MCV-M3: Image Classification (week 1)

Jonatan Poveda, Martí Cobos

Image Classification Pipeline

Train Database

Input Images and Labels

Feature Extraction

Selectable Features

Save Features

Save Features to test several Classifiers

Classifier Training

Selectable Classifier

Classification model

Test Database

Input Images and Labels

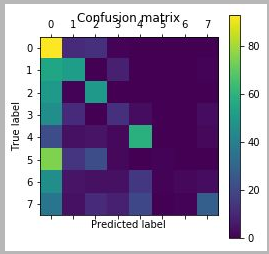
Feature Extraction

Classify Image

Feature extraction and classification performed per image in parallel

Evaluate Code Performance

Compute metrics using ground truth



Task 1: Modularize the code

- Implemented various classes to separate code by domain
  - Database
  - FeatureExtractor
  - Classifier
  - Evaluator
- Generalize code by implementing inheritance and polymorphism
  - Allows to switch descriptor or classifier with minimal changes in the code
- Implement multi-threading on image prediction
  - Using all the cores available
- Database improve speed for testing multiple classificators
  - Allows to load already computed descriptors instead of recomputing each test

Task 1: Modularize the code

Multithreading:

- Multithreading implemented on image prediction using a pool of workers.
- Pool.map method used for running the pool of workers.
- Code efficiency summarized on the following table (using several classification algorithms):

Efficiency	Method 1	Method 2	Method 3	Method 4
Single threading [s]	832,49	67,68	117,88	13,83
Multi-threading [s] (*)	476,73	55,61	21,04	10,80
Improvement [%]	42,7	17,8	82,2	21,9

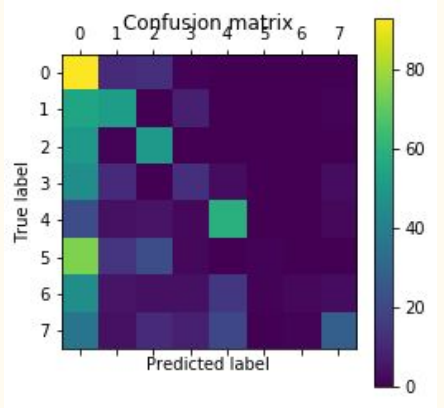
Method 1: SIFT+KNN
Method 2: SIFT+gaussian bayesian
Method 3: Color_Hist+Knn bayesian
Method 4: Color_Hist+gaussian bayesian

(*) In a 4-core CPU

Task 2: Implement performance evaluation measures and experimental protocols

Performance evaluation metrics:

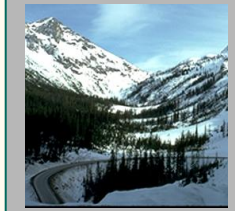
- Evaluator class has been implemented to assess code performance after assessing test images. Implementation based on sklearn.metrics class.
- Code performance metrics implemented:
  - Confusion matrix: Great for visualize similar classes on the feature space in use.
  - Accuracy; Metric used to assess correct classifications vs entire dataset
  - Precision
  - Recall
  - F-score: Metric used to asses overall classifier performance
- Image Classification algorithm evaluated on every iteration



Task 3: Global description of the image

Color Histogram

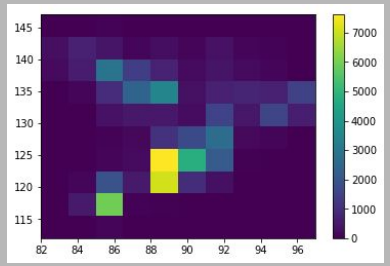
- A global feature descriptor based on the image color histogram has been implemented. Luv color space used for color histogram.
- Number of descriptors per image = (number of bins)^2
- The following flowchart summarizes the process to obtain the color histogram descriptor:



Input Images

Convert image to Luv

Compute 2D histogram of u, v



Convert matrix to 1D array

Task 4: Classifiers testing

Classifiers

- A Classifier builder has been implemented to select between the following classifiers:
  - K-NN
  - Random Forest
  - Gaussian Bayesian Classifier
  - Bernoulli Bayesian Classifier
  - SVM
  - Logistic Regression: Custom implementation described on task 5
- All classifiers are implements using sklearn library except Logistic Regression
- Performance results described on performance evaluation metrics slide

Task 5: Logistic regression

Custom Implementation

- A custom implementation of the logistic regression classifier has been implemented with the following features:
  - Convergence detection
  - Selectable maximum number of iterations
  - Lasso regularization with selectable lambda parameter
  - One vs All multiclass classification
- The following equation defines the cost function of the gradient descent algorithm implemented

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} \theta_j$$

Classification performance evaluation

Procedure to obtain the best combination of descriptor-classifier

1. With SVM classifier, apply 3 variants of SIFT.
2. With SVM classifier, apply 3 variants of color histogram descriptor.
3. With the best descriptor, apply all classifiers implemented.
4. With the best classifier and descriptor, apply 3 variants to the classifier.

Results of every test case shown on following slide.

Classification performance evaluation

Method 1: SIFT (50 features) + SVM
Method 2: SIFT(100 features) + SVM
Method 3: SIFT(200 features) + SVM
Method 4: ColorHist (64 bins) + SVM
Method 5: ColorHist (32 bins) + SVM
Method 6: ColorHist (16bins) + SVM
Method 7: ColorHist (32 bins) + K-NN
Method 8: ColorHist (32 bins) + Random Forest
Method 9: ColorHist (32 bins) + Gaussian Bayes
Method 10: ColorHist (32 bins) + Bernoulli Bayes
Method 11: ColorHist (32 bins) + LogisticRegression
Method 12: ColorHist (32 bins) + SVM (C=0.5)
Method 13: ColorHist (32 bins) + SVM (C=1)
Method 14: ColorHist (32 bins) + SVM (C=5)

Pixel based	Accuracy	Precision	Recall	F1-score	time[s]
Method 1	0.094	0.012	0.125	0.022	46.72
Method 2	0.193	0.110	0.177	0.136	82.73
Method 3	0.149	0.035	0.129	0.055	186.50
Method 4	0.234	0.246	0.238	0.241	9.83
Method 5	0.264	0.271	0.266	0.268	2.48
Method 6	0.263	0.222	0.264	0.242	2.23
Method 7	0.196	0.234	0.192	0.211	17.82
Method 8	0.221	0.224	0.224	0.224	1.26
Method 9	0.247	0.251	0.245	0.248	1.21
Method 10	0.252	0.213	0.247	0.229	1.20
Method 11	0.161	0.167	0.150	0.158	3.06
Method 12	0.257	0.256	0.257	0.257	1.25
Method 13	**0.265**	**0.274**	**0.267**	**0.270**	**1.17**
Method 14	0.255	0.248	0.256	0.252	1.37

Conclusions

- Multiprocessing decreases drastically code execution when the number of features is high.
- Color histogram descriptor faster to compute because it has less number of features than SIFT descriptor
- Performance results are not good because the descriptors in use do not separate well enough the different classes.
- Best results found using 32 bins color histogram descriptor and SVM (C=1) classifier. Those results did not improve (F1) initial classification results with SIFT descriptor and KNN classifier:

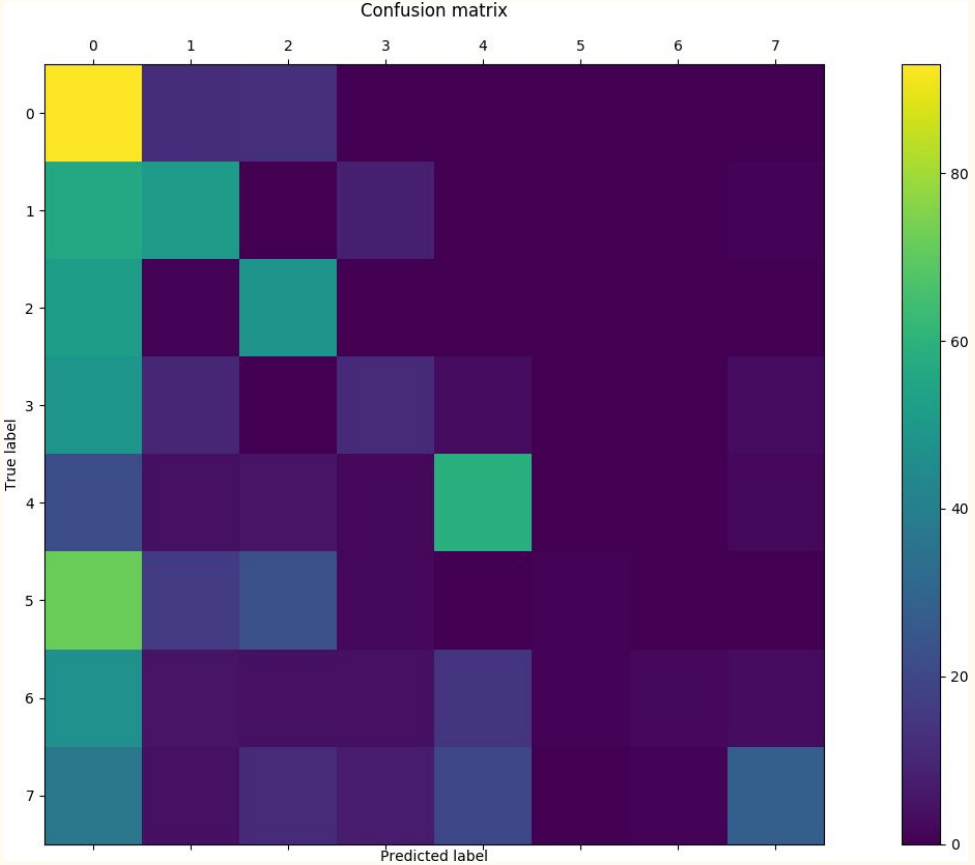
Pixel based	Accuracy	Precision	Recall	F1-score	time[s]
Method 0	0.363	0.504	0.346	0.411	112.34
Method 13	0.265	0.274	0.267	0.270	1.17

Method 0:
SIFT (100 features) + K-NN (5 neighbours)
Method 13:
ColorHist (32 bins) + SVM (C=1)

Conclusions

- Method 0: SIFT (100 features) + K-NN (5 neighbours)

- The Confusion Matrix shows that there are problems specially when classifying nearly all classes with class-0 (Opencountry).



Conclusions

- Even the log of the Confusion Matrix shows there are a lot of mistakes when classifying amongst all the classes