Convolutional Neural Networks and Scene Recognition

Towner Hale and Nathan Francis

1 Introduction

This report will be discussing the effect of a bag of SIFT features, spatial pyramids and feature extraction with convolutional neural networks (CNNs) with both a Euclidean distance classifier and support vector machine (SVM) for scene recognition. Part of the SUN database was used for the training and test data; 15 scenes from the dataset were used as the scenes to be recognised by the classifier. The bag of SIFTs calculates SIFT features for each image and then clusters them to create a vocabulary; this vocabulary is then used to create features for each image by creating a histograms of visual words. The spatial pyramids calculates bags of SIFT features on decreasing resolution sizes of an image and combines the bags of SIFT features for feature extraction. The CNN feature extraction method uses a pretrained CNN for feature extraction by extracting features from the CNN at one of the fully connected layers. Various parameters were changed for each feature extraction technique in order to see how susceptible they are to parameter changes and also to find the best performing set of parameters. Both the Euclidean distance classifier and SVM used the features provided by the feature extraction techniques to classify each test image.

Section 2 gives a brief description of SIFTs, Spatial Pyramids, and CNNs for feature extraction. Section 3 gives a brief description of how the Euclidean distance and SVM classifiers work and Section 4 evaluates the performance of the feature extraction techniques with the classifiers for scene recognition. Finally, Section 5 concludes the findings from the evaluation.

2 Feature Extraction Techniques

2.1 SIFTs

Scale Invariant Feature Transform (SIFT) was developed as a way of getting image features to be used for object recognition, and they have also been used for scene recognition (Xiao et al., 2010). Due to research leading to the belief that visual systems for primates are invariant to location, scale, and illumination, but are very sensitive to combinations of local shape, colour, and texture, a way of doing this in a computer visual system was sought after and SIFT was one of these approaches. In this approach images are transformed into a large collection of local image feature vectors. These feature vectors are invariant to many image parameters such as image scaling, translation, illumination, rotation and 3D projection. This approach was different to previous local feature techniques as previous approaches were negatively affected by changes to scale and illumination. The performance time of this approach is very fast as the SIFT feature vectors can be computed in less than one second. The local features are found by a stage filtering approach, in this approach the maxima or minima of a difference-of-Gaussian function is used to find key feature points and feature vectors are made which describe the regions around those points in a way that is similar to the responses of complex cells in the primary visual cortex in primates. This approach recognises objects by bounding the features to object interpretations, in a process called indexing, and then going through a best-fit solution, which is similar to part of the process of human object recognition. This further demonstrates how this approach is heavily modelled on a primates visual system. Due to this process objects are able to be recognised in cluttered backgrounds from any viewpoint and with varying levels of illumination (Lowe, 1999)

In this project 128 dimensional SIFT features were used to create a bag of quantised SIFTS. First, a vocabulary is created by calculating and putting the SIFT features for each image in the training set next to each other so that it creates a 128xn matrix where n is the total number of features extracted from the images. The features were then clustered with a kmeans algorithm, which finds a set number of centroids in the data. These clusters then become the vocabulary. To extract training features for each image in the training and test sets, SIFT features are extracted from the image and the distance between each SIFT feature and cluster in the vocabulary is calculated and a histogram is made, which is of size $clusterSize$x1 and sums to the amount of SIFT descriptors extracted, where each bin in the histogram indicates the number of times that cluster was the nearest to a SIFT feature. This histogram of visual words is used as the feature for that image. For colour images, the colour channels were normalized before calculating the SIFT features as was discussed in van de Sande et al. (2004), and then for every normalized channel, a SIFT descriptor is computed with the red, blue, and green SIFTs, and then combined to create a RGB SIFT. The SIFT features and kmeans algorithm are both implemented using the VLFeat toolbox[1].

[1]VLFeat Toolbox, 'http://www.vlfeat.org/index.html'

2.2 Spatial Pyramids

Spatial Pyramid Matching builds upon the idea of pyramid matching and works well for scene recognition. A spatial pyramid works by partitioning an image into increasingly fine sub-regions, locating local features in the sub regions and then creating histograms based off of the local features. In this approach features are no longer an orderless set, as they are in pyramid matching, as it uses global cues as indirect evidence about the presence of an object. The feature vectors of each image are quantised into a set number of discrete types, which can be seen in Figure 1, and make the assumption that only features of the same type can be matched to each other. Each histogram is then weighted inversely to the pyramid level. During feature extraction two kinds of features can be extracted; these two types of features are weak features which are similar to global SIFT descriptors and strong features. They also provide more discriminative power (Lazebnik et al., 2006). A support vector machine has been shown to work well with spatial pyramids for classification of imagesLazebnik et al. (2006). Spatial pyramids have been shown to perform better than state of the art and more sophisticated techniques on the Caltech-101 database and performed better than orderless bag of features methods in Lazebnik et al. (2006).

In this project, a three level spatial pyramid is implemented where the local features in each sub region of the image is calculated by creating a bag of quantised SIFTs as described in Section 2.1. Each histogram that the bag of quantised SIFTs creates is weighted and put in a vector, which contains the histograms for each level of the pyramid and becomes the vector of features for the image. This is done for both the training and test images and the feature vectors are sent to a classifier for classification.

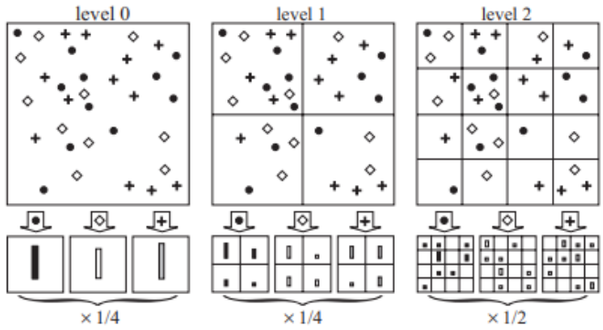

Figure 1: Example of a three-level pyramid that has three feature types, indicated by circles , diamonds and crosses. Each image at the top shows the image being subdivided to a smaller resolution. For each level of resolution and channel the number of features that fall into each spatial bin is counted

2.3 Convolutional Neural Network for Feature Extraction

Convolutional neural networks (CNNs), which are deep feed-forward artificial neural networks with convolutional filters, have been useful at solving the problem of object classification, however, they haven't been widely applied to the problem of scene recognition. Zhou et al. (2014) applied CNNs to the problem of scene recognition and achieved state of the art results and due to this success they were also used in this project. Convolutional neural networks have improved upon the performance of current object recognition approaches by using a larger database, learning more effective models, and employing different techniques to prevent overfitting. A CNN can make accurate assumptions on the nature of images because they have less parameters and consequently are easier to train. CNNs have been used for scene recognition in (Zhou et al., 2014), where the CNN learns deep features in each scene. CNNs need an input of fixed size images and each image will then have to change its resolution to the fixed size, which can be achieved by down-sampling or data augmentation among other techniques. The deep features learned from a CNN can be also be used for other visual recognition tasks or used with a classifier such as a support vector machine, which has been shown to exceed performance of state-of-the-art techniques, such as was done in Zhou et al. (2014) where it was evaluated on the SUN database and an accuracy of 54.32% was achieved. Feature extraction with a CNN has also been shown to outperform state-of-the-art techniques in Razavian et al. (2014).

Due to the success of feature extraction with CNNs, it was used in this project. A pretrained CNN was used for feature extraction and these features were then sent to an SVM for scene recognition. The pretrained CNN is called alexnet which accepts images of input size 227x227 and is a 25 layer CNN with 5 convolutional layers and 3 fully connected layers that was implemented in Krizhevsky et al. (2012) with large success. The convolutional layers apply a convolution filter to the image and detects features in a position independent manner. The architecture of alexnet can be seen in Figure 2. Alexnet was trained on 1.2 million images from the ImageNet[2] dataset. The ImageNet dataset contains objects and therefore the alexnet CNN was trained for object recognition, which is different from the problem that is trying to be solved in this project, which is scene recognition. However, as the two tasks aren't

[2]ImageNet, http://www.image-net.org/

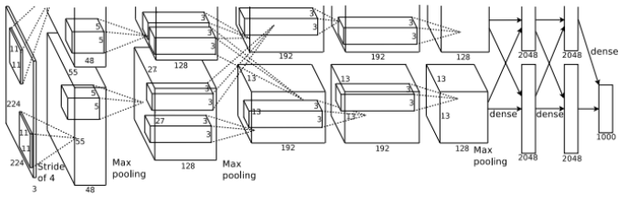

Figure 2: Results of the cluster size experiments for spatial pyramids

too dissimilar, the CNN should be able to extract features that are important for scene recognition as it may be able to find objects that are specific to each scene, and it would be interesting to see how the CNN would perform for feature extraction. For this project, the CNN was sent RGB images from the SUN database, which it is assumed that it would be better than grey images, as a colour can be important in differentiating two scenes. For example, for the scene called Underwater, a lot of blue colours are expected. Features can be extracted from any of the fully connected and convolutional layers of alexnet as they all extract features that could be used for classification. The CNN extracts 4096 features for each image and these features are then sent to an SVM for classification. An SVM was used for classification because it has been shown to have good performance for image classification when the features used have been extracted from a CNN (Zhou et al., 2014).

3 Classifiers

3.1 K-Nearest Neighbour Classifier

A Euclidean distance classifier which is also known as a k-nearest neighbour classifier (KNN) can be used for image and scene recognition. The KNN works by using the features of a test image, which are in this case from either the bag of SIFTs or spatial pyramids, and then computing the distance between the test image and every train image. The class of the k training examples closest to the test point determine the test images class through a most vote method. For small k values, training samples similar to the test sample will command its classification, while large k values will have training samples less like the test sample influence its classification. Therefore, choosing a good value for k involves an odd value that isn't too small (over-fitting) and isn't too large (under-fitting). In this project the city block distance metric was was used to calculate the distance between the train and test images, and this was due to city block outperforming other distance metrics in experiments with the SUN database that were conducted in a previous project.

3.2 Support Vector Machine

The support vector machine classifier, which is also known as SVM, is a binary linear classifier. It is has shown to perform well for image recognition and is used in this project for scene recognition. SVMs work by marking the training data into two different categories, and constructing a dimensional vector that will separate the greatest distance, known as the margin, between the points of the categories. The greater the margin, the greater the chance of the new data being classified correctly. This is known as the maximum-margin hyperplane, and it divides and classifies data while minimizing the risk of overfitting. The further from the hyperplane the data point is, the greater chance that data has been correctly classified. If the data sent to an SVM is not linearly separable then it uses a kernel to transform the data to a higher dimension where it can be linearly separated.

Figure 3: Example of a support vector machine classifier and its dimensional vector that separates the points of the categories

The benefits of SVM's include performing at a high level when there is a defined separation between the categories and the margin, as well as being more memory efficient than alternative classifiers.

However, one drawback is that the training time required is higher than other classifier alternatives, which might not making it the best option when using real world applications under time constraints.

In this project 15 SVMs are used for classification where each SVM is trained in the format category vs all; for example the first SVM is trained on kitchen vs all other categories. The confidence of each SVM is returned and the SVM with the strongest confidence is chosen as the category for the test image. The SVMs are implemented using the VLFeat toolbox.

4 Experiments

The performance of SIFT features, spatial pyramids and a CNN, was evaluated with KNN and SVM classifiers. The data was partitioned into two sets, a training set and test set, with both sets containing 1500 images. Parameters were altered in the experiments for both the feature extraction techniques and classifiers to see how it affected their performance, and also in attempt to find the parameters that gave optimal performance. The SUN database was used for the train and test data, and it contains images of the following 15 scenes: Kitchen, Store, Bedroom, Living Room, House, Industrial, Stadium, Underwater, Tall Building, Street, Highway, Field, Coast, Mountain, and Forest. The classifier's goal was to correctly classify a test image as one of the scenes while using the features provided by the feature extraction techniques. A set of benchmark parameters were given to the feature extraction techniques to compare the effect of parameter changes.

4.1 SIFT Experiments

For the get bag of sifts feature extraction technique, experiments were carried out to determine the effect of changing the SIFT features step size, the number of clusters, colour and grey images, and the KNN and SVM classifiers upon performance. Experiments were carried out with the following benchmark parameters: a SVM classifier, greyscale images, a step size of five, and a cluster size of 50. The SVM classifier was chosen because it was expected to have very good performance, while greyscale images allowed the training data to be computed faster. A step size of five and cluster size of 50 was chosen because it was determined to have good performance while not taking a large amount of time to run.

4.1.1 Step sizes

Changing the step sizes changes the number of pixels used to make each SIFT descriptor. For example, if the step size is five, it extracts a SIFT descriptor each five pixels. It is believed that if the step size is too small, it will take longer to extract each descriptor, and said descriptor will be too specific to the image which could lead to the training data overfitting the classifier. If the step size is too large, then the descriptor will capture too much of the whole image and consequently not be specific enough. Therefore, a step size of 5 was picked as it shouldn't take too long to compute.

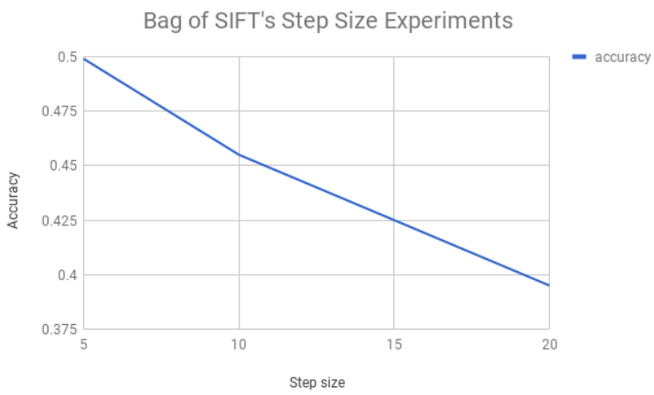

Figure 4: Results of the step size used for the SIFT features in the bag of SIFT's

When comparing step sizes, a step size of 5, the smallest out of the steps that was tested, received the highest accuracy at 49.9%, which contradicts the belief that a small step size would make the descriptor too specific. The step sizes 10, 15, and 20 received accuracies at 45.5%, 42.5%, and 39.5% respectively, demonstrating a clear correlation between an increase in step size and decrease in performance. One reason why this might be the case is that the SIFT features don't capture enough information specific to the image, and subsequently decrease the accuracy.

4.1.2 Clusters

The number of clusters was altered to determine its effect upon the performance of the bag of SIFTs. A low number of centroids as a result of low cluster size can negate the model's accuracy because there are not enough clusters to describe the data well. A very large cluster size can lead to over-fitting, and

subsequently decrease classifier performance because there are too many clusters described key points in the data. It is then believed that a cluster size that is not too small or large will lead to the best performance by the classifier.

Figure 5: Results of the cluster size used for the SIFT features in the bag of SIFT's

After performing experiments altering cluster size, the largest cluster at 300 had the highest accuracy at 61.8%, followed by 150 clusters at 58.6%. Clusters 100, 75, 50, and 25 received accuracies of 56.1%, 52.1%, 49.9%, and 47.8% respectively. The results demonstrate that accuracy improves as cluster size increases, and that a large cluster size 300 does not lead to overfitting the data. Experiments with larger clusters sizes couldn't be conducted due to it being very computationally expensive, however if they could, an overfitting of the data may have been observed.

4.1.3 KNN Classifier

The KNN classifier used a set of k values ranging from one to 15 to test which one gave the highest degree of accuracy on the data set. Low k values were not expected to perform well because of the chance of overfitting, as well as training data dominating the classification unless training and test data are similar. Large k values give better performance when training data is less like test data, which otherwise could lead to underfitting. The experiments were run on both grey and coloured images s to determine there effect on performance. As colour images give more information about the scene it is assumed that they would lead to higher accuracy than its grey alternative. Colour information can also be important for scenes; if there is a scene of the sea, the image is likely to have significant amounts of blue, which can aid a classifier by telling it this information.

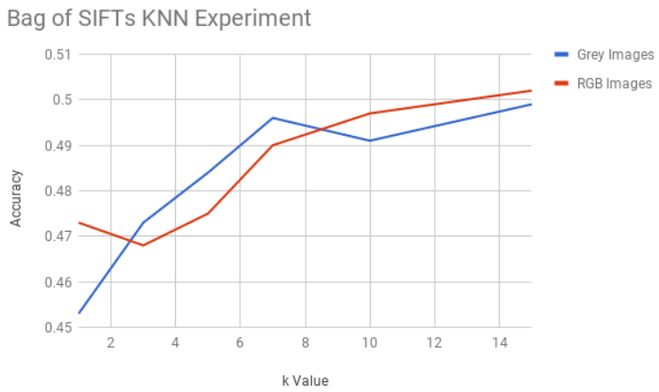

Figure 6: Results of the altered nearest neighbours used for coloured and grey SIFT features in the bag of SIFT's

The results of the nearest neighbour classifier on both coloured and grey images demonstrate that there is no clear decision on which has better performance. The grey KNN's k values of 1, 3, 5, 7, 10, and 15 gave accuracies of 45.3%, 47.3%, 48.4%, 49.6%, 49.1%, and 49.9% respectively. For the most part, an increase in nearest neighbours led to an increase in accuracy, with the largest nearest neighbour outperforming the others. When using colour, k values of 10 and 15 gave the highest accuracies at 49.7% and 50.2% respectively. The k values 1, 3, 5, 7, accuracies of 47.3%, 46.8%, 47.5%, 49.0%, respectively. While the colour k value of 15 gave the highest accuracy out of both colour and grey images, there was not a large difference in accuracies between the grey and colour k values, therefore the hypothesis that colour would lead to a higher accuracy is not substantiated. However, the colour images did give the highest overall accuracy. The results for the KNN experiments can be seen in Figure 6.

4.1.4 SVM Classifier

When comparing the SVM and KNN classifiers, it was assumed that the SVM, used in modern computer vision applications, would out perform the more simplistic KNN classifier. The SVM must optimize it's lambda, which influences the number of miss-classifications. A high lambda will minimize the amount of miss-classifications by maximizing the margin between classes, however, there is a trade off between a small lambda, which can lead to a high possibility of underfitting, and a large lambda, which can lead to overfitting. Due to this trade off, a lambda value between 0.001 and 0.0001 is expected to have good performance. The SVM classifier was tested on grey images and RGB images in order to see if colour information aids classification of scenes. It was hypothesised that RGB images would improve the classifiers performance because colour information is useful for some scenes, for example if a scene usually has lots of green in it, it will be important for the classifier to know this.

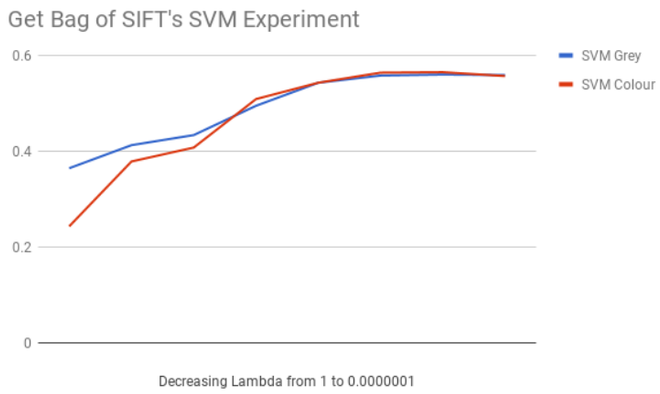

Figure 7: Results of colour and grey images with different lambda values

The results of the experiments showed that colour images with a lambda value of 0.000001 gave the highest accuracy at 56.5%, however, it was only marginally better than grey images, which received a highest accuracy at 56.0%. As can be seen in 12, the grey images received higher accuracies than the colour images for higher values of lambda, but as the lambda value decreased, both grey and colour images accuracies increased. The colour images also increased at a faster rate with higher accuracies than the grey images. For example, grey images with a lambda value of 1 achieved 36.5% accuracy, which is 12.1% more than the RGB images achieved. Lambda values 0.00001 and 0.000001 achieved 55.8% and 56% accuracy respectively, while the RGB images achieved accuracies of 56.4% and 56.5%. This experiment demonstrated that colour images have marginally better performance than grey images when the SVM has been optimised.

4.1.5 Optimal Parameters

The best performing parameters were put together in order to see if performance would be better than the other experiments. It was expected that these parameters would have the best performance, however, the lambda value of the SVM classifier may have had to be changed because it is highly dependent upon the data it is trained and tested on. The parameters chosen for the optimal bag of SIFTs was 300 clusters, RGB images and an SVM classifier with a lambda of 0.000001. This gave an accuracy of 58.9%, which was lower than the 59.5% received for the grey images. The best optimal lambda for grey images was 0.001 at an accuracy of 61.8%, however, after optimizing the lambda for RGB images to 0.0001, an accuracy of 67.1% was received, which was the highest accuracy out of these parameters. Therefore, RGB images with an optimized lambda gives the best performance, which demonstrates that RGB images gives a better image description and classification accuracy when compared to grey images.

Referring to the confusion matrix in Figure 8, yellow squares represent images that have been classified very often, while the darker the square the less the image is classified with that particular image. It is clear that underwater, store, street, and forest images were classified well because of their bright yellow colour, while kitchen, bedroom, and living room images were occasionally classified as each other. Coast was also occasionally classified as field, while mountain was mistaken for coast.

4.2 Spatial Pyramids Experiments

Experiments were conducted with Spatial Pyramids to see how parameter changes affected them and also to see which classifier resulted in the best performance. Experiments were conducted to see the effect of the SIFT features step size, the cluster size when clustering the SIFTs, creating spatial pyramids with both colour and grey images and the performance of a KNN classifier and SVM classifier with spatial pyramids. The step size and cluster size experiments were computed with grey images as grey images are faster to compute and will give a good benchmark on which values give good performance and which

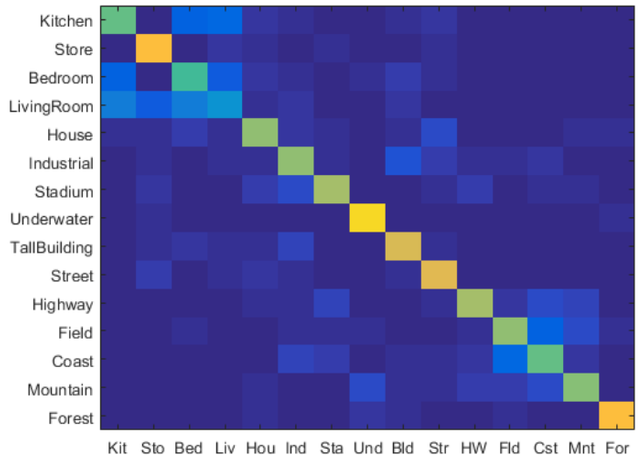

Figure 8: The confusion matrix generated with the parameters: 300 cluster size, RGB, and a 0.0001 lambda

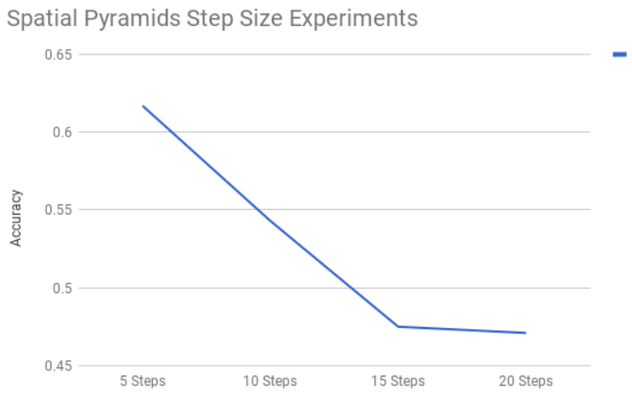

Figure 9: Results of the step size used for the SIFT features in the spatial pyramids

values can decrease performance. The classifier experiments however used both colour and grey images in order to see which classifier performed best and to see if colour aids the classifiers performance.

Each experiment used benchmark parameters and changed only the parameter that the experiment was for. This was done so that each experiment and parameter could be compared fairly. The benchmark parameters for the spatial pyramids were a step size of 5, 50 clusters, grey images and a SVM classifier with a lambda value of 0.001. The SVM classifier was used as it is an approach that is used in many modern computer vision applications, 50 clusters were used because its run time will not be too long while still having good performance.

4.2.1 Step sizes

The step size changes the number of pixels a SIFT descriptor is extracted for. For spatial pyramids the step sizes control the number of pixels used to extract SIFT features in each division of the image for each pyramid level. It is believed that if the step size is too low then then SIFT features may be too specific to the training data, however if the step size is too big then the SIFT features will be too general and won't capture information specific to each scene. For this reason it was believed that between 5 and 10 step sizes would be the ideal number of steps. The smaller the number of steps the slower the SIFT features take to compute, and this is because it creates less SIFT features. For applications that use SIFT features they will have to balance the time it takes to compute the SIFT features with the performance of the step size.

The hypothesis proved to be true as increasing the step size led to a decrease in performance of the spatial pyramids.The results of the step size experiments can be seen in Figure 9. A step size of five achieved an accuracy of 61.7% and this took a sharp drop to 54.3% with a step size of 10. The accuracy continued to drop sharply until it reach 15 steps where the accuracy only dropped by 0.4%. This proves that increasing the step size, which leads to less SIFT features being created means there is less information for the classifier to use to correctly classify each scene, which leads to a decrease in accuracy. If step sizes less than 5 were tested than the accuracy may have started to decrease as was hypothesised because the features may have been to specific to the training data.

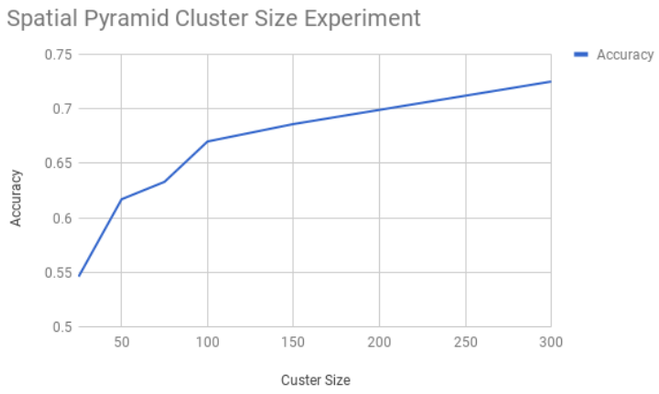

Figure 10: Results of the cluster size experiments for spatial pyramids

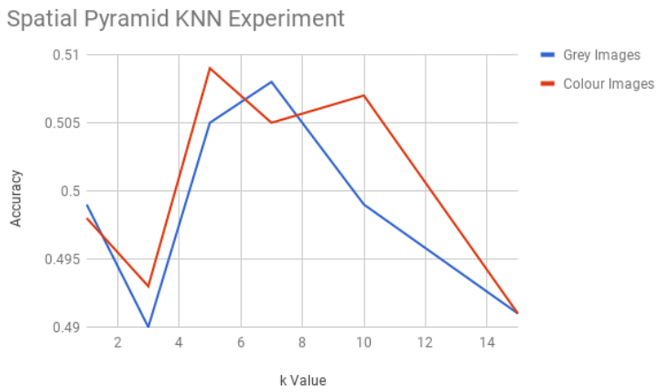

Figure 11: Results of the number of neighbours used in the KNN classifier for spatial pyramid features with RGB and grey images

4.2.2 Clusters

The number of clusters created for the SIFT features at each level of the pyramid was changed to see how they would effect performance. Changing the number of clusters changes the number of centroids created in the feature space and therefore having a low number centroids will result in the feature space not being modelled by the centroids well, which results in the images not being described by the spatial pyramid histograms well. On the other hand if the number of clusters used is too high there will be too many centroids and it will overfit the training data and decrease performance. Due to this it was believed performance would increase until a certain number of clusters and then performance would start to decrease.

As was hypothesised, increasing the cluster size increased performance of the spatial pyramids as accuracy increased from 54.6% with 25 clusters to 72.5% with 300 clusters as can be seen in Figure 10. However the accuracy didn't decrease when the cluster size got too big, but this may be because big enough cluster sizes were not tested due to them taking an extremely long time to run. Due to the problem of bigger cluster sizes greatly increasing the runtime a decision will have to be made between the cluster size and the step size used when creating a spatial pyramid as increasing the step size makes the runtime shorter but decreases accuracy.

4.2.3 KNN Classifier

A range of k values for the KNN classifier was tested to see which K value has the best performance. It was expected that low k values wouldn't perform well because due to over fitting the training data. However if the value of k is too large the it can lead to under-fitting the data and decrease the classifiers performance. The KNN classifier was also tested with both RGB and grey images to see if there is a big difference in performance between RGB and grey images.

The results of the KNN classifier can be seen in Figure 11 and they show the colour images having the optimal performance with the KNN classifier when it used 5 nearest neighbours and achieved an accuracy of 50.9% which is marginally better than the grey images best accuracy of 50.8% with 7 neighbours. The colour images performed better than the grey images for every value of k apart from values 1 and 7 and this shows that spatial pyramids that use colour information give better performance than grey images with a KNN classifier. The results also prove the hypothesis true that a low value of k will underfit the traiing data and not give good results but as k increases the accuracy will increase until the value of k is too big and overfits the data and the accuracy starts to decrease.

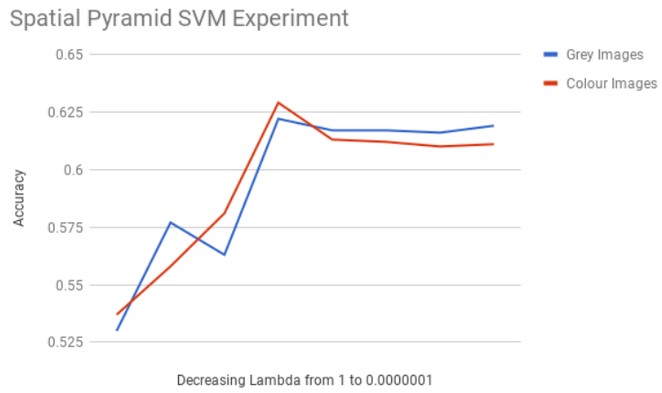

Figure 12: Results of the SVM classifier with decreasing lambda values for spatial pyramid features with RGB and grey images

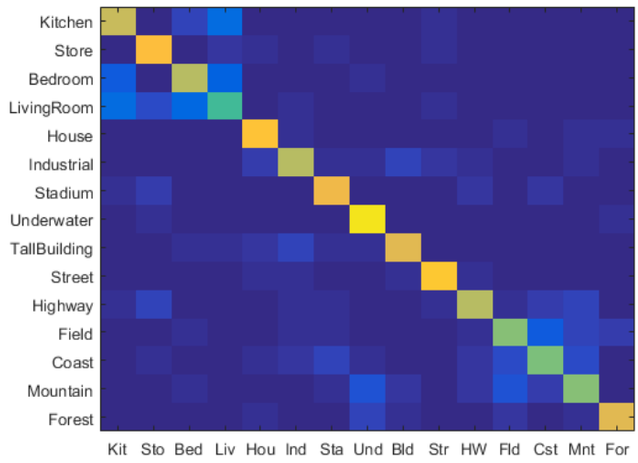

Figure 13: Confusion matrix of the best performing parameters with a spatial pyramid for feature extraction and SVM for classification

4.2.4 SVM Classifier

The SVM classifier was expected to have higher accuracy than the KNN as it is a more sophisticated approach than the KNN classifier and is used in many modern computer vision applications, including state of the art applications. For a SVM to perform well however, its lambda parameter, which sets its regularisation, need to be optimised. The higher the value of lambda the less misclassification's are allowed and the smaller the lambda the bigger the margin is and more misclassification's are allowed. The optimum value of lambda is thought to be a lambda value that is in the middle as it would have a good tradeoff between bias and variance. Due to this a lambda between 0.001 and 0.0001 was thought to give the best performance.

The results of the SVM classifier showed that a lambda value of 0.001 gave the best performance for both spatial pyramids that used grey images and RGB images. The spatial pyramids that used the RGB information gave the highest accuracy as it achieved 62.9% compared to the grey images 62.2%. However for half of the other values of lambda that were tested, the grey images achieved marginally higher accuracy than the colour images. For example for lambda values between 0.0001 to 0.000001 the grey images achieved an accuracy of 61.7%, 61.7% and 61.6% respectively while the spatial pyramids with colour information achieved an accuracy of 61.3%, 61.2% and 61% respectively. This shows that spatial pyramids with colour information only have better performance than grey images for some values of lambda but also that colour images give the best accuracy overall.

4.2.5 Optimal Parameters

The best performing parameters for each experiment for the spatial pyramids were used together in order to see how they would perform together. It was believed that using all of the best parameters together would give the best performance because these were seen as the optimal experiments in each previous experiment. The lambda value for the SVM classifier however will most likely have to be changed with the optimal parameters because the best lambda value is highly dependent upon the data it is trained and tested on. The parameters chosen for the optimal spatial pyramid were 300 clusters, RGB images and an SVM classifier with a lambda of 0.001.

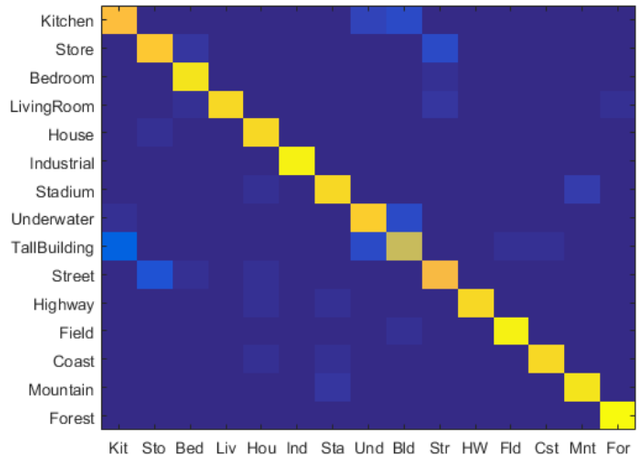

Figure 14: Confusion matrix of a convolutional network used for feature extraction with SVM for classification

As was expected, these parameters gave the best results as an accuracy of 73.4% was achieved using a spatial pyramid with the optimal parameters. The lambda was changed to test the hypothesis that the optimal lambda is dependent on the features, and to see if better performance could be achieved. The hypothesis was proven true as a lambda value of 0.0001 achieved 74.1% accuracy. To ensure RGB images performed better than grey images, the optimal parameters were used with grey images and the best lambda was searched for, which resulted in a lambda of 0.001 and 72.5% accuracy, almost 2% lower than that of the RGB images. This shows that the optimal parameters improve performance of spatial pyramids and classifier performance improves when using the images colour information.

The confusion matrix of the best performing parameters demonstrate that the spatial pyramids achieved good accuracy, although there were a lot of misclassification's of a living room as it was often misclassified as a kitchen and bedroom. One explanation could be that there are many colours that could be similar in these scenes, as opposed to scene called field which would predominantly be green and it was never misclassified as a living room. The Kitchen, Bedroom, and Living Room scenes substantiate this explanation because they were often misclassified as each other. It is surprising that bedroom was misclassified because the vast majority of bedrooms contain a bed which should be a strong discriminant. Fields, Mountains, Coasts and Mountains were also misclassified as each other quite often and this is mostly likely because there are many similar objects in those scenes, as well as the colour green, which shows the downside to using colour with the spatial pyramids.

4.3 CNN Experiments

The CNN experiments were expected to outperform SVM and KNN because CNNs are state-of-the-art models and learn deep features of each object. One potential drawback is that the CNN model that was used is a pretrained network called alexnet, which was trained on object recognition instead of scene recognition, which could mean the features extracted might not be useful for scene recognition.

When conducting the CNN experiments, a SVM classifier was used on the extracted features provided by the CNN because it is a state of the art classifier that has had better performance than the KNN and has been used for scene recognition with features extracted from a CNN in the past. The lambda parameters in the SVM were changed to see their effect on accuracy, while different layers from the network were used to extract higher level and lower level features. It was expected that the higher level layers would give the best performance in comparison to lower level features, and a lambda around the value of 0.001 would give the highest accuracy because it gave the best results when in the SIFT and spatial pyramid experiments. The 6th and 7th fully connected layers were used in the experiments and 5th and 3rd convolutional layers were also experimented with. It was expected that the fully connected layers would have the best performance as fully connected layers are good for classification.

After performing the experiments, the fully connected 7th layer received an accuracy of 90.3% with a lambda of 0.0001, which was higher than its 89.3%. accuracy when it had a lambda of 0.001. The 6th layer had the same accuracy at 90.3% for the 0.0001 lamda, which was lower than its accuracy with a 0.001 lambda at 89.8%. Since there was not a large difference between both layers, it can be concluded that since the layers are deep within the network they would have similar features, as opposed to the layers in the beginning of the network. The 5th convolutional layer received a lower accuracy when compared to the fully connected layers at 86.1%, and the 3rd layer received the lowest accuracy of all at 79.6%, which supports the hypothesis that convolutional layers would perform worse than fully connected layers. It is also clear that there is a significant difference between layers in terms of performance; it can be assumed from this experiment that latter layers outperform the earlier layers. Referring to the confusion matrix in Figure 14, it is interesting to note that there was not a lot of

misclassification between store, kitchen, and bedroom, which was a common misclassification when using the SVM classifier without the neural network, which can be seen in the confusion matrix in Figure 8. There was, however, misclassification between underwater and building, as well as building and kitchen, and store and street, which is surprising because there are not many similarities between the images.

One potential improvement upon the CNN is using a method called transfer learning, which removes the last few layers from a CNN, such as the last fully-connected layers, from the pre-trained network and replaces them with randomly initialized layers. It will then train the network on the training data and only learn the parameters for the new layers, and after learning these parameters for the new layers, it will fine tune the network by learning parameters for the whole network. Since it only retrains the last layers, it is quicker than training on the whole network and it can be used on object and scene recognition without training a whole CNN which is time consuming.

Data augmentation of the features is another potential improvement upon the CNN, where the features are modified to make them less similar to the training set. Some methods of data augmentation include: duplicating each image while shifting, rotating, distorting, or shading the image. Another option would be modifying the neural net to take in two random images from training, and produce a single image from them. This augmented image is then put into a second network with the training data, and subsequent training loss is is backpropagated to train the network's layers. Consequently, the model will recognize the data set's best augmentation.

5 Conclusion

In this paper, the performance of bag of SIFTS, spatial pyramids, and CNNs were evaluated using a SVM and KNN classifier to determine which technique had the highest accuracy for scene recognition. The CNN paired with the SVM classifier and using a lambda of 0.001, had the best performance with an accuracy of 90.3%, which was higher than the 74.1% received using spatial pyramids with optimal parameters. It was determined that the SVM classifier outperforms a KNN classifier, and that increased the cluster size for both bag of SIFTS and spatial pyramids improves accuracy. Using colour information for scene recognition was also seen as useful for scene recognition as for each feature extraction method it increased performance. Both members contributed 50%.

References

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2169–2178. IEEE.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee.

Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE.

van de Sande, K. E., Gevers, T., and Snoek, C. G. (2004). Evaluation of color descriptors for object and scene recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, USA (June 2008)*.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014). Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495.