# Deep Fisher Networks for Large-Scale Image Classification

Karen Simonyan, Andrea Vedaldi, Andrew Zisserman
{karen, vedaldi, az}@robots.ox.ac.uk
Visual Geometry Group, University of Oxford, UK

## 1. OVERVIEW

**Motivation**

Deep convolutional networks achieve excellent performance in large-scale image classification tasks [Krizhevsky et al., 2012].

***Do traditional architectures benefit from the increased depth?***

**Objectives**

- Extend a state-of-the-art shallow image classification pipeline to a deep architecture.
- Evaluate the benefit of the increased depth.

## 2. FISHER VECTOR ENCODING

The state-of-the-art shallow image classification pipeline [Perronnin et al., 2010] comprises the following steps.

**Low-level feature extraction**

Visual features (SIFT and colour) are densely extracted at several scales in the image, resulting in a set of feature vectors $x_p \in \mathbb{R}^D$.

**Fisher Vector (FV) encoding**

Local features $x_p$ are PCA-decorrelated and pooled into a Fisher vector $\Phi$ by soft-assignment to a GMM

$$\alpha_k(x_p) \propto \mathcal{N}(x_p|\mu_k, \sigma_k),$$

followed by computing the first and second order statistics over the pooling window:

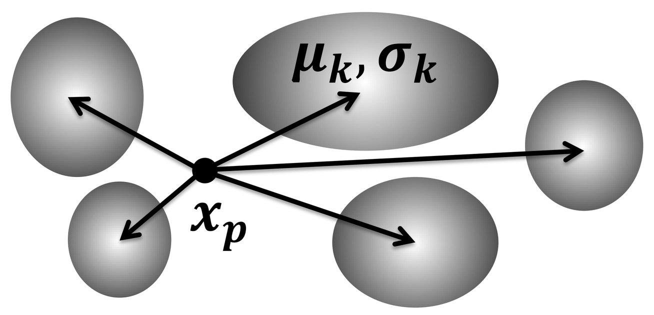
$$\Phi_k^{(1)} = \frac{1}{N\sqrt{\pi_k}} \sum_{p=1}^{N} \alpha_k(x_p)\left(\frac{x_p - \mu_k}{\sigma_k}\right),$$

$$\Phi_k^{(2)} = \frac{1}{N\sqrt{2\pi_k}} \sum_{p=1}^{N} \alpha_k(x_p)\left(\frac{(x_p-\mu_k)^2}{\sigma_k^2} - 1\right).$$

This can be seen as an approximation of the Fisher kernel [Jaakkola and Haussler, 98] using the diagonal-covariance GMM as a generative model.

FV has a high dimensionality: $2KD$, where $K$ is a number of Gaussians in the GMM.

**GMM assignment**




**Linear SVM classification**

Image classification is performed using one-vs-rest linear SVMs on top of FV image descriptors.

## 3. FISHER LAYER

*A **Fisher layer (FL)** transforms dense low-dimensional input features into more discriminative, but still dense and low-dimensional features with larger spatial support.*

It includes three sub-layers:

1. Compressed local FV encoding
2. Spatial stacking
3. Normalisation & PCA decorrelation

**1. Compressed local FV encoding**

*FV is employed to encode local image windows, rather than the whole images.*

- FVs are pooled over densely sampled local windows of different size.
- To prevent dimensionality explosion in the next layer, FV dimensionality must be reduced.
- **Discriminative dimensionality reduction** is done by projection onto the space of classifier scores (alternative: WSABIE [Weston et al., 2011]).

**2. Spatial stacking**

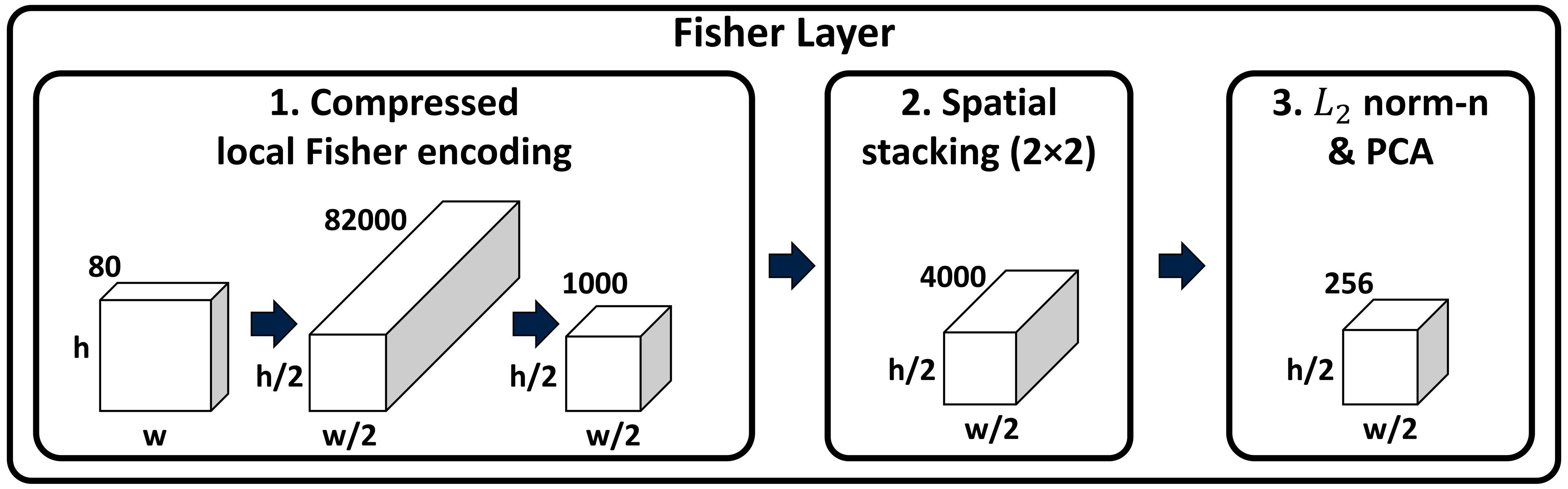
*Weak geometric information is encoded at each location by stacking spatially adjacent features.*

- Spatially adjacent low-dim FV, pooled with the same window size, are stacked in a 2×2 window.

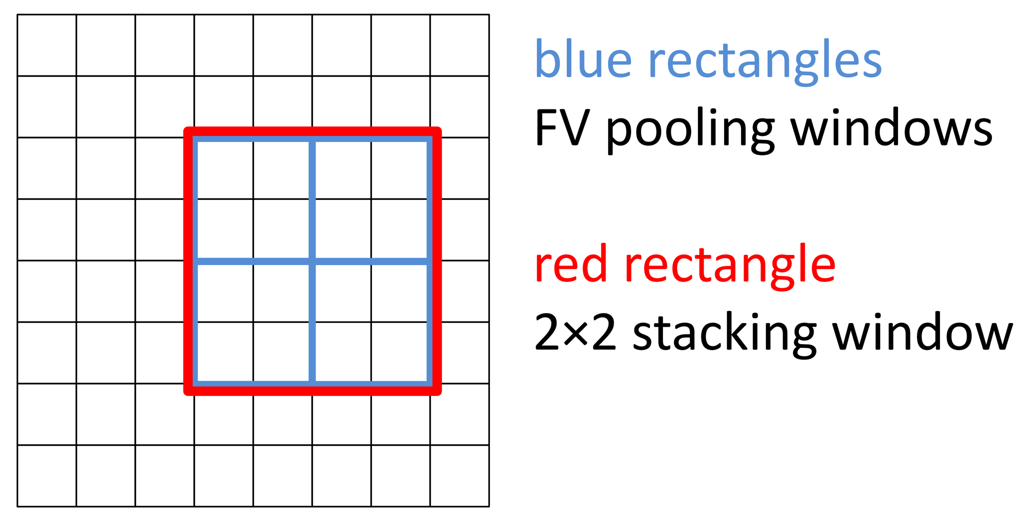
**3. Normalisation & PCA decorrelation**

*Feature post-processing*

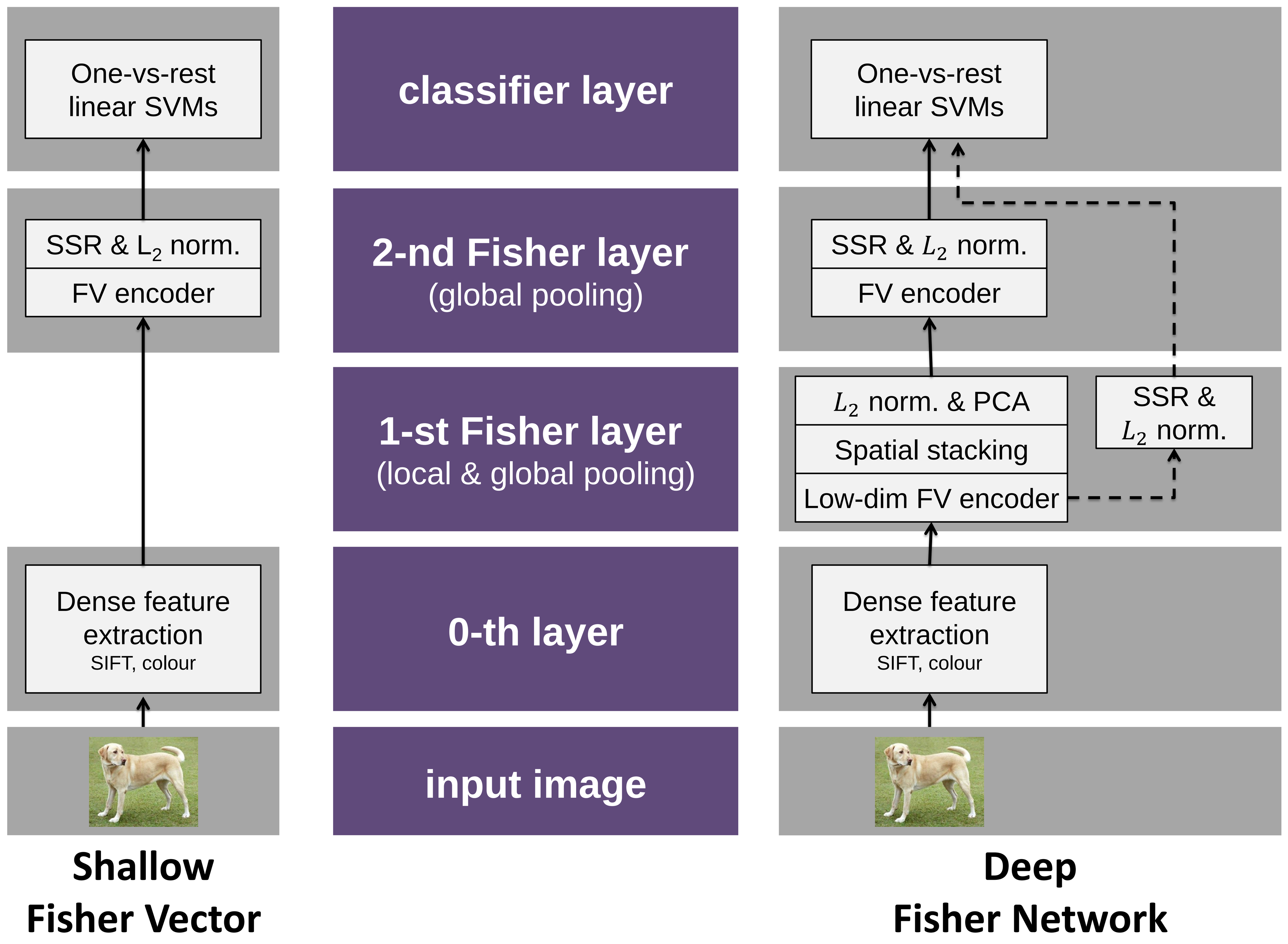
- $L_2$ normalisation to improve generalisation.
- Decorrelation is required for the next Fisher layer.
- PCA can be used for additional dimensionality reduction.




**Spatial stacking**




## 4. FISHER NETWORK




**Fisher network** – a composition of several (≥ 2) Fisher layers on top of dense features, followed by linear SVMs.

- **Training is performed greedily** (layer-by-layer).
- Globally pooled FVs are branched out of each Fisher layer and concatenated to produce a multi-scale image descriptor.
- Signed Square-Rooting (SSR) is applied after global pooling to further improve the performance.
- Large-scale FV computation is speeded-up by the hard assignment to the GMM.

**ImageNet 2010 & 2012 classification accuracy (%)**

| Method | 2010 | | 2012 | |
|---|---|---|---|---|
| | top-1 | top-5 | top-1 | top-5 |
| 1st FL (shallow baseline) | 55.4 | 76.4 | 50.6 | 72.7 |
| 2nd FL | 56.2 | 77.7 | | |
| **1st and 2nd FL (FishNet)** | **59.5** | **79.2** | **55.3** | **76.6** |
| ConvNet [Krizhevsky et al., 2012] | 62.5 | 83.0 | 59.4 | 81.8 |
| ConvNet, 5 instances | | | 61.9 | 83.6 |
| ConvNet (1 instance, our implement.) | 62.9 | 83.2 | 60.3 | 82.3 |
| **FishNet & ConvNet** | **66.8** | **85.6** | **63.8** | **84.7** |

This work was supported by ERC grant VisRec no. 228180