Coursework on Randomised Decision Forests

Alexis Othonos Koral Hassan
Imperial College London
{ao4818, kbh15}@ic.ac.uk

Abstract

We pursue a bag-of-visual-words approach to the multi-class image categorisation problem. We use dense SIFT to obtain descriptors. We experiment with K-means clustering and Random Forests for codebook creation. We again use Random Forests for classification.

1. Dataset

We will be attempting image categorisation using a subset of the Caltech 101 dataset. [1] Our subset contains 10 labelled classes. Each class contains between 34 – 239 images. Please see Appendix A for a full list of class sizes and an exemplar image from each class.

From each class, we randomly select $N_{train} = 15$ images for our training set and $N_{test} = 15$ other images for our testing set.

2. Feature Extraction

We apply a very popular feature extraction algorithm called the *Scale Invariant Feature Transform* (SIFT). [3] It analyses a subsection of an image (referred to as a $patch$) in terms of Difference of Gaussians (DoG). [**?**] Specifically, Gaussian derivatives are computed at 8 orientation planes over a 4x4 grid of spatial locations, giving a $d = 128$ dimensional *descriptor vector* for that patch.

The VL_PHOW algorithm we utilised [**?**] computes the grey-scale variant of the descriptor.

We opted not to use an interest point detector for filtering down to a sparse set of patches. Instead, we used a $dense$ grid where patches are collected on each segment of the grid.

SIFT can be performed on different DoG scales, which we have illustrated on Figure 1.

3. Codebook

Due to memory concerns, we randomly select $N' = 100000$ descriptors and only use this subset for creating our codebook.

The codewords in our codebook are created by clustering the descriptors found in the multi-scale dense SIFT. In this case, the K-means clustering algorithm is used to find K different group centres such that the overall distance between clusters is maximised while the distance of descriptors to their group centre is minimised, according to the equation under optimisation:

$$J = \sum_{n=1}^{N'} \sum_{i=1}^{K} r_{ni} ||d_n - \mu_i||^2 \quad (1)$$

where d_n are the data, μ_i are the cluster centres and r_{ni} is a boolean membership indicator.

K represents our visual vocabulary size. If it is too low, we will not have have a codebook that is representative of all patches. However if it is too high, we will see large quantization artifacts and overfitting.

We can see in Figure 2 that our testing accuracy initially increases but then starts to plateu. Values of K that are much higher than 500 will start to see a decline, so this will be avoided.

4. K-means Classifier

After the codebook is created, the images are encoded into a membership histogram of the K codewords by performing a euclidian Nearest Neighbour search for each image descriptor. Figure 3 shows an image and its corresponding histogram after the quantisation process.

5. RF Classifier

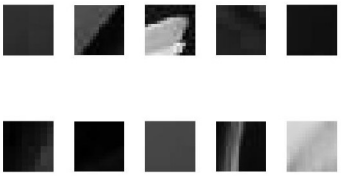
5.1. Random Forest

The Randomised Forest (RF) is implemented with a modified version of external toolbox [2]. The trees are assumed to be balanced.

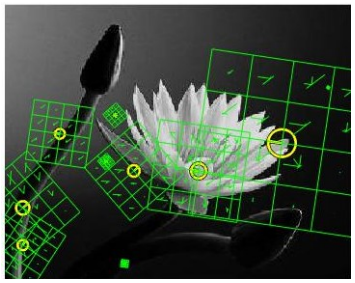
5.1.1 Number of Trees

The Random Forest is an ensemble of binary trees. The committee of different trees enables the overall system to

(a) Image analysed on DoG scales of 4, 8, and 10 respectively.



(b) 10 randomly selected image patches.



(c) Descriptors visualised on top of their patches.

Figure 1: Dense SIFT performed on the image of a water lily.

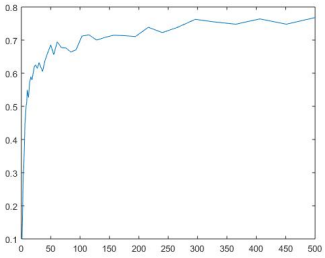
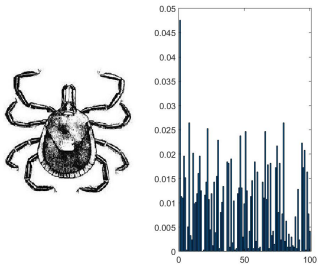
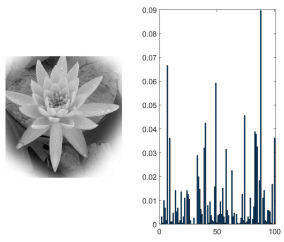


Figure 2: Testing accuracy of RF classifier with increasing values of K



(a) Image from testing set.



(b) Image from training set.

Figure 3: Bag-of-words histograms.

generalise the output decisions by adding a measure of uncertainty over the outcomes. The uncertainty stems from the contribution of each binary classification tree to the decision. It is generally known that binary classification trees are prone to over fitting, therefore, by having a randomisation factor, the cross correlation of the tree outputs will be reduced. Finally, by combining all the tree outputs with a certain policy, the resulting decisions will be a generalisation of the training data. The above notion can be visualised in Figure **??**

5.1.2 Weak Learner

The weak learner, also known as split function, is an important part in the creation of the binary trees. That is, each node has to decide as to which subset of the data S to propagate to the left or right child. These decisions are described by the split function. While a tree can have different kinds of weak learners in each node, in the current implementation, all nodes of the trees in the forest are of the same weak learner type. The most popular split functions are the axis aligned and the two-pixel test. The first function involves simple thresholding of a particular dimension, while the second involves comparing the values of two pixels, which is a rudimentary discrete derivative. Additionally, some nonlinear split functions are explored. A comparison between different split functions can be seen in Figure **??**. It should be noted that the performance of the weak learners is heavily reliant on the type of data under question, as will be demonstrated in later results.

5.1.3 Depth of trees

The depth of trees also plays a significant role on the decision bounds. In essence, as the depth of the decision trees increases, the decision bounds can have a more 'complex shape. This can be demonstrated with the axis-aligned split function for different depths. As can be seen in Figure **??**, with less levels, the forest can only form simple decision

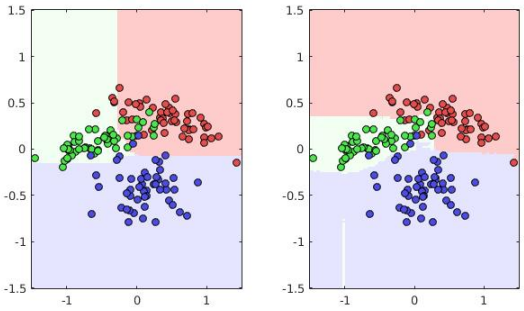


Figure 4: Decision bounds for a 2 level (left) and 8 level (right) depth forest respectively

	Actual Yes	Actual No
Predicted Yes	10.8	4.2
Predicted No	4.2	130.8

Table 1: Ranks of Scatter Matrices

bounds, unlike the 8 levelled forest. However, as the depth increases, the decision bound may be over fitted, something that can be observed in the figure.

5.2. Random Forest as Classifier

The RF can be used as a classifier for the image histograms acquired with the Bag-of-Words method with a good performance. A prediction accuracy of 0.8 was achieved with this procedure with specific parameter values. An average confusion matrix for all classes is shown in Table 1. The RF classifier works by having the histograms obtained by the Bag-of-Words method train the forest. The trained model is then used to test query images, where the output of the leaves is a probability density of the classes. The decision is therefore made by taking the class with the most confidence.

6. RF Codebook

The K-means algorithm is simple in its implementation and will guarantee a local optimal solution. However the large number of computations, mainly the euclidean distance comparisons between the descriptors means that it is not very efficient to use, especially for a large number of images. The use of a modified RF is explored as a substitute to the K-means algorithm for the creating a histogram representation of the images. This can be achieved by 'passing' all descriptors of an image through an RF and then count a vote for each descriptor ending up in a unique leaf id. It should be mentioned that the leafs.

As can be seen in Table 2, even if the testing accuracy

	RF Classifier	RF Codebook and Classifier
Testing Accuracy	0.8	0.7
Time Efficiency	120	50

Table 2: Ranks of Scatter Matrices

	Actual Yes	Actual No
Predicted Yes	9.2	5.8
Predicted No	5.8	129

Table 3: Ranks of Scatter Matrices

achieved by the RF Codebook is smaller than the K-means Codebook, the timing efficiency is less than half. It should be mentioned, that not all the parameters have been tested, and therefore, it possible that the RF Codebook algorithm could achieve even better results. In Table 3 the confusion matrix of the RF codebook is shown. 9.2000 5.8000 5.8000 129.2000

7. Conclusion

In this report images were analysed in codewords using a multi-scale dense SIFT algorithm to extract features, and then performing either a K-means algorithm or passing the data through an RF. The K-means codewords with RF classifier had the best performance with a maximum testing accuracy of 0.8 but with a large computational cost. On the other hand, the RF codebook and classifier performed well with a 0.7 testing accuracy and with a good time efficiency, which was less than half of the K-means algorithm.

References

[1] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 178–178, June 2004.

[2] karpathy. Random-forest-matlab.

[3] D. G. Lowe. 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, 1999.

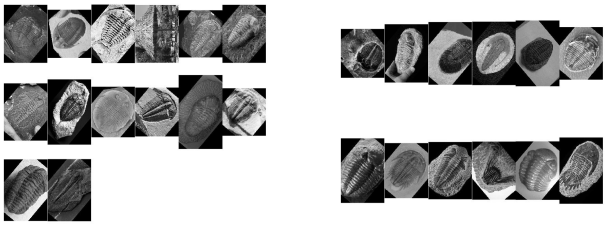
Appendix A. Size of Each Class in Dataset

The number of images in each labelled class of the dataset are given below:

- tick: 49 images
- trilobite: 86 images
- umbrella: 75 images
- watch: 239 images
- water lilly: 37 images
- wheelchair: 59 images
- wild cat: 34 images
- windsor chair: 56 images
- wrench: 39 images
- yin yang: 60 images

Figure 5: One exemplar image picked at random from each of the 10 classes.

Appendix B. Kmeans Classification



(a) Train (b) Test

Figure 6: Example 1

(a) Train (b) Test

Figure 7: Example 2

(a) Train (b) Test

Figure 8: Example 3

(a) Train (b) Test

Figure 9: Example 4

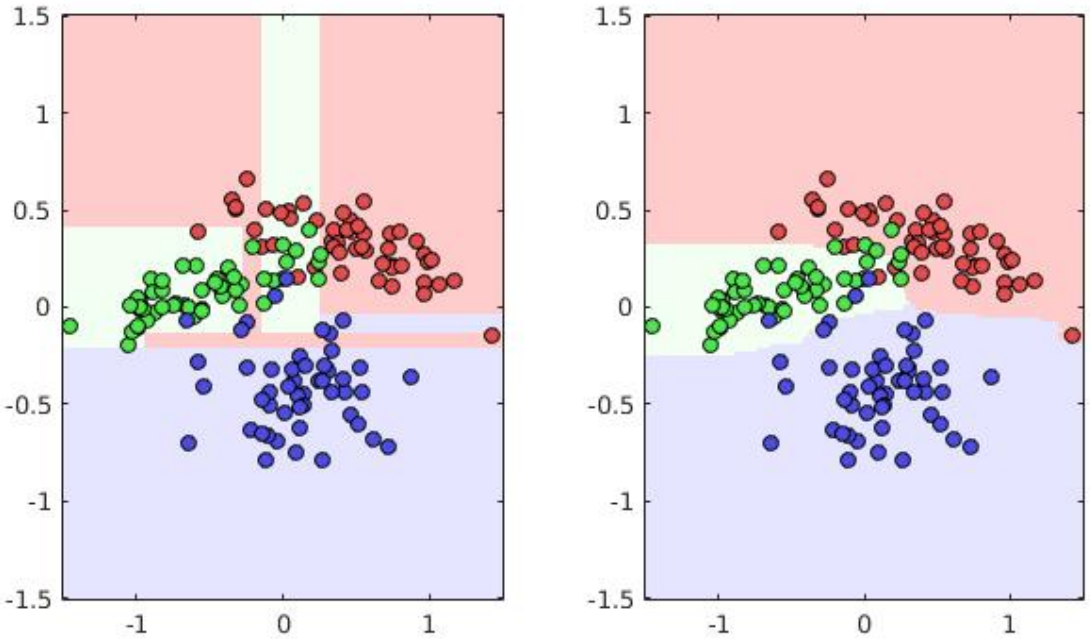


Figure 10: Decision bounds represented by colour for a single tree (left) and 200 trees (right)

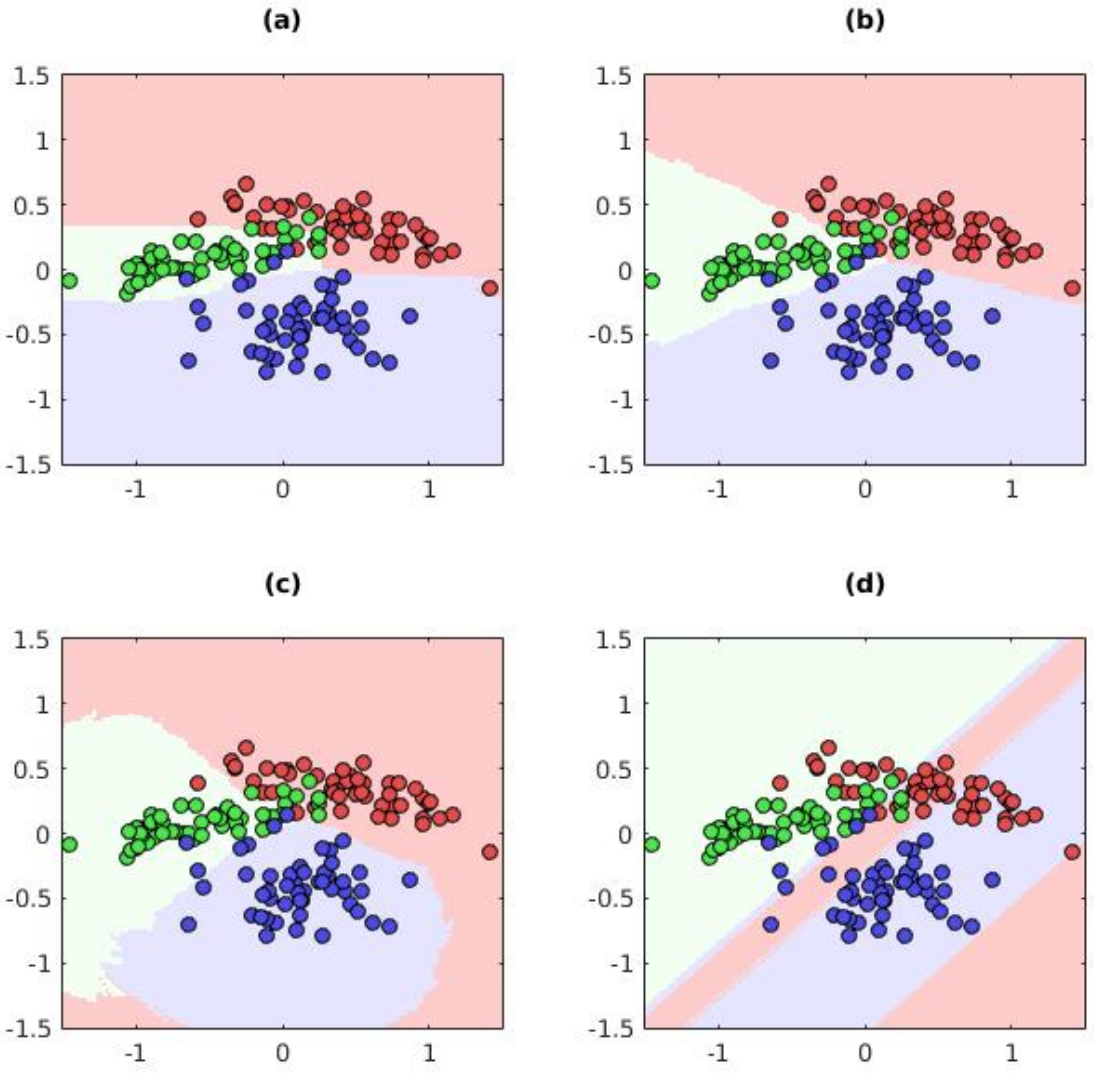


Figure 11: Results for different split functions: (a) Axis-aligned, (b) 2D Axis-aligned, (c) Euclidean distance (d) Two-pixel test