Feature Encoding Strategies for Multi-View Image Classification

M.E.Sc Candidate: Kyle Doerr
Supervisor: Dr. Samarabandu
Co-Supervisor: Dr. Wang

January 26, 2016

Western Engineering

Electrical and Computer Engineering

Outline

- Motivation
- Contributions
- Background on image classification
- Feature Encoding techniques
- Triangulation Embedding
- Results
- Conclusion
- Future Work

Motivation

- Recent advances in computer vision
- Limited active research on machine vision problems
- Many manufacturing industries still rely on human inspection

Contributions

- A machine vision framework that can classify very similar automotive parts:
  - High classification accuracy is required to be useful in manufacturing environments
  - Near real-time classification
- An evaluation of feature encoding techniques:
  - With multiple view data
  - Results that generalize from machine vision to computer vision problems

Contributions

- A new object classification framework using binary ORB features that can classify very similar automotive parts real-time and provide a meaningful, robust solution for manufacturing environments

- A low cost view vision system using web cameras that can handle missing input

Introduction to the Problem

- The problem focuses on the correct classification of window pillars and guides
- Poke-yoke system

Figure 1.1: Images of window pillars, guides and their location on a vehicle

Introduction to the Problem

- Variations in length, orientation and luster

Introduction to the Problem

- A typical automotive plant can manufacturer several thousand of these parts a day
- Previously, correct labelling of these parts was left to humans:
  - Difficult to identify small similarities on a moving belt
  - Human fatigue during an 8+ hour shift
  - More than one vehicle make with several models

Background: ORB Features

- Perform feature detection by finding corners
- Using a pre-defined sampling pattern compare patches, patch **p** at point **x** to build the descriptor

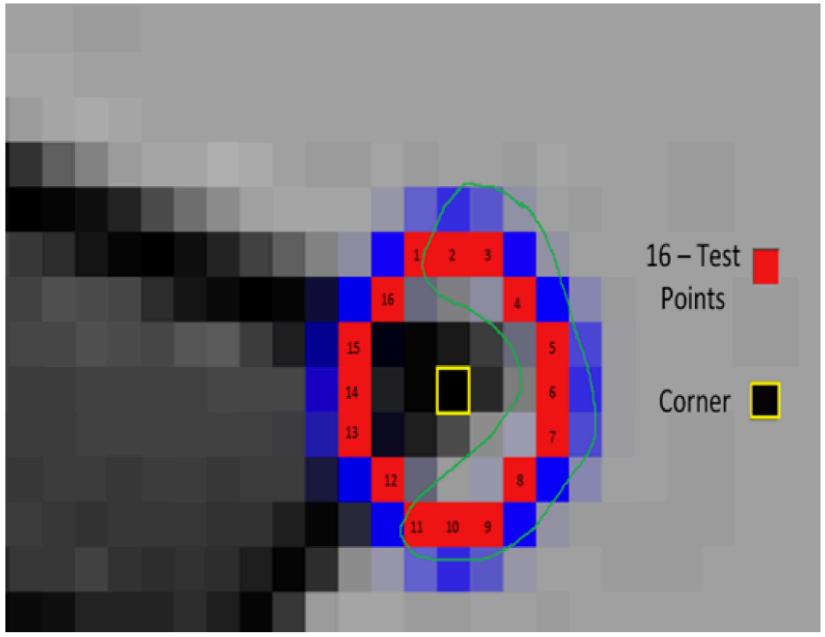


Figure 2.3: A visual of the FAST corner test

$$\tau(\mathbf{p};\mathbf{x},\mathbf{y}) = \begin{cases} 1 : \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 : \mathbf{p}(\mathbf{x}) \geq \mathbf{p}(\mathbf{y}) \end{cases}$$

$$f_n(\mathbf{p}) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(\mathbf{p};\ \mathbf{x}_i,\ \mathbf{y}_i)$$

Background: ORB Features

- Rotational Invariance by rotating the sampling points by an intensity centroid
- Scale invariance by performing this process using image pyramids

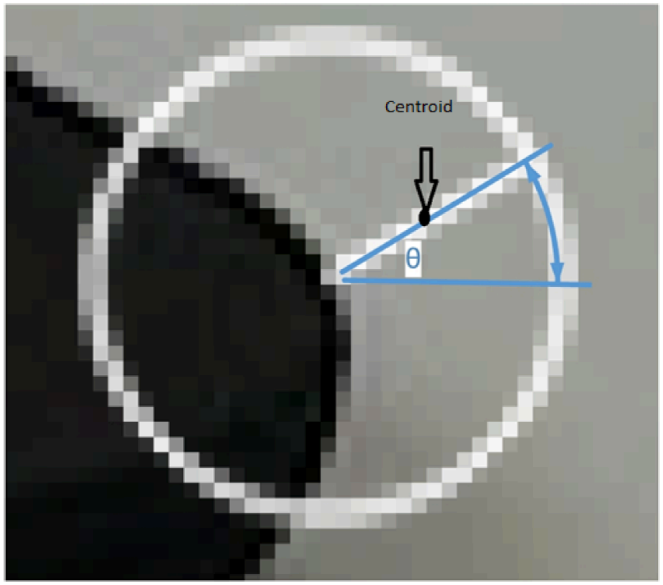


Figure 2.6: A sample showing where the centroid is located within an image

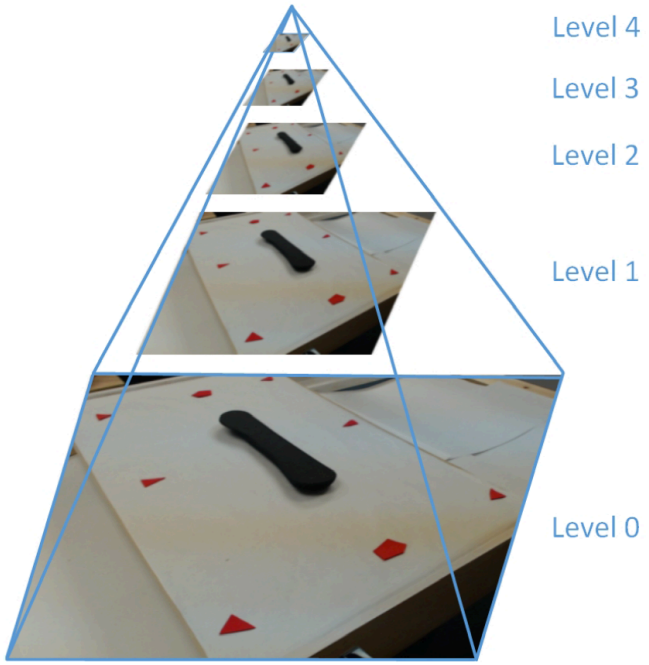
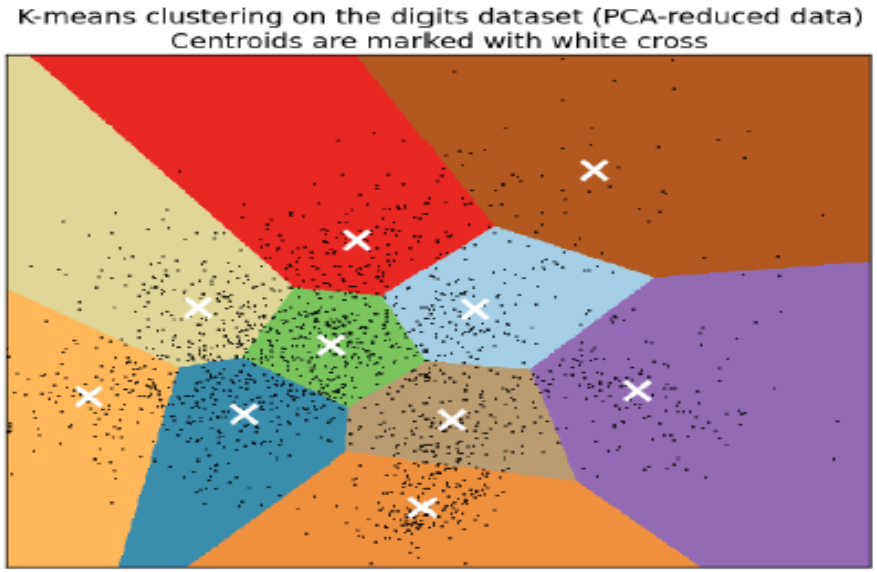


Figure 2.5: Features are produced at each level of the pyramid

Background: Bag of Words

- On an offline dataset perform feature extraction on a set of images to build a feature set
- Run the unsupervised K-means clustering algorithm to construct the Visual Vocabulary
- The centroids form the visual words



Background: Bag of Words

- The formulation of the visual vocabulary now allows for building a training set
- Features are matched to centroids by Nearest-Neighbour assignment

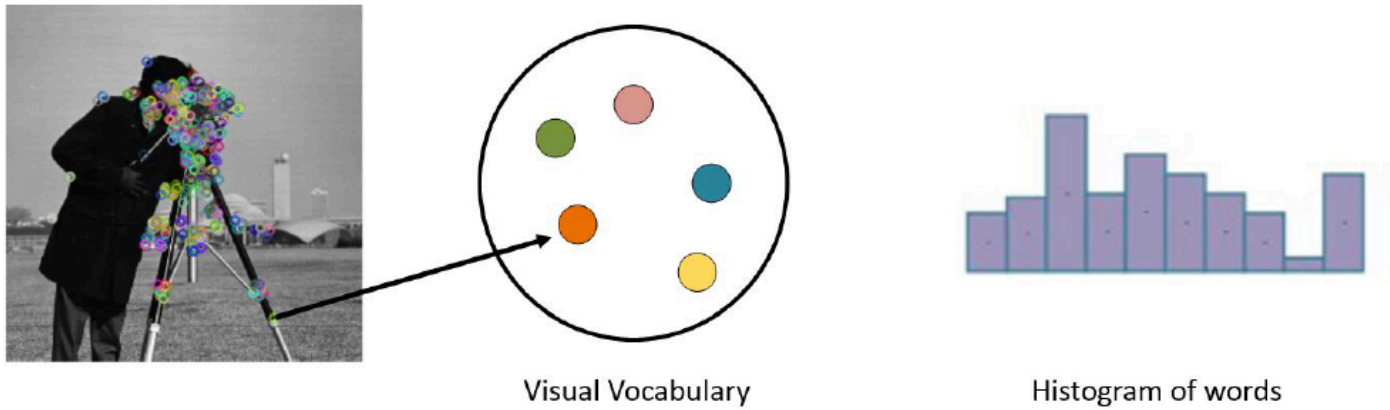
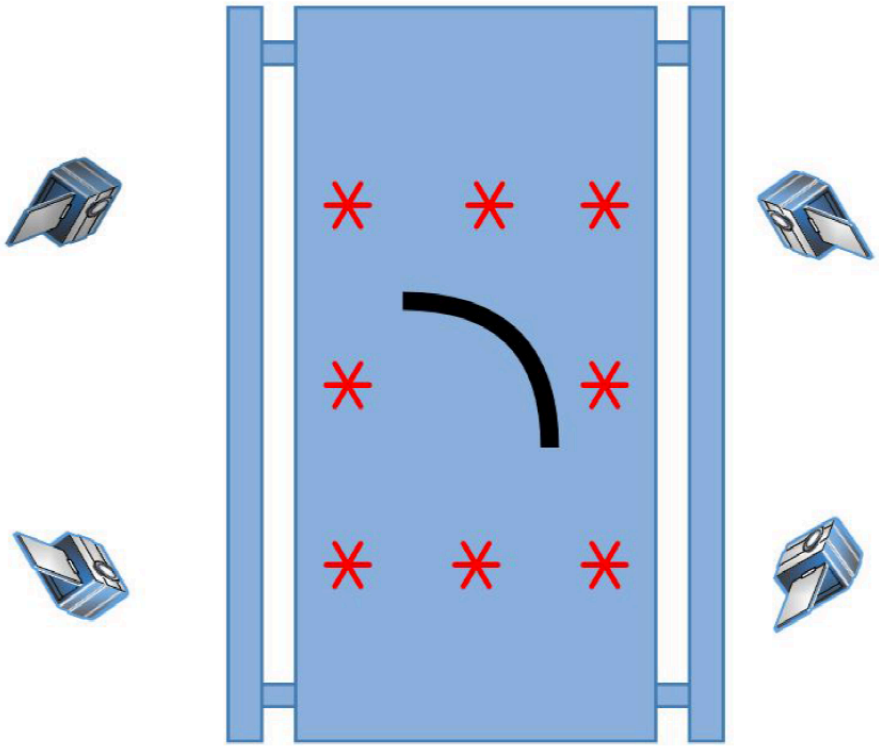
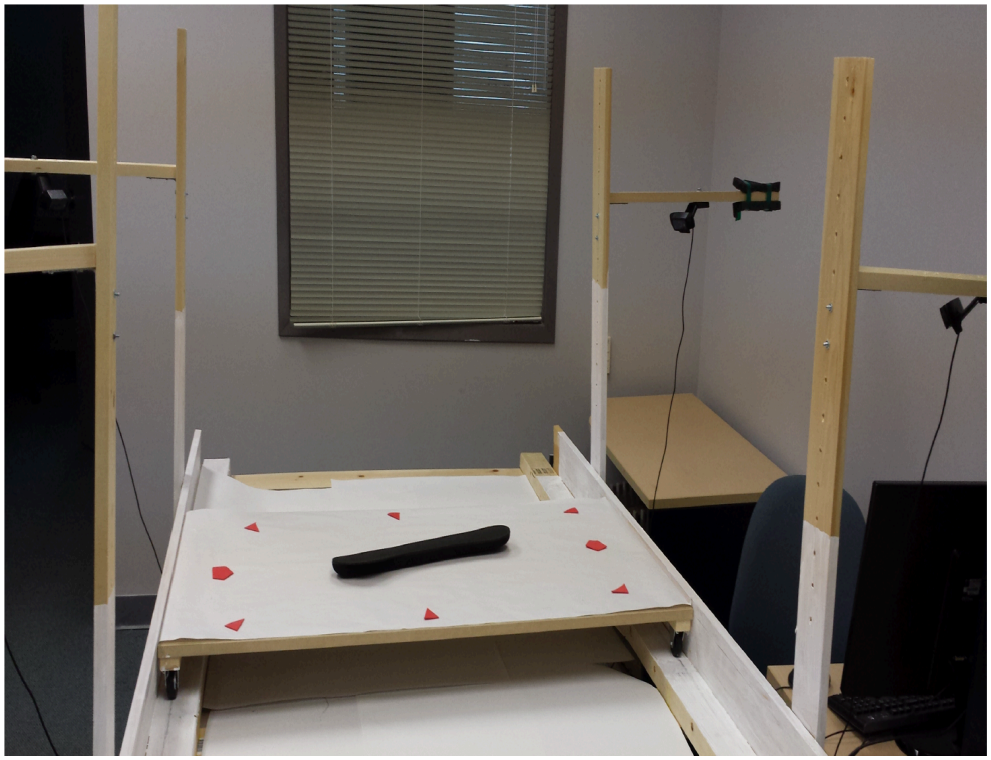


Figure 2.8: The process of descriptor encoding

Proposed System





Datasets – in house

-39 classes of automotive parts – window & guides

-Differences in length, orientation, and luster

Figure 4.3: Window guides of varying length

Figure 4.4: Window pillars with left and right orientation

Datasets – CalTech3D (Fei et al.)

8 view dataset with 8 object categories

- 10 object instances with varying heights and scales

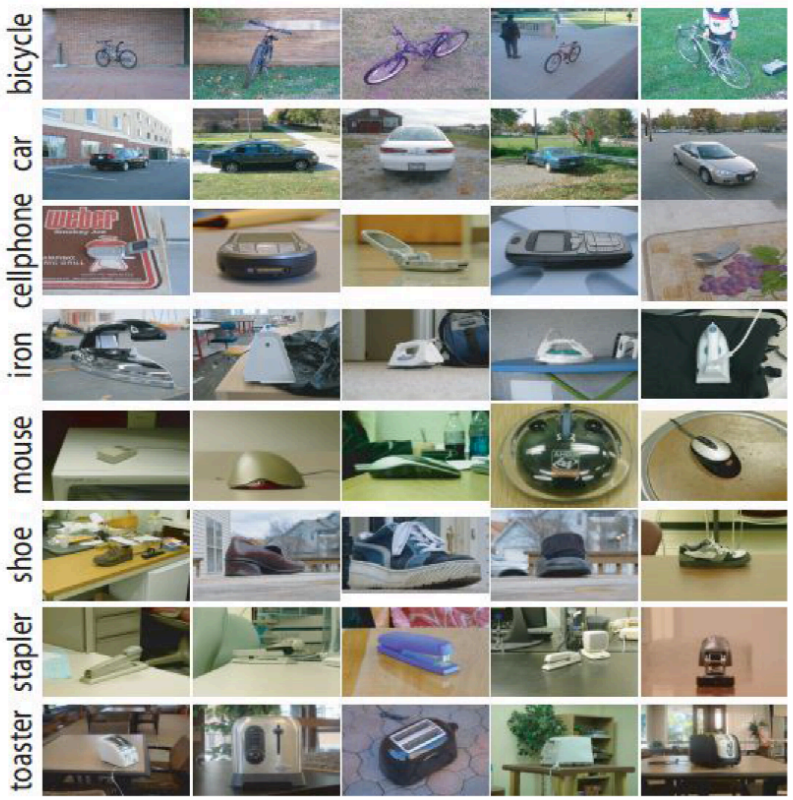


Figure 4.5: A diagram showing samples of the CalTech 3D dataset

Problem 1 – Feature Selection

- How does the selection of feature types affect classification performance?
  - ORB, SIFT, SURF

Results: Single View (in-house)

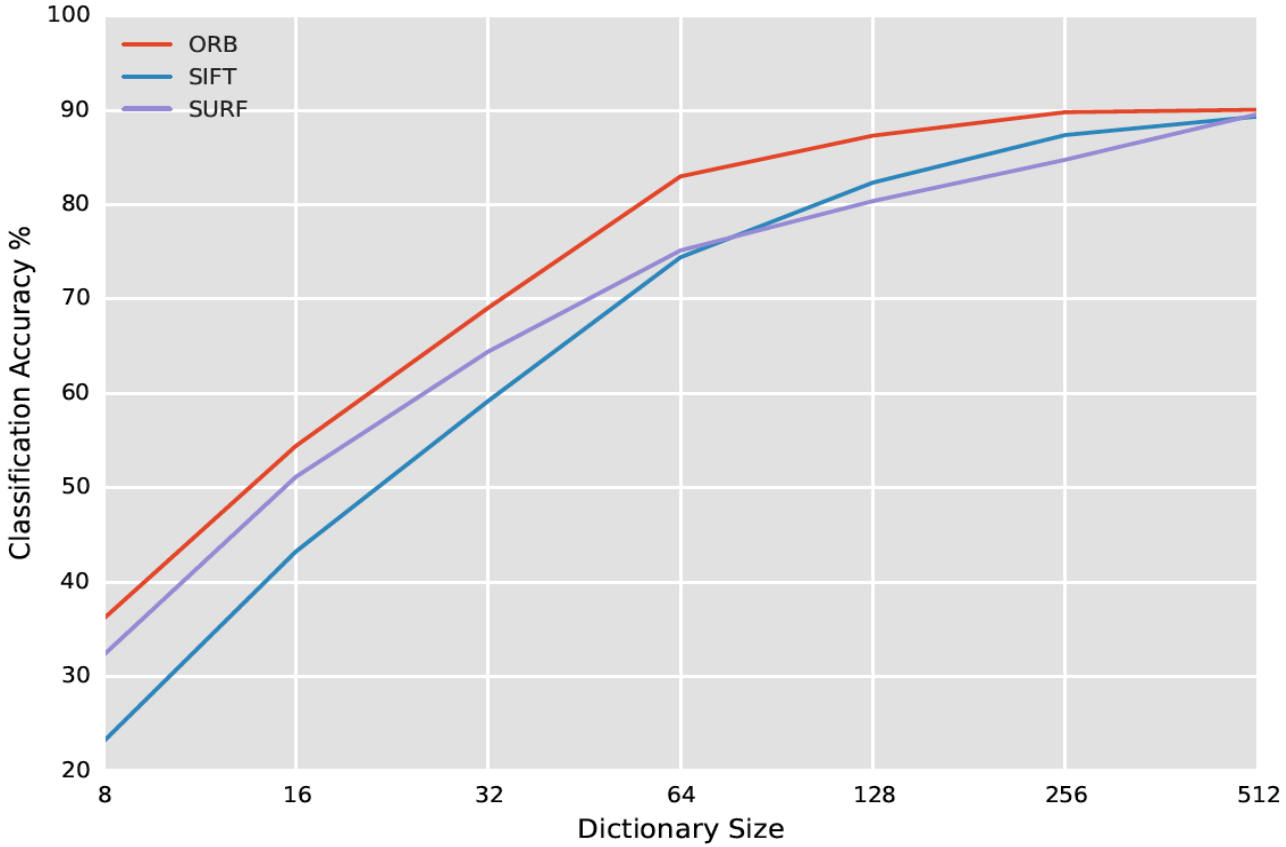
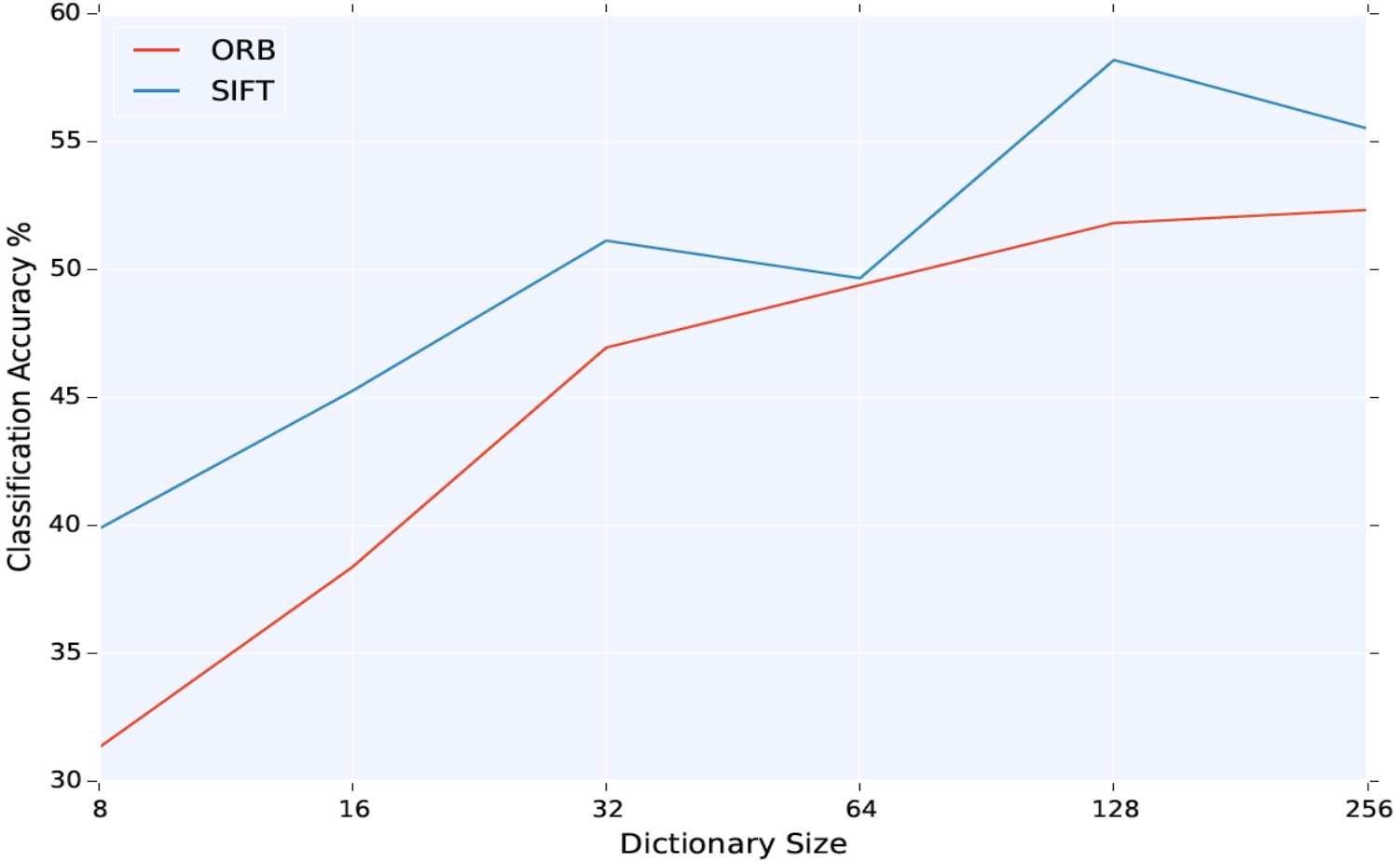


Figure 4.6: Classification performance on in-house dataset using the BoW model comparing SIFT, SURF and ORB features against vocabulary size using a single view

Results: Single View (CalTech)



Analysis

- ORB showed superior classification performance for in-house dataset but not CalTech:
  - CalTech dataset had much larger changes in scale and viewing angle
  - ORB appears to be more suitable in controlled environments
- Accuracy as a function of dictionary size:
  - Diminishing returns
  - Trade off of increased classification time
- ORB showed a classification speed increase of ~6.5x SIFT and ~3.4x SURF
- Avoidance of license restrictions – both SIFT & SURF are patent protected

Problem 2 – Single vs Multiple Views

- Car parts ~90% classification accuracy
  - Not acceptable if thousands of parts are processed per day
- Discriminating features are occluded
  - Discriminative physical features may not be visible in the one view

- Multi view representation is challenging:
  - Depth information / 3D reconstruction
  - Computationally expensive
  - Many techniques require camera calibration
  - Feature correspondence is challenging for this particular dataset

Problem 2 – Single vs Multiple Views

- Proposed simple approach of concatenation of feature vectors over multiple views

- Construct a feature vector for each view independently then concatenate

For i =1 to n views:

Fvec := $fvec_i$

Results – Single vs Multiple views

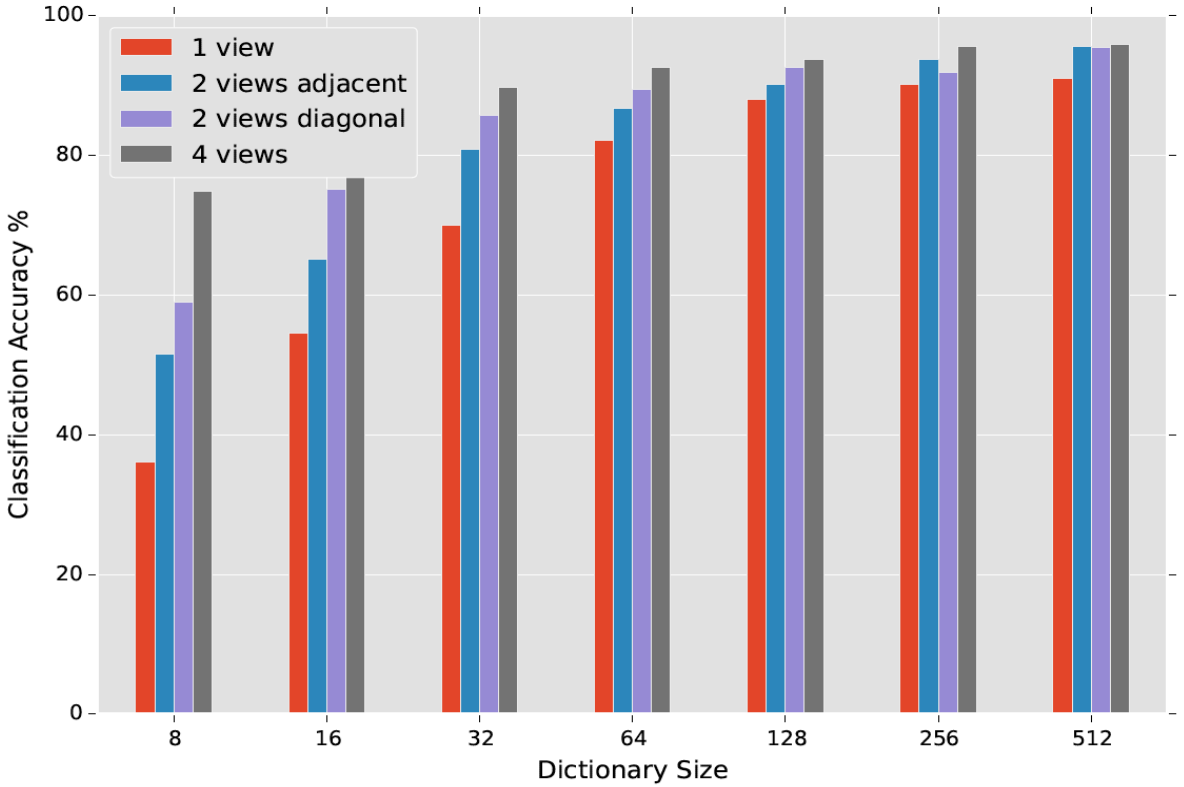
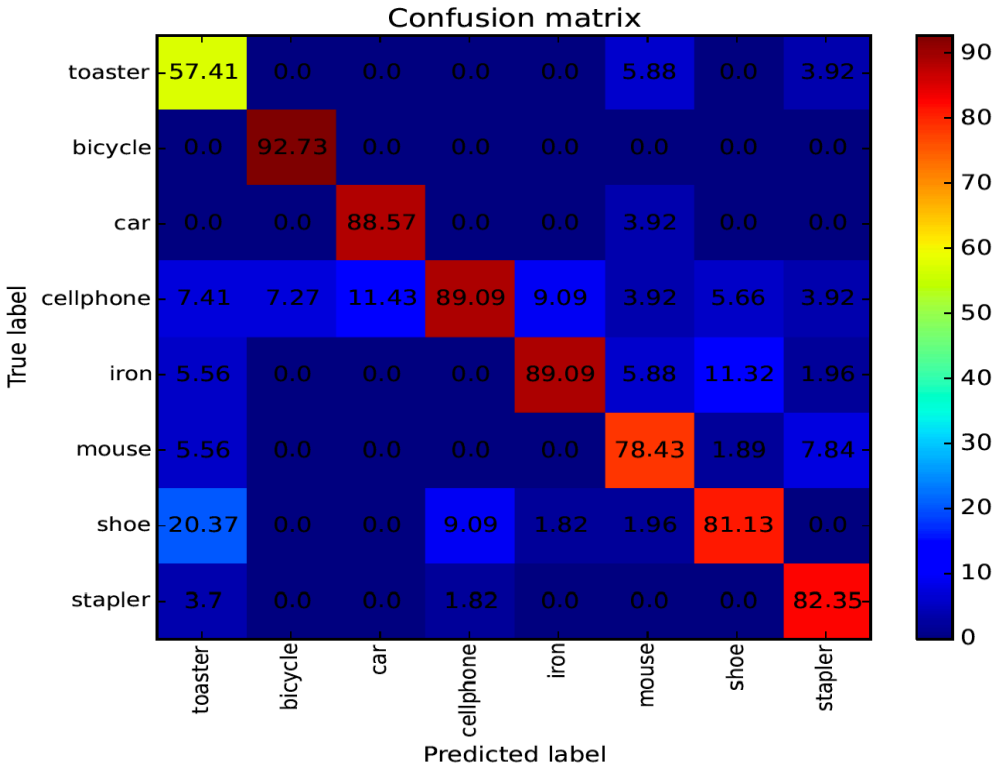
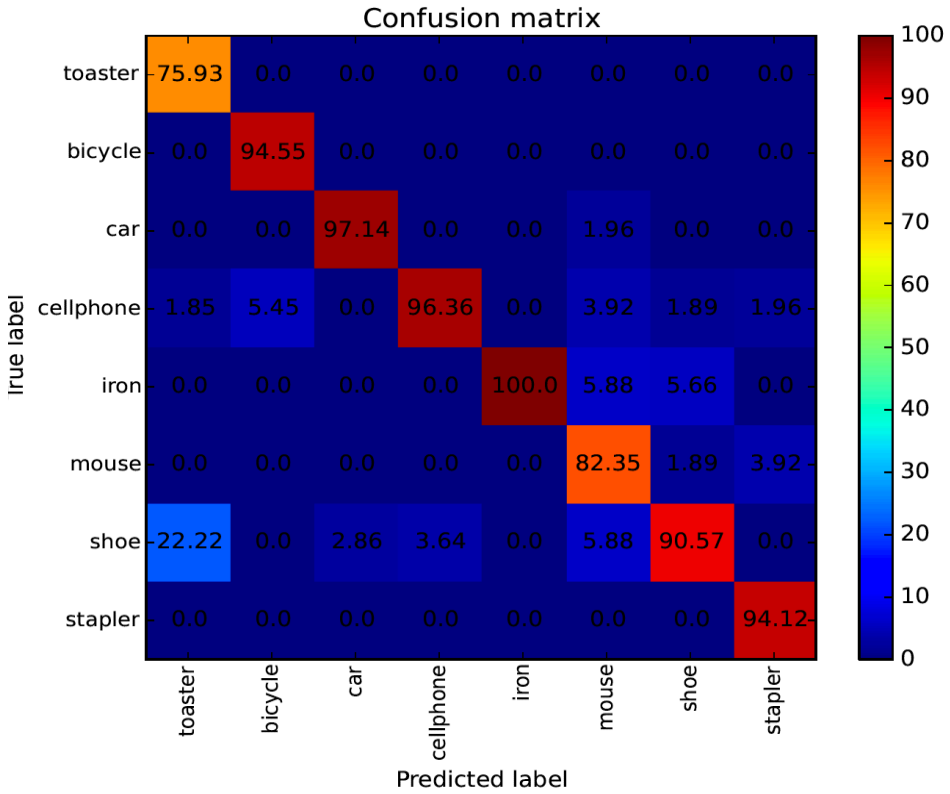


Figure 4.8: Classification performance showing the impacts of using different combinations of views and dictionary size on in-house dataset

Results – Single vs Multiple views





Significance

- Classification on both datasets improved when there were more views
- Increasing the number of views increases the chance of capturing features that helps discriminate among similar objects
- Analogous to spatial histograms but incorporates geometry at the view level (next slide)

Spatial Histograms

- The Bag of Words model is order less representation
- Spatial histograms help circumvent this

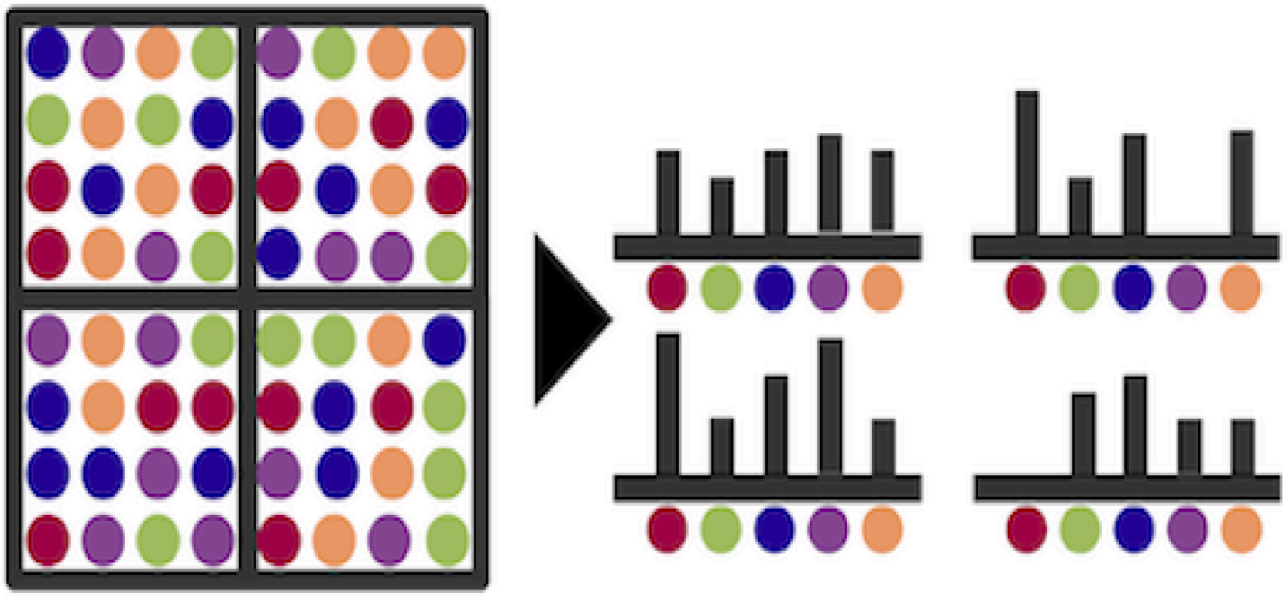


Figure 4.17: An example of spatial histograms

Spatial Histograms

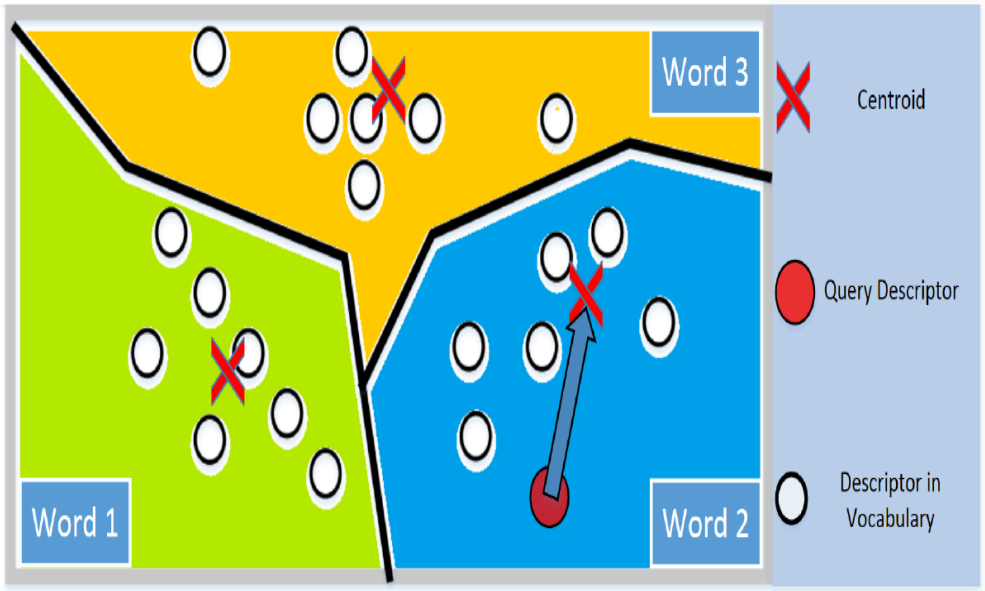
- It is widely understood that spatial histograms improve accuracy

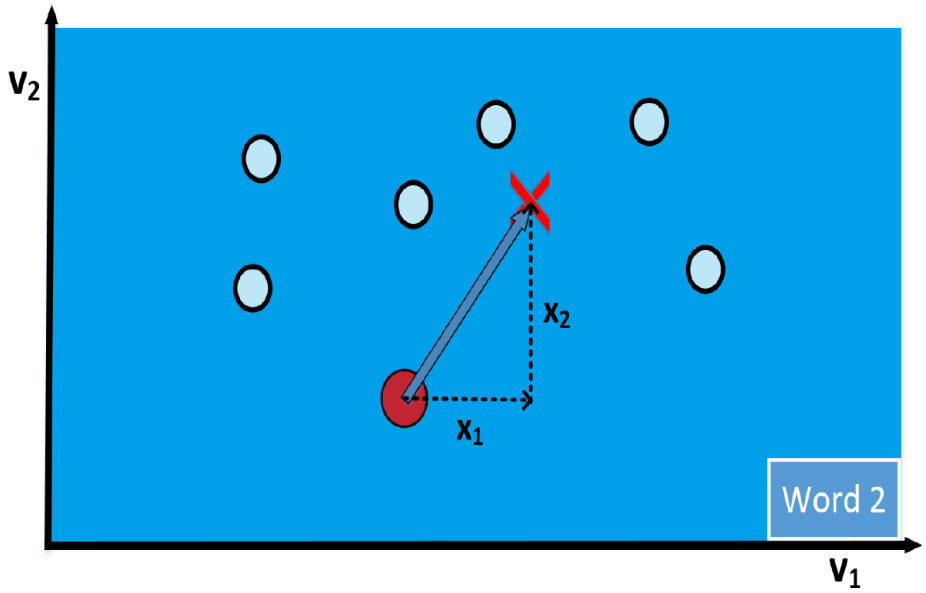
- Using this ‘spatial histograms’ among views also achieves this goal

Vector of Locally Aggregated Descriptors

- Extension to the Bag of words model
- No longer encodes features as just a histogram of word counts
- Contains ‘first order information’
  - Word assignment + distances to centroid
- Dimensionality k*d:
  - k = number of centroids
  - d = dimensionality of descriptor

Vector of Locally Aggregated Descriptors





$$v_{i,j} = \sum_{NearestN(x)=c_i} \mathbf{x}_j - \mathbf{c}_{i,j}$$

$v = [0, 0, x_1, x_2, 0, 0]$

VLAD Experiment

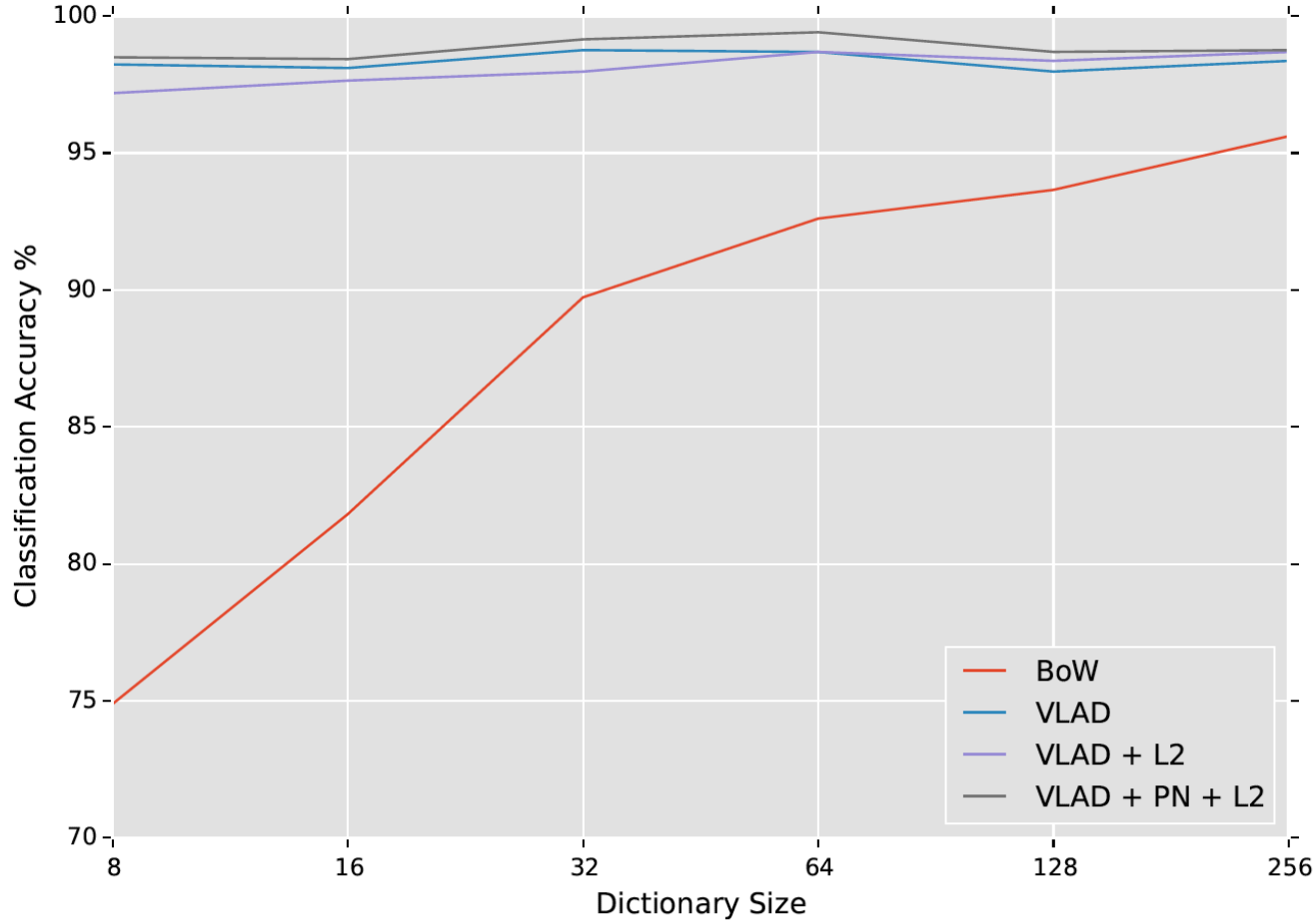


Figure 4.21: The results of the in-house dataset using the VLAD encoding approach

VLAD Results

- Great accuracy improvement even when the visual vocabulary is small
  - Peak accuracy of 99.4% compared to 95.6%
  - Accuracy is great enough to be of value for machine vision applications

- Demonstration that VLAD encoding can be used with multiple views

- Evidence that despite being binary, ORB features can still undergo VLAD encoding
  - Prior works used primarily SIFT
  - Considerable speedup for training and classification as resulting feature vector is always ¼ that of SIFT

Triangulation Embedding

- Extension of the VLAD representation but uses directional information over absolute distances
- No longer performs a brute-force search to assign descriptors to centroids – but rather uses residual distances among all words
- Two main steps: 1-Embedding and 2-Aggregation

Triangulation Embedding

- Embedding:
  - Define an embedding function

$\phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$ $(d < D)$ maps each descriptor $x \in \mathcal{X}$ as:

$$x \mapsto \phi(x)$$

- Compute residuals and unit normalize:

$$r_j(x) = \left\{ \frac{x - c_j}{\|x - c_j\|} \right\}, \quad \text{for } j = 1...|C|, \; x \neq c_j$$

Forming the set R(x)

Triangulation Embedding

- The set of residuals then undergo a *whitening* process: center, scale and rotate

$$\phi_{\Delta}(x) = \Sigma^{-\frac{1}{2}}(\mathrm{R}(x) - \mathrm{R}_0)$$

- $\mathrm{R}_0 = E_X[\mathrm{R}(x)]$ and Σ – covariance matrix both learned on a training set → looks very similar to PCA

$$\phi_{\Delta}(x)' = diag(\lambda_1^{-\frac{1}{2}}, ..., \lambda_D^{-\frac{1}{2}})\ P^T\ \phi(x)$$

Triangulation Embedding

$$\phi_{\Delta}(x)' = diag(\lambda_1^{-\frac{1}{2}}, ..., \lambda_D^{-\frac{1}{2}}) \; P^T \; \phi(x)$$

- ϕ(x) - encoded features
- P – DxD matrix of eigenvectors of covariance matrix, sorted by corresponding eigenvalue from greatest to least
- λ – Eigenvalues

*assuming features are stored as column vectors
*mean subtraction of ϕ(x) has already taken place

Triangulation Embedding

- Aggregation by summation:

$$\Phi_{\Delta} = \Sigma \, \phi_{\Delta}(\mathrm{x})'$$

- Feature vector of length k*d where k is the number of centroids and d is the feature dimensionality
- Typically setting k=d is the standard

Triangulation Embedding Results

Approach	Accuracy (%)
BoW + l^2	95.62
Histogram Intersection Kernel	98.37
X^2 (additive)	98.43
Spatial Pyramids	96.08
VLAD + l^2	98.76
$\Sigma R(x)$	96.08
Φ_Δ	99.28

Table 4.5: Comparison of the Triangulation Embedding method on the in-house dataset

Triangulation Embedding Results

Approach	Accuracy (%)
BoW $+l^2$	82.15
Spatial Pyramids	93.63
VLAD + l^2	88.74
$\Sigma R(x)$	83.53
Φ_{Δ}	94.38

Table 4.6: Comparison of the Triangulation Embedding method on the CalTech 3D set

Triangulation Embedding Analysis

- The comparison of images is done by use of a similarity score:
  - Compared by feature vectors
  - Cosine similarity is the most common metric
  - Similarity is between [0 – 1] from least to most similar

- Classification is ideal when an instance is similar to its members within the same class and dissimilar to objects of other classes

Triangulation Embedding Analysis

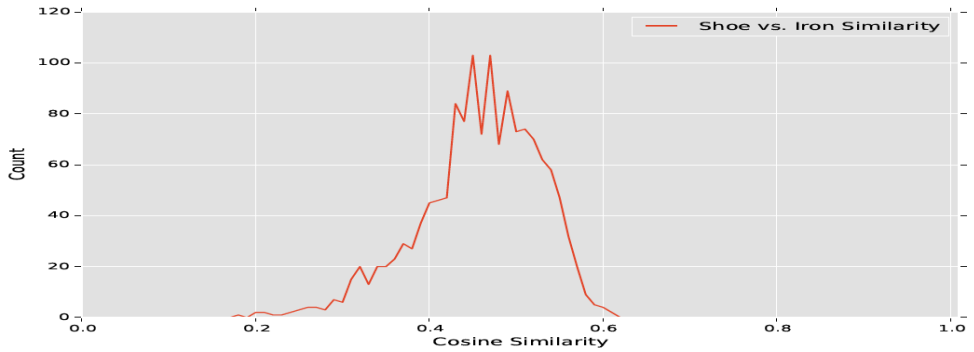


Figure 4.25: Similarity scores histogram of Shoe vs. Iron classes using BoW

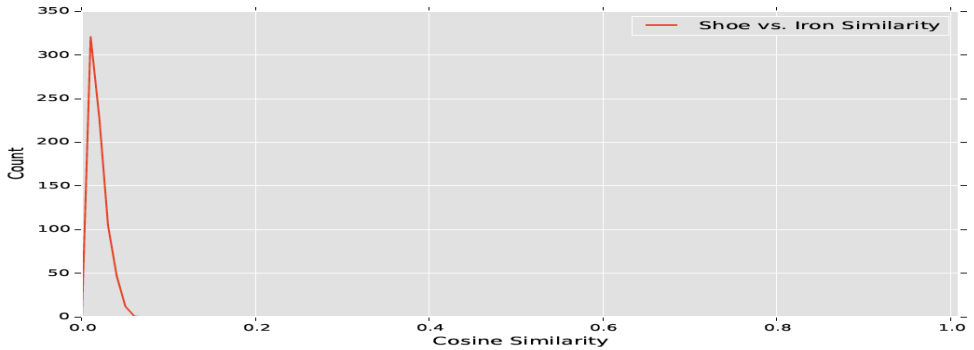


Figure 4.26: Similarity score histogram of Shoe vs. Iron classes using Triangulation Embedding

Triangulation Embedding Significance

- Outperforms other encoding methods
  - Better accuracy then BoW & VLAD
  - Best accuracy on both datasets

- This method generates a more discriminative feature vector

- This encoding method is nearly guaranteed to perform better over the standard BoW model

Camera Downtime

- All machines break down eventually
- Harsh manufacturing environments:
  - Dust
  - Heat
  - Physical damage
  - Strenuous use (i.e. running 24x7)

- Downtime can stop inspection or manufacturing

Camera Downtime

- Propose to see how well the system can maintain classification by disabling an input on a random view
- Apply power-law normalization to handle bursty features

- Power-law normalization (PN):

$$f(z) = sign(z)\,|z|^{\alpha} \qquad 0 \le \alpha \le 1$$

Camera Downtime Results

Approach	Accuracy (%)
BoW	94.51
BoW + l^2	93.72
BoW + PN	96.60
$\Phi_\triangle$	98.56
$\Phi_\triangle$ + l^2	98.56
$\Phi_\triangle$ + PN	98.89

Table 4.7: Comparison of the Triangulation Embedding method on the in-house dataset

Approach	Accuracy (%)
BoW	75.59
BoW + l^2	75.57
BoW + PN	74.87
$\Phi_\triangle$	91.69
$\Phi_\triangle$ + l^2	91.21
$\Phi_\triangle$ + PN	91.94

Table 4.8: Comparison of the Triangulation Embedding method on the CalTech dataset

Camera Downtime Significance

- Minor decrease in accuracy in the event of a damaged camera:
  - Indicates a damaged input would not have to result in disabling the vision system
  - In-house dataset only experienced a 0.6% drop in accuracy using triangulation embedding method with a missing input
  - On both datasets the reduction in accuracy was only half that compared to the BoW method

Encoding Rates

- The time it takes from feature detection to generating a final feature vector
- Intel i7 machine using Python3 with openCV & Numpy libraries

Approach	Encoding Rate (images/sec) (%)
BoW	18.64
VLAD	17.78
$\Phi_{\triangle}$	12.90

Table 4.9: Comparison of the feature encoding rates

Conclusions

- This work evaluated the usefulness of the binary ORB features and how it compares to SIFT and SURF

- The presentation of a framework for a machine vision systems for object identification

- This framework is extended to use multiple views while also evaluating newer feature encoding strategies that greatly outperform the traditional BoW model

Conclusions

- The system itself can classify fine-grained object categories at near real-time speeds

- The vision system is robust enough to handle a damaged camera without disrupting operations

- Engineering perspective:
  - Improved quality control
  - Addressed the risk of possible component failure
  - Designed an inexpensive solution – total cost of cameras ~$400

Future Work

- Integration of depth sensors
- Convolutional Neural Networks
- Increase processing speed:
  - GPU computing
  - Evaluate ORB descriptors in Hamming Space

Acknowledgements

- Dr. Samarabandu & Dr. Wang
- Sightline Innovation Inc.

Q&A

Western

UNIVERSITY · CANADA