パターン認識・メディア理解研究会 2月17日

大規模一般画像認識と画像表現
Large-Scale Generic Image Recognition and Image Representation

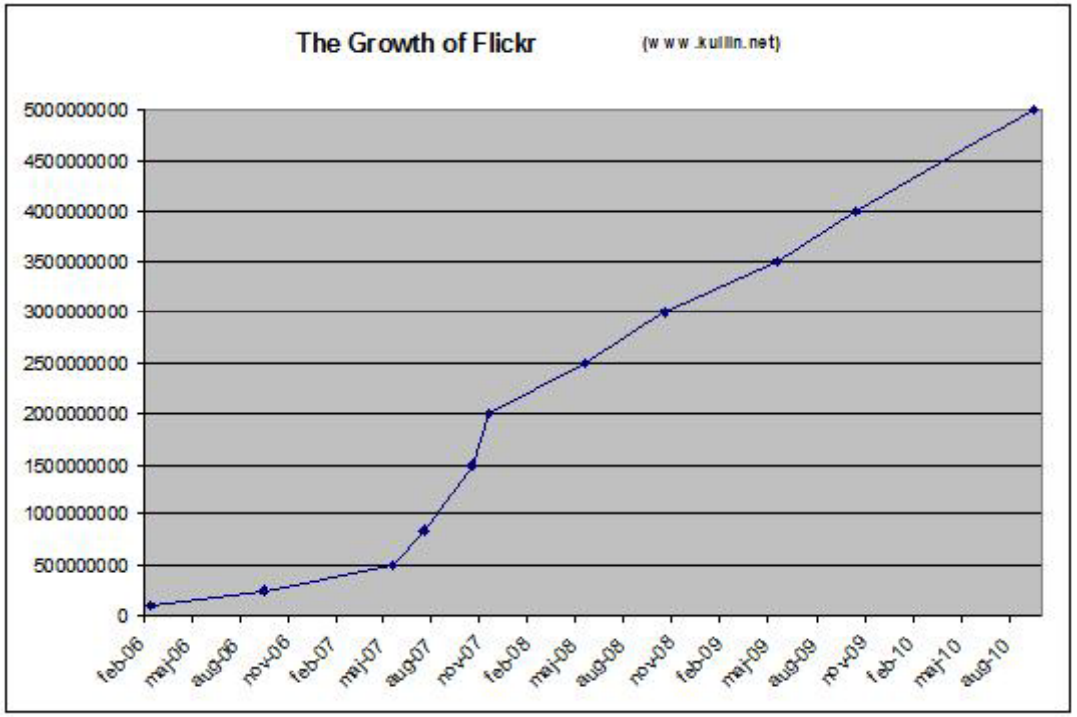
東京大学/JSTさきがけ
原田達也

Flickr reached 5,000,000,000 photos on September 19, 2010.

http://blog.flickr.net/en/2010/09/19/5000000000/

The Growth of Flickr

- Over 5,000,000,000 photos
- 4,596 uploads in the last minute
- 134,362,183 geotagged items



http://www.flickr.com/photos/kullin/4999988381/

Facebook

http://twitter.com/randizuckerberg/status/2218740721857740 8#

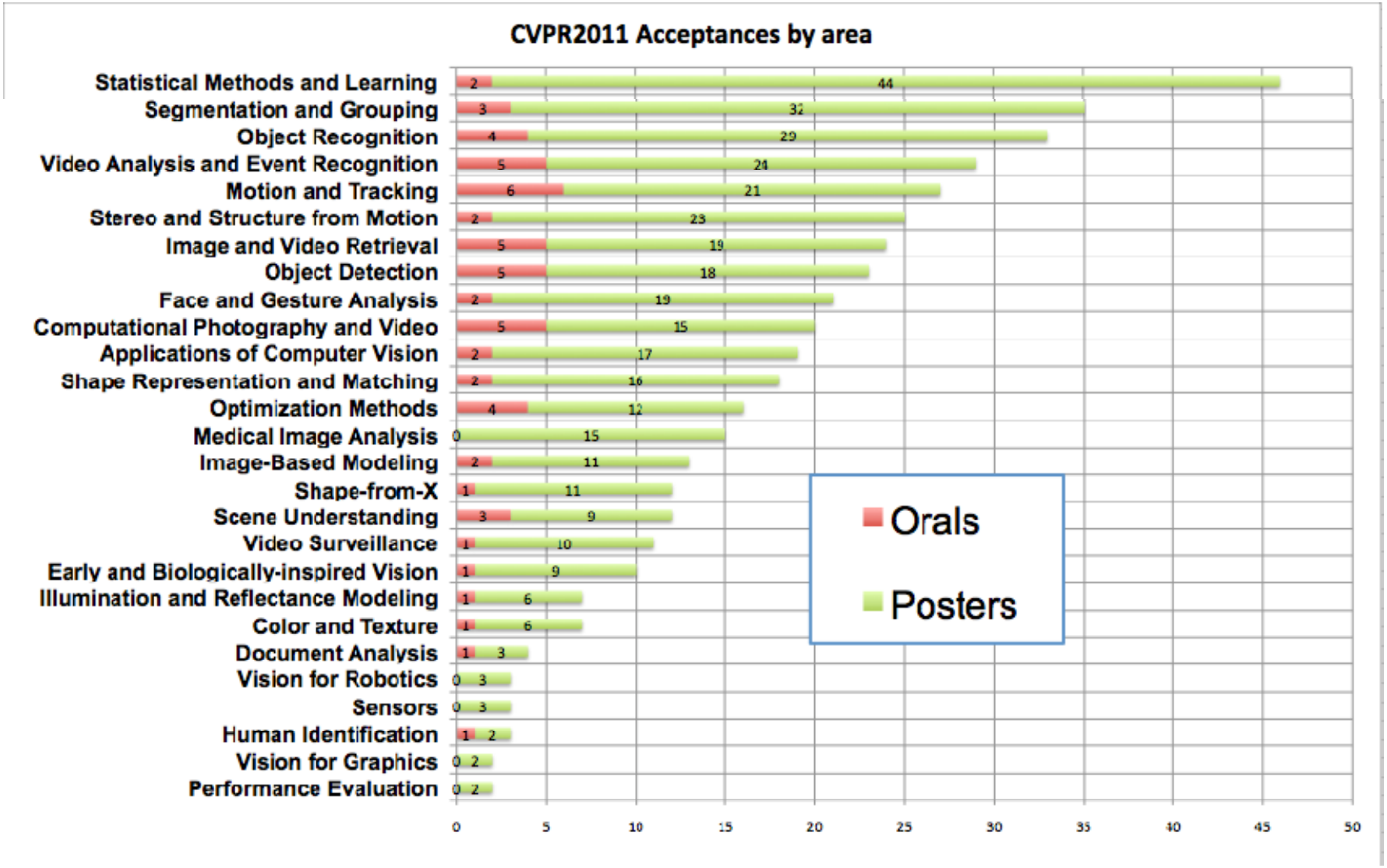
ImageShack：2009年時点で1億枚/月

ECCV2010の統計

- 物体・シーン認識はComputer Visionでも競争が激しい
- 1，2年前の常識が通用しない！

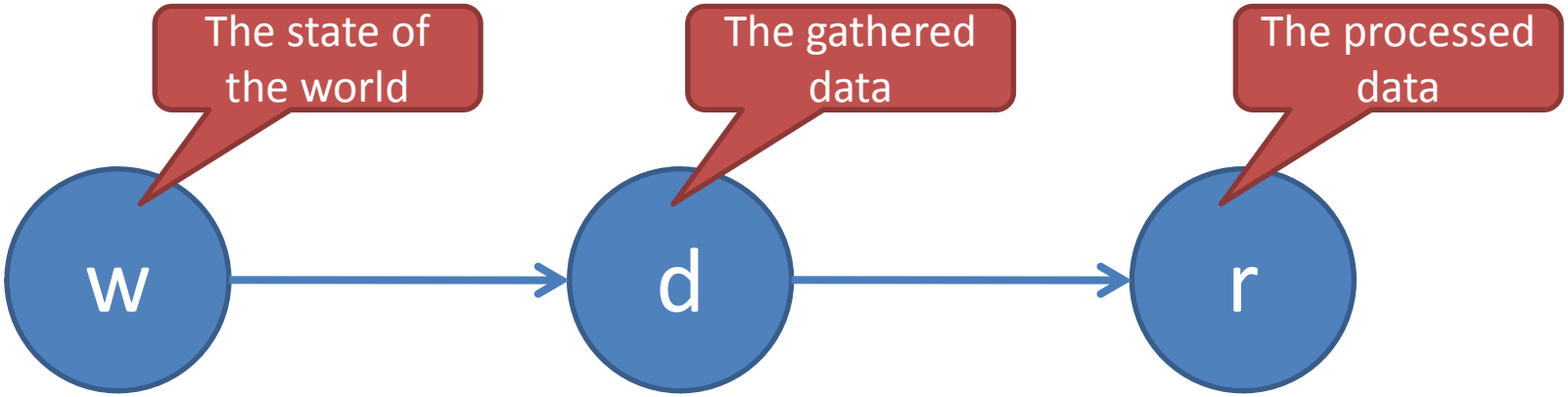
Thematic area	# submitted	% over submitted	# accepted	% over accepted	% acceptance in area
Object and Scene Recognition	192	16.4%	66	20.3%	34.4%
Segmentation and Grouping	129	11.0%	28	8.6%	21.7%
Face, Gesture, Biometrics	125	10.6%	32	9.8%	25.6%
Motion and Tracking	119	10.1%	27	8.3%	22.7%
Statistical Models and Visual Learning	101	8.6%	30	9.2%	29.7%
Matching, Registration, Alignment	90	7.7%	21	6.5%	23.3%
Computational Imaging	74	6.3%	24	7.4%	32.4%
Multi-view Geometry	67	5.7%	24	7.4%	35.8%
Image Features	66	5.6%	17	5.2%	25.8%
Video and Event Characterization	62	5.3%	14	4.3%	22.6%
Shape Representation and Recognition	48	4.1%	19	5.8%	39.6%
Stereo	38	3.2%	4	1.2%	10.5%
Reflectance, Illumination, Color	37	3.2%	14	4.3%	37.8%
Medical Image Analysis	26	2.2%	5	1.5%	19.2%
Total	1174		325		

CVPR2011の統計



http://cvpr2011.org/index.html

The data processing theorem



Markov chain

$$P(w,d,r) = P(w)P(d\,|\,w)P(r\,|\,d)$$

The average information

$$I(W;D) \geq I(W;R)$$

The data processing theorem states that data processing can only destroy information.

David J.C. MacKay. Information Theory, Inference, and Learning Algorithms. Cambridge University Press 2003.

画像認識のプロセス

訓練時

識別時

- 処理を重ねる毎にデータの持つ情報は減少する.
  - データ，特徴抽出，モデルの順に高い質が求められる.
- 従来の画像認識研究の多くはモデル化に重点が置かれていた
  - 小さな実験環境，スモールワールド
- 複雑なモデルは大規模データの前では役に立たない
  - スケーラビリティの重要性
- 高い質のデータ，特徴抽出が適切に行われていればシンプルなモデルで十分な性能が出せる

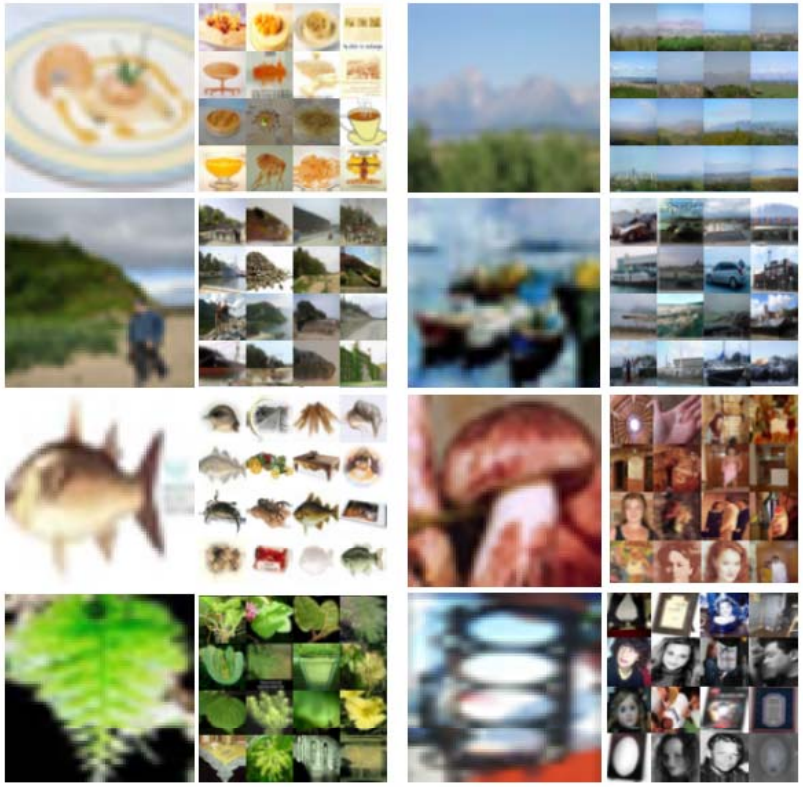
画像認識の分類

- 特定物体認識，Specific Object Recognition
  - データベースには認識対象とする物体の画像をすでに持つことを前提として，入力画像に写る物体とデータベース内の画像を照合すること
- 一般物体認識，Generic Object Recognition
  - データベースに存在しない入力画像の物体のカテゴリを予測すること
- 画像アノテーション，Image Annotation
  - 狭義：複数ラベルが付与された画像データセットから，入力画像に複数のラベルを付与すること
  - 広義：特定物体認識，一般物体認識を含む広い概念

一般画像認識：
上記の分類を包含したセマンティクスレベルの画像認識

TinyImages

- A. Torralba, R. Fergus, W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.30(11), pp. 1958-1970, 2008.
- 8000万枚の画像データセット
- データが大量にあれば最近傍法のみで十分認識可能



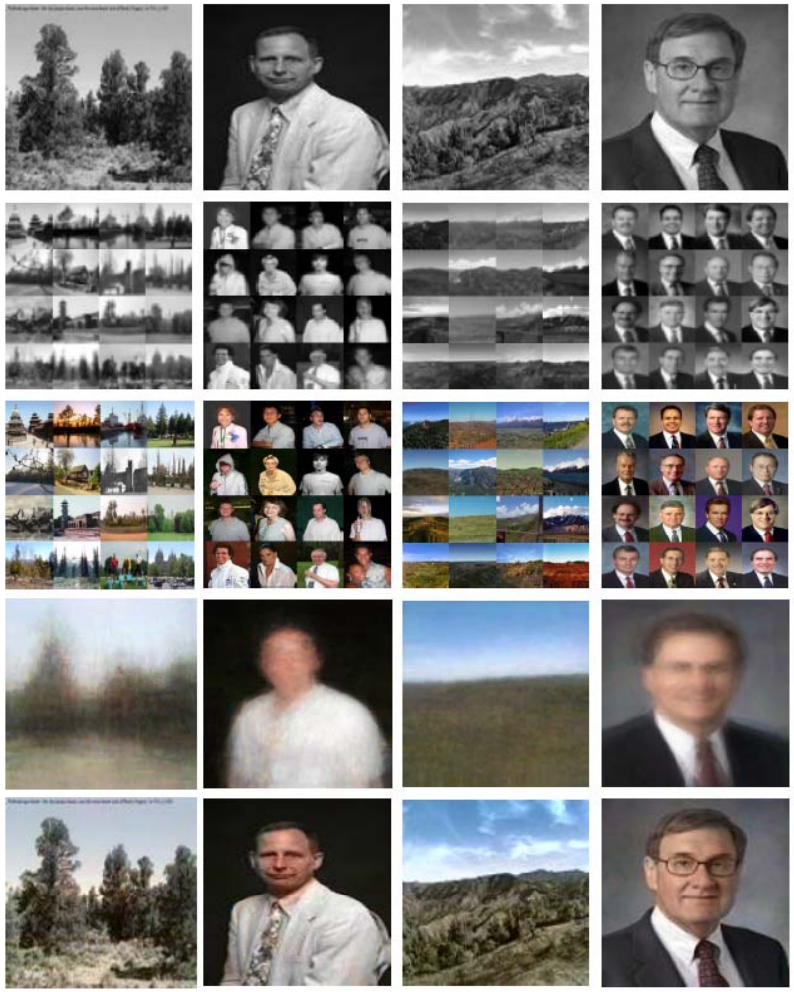


Fig. 1. 1st & 3rd columns: Eight 32 × 32 resolution color images. Despite their low resolution, it is still possible to recognize most of the objects and scenes. These are samples from a large dataset of 10^8 32 × 32 images we collected from the web which spans all visual object classes. 2nd & 4th columns: Collages showing the 16 nearest neighbors within the dataset to each image in the adjacent column. Note the consistency between the neighbors and the query image, having related objects in similar spatial arrangements. The power of the approach comes from the copious amount of data, rather than sophisticated matching methods.

ARISTA

- Xin-Jing Wang, Lei Zhang, Ming Liu, Yi Li, Wei-Ying Ma. ARISTA - Image Search to Annotation on Billions of Web Photos. In CVPR, 2010.
- 20億枚の画像データセットを利用した画像認識
- Near duplicated imageの活用．特定の名称まで認識可能.

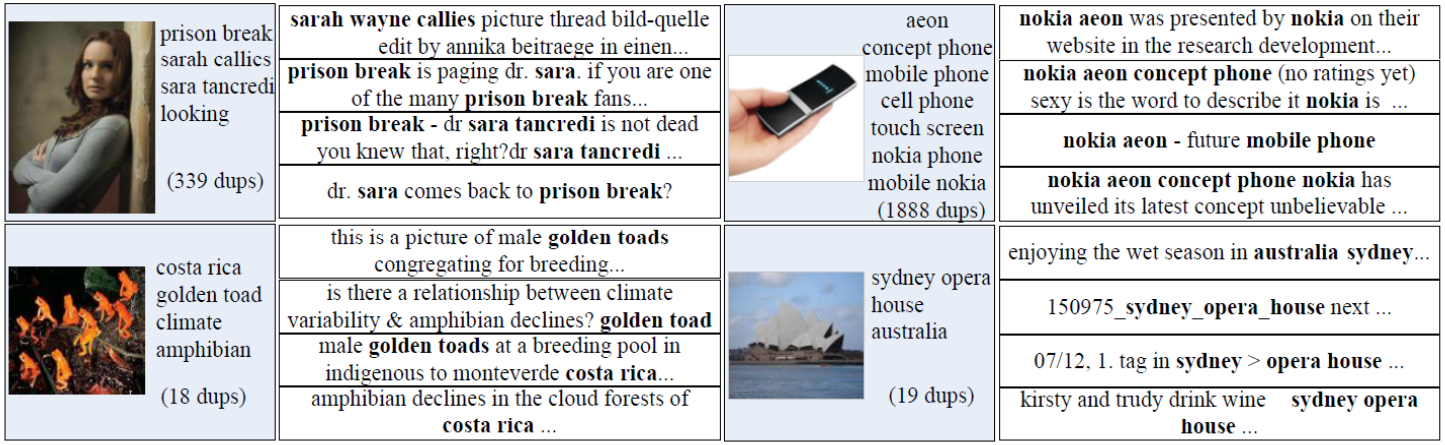


Figure 1. Examples showing that surrounding texts of near-duplicates have common terms which hit the semantics of a query image. The tags inside the image blocks are our annotation outputs. The common terms of each near-duplicate are highlighted in bold. Note that the detected tags are very specific. This is in contrast to most existing works that tend to generate general terms like sky, city, etc.

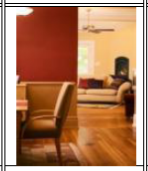
	2.4 M	80M	2B		2.4M	80M	2B
	(no results)	(no results)	*prison break*, **sarah callies**, **sara tancredi**, looking		(no results)	house paint, color	*house*, *paint*, wanta-toos, house painting, hardwood floor, interior design
	**michael jackson**	**michael jackson**, *rock pop*	**michael jackson**, sony music, *cd dvd*, *enter-tainment music*, *pop rock*		linu, *logo*	server, *software*, *logo*, credit card processing, *op-erating system*	**penguin**, *open source*, *virtual server*, *logo*, *operating system*
	**ipod touch**	**apple ipod**, *mp3 player*, **iphone**, wi fi, touch screen	**apple ipod**, *mp3 player*, wi fi, media player, **touch screen**, **mobile phone**		(no results)	(no results)	**bald eagle**, **haliaee-tus leucocephalus**, endangered species, fish wildlife, *eagle flight*

Figure 9. Annotation examples vs. dataset size. Bold-faced tags are perfect terms labeled by human subjects and italic ones are correct terms. Due to space limit, only the top five tags are shown. This figure suggests that larger dataset size ensures more accurate tags.

ImageNet

- ImageNet
  - 12 million images, 15 thousand categories
  - Image found via web searches for WordNet noun synsets
  - Hand verified using Mechanical
  - All new data for validation and testing this year
- WordNet
  - Source of fraction of English nouns
  - Also used tthe labels
  - Semantic hierarchy
  - Contains large o collect other datasets like tiny images (Torralba et al)
  - Note that categorization is not the end goal, but should provide information for other tasks, so idiosyncrasies of WordNet may be less critical

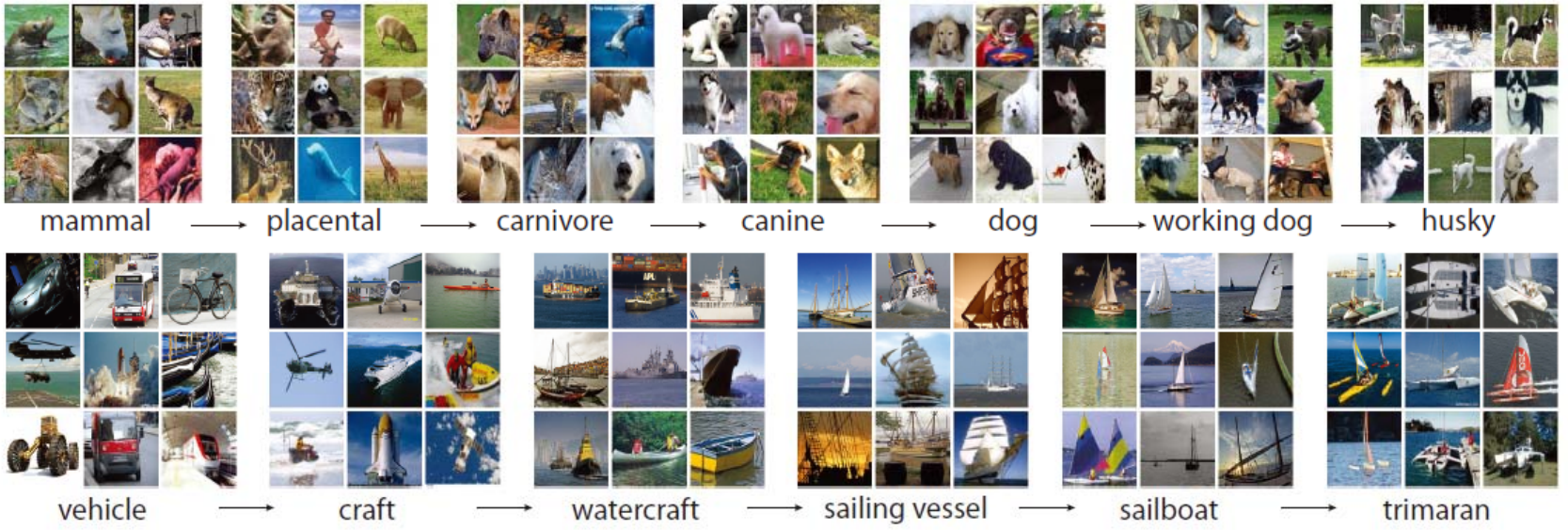


Figure 1: A snapshot of two root-to-leaf branches of ImageNet: the **top** row is from the mammal subtree; the **bottom** row is from the vehicle subtree. For each synset, 9 randomly sampled images are presented.

Deng et al., CVPR2009

大規模画像認識コンテスト

- ImageNet
  - http://www.image-net.org/challenges/LSVRC/2010/index

- 今年度の挑戦課題
  - 画像識別
  - 1000 カテゴリ
  - 120万枚の訓練画像
  - 5万枚の検証画像
  - 15万枚のテスト画像

テスト画像



















カテゴリ

french fries mashed potato black olive face powder crab apple Granny Smith strawberry blueberry cranberry currant blackberry raspberry persimmon mul berry orange kumquat lemon grapefruit plum fig pineapple banana jackfruit cher ry grape custard
apple durian mango elderberry guava litchi pomegranate quince kidney
bean soy green pea chickpea chard lettuce cress spinach bell
pepper pimento jalapeno cherry tomato parsnip turnip mustard bok choy head
cabbage broccoli cauliflower brussels sprouts zucchini spaghetti squash acorn
squash butternut squash cucumber artichoke asparagus green
onion shallot leek cardoon celery mushroom pumpkin cliff lunar
crater valley alp volcano promontory sandbar dune coral
reef lakeside seashore geyser bakery juniper
berry gourd acorn olive hip ear pumpkin seed sunflower seed coffee
bean rapeseed corn buckeye bean peanut walnut cashew chestnut hazelnut coco
nut pecan pistachio lentil pea peanut okra sunflower lesser celandine wood
anemone blue columbine delphinium nigella calla lily sandwort pink baby‘s
breath ice plant globe amaranth four o’clock Virginia spring beauty wallflower damask
violet candytuft Iceland poppy prickly poppy oriental poppy celandine blue
poppy Welsh poppy celandine poppy corydalis pearly everlasting strawflower yellow
chamomile dusty miller tansy daisy common marigold China
aster cornflower chrysanthemum mistflower など

カテゴリ （Google翻訳後，，，）

フライドポテトは ジャガイモのマッシュアップ ブラックオリーブ 白粉の カニリンゴ グラニースミスの イチゴ ブルーベリー クランベリー カラント ブラックベリー ラズベリーの 柿 桑 オレンジ キンカン レモン グレープフルーツ 梅 図 パイナップル バナナ ジャックフルーツ 桜の ブドウ カスタードアップルを ドリアン マンゴー ニワトコ グアバ ライチ ザクロ カリン インゲン 大豆 グリーンピースの ひよこ豆の フダンソウの レタス クレソンの ほうれん草 ピーマン ピーマン ハラペーニョ ミニトマト パースニップ カブ マスタード ボクは菜 ヘッドキャベツ ブロッコリー カリフラワー 芽キャベツの ズッキーニの スパゲティは、スカッシュ スカッシュドングリ バタースカッシュ キュウリは アーティチョーク アスパラガス ねぎ エシャロットを ネギ カルドンの セロリ マッシュルーム カボチャの 崖の 月面のクレーターの 谷 アルプスの 火山 岬 砂州の 砂丘に サンゴ礁を 湖畔 海岸 間欠泉の パン屋さん ジュニパーベリーは、 ヒョウタン ドングリ オリーブ ヒップ 耳 カボチャの種 ヒマワリの種 コーヒー豆の 菜種 トウモロコシ バックアイ 豆 ピーナッツ クルミ カシューナッツ 栗 ヘーゼルナッツ ココナッツ ピーカンナッツ ピスタチオ 豆 豆 ピーナッツ オクラ ヒマワリ 低いクサノオウの 木のクマノミ ブルーコロンバイン デルフィニウム ニゲラの カーラリリー sandwort ピンク 赤ちゃんの呼吸 アイスプラントの 世界をアマランサス 四oの'クロック バージニア春の美しさの 壁の花の ダマスクバイオレット キャンディータフト アイスランドポピー 厄介ポピー オリエンタルポピー クサノオウ 青いケシ ウェルシュポピー クサノオウケシ キケマン 真珠のような永遠の ストローフィールド 黄色のカモミール ダスティーミラーの ヨモギギクに デイジーチェーン 共通マリーゴールド エゾギク コーンフラワー キク キク科ヒヨドリバナ属の多年草の など

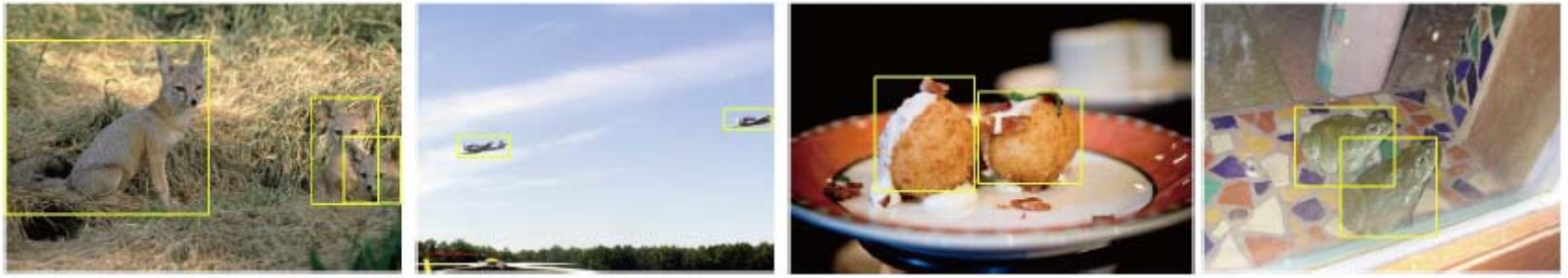
結果

- 参加チーム
  - 150以上参加，最終的には11チームの結果報告
    - データ規模が膨大！ダウンロードだけで1週間以上！
    - ベースラインの結果が決まっている

- 順位
  1. NEC-UIUC, USA
  2. XRCE, France
  3. ISIL, University of Tokyo, Japan
  4. UC Irvine, USA
  5. MIT, USA
  6. Nanyang Technological University, Singapore
  7. LIG Grenoble, France
  8. IBM-ensemble, USA
  9. SRI International, USA
  10. National Institute of Informatics, Tokyo, Japan
  11. Harbin Institute of Technology, China

2011 Large Scale Visual Recognition Challenge!

- Data:
  - Bounding boxes
  - Not yet full parsing of images
- Task:
  - Image categorization
  - Object detection/localization



http://www.image-net.org/challenges/LSVRC/2010/pascal_ilsvrc.pdf

Pipeline

NEC: LLC, SV
XRCE: Fisher Vector

Image representation
画像表現

Local Descriptors

Semi-local feature

Spatial Pyramid

Pooling

Image Feature

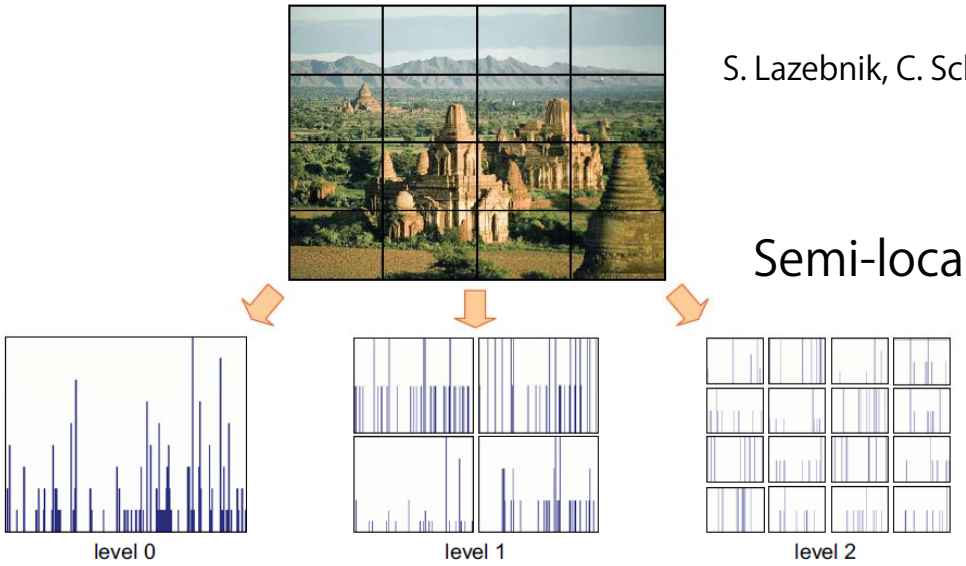
Linear Classifier

Category

NEC: HOG, LBP
XRCE: SIFT, Color

Linear SVMs

Spatial Pyramid Representation



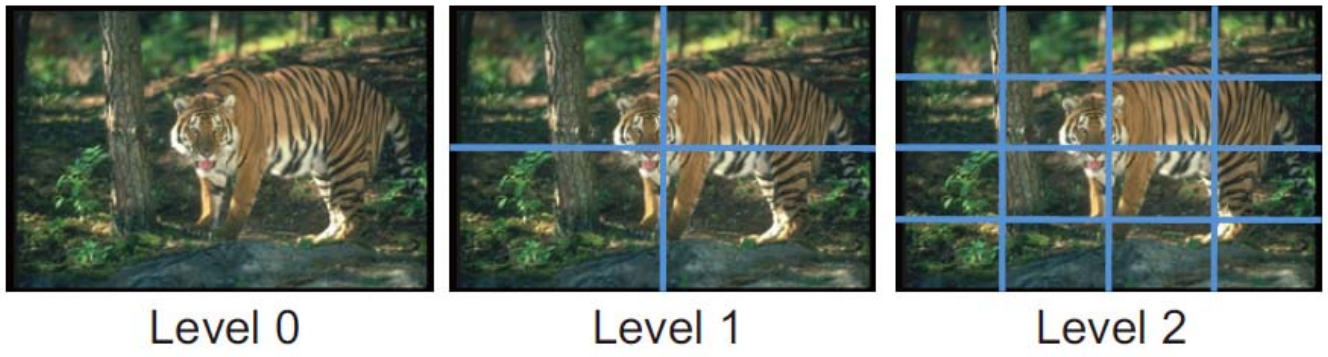
S. Lazebnik, C. Schmid, and J. Ponce

Semi-local feature (SLF)

Fig. 1.1. A schematic illustration of the spatial pyramid representation. A spatial pyramid is a collection of orderless feature histograms computed over cells defined by a multi-level recursive image decomposition. At level 0, the decomposition consists of just a single cell, and the representation is equivalent to a standard bag of features. At level 1, the image is subdivided into four quadrants, yielding four feature histograms, and so on. Spatial pyramids can be matched using the *pyramid kernel*, which weights features at higher levels more highly, reflecting the fact that higher levels localize the features more precisely (see Section 1.2).

- Level0: Global featureと同じ
- Level1: 2x2のcellに分割し各cellでSLFを計算
- Level2: 4x4のcellに分割し各cellでSLFを計算

Variations of SPR



(a) Spatial pyramid representation in [12]

(b) Spatial pyramid representation in [15]

- SPR
  - アドホック，高次元
- 解決策
  - T. Harada, Y. Ushiku, Y. Yamashita, and Y. Kuniyoshi. Discriminative Spatial Pyramid. In CVPR, to appear, 2011.

Bag of Visual Words?

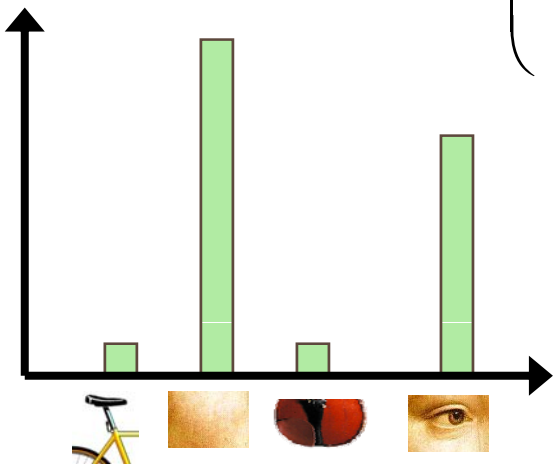
Li Fei Fei, cvpr07 tutorial
より抜粋

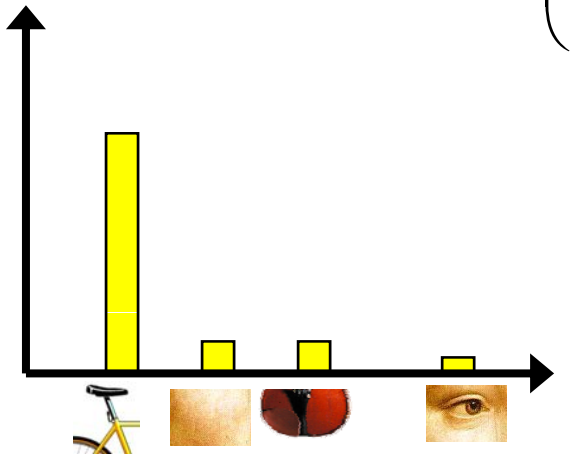
Visual words

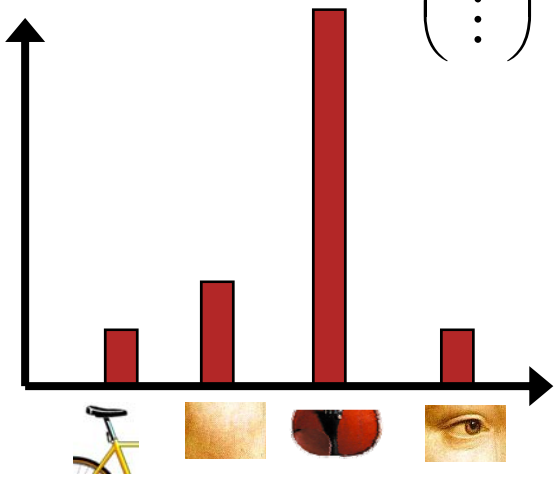
$$\begin{pmatrix} 1 \\ 10 \\ 1 \\ 7 \\ \vdots \end{pmatrix}$$

$$\begin{pmatrix} 7 \\ 2 \\ 2 \\ 1 \\ \vdots \end{pmatrix}$$

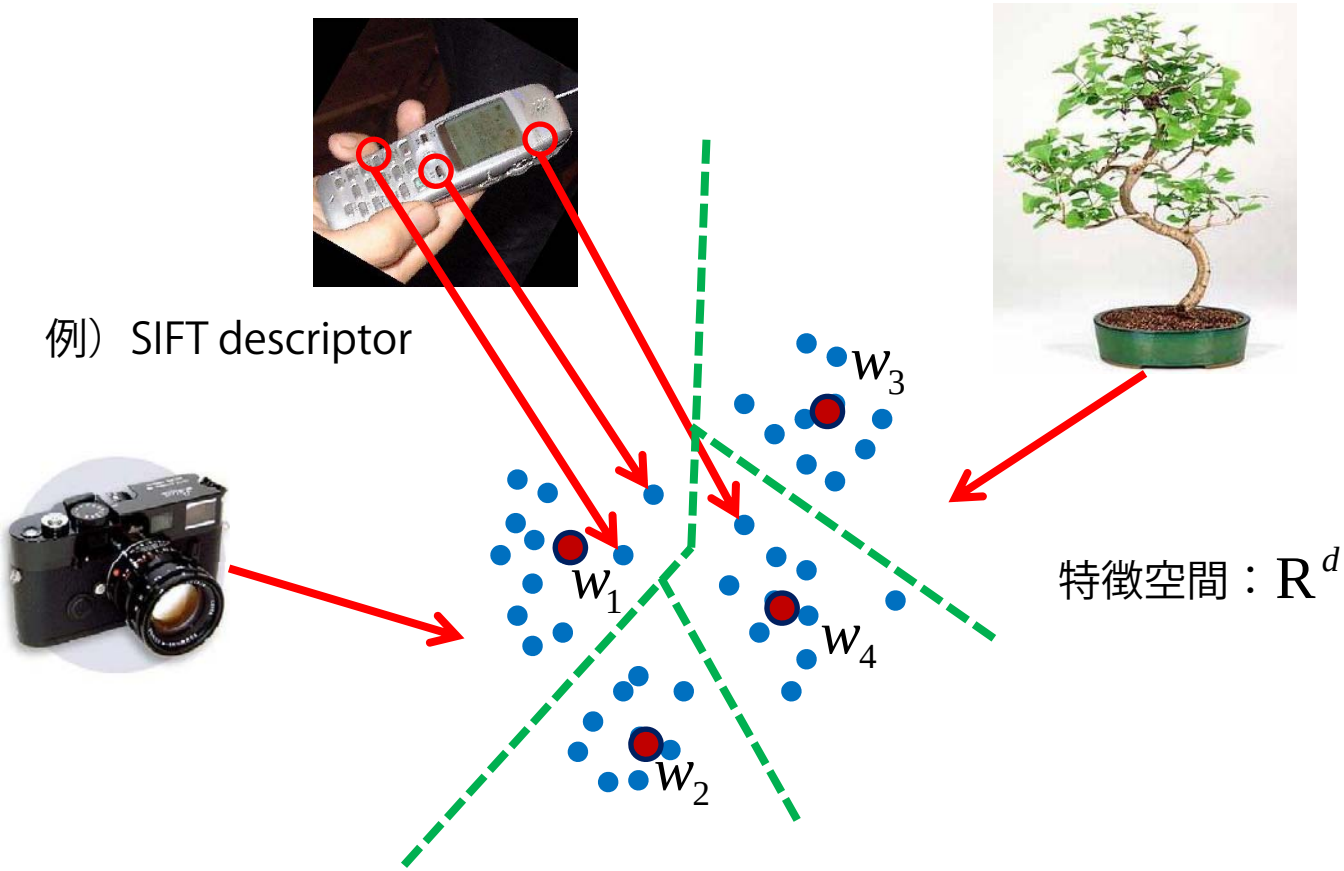
$$\begin{pmatrix} 3 \\ 4 \\ 10 \\ 2 \\ \vdots \end{pmatrix}$$







Code wordsの生成：clustering



- ベクトル量子化と呼ばれるプロセス
- 一般的にk-meansによるクラスタリング
  - 階層的クラスタリング：Vocabulary Tree
- 局所記述子にはSIFTがよく用いられる
  - もちろんSURFやRGB，Self Similarityでもよい

BoFのGMM利用による改善

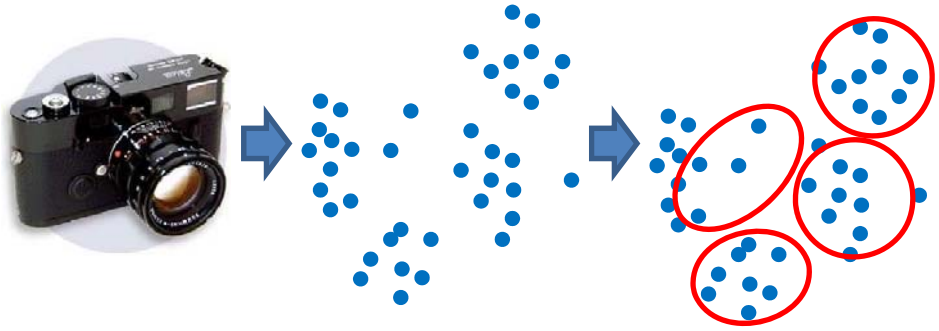


Image　　Local descriptors in feature space　　PDF estimation

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \sum_{k=1}^{K} \pi_k p_k(\boldsymbol{x})$$

$$\gamma_n(k) = p(k|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) = \frac{\pi_k p_k(\boldsymbol{x}_n)}{\sum_{j=1}^{K} \pi_j p_j(\boldsymbol{x}_n)}$$

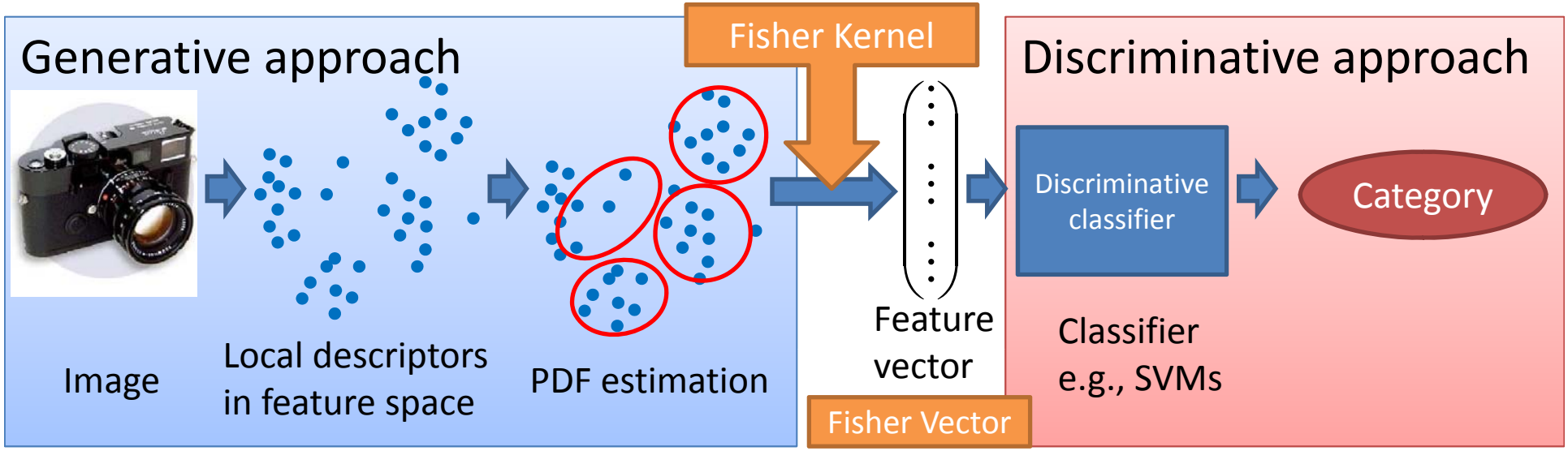
$$\boldsymbol{f} = \frac{1}{N} \sum_{n=1}^{N} [\gamma_n(1), \cdots, \gamma_n(K)]^\top \in R^K$$

- メリット
  - 混合ガウス分布を構成する各ガウス分布がそれぞれ共分散を持つため，共分散を考慮した距離計量を利用できる
  - 混合ガウス分布では局所特徴と多くのコードワードとの関係を表現できるので，特徴空間における局所特徴の位置に関する情報をエンコードできる

- デメリット
  - 混合ガウス分布表現はBoF と比較してパラメータが多い
    - 混合ガウス分布：O(K(D^2/2 + D))，BoF：O(KD)
  - 混合ガウス分布は訓練データに対して過剰適合する可能性があり，学習時に正則化を行う必要

フィッシャーベクトル

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



- 混合ガウス分布を用いた確率密度分布推定によるBoF の改良
  - 生成モデル（generative model）
- 生成モデルを識別的なアプローチに適応可能なより洗練された手法があれば識別性能の改善につながる.
- フィッシャーカーネル（Fisher Kernel)
  - 生成的アプローチ（generative approach）と識別的アプローチ（discriminative approach）を結合させる強力な枠組み
  - 手順

1. 局所特徴を生成する確率密度分布から導出される勾配ベクトルの計算
2. 画像を表現する一つの特徴ベクトルの計算
   →フィッシャーベクトル（Fisher Vector）
3. 得られた特徴ベクトルを識別的分類機に入力する.

フィッシャーベクトルのメリット

- 豊かな特徴ベクトル表現
  - BoF と比較してフィッシャーカーネルを利用するメリットは，コードブックサイズが同じであればより要素数の多い特徴ベクトルが得られる.
    - コードブックサイズ：K，局所特徴の次元：d
    - BoFの次元：K
    - フィッシャーベクトル： (2d+1)K-1
  - 特徴ベクトルの表現する情報が多いため計算コストの高いカーネル法を利用して高次元空間へ射影する必要がなく，線形識別機でも十分な識別性能を出すことが可能となる.

フィッシャーベクトル詳細

- 局所特徴群

$$\mathcal{X} = \{\boldsymbol{x}_n \in R^D\}_{n=1}^N$$

- あらゆる画像内容を表現する局所特徴の確率密度分布

$$u_\theta$$

- 対数尤度の勾配

$$G_\theta^{\mathcal{X}} = \frac{1}{N}\nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta})$$

  - データに最も適合するように確率密度関数のパラメータが修正すべき方向を表現
  - 異なるデータサイズ集合をパラメータ数に依存した特定の長さの特徴ベクトルに変換
  - 内積を利用する識別機には正規化が必要！！

- フィッシャー情報行列

$$F_\theta = E_X[\nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta})\nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta})^\top]$$

- フィッシャーベクトル（Fisher Vector）

$$\mathcal{G}_\theta^X = F_\theta^{-1/2}\nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta})$$

フィッシャー情報行列による対数尤度の勾配の正規化

混合ガウス分布におけるフィッシャーベクトル

- 確率密度分布を混合ガウス分布とする
  - 共分散行列は対角行列と仮定

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \sum_{k=1}^{K} \pi_k p_k(\boldsymbol{x})$$

あらゆる画像を生成する確率密度分布

- 対数尤度の微分

$$G_\theta^{\mathcal{X}} = \frac{1}{N} \nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta})$$

画像1枚から得られる局所特徴の集合

負担率：局所特徴x_nがGMMのコンポーネントkに属する確率

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \pi_k} = \sum_{n=1}^{N} \left[\frac{\gamma_n(k)}{\pi_k} - \frac{\gamma_n(1)}{\pi_1} \right]$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k^d} = \sum_{n=1}^{N} \gamma_n(k) \left[\frac{\boldsymbol{x}_n^d - \boldsymbol{\mu}_k^d}{(\boldsymbol{\sigma}_k^d)^2} \right]$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\sigma}_k^d} = \sum_{n=1}^{N} \gamma_n(k) \left[\frac{(\boldsymbol{x}_n^d - \boldsymbol{\mu}_k^d)^2}{(\boldsymbol{\sigma}_k^d)^3} - \frac{1}{\boldsymbol{\sigma}_k^d} \right]$$

GMMのBoFとほぼ同じ

$$\boldsymbol{f} = \frac{1}{N} \sum_{n=1}^{N} [\gamma_n(1), \cdots, \gamma_n(K)]^\top \in R^K$$

局所特徴x_nとGMMの各コンポーネントkの平均との差分

- 混合比：BoFとほぼ同じ
- 平均，分散：あらゆる画像を表現するpdfの平均との差分
- BoFは0次，Fisher Vectorは1次，2次の統計量を含む
- 分散の表現は平均の表現とあまり差がない？本来は各コンポーネント間の相関が必要

フィッシャー情報行列

- フィッシャー情報行列

$$F_\theta = E_X[\nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta}) \nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta})^\top]$$

- 混合ガウス分布において近似的に閉じた解が得られる
- 仮定
  - フィッシャー情報行列は対角行列
  - 共分散行列は対角行列
  - 負担率はピーキー
  - 一枚の画像から得られる局所特徴数は一定

$$\frac{\partial \mathcal{L}(\mathcal{X}|\theta)}{\partial \pi_k} \Rightarrow f_{\pi_k} = N\left(\frac{1}{\pi_k} + \frac{1}{\pi_1}\right)$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\theta)}{\partial \mu_k^d} \Rightarrow f_{\mu_k^d} = \frac{N\pi_k}{(\boldsymbol{\sigma}_k^d)^2}$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\theta)}{\partial \sigma_k^d} \Rightarrow f_{\sigma_k^d} = \frac{2N\pi_k}{(\boldsymbol{\sigma}_k^d)^2}$$

フィッシャー情報行列の要素

フィッシャーベクトルの直感的解釈

http://www.image-net.org/challenges/LSVRC/2010/ILSVRC2010_XRCE.pdf

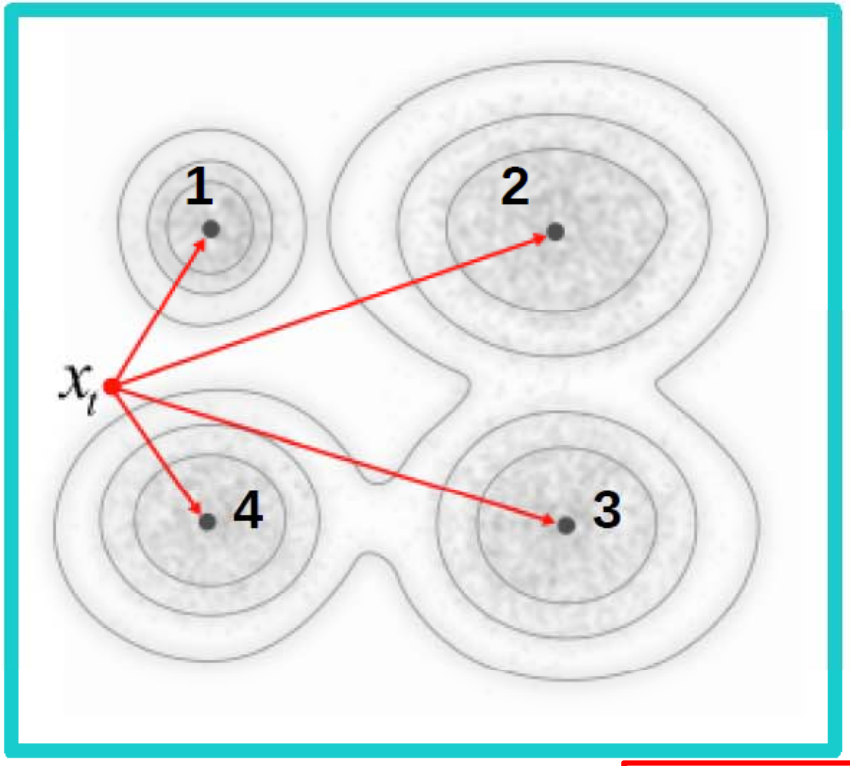
BOV

Hard Assignment

[0 0 0 1]

Soft Assignment

[.3 .1 .1 .5]



Fisher Vector

Gradient wrt w

[.15 -.2 -.35 .2]

Gradient wrt mean

[.8 -1.5 -3.7 -1.3 -3.8 1.2 -.9 1.4]

Gradient wrt var

[-1.2 -.9 1.4 -.8 1.5 -3.7 1.3 -3.8]

Bag of Visual Words (GMM)

$$\boldsymbol{f} = \frac{1}{N}\sum_{n=1}^{N}[\gamma_n(1), \cdots, \gamma_n(K)]^\top \in R^K$$

Fisher Vector

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \pi_k} = \sum_{n=1}^{N}\left[\frac{\gamma_n(k)}{\pi_k} - \frac{\gamma_n(1)}{\pi_1}\right]$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k^d} = \sum_{n=1}^{N}\gamma_n(k)\left[\frac{\boldsymbol{x}_n^d - \boldsymbol{\mu}_k^d}{(\boldsymbol{\sigma}_k^d)^2}\right]$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\sigma}_k^d} = \sum_{n=1}^{N}\gamma_n(k)\left[\frac{(\boldsymbol{x}_n^d - \boldsymbol{\mu}_k^d)^2}{(\boldsymbol{\sigma}_k^d)^3} - \frac{1}{\boldsymbol{\sigma}_k^d}\right]$$

フィッシャーベクトルの改善

- フィッシャーベクトルはBoF と比較して豊かな表現
  - しかしながら，そのまま画像識別に利用してもBoF とさほど性能に差がない.

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \pi_k} = \sum_{n=1}^{N} \left[\frac{\gamma_n(k)}{\pi_k} - \frac{\gamma_n(1)}{\pi_1} \right]$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k^d} = \sum_{n=1}^{N} \gamma_n(k) \left[\frac{\boldsymbol{x}_n^d - \boldsymbol{\mu}_k^d}{(\boldsymbol{\sigma}_k^d)^2} \right]$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\sigma}_k^d} = \sum_{n=1}^{N} \gamma_n(k) \left[\frac{(\boldsymbol{x}_n^d - \boldsymbol{\mu}_k^d)^2}{(\boldsymbol{\sigma}_k^d)^3} - \frac{1}{\boldsymbol{\sigma}_k^d} \right]$$

GMMのBoFとほぼ同じ

局所特徴x_nとGMMの各コンポーネントkの平均との差分

- 改善方法
  - L2正規化
  - パワー正規化
  - 空間ピラミッドの導入

- F. Perronnin, J. Sanchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. ECCV, 2010.

L2正規化によるフィッシャーベクトルの改善

- 対数尤度の勾配

$$G_\theta^{\mathcal{X}} = \frac{1}{N}\nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta}) \longrightarrow G_\theta^{\mathcal{X}} \approx \nabla_\theta \int_x p(\boldsymbol{x}) \log u_\theta(\boldsymbol{x}) d\boldsymbol{x}$$

1枚の画像から得られた局所特徴群Xはp(x)に従うとする

- 確率密度分布の分解

u(x)

q(x)

画像に特定の分布：前景

あらゆる画像を表現する分布：背景

$$p(\boldsymbol{x}) = \omega q(\boldsymbol{x}) + (1-\omega) u_\theta(\boldsymbol{x})$$

前景・背景の混合比

最尤法によりパラメータを求めた場合ゼロとなる！

$$G_\theta^{\mathcal{X}} \approx \omega \nabla_\theta \int_x q(\boldsymbol{x}) \log u_\theta(\boldsymbol{x}) d\boldsymbol{x} + (1-\omega) \nabla_\theta \int_x u_\theta(\boldsymbol{x}) \log u_\theta(\boldsymbol{x}) d\boldsymbol{x}$$

画像に特定の分布のみが残る！！！
ただし前景・背景の混合比の影響が残るのでL2正規化を行う.

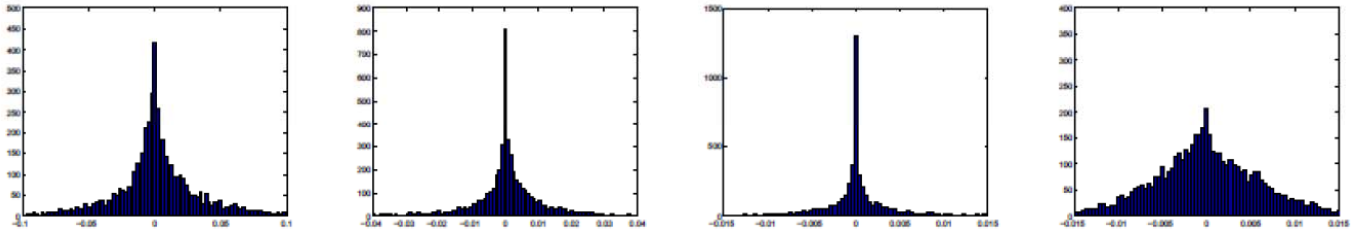
$$G_\theta^{\mathcal{X}} \approx \omega \nabla_\theta \int_x q(\boldsymbol{x}) \log u_\theta(\boldsymbol{x}) d\boldsymbol{x}$$

パワー正規化，空間ピラミッドによるフィッシャーベクトルの改善

http://www.image-net.org/challenges/LSVRC/2010/ILSVRC2010_XRCE.pdf

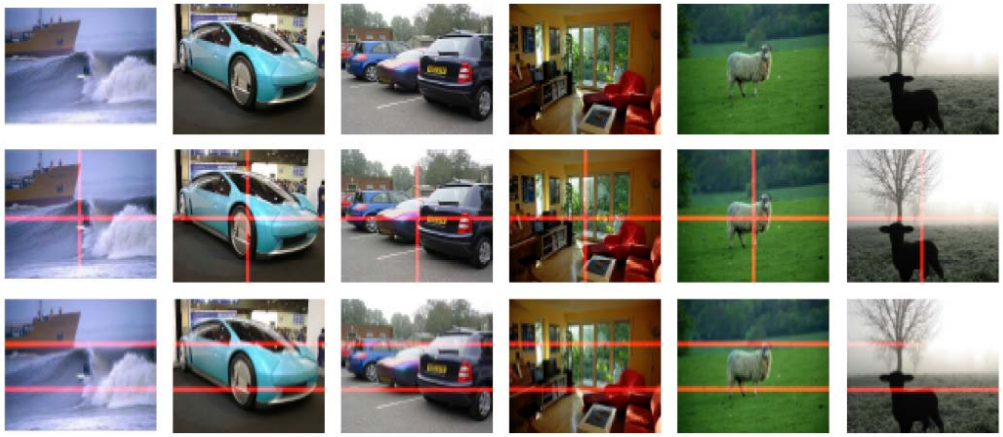
- パワー正規化
  - 混合数の増加に伴いフィッシャーベクトルがスパースになる
  - スパースベクトルにおけるL2距離は性能が悪い
  - 方針１：カーネル法は計算コストが高い
  - 方針２：スパースにしない

$$f(z) = \mathrm{sign}(z)|z|^{\alpha}$$



K = 16　K = 64　K = 256　K = 256 Power Normalized

- 空間ピラミッド



画像1枚あたり8個のフィッシャーベクトルを抽出

フィッシャーベクトルの性能

http://www.image-net.org/challenges/LSVRC/2010/ILSVRC2010_XRCE.pdf

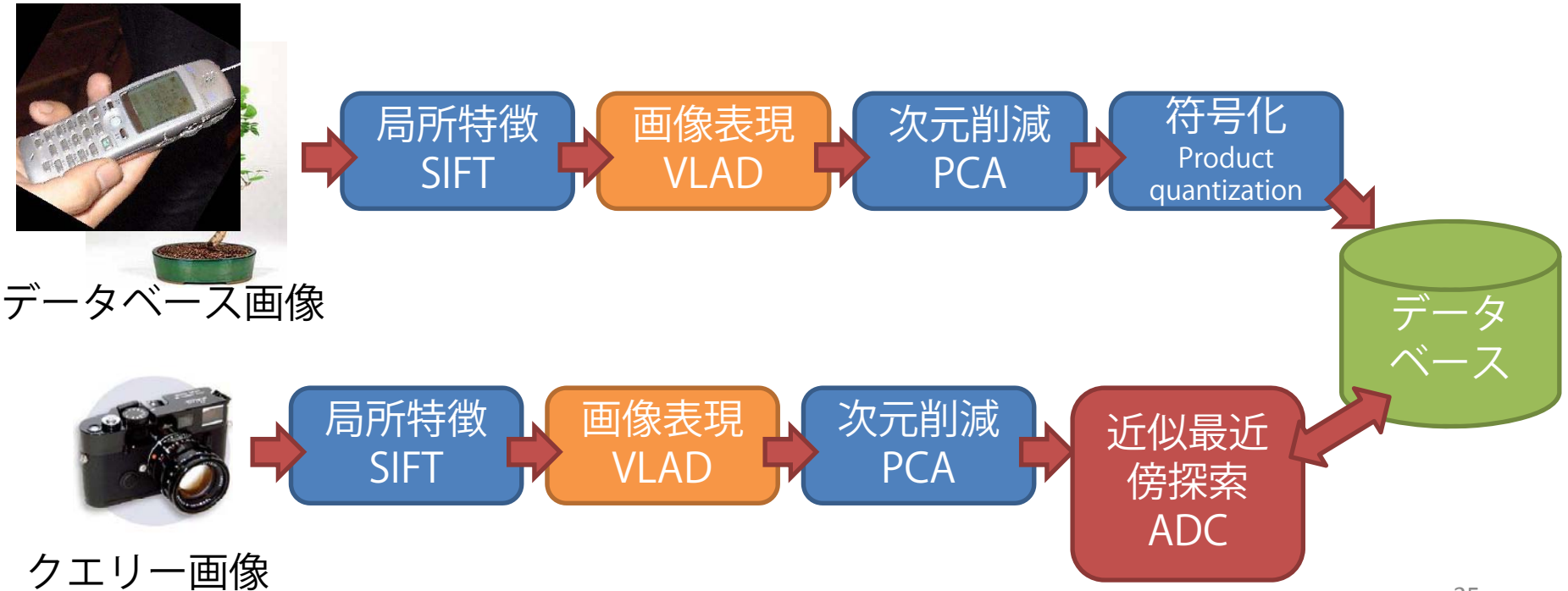
- Pascal VOC 2007
- 改良されたフィッシャーベクトルを利用
- 識別機：線形SVM

PN	L2	SP	SIFT	Col	S+C
-	-	-	47.9	34.2	45.9
✓	-	-	54.2	45.9	57.6
-	✓	-	51.8	40.6	53.9
-	-	✓	50.3	37.5	49.0
✓	✓	✓	58.3	50.9	60.3

パワー正規化＞L2正規化＞空間ピラミッド，の順で改善の効果が高い

フィッシャーベクトルの画像検索への応用例

- H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. CVPR, 2010.
- 20bitに画像表現しても，生のBoFを使った検索と同じ検索性能
- パイプライン



VLAD

H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. CVPR, 2010.

- Vector of Locally Aggregated Descriptors

VLADのd番目要素

局所特徴のd番目要素

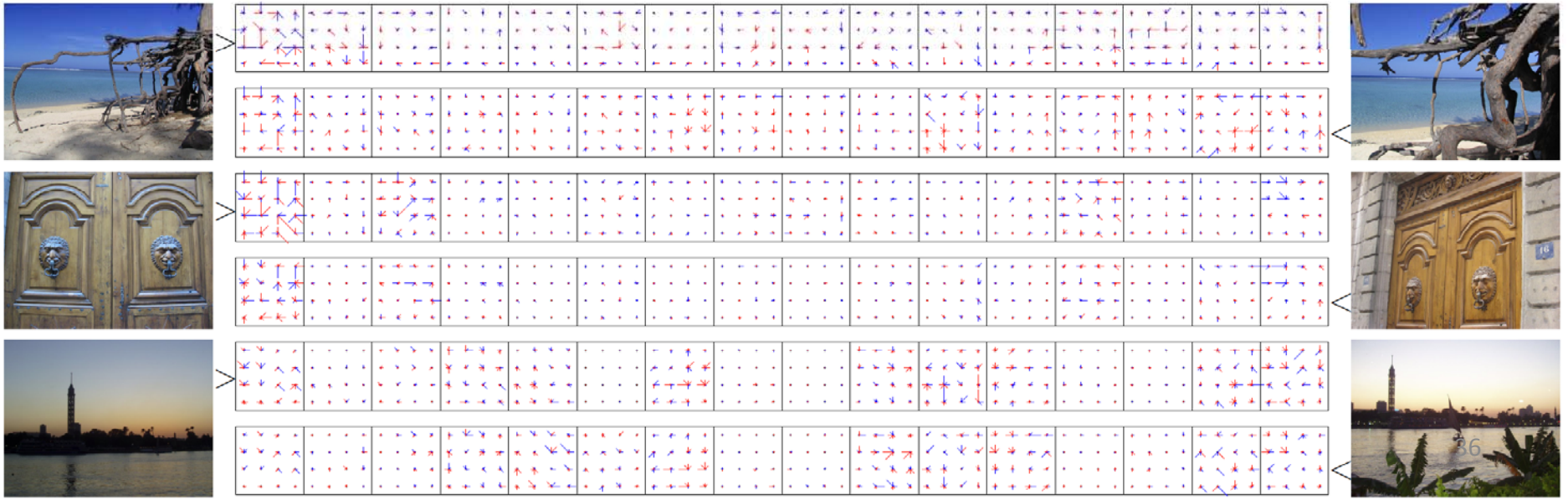
局所特徴が割り当てられたコードワードiのベクトル

$$\boldsymbol{z}_i^d = \sum_{x \in \mathcal{X}_i} \left(\boldsymbol{x}^d - \boldsymbol{v}_i^d \right)$$

この後 L2正規化

コードワードiに属する局所特徴集合

VLADの例，コードワード数：16



VLADとフィッシャーベクトル

- フィッシャーベクトル

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \pi_k} = \sum_{n=1}^{N} \left[\frac{\gamma_n(k)}{\pi_k} - \frac{\gamma_n(1)}{\pi_1} \right]$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k^d} = \sum_{n=1}^{N} \gamma_n(k) \left[\frac{\boldsymbol{x}_n^d - \boldsymbol{\mu}_k^d}{(\boldsymbol{\sigma}_k^d)^2} \right]$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\sigma}_k^d} = \sum_{n=1}^{N} \gamma_n(k) \left[\frac{(\boldsymbol{x}_n^d - \boldsymbol{\mu}_k^d)^2}{(\boldsymbol{\sigma}_k^d)^3} - \frac{1}{\boldsymbol{\sigma}_k^d} \right]$$

GMMのBoFとほぼ同じ

局所特徴x_nとGMMの各コンポーネントkの平均との差分

- VLAD

$$\boldsymbol{z}_i^d = \sum_{x \in \mathcal{X}_i} \left(\boldsymbol{x}^d - \boldsymbol{v}_i^d \right)$$

VLADのd番目要素

局所特徴のd番目要素

局所特徴が割り当てられたコードワードiのベクトル

コードワードiに属する局所特徴集合

- 負担率：ハードな割り当て
- 分散：全てのコンポーネントで同じ

- →VLADはフィッシャーベクトルの平均に関する要素と同じ

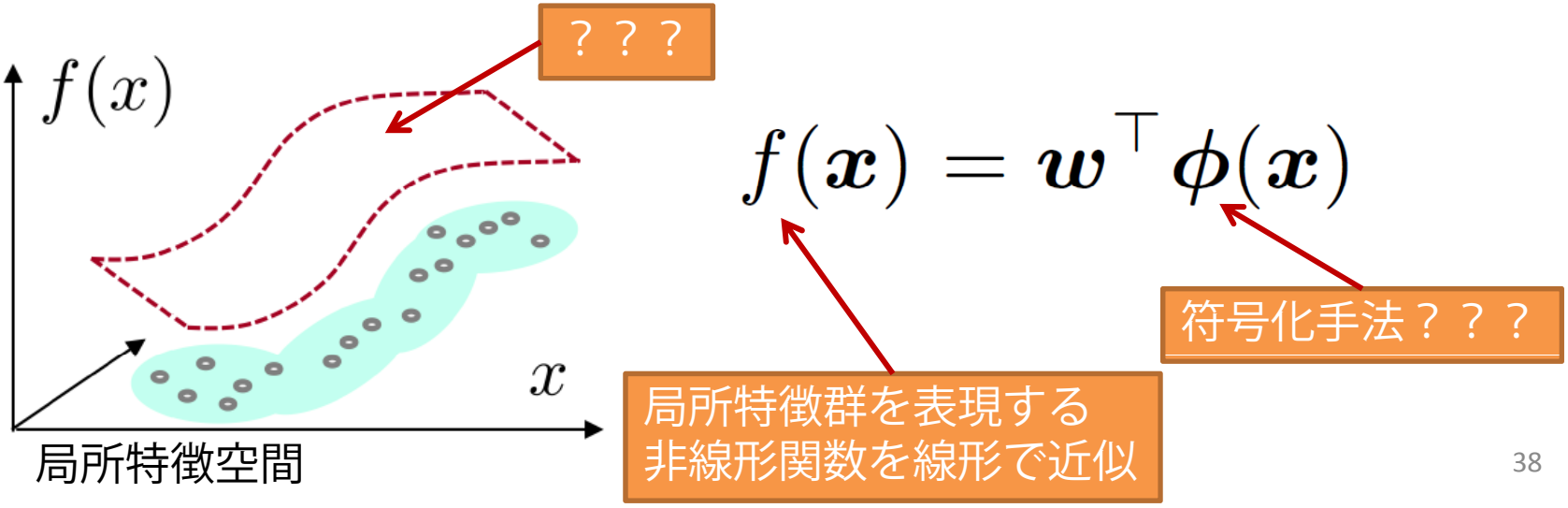
スーパーベクトル符号化
Super-Vector Coding

- X. Zhou, K. Yu, T. Zhang, and T.S. Huang. Image classification using super-vector coding of local image descriptors. ECCV, 2010.

- BoF や混合ガウス分布を用いたBoF の改善手法
  - 特徴空間における局所特徴の分布の表現を得るプロセスと解釈できた.

- ここでも高次元空間における局所特徴分布を表現する，なめらかな非線形関数 *f(x)* の学習について考える.

- 非線形関数 *f(x)* を線形表現可能な符号化手法 *Φ(x)* を求める.



スーパーベクトル符号化の導出

- 局所特徴をコードブックを利用して近似

$$\boldsymbol{x} \approx \sum_{k=1}^{K} \gamma_x(k)\boldsymbol{v}_k \qquad \boldsymbol{\gamma}_x = [\gamma_x(1), \cdots, \gamma_x(K)], \sum_{k=1}^{K} \gamma_x(k) = 1$$

負担率のようなもの

コードワードk

- β Lipschitz derivative smooth

関数f(x)の1次近似のUpper boundに関する式

$$|f(\boldsymbol{x}) - f(\boldsymbol{x}') - \nabla f(\boldsymbol{x}')^\top (\boldsymbol{x} - \boldsymbol{x}')| \leqq \frac{\beta}{2}||\boldsymbol{x} - \boldsymbol{x}'||^2$$

コードワードの代入 $\boldsymbol{x}' = \boldsymbol{v}^x$

$$|f(\boldsymbol{x}) - f(\boldsymbol{v}^x) - \nabla f(\boldsymbol{v}^x)^\top (\boldsymbol{x} - \boldsymbol{v}^x)| \leqq \frac{\beta}{2}||\boldsymbol{x} - \boldsymbol{v}^x||^2$$

$$f(\boldsymbol{x}) = f(\boldsymbol{v}^x) + \nabla f(\boldsymbol{v}^x)^\top (\boldsymbol{x} - \boldsymbol{v}^x) \cdots (\star)$$

||x-v||が小さければ近似精度が向上

- スーパーベクトル符号化

$$f(\boldsymbol{x}) \approx \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x})$$

式(☆)を分解！

Super Vector Coding

$$\boldsymbol{\phi}(\boldsymbol{x}) = \left[s\gamma_x(k), \gamma_x(k)(\boldsymbol{x} - \boldsymbol{v}_k)^\top \right]^\top_{v_k \in \mathcal{V}}$$

$$\boldsymbol{w} = \left[\frac{1}{s} f(\boldsymbol{v}_k), (\nabla f(\boldsymbol{v}_k))^\top \right]^\top_{v_k \in \mathcal{V}}$$

スーパーベクトル符号化の解釈

- スーパーベクトル符号化の例
  - コードワード数：3，γ =[0 1 0]'

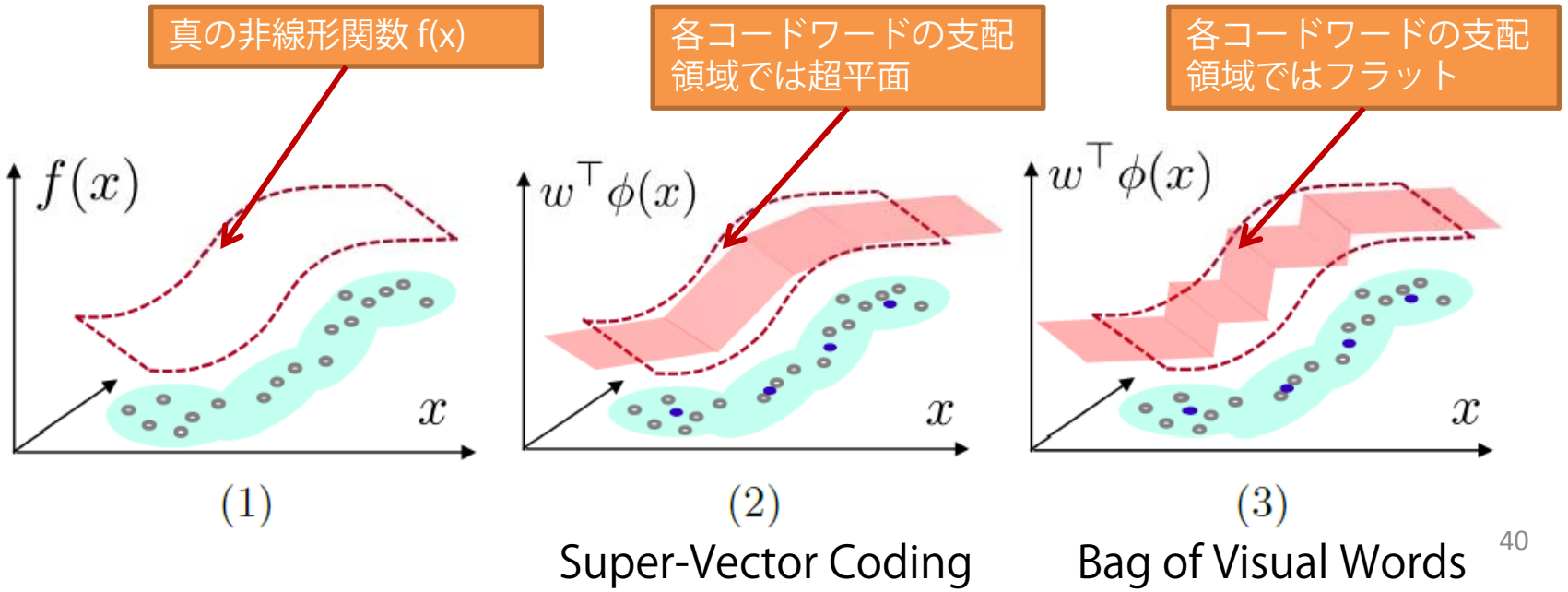
X. Zhou, K. Yu, T. Zhang, and T.S. Huang. Image classification using super-vector coding of local image descriptors. ECCV, 2010.

Super Vector Coding

$$\boldsymbol{\phi}(\boldsymbol{x}) = \left[s\gamma_x(k), \gamma_x(k)(\boldsymbol{x} - \boldsymbol{v}_k)^\top \right]^\top_{v_k \in \mathcal{V}}$$

$$\Rightarrow \quad \phi(x) = \left[\underbrace{0, \ldots, 0}_{d+1 \text{ dim.}}, \underbrace{s, (x-v)^\top}_{d+1 \text{ dim.}}, \underbrace{0, \ldots, 0}_{d+1 \text{ dim.}} \right]^\top$$

- スーパーベクトル符号化とBoF



スーパーベクトル符号化とフィッシャーベクトル

- フィッシャーベクトル

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \pi_k} = \sum_{n=1}^{N} \left[\frac{\gamma_n(k)}{\pi_k} - \frac{\gamma_n(1)}{\pi_1} \right]$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k^d} = \sum_{n=1}^{N} \gamma_n(k) \left[\frac{\boldsymbol{x}_n^d - \boldsymbol{\mu}_k^d}{(\boldsymbol{\sigma}_k^d)^2} \right]$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\sigma}_k^d} = \sum_{n=1}^{N} \gamma_n(k) \left[\frac{(\boldsymbol{x}_n^d - \boldsymbol{\mu}_k^d)^2}{(\boldsymbol{\sigma}_k^d)^3} - \frac{1}{\boldsymbol{\sigma}_k^d} \right]$$

GMMのBoFとほぼ同じ

局所特徴x_nとGMMの各コンポーネントkの平均との差分

- スーパーベクトル符号化

$$\boldsymbol{\phi}(\boldsymbol{x}) = \left[s\gamma_x(k), \gamma_x(k)(\boldsymbol{x} - \boldsymbol{v}_k)^\top \right]^\top_{v_k \in \mathcal{V}}$$

負担率

局所特徴x_nとコードワードとの差分

- 混合比：一定
- 分散：一定

→スーパーベクトル符号化はフィッシャーベクトルの混合比と平均に関する要素と同じ

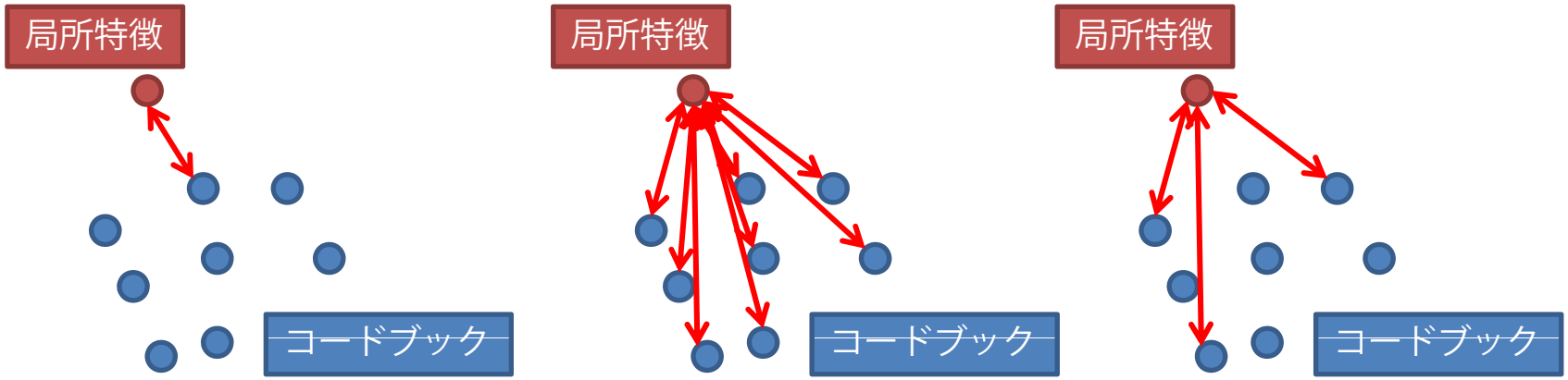
スパース符号化（Sparse Coding）

- J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. CVPR, 2009.

- BoF
  - 局所特徴が一つのコードワードに割り当てられる

- BoFのGMMによる表現
  - 局所特徴が全てのコードワードと関係を持つ

- スパース符号化
  - 局所特徴が少数のコードワードと関係を持つ



(a) BoF　(b) GMM　(c) Sparse Coding

スパース符号化の定式化

- Bag of Visual Words
  - ベクトル量子化（VQ）

$$\min_{U,V} \sum_{n=1}^{N} ||\boldsymbol{x}_n - \boldsymbol{V}\boldsymbol{u}_n||^2$$

$$\text{s.t. } \text{Card}(\boldsymbol{u}_n) = 1, |\boldsymbol{u}_n| = 1, \boldsymbol{u}_n \succeq 0, \forall n$$

局所特徴

コードブック

局所特徴がどのコードワードに所属するかを示す指標

一つのコードワードに属する制約→厳しすぎる！！！

- スパース符号化（Sparse Coding）

$$\min_{U,V} \sum_{n=1}^{N} ||\boldsymbol{x}_n - \boldsymbol{V}\boldsymbol{u}_n||^2 + \lambda|\boldsymbol{u}_n|$$

$$\text{s.t. } ||\boldsymbol{v}_k|| \leqq 1, \forall k$$

L1ノルム正則化項
→少数のコードワードへの所属を許容

L1正則化の役割

- コードブックは局所特徴の次元数よりも多く，過剰（K > D）なため，under determinedな系である．つまり情報が不足して解を定められない状況にある．そのためL1 正則化により解を定めることが可能となる．

- *スパース性の事前知識を用いることによって*局所特徴の顕著なパターンを捉えることができる．

- *ベクトル量子化よりもスパース符号化の方が*量子化誤差を低減させられる．

スパース符号化空間ピラミッド

- 空間ピラミッド
  - 符号化された局所特徴群 $\boldsymbol{U}$ から一つの特徴ベクトル $\boldsymbol{f}$ を得る手段

- プーリング（pooling）

$$\boldsymbol{f} = \mathcal{F}(\boldsymbol{U})$$

局所特徴集合

プーリング関数

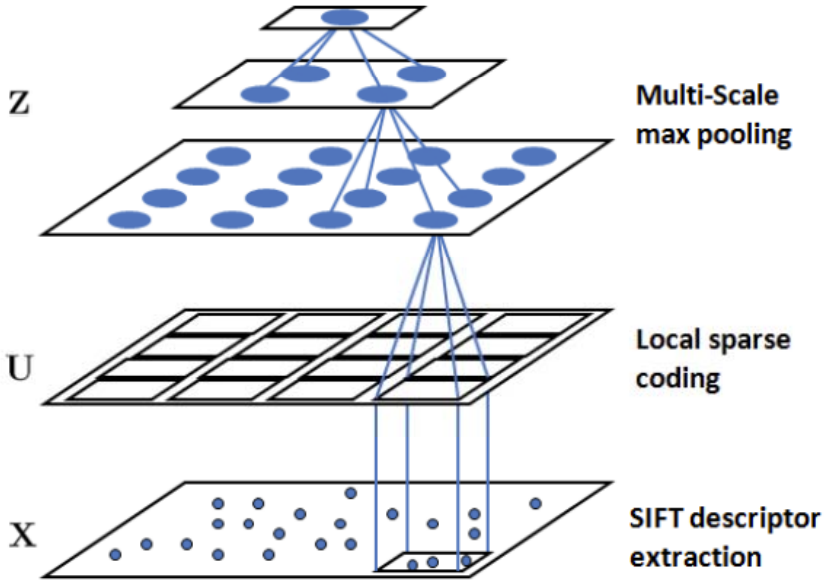
  - 平均プーリング
    average pooling

$$\boldsymbol{f} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{u}_n$$

BoFはこれを利用

  - 最大値プーリング
    max pooling

$$\boldsymbol{f}^d = \max\{|\boldsymbol{u}_1^d|, |\boldsymbol{u}_2^d|, \cdots, |\boldsymbol{u}_N^d|\}$$



J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. CVPR, 2009.

最大値プーリングの効果

- Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. CVPR, 2010.

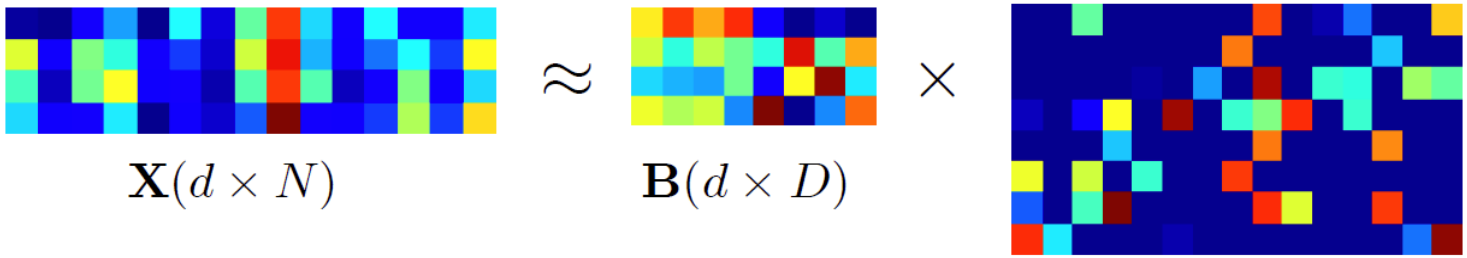
Method	Caltech-101, 30 training examples		15 Scenes, 100 training examples	
	Average Pool	Max Pool	Average Pool	Max Pool
	Results with basic features, SIFT extracted each 8 pixels			
Hard quantization, linear kernel	51.4 ± 0.9 [256]	64.3 ± 0.9 [256]	73.9 ± 0.9 [1024]	80.1 ± 0.6 [1024]
Hard quantization, intersection kernel	64.2 ± 1.0 [256] **(1)**	64.3 ± 0.9 [256]	80.8 ± 0.4 [256] **(1)**	80.1 ± 0.6 [1024]
Soft quantization, linear kernel	57.9 ± 1.5 [1024]	69.0 ± 0.8 [256]	75.6 ± 0.5 [1024]	81.4 ± 0.6 [1024]
Soft quantization, intersection kernel	66.1 ± 1.2 [512] **(2)**	70.6 ± 1.0 [1024]	81.2 ± 0.4 [1024] **(2)**	83.0 ± 0.7 [1024]
Sparse codes, linear kernel	61.3 ± 1.3 [1024]	$\mathbf{71.5 \pm 1.1}$ [1024] **(3)**	76.9 ± 0.6 [1024]	83.1 ± 0.6 [1024] **(3)**
Sparse codes, intersection kernel	70.3 ± 1.3 [1024]	$\mathbf{71.8 \pm 1.0}$ [1024] **(4)**	83.2 ± 0.4 [1024]	$\mathbf{84.1 \pm 0.5}$ [1024] **(4)**
	Results with macrofeatures and denser SIFT sampling			
Hard quantization, linear kernel	55.6 ± 1.6 [256]	70.9 ± 1.0 [1024]	74.0 ± 0.5 [1024]	80.1 ± 0.5 [1024]
Hard quantization, intersection kernel	68.8 ± 1.4 [512]	70.9 ± 1.0 [1024]	81.0 ± 0.5 [1024]	80.1 ± 0.5 [1024]
Soft quantization, linear kernel	61.6 ± 1.6 [1024]	71.5 ± 1.0 [1024]	76.4 ± 0.7 [1024]	81.5 ± 0.4 [1024]
Soft quantization, intersection kernel	70.1 ± 1.3 [1024]	73.2 ± 1.0 [1024]	81.8 ± 0.4 [1024]	83.0 ± 0.4 [1024]
Sparse codes, linear kernel	65.7 ± 1.4 [1024]	$\mathbf{75.1 \pm 0.9}$ [1024]	78.2 ± 0.7 [1024]	83.6 ± 0.4 [1024]
Sparse codes, intersection kernel	73.7 ± 1.3 [1024]	$\mathbf{75.7 \pm 1.1}$ [1024]	83.5 ± 0.4 [1024]	$\mathbf{84.3 \pm 0.5}$ [1024]

Table 1. Average recognition rate on Caltech-101 and 15-Scenes benchmarks, for various combinations of coding, pooling, and classifier types. The codebook size shown inside brackets is the one that gives the best results among 256, 512 and 1024. Linear and histogram intersection kernels are identical when using hard quantization with max pooling (since taking the minimum or the product is the same for binary vectors), but results have been included for both to preserve the symmetry of the table. Top: Results with the baseline SIFT sampling density of 8 pixels and standard features. Bottom: Results with the set of parameters for SIFT sampling density and macrofeatures giving the best performance for sparse coding.

局所座標符号化
Local Coordinate Coding (LCC)

- K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. NIPS, 2009.

http://www.image-net.org/challenges/LSVRC/2010/ILSVRC2010_NEC-UIUC.pdf



$\mathbf{X}(d \times N)$ $\approx$ $\mathbf{B}(d \times D)$ $\times$ $\mathbf{Z}(D \times N)$

Assume $\mathbf{B}$ is given.

Sparse coding:

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{x} - \mathbf{B}\mathbf{z}\|^2 + \lambda \sum_{i=1}^{D} |z_i|$$

局所性がスパースネスよりも本質！！

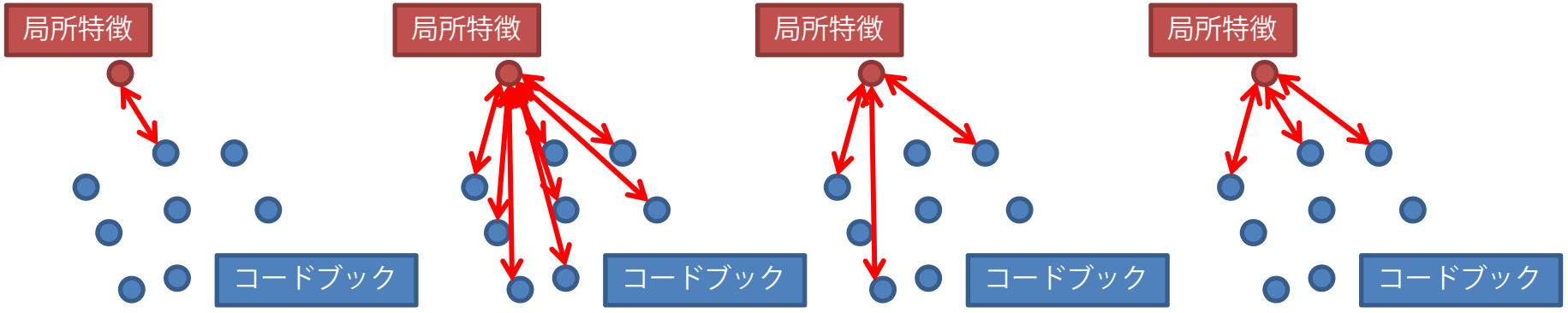
LCC: K. Yu et. al, NIPS 2009

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{x} - \mathbf{B}\mathbf{z}\|^2 + \lambda \sum_{i=1}^{D} \|\mathbf{x} - \mathbf{b}_i\|^2 |z_i|$$

Explicitly enforcing locality constraint

局所線形制約符号化と他符号化の比較

- BoF
  - 局所特徴が一つのコードワードに割り当てられる
- BoFのGMMによる表現
  - 局所特徴が全てのコードワードと関係を持つ
- スパース符号化
  - 局所特徴が少数のコードワードと関係を持つ
- 局所線形制約符号化
  - 局所特徴が局所の少数コードワードと関係を持つ



(a) BoF (b) GMM (c) Sparse Coding (d) LCC

なぜ局所座標符号化が良いのか？

http://www.image-net.org/challenges/LSVRC/2010/ILSVRC2010_NEC-UIUC.pdf

$$f(\mathbf{x}) \approx \sum_{i=1}^{D} z_i(\mathbf{x}) w_i$$

e.g. nonlinear separating hyperplane

$$|f(\mathbf{x}) - \sum_{i=1}^{D} z_i(\mathbf{x}) f(\mathbf{b}_i)|$$ ← Functional approximation error

$$\leq \alpha \|\mathbf{x} - \mathbf{B}\mathbf{z}(\mathbf{x})\| + \beta \sum_{i=1}^{D} \|\mathbf{x} - \mathbf{b}_i\|^2 |z_i(\mathbf{x})|$$

Coding error

Locality term

- よりよく近似するためには
  - 局所特徴に対して局所性を有すること
  - 局所特徴の再構築誤差を減らすこと

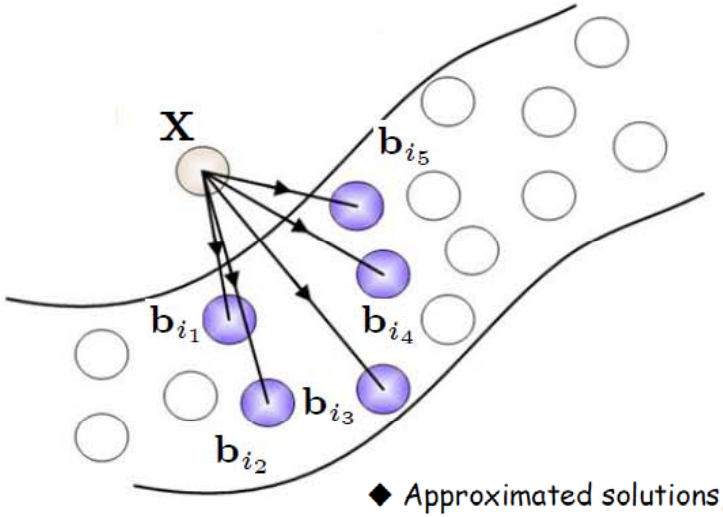
局所座標符号化の高速な実装

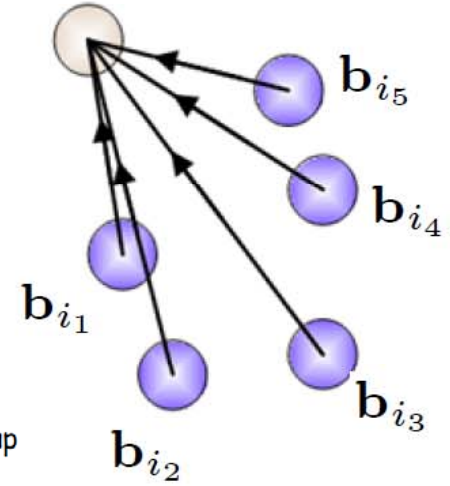
- 局所制約線形符号化
  - Locality-constrained Linear Coding (LLC)
  - J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. CVPR, 2010.

Step 1: be local to the test point $\mathbf{x}$

-- given $\mathbf{x}$, find its KNNs.

Step 2: small reconstruction error -- solve LMS fitting using only the KNNs





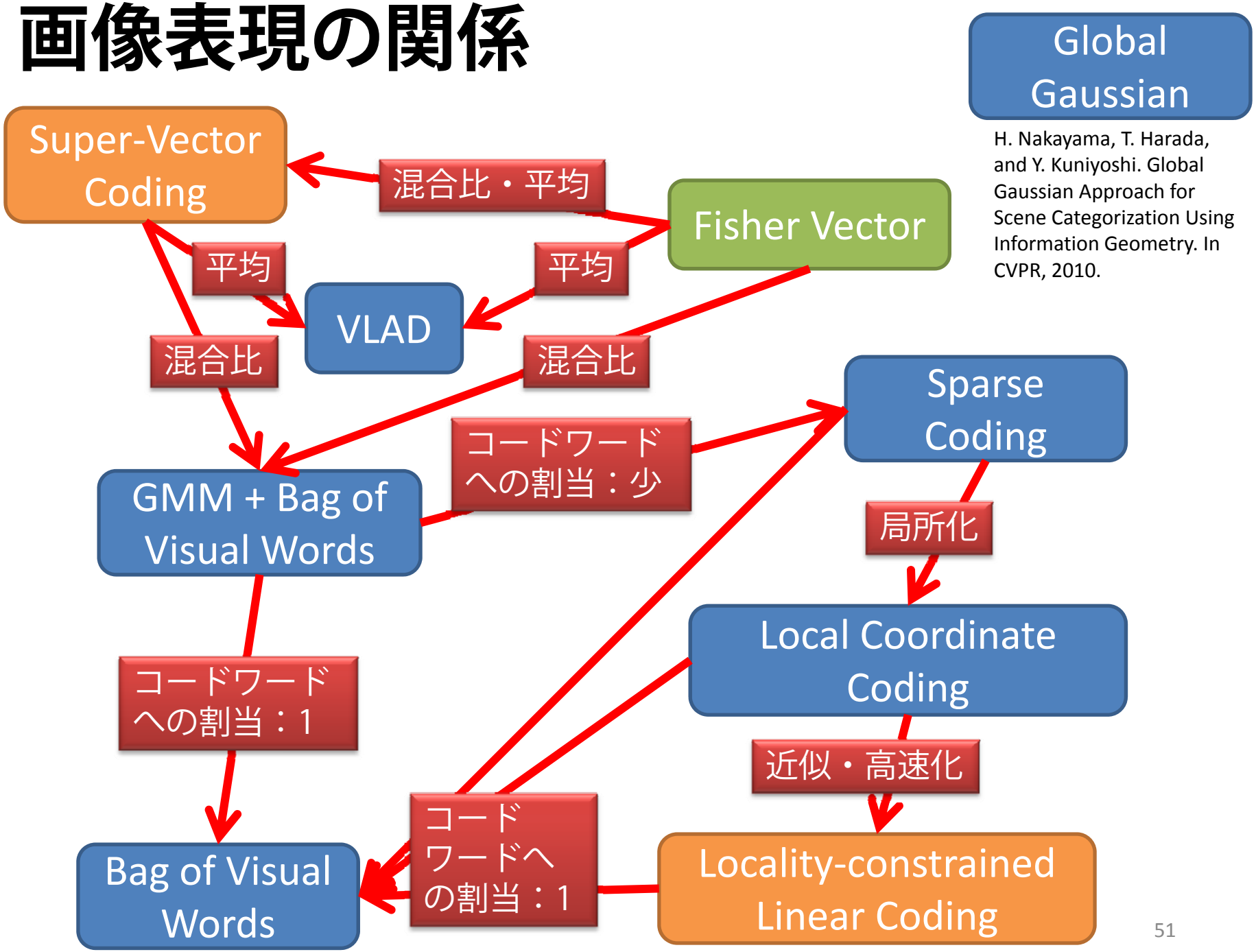
◆ Approximated solutions, but significant speedup

For a regular image (7k patches), with D=8192:
sparse coding needs ~10mins, (approximate) LCC needs only ~2s

http://www.image-net.org/challenges/LSVRC/2010/ILSVRC2010_NEC-UIUC.pdf

局所線形埋込み（Local Linear Embedding, LLE）と比較して，
局所制約線形符号化はコードブックの学習が入る点で異なる．

画像表現の関係


Global Gaussian
H. Nakayama, T. Harada, and Y. Kuniyoshi. Global Gaussian Approach for Scene Categorization Using Information Geometry. In CVPR, 2010.
Super-Vector Coding
混合比・平均
Fisher Vector
平均
平均
VLAD
混合比
混合比
Sparse Coding
コードワードへの割当：少
GMM + Bag of Visual Words
局所化
Local Coordinate Coding
コードワードへの割当：1
近似・高速化
コードワードへの割当：1
Bag of Visual Words
Locality-constrained Linear Coding

まとめ

- 大規模画像データセットを用いた画像認識のトレンドについて紹介した.

- 近年，大規模画像識別に用いられている画像表現を紹介し，それらの体系化の試みを解説した.

謝辞

- 東京大学大学院情報理工学系研究科
  - 國吉康夫　教授
  - 博士3年　中山君
  - 修士2年　牛久君
  - 修士1年　山下君
  - 学部4年　井村君

- JSTさきがけ
- 科研「情報爆発IT基盤」