M3 - Image Classification

Week 1 - Basic Image Classification with Local Features and SVMs

Team 3: María Cristina Bustos Rodríguez
Lidia Garrucho Moras
Xián López Álvarez
Xènia Salinas Ventalló

1. Modularize the code

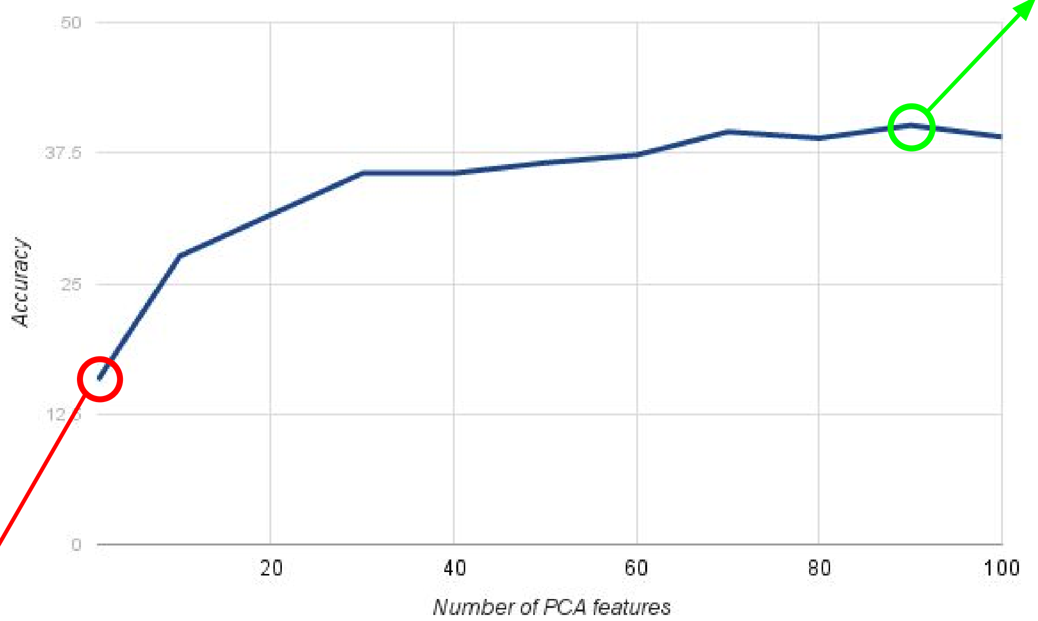
Generate functions for each system step:

- **Features extraction**: function to extract the features from input images.
- **Train classifier:** function to train the classifier with the obtained features.
- **Test system:** predict the classification for the test images using the resulting classifier.

All the functions were stored at file **session1.py**. Another function, in **main.py**, calls and runs the system. To run the system in the cluster in parallel, other functions were created. These allow to create job arrays.

2. Apply PCA

<u>System configuration:</u>
- **Local features:** SIFT
- **# features:** 100
- **SVM:**
  - **Kernel:** linear
  - **C param:** 1

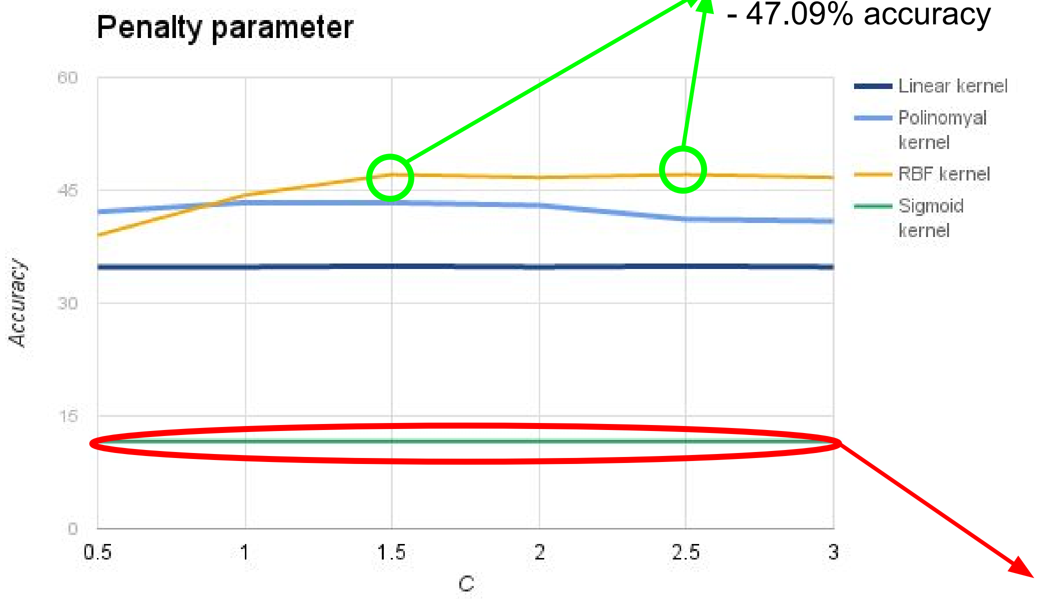


Best result
- 90 PCA features
- 40.14% accuracy

Worst result
- 1 PCA feature
- 15.74% accuracy

Without applying PCA → 38.17% accuracy

3. SVM parameters and kernels (I)



Best result
- RBF kernel and C = 1.5 or 2.5
- 47.09% accuracy

Worst result
- Sigmoid kernel
- 11.69% accuracy

System configuration:
- **Local features:** SIFT
- **# features:** 100
- **PCA:** 30

Polynomial kernel:
- **Degree:** 3
- **Coef:** 0

RBF kernel:
- **Gamma:** auto

Sigmoid kernel:
- **Coef:** 0

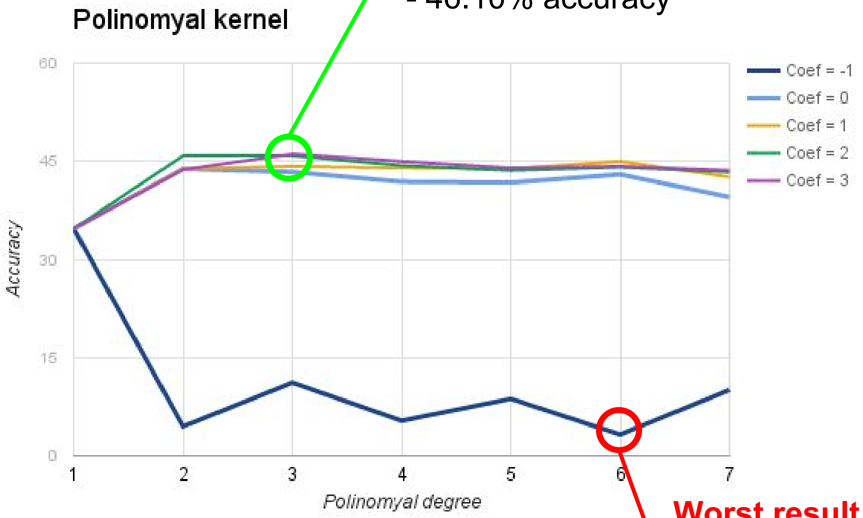
3. SVM parameters and kernels (II)

System configuration:

- **Local features:** SIFT
- **# features:** 100
- **PCA:** 30
- **SVM C param:** 1

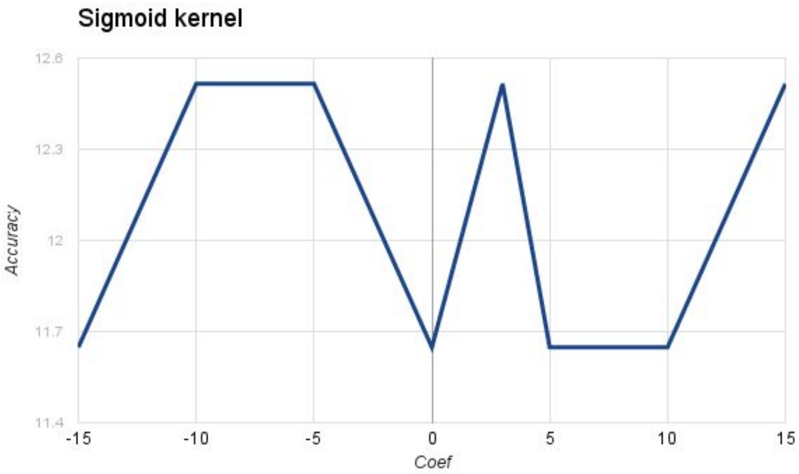
Best result

- Coef = 3 and degree = 3
- 46.10% accuracy



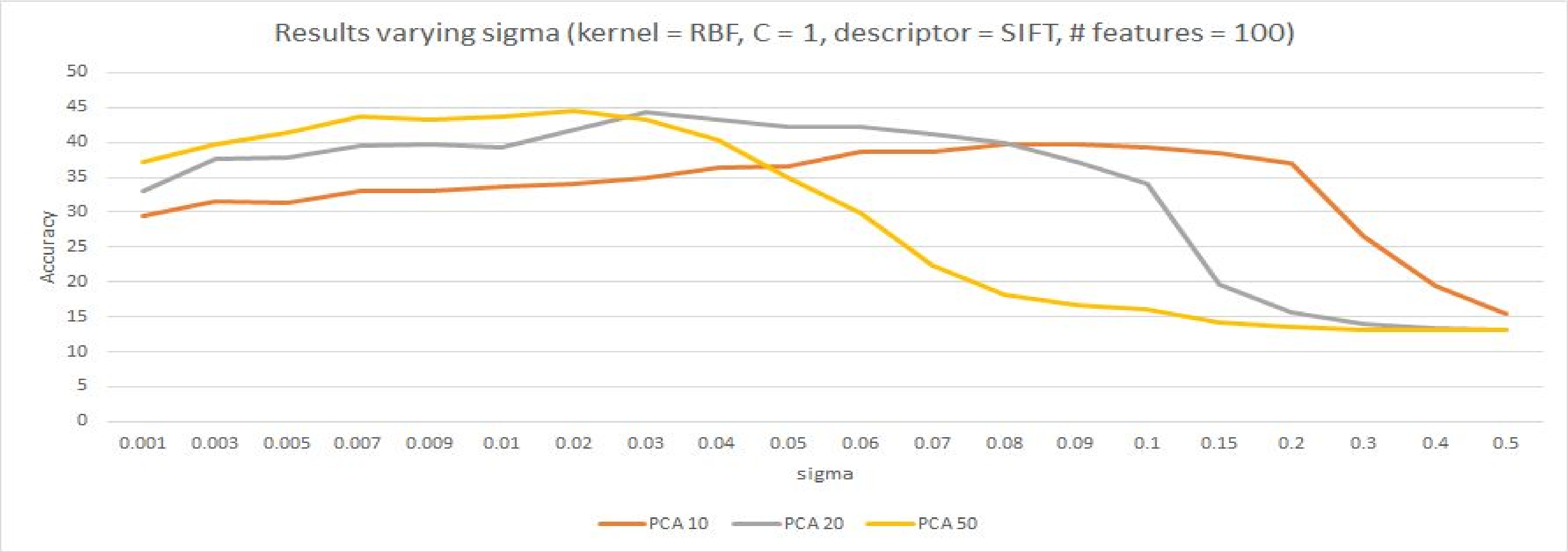
Worst result

- Coef = -1 and degree = 6
- 3.22% accuracy



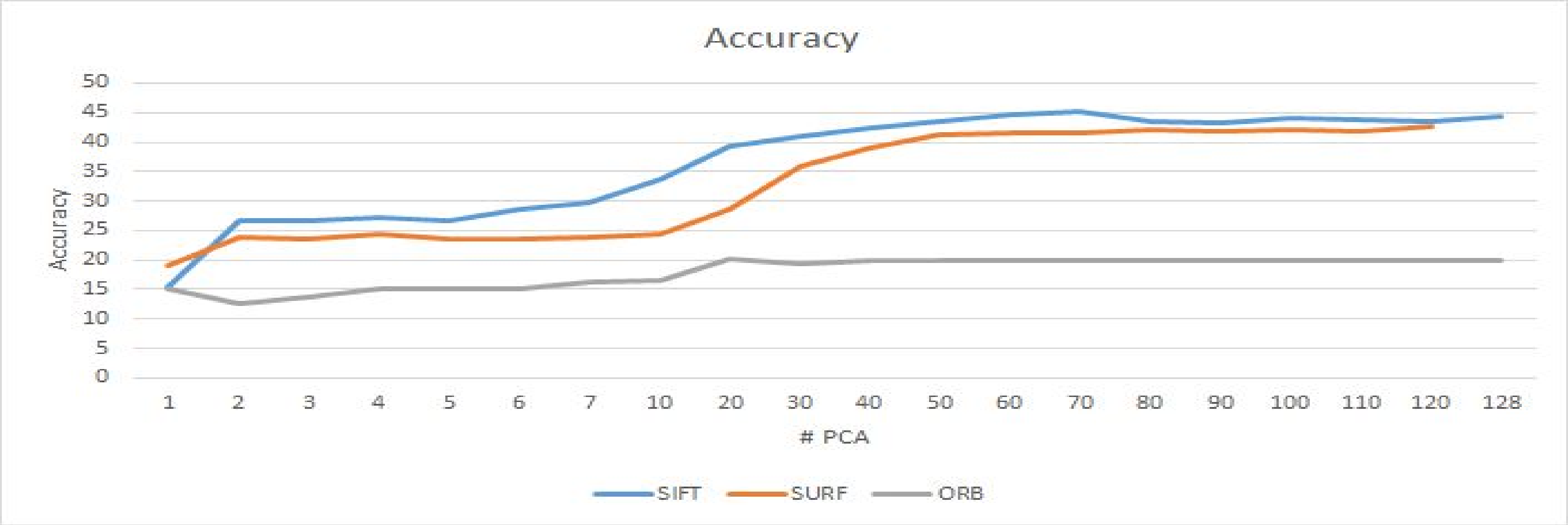
Extremely bad results

3. SVM parameters and kernels (III)



Conclusion: for less # of variables, SVM with RBF kernel needs smaller sigma.

Varying the number of principal components (I)



RBF kernel with sigma = 0.01.

SIFT: 100 features

SURF: Hessian ths = 400

ORB: 100 features

Varying the number of principal components (II)

RBF kernel with sigma = 0.01.

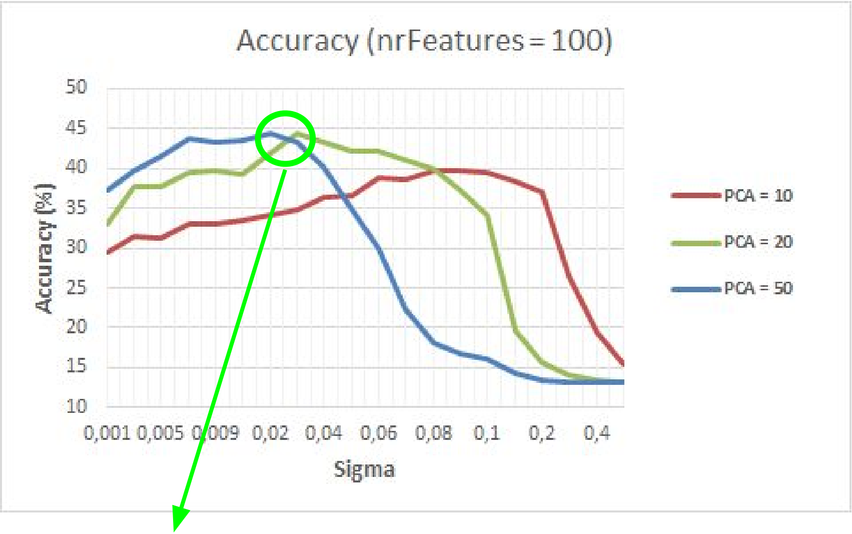
SIFT: 100 features

SURF: Hessian ths = 400

ORB: 100 features

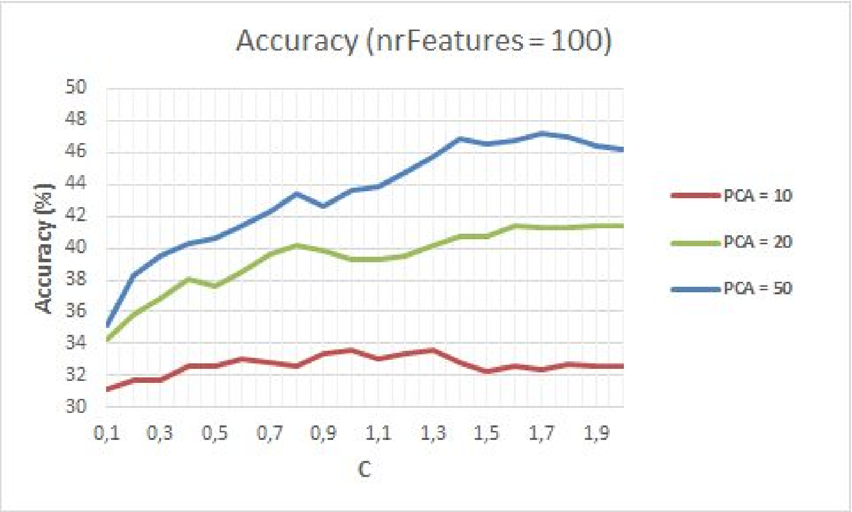
4.1 SIFT

SIFT with SVM with RBF Kernel (C = 1)



For a sigma around 0.02 the results using PCA with 20 features or with 50 are very similar.

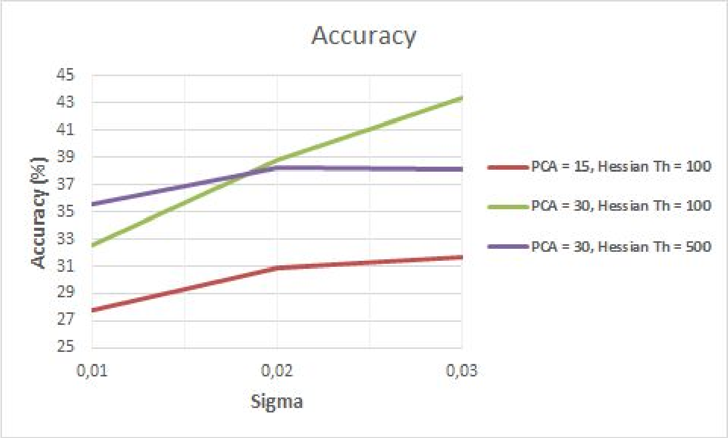
SIFT with SVM with RBF Kernel (sigma = 0.01)

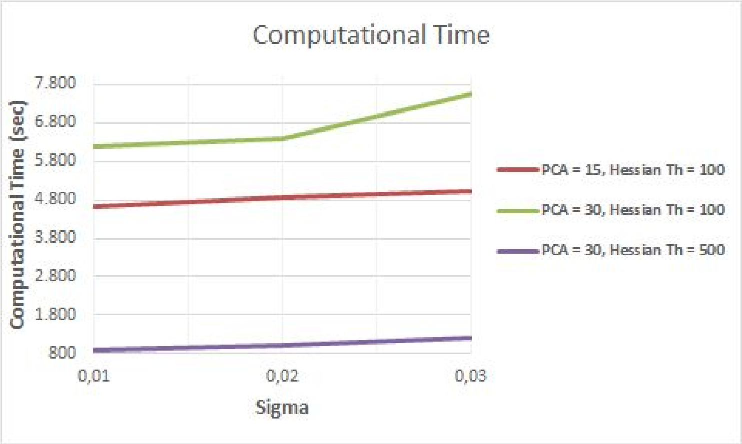


The effect of varying C is more significant when the number of PCA features increases. A value of C = 1.5 gives a good result, and increasing it does not increase the accuracy.

4.2 Other local features: SURF

SURF with SVM with RBF Kernel (C = 1)





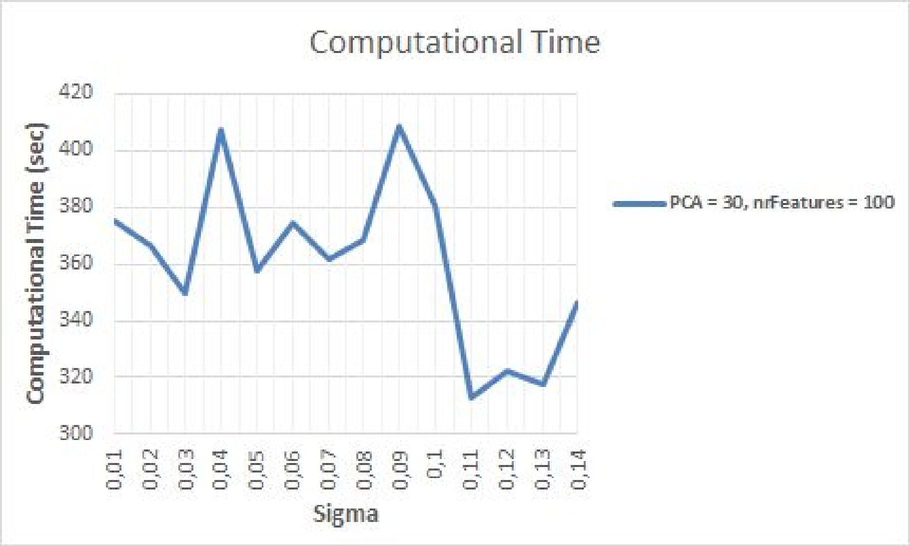
Using SURF, increasing the Hessian Threshold turns out in decreasing the computational time of the local feature descriptor. This makes sense since that way more keypoints are considered, being a higher amount of data to process.
As the algorithm performance is a trade-off between the accuracy and the computational time, the best parameters for SURF would be:

- Hessian Threshold = 500
- PCA = 30 features

4.3 Other local features: ORB

ORB with SVM with RBF Kernel (C = 1)

Best result
- **Sigma**: 0.09
- **Accuracy**: 25.52%



4.3 Comparison of local features

- The best trade-off between accuracy and computational time was found using SIFT.
- We obtained very high computational times with SURF. This is due to the fact that we did not limit the amount of keypoints it extracts from each image. In order to obtain a more fair comparison with the other descriptors (in accuracy and in time), this should be addressed.
- The computational time of ORB is similar to the one of SIFT but the accuracy is lower.

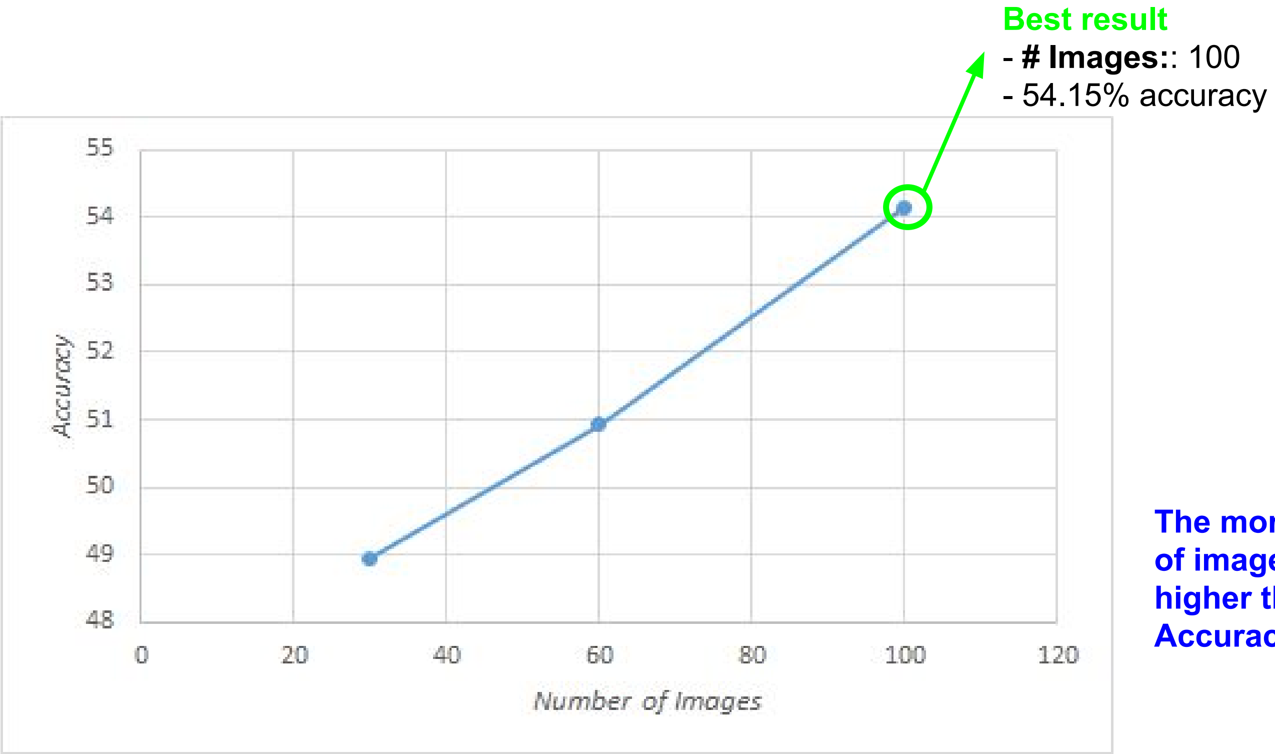
5.1. SIFT: varying the number of local features

Best result
- **# features**: 200
- 45.97% accuracy

__**System configuration:**__
- **Local features:** SIFT
- **PCA Components:** 50
- **SVM:**
  - **Kernel:** rbf
  - **Sigma:** 0.01

5.2. Varying the number of training set images

System configuration:
- **Local features:** SIFT
- **# features:** 100
- **PCA Components:** 50
- **SVM:**
  - **Kernel:** rbf
  - **Sigma:** 0.01
  - **C:** 20



The more number of images, the higher the Accuracy

5.3. Aggregating predictions

Instead of using the predicted class for each keypoint, we take into account the probability it has for each class. For a given image, we sum over all the keypoints the probabilities found for each class, and select the highest.

	Simple aggregation	Probability aggregation
Accuracy	33.0855018587	43.7422552664

Results with RBF kernel, sigma = 0.01, SIFT descriptor, 100 features, and # PCA = 10.

Conclusions:

- The more number of images used in the training, the higher the accuracy.
- Principal Component Analysis: using more PCA features increases the accuracy, a good trade-off between the accuracy and the computational time is around 30 PCA features.
- In the case of SURF, increasing the PCA has a significant effect in the computational cost of the algorithm.
- The best results were obtained using SIFT and SURF local feature descriptors.
- The best accuracy was obtained with the following SVM kernels:
    - Polynomial of degree 3
    - Radial basis function (RBF) with C=1,5
- The probability aggregation was found to be a better technique to classify the test images.