BUILDING IDENTIFICATION USING AR. DRONE

Dhruv Kumar Yadav 11253
dhruvkr@
Smith Gupta 11720
smith@

CS365A Artificial Intelligence
Mentor: Prof Amitabha Mukerjee
Department of Computer Science and Engineering

Abstract

Quadcopters have grown immense popularity in the recent times in various fields of active research since they have huge potential in search and rescue operations, sports and autonomous tracking. One of them is to understand the terrain and identify its location in GPS denied environment. In this project we propose a novel Building detection method in the terrain of IIT Kanpur. Training data is collected by segmenting recorded aerial videos of buildings of IIT Kanpur using front camera of AR Drone. We first employed scale invariant feature extraction from the training images using SIFT algorithm. Then bag of Visual Words is applied for representing each image in the form of a visual word and these words are then clustered using K-means algorithm. Finally One vs. One Binary SVM Classification algorithm is used for learning the classifier. We finally show how proposed method give interesting performances.

Keywords: *AR Drone, Identification, Multi-class Classification, Bag of Visual Words, Support Vector Machine (SVM), Posterior Probability Estimation*

1. INTRODUCTION

In recent years Unmanned Aerial Vehicles (UAVs) have attracted many research scholars and has become a research interest topic. This is mainly due to the fact that they have a subtle advantage of movement over any terrain that helps them to navigate easily in hostile environments making them essential in tasks like autonomous navigation, object following, position stabilisation, and has wide applications in disaster management, surveillance and terrain mapping.[1]
AR Drone is one such UAV and in particular a quadcopter which is being widely used mainly due to the open source features that come along. One can get all the required software packages that are available freely on internet so that they overcome the initial glitches and directly focus on more advanced problem. There are exhaustive libraries available for AR Drone which can be used for its smooth control as well as for performing advanced functions through Drone.[1]
We focus on determining location of the drone in a GPS denied environment which is achieved using identification of surrounding environment through image classification of surrounding buildings. Our project has been inspired from Mission 4 in International Aerial Robotics Competition (IARC) which involved flying to an abandoned building and identifying a particular structure based on a symbol on the building. Our Project not only aims to identify the structure but also differentiates between sides of the structure. The identification task is done at run time i.e., during the flight of the drone.

Image classification problem deals with assigning a class label to the image. Difficulty faced during classification task is the inconsistency due to external factors like irregular movement of Drone, camera position, changes in illumination or internal parameters like variations within object. For the task, we use One vs. One binary classification where among any two classes positive training images consist of images of the object class whereas negative training images consist of images not containing the object class. Given a test image, we classify whether it contains object class or not.

2. MODEL

Model consist of two phrases-Testing and Training.

2.1. ***Training Model***

We used AR Drone to capture high resolution videos of various buildings across Indian Institute of Technology Kanpur (H. R. Kadim Department of Computer Science and Engineering, P. K. Kellkar Library and Department of Industrial and Management Engineering). Our objective is to enable A R Drone to automatically identify its location in GPS denied terrain by exploring the environment around. In particular, we focus on identification of various buildings across IITK Campus.

First step is o obtain the images from the training video. We used ffmpeg liberary for the same. Refer section 3 for more details.
The image data set is then used to model the high level characteristics of buildings so that we can distinguish between them. We employ three step approach to learn the features corresponding to each building i.e. extracting the feature vectors, create bag of visual models and classify using Multi-class Support Vector machine.

- ***Feature Extraction***
  Features are the interesting points that can be easily extracted from an image and are used to locate the object. Desired properties of local features are [2]
  - Repeatability : Inspite of geometric or photometric shift, we should be able to find same features in two images of same object.
  - Distinctiveness : Each feature should have distinctive property
  - Compactness and Efficiency : To take into account real time applications detect fewer features than number of pixels.
  - Locality : Feature should occupy small area of image so that it is robust to clutter or occlusion

  Since our project is based on categorisation based on different classes of buildings, we wish to find the features of buildings in a specific class.
  We used Scale Invariant Feature Transform (SIFT) descriptor in the given image to compute key points and descriptors. SIFT allows to detect the interest points under scale, rotation and noise. David Lowe first gave the idea of SIFT features and since then SIFT features have been extensively used in various fields of Computer Vision since then.
  [3] SIFT algorithm, as given in Lowes paper, involves extraction as a four step process. The details of the extraction can be found at David Lowes paper. [3]
  We employed dense sift rather than the sift described by Lowes algorithm. Dense sift compute descriptor over dense grid of pixels. For category recognition, it has been found that dense sift features give better result than sift features. Note that this typically generates large set of features.

- ***Bag of Visual Words***
  Bag of visual words model is analogous to bag of words model in documents. In the later, we define a dictionary having total of k words. Each document is represented as histogram of words, storing the frequency of each word call bag of words. Note that order does not matter in document representation using bag of words model. In the images we need to define vocabulary or visual words in order to represent each image in

terms of those visual words. We take the output (feature descriptors) from feature extraction in the first step using SIFT features. Next step is to quantize descriptors into visual vocabulary. K-means clustering algorithm is employed and each cluster is assigned to words to obtain dictionary of k-visual words. K-means algorithm is explained below. [4]
K-means Algorithm for clustering

- Choose k data points to act as a cluster centres
- Until the cluster centres are unchanged
  - Allocate each data point to the cluster whose center is nearest
  - Replace each cluster center with the mean of the elements in their cluster

  end

Next each image represented as histogram of these visual words. These histograms are used for classification.

- ***Multi-class SVM*** In this step we need, we need to learn a classifier which assign bag of visual words into different classes. Positive and negative vectors for the classifier are the histograms, one from each training image. The classifier here learns all the patterns and regularities in the input vectors. With successful training, classifier is able to classify a test sample into correct class with high accuracy. As dimension of input vectors is large, we use support vector machine (SVM) for classification. In the binary classification task, we need to learn the classifier for two classes only (assign label +1 and -1 to the classes). The job of the SVM classifier is to find the best suitable hyperplane to distinguish between the two classes. The details of Linear SVM Classification and mathematics involved can be found at A SIFT-SVM method for detecting cars in UAV images.[5]
For multi-class classification, there exist two strategies -One vs all or one vs one. Since one vs all classifier needs to deal with all the data of all the samples thus consuming much time; we are planning to adopt one vs one method which improves the speed of classification a lot.[5]

The Final Algorithm can be summarised as below:

1. Segment each video after fixed number of frames to obtain training images
2. Randomly sample 1000 points in each image and compute SIFT descriptors at every point
3. Cluster the descriptors into visual words using k-means thus generating vocabulary of words
4. For training image again compute dense sift at all the grids.
5. Each image is described as histograms of words. Since total words are fixed each image is described in same vector space
6. One-vs One Binary SVM Classifier is learnt for each class. Thus for c classes; c(c-1)/2 classifiers are learnt.

2.2. ***Testing Model***

For the test image, we first represent the image as histogram using the original vocabulary. Histogram is given as input to each learnt classifier and corresponding scores are obtained. Given many input test images; the task is to model the output from multi-class SVM to select one class over the other.
The general way is to use voting of multiple one vs one classifiers. However, that can be affected by noise, so one way is to vote by using magnitude or thresholded magnitude (i.e. distance from hyperplane). Other more sophisticated techniques involve converting the distance from margin to probabilities and taking into account the probabilities.
Given a test image, its test score is calculated through which probabilities are assigned to each class and maximum out of them is selected. Method proposed in [6] is used which assigns probability to each class for given test score in the following manner,

$$P(w_i|(X=x)) = \frac{1}{\sum_{j=1}^{K} \frac{1}{P(w_i|w_{ij}^{l}(X=x))} - (k-2)} \quad (1)$$

where K is the total number of classes and $Pr(wi|wij^(X = x))$ refers to probability of test image belonging to class wi given the test score x when classes wi and wj are pairwise considered in One vs. One scenario. These $Pr(wi|wij^(X = x))$ is calculated via the Bayes rule,

$$P(w_i|w'_{ij}(X = x)) = \frac{p(x|w_i)P(w_i)}{p(x|w_i)P(w_i) + p(x|w_j)P(w_j)} \quad (2)$$

where $p(X = x|w_i)$ and $p(X = x|w_j)$ are class-conditional densities that are governed by probability distribution of classes in each of the One vs. One classification pairs which we calculate using distribution of training images over the range of test score. Normalization is done at various steps to make probabilities of classes exhaustive.

3. IMPLEMENTATION DETAILS

3.1. Training

- **Video Segmentation**
  The implementation of project involves use of matlab and ffmpeg liberary. The video is first segmented into various images at the frame rate of .5 (i.e. a image per 2 seconds of clip) using ffmpeg library. The ffmpeg is a software that handles multimedia data (decode, encode, transcode, mux, demux, stream, filter) [7].

- **SIFT and Bag Of Visual Words**

  – First we need to construct vocabulary of visual words from the input images. Vocabulary is a structure with fields:
  WORDS:: 128 x K matrix of visual word centers
  KDTREE:: KD-tree indexing the visual word for fast quantization

  – Number of words are fixed to 1000 and number of features is 100 times number of words. Only 100000 overall descriptors are retrieved as more do not really improve the estimation of the visual dictionary but slow down computation.[8]

  – We used vl-phow from VL-FEAT library[9] in matlab that extracts phow features(dense sift applied at various resolutions) from images.

  – This clusters the descriptors into NUMWORDS visual words by using KMEANS. It then computes a KDTREE to index them. This speeds-up quantization significantly

  – Vocabulary is used to construct spatial histograms for each class of the image (H.R. Kadim CSE department, P.K. Kellkar Library and IME) and also for the subclasses of the image(front of CSE Department or Left of CSE Department)

  – As classifier, we used Support Vector Machine (SVM). It is implemented using vl-svmtrain from VL-FEAT library. this library trains a linear Support Vector Machine (SVM) from the data vectors X and the labels [9]

  As we employed multi-class classification; for c classes we learnt on c(c-1)/2 binary classes.

3.2. Testing

We again used ffmpeg library to segment the real-time video into various test images. AR Drone is connected to server via WiFi and streamed video is segmented using ffmpeg.
Using the vocabulary computed at the time of testing, we represent the test image as histogram. This histogram is fed to classifiers and output scores are obtained. For multiple test images as Explained in the model we take into account the probabilities. We calculated the probability for each building for each classifier 2. At testing stage we use these probability to calculate what is probability of given image to be of given building 1.
To increase the confidence measure, we look at second highest probability and if it is less than .9(first highest probability) then we increase the score for that building.
Given sequence of images we find out the one with maximum score.

4. EXPERIMENTAL RESULTS

We experiment on the videos of H. R. Kadim Department of Computer Science and Engineering (CSE Department), P. K. Kellkar Library and Department of Industrial and Management Engineering (IME). The following table shows the number of images in the training set obtained after segmentation via ffmpeg.

Training Video	Total Images	Front Images	Back Images
H.R. Kadim CSE Dept.	460	224	236
IME	213	176	137
Library	333	189	144
Total	1006		

Figure 1: Training Statistics

We apply our model explained in Section 2 and Section 3. Figure[2] and Figure [3] show the scores of training images captured by the classifier.

Figure 2: Scores-H.R. Kadim

We then applied our training model to video captured by AR Drone in the realtime. We tested by flying AR drone near CSE building and IME building. Performance measure was used to calculate result. Results are shown in Figure[4].

Performance Measure is Modelled as $P(A) = \frac{\text{Number of test images satisfying A}}{\text{Total Test Image}}$

Note that in our implementation we are considering only those images from video stream with high probability compared to second best and rejecting

Figure 3: Scores-IME

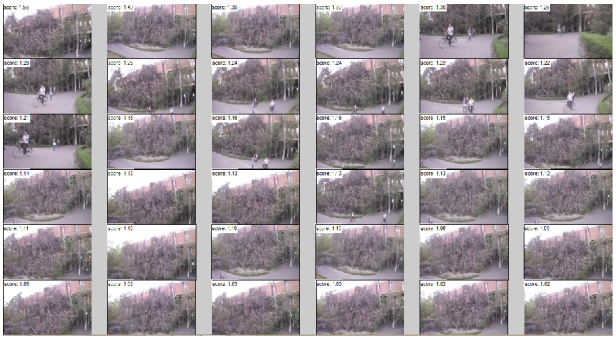

Figure 4: Scores-Library

the rest in the video. Thus here is high drop rate among the image and only those images are considered which distinguish with high confidence. Thus we are able to recognise successfully even when accuracy among the total images is 50-60%.

We were quite successfully able to classify which building A R drone is near. We can infer from the results that CSE Department is quite distinctive from other buildings and hence then we were able to categorise into the same with high confidence. High accuracy is also visible in case of IME.

Next we categorise between the front or the back. High accuracy was not found in case of distinguishing IME back due to high similarities between the front side of the building and the back side. Similarly there is big difference between the accuracy when recognising the CSE front and CSE back. CSE

Test Video	Building Identification Performance with front test video	Front Side Detection Performance	Building Identification Performance with back test video	Back Detection Performance Method 1
H.R. Kadim CSE Dept.	100	81.81	59.52	45.23
IME	88.37	93.02	84.21	31.55

Figure 5: Test Statistics

front is quite distinguished as compared to other buildings but CSE back has features similar to IME. We expect the problem to intensify as we take into account more buildings.

Another difficulty we faced is A R drone need open space to have safe flight. Thus we were not able to capture the buildings or side of a building with high density of trees. Also AR drone has best flight precision in joy-pad mode, but battery backup in the same is quite poor.

5. CONCLUSION and FUTURE WORK

The project demonstrated that Building Identification is feasible using an quadcopter equipped with an on-board camera. This is achieved by bag of Visual Words for representing each image in the form of a visual word and these words are then clustered using K-means algorithm. Finally One vs. One Binary SVM Classification algorithm is used for learning the classifier. The problem of inability to distinguish between the similar buildings is addressed in the Results section. The real issue, we believe, is with feature representation over image rather than video. We would like to explore the idea of classification algorithm that takes into account the representation of features over the video stream which can store the spatial information of features w.r.t. to others.

Motivation behind the project is to understand the terrain and identify its location in GPS denied environment. A similar project , SkyCall, is under process at MIT. Skycall is an autonomous flying quadcopter and a personal tour guide build at MIT Senseable City Lab that uses GPS to locate the user and flies to them. It flies at a walking speed and leads the user to its destination. The guide also provide interesting information about the sights they pass. We would like to build the similar guiding system for IIT Kanpur [10]
Finally we would like to extend our approach to more buildings at IIT Kanpur.

6. ACKNOWLEDGEMENT

We would like to thank Andea Vedaldi for sharing the code for image classification[8]. We would also like to thank Dr. Amitabha Mukerjee for his guidance and insight.

References

[1] T. Krajnk, V. Vonsek, D. Fier, J. Faigl, Ardrone as a platform for robotic research and education 161 (2011) 172–186.

[2] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: A survey, Foundations and Trends in Computer Graphics and Vision 3 (2008) 177–280.

[3] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision (2004) 91–110.

[4] D. N. N. P. C. S. R. W. A. Kanungo, Tapas; Mount, An efficient k-means clustering algorithm: analysis and implementation (2002).

[5] T. Moranduzzo, F. Melgani, A sift-svm method for detecting cars in uav images (2012) 6868–6871.

[6] L. P. D. Price, S. Knerr, G. Dreyfus, Pairwise nerual network classiers with probabilistic outputs 7 (1995) 11091116.

[7] FFmpeg, FFmpeg Software, Version 2.2.1, 2014.

[8] A. Vedaldi, A. Zisserman, Image classification practical, 2011, http://www.robots.ox.ac.uk/ vgg/share/practical-image-classification.htm, 2011.

[9] A. Vedaldi, B. Fulkerson, VLFeat: An open and portable library of computer vision algorithms, http://www.vlfeat.org/, 2008.

[10] M. S. C. Lab, Skycall, 2014.