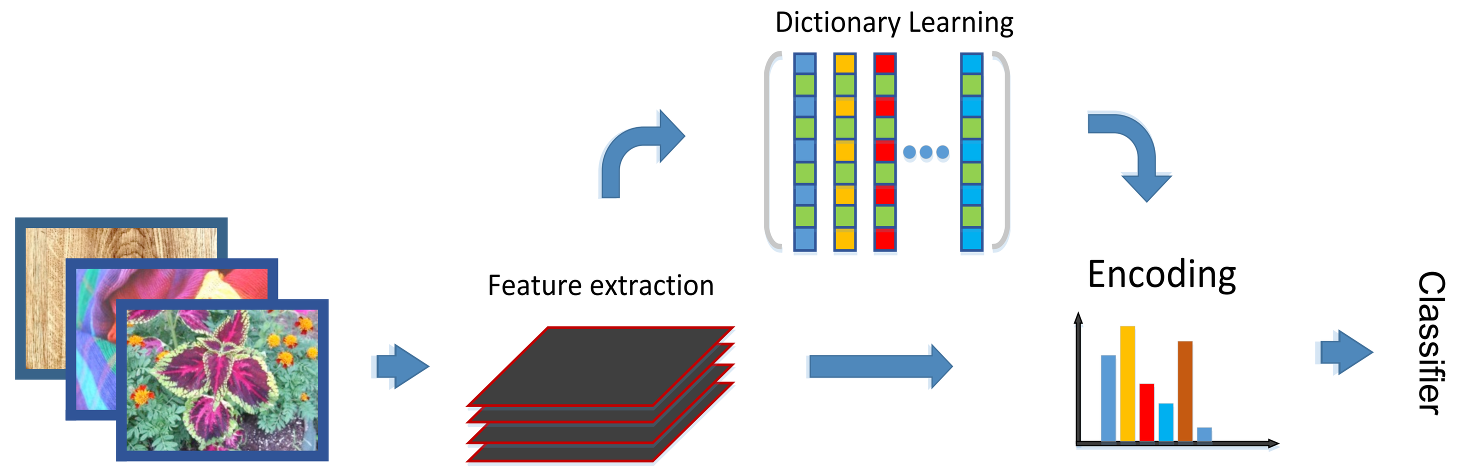
Deep TEN: Texture Encoding Network

Hang Zhang, Jia Xue, Kristin Dana
{zhang.hang, jia.xue}@rutgers.edu, kdana@ece.rutgers.edu, Department of Electrical and Computer Engineering, Rutgers University.

CVPR July 21-26 HONOLULU 2017

Overview:

- Encoding-Net (a new CNN architecture) with a novel Encoding Layer.
- State-of-the art results on texture recognition *(minc-2500, FMD, GTOS, 4D-light datasets).*
- Flexible deep learning framework (arbitrary image size and easy to transfer learned features).

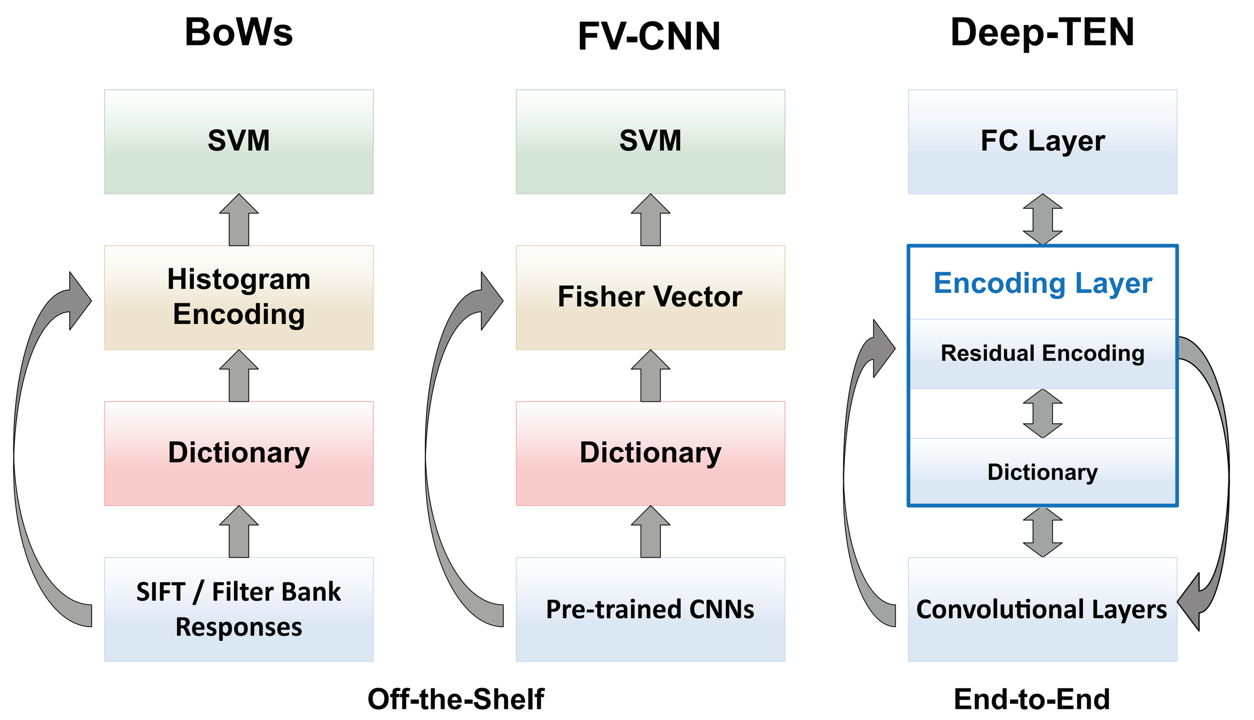


Classic Vision Approaches

- Flexible by allowing arbitrary input image size.
- No problem of domain transfer (features are generic).
- Dictionary encoding usually carries domain information.

Deep learning

- Preserving spatial information (texture needs orderless).
- Fixed image size.
- Difficulties in domain transfer.



Encoding Layer :

Residual Encoder

- Given a set of visual descriptors $X = \{x_1, ..x_N\}$ and a learned codebook $C = \{c_1, \dots c_K\}$.
- Each descriptor x_i can be assigned with a weight a_{ik} to each codeword c_k.
- The residual encoder aggregate the residuals with assignment weights

$$e_k = \sum_{i=1}^{N} e_{ik} = \sum_{i=1}^{N} a_{ik} r_{ik}$$

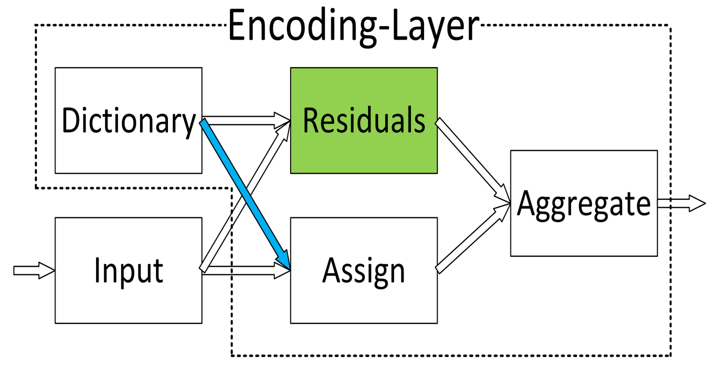
Assignment Weights:

- Soft-weighting

$$a_{ik} = \frac{\exp(-\beta \|r_{ik}\|^2)}{\sum_{j=1}^{K} \exp(-\beta \|r_{ij}\|^2)}$$

- Learnable smoothing Factor

$$a_{ik} = \frac{\exp(-s_k \|r_{ik}\|^2)}{\sum_{j=1}^{K} \exp(-s_j \|r_{ij}\|^2)}$$



Relation to Other Approaches:

- Dictionary learning: K-means or K-SVD
- BoW, VLAD, Fisher Vector & Net-VLAD
- Global Pooling: Avg-pool, SPP-Net, Bilinear pool

Domain Transfer

- The Residual Encoding discards the frequently appearing features, which is likely to be domain specific.
- For a visual feature x_i that appears frequently in the data, it is likely close to a visual center d_k
  a). $e_k \approx 0$, since $r_{ik} = x_i - d_k \approx 0$
  b). $e_j \approx 0$ $(j \neq k)$, since $a_{ik} \approx 0$ (soft-assignment)

Compare to State-of-the-art

	MINC-2500	FMD	GTOS	KTH	4D-Light
Deep-TEN* (ours)	**81.3**	80.2$_{\pm 0.9}$	**84.5**$_{\pm 2.9}$	**84.5**$_{\pm 3.5}$	**81.7**$_{\pm 1.0}$
State-of-the-Art	76.0$_{\pm 0.2}$ [2]	**82.4**$_{\pm 1.4}$ [5]	N/A	81.1$_{\pm 1.5}$ [4]	77.0$_{\pm 1.1}$ [43]

Experiments:

Dataset

- Material & texture datasets: *MINC-2500, KTH, FMD, 4D-light, GTOS*
- General recognition datasets: *MIT-Indoor, Caltech-101*

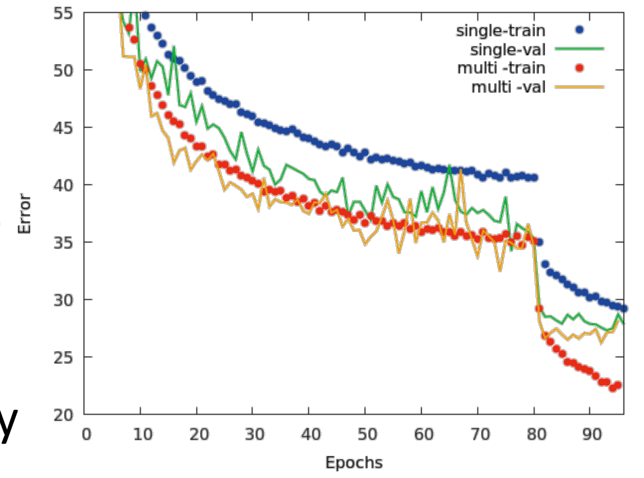
Baselines

- FV-SIFT (128 Gaussian Components, $32K \Rightarrow 512$)
- FV-CNN (Cimpoi *et al*. VGG-VD & ResNet, 32GMM)

	MINC-2500	FMD	GTOS	KTH	4D-Light	MIT-Indoor	Caltech-101
FV-SIFT	46.0	47.0	65.5	66.3	58.4	51.6	63.4
FV-CNN (VGG-VD)	61.8	75.0	77.1	71.0	70.4	67.8	83.0
Deep-TEN (ours)	**80.6**	**80.2**$_{\pm 0.9}$	**84.3**$_{\pm 1.9}$	**82.0**$_{\pm 3.3}$	**81.7**$_{\pm 1.0}$	**71.3**	**85.3**

Effect of Multi-size Training

- Ideally arbitrary image sizes
- Training with pre-defined sizes iteratively w/o modifying solver
- Single-size testing for simplicity



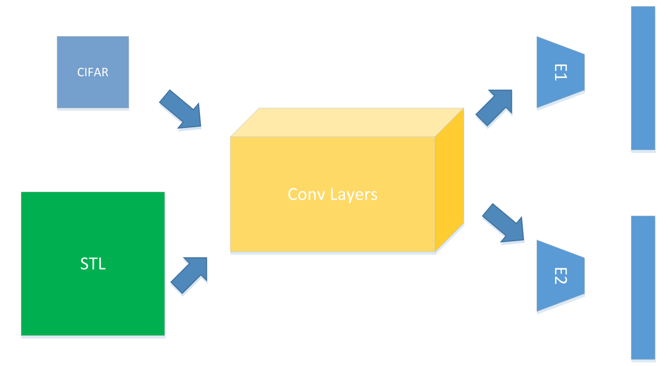
	MINC-2500	FMD	GTOS	KTH	4D-Light	MIT-Indoor
FV-CNN (VGG-VD) multi	63.1	74.0	79.2	77.8	76.5	67.0
FV-CNN (ResNet) multi	69.3	78.2	77.1	78.3	77.6	76.1
Deep-TEN (ours)	80.6	**80.2**$_{\pm 0.9}$	84.3$_{\pm 1.9}$	82.0$_{\pm 3.3}$	**81.7**$_{\pm 1.0}$	71.3
Deep-TEN (ours) multi	**81.3**	78.8$_{\pm 0.8}$	**84.5**$_{\pm 2.9}$	**84.5**$_{\pm 3.5}$	81.4$_{\pm 2.6}$	**76.2**

Table 4: Comparison of single-size and multi-size training.

Joint Encoding

- Dictionary encoding representation is likely to carry domain information.
- The features are likely to be generic.
- CIFAR-10: 36×36
  STL-10: 96×96

Code here!



(Only shallow architectures considered.)

	STL-10	CIFAR-10
Deep-TEN (Individual)	76.29	91.5
Deep-TEN (Joint)	**87.11**	91.8
State-of-the-Art	74.33 [49]	-

Gradients w.r.t Input X The encoder $E = \{e_1, ...e_K\}$ can be viewed as k independent sub-encoders. Therefore the gradients of the loss function ℓ *w.r.t* input descriptor x_i can be accumulated $\frac{d_\ell}{d_{x_i}} = \sum_{k=1}^{K} \frac{d_\ell}{d_{e_k}} \cdot \frac{d_{e_k}}{d_{x_i}}$. According to the chain rule, the gradients of the encoder $w.r.t$ the input is given by

$$\frac{d_{e_k}}{d_{x_i}} = r_{ik}^T \frac{d_{a_{ik}}}{d_{x_i}} + a_{ik} \frac{d_{r_{ik}}}{d_{x_i}}, \tag{4}$$

where a_{ik} and r_{ik} are defined in Sec 2, $\frac{d_{r_{ik}}}{d_{x_i}} = 1$. Let $f_{ik} = e^{-s_k \|r_{ik}\|^2}$ and $h_i = \sum_{m=1}^{K} f_{im}$, we can write $a_{ik} = \frac{f_{ik}}{h_i}$. The derivatives of the assigning weight *w.r.t* the input descriptor is

$$\frac{d_{a_{ik}}}{d_{x_i}} = \frac{1}{h_i} \cdot \frac{d_{f_{ik}}}{d_{x_i}} - \frac{f_{ik}}{(h_i)^2} \cdot \sum_{m=1}^{K} \frac{d_{f_{im}}}{d_{x_i}}, \tag{5}$$

where $\frac{d_{f_{ik}}}{d_{x_i}} = -2 s_k f_{ik} \cdot r_{ik}$.

Gradients w.r.t Codewords C The sub-encoder e_k only depends on the codeword c_k. Therefore, the gradient of loss function *w.r.t* the codeword is given by $\frac{d_\ell}{d_{c_k}} = \frac{d_\ell}{d_{e_k}} \cdot \frac{d_{e_k}}{d_{c_k}}$.

$$\frac{d_{e_k}}{d_{c_k}} = \sum_{i=1}^{N} (r_{ik}^T \frac{d_{a_{ik}}}{d_{c_k}} + a_{ik} \frac{d_{r_{ik}}}{d_{c_k}}), \tag{6}$$

where $\frac{d_{r_{ik}}}{d_{c_k}} = -1$. Let $g_{ik} = \sum_{m \neq k} f_{im}$. According to the chain rule, the derivatives of assigning *w.r.t* the codewords can be written as

$$\frac{d_{a_{ik}}}{d_{c_k}} = \frac{d_{a_{ik}}}{d_{f_{ik}}} \cdot \frac{d_{f_{ik}}}{d_{c_k}} = \frac{2 s_k f_{ik} g_{ik}}{(h_i)^2} \cdot r_{ik}. \tag{7}$$

Gradients w.r.t Smoothing Factors Similar to the codewords, the sub-encoder e_k only depends on the k-th smoothing factor s_k. Then, the gradient of the loss function *w.r.t* the smoothing weight is given by $\frac{d_\ell}{d_{s_k}} = \frac{d_\ell}{d_{e_k}} \cdot \frac{d_{e_k}}{d_{s_k}}$.

$$\frac{d_{e_k}}{d_{s_k}} = -\frac{f_{ik} g_{ik} \|r_{ik}\|^2}{(h_i)^2} \tag{8}$$

Note In practice, we multiply the numerator and denominator of the assigning weight with e^{ϕ_i} to avoid overflow:

$$a_{ik} = \frac{\exp(-s_k \|r_{ik}\|^2 + \phi_i)}{\sum_{j=1}^{K} \exp(-s_j \|r_{ij}\|^2 + \phi_i)}, \tag{9}$$

where $\phi_i = \min_k \{s_k \|r_{ik}\|^2\}$. Then $\frac{d_{\bar{f}_{ik}}}{d_{x_i}} = e^{\phi_i} \frac{f_{ik}}{d_{x_i}}$.