Image Categorisation II

James Rogers, 100062949

1 Introduction

Image categorisation is a field of study that has seen much attention in the recent years. The ability to identify scenes in images has huge potential for useful applications that can provide many benefits.

This report will detail several techniques in computer vision to categorise images. Image descriptors such as SIFT [1] and HOG2X2 [2] features are used to describe the characteristics of a scene. These descriptors describe interesting aspects of an image, such as corners and T-junction. Feature detection algorithms such as a bag of words [8] and spatial pyramids [11] have also been investigated which are able to extract feature histograms from images. Lastly, classifiers such as the K-Nearest-Neighbour algorithm (K-NN) [13] and Support Vector Machine (SVM) [14] are used to classify the image feature histograms.

Each algorithm used will be tested with varying parameters that may or may not affect the image categorisation accuracy. The parameters that achieve the best accuracies are combined to produce an algorithm with the maximum potential.

2 Descriptors

In this section, two types of feature descriptors SIFT [1], and HOG2x2 [2] are presented. A brief description of each descriptor will be given, as well as a basic outline of how they will be implemented.

SIFT SIFT features [1], an abbreviated term for Scale Invariant Feature Transform, are used to describe interesting aspects of an image, such as corners and T-junctions. As already suggested by it's name, these features are invariant to image scaling, translation, and rotation, as well as robust to illumination and occlusion.

The SIFT features of an image are calculated mainly by four steps. Those are:

1. Scale-space extrema detection
2. Keypoint localisation
3. Orientation assignment
4. Keypoint matching

The scale-space extrema detection is performed by using a Difference of Gaussians (DoG) for different octaves of the image, which is an approximation of the costly Laplacian of Gaussian (LoG). Once this is found, the images are searched for local extrema over scale and space. These local extrema are potential keypoints. Keypoint localisation uses Taylor expansion is used to eliminate potential extrema if its intensity is below some threshold. Orientation assignment works by forming a histogram by quantising the orientations into 36 bins. These gradients are then normalised so that their length is of 1 to avoid the effects of illumination change. The keypoint descriptor is a vector for each keypoint that is distinctive and invariant to illuminant changes.

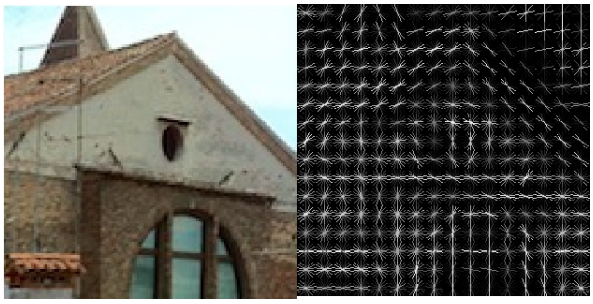

Figure 1: An example of HOG features [3]. The right image is a visual representation of the HOG features from the original image to the left.

The SIFT features are retrieved by using the VLFeat [3] library method, vl_dsift. This returns a 128 x n matrix where n is the number of features retrieved, and the 128 is a fixed dimension that describes the feature.

HOG2x2 Histogram of Gradient orientations (HOG) [2], is a method that divides an image into cells, typically 6-8 pixels wide. For each cell, a 1-dimensional histogram of directions is computed from all of the pixels within the cell, Figure 1. The result will be a matrix of size cx x cy x 31, where cx and cy are the cells along the rows and columns, and the fixed 31 dimension contains the HOG descriptors. The HOG descriptor is the same as step 3 of the SIFT detection process.

HOG2x2 [4], is a method that stacks 2x2 neighbouring HOG descriptors that spatially overlap to form a higher dimensional feature that is more descriptive. This ultimately leads to more detailed representations of image features.

The VLFeat [3] method, vl_hog is used to retrieve the HOG features which returns a 3-dimensional matrix of size cx x cy x 31. Next, the HOG2x2 feature descriptor matrix needs to be computed, which is done by iterating over every cell in the HOG matrix. The three neighbouring cells: east, south-east, and south are concatenated onto the current cells 3rd dimension. This leads to a matrix of size cx x cy x 124, where 124 is stacked neighbouring cells. Next, this matrix is reshaped so that it becomes 2-dimensional so that the cells are aligned along columns, and the rows occupy the feature descriptors, resulting in a 124 x $cxcy$

matrix.

3 Colour Descriptors

SIFT features can be extracted from an image using multiple types of colour channels. Varying colour channels may yield benefits such as intensity and illuminant invariance. In this section, these multiple colour variations of the SIFT descriptors will be highlighted and discussed.

RGB-SIFT RGB-SIFT computes SIFT descriptors for each RGB channel of the image. Instead of calculating the SIFT descriptors for a single grayscale channel, the descriptors are calculated independently for each RGB channel, these three channel features are then concatenated together to form a large feature matrix.

HSV-SIFT HSV consists of three channels, Hue, Saturation and Intensity. The V channel is invariant to light colour changes whereas as the H and S channels are not.

Hue-SIFT The hue-SIFT descriptor is a method proposed by [5] that concatenates a hue histogram on to the end of a SIFT histogram. The hue histogram is used to address the instability of the hue around the grey axis when compares to HSV-SIFT [6]. This is done by weighting each sample of the hue by its saturation.

Opponent-SIFT The Opponent-SIFT descriptor describes the SIFT features using three channels. The opponent colour is given using Equation 1, where R, G and B are the colour channels of the image, and $O1$, $O2$ and $O2$ are the opponent colour channels. However, the R, G and B colours must first be normalised by dividing each channel by its average [7]. The $O1$ and $O2$ colour channels retain colour information where as $O3$ is equal to the intensity information. This method is also said to be the most efficient in scene recognition [6].

$$\begin{pmatrix} O1 \\ O2 \\ O3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix} \tag{1}$$

W-SIFT Where the opponentSift descriptors $O1$ and $O2$ channels still contain some intensity information, W-SIFT looks to remove these intensities by using the W-Invariant given by Equation 2. This W-Invariant makes these channels become scale-invariant with respect to light intensity [6].

$$\begin{pmatrix} O1' \\ O2' \\ O3' \end{pmatrix} = \begin{pmatrix} \frac{O1}{O3} \\ \frac{O2}{O3} \\ O3 \end{pmatrix} \tag{2}$$

rg-SIFT rg-SIFT normalises the RGB channels of the colour image, Equation 3. Since $R + G + B = 1$, the B channel can be discarded. Where SIFT uses the channels derivatives to get the features, this becomes shift invariant. However, this techniques is still not invariant to varying illuminance, [6].

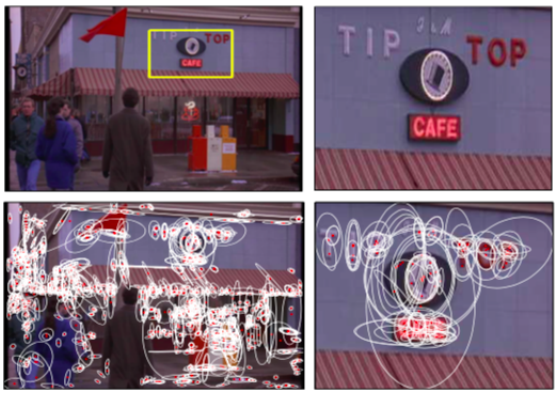

Figure 2: An image visualising all detected visual words.

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} \frac{R}{R+G+B} \\ \frac{G}{R+G+B} \\ \frac{B}{R+G+B} \end{pmatrix} \tag{3}$$

4 Feature Detection

This section will describe the feature detection methods used in this implementation. Each of the feature detection methods will use image descriptors that can either be obtained from SIFT of HOG2x2.

Bag of Words Local neighbourhoods of features can be represented by what is called a visual word [8], which describes the shape of an object, Figure 2. A list of visual words can be identified programmatically by vector quantising features retrieved from the images, using an unsupervised machine learning algorithm, k-means clustering [9]. This list of visual words can be thought of as a vocabulary, or a bag of words. A bag of words is used to identify image features and can provide accurate results, however, occlusion can greatly affect performance [8].

This algorithm has been implemented by extracting features using the VLFeat [3] library methods for each image in the training set. This matrix is then concatenated onto the end of a master feature matrix that holds all of the features found in the training data. Next, the master feature matrix is vector quantised using a k-means clustering method from the VLFeat library, vl_kmeans. This method returns a nxd matrix where n is the number of quantised visual words and d is the feature descriptors. An implementation of this algorithm can be found in Algorithm 1.

In order to build image feature histograms, the image features are first extracted from the image. Histogram intersection [10] is then performed between the vocabulary and the image features to compute a histogram of the closest words. Equation 4 demonstrates histogram intersection, where d is the number of bins, I and M are the histograms of the image features and the vocabulary, and HI is the resulting histogram.

Algorithm 1 BuildVocab(ImgPaths, numWords) **return** Vocab

```
for i ← 1 to numImgs do
    img = read(ImgPaths_i)
    imgFeats = getImageFeats(img)
    masterImgFeats = cat(masterImgFeats, imgFeats)
vocab = vl_kmeans(masterImgFeats, numWords)
return vocab
```

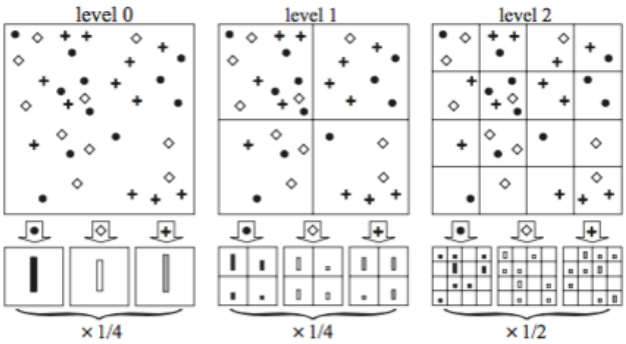

Figure 3: An example of constructing a three level pyramid [11]. There are three freature types in this figure, circles, diamonds, and crosses. For each level of resolution and each channel, we count the features that fall in each spatial bin.

$$HI = \sum_{i=1}^{d} \min(\mathbf{I}_i, \mathbf{M}_i) \tag{4}$$

Spatial Pyramid The spatial pyramid matching algorithm [11] is an adaptation of the pyramid kernel [12], that considers spatial information. Instead of computing a normal histogram intersection between the image features and the vocabulary, the pyramid matching function uses spatial information to compute a weighted histogram intersection. It does this by dividing the image up into equal sized cells, Figure 3. The number of cells is determined by the level specified, $2L^2$, where L is the level. Level 0 will result in no subdivision and behave as if it was a bag of words. A histogram intersection is then computed for each corresponding cell, but instead of incrementing the histogram by 1, it is incremented by a weighted value given by $\frac{1}{2^{L-l}}$ where l is the current level. An implementation of this algorithm can be found in Algorithm 2.

5 Classifiers

This section will detail the machine learning algorithms used to classify images given a set of feature histograms, as well as highlight how they were implemented with pseudo code examples. Those algorithms are K-Nearest-Neighbour (K-NN) [13] and Support Vector Machine (SVM) [14].

Algorithm 2 getSpatialPyramid(ImgPaths, vocab) **return** imageFeatures

```
for i ← 1 to numImgs do
    img = read(ImgPaths_i)
    ( imgFeats, locations ) = getImageFeats(img)
    D = vl_alldist2(imgFeats, vocab)
    I = min(D)
    for l ← 1 to levels do
        cells = getCells(img, l)
        for f ← 1 to numImgFeats do
            featLoc = locations_f
            cell = intersectCell(cells, featLoc)
            histIndex = cell_index + I_f
            hist_histIndex += 1/2^(L-l)
    masterImgFeats_i = hist
norm(masterImgFeats)
return masterImgFeats
```

K-Nearest Neighbour The Nearest-Neighbour (NN) [13], is a classifier that uses a supervised training model. It classifies data by finding the smallest distance between the test sample, and each of the training samples. The test sample will be classified as the classification of the closest training sample. To reduce the generalisation error, a modification of the algorithm called K-Nearest-Neighbour (K-NN) has been introduced which is what is used in this implementation. Instead of classifying data based on the closest training sample, a majority vote of the K nearest neighbours will be chosen to produce the classification. To eliminate an equal number of votes, K is usually an odd number, however, this only works with a binary class problem. If an even number of votes do arise on a multi-class problem, a random classification will be picked.

Support Vector Machine The Support Vector Machine (SVM) [14], is a classifier that uses a supervised learning model to classify test data. To build the classifier on a binary, linearly separable problem, a hyperplane must be defined to separate the data sets. To do so, find the shortest line segment between the convex hulls of each data set. The training patterns that define the closest points are known as "Support Vectors". The hyperplane will be perpendicular to the mid point of this line segment. This hyperplane maximises the margin between the datasets to lower the generalisation error [15].

In this implementation, the VLFeat [3] method, vl_svmtrain is used to produce a classification base on the training data. An SVM is trained for every image category, as SVMs are only able to classify a binary class problem. The resulting hyper planes given by this function is stored in a vector, ready to classify the test data. Once all of the SVMs have been trained, every image in the testing data is classified against each of the SVMs. The SVM classification with the most confidence will be chosen as the image

Algorithm 3 SVMClassify(trainImgPaths, trainLabels, testImgPaths) **return** predictedCats

```
for i ← 1 to categories do
    y = initClassifications(i)
    ( w, b ) = vl_svmtrain(trainImgFeats, y)
    SVMs_i = ( w, b )
for i ← 1 to numTestImgPaths do
    convSVM = -∞
    for s ← 1 to categories do
        c = SVMs_{s_w} * testImgFeats_i + SVMs_{s_b}
        if c > convSVM then
            convSVM = c
    predictedCats_i = convSVM
return predictedCats
```

Figure 4: The SUN image database[1] consists of 1500 training and 1500 test images. These images are split into 15 categories: Kitchen, Store, Bedroom, LivingRoom, House, Industrial, Stadium, Underwater, TallBuilding, Street, Highway, Field, Coast, Mountain, Forest.

category, this method is known as a one-vs-all SVM classifier. An implementation of this algorithm can be found in Algorithm 3 .

6 Testing

In this section, the data these algorithms are tested on will be described and their characteristics highlighted. Details of the types of tests carried out will also be given.

6.1 Data

The SUN database [1], Figure 4. was used to perform tests on these algorithms. The images in the database fall into one of fifteen categories, kitchen, store, bedroom, living room, house, industrial, stadium, underwater, tall building, street, highway, field, coast, mountain, forest. These categories are distributed equally across the database, meaning 200 images represent each category. This database is split equally into two subsets for training and testing data.

These images vary in characteristics as some categories represent a room indoors, and some represent urban and natural outdoor environments. Certain categories such as bedrooms and living rooms should be very difficult for these algorithms to distinguish as their features are likely to be very similar. Categories such as forests and under water should be easier to distinguish as their feature representations are expected to be considerably different.

6.2 Tests

The tests performed are used to identify weaknesses or strengths, in each of the feature descriptors and classifiers.

6.2.1 Feature Descriptors

The SIFT and HOG2x2 feature descriptors will be tested using the bag of words and the spatial pyramids. Multiple sizes of vocabulary's are also compared to determine an optimal number of words to describe all of the training images.

SIFT Tests will be carried out with various types of colour SIFTs. Judging by the results from [6], the W-SIFT colour descriptor should give results with the highest accuracy when compared to the other colour descriptors. Multiple variations in SIFT sampling rates will also be evaluated when retrieving the SIFT features.

HOG2x2 The HOG2x2 feature descriptor will see multiple variations in cell sizes to determine the optimal cell size to describe an image.

6.2.2 Feature Detection

The testing for the feature detection methods: bag of words and spatial pyramids will see variations in parameters that may alter the accuracy.

Bag of words The size of the vocabulary will be tested with the bag of words for the SIFT and HOG2X2 feature descriptors to identify accuracy changes.

Spatial pyramids Like the tests with a bag of words, the size of the vocabulary will be tested to identify changes in accuracies. The number of levels in which the spatial pyramid is divided will be tested ranging from 1 to 3. Level 0 will not be tested as this will effectively become the same as a bag of words. Like Lazebnik [11] found, it is expected that the spatial pyramid at level 2 will achieve the greatest accuracies.

6.2.3 Classifier

The classifiers will be tweaked and tested to determine the most optimal parameters for scene recognition.

K-NN The K-Nearest Neighbours will be altered in these tests ranging from 1, which will effectively become a Nearest Neighbour (NN) method, and 30.

[1] See `http://vision.princeton.edu/projects/2010/SUN/`

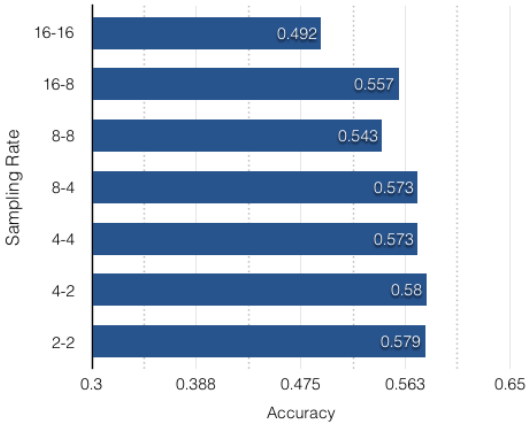

Figure 5: Results with varying SIFT sampling rate (vocab, image).

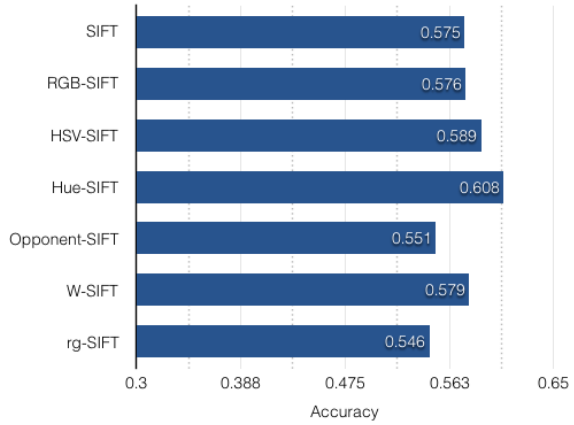

Figure 6: Results of the different colour-SIFT features.

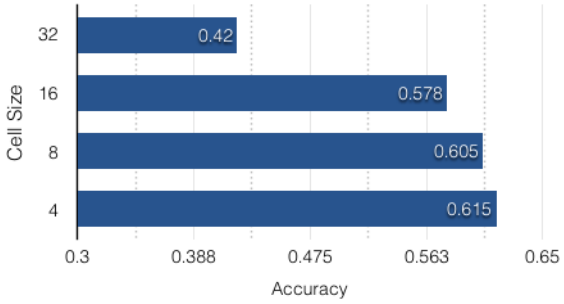

Figure 7: Results of a varying HOG2X2 cell size.

SVM Different variations of the lambda attribute in the SVM will be tested. The tests of the lambda attribute will vary from 10 to 0.00001.

7 Results

This section will highlight the results achieved using the tests specified in the previous section.

7.1 Feature Descriptors

7.1.1 SIFT

These tests will be performed while using the bag of words feature detection method, as well as an SVM with a lambda value set to 0.0001.

Sampling Rate As seen in Figure 5, the results show that the rate inwhich the SIFT features are sampled alter the accuracy. The sampling rate of 16 for the vocabulary and image feature histograms provided the worst accuracies with 0.492. The sampling rate for the vocabulary and image histograms ranging from 8 to 2 gave similar results; however, the best results were achieved with a configuration of 4 for the vocabulary and 2 for the image histograms with an accuracy of 58 %.

Colour SIFT The results shown in Figure 6 display the results achieved with each type of SIFT descriptor as mentioned in Section 3. The SIFT and RGB-SIFT features produce results that are very similar, this was to be expected as [6] also saw similar results. The HSV-SIFT saw improvements, this could be due to the hue and saturation channels being scale-invariant and shift-invariant. Hue-SIFT saw further improvements over the previous three, this is expected to be because the hue histogram addresses the instability of the gray axis. Opponent-SIFT saw a decline in accuracies. This was not to be expected as according to [6], this SIFT descriptor saw some of the highest accuracies. The decline in performance could be due to an implementation error. The W-SIFT improves upon Opponent-SIFT due to the removal of intensity information.

7.1.2 HOG2X2

Cell Size These results, Figure 7 show that the cell size in which the gradients are sampled affects the overall performance. A cell size of 32x32 gives the worst results with 0.42, this may be because and the high was 0.615 with a cell size of 4x4. It appears that the finer the cell size, the greater the accuracies. This is due to the increased amount of detail obtained from the image.

7.2 Feature Detection

7.2.1 Bag of Words

Vocabulary Size Figure 8 displays the results gathered with varying vocabulary sizes with SIFT and HOG2X2 features. Here you can see that the accuracies increase as the vocabulary size gets larger. The HOG2X2 feature descriptor sees higher accuracies than SIFT when using vocabulary sizes 50, 100 and 200. However, SIFT begins to outperform HOG2X2 for all vocabularies from 400 onwards. The reasons for this may be due to SIFT being able to capture more individual words from the images, where as HOG2X2 struggles to identify unique clusters.

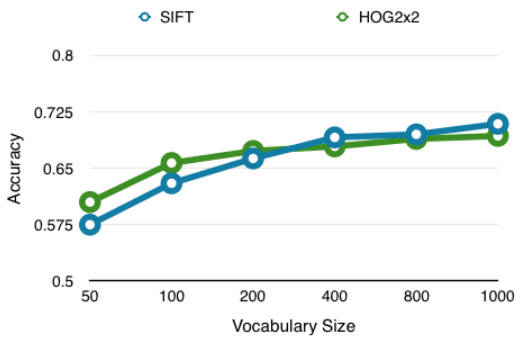

Figure 8: Results of the bag of words feature detection and varying vocabulary sizes using the SIFT and HOG2X2 features. The green plot indicates the SIFT results and blue represents HOG2X2.

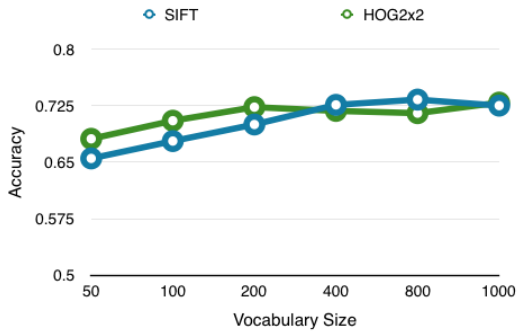

Figure 9: Results of the spatial pyramid and varying vocabulary sizes with the SIFT and HOG2X2 features. The green plot indicates SIFT and blue represents HOG2X2.

7.2.2 Spatial Pyramids

Vocabulary Size Similarly to the results obtained with varying vocabulary sizes in a bag of words, the spatial pyramid saw improvements in SIFT and HOG2X2 as the vocabulary grew, Figure . However, the SIFT descriptor saw improvements up until 800 words, until it began to decline at 1000 words. The HOG2X2 descriptor on the other hand began to drop off and get worse as the vocabulary passed 200 words, however, it saw a sudden increase at 1000 words.

Levels Figure shows that the levels in a spatial pyramid cause a drastic impact on the accuracies achieved. Level 0 provides the worst results as it is essentially a bag of words as the image is not spatially partitioned. Level 1 splits the image into four regions and improves the accuracies by approximately 10%. Level 2 sees the best results. This was to be expected as [11] also reported the same. Level 3 begins to decline, this could be due to over fitting the data. It also takes a lot longer to compute as each final image feature has 64 more histograms.

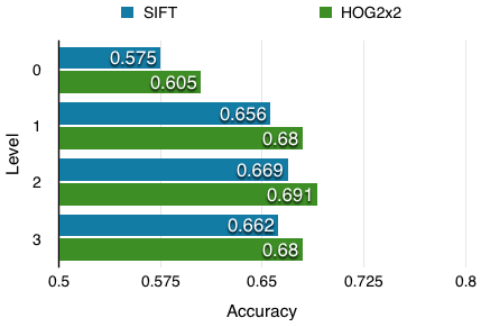

Figure 10: Results of varying spatial pyramid levels using the SIFT and HOG2X2 features. SIFT is represented by green and blue represents HOG2X2.

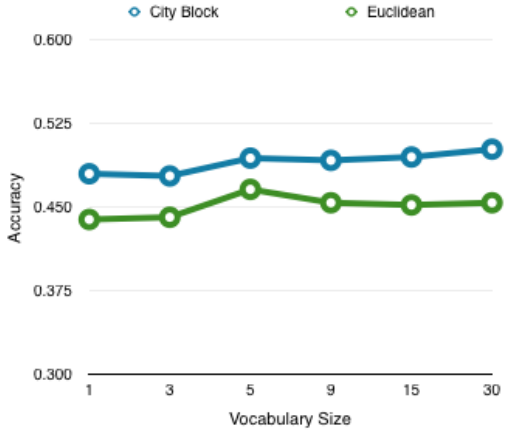

Figure 11: The results of the distance metric and number of neighbours in the K-NN classifier. These tests were performed with a SIFT bag of words and a vocabulary of size 50.

7.3 Classifier

7.3.1 K-NN

K Neighbours As seen in Figure 11 the number of nearest neighbours does cause an impact on the performance of the classifier. The accuracy of the classifier appears to increase slightly as the number of k nearest neighbours rise. However, it can be seen that accuracy of the classifier begins to drop as k becomes larger than 5 for the euclidean distance metric.

Distance Metric In Figure 11, it can be seen that the distance metric used when calculating the distances alters the performance of the classifier. The city block distance metric produces consistently higher results than the euclidean distance for all k nearest neighbours sampled.

7.3.2 SVM

Lambda The lambda parameter in the support vector machine has proved to provide drastic results. As seen in Fig-

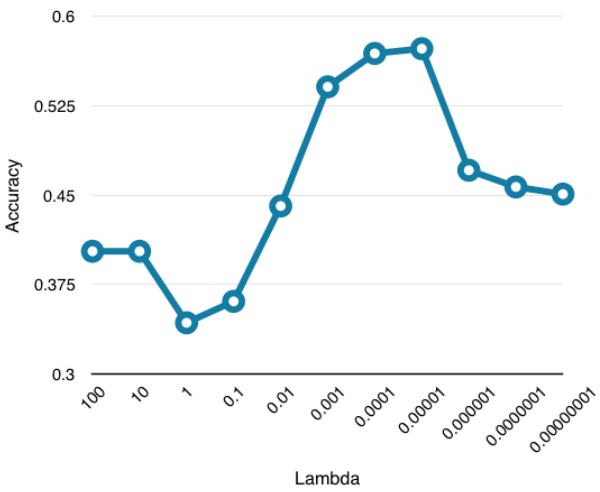

Figure 12: The results of a varying lambda value in the support vector machine.

ure 12, the optimal value of Lambda should range from 0.001 to 0.0001, as these saw the highest results. However, the performance quickly plummets as the value of lambda decreases below 0.0001.

7.4 Final Result

After testing the individual parameters of the feature descriptors, feature detection and classifiers, two final test have been performed of SIFT and HOG2X2 that combine each of the parameters with the best accuracies.

SIFT Firstly, the Hue-SIFT feature descriptor was used with a vocabulary size of 1000. The sampling size of the vocabulary was set to 4 and the image feature sampling was set to 2. The feature detector used was the spatial pyramid with a level of 2. Lastly, the support vector machine was chosen with lambda of 0.00001.

The results obtained from this configuration saw an accuracy of 0.761. The confusion matrix, Figure 13 shows the distribution of the results for each individual category. The category with the highest accuracy was the underwater category with a 92% success rate. The category with the lowest success rate was the living room at 57%. As observed in the confusion matrix, it appears that there is a block in the upper left corner that shows some uncertainty. The categories that fall into this block are: Kitchen, Store, Bedroom and Living room, which are all images of rooms inside a building. These scenes were expected to be harder to recognise as they are all very similar in characteristics and lighting.

HOG2X2 The first parameter of this tests was the HOG2X2 with a cell size of 2, and a vocabulary size of 1000. Like the SIFT test, the spatial pyramid was chosen with a level of 2. Again the support machine was chosen with a lambda of 0.00001.

The results achieved from this test saw an accuracy of

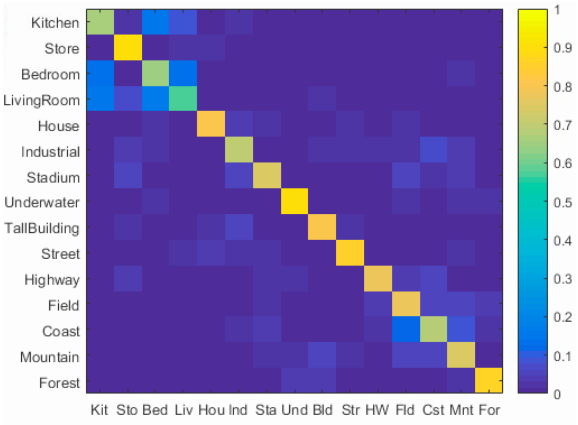

Figure 13: Confusion matrix of Hue-SIFT final test. Success rate: 0.757.

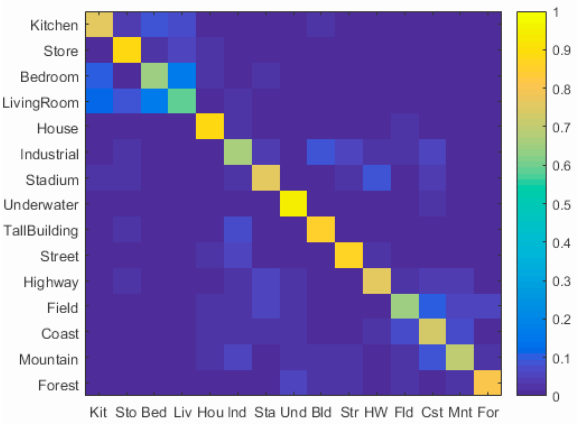

Figure 14: Confusion matrix of HOG2X2 final test. Success rate: 0.765.

0.765.The confusion matrix, Figure 14 shows the distribution of the results for each individual category. Similarly to the Hue-SIFT, the category with the highest accuracy was the underwater category, but with a 2% increase, 95% success rate. Again, the category with the lowest success rate was the living room at 57%. This confusion matrix also has similar results when distinguishing the inside rooms.

8 Conclusion

In this report, many algorithms have been demonstrated, those are two feature descriptors, SIFT and HOG2X2, two feature retrieval algorithms, a bag of words and spatial pyramids, and two classifiers, k-nearest-neighbour and support vector machine. Each algorithm has been thoroughly tested to find the optimal parameters for the maximum success rates. These optimal parameters were then combined to achieve the maximum possible success rate for SIFT and HOG2X2 features. The optimal SIFT feature algorithm saw an accuracy of 0.757 and the HOG2X2 descriptor saw 0.765. Both of these algorithms used a vocabulary size of

1000 and used the spatial pyramid with a level of 2.

References

[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, 1999, pp. 1150–1157 vol.2.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893.

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

[4] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 3485–3492.

[5] J. van de Weijer, T. Gevers, and A. D. Bagdanov, "Boosting color saliency in image feature detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 150–156, Jan 2006.

[6] K. van de Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, Sept 2010.

[7] J. Delon, Y. Gousseau *et al.*, "Combining color and geometry for local image matching," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 2667–2680.

[8] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 591–606, Apr. 2009.

[9] S. Lloyd, "Least squares quantization in pcm," *IEEE Trans. Inf. Theor.*, vol. 28, no. 2, pp. 129–137, Sep. 2006.

[10] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vision*, vol. 7, no. 1, pp. 11–32, Nov. 1991.

[11] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2169–2178.

[12] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *IN ICCV*, 2005, pp. 1458–1465.

[13] E. Fix and J. L. Hodges Jr, "Discriminatory analysis-nonparametric discrimination: consistency properties," DTIC Document, Tech. Rep., 1951.

[14] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297. [Online]. Available: http://dx.doi.org/10.1023/A:1022627411411

[15] J. Kim, B.-S. Kim, and S. Savarese, "Comparing image classification methods: K-nearest-neighbor and support-vector-machines," in *Proceedings of the 6th WSEAS International Conference on Computer Engineering and Applications, and Proceedings of the 2012 American Conference on Applied Mathematics*, ser. AMERICAN-MATH'12/CEA'12. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2012, pp. 133–138.