Object Classification using Ensemble of Local and Deep Features

Siddharth Srivastava*, **Prerana Mukherjee***, Brejesh Lall, Kamlesh Jaiswal

Department of Electrical Engineering
Indian Institute of Technology Delhi

* Equal Contribution

Introduction

- Convolutional Neural Networks have become the de-facto standard for a majority of image related tasks
- A standard network is composed of Convolution Layers, Pooling and Fully Connected Layers.
- There are many architectures- VGGNet, GoogleNet, ResNet etc.
- Our research focusses on compare and contrast the effectiveness of various components of these architectures.

Research Questions

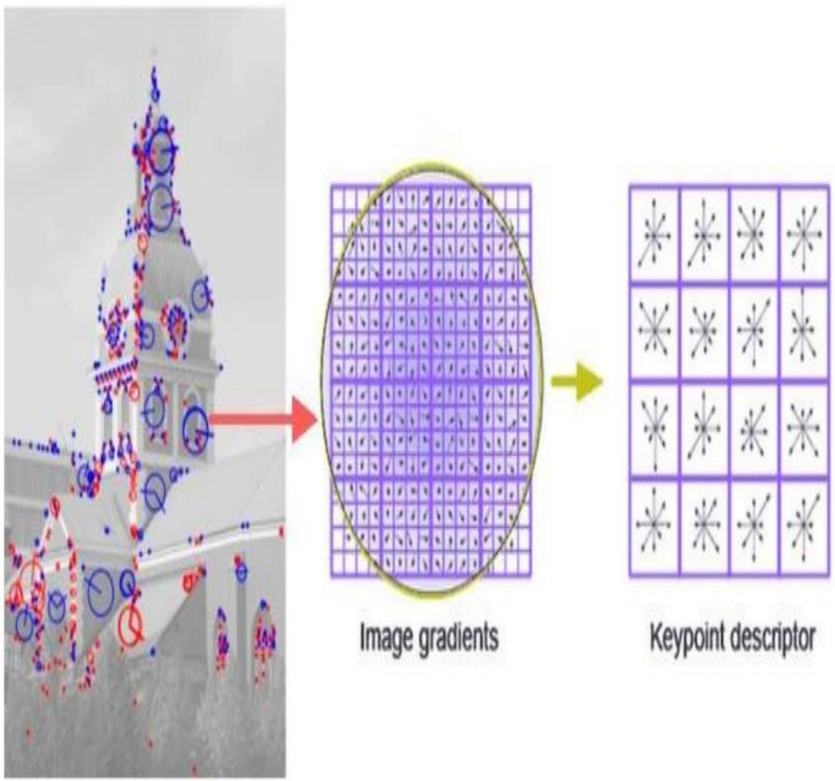
- What is the feature representation ability of intermediate layers of a CNN i.e. are features from fully connected layers always better ?
- Can local features complement the performance of deep features ?
- Are deep features complementary i.e. do the advanced networks subsume information represented by prior networks ?

Contributions

- We compare and contrast effectiveness of feature representation capability of various layers of a convolutional neural network
- We demonstrate with extensive experiments for object classification that the representation capability of features from deep networks can be complemented with information captured from local features
- We also find out that features from various deep convolutional networks encode distinctive characteristic information
- Finally, we propose an ensemble of local and deep features for object classification

Background

Scale Invariant Feature Transform (SIFT)

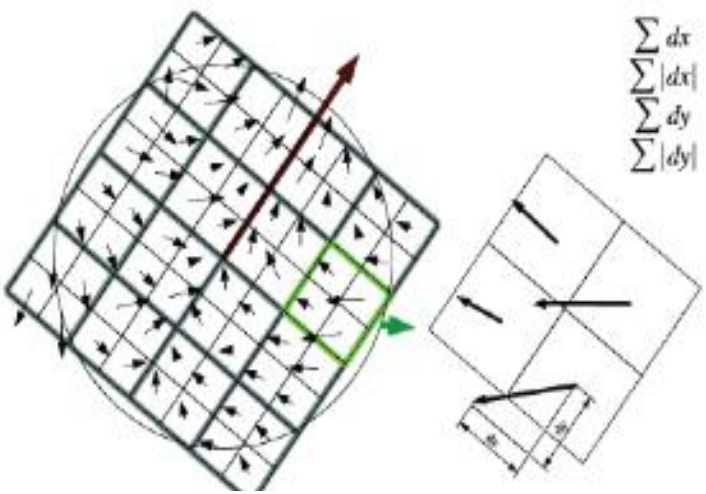


- Invariant to scaling, rotation and translation of image
- Scale space extrema detection
- Keypoint localization
- Orientation assignment
- Keypoint description

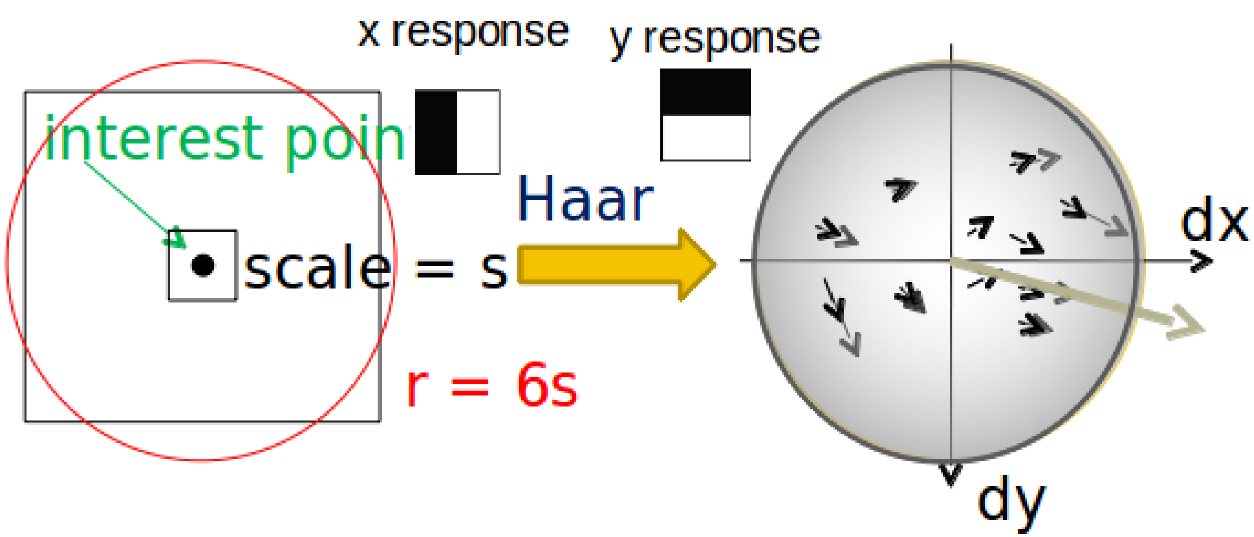
- Make a **(16x16)** around a key-point.
  - Based on 16*16 patches
  - 4*4 subregions
  - 8 bins in each subregion
  - 4*4*8=128 dimensions in total

http://docs.opencv.org/master/sift_inv.jpg

Speeded-up Robust Features (SURF)



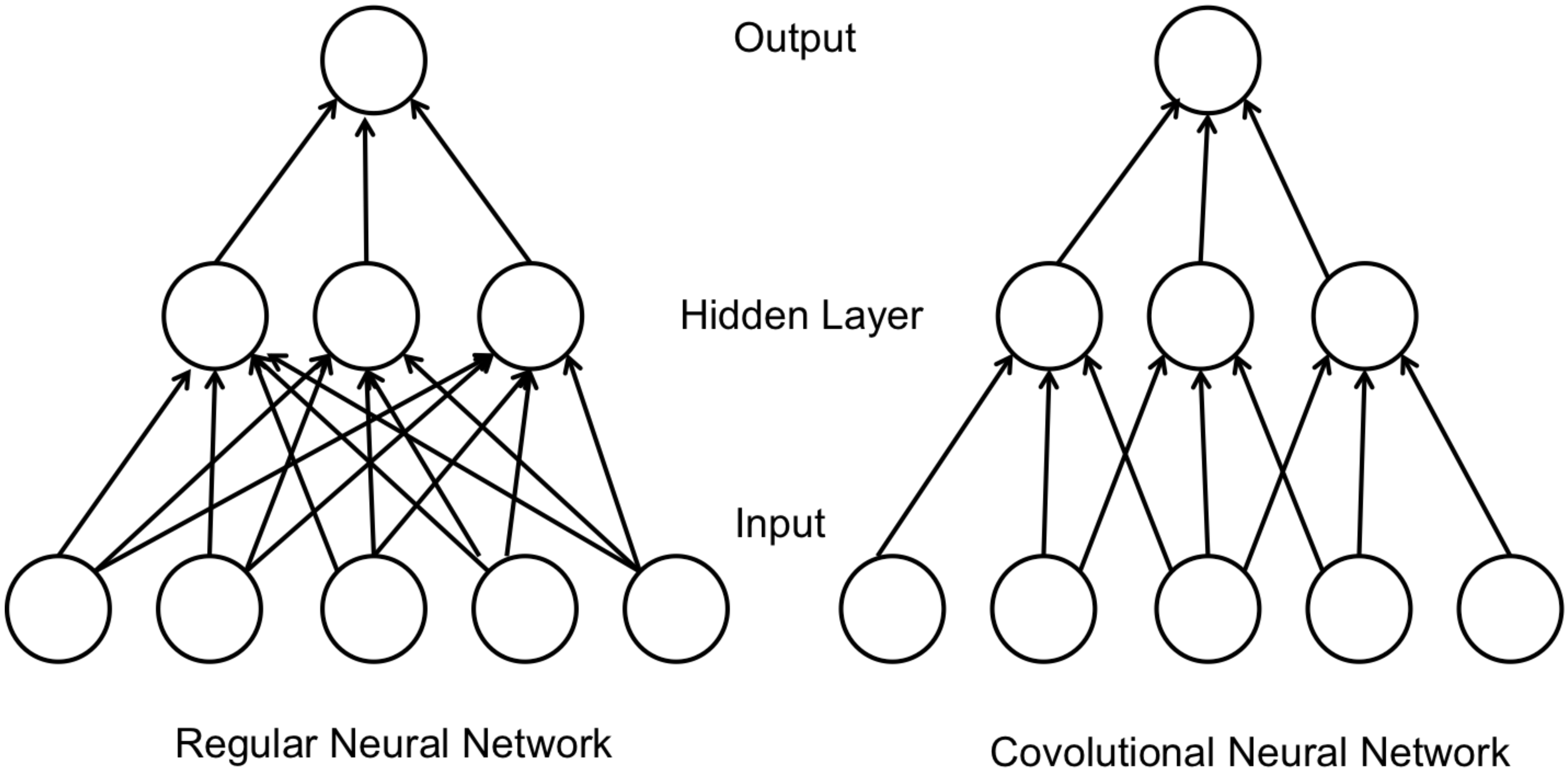
- The feature vector of SURF is almost identical to SIFT. It creates a grid around the keypoint and divide each grid cell into sub-grid.
- At each sub-grid cell, the grid histogram is calculated by Haar wavelet responses.
- These grid-histogram are concatenated into 64-d vector



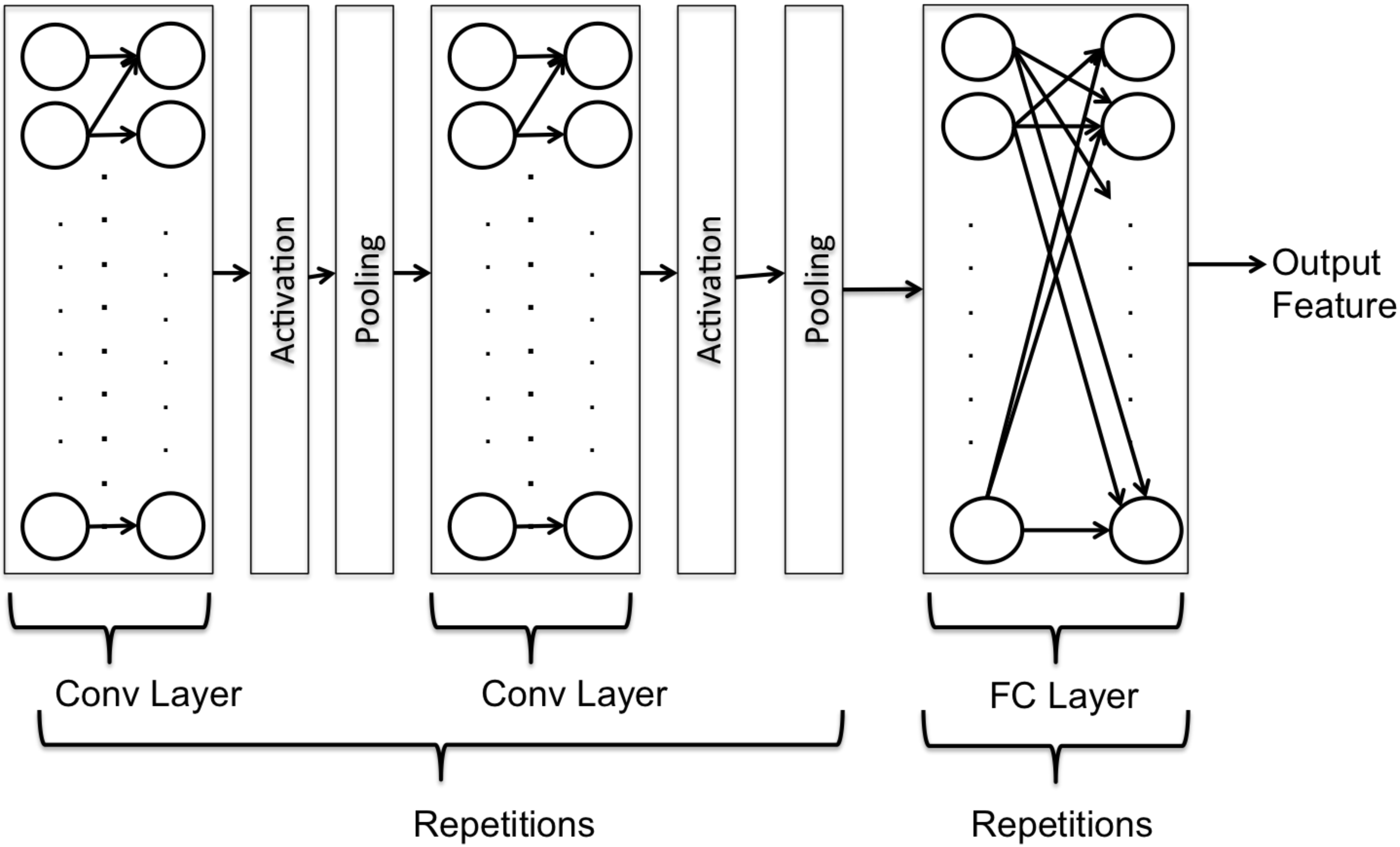
Convolutional Neural Networks (CNNs)

- First practical application of CNN was proposed by Yann Lecun in 1989. The architecture is popularly known as LeNet.
- Due to high computational complexity and advent of SVM, it lost popularity.
- In 2012, Alex Krizhevsky proposed AlexNet on ImageNet Challenge showing superlative performance to previous methods.
- Since then, many variations have been proposed and they are believed to achieve human level efficiency (though not exactly as intelligent as humans !!!)

Regular Neural Network vs CNN



Architecture



Literature Survey

Deep Features are ultimate features

- Razavian et. al. [5], through rigorous experiments suggest that deep convolutional features should be primary features for vision related tasks.

Classification (ImageNet)

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
GHM[8]	76.7	74.7	53.8	72.1	40.4	71.7	83.6	66.5	52.5	57.5	62.8	51.1	81.4	71.5	86.5	36.4	55.3	60.6	80.6	57.8	64.7
AGS[11]	82.2	83.0	58.4	76.1	**56.4**	**77.5**	**88.8**	69.1	**62.2**	61.8	64.2	51.3	**85.4**	**80.2**	91.1	48.1	61.7	**67.7**	86.3	70.9	71.1
NUS[39]	82.5	79.6	64.8	73.4	54.2	75.0	77.5	79.2	46.2	62.7	41.4	74.6	85.0	76.8	91.1	53.9	61.0	67.5	83.6	70.6	70.5
CNN-SVM	88.5	81.0	83.5	82.0	42.0	72.5	85.3	81.6	59.9	58.5	66.5	77.8	81.8	78.8	90.2	54.8	71.1	62.6	87.2	71.8	73.9
CNNaug-SVM	**90.1**	**84.4**	**86.5**	**84.1**	48.4	73.4	86.7	**85.4**	61.3	**67.6**	**69.6**	**84.0**	**85.4**	80.0	**92.0**	**56.9**	**76.7**	67.3	**89.1**	**74.9**	**77.2**

Method	mean Accuracy
HSV [27]	43.0
SIFT internal [27]	55.1
SIFT boundary [27]	32.0
HOG [27]	49.6
HSV+SIFTi+SIFTb+HOG(MKL) [27]	72.8
BOW(4000) [14]	65.5
SPM(4000) [14]	67.4
FLH(100) [14]	72.7
BiCos seg [7]	79.4
Dense HOG+Coding+Pooling[2] w/o seg	76.7
Seg+Dense HOG+Coding+Pooling[2]	80.7
CNN-SVM w/o seg	74.7
CNNaug-SVM w/o seg	**86.8**

Segmentation (Oxford 102 flowers)

	Dim	Oxford5k	Paris6k	Sculp6k	Holidays	UKBench
BoB[3]	N/A	N/A	N/A	**45.4**[3]	N/A	N/A
BoW	200k	36.4[20]	46.0[35]	8.1[3]	54.0[4]	70.3[20]
IFV[33]	2k	41.8[20]	-	-	62.6[20]	83.8[20]
VLAD[4]	32k	55.5 [4]	-	-	64.6[4]	-
CVLAD[52]	64k	47.8[52]	-	-	81.9[52]	89.3[52]
HE+burst[17]	64k	64.5[42]	-	-	78.0[42]	-
AHE+burst[17]	64k	66.6[42]	-	-	79.4[42]	-
Fine vocab[26]	64k	74.2[26]	74.9[26]	-	74.9[26]	-
ASMK*+MA[42]	64k	80.4[42]	77.0[42]	-	81.0[42]	-
ASMK+MA[42]	64k	**81.7**[42]	78.2[42]	-	82.2[42]	-
CNN	4k	32.2	49.5	24.1	64.2	76.0
CNN-ss	32-120k	55.6	69.7	31.1	76.9	86.9
CNNaug-ss	4-15k	**68.0**	**79.5**	42.3	**84.3**	**91.1**
CNN+BOW[16]	2k	-	-	-	**80.2**	-

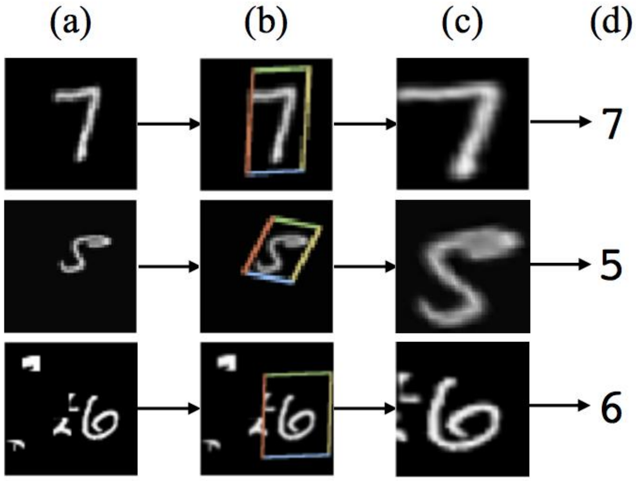
Image Retrieval

Complementarity of deep features (same network)

- Krizhevsky et. al. [6], Zeiler and Fergus [7], Simonyan et. al. [8] exploit complementarity of features in by modifying same network
- Such networks obtain better classification

Impact of Image Transformations

- Authors in [10] [11] show that the output of the convolution layers are not invariant to large image transformations
- Jaderberg et. al. [12] alleviate this problem with Spatial Transformer Network which can be added to existing CNN architecture.



The result of using a spatial transformer as the first layer of a fully-connected network trained for distorted MNIST digit classification. (a) The input to the spatial transformer network (b) The localisation network of the spatial transformer predicts a transformation to apply to the input image. (c) The output of the spatial transformer (d) The classification prediction

- Perronnin et. al. [13] show that using encoded local features with fully connected layers is computationally less expensive than CNN while outperforming traditional approaches.

Methodology

Evaluated Architectures

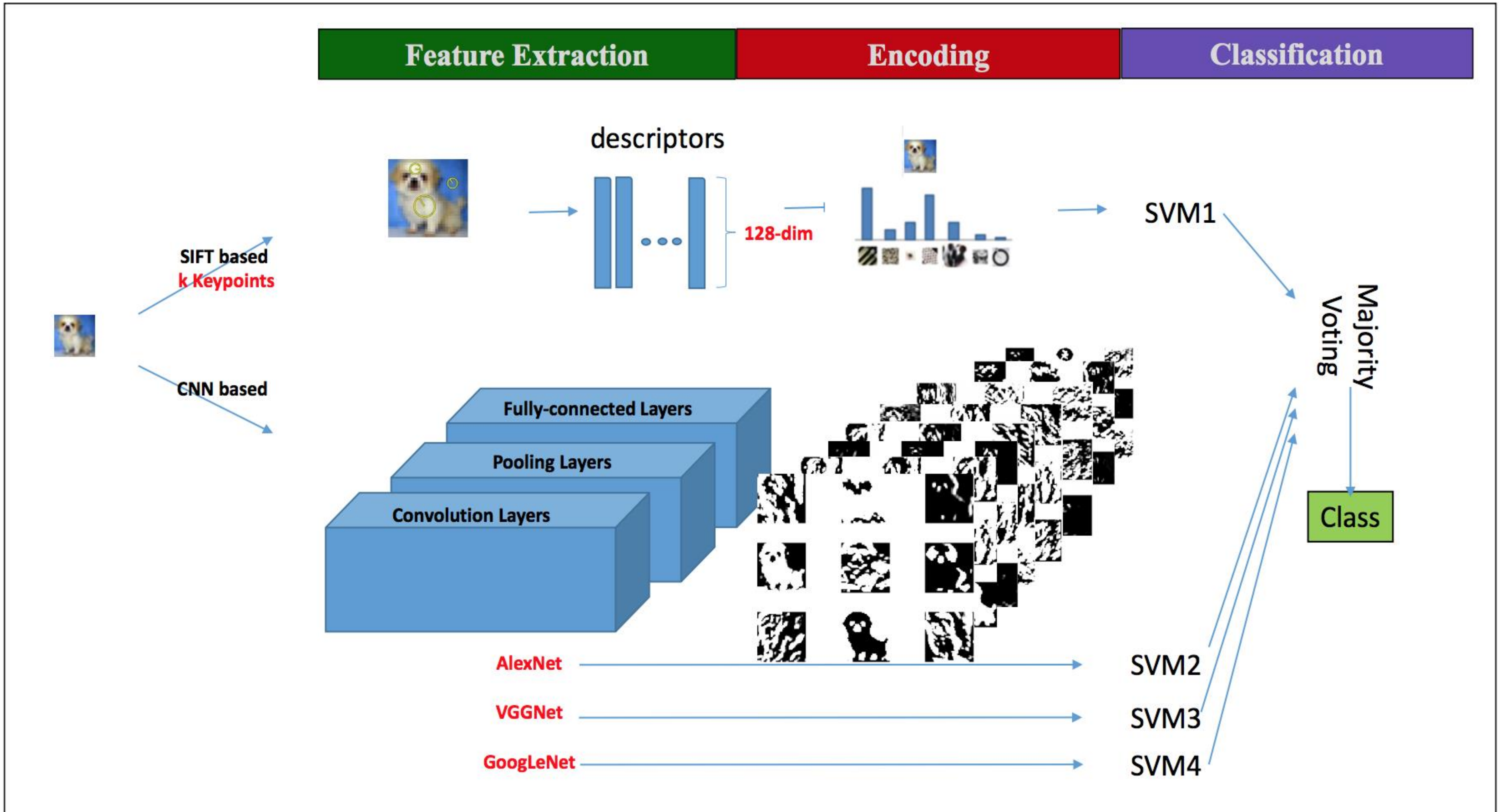
- Deep Ensemble: Ensemble of deep features
- Ensemble of Intermediate Layers
  - Individual Intermediate Layers
  - Fusion of Intermediate Layers
  - SIFT with Deep Ensemble

CNN Model (Layer)	Dimension	Dimension (PCA)
AlexNet (4)	18432	2500
AlexNet (5)	4096	1000
AlexNet (7)	4096	1000
VGGNet (5)	18432	2500
VGGNet (6)	4096	1000
VGGNet (7)	4096	1000

TABLE I

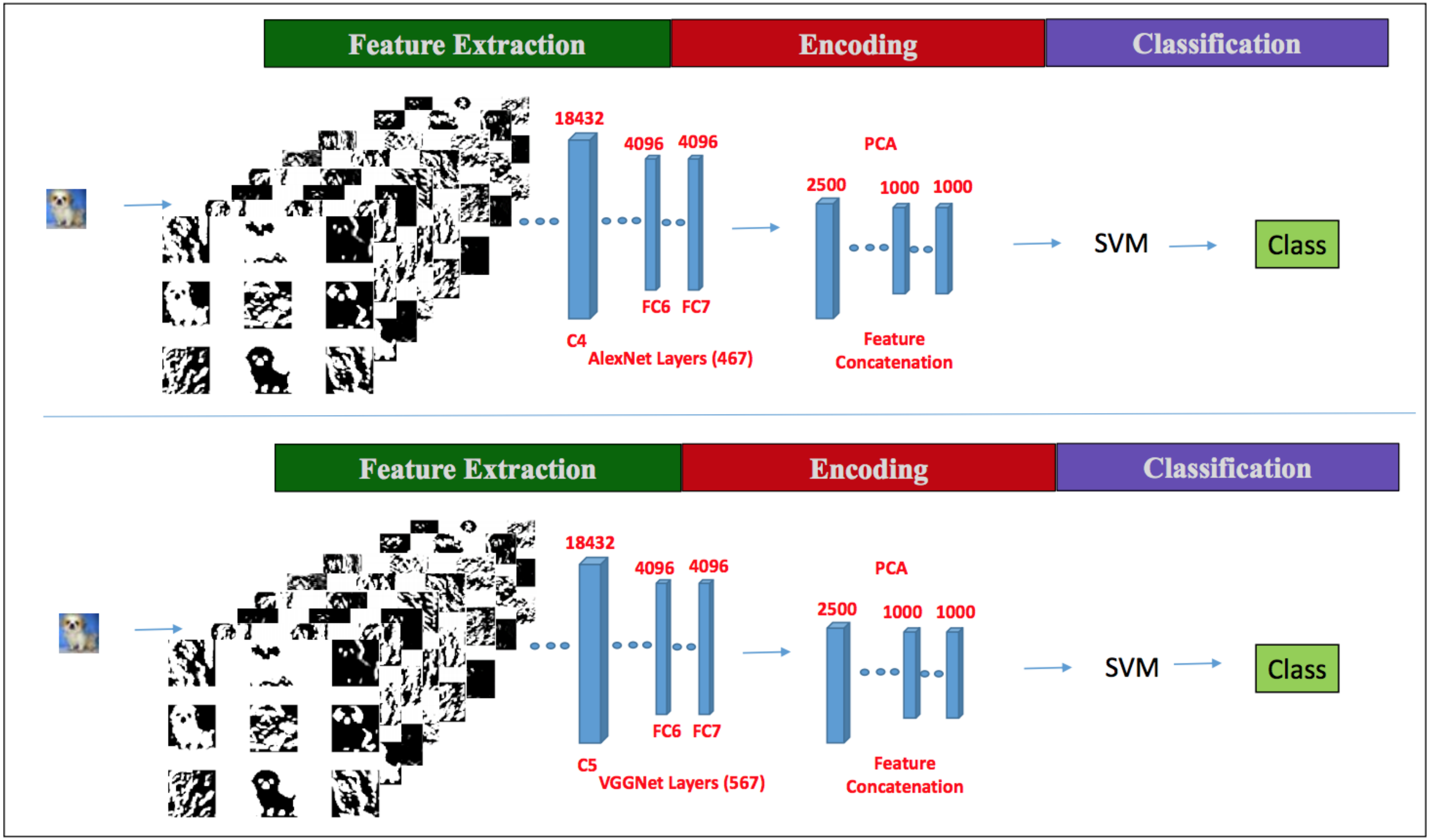
Size of output features from various layers

SIFT + Deep Ensemble



SIFT + Deep Ensemble

Fusion of intermediate layers



Results

CNN Model (Layer)	Accuracy (SVM)	Accuracy (PCA+SVM)
VGGNet (7)	87.6	86.9
VGGNet (6)	90.1	88.3
VGGNet (5)	80.1	85.9
AlexNet (7)	86.1	86.5
AlexNet (6)	84.3	84.2
AlexNet (4)	87.1	88.3
VGGNet (567)	-	89.8
AlexNet (457)	-	88.9
Deep Ensemble	90.8	-
SIFT + Deep Ensemble	**91.1**	-

TABLE II

CLASSIFICATION ACCURACY (%) OF VARIOUS CNN MODELS. VGGNET (567) REPRESENTS THE CONCATENATION OF FEATURES FROM LAYERS 5^{th},6^{th} AND 7^{th} WHILE ALEXNET (467) REPRESENTS THE CONCATENATION OF FEATURES FROM 4^{th}, 5^{th} AND 7^{th} LAYERS

Results

CNN Model (Layer)	Accuracy (SVM)	Accuracy (PCA+SVM)
VGGNet (7)	87.6	86.9
VGGNet (6)	90.1	88.3
VGGNet (5)	80.1	85.9
AlexNet (7)	86.1	86.5
AlexNet (6)	84.3	84.2
AlexNet (4)	87.1	88.3
VGGNet (567)	-	89.8
AlexNet (457)	-	88.9
Deep Ensemble	90.8	-
SIFT + Deep Ensemble	**91.1**	-

TABLE II

CLASSIFICATION AC
(567) REPRESENTS
5^{th},6^{th} AND 7
CONCATENATION

It can be observed that VGGNet (6) performs better than other VGGNet features. Since, VGGNet (6) represents the penultimate layer of the architecture, it indicates that the last fully connected layer results in loss of feature distinctiveness.

Results

CNN Model (Layer)	Accuracy (SVM)	Accuracy (PCA+SVM)
VGGNet (7)	87.6	86.9
VGGNet (6)	90.1	88.3
VGGNet (5)	80.1	85.9
AlexNet (7)	86.1	86.5
AlexNet (6)	84.3	84.2
AlexNet (4)	87.1	88.3
VGGNet (567)	-	89.8
AlexNet (457)		
Deep Ensemble		
SIFT + Deep Ensem		

CLASSIFICATION AC
(567) REPRESENTS
5^{th},6^{th} AND
CONCATENATIO

- Higher accuracy with PCA for AlexNet (4) demonstrates that 4th layer, which is the last convolution layer has redundant features and further layers reduce the strength of the descriptor
- while 4th layer provides highest accuracy, the size of the raw descriptor is nearly 4 times the subsequent layers while we still achieve a 3.5% higher mean accuracy than other PCA reduced AlexNet descriptors

Results

CNN Model (Layer)	Accuracy (SVM)	Accuracy (PCA+SVM)
VGGNet (7)	87.6	86.9
VGGNet (6)	90.1	88.3
VGGNet (5)	80.1	85.9
AlexNet (7)	86.1	86.5
AlexNet (6)	84.3	84.2
AlexNet (4)	87.1	88.3
VGGNet (567)	-	89.8
AlexNet (457)	-	88.9
Deep Ensemble	90.8	-
SIFT + Deep Ensemble	**91.1**	-

CLASSIFICATION AC
(567) REPRESENTS
5^{th},6^{th} AND 7
CONCATENATION

- Combination of features from intermediate layers on an average achieves approximately 3% improvement over other AlexNet and VGGNet features.
- This is a significant gain given that no additional complexity has been introduced for combining or fine-tuning the descriptors.

- The Deep Ensemble shows an average improvement of 4.5%, 4.2% and 8.8% over 7th, 6th and 5th/4th layers of vanilla VGGNet and AlexNet architectures respectively.
- Similarly, the (SIFT+ Deep Ensemble) results in respective improvements of 4.8%, 4.5%, 9.2%.

CNN Model (Layer)	Accuracy (SVM)	Accuracy (PCA+SVM)
VGGNet (7)	87.6	86.9
VGGNet (6)	90.1	88.3
VGGNet (5)	80.1	85.9
AlexNet (7)	86.1	86.5
AlexNet (6)	84.3	84.2
AlexNet (4)	87.1	88.3
VGGNet (567)	-	89.8
AlexNet (457)	-	88.9
Deep Ensemble	90.8	-
SIFT + Deep Ensemble	**91.1**	-

TABLE II

CLASSIFICATION ACCURACY (%) OF VARIOUS CNN MODELS. VGGNET (567) REPRESENTS THE CONCATENATION OF FEATURES FROM LAYERS 5^{th},6^{th} AND 7^{th} WHILE ALEXNET (467) REPRESENTS THE CONCATENATION OF FEATURES FROM 4^{th}, 5^{th} AND 7^{th} LAYERS

Conclusion

- We proposed and evaluated an ensemble of local and deep features for object classification.
- We performed extensive evaluation on CIFAR-10 dataset and demonstrated that local features such as SIFT can complement the deep features
- We also found that different deep architectures characterize distinctive information of an image.
- Additionally, we evaluated features from intermediate layers and their combination, which led us to conclude that such features also complement features from fully connected layers.

References

[1]: Lowe, David G. "Object recognition from local scale-invariant features." Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Vol. 2. IEEE, 1999.

[2]: Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." Computer vision–ECCV 2006 (2006): 404-417.

[3]: J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In Proc. ICCV, Oct 2003.

[4]: T. Majtner, S. Yildirim-Yayilgan, and J. Y. Hardeberg, “Combining deep learning and hand-crafted features for skin lesion classification,” in Image Processing Theory Tools and Applications (IPTA), 2016 6th International Conference on. IEEE, 2016, pp. 1–6.

[5]: A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn
features off-the-shelf: an astounding baseline for recognition,” in Proceedings
of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 806–813.

[6]: A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification
with deep convolutional neural networks,” in Advances in neural information
processing systems, 2012, pp. 1097–1105.

[7]: M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in European conference on computer vision. Springer, 2014, pp. 818–833.

[8]: K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in Advances in neural information processing systems, 2014, pp. 568–576.

[9]: T. Majtner, S. Yildirim-Yayilgan, and J. Y. Hardeberg, “Combining deep learning and hand-crafted features for skin lesion classification,” in Image Processing Theory Tools and Applications (IPTA), 2016 6th International Conference on. IEEE, 2016, pp. 1–6.

[10]: K. Lenc and A. Vedaldi, “Understanding image representations by measuring their equivariance and equivalence,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 991–999.

References

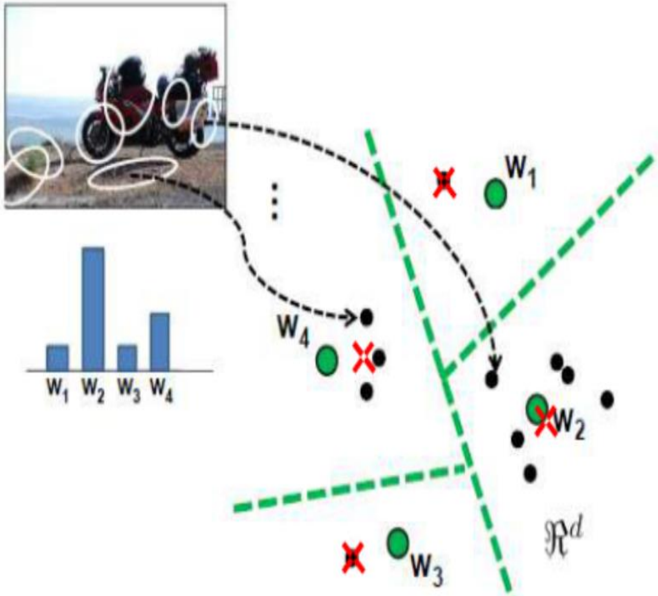
[11]: T. S. Cohen and M. Welling, "Transformation properties of learned visual representations," ICLR, 2015.

[12]: M. Jaderberg, K. Simonyan, A. Zisserman et al., "Spatial transformer networks," in Advances in Neural Information Processing Systems, 2015, pp. 2017–2025.

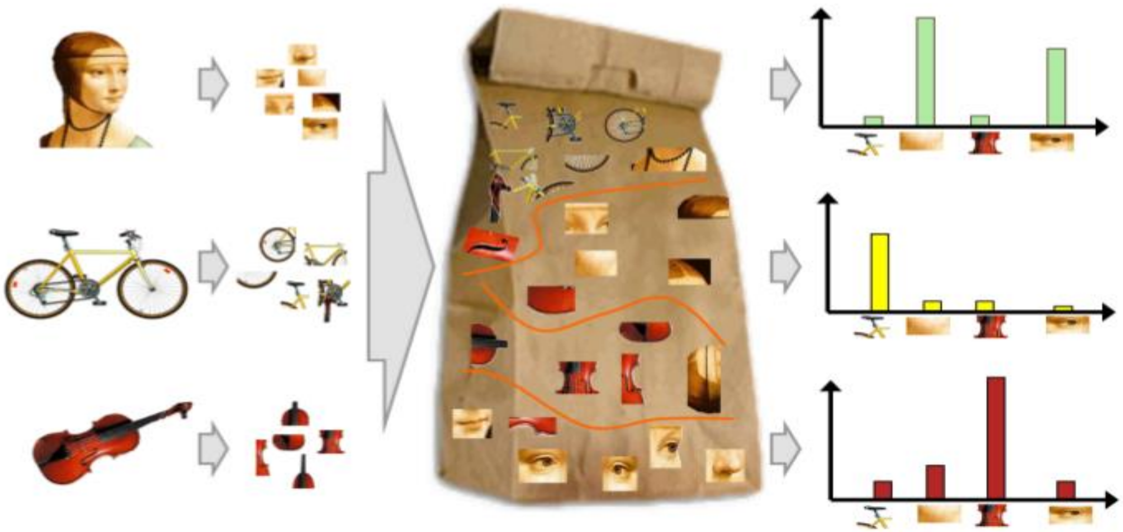
[13]: F. Perronnin and D. Larlus, "Fisher vectors meet neural networks: A hybrid classification architecture," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3743–3752.

THANK YOU

Bag of Visual Words (BoVW)



- SIFT or SURF feature are quantised into Bag of visual words with k-means clustering.
- The nearest point are encoded into centroid point.
- Image encoded into histogram with the help this BoW .
- The dimension of histogram is number of cluster.



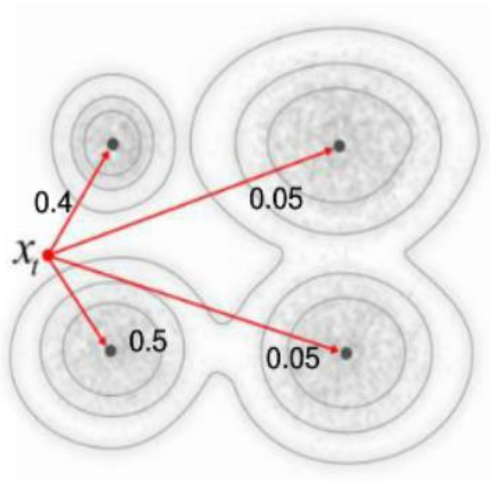
Fisher Vector (FV)

FV formulas:

- gradient wrt to w

≈ $\boxed{\frac{1}{T}\sum_{t=1}^{T}\gamma_t(i)}$

→ **soft BOV**



- gradient wrt to μ and σ

$$\boxed{\begin{aligned}\mathcal{G}_{\mu,i}^{X} &= \frac{1}{T\sqrt{w_i}}\sum_{t=1}^{T}\gamma_t(i)\left(\frac{x_t-\mu_i}{\sigma_i}\right)\\ \mathcal{G}_{\sigma,i}^{X} &= \frac{1}{T\sqrt{2w_i}}\sum_{t=1}^{T}\gamma_t(i)\left[\frac{(x_t-\mu_i)^2}{\sigma_i^2}-1\right]\end{aligned}}$$

$\gamma_t(i)$ = soft-assignment of patch t to Gaussian i

→ compared to BOV, include **higher-order statistics** (up to order 2)

→ FV **much higher-dim** than BOV for a **given visual vocabulary size**

→ FV **much faster to compute** than BOV for a **given feature dim**

Convolution with Images

Mathematical Representation

$$response = I * K = \sum_{c=-\frac{n-1}{2}}^{\frac{n+1}{2}} \left(\sum_{r=-\frac{m-1}{2}}^{\frac{m+1}{2}} I(a+r, b+c)K(r,c) \right)$$

Feature Map

Image

Convolution Kernel

Example

$$\begin{bmatrix} 6 & 8 & 7 & 3 & 6 \\ 4 & 1 & 4 & 2 & 4 \\ 8 & 2 & 1 & 5 & 8 \\ 9 & 1 & 6 & 0 & 9 \\ 9 & 1 & 6 & 0 & 9 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ -1 & -1 & -1 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 25 \end{bmatrix}$$

Input Image

Convolution Kernel

response = 1*1 + 4*2 + 2*1 + 2*(-1) + 1*(-1) + 5*(-1) + 1*1 + 6*2 + 0*1 = 25

Convolution Layer

- Purpose: To detect features from images (lines, edges etc.).
- It is achieved by a set of filters which are learned to detect these features.
- The filters are small in terms of width and height but extend to the complete depth of the input image.
- **Convolution** between the input volume and filter is performed by sliding the filter across the width and height of the input volume while computing the dot product on the overlapping values at a location.

Convolution Layer

- Hyper-Parameters
  - Depth
  - Stride
  - Size of Filter
  - Zero-padding
- Input and Output Volume Size (see example below)

Size of input image (I) = 227X227 Size of Filter or Receptive Field Size (F_S) = 13

Stride (S) = 2

Padding (P) = 0

Depth of Convolution Layer (D) = 96

the size of the output volume can be computed as

$$O = \frac{227-13+0}{2} + 1 = 108$$

which results in a output volume of size

Size of Output Volume = 108 X 108 X 96

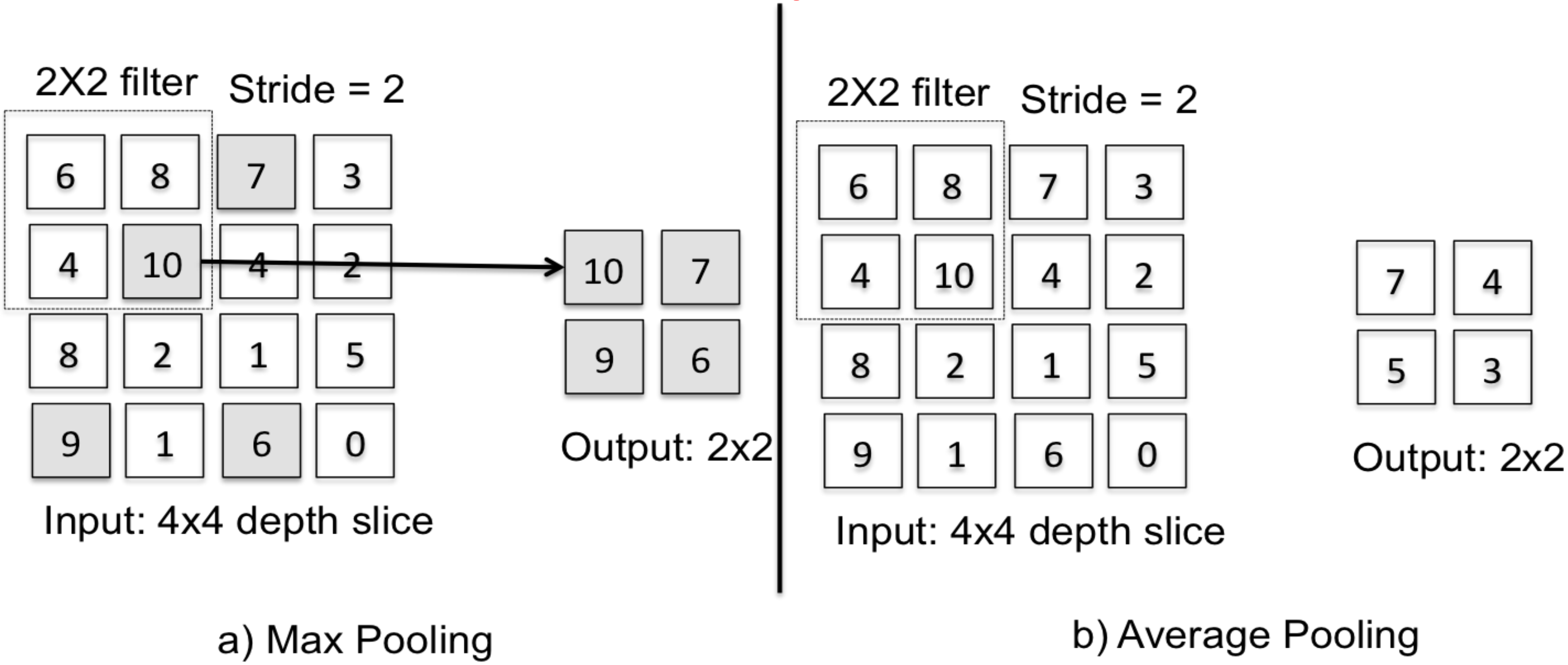
Pooling Layer

- Purpose: Progressively down sample the input volume from the Convolution Layer.
- It is an optional layer and is put between successive convolution layers.
- It results in reduction of the number of parameters and avoids overfitting

Pooling Layer

- Hyper-parameters
  - Spatial Extent
  - Stride

Example



Fully Connected Layer

- Fully Connected layer is the final layer in a CNN.
- It is a fully connected layer from the output volume of convolutional/pooling layer to neurons in this layer.
- The CNN architecture can contain multiple dense layers
- The reason that fully-connected layers are used towards the end is:
  - Convolution layer is exploits the spatial structure in the input image.
  - fully connected layers require huge number of parameters which would make the architecture computationally inefficient if used towards the beginning.