# ImageNet Classification

Sunpreet Arora & Josh Klontz

April 22, 2013

# Challenges

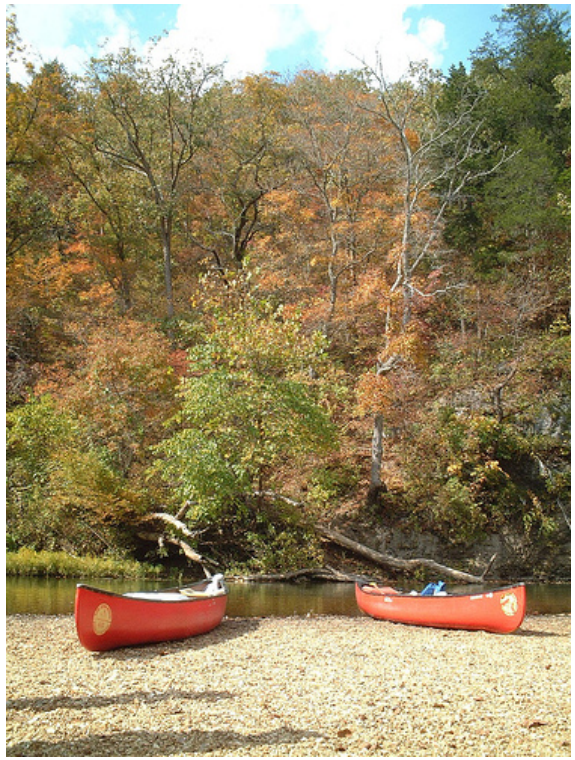
- Intraclass variation

# Challenges

- Intraclass variation
- Interclass similarity

# Challenges

- Intraclass variation
- Interclass similarity
- Scale



# Challenges

- Intraclass variation
- Interclass similarity
- Scale
- Pose

# Challenges

- Intraclass variation
- Interclass similarity
- Scale
- Pose
- Illumination

# Challenges

- Intraclass variation
- Interclass similarity
- Scale
- Pose
- Illumination
- Occlusion


# Challenges

- Intraclass variation
- Interclass similarity
- Scale
- Pose
- Illumination
- Occlusion
- Clutter

# Methodology

Image

Multi-scale SIFT

Color

Multi-scale LBP

SIFT codebook

Color codebook

LBP codebook

Non-linear SVM

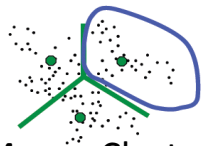
Non-linear SVM

Non-linear SVM

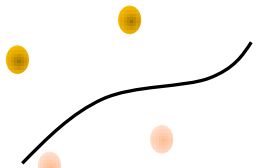
Class1
Class2
...

Class labels

Visual Vocabulary



K-Means Clustering

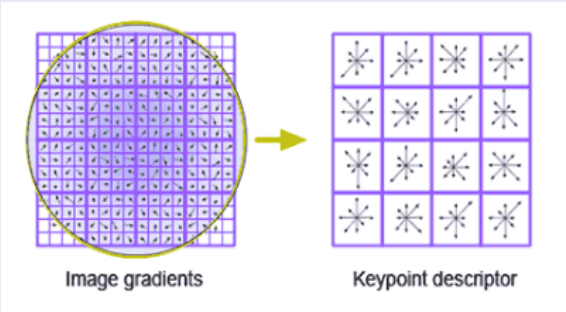


Classification

# Feature Descriptors

## SIFT

128 dimensions

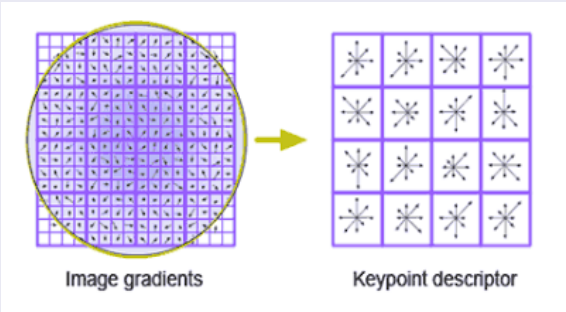



16 orientation histograms, 8 bins each.

# Feature Descriptors

## SIFT

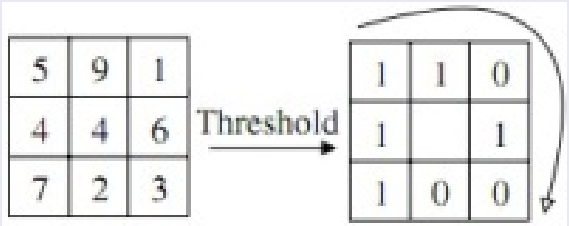
128 dimensions

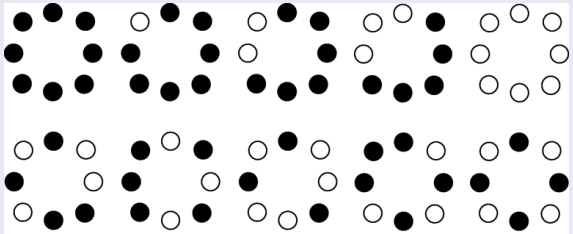



16 orientation histograms, 8 bins each.

## LBP

59 dimensions



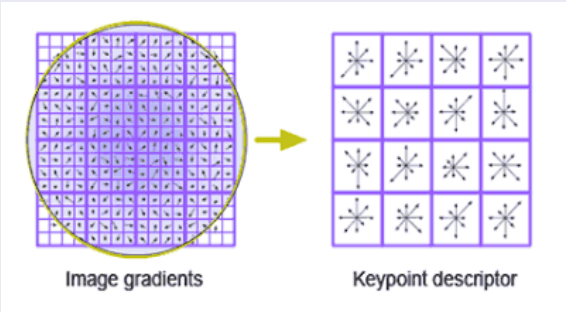



Compute histogram of the 59 “U2” patterns.

# Feature Descriptors

## SIFT

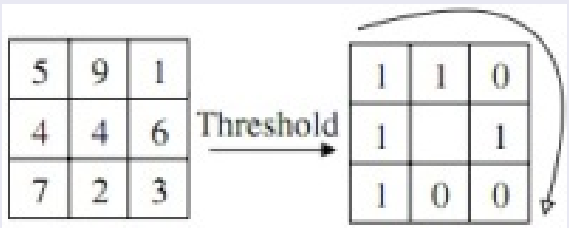
128 dimensions

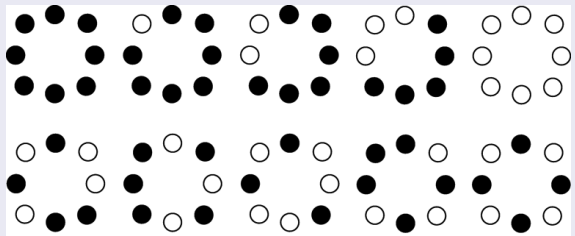



16 orientation histograms, 8 bins each.

## LBP

59 dimensions



Compute histogram of the 59 "U2" patterns.

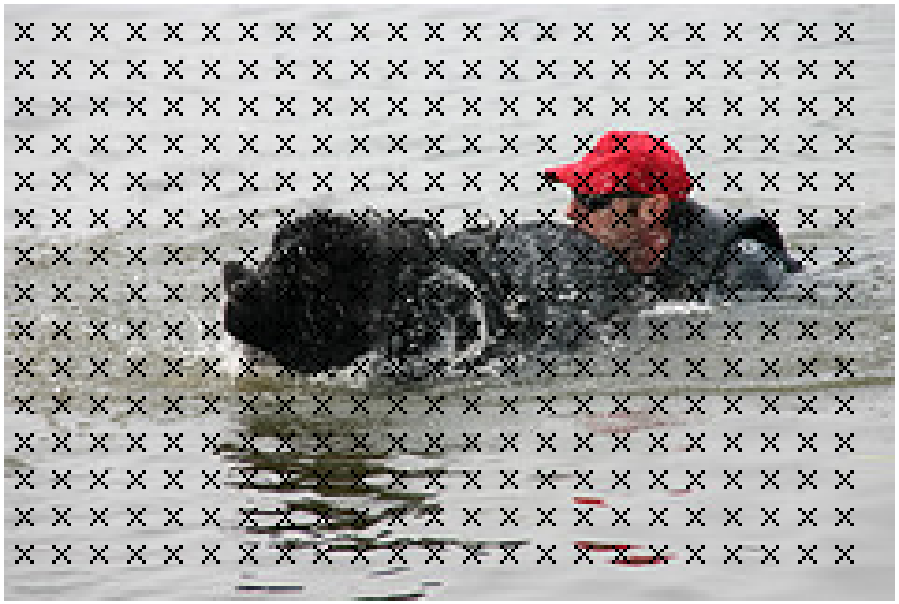
## RG

64 dimensions

$$R = \frac{r}{r+g+b} \qquad G = \frac{g}{r+g+b} \tag{1}$$

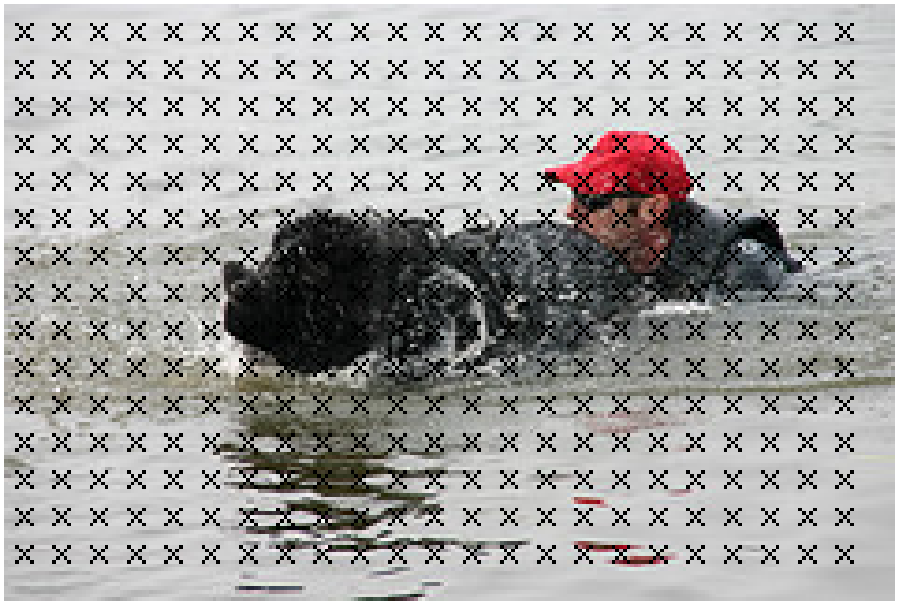
$R$ and $G$ each quantized into 32-bin histogram.

# Multi-Scale Dense Feature Extraction



Dense sampling at 3 scales

# Multi-Scale Dense Feature Extraction

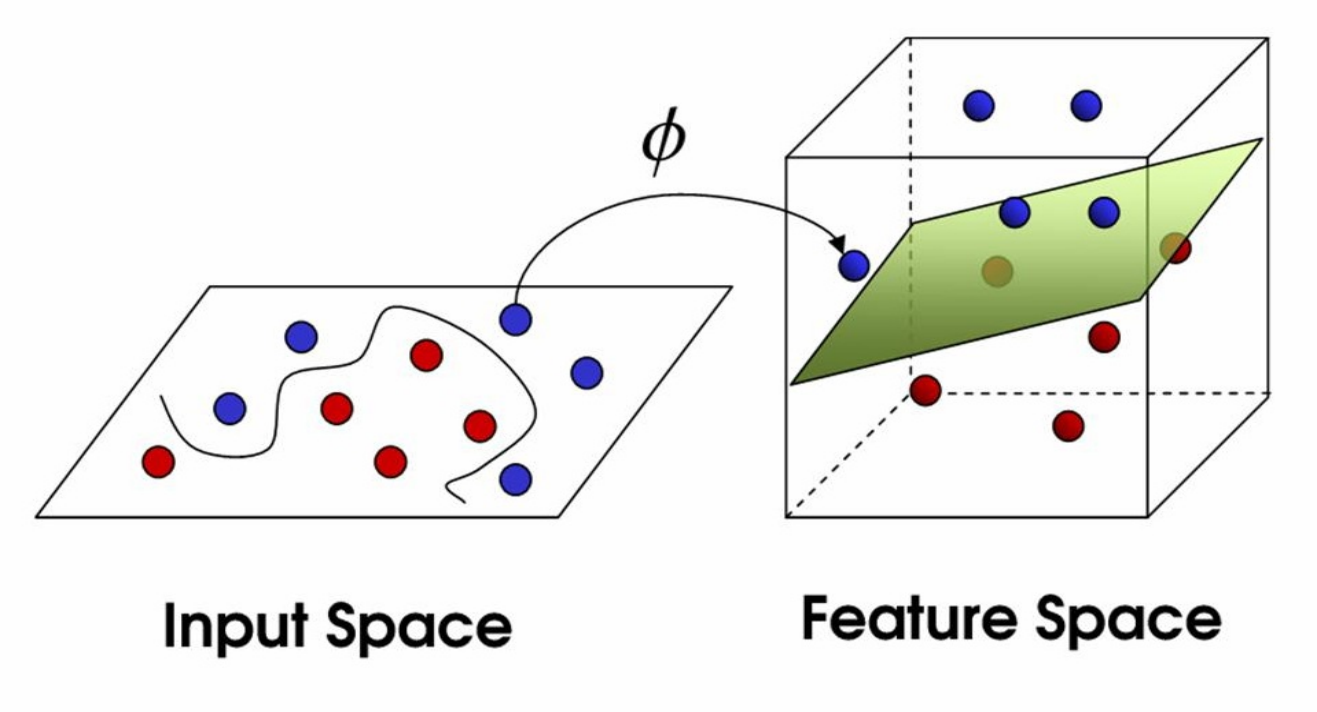


Dense sampling at 3 scales

Bag of words

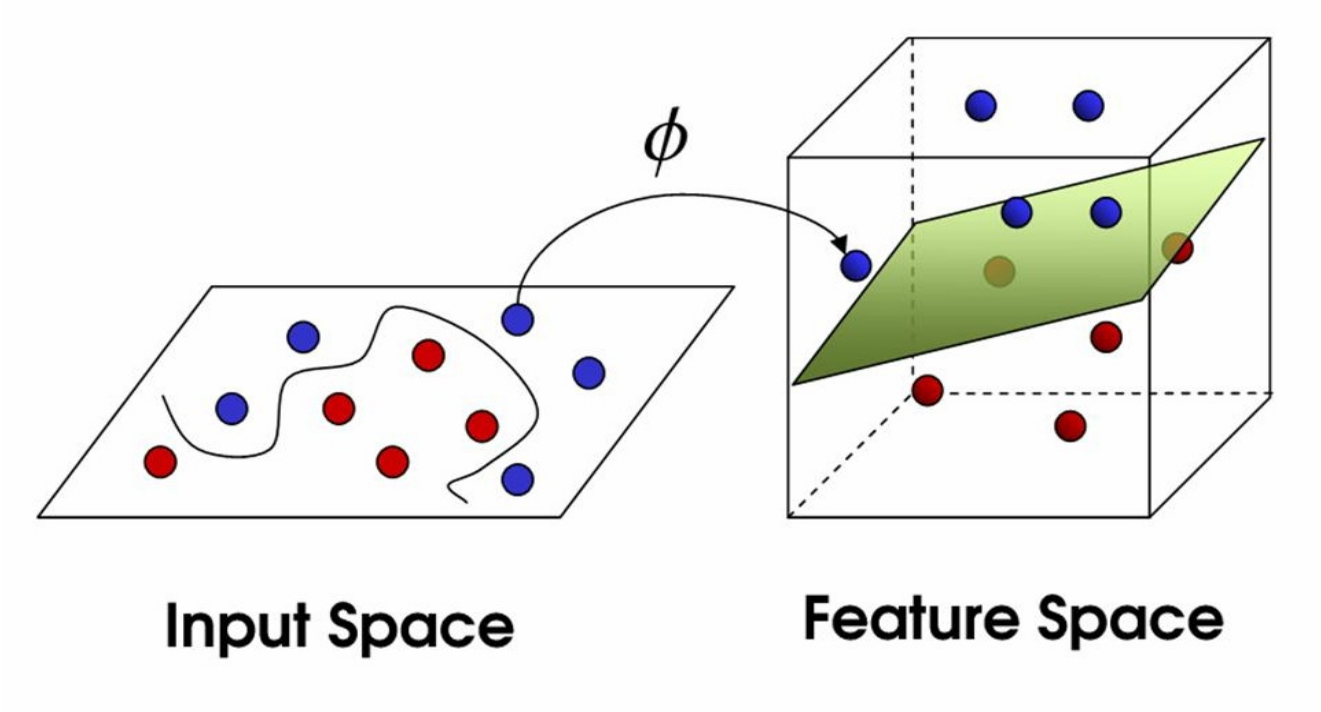
# Classifiers

- Train three different SVM classifiers for each of the codebooks.



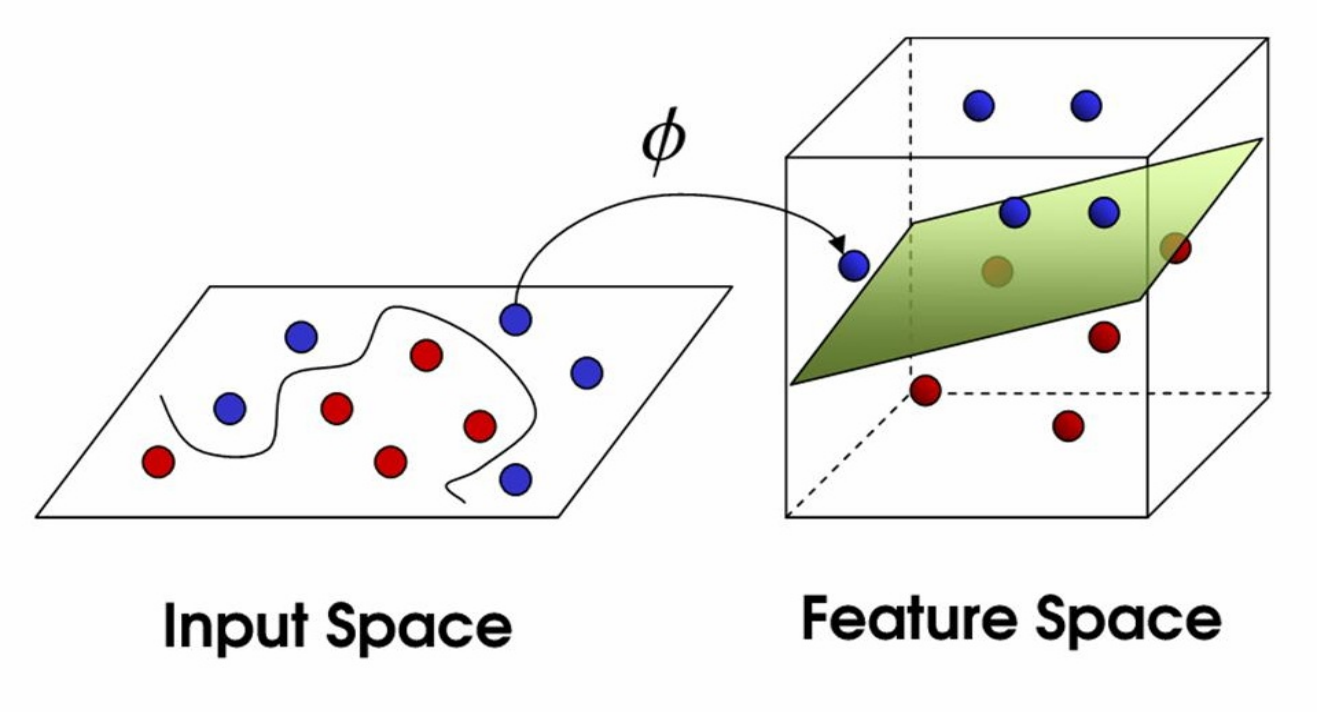
# Classifiers

- Train three different SVM classifiers for each of the codebooks.
- Linear kernel does not work very well because the classes are not well separable in the feature space.



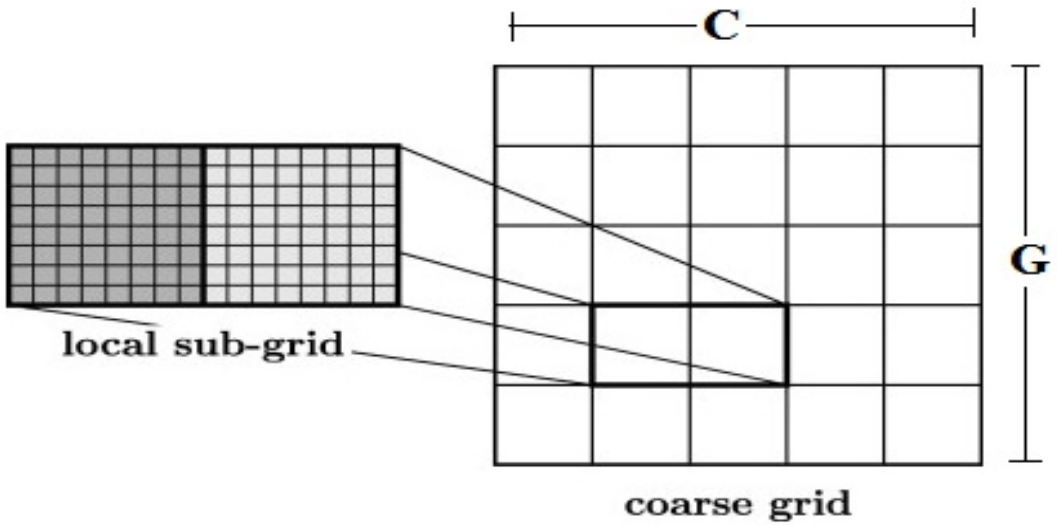
# Classifiers

- Train three different SVM classifiers for each of the codebooks.
- Linear kernel does not work very well because the classes are not well separable in the feature space.
- Need to map the data to a higher dimensional space: **RBF kernel**.



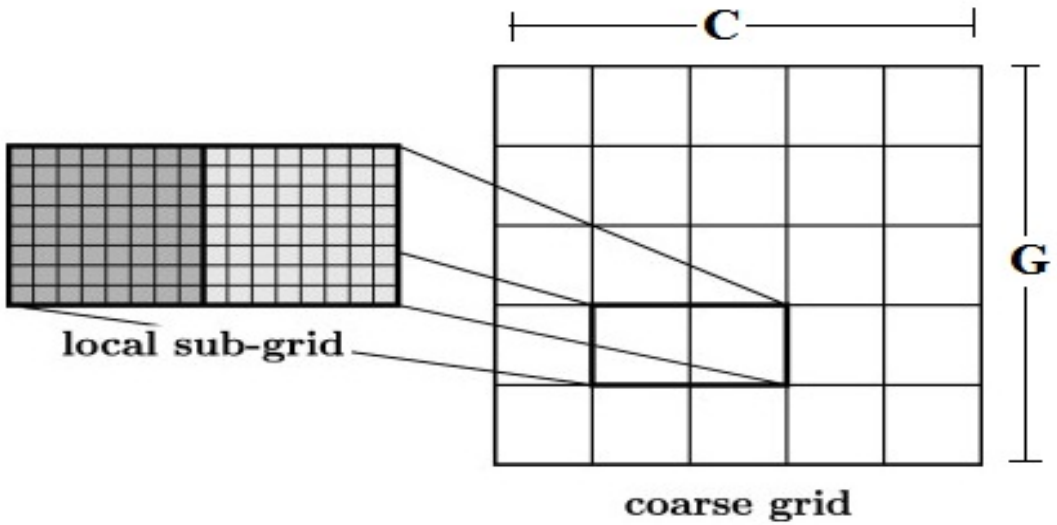
# Parameter Selection

- Effectiveness of SVM classifiers depends on the selection of the right set of parameters.



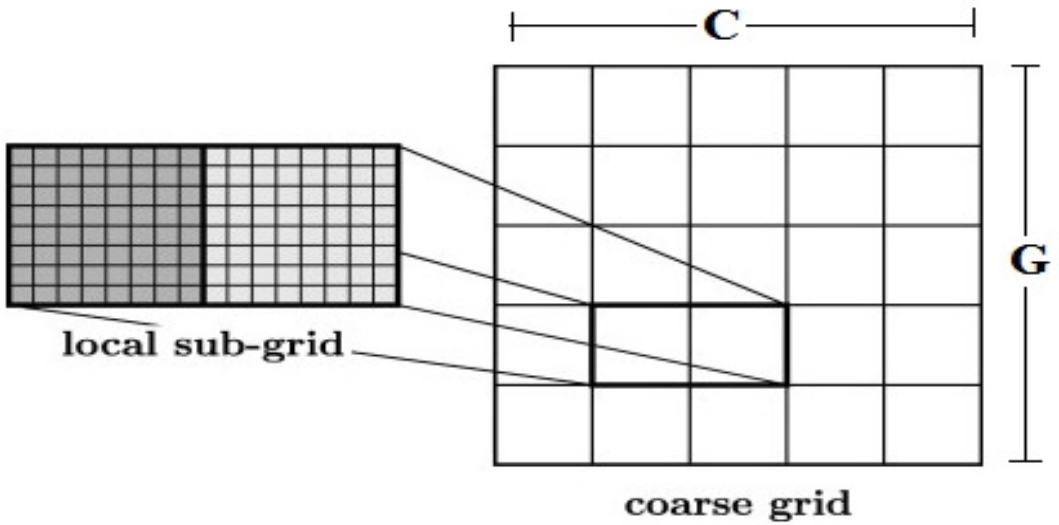
# Parameter Selection

- Effectiveness of SVM classifiers depends on the selection of the right set of parameters.
- Two different parameters for RBF-SVM: **soft margin parameter C** and **kernel width G**.

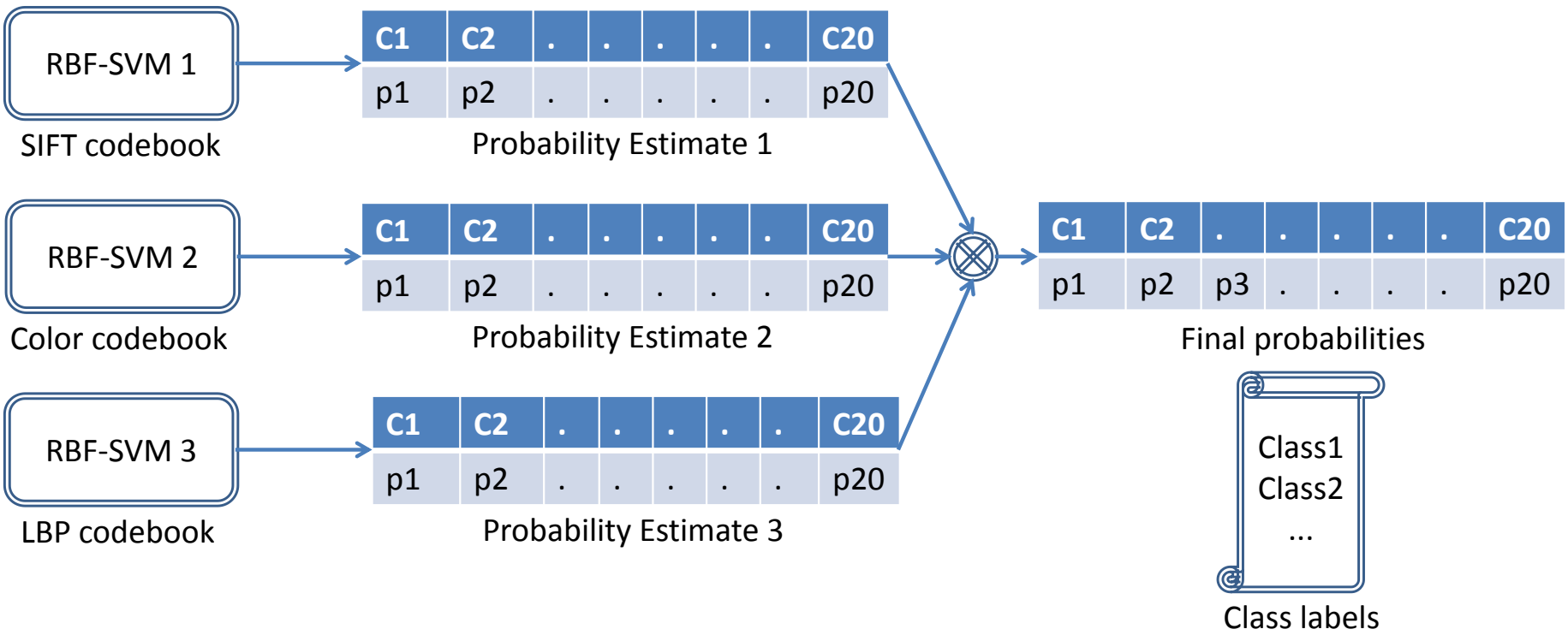


# Parameter Selection

- Effectiveness of SVM classifiers depends on the selection of the right set of parameters.
- Two different parameters for RBF-SVM: **soft margin parameter C** and **kernel width G**.
- Coarse-to-fine grid search (with 5 fold cross validation) for selecting the optimum set of parameters.



# Classifier Combination



# Results

- Top-5 accuracy:
  - **Training Set**: 99.99%
  - **Validation Set**: 88.55%
  - **Test Set**: 88.94%

| **Validation Set Accuracy (%)** | |
|---|---|
| SIFT | 81.55 |
| LBP | 82.6 |
| Color | 77.58 |
| All | **88.55** |

# Results

- Top-5 accuracy:
  - **Training Set**: 99.99%
  - **Validation Set**: 88.55%
  - **Test Set**: 88.94%
- Classes where the approach really works well: *odometer*, *rapeseed*, *website*.

| **Validation Set Accuracy** (%) | |
|---|---|
| SIFT | 81.55 |
| LBP | 82.6 |
| Color | 77.58 |
| All | **88.55** |

# Results

- Top-5 accuracy:
  - **Training Set**: 99.99%
  - **Validation Set**: 88.55%
  - **Test Set**: 88.94%
- Classes where the approach really works well: *odometer*, *rapeseed*, *website*.
- Classes where the approach does not work very well: *lopener*, *hatchet*, *cleaver*.

| **Validation Set Accuracy** (%) | |
|---|---|
| SIFT | 81.55 |
| LBP | 82.6 |
| Color | 77.58 |
| All | **88.55** |

# Success


{'odometer','spatula','gondola','hook','elocomotive'}

# Success

{'yflower','daisy','flamingo','ladle','plunger'}

# Failure

{'plunger','ladle','spatula','hook','cleaver'}

# Failure

{'odometer','daisy','flamingo','ladle','spatula'}

Questions?