THE UNIVERSITY OF TOKYO

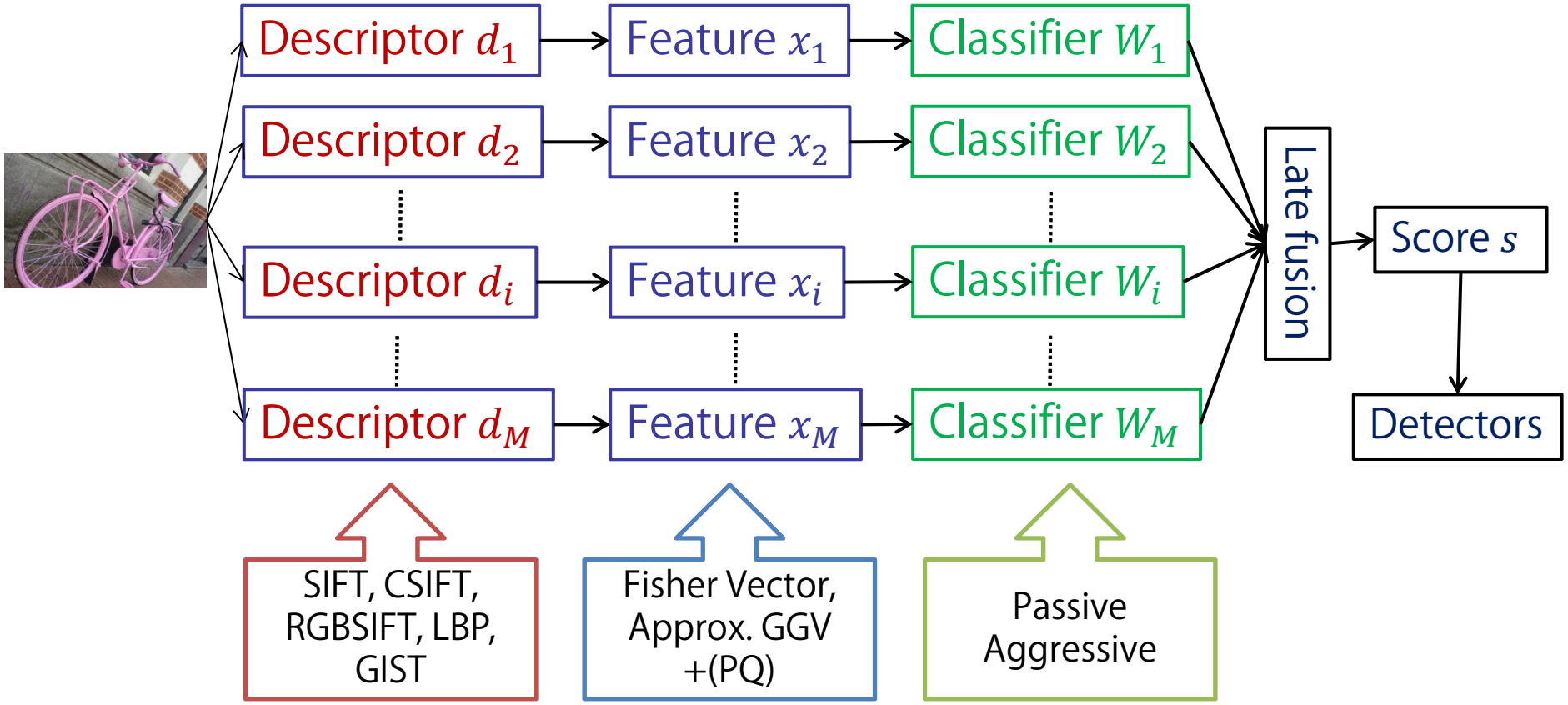
The Univ. of Tokyo, ILSVRC2012 Scalable Multiclass Object Categorization with Fisher Based Features

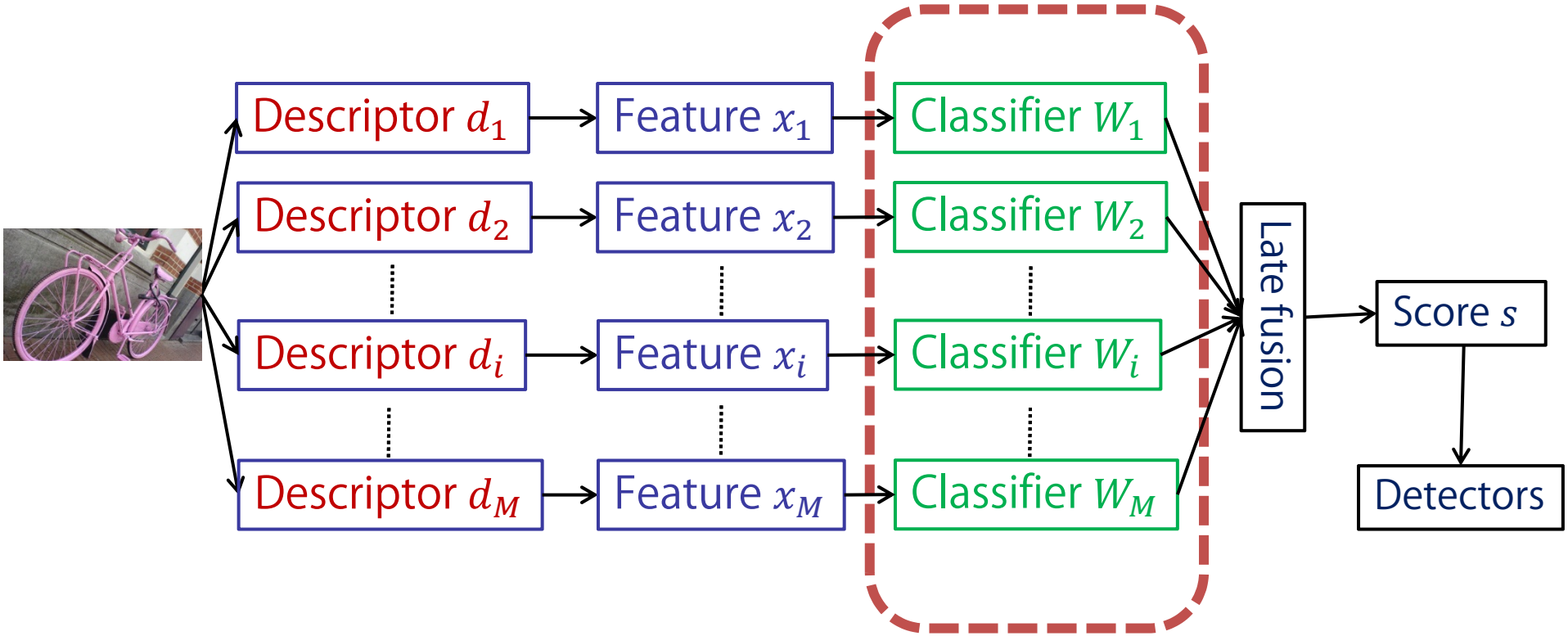
N. Gunji, T. Higuchi, K. Yasumoto, H. Muraoka,
Y. Ushiku, T. Harada, and Y. Kuniyoshi

Intelligent Systems and Informatics Lab.,
the University of Tokyo, Japan

Overview

Fisher based features + Multi class linear classifiers




Descriptor d_1
Feature x_1
Classifier W_1
Descriptor d_2
Feature x_2
Classifier W_2
Descriptor d_i
Feature x_i
Classifier W_i
Descriptor d_M
Feature x_M
Classifier W_M
Late fusion
Score s
Detectors

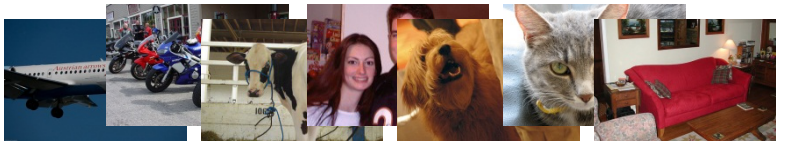
Linear Classifiers

- One-vs.-the-rest SVMs are usually used for object categorization.
- The classifiers are learned independently.

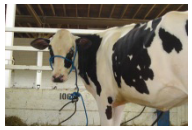
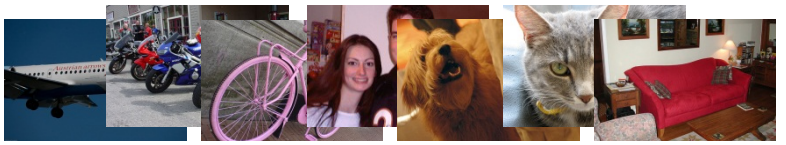
F. Perronnin, Z. Akata, Z. Harchaoui and C. Schmid. Towards Good Practice in Large-Scale Learning for Image Classification. CVPR 2012.

One-vs.-the-rest manner

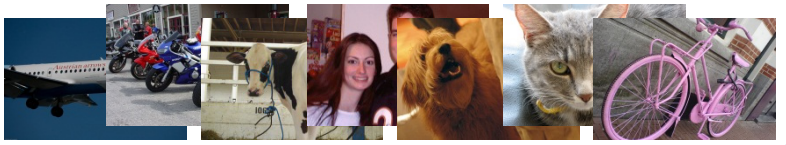
"Bicycle" classifier

positive  vs. negative 

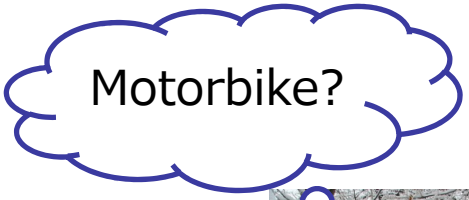
"Cow" classifier

positive  vs. negative 

"Sofa" classifier

positive  vs. negative 

Problem of one-vs.-the-rest



There is no guarantee that the scores for different classifiers will have appropriate scales.

Input

Predicted labels

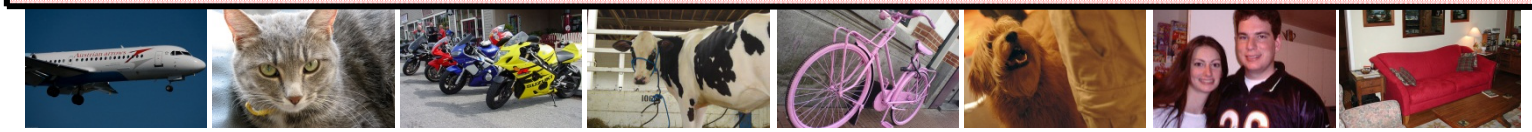
Correct labels

prediction score

YES

NO

0



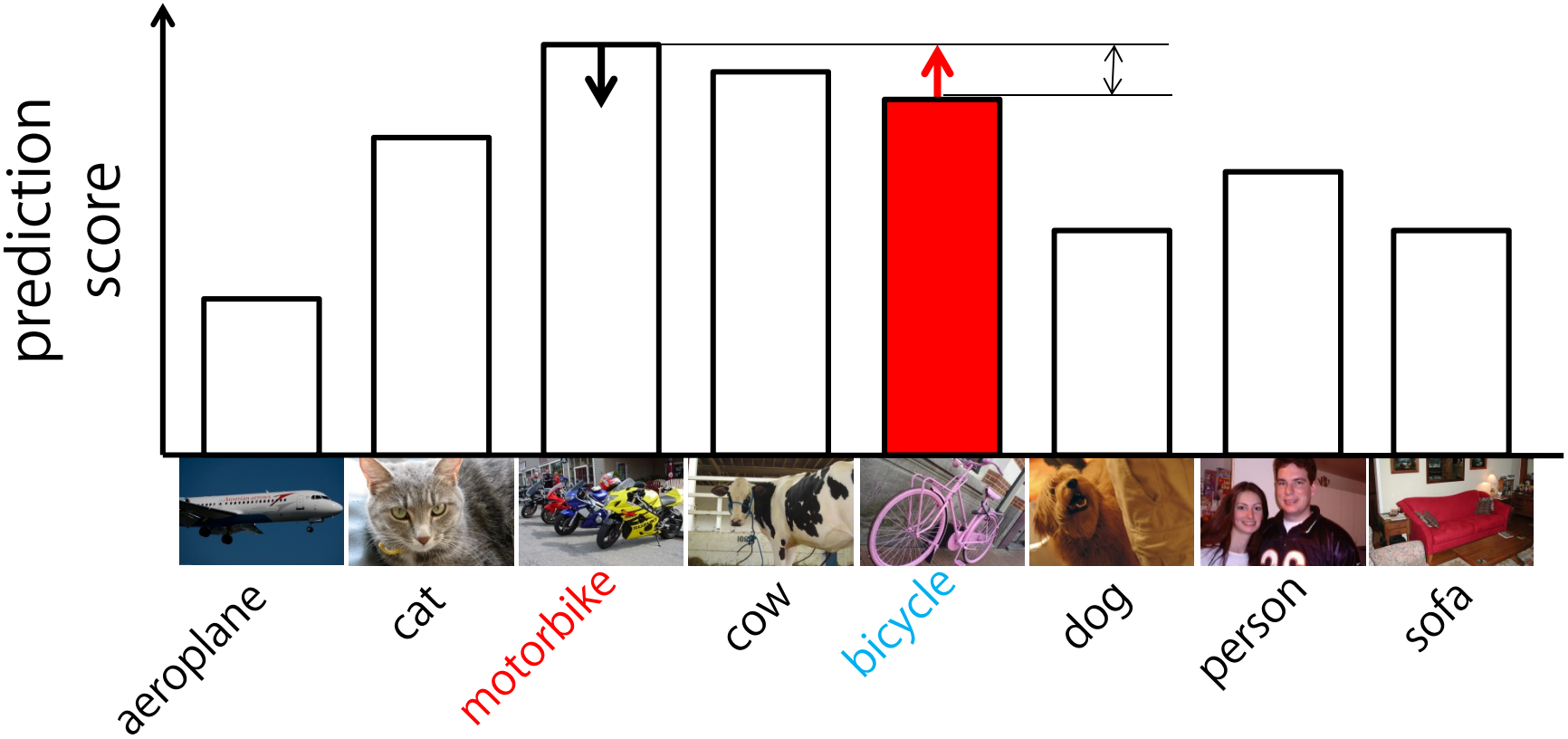
aeroplane cat motorbike cow bicycle dog person sofa

Multiclass Learning

Input

Regularizing their magnitude relation
cf. SVM Multiclass

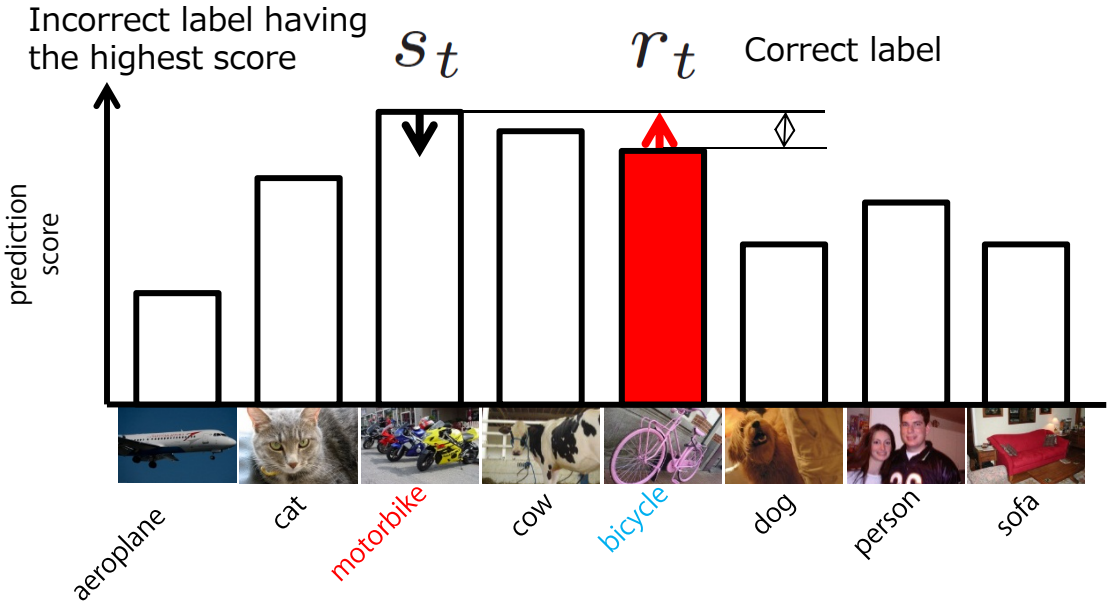
K. Crammer and Y. Singer. On the Algorithmic Implementation of Multi-class SVMs. JMLR, Vol.2 , pp.265-292, 2001.



Passive Aggressive

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, Y. Singer. Online Passive-Aggressive Algorithms. JMLR, Vol.7, pp.551-585, 2006.

Input



- Pairwise loss

$$l\left(\boldsymbol{w}_t^{r_t}, \boldsymbol{w}_t^{s_t}; (\boldsymbol{x}_t, \boldsymbol{Y}_t)\right) = \begin{cases} 0 & \text{if } \boldsymbol{w}_t^{r_t} \cdot \boldsymbol{x}_t - \boldsymbol{w}_t^{s_t} \cdot \boldsymbol{x}_t \geq 1 \\ 1 - (\boldsymbol{w}_t^{r_t} \cdot \boldsymbol{x}_t - \boldsymbol{w}_t^{s_t} \cdot \boldsymbol{x}_t) & \text{otherwise} \end{cases}$$

- Optimization problem

$$\boldsymbol{w}_{t+1}^{r_t}, \boldsymbol{w}_{t+1}^{s_t} = \arg \min_{w^{r_t}, w^{s_t}} Cl\left(\boldsymbol{w}^{r_t}, \boldsymbol{w}^{s_t}; (\boldsymbol{x}_t, \boldsymbol{Y}_t)\right)^2 + ||\boldsymbol{w}^{r_t} - \boldsymbol{w}_t^{r_t}||^2 + ||\boldsymbol{w}^{s_t} - \boldsymbol{w}_t^{s_t}||^2$$

- Online learning

$$\boldsymbol{w}_{t+1}^{r_t} = \boldsymbol{w}_t^{r_t} + \tau_t \boldsymbol{x}_t, \quad \boldsymbol{w}_{t+1}^{s_t} = \boldsymbol{w}_t^{s_t} - \tau_t \boldsymbol{x}_t \qquad \tau_t = \frac{l\left(\boldsymbol{w}^{r_t}, \boldsymbol{w}^{s_t}; (\boldsymbol{x}_t, \boldsymbol{Y}_t)\right)}{2||\boldsymbol{x}_t||^2 + 1/C}$$

Online Learning

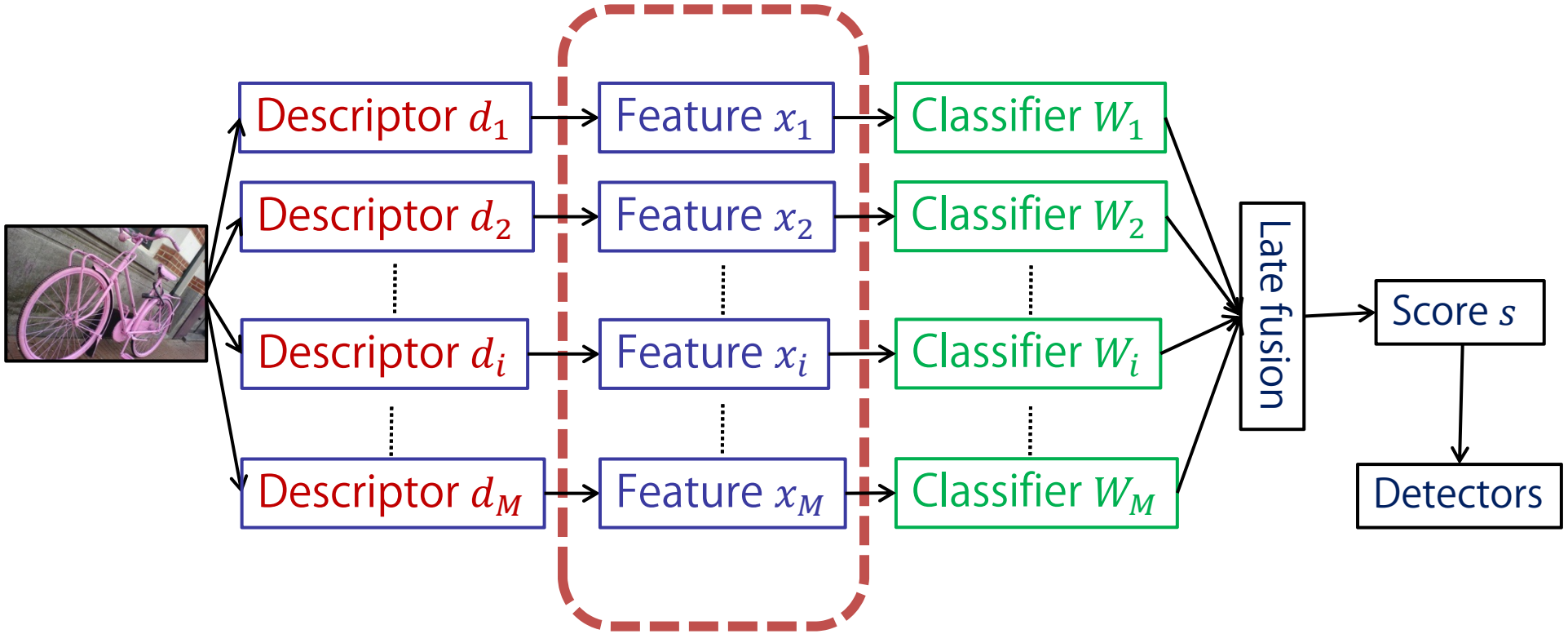
$$w_{t+1} = w_t + \tau_t x_t$$

updated classifier　current classifier　Gain　current datum

	Multiclass	tuning
• SGD-SVM [Bottou et al., 2010]		
+one-vs.-the-rest	✗	✗
+multiclass	✓	✗
• PA [Crammer et al., 2006]	✓	✓
• CW [Crammer et al., 2009]	✓	✓
• AROW [Crammer et al., 2009]	✓	✓
• NHERD [Crammer et al., 2010]	✓	✓

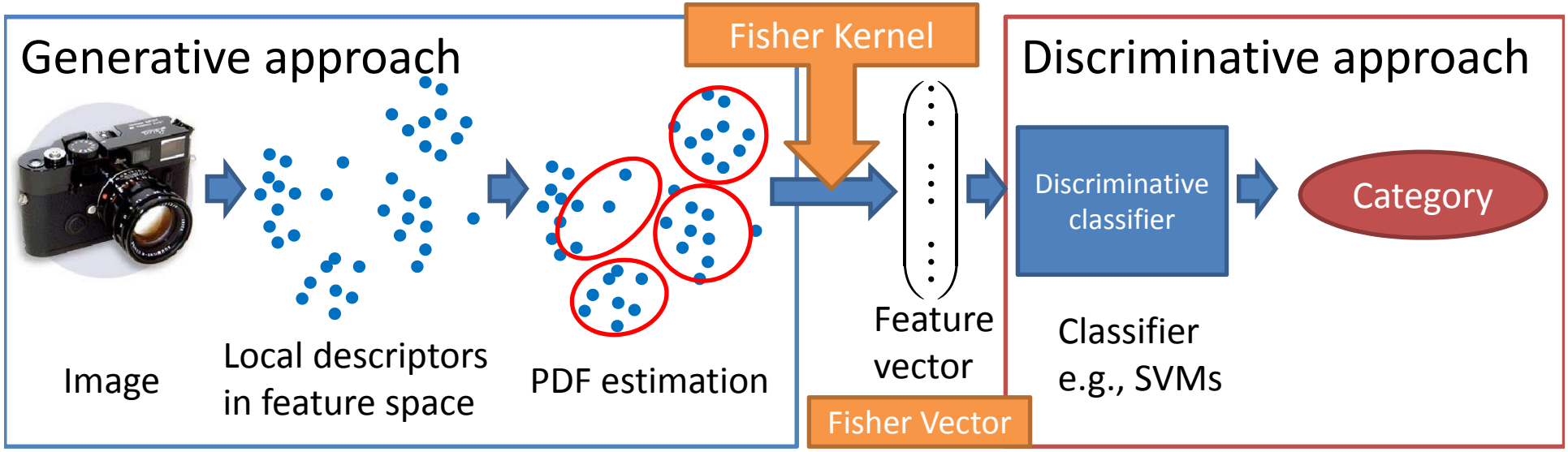
However, there is a little trick to obtain good results.

Unpublished work


Descriptor d_1
Descriptor d_2
Descriptor d_i
Descriptor d_M
Feature x_1
Feature x_2
Feature x_i
Feature x_M
Classifier W_1
Classifier W_2
Classifier W_i
Classifier W_M
Late fusion
Score s
Detectors

Fisher Vector

F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. CVPR, 2007.



- Gaussian Mixture Models u_θ
- Gradient of log-likelihood $G_\theta^{\mathcal{X}} = \frac{1}{N}\nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta})$
- Fisher Information Matrix $F_\theta = E_X[\nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta})\nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta})^\top]$
- Fisher Vector $\mathcal{G}_\theta^X = F_\theta^{-1/2}\nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta})$

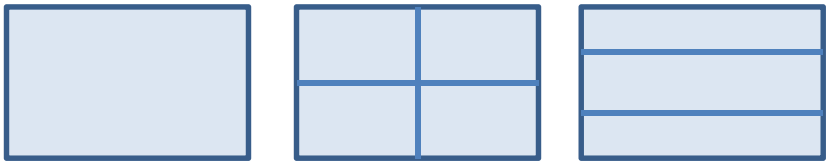
Improved Fisher Vector

- We used the improved Fisher Vectors.
  - F. Perronnin, J. Sanchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. ECCV, 2010.
  - INRIA's Fisher vector implementation
    - http://lear.inrialpes.fr/src/inria_fisher/

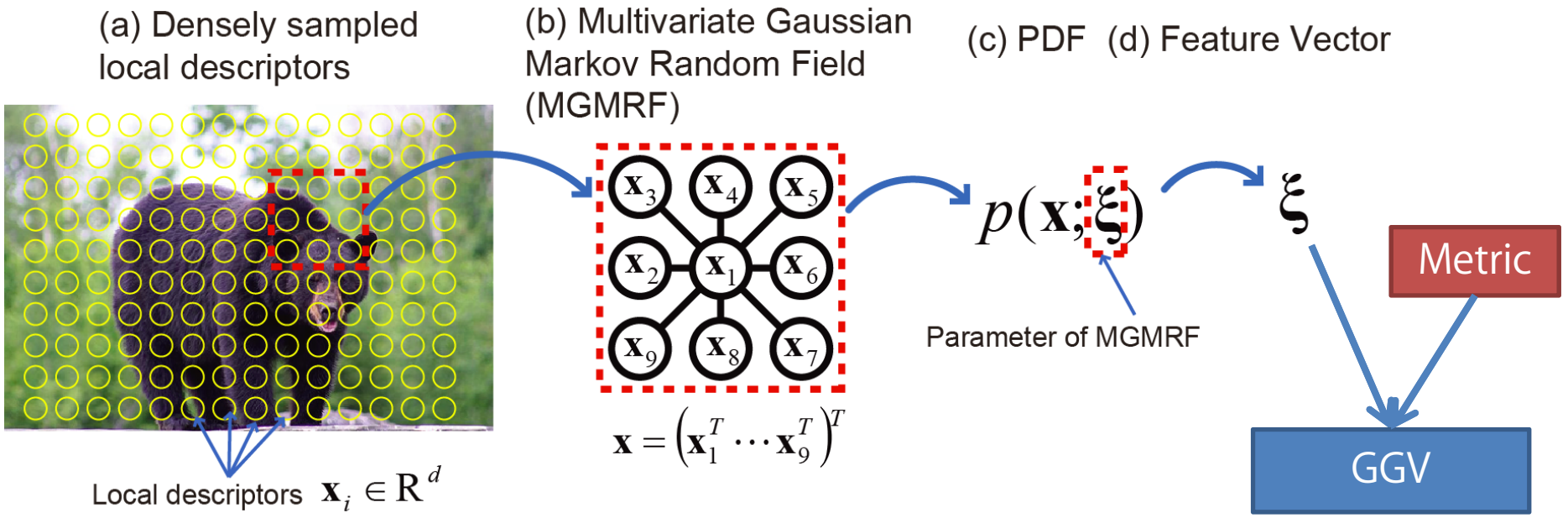
- Improved Fisher Vector
  - L2 normalization
  - Power normalization
  - Spatial pyramid

- Parameters of IFV for all local features in our system
  - Dimension reduction of local feature (D): 64 dim
  - # of components in GMM (K): 256
  - 5 scales of local patches
  - Spatial pyramid (P): 1x1 + 2x2 + 3x1 = 8
  - Dimension of IFK: 2PKD=262,144 dim

Smaller than another methods (e.g., 1024)



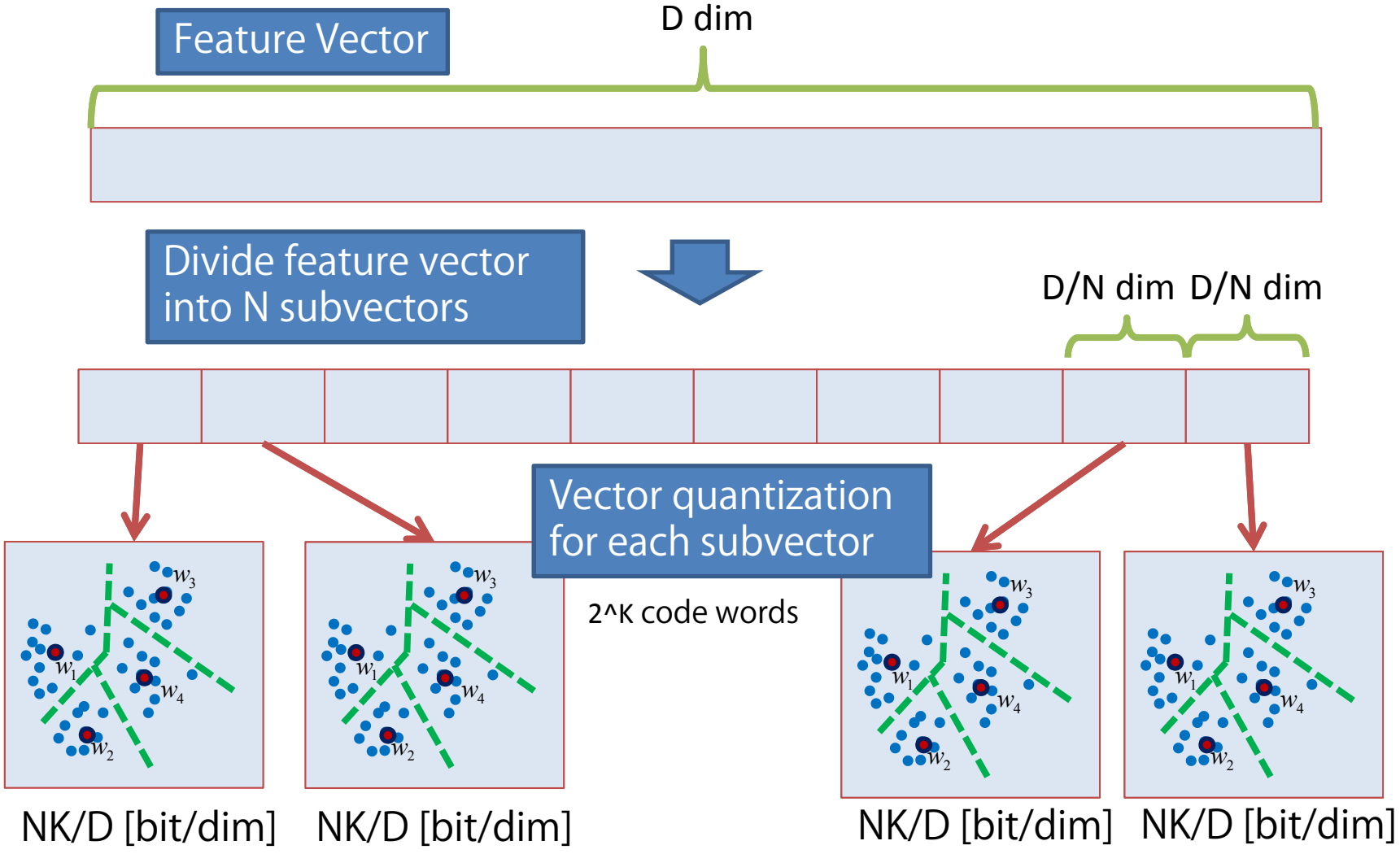
Approx. Graphical Gaussian Vector

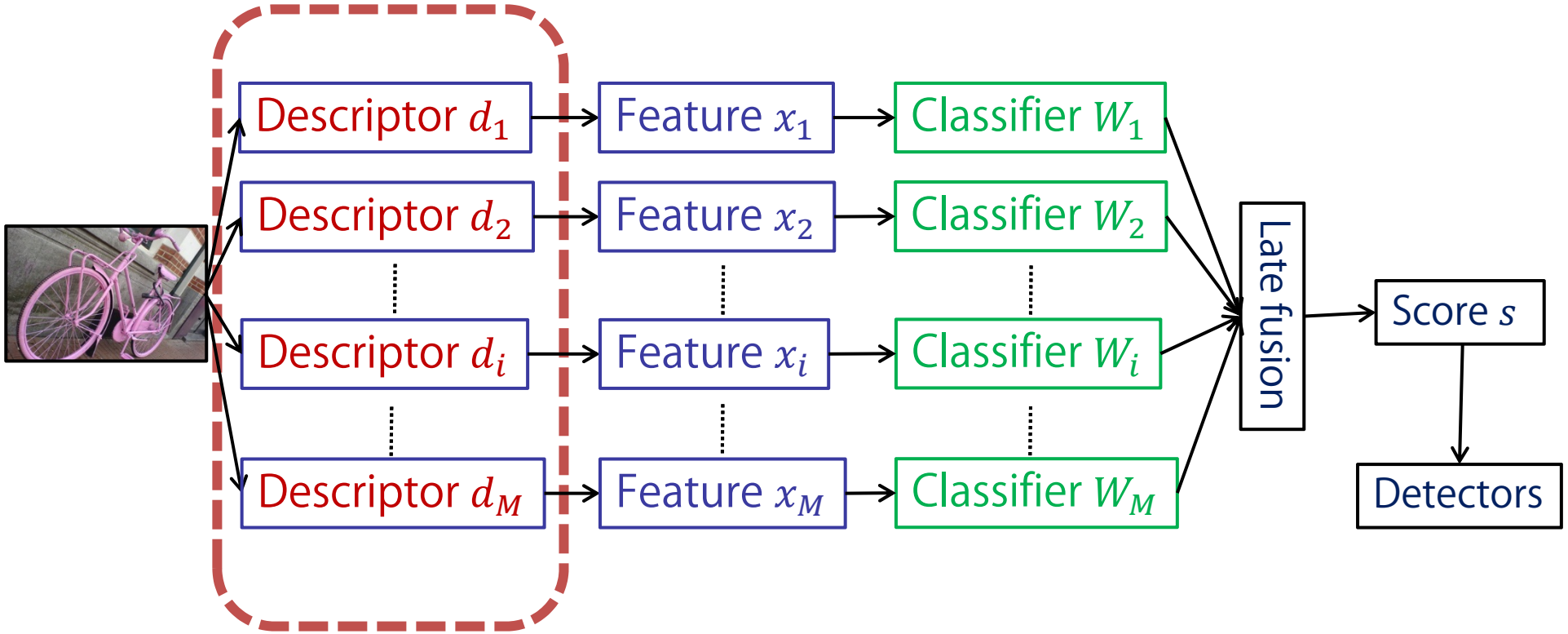


- T. Harada. Graphical Gaussian Vector for Object Categorization. NIPS, 2012.
- Spatial relationship among local descriptors is modeled with Gaussian Markov Random Field (GMRF).
- Approx. GGV is a simpler version of GGV (unpublished work).
- Parameters (not optimized)
  - Dimension reduction of local feature: 80 dim
  - No codebook
  - 3 scales of local patches
  - Spatial pyramid: 1x1 + 2x2 + 3x1 = 8

Product Quantization

- H. Jegou, M. Douze, and C. Schmid. Product Quantization for Nearest Neighbor Search. IEEE Trans. on PAMI, Vol.33, pp.117-128, 2011.
- D/N = 8 dim, 256 centroids in each sub-space, 1 bit per dimension
- We used PQ for task 1, but didn't use it for task 3.




Descriptor d_1
Descriptor d_2
Descriptor d_i
Descriptor d_M
Feature x_1
Feature x_2
Feature x_i
Feature x_M
Classifier W_1
Classifier W_2
Classifier W_i
Classifier W_M
Late fusion
Score s
Detectors

Local Descriptors

- Task 1: 1000 classes categorization
  - Fisher Vector + PQ
    - 4 descriptors: SIFT, C-SIFT, GIST, LBP
    - 5 scales of local patches
    - Sampling: each 6 grid step
  - Approx. GGV + PQ
    - 2 descriptors: SIFT, GIST
    - 3 scales of local patches
    - Sampling: 6 grid step
    - We start this work 1 week before the deadline. This is a preliminary trial.

- Task 2: Detection
  - (Results of Task 1)+DPM root filter
    - HOG

- Task 3: Fine-grained categorization
  - Fisher Vector without PQ
    - 5 descriptors: SIFT, C-SIFT, RGB-SIFT, GIST, LBP
    - 5 scales of local patches
    - Sampling: each 3 or 6 grid step

We used the following codes:

VLFeat
http://www.vlfeat.org/

ColorDescriptor
http://koen.me/research/colordescriptors

GIST
http://people.csail.mit.edu/torralba/code/spatialenvelope/

Thank you!

Computation Time for FV

- Fisher Vector
  - We extracted SIFT, LBP, Gist, and CSIFT at each 6 grid step.
    - All descriptors are reduced to 64 dim with PCA.
    - # of Gaussians: 256, # of spatial pyramid regions: 8

    → 64 dim x 2 x 256 x 8 = 262,144 dim
  - Descriptor extraction + FV coding: 200 CPU days/descriptor
    - 4 kinds of descriptors: 200 x 4 = 800 CPU days!
    - Our computer system: 16 cores x 4 + 12 cores x 5 = 124 cores
    - Total feature extraction time: 800/124 = 6.5 days= 1 week

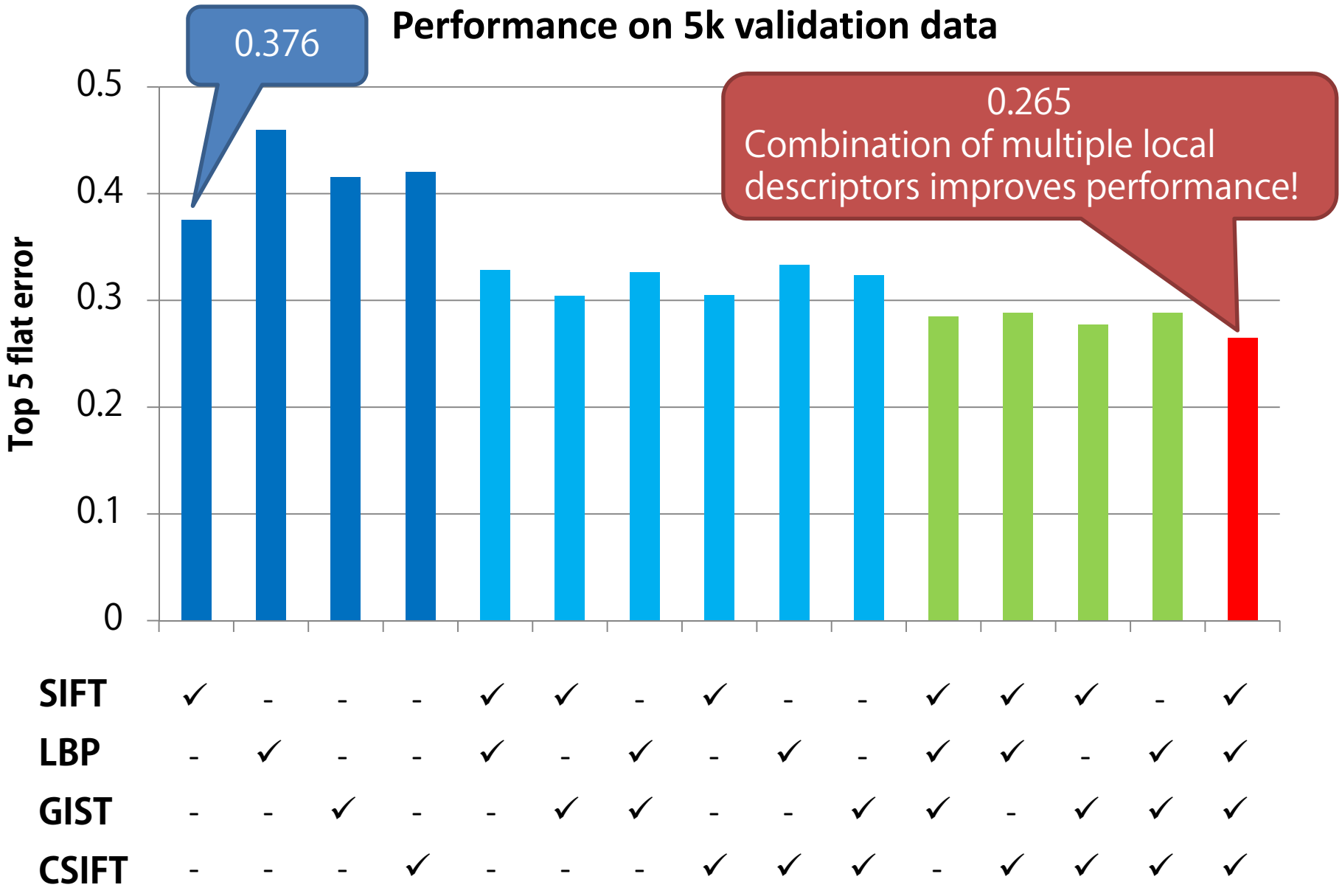
- Product Quantization
  - Compression to fit all FVs in memory: 12 CPU days for learning
    - 1 bit per dimension, 256 centroids in each sub-space
    - Total training time: 1 day by using 4 workstations

- Multi-class Passive-Aggressive algorithm
  - Training time: 16 CPU days per 1 iteration

  → Up to 5 iterations
  → Total training time: 5 days by using 4 workstations

- Computation time from descriptor extraction to model learning
  - 1 week + 1 day + 5 day = about 2 weeks

Results of FVs



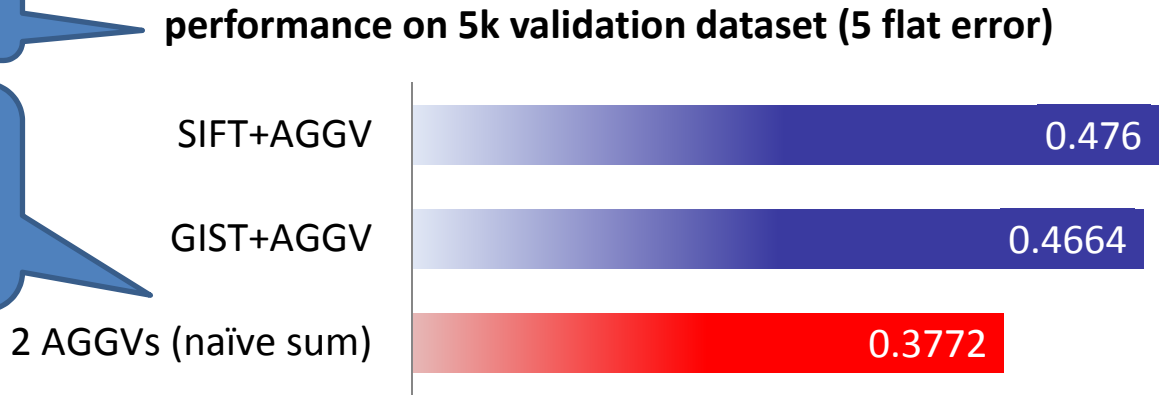
Computation Time and Scores for AGGV

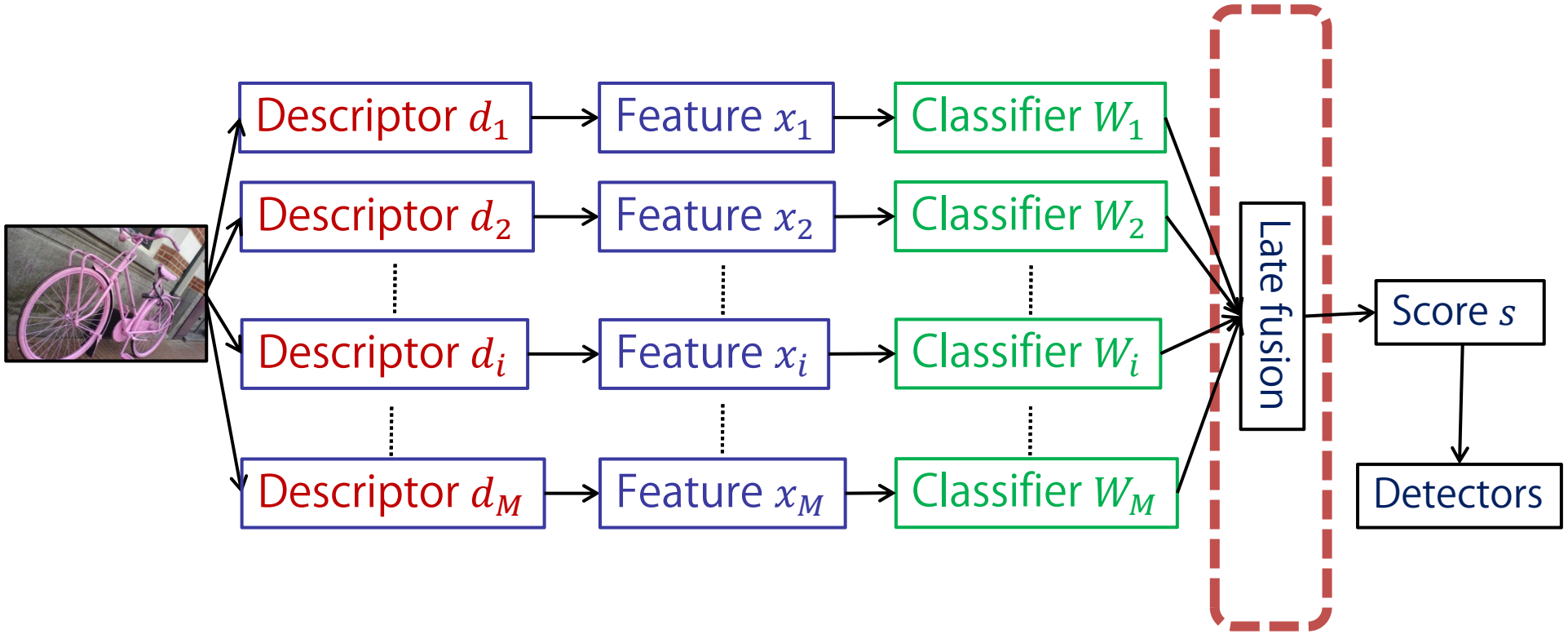
- AGGV: Approximated ver. of GGV [unpublished]
  - We extracted SIFT and Gist as local descriptors at each 6 grid step
    - all reduced to 80 dim (PCA), # of neighbor descriptors: 2, # of spatial pyramid regions: 8

    → (80 dim + 80 dim x (80 dim + 1) / 2 + 80 dim * 80 dim * 2) * 8 = 128,960 dim
  - Descriptor extraction + AGGV coding: 160 CPU days/descriptor
  - Total time: 2x160/124 = about 2.6 days
- Product Quantization
  - Compression to fit all AGGVs in memory: 6 CPU days for learning
    - 1 bit per dimension, 256 centroids in each sub-space
    - Total training time: 0.5 day by using 2 workstations
- Multi-class Passive-Aggressive algorithm
  - Training time: 6 CPU days per 1 iteration→ up to 3 iterations (not converged)
  - Total training time: 1.5 days by using 2 workstations
- Computation time from descriptor extraction to model learning
  - 2.6 days + 0.5 day + 1.5 days = about 5 days

Preliminary results:
Learning didn't finish.

The score is not bad, since this score is comparable to our last year's score, and its comp. cost is smaller than FV.




Descriptor d_1
Feature x_1
Classifier W_1
Descriptor d_2
Feature x_2
Classifier W_2
Descriptor d_i
Feature x_i
Classifier W_i
Descriptor d_M
Feature x_M
Classifier W_M
Late fusion
Score s
Detectors

Late Fusion

- Task 1: 1000 categories classification
  - Comparison between equal weighting and optimized weighting
  - Learning the weights is fast (3 minutes!).
  - This weighting method is unpublished.
  - Weighted sum of scores improves the classification performance.

- Task 3: Fine-grained categorization
  - We used the averaging sum of scores, because we had no time to test the weighted sum of scores. ☹

Performances of Late Fusion

performance on 5k validation dataset (5 flat error)

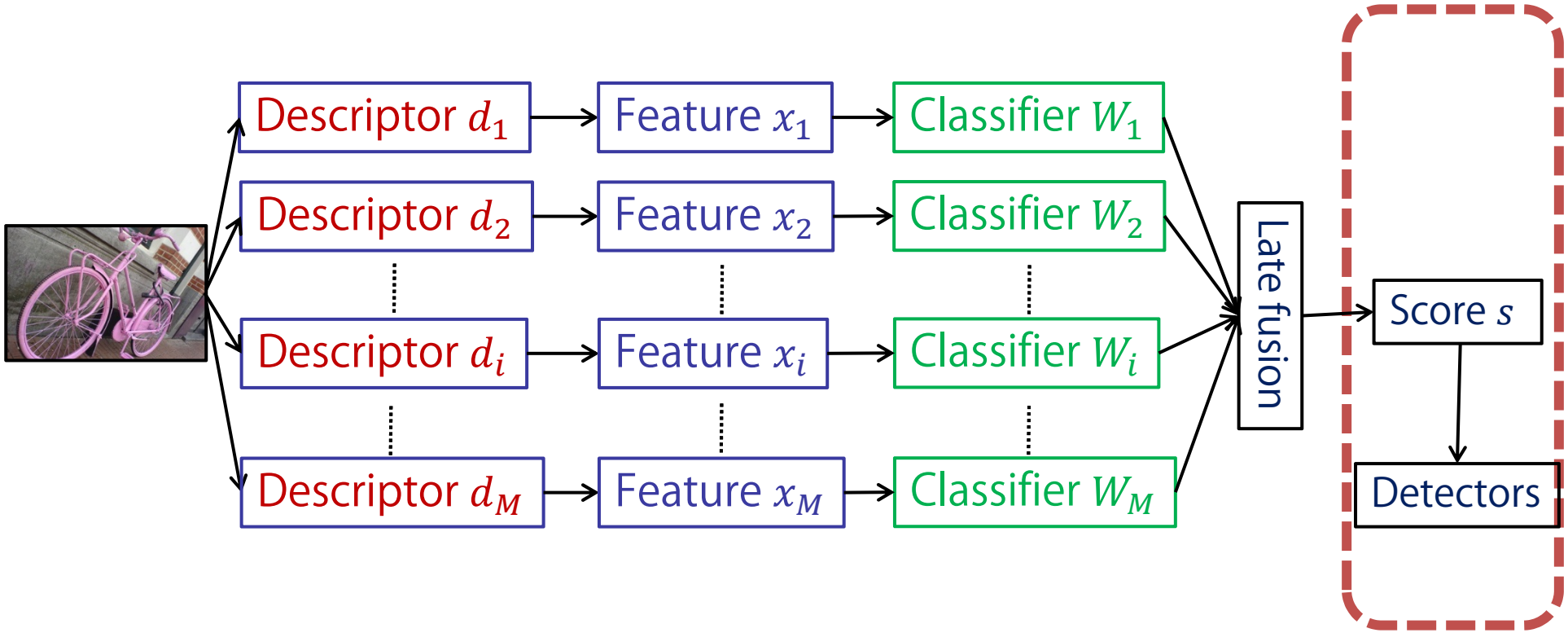
Flat error of 4 FVs (equally weighted sum of scores)

Flat error of 4 FVs + 2 AGGVs (equally weighted sum of scores)
→ worse than only 4 FVs ☹

Flat error of 4 FVs (optimally weighted sum of scores)
Weights are learnt by an online learning (unpublished).

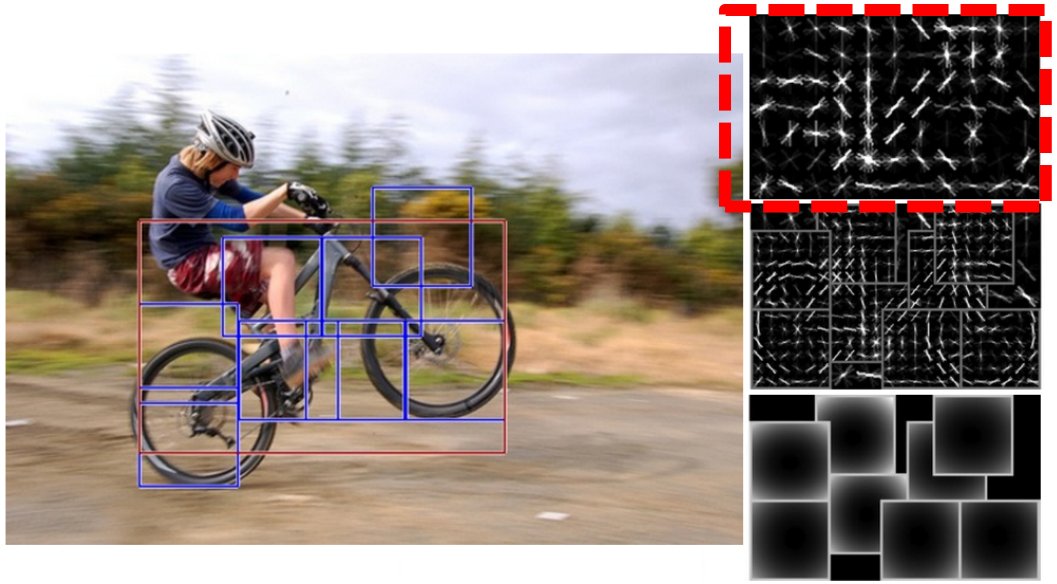
Flat error of 4 FVs + 2 AGGVs (optimally weighted sum of scores)
Weights are learnt by an online learning (unpublished).

AGGV is a complement of FV.


Descriptor d_1
Feature x_1
Classifier W_1
Descriptor d_2
Feature x_2
Classifier W_2
Descriptor d_i
Feature x_i
Classifier W_i
Descriptor d_M
Feature x_M
Classifier W_M
Late fusion
Score s
Detectors

Task 2: Detection

- Cascade object detection with Deformable Part Model [Felzenswalb et al, PAMI, 2010]
  - http://people.cs.uchicago.edu/~rbg/latent/
- Only root filters are used to avoid high computational learning cost.
- Only 108 (/1000) object detectors were learnt in ILSVRC2012.
- The other 892 detectors are same as ours in ILSVRC2011.
- Learning time: 90 CPU min/class→0.5 day by using 1 workstation.
- Detection time: 15 sec/image/class
- We start this task 1 day before the deadline.
- 5 object detectors are nominated using the results from Task 1.

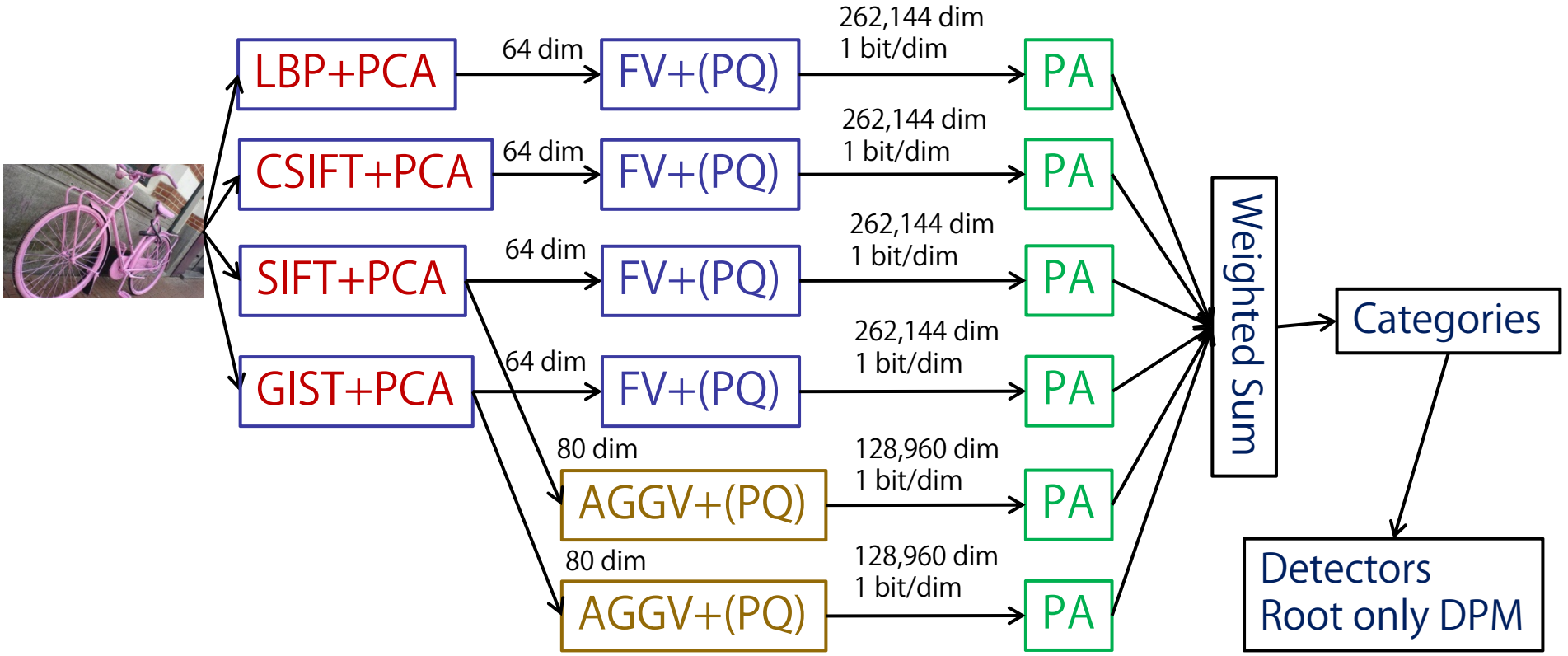


http://people.cs.uchicago.edu/~rbg/latent/

Team	Error
SuperVision	0.335
Oxford_VGG	0.500
ISI (ours)	0.536

The score is not bad, although our method is simple.

Final Pipeline for Task 1 and 2



Team	Flat Error
SuperVision	0.153
ISI (ours)	0.262
OXFORD_VGG	0.270
XRCE/INRIA	0.271

Task 3: Fine-grained categorization

- The pipeline of task 3 is almost same as task 1.

- Image feature: Fisher Vector
  - Without PQ
  - In training, only bounded image regions are used.
  - Local descriptors
    - LBP and CSIFT are sampled at each 6 grid step.
    - GIST, SIFT and RGBSIFT are sampled at each 3 grid step.
    - SIFT reduced to 32 dim, the others reduced to 64 dim (PCA)
    - # of Gaussians: 256
    - # of spatial pyramid regions: 8

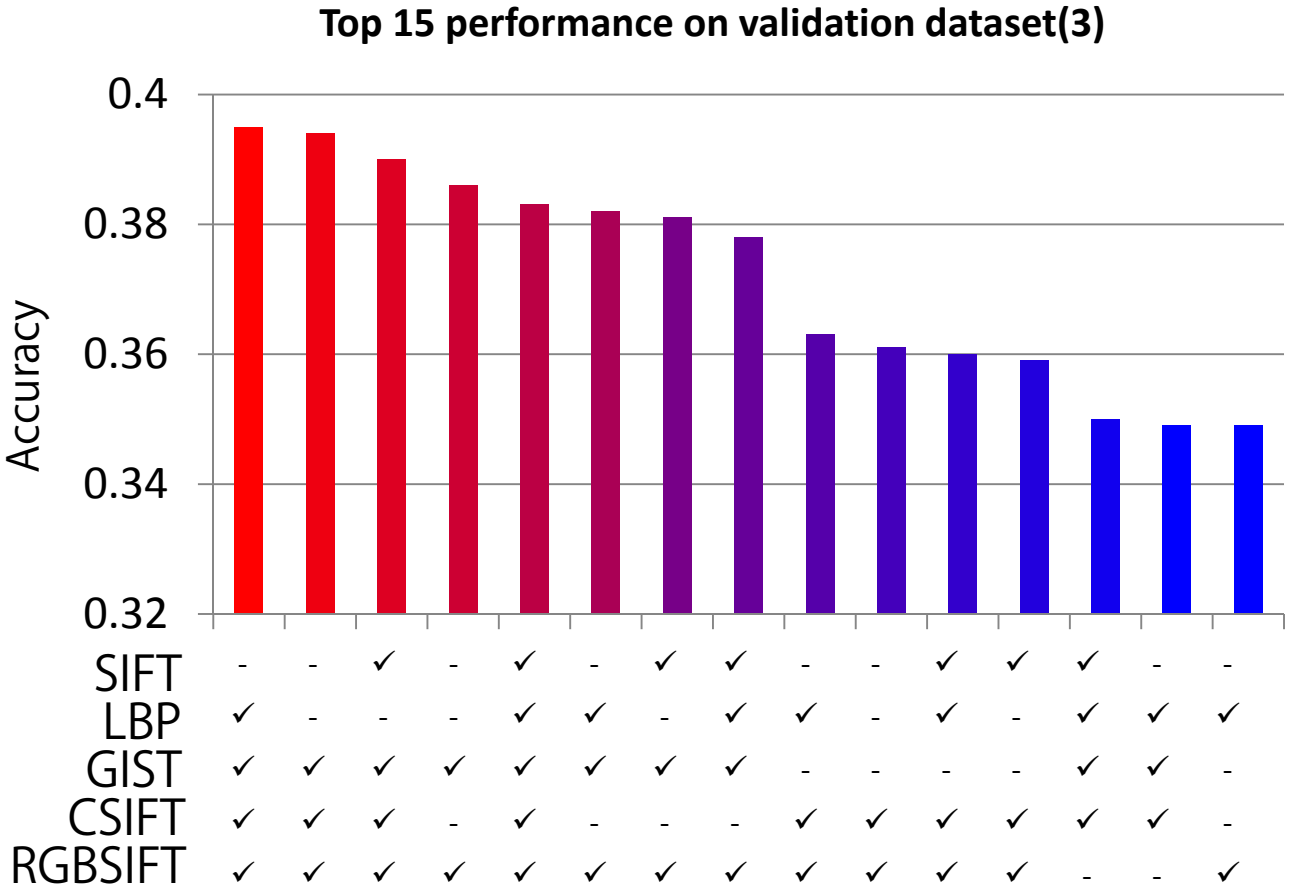
    → 32 dim x 2 x 256 x 8 = 131,072 dim (SIFT)

    → 64 dim x 2 x 256 x 8 = 262,144 dim (others)

- Classifier: Multi-class PA
  - Up to 10 iterations, C = 10^6

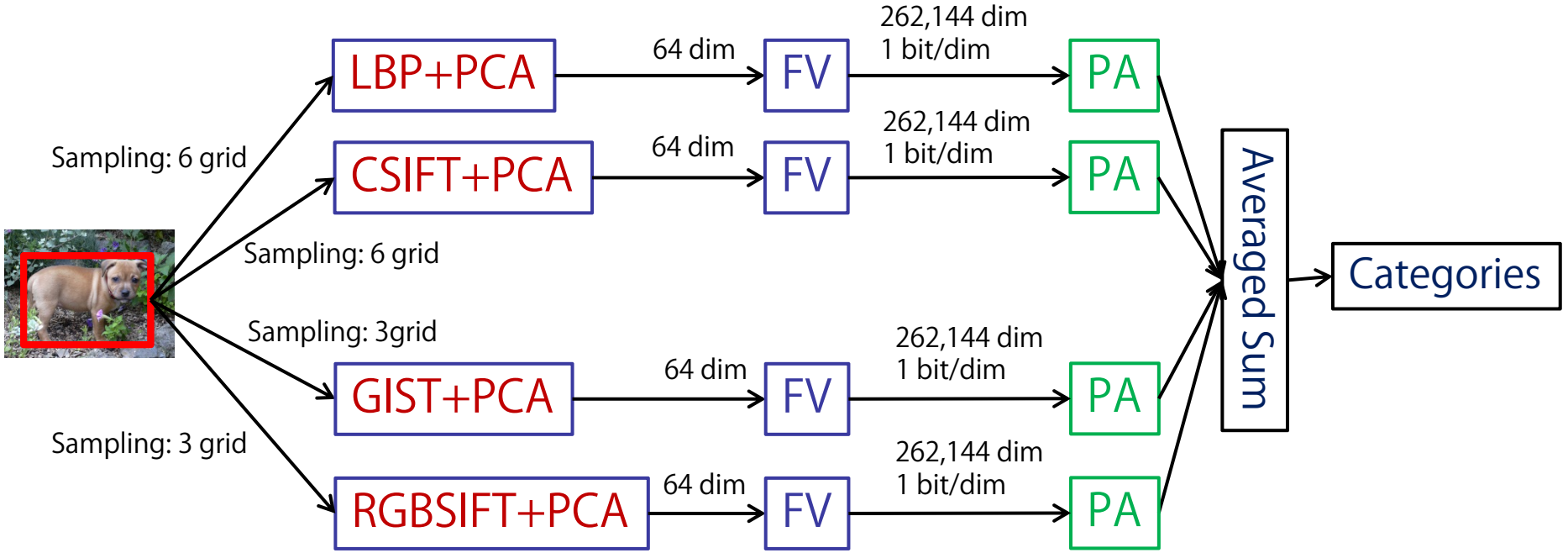
- Late Fusion
  - Equally weighted sum of scores

Results of Task 3



- RGBSIFT+CSIFT+GIST+LBP or RGBSIFT+CSIFT+GIST is good.
- Finer grid sampling improves performance.
- Training image features within the bounding boxes also improves performance.

Final Pipeline for Task 3



Team	mAP
ISI (ours)	0.323
XRCE/INRIA	0.310
Uni Jena	0.246

Summary

- Our pipeline
  - Fisher based features + Multi-class linear classifiers
- Local Descriptors
  - SIFT, CSIFT, RGBSIFT, GIST and LBP
- Image Features
  - FV and AGGV
- Classifier
  - Multi-class Passive-Aggressive Algorithm
- Late Fusion
  - Weighted sum of scores from the classifiers