

# Bag of words based image classification

## Introduction

For this project our goal is build a proper SVM classifier able to classify and distinguish between 5 different images: images containing an airplane, images containing a bird, images containing a ship, images containing a horse and images containing a car. The idea is to create 5 binary SVM classifiers for this. Each being able to tell whether an image for example is the image of a car or not. As the title indicates, we used the bag of words implementation to train and test our SVM classifiers. In this report we will evaluate differently trained classifiers with different configurations to see how our SVM classifiers perform under different circumstances. In the end we do expect a classifier trained on RGB or opponent images, using a broad vocabulary with a size of 4000 and dense SIFT for feature extraction to show the best performance. Dense sift gives a constant amount of descriptors of a predefined set of patches, a broad vocabulary can make the classifier more precise and RGB or opponent give more information about an image since they contain 3 channels.

*Demo.m trains all configurations and shows the results afterwards.*

## Training and testing settings

In this section we will elaborate our training configurations to get our SVM classifiers. First of all, we used two different sift operations: keypoint SIFT and dense SIFT. Keypoint SIFT looks for keypoints in a given image and then takes the descriptors of the found keypoints. This will result in a different amount of descriptors per image. Dense SIFT takes descriptors from a predefined set of image patches, which results in a constant amount of descriptors per image. For dense SIFT, we used 2 different configurations: size 8 with step 4 and size 16 with step 8. Further we also used different image types to train our classifiers with: RGB, opponent and grayscale.

To train our SVM classifiers we first need to cluster all image descriptors from a predefined image set, to create a vocabulary. We used K-means to cluster the image descriptors with 3 different hyperparameters: 400, 1000 and 4000 different cluster. This means that we have  $3 \times 3 \times 2 + 9$  different models which we will evaluate all (+9 for the 9 extra for different dense SIFT configurations). Our training set consists of 2500 images and our testing set contains 4000 images. Due to limited computational resources, we were not able to create a K-means cluster with more than 600 images (it took around 30GB of ram at some moments). However we were able to use half of the 2500 images to perform bag of words and all of the 4000 images for testing.

For testing we will evaluate the precision, recall and accuracy of each of the 27 different configurations.

## Testing results and evaluations

### Keypoint SIFT - Precision

color\cluster size	400	1000	4000
RGB	0.41431	0.45392	0.54006
OPPONENT	0.40145	0.43005	0.48031
GRAYSCALE	0.4025	0.39214	0.45582

### Keypoint SIFT - Recall

color\cluster size	400	1000	4000
RGB	0.47925	0.46175	0.423
OPPONENT	0.45825	0.4465	0.4055
GRAYSCALE	0.435	0.41175	0.35725

### Keypoint SIFT - Accuracy

color\cluster size	400	1000	4000
RGB	0.76035	0.78125	0.81255
OPPONENT	0.755	0.77095	0.79335
GRAYSCALE	0.75785	0.7547	0.78615

### Dense SIFT (size: 8, step: 4) - Precision

color\cluster size	400	1000	4000
RGB	0.69212	0.76795	0.8482
OPPONENT	0.62228	0.72797	0.81316
GRAYSCALE	0.64505	0.75972	0.82777

### Dense SIFT (size: 8, step: 4) - Recall

color\cluster size	400	1000	4000
RGB	0.72725	0.74875	0.7655
OPPONENT	0.69275	0.7165	0.7355
GRAYSCALE	0.68875	0.7525	0.74274

### Dense SIFT (size: 8, step: 4) - Accuracy

color\cluster size	400	1000	4000
RGB	0.88075	0.9045	0.9257
OPPONENT	0.85445	0.88975	0.9133
GRAYSCALE	0.86195	0.9029	0.9178

### Dense SIFT (size: 16, step: 8) - Precision

color\cluster size	400	1000	4000
RGB	0.60522	0.6761	0.76543
OPPONENT	0.58443	0.65858	0.74262
GRAYSCALE	0.62735	0.64879	0.74372

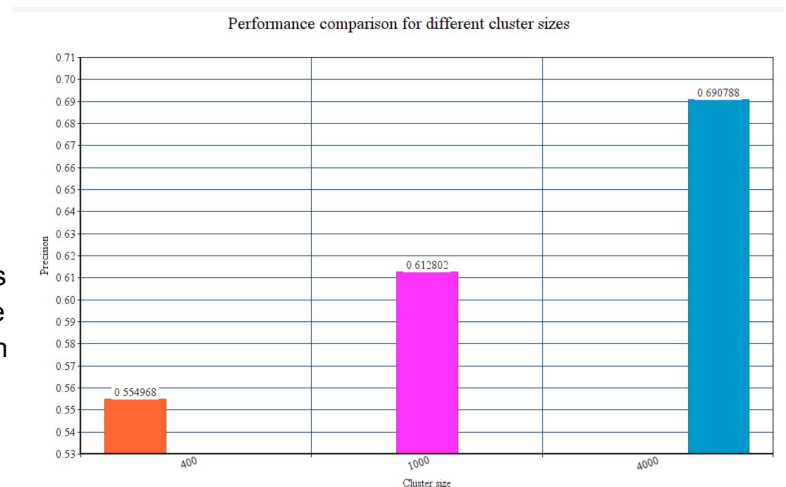
### Dense SIFT (size: 16, step: 8) - Recall

color\cluster size	400	1000	4000
RGB	0.678	0.6925	0.70075
OPPONENT	0.66325	0.6785	0.6665
GRAYSCALE	0.6755	0.6715	0.68125

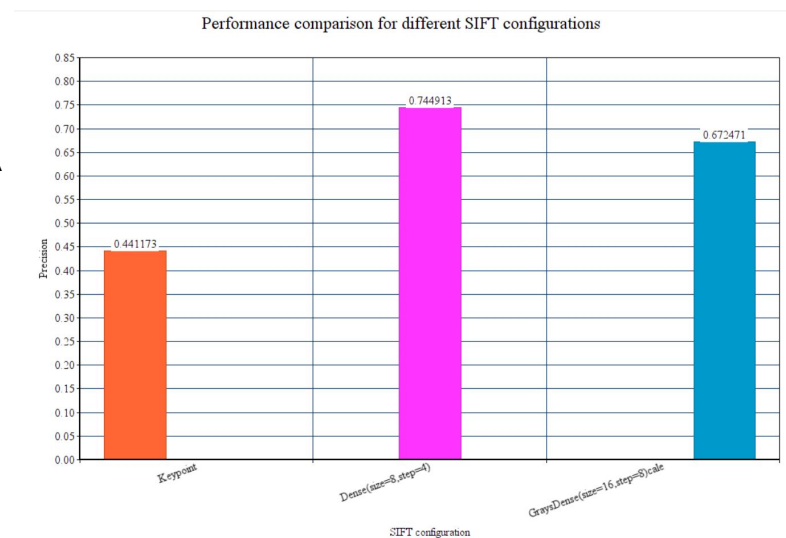
### Dense SIFT (size: 16, step: 8) - Accuracy

color\cluster size	400	1000	4000
RGB	0.84715	0.87215	0.8972
OPPONENT	0.83775	0.86535	0.8871
GRAYSCALE	0.85485	0.8616	0.8893

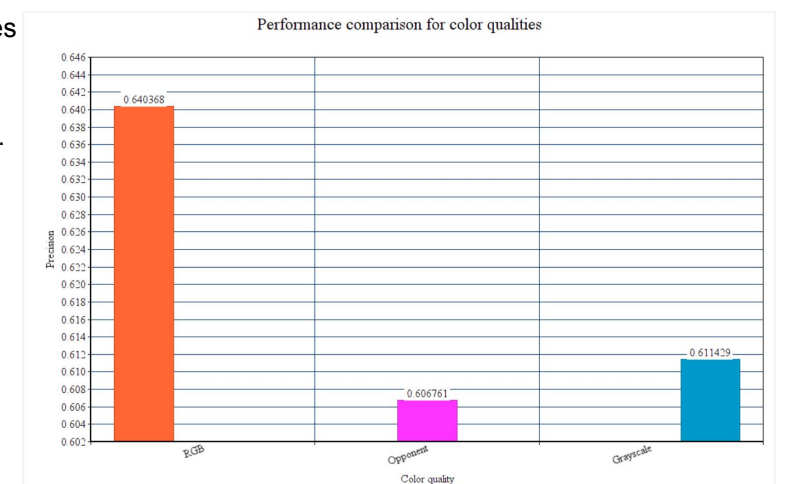
When we look at the performance of different cluster sizes, we can see that generally a higher amount of clusters correlates with a higher precision. When there are more clusters to categorize features to, more detail can be classified. This might be the reason why more clusters can classify an image better, because there are more distinct points that characterize an object. However, while the correlation seems linear in the graph, the amount of clusters are not linear.



For the different SIFT configurations there are quite some differences in performance. Dense SIFT with a size of 8 and a step of 4 performs the best of our 3 configurations. A smaller size and step lead to more details, which can help in categorizing an object. Overall, dense SIFT seems to outperform keypoint SIFT. This might be because dense SIFT looks more to local features than keypoint SIFT. This again gives more detail about the objects which can help in categorizing it.



Our models perform better with RGB images than with grayscale or opponent images. Grayscale performs slightly better than opponent, but this difference is neglectable. RGB can give more information about an object by showing its colors. This might be useful in some situations, where colors are distinct for a certain object. This might explain our results. RGB does not perform much better than the others, as there is a difference of only 3-4%.



The values in the charts are based on the average of our data. This generalizes the data to make it more reliable.

### mAP of all classes of all configurations

confclass	airplane	bird	ship	horse	car
400_dense_rgb	0.42294	0.51306	0.50284	0.54366	0.52036
400_dense_opponent	0.33536	0.48115	0.46674	0.4986	0.38787
400_dense_gray	0.37853	0.4614	0.4651	0.48281	0.42996
400_key points_rgb	0.17711	0.17243	0.24369	0.23129	0.20905
400_key points_opponent	0.17754	0.15981	0.19342	0.21499	0.19787
400_key points_gray	0.15684	0.14219	0.19921	0.20707	0.18617
1000_dense_rgb	0.45013	0.53957	0.67166	0.63075	0.57039
1000_dense_opponent	0.40909	0.53834	0.58502	0.62612	0.46453
1000_dense_gray	0.45344	0.53015	0.62308	0.64953	0.57721
1000_key points_rgb	0.18603	0.20477	0.24455	0.2235	0.20878
1000_key points_opponent	0.16166	0.17401	0.18212	0.22663	0.20976
1000_key points_gray	0.14379	0.12879	0.18322	0.18119	0.16591
4000_dense_rgb	0.57787	0.60989	0.69897	0.70209	0.6454
4000_dense_opponent	0.49944	0.59983	0.67344	0.70545	0.53
4000_dense_gray	0.53615	0.57043	0.65454	0.68034	0.60553
4000_key points_rgb	0.26837	0.17796	0.26051	0.24321	0.25203
4000_key points_opponent	0.19759	0.19331	0.19376	0.2545	0.16814
4000_key points_gray	0.15561	0.1094	0.20417	0.175	0.1804
Average per class	0.315972	0.350361	0.402558	0.415374	0.361631

In the table above we can see mAP values for each class for all configurations. As the tables with the precision of the classifiers combined, the sift method has great influence on the performance of the models. Again, this is most likely caused by the fact that dense sift gives much more information about the images compared to keypoint SIFT. We can also see performance differences per class classifier. In the averages per class we can see that the classifier for horses performs better compared to the airplanes. This could be caused by the fact that the images of horses are usually in very similar situations like standing in a field. While an airplane can be in the air, on the ground, on the water (water airplane) which makes it more difficult to distinguish and classify.

## Conclusion

From the results of our testing we can conclude that the sift method has great influence on the performance of our models. We expect this being caused by the fact that finding keypoints in an image can be quite difficult and result in relatively few descriptors. Dense SIFT gives a constant number of descriptors from an image, such that descriptors from less interesting areas are also retrieved. Therefore dense SIFT also gives a tremendous amount of information about each image. We also noticed this during the creation of our vocabulary, where close to 30GB of RAM was used at some moments, and with the creation of the bag of words.













It was also interesting to notice that having three channels and therefore more information per image, does not always lead to better performance. Grayscale outperforms opponent in many occasions. However, RGB seems to be the best representation of the image in terms of information to create a good classifier with. Image type also has some influence on the performance of a classifier, as shown by the mAP. It is shown that a horse is easier classified than a plane. mAP also shows again that dense SIFT, 4000 cluster size and RGB images make the SVM models perform at their best.

Rico Mossinkoff - 12805157  
Ewoud Vermeij - 11348860

Hannah Lim - 10588973  
Yke Rusticus - 11306386

## Dense configuration:

- Size = 16
- Step = 8

<p>400 dense gray</p> <p>Best classified</p> <p>score: 22.8014   score: 27.8739   score: 23.5981   score: 22.2812   score: 23.6786</p>  <p>Worst classified</p> <p>score: 1.6084   score: 1.8225   score: 1.9673   score: 2.1529   score: 1.4723</p>  <p>Precision: 0.62735. Recall: 0.6755. Accuracy: 0.85485</p>	<p>400 dense opponent</p> <p>Best classified</p> <p>score: 55.2313   score: 68.6447   score: 54.654   score: 57.642   score: 57.6111</p>  <p>Worst classified</p> <p>score: 2.1354   score: 2.3467   score: 2.7863   score: 2.6635   score: 1.0775</p>  <p>Precision: 0.58443. Recall: 0.65325. Accuracy: 0.83775</p>
<p>400 dense rgb</p> <p>Best classified</p> <p>score: 51.5157   score: 62.3542   score: 52.2297   score: 51.0316   score: 51.1837</p>  <p>Worst classified</p> <p>score: 3.3826   score: 2.1721   score: 3.5645   score: 3.6041   score: 3.2762</p>  <p>Precision: 0.60522. Recall: 0.678. Accuracy: 0.84715</p>	<p>400 key points gray</p> <p>Best classified</p> <p>score: 18.9973   score: 16.4773   score: 16.7488   score: 16.8492   score: 16.227</p>  <p>Worst classified</p> <p>score: 2.0725   score: 1.6282   score: 2.1138   score: 1.8864   score: 2.1302</p>  <p>Precision: 0.40491. Recall: 0.421. Accuracy: 0.76045</p>
<p>400 key points opponent</p> <p>Best classified</p> <p>score: 28.8756   score: 30.3306   score: 38.0573   score: 37.002   score: 35.5636</p>  <p>Worst classified</p> <p>score: 2.4659   score: 1.9147   score: 1.8826   score: 1.9523   score: 2.3852</p>  <p>Precision: 0.41615. Recall: 0.45725. Accuracy: 0.76315</p>	<p>400 key points rgb</p> <p>Best classified</p> <p>score: 42.0783   score: 34.251   score: 35.3056   score: 43.6775</p>  <p>Worst classified</p> <p>score: 3.0701   score: 3.2332   score: 3.1023   score: 3.2416   score: 3.2894</p>  <p>Precision: 0.42027. Recall: 0.48625. Accuracy: 0.7631</p>















Rico Mossinkoff - 12805157  
Ewoud Vermeij - 11348860

Hannah Lim - 10588973  
Yke Rusticus - 11306386

## Dense configuration:

- Size = 16
- Step = 8

<p>1000 dense gray</p> <p>Best classified</p> <p>score: 12.881   score: 11.7813   score: 10.8244   score: 11.6105   score: 10.8367</p>  <p>Worst classified</p> <p>score: 1.1859   score: 1.3573   score: 1.3458   score: 1.1   score: 1.3224</p>  <p>Precision: 0.64879. Recall: 0.6715. Accuracy: 0.8616</p>	<p>1000 dense opponent</p> <p>Best classified</p> <p>score: 8.6446   score: 8.5885   score: 9.0016   score: 13.0836</p>  <p>Worst classified</p> <p>score: 0.85322   score: 0.90985   score: 0.79129   score: 0.86639   score: 0.87258</p>  <p>Precision: 0.65858. Recall: 0.6785. Accuracy: 0.86535</p>
<p>1000 dense rgb</p> <p>Best classified</p> <p>score: 9.2711   score: 8.7327   score: 8.6337   score: 8.6916   score: 8.6724</p>  <p>Worst classified</p> <p>score: 1.4231   score: 1.3742   score: 1.3906   score: 1.2952   score: 1.3842</p>  <p>Precision: 0.6761. Recall: 0.6925. Accuracy: 0.87215</p>	<p>1000 key points gray</p> <p>Best classified</p> <p>score: 9.4927   score: 9.4677   score: 10.9814   score: 10.4321   score: 10.8692</p>  <p>Worst classified</p> <p>score: 1.0263   score: 1.3169   score: 1.2143   score: 1.2471   score: 1.2616</p>  <p>Precision: 0.40692. Recall: 0.41475. Accuracy: 0.76205</p>
<p>1000 key points opponent</p> <p>Best classified</p> <p>score: 15.3507   score: 15.5264   score: 15.2382   score: 14.3044   score: 14.3668</p>  <p>Worst classified</p> <p>score: 1.2594   score: 1.3   score: 1.1144   score: 1.2958   score: 1.2839</p>  <p>Precision: 0.41542. Recall: 0.44325. Accuracy: 0.7639</p>	<p>1000 key points rgb</p> <p>Best classified</p> <p>score: 9.3299   score: 9.5838   score: 10.0294   score: 9.2899   score: 10.3771</p>  <p>Worst classified</p> <p>score: 0.97352   score: 0.97744   score: 1.0599   score: 1.1242   score: 1.1908</p>  <p>Precision: 0.45754. Recall: 0.489. Accuracy: 0.78185</p>











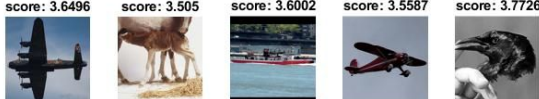



Rico Mossinkoff - 12805157  
Ewoud Vermeij - 11348860

Hannah Lim - 10588973  
Yke Rusticus - 11306386

## Dense configuration:

- Size = 16
- Step = 8

<p>4000 dense gray</p> <p>Best classified</p> <p>score: 4.1415   score: 4.6396   score: 4.8802   score: 4.8703   score: 4.1561</p>  <p>Worst classified</p> <p>score: 0.81831   score: 0.81668   score: 0.85171   score: 0.79148   score: 0.78849</p>  <p>Precision: 0.74372. Recall: 0.68125. Accuracy: 0.8893</p>	<p>4000 dense opponent</p> <p>Best classified</p> <p>score: 3.7881   score: 3.7832   score: 3.7661   score: 4.0019   score: 3.7406</p>  <p>Worst classified</p> <p>score: 0.80698   score: 0.80901   score: 0.78091   score: 0.81103   score: 0.78194</p>  <p>Precision: 0.74262. Recall: 0.6665. Accuracy: 0.8871</p>
<p>4000 dense rgb</p> <p>Best classified</p> <p>score: 4.2605   score: 4.6014   score: 4.4358   score: 4.4614   score: 5.0681</p>  <p>Worst classified</p> <p>score: 0.87162   score: 0.88214   score: 0.8375   score: 0.86388   score: 0.85713</p>  <p>Precision: 0.76543. Recall: 0.70075. Accuracy: 0.8972</p>	<p>4000 key points gray</p> <p>Best classified</p> <p>score: 4.3023   score: 3.9815   score: 4.0181   score: 4.2721   score: 3.9774</p>  <p>Worst classified</p> <p>score: 0.81611   score: 0.63856   score: 0.83623   score: 0.8007   score: 0.85112</p>  <p>Precision: 0.44836. Recall: 0.35275. Accuracy: 0.78375</p>
<p>4000 key points opponent</p> <p>Best classified</p> <p>score: 6.41   score: 7.422   score: 7.2612   score: 7.0378   score: 6.5684</p>  <p>Worst classified</p> <p>score: 0.87573   score: 0.8796   score: 0.86648   score: 0.89212   score: 0.83477</p>  <p>Precision: 0.4654. Recall: 0.4035. Accuracy: 0.788</p>	<p>4000 key points rgb</p> <p>Best classified</p> <p>score: 3.6496   score: 3.505   score: 3.6002   score: 3.5587   score: 3.7726</p>  <p>Worst classified</p> <p>score: 0.84236   score: 0.78538   score: 0.75885   score: 0.7219   score: 0.8377</p>  <p>Precision: 0.54263. Recall: 0.43275. Accuracy: 0.8136</p>

Rico Mossinkoff - 12805157  
Ewoud Vermeij - 11348860

Hannah Lim - 10588973  
Yke Rusticus - 11306386

## Dense configuration:

- Size = 8
- Step = 4.

<p>400 dense gray</p> <p>Best classified</p> <p>score: 17.4268   score: 18.2248   score: 17.7925   score: 25.4433   score: 20.7526</p>  <p>Worst classified</p> <p>score: 2.4916   score: 2.54   score: 1.2199   score: 2.4192   score: 1.0618</p>  <p>Precision: 0.64505. Recall: 0.68875. Accuracy: 0.86195</p>	<p>400 dense opponent</p> <p>Best classified</p> <p>score: 24.8892   score: 23.757   score: 24.4018   score: 23.908   score: 29.7392</p>  <p>Worst classified</p> <p>score: 2.3661   score: 2.5183   score: 2.2454   score: 2.6112   score: 2.1743</p>  <p>Precision: 0.62228. Recall: 0.69275. Accuracy: 0.85445</p>
<p>400 dense rgb</p> <p>Best classified</p> <p>score: 16.577   score: 17.3957   score: 16.3882   score: 16.7276   score: 17.1478</p>  <p>Worst classified</p> <p>score: 2.0889   score: 2.2713   score: 1.8258   score: 2.0856   score: 2.1671</p>  <p>Precision: 0.69212. Recall: 0.72725. Accuracy: 0.88075</p>	<p>1000 dense gray</p> <p>Best classified</p> <p>score: 6.2745   score: 6.1007   score: 7.2651   score: 6.0584   score: 6.0831</p>  <p>Worst classified</p> <p>score: 1.1392   score: 1.2176   score: 1.0171   score: 1.2147   score: 1.1575</p>  <p>Precision: 0.75972. Recall: 0.7525. Accuracy: 0.9029</p>
<p>1000 dense opponent</p> <p>Best classified</p> <p>score: 7.9753   score: 8.032   score: 8.455   score: 9.8215   score: 8.4422</p>  <p>Worst classified</p> <p>score: 1.1392   score: 1.0895   score: 1.1706   score: 1.0778   score: 1.0235</p>  <p>Precision: 0.72797. Recall: 0.7165. Accuracy: 0.88975</p>	<p>1000 dense rgb</p> <p>Best classified</p> <p>score: 6.6517   score: 6.2343   score: 6.6025   score: 6.3532   score: 6.2201</p>  <p>Worst classified</p> <p>score: 1.1585   score: 1.1234   score: 1.2006   score: 1.1368   score: 1.1628</p>  <p>Precision: 0.76795. Recall: 0.74875. Accuracy: 0.9045</p>
<p>4000 dense gray</p> <p>Best classified</p> <p>score: 4.0852   score: 3.5814   score: 3.7115   score: 3.7432   score: 3.8697</p>  <p>Worst classified</p> <p>score: 0.88175   score: 0.87012   score: 0.83545   score: 0.80627   score: 0.85364</p>  <p>Precision: 0.82777. Recall: 0.74375. Accuracy: 0.9178</p>	<p>4000 dense opponent</p> <p>Best classified</p> <p>score: 3.7655   score: 3.7842   score: 3.6873   score: 4.1655   score: 3.7274</p>  <p>Worst classified</p> <p>score: 0.94568   score: 0.85842   score: 0.96421   score: 0.93111   score: 0.96445</p>  <p>Precision: 0.81316. Recall: 0.7355. Accuracy: 0.9133</p>
<p>4000 dense rgb</p> <p>Best classified</p> <p>score: 3.6533   score: 3.754   score: 3.7146   score: 3.8276   score: 3.5387</p>  <p>Worst classified</p> <p>score: 0.80069   score: 0.7966   score: 0.87432   score: 0.7699   score: 0.86793</p>  <p>Precision: 0.8482. Recall: 0.7655. Accuracy: 0.9257</p>	