Bag of Features Model for Object Recognition

CAP 5415 - Project

Kishan S Athrey

Overview – Bag of features model

- Extract features from the images at interest points using standard feature descriptors
  - SIFT
  - SURF
  - HOG
- Code word (Vocabulary generation)
  - K-means clustering on feature descriptors
- Image representation
  - Histogram
- Training a Classifier
  - Generative model – Naïve Bayes
  - Discriminative model – Random forest, SVM

Feature Extraction

- SURF (Speeded Up Robust Features) is used for feature descriptor
  - Same performance as SIFT, but faster than SIFT
  - Harris corner to detect interest points
  - LoG in SIFT – Approximated by box filter
  - Use of Integral images – For speed
  - OpenCV – SURF gives 64 and 128 dimension descriptor
  - Robust to scale, rotation and translation variance

Code-Word (Visual vocabulary generation)

- The (128 dim) feature descriptors are clustered using k – means clustering technique
- In my implementation
  - K = 500 [CUB Birds dataset]
  - K = 250 [Caltech 101]
- Value of K represents the size of the vocabulary
  - K = 100 -> 100 code words

Code-Word (Visual vocabulary generation)

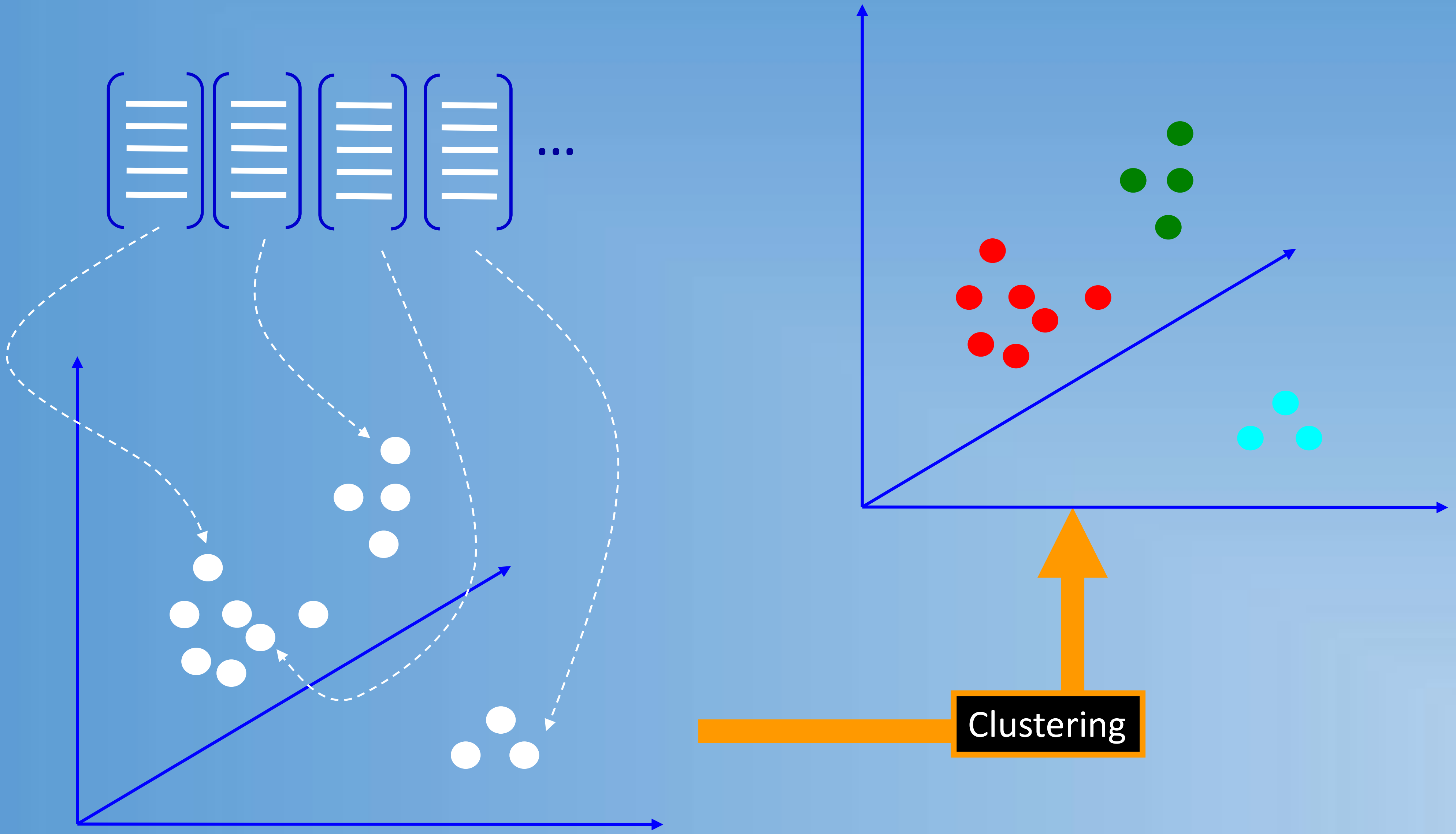
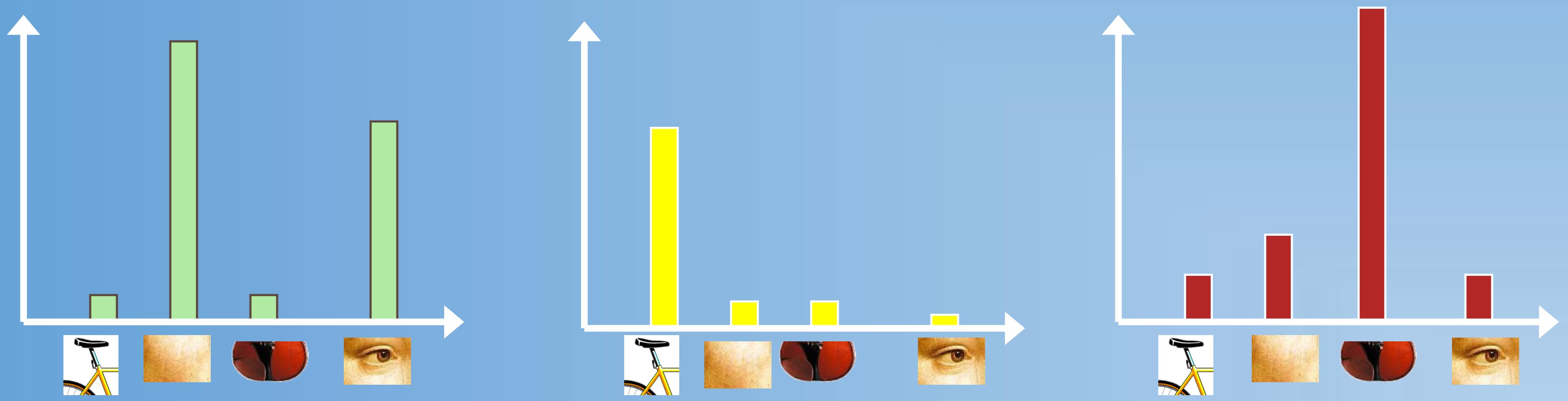


Image representation

- Each training image is represented as Histogram of code words
  - Feature descriptors of training images are compared with K-mean centers
  - These features are then quantized to compute histogram
  - Number of bins in histogram – Same as ‘K’ value



Training

- Classifier used – Naïve Bayes and Random forests
- Features (Input) for the classifiers are histogram vectors
- Generative Classifiers – Naïve Bayes
  - Multinomial NB
- Discriminative Classifiers – Random Forests
  - Trees – 500,100
  - Splitting Criteria – Entropy

Implementation and Results

- Datasets used
  - CUB Birds dataset
    - Trained on 10 classes of birds (approx. 650 images)
    - Tested on 10 classes of birds (approx. 480 images)
    - Accuracy using NB – 20.2%
    - Accuracy using RF – 21.4%

  - Caltech 101
    - Trained on 15 classes of objects (Airplane, animals, chairs, watches etc.)(approx. 1700 images)
    - Tested on 15 classes of objects (approx. 800 images)
    - Accuracy using NB – 66.4%
    - Accuracy using RF – 78.6% with K = 250

Please check last slide for detailed results

Observations

- BoF model works best when –
  - High inter class and intra class variation (Highly discriminative objects)
- BoF model suffers when –
  - Low intra class variation (Fails to discriminate b/w different species of birds)
- BoF can be improved by using global descriptors or by embedding spatial information
- Unbalanced dataset – Accuracy of under-represented class suffers
  - Use under sampling to increase accuracy of under-represented class
  - Use other evaluation metrics such as confusion matrix, precision and recall
  - Random forests works better than NB even with unbalanced data set

Implementational details

- Run time
  - CUB dataset
    - To generate code words with k = 500 – approx. 6min
    - To train and test with NB – approx. 3min
    - To train and test with RF – approx. 5min
  - Caltech101
    - To generate code words with k = 100 – approx. 12min (for k > 100 – 45min+)
    - To train and test with NB – approx. 4min
    - To train and test with RF – approx. 7min

Results

- CUB Birds Dataset

Category	Sensitivity	Specificity
Black_footed_Albatross	0.315	0.95
Laysan_Albatross	0.078	0.91
Sooty_Albatross	0.23	0.92
Groove_billed_Ani	0.2	0.88
Crested_Auklet	0.1	1.0
Least_Auklet	0.15	0.982
Parakeet_Auklet	0.5	0.83
Rhinoceros_Auklet	0.1	0.947
Brewer_Blackbird	0.4	0.914
Red_winged_Blackbird	0.4545	0.79

Results

- Caltech101 dataset

Category	Sensitivity	Specificity
Watches	0.72	0.87
Airplanes	0.56	0.88
Plant	0.76	0.95
Camera	1.0	1.0
Cars	0.514	0.99
Chairs	0.727	0.99
Coffee mug	0.731	1.0
Kangaroo	0.523	0.9
Sail boat	0.9428	0.94
Motorcycle	0.8194	0.98
Gun	0.2424	1.0
Football	0.6767	0.9
Starfish	0.357	1.0
Stop sign	0.75	1.0
Umbrella	0.357	0.99