VISUAL RECOGNITION

Eric López, Gerard Martí, Sergio Sancho, Adriana Fernández

Master in Computer Vision, Barcelona
UAB, UOC, UPC, UPF

April 2015

Contents

Summary

Hands on ConvNet

Conclusions

Summary

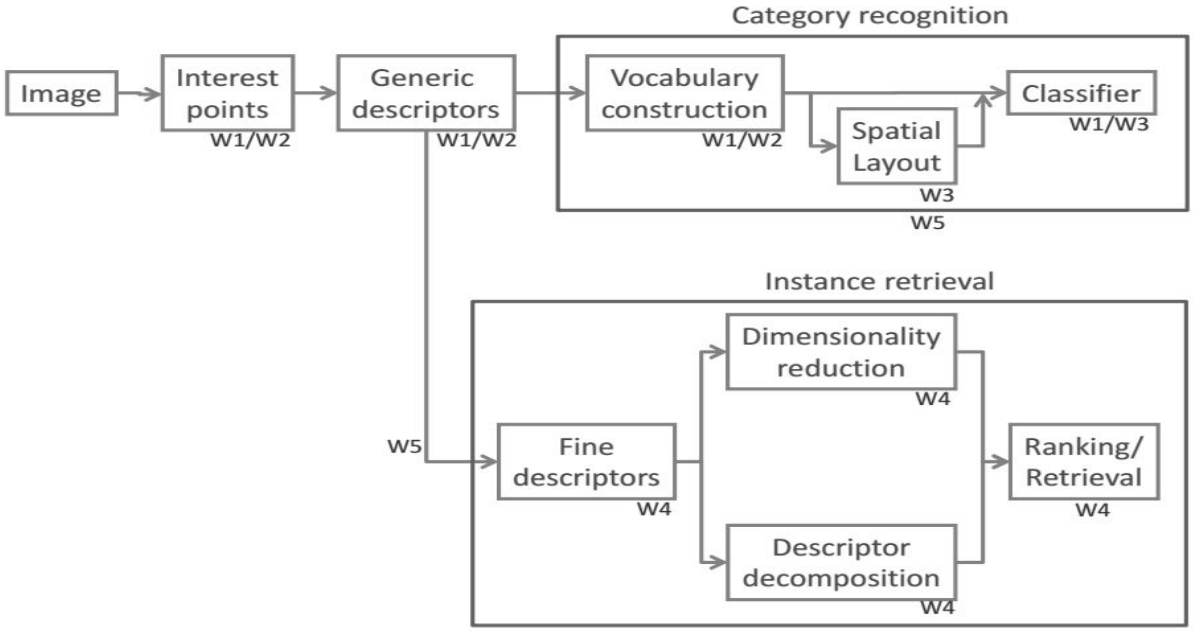

Figure 1.1 Flowchart of the system.

Summary W1-3 II

#Week	K value	Weighting	Pyramid	Detector	Descriptor	Classifier	Acc.
1	K = 1000	-	-	-	SIFT	SVM	0.74
2	K = 5000	-	-	FAST	SIFT	SVM	0.84
3	K = 5000	0.65/0.35	SPM2X2	FAST	SIFT + ColorNaming	HISVM	0.89

Table 1.1 Shows our best performance obtained during the first three weeks. The number of samples was 50000, the cross validation with 5 folds and the SVM with C = 0.01.

Conclusions

- SPM2X2, weighted Intermediate fusion and HISVM achieves the best performance (more than a SPM3X1).
- FAST detector outperforms the others.
- SIFT descriptor works pretty well.

Summary W4 I - Query methods

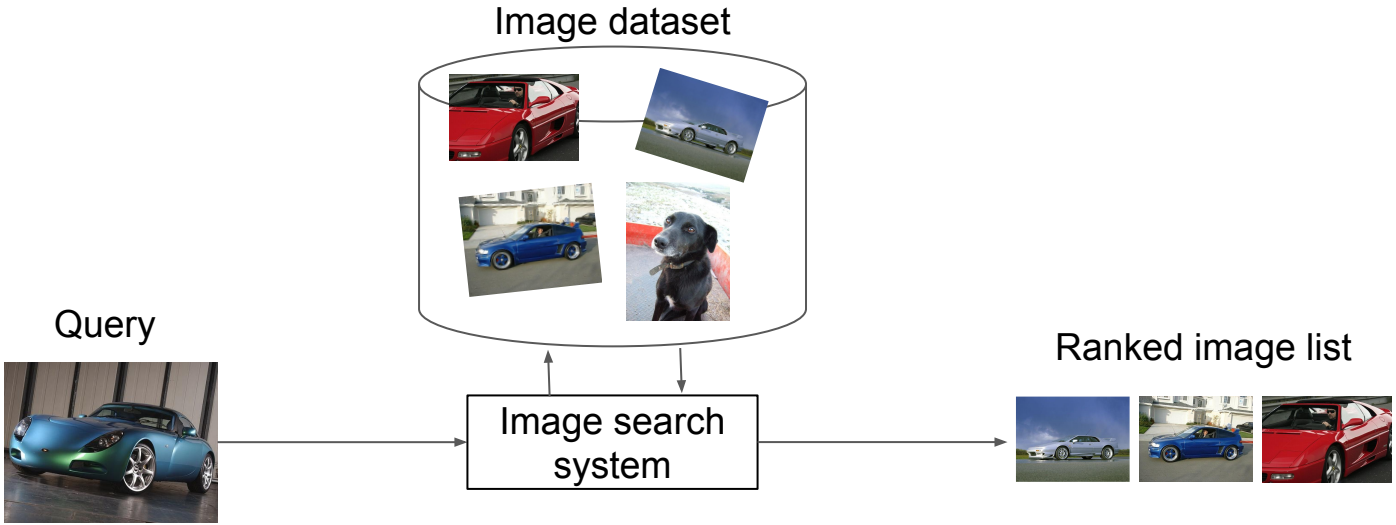

Figure 1.2 Flowchart of the image retrieval system. The objective is to find the most similar images to the query.

Summary W4 II - Query methods

Measure	Acc.
Top5	0.82
Top10	0.80
Top25	0.77

Table 1.2 Our best combination is: **Fisher + FastNN, using FAST+SIFT and K = 64.**

Comments

- The best performance is obtained by **Fisher + FastNN (using FAST, SIFT and K = 64)**.
- Product quantization does not improve results.
- Combined with PQ, inverted files perform better than LSH, but not overcome FastNN.
- Inverted files technique improves computation time.

Summary W4 III

Visual recognition: an example

Figure 1.3 Query obtained. These images were retrieved using Fisher + FastNN, our best configuration. We have chosen a Top-10 performance to show our results.

Observation

In the Fig.1.3 there are several dogs, and one of the dogs has really fluffy hair.

The system give us dogs which has a similar fluffy hair, but also dogs near patches of grass that have a similar texture.

Summary

Hands on ConvNet

Conclusions

Hands on ConvNet W5

- We compared performance of a pretrained AlexNet on the MIT Dataset for classification.
- Two networks: one for 227x227 images and another for 37x37 images.
- The accuracy was **0.71** for AlexNet_small and **0.9281** for normal AlexNet

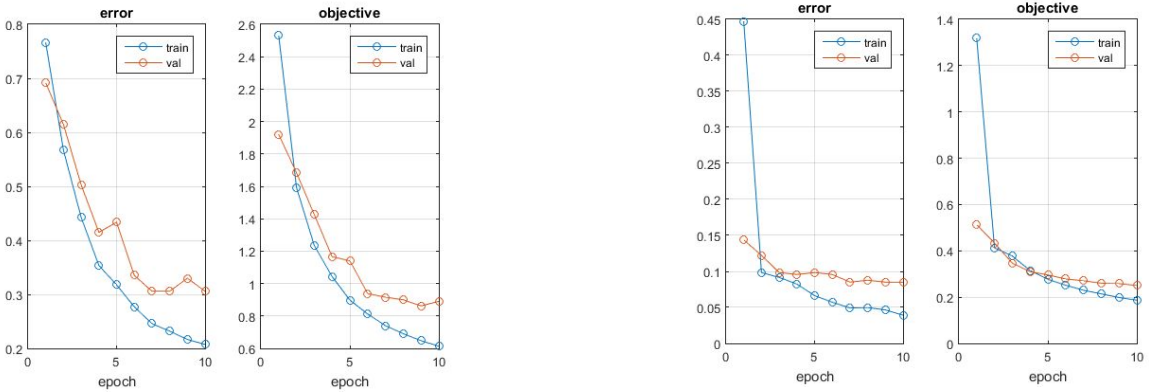

Figure 2.1 Error and objective plots for both AlexNet, (left smaller, right larger) for 10 epochs

Hands on ConvNet W5

- Using the values in several layers of the pretrained AlexNet as an input to our Bag-Of-Words framework.
- Codebooks for the Convolutional layers
- Classification is done with a Linear SVM and the best parameters from week 2.

Layer	Conv1	Conv2	Conv3	Conv4	Conv5	Fc6	Fc7
Accuracy	0.8327	0.8859	0.9083	0.9058	0.6282	0.9504	0.9368

Table 2.1 Shows the accuracy of the BoW framework using as input the values in the layers.

- Best results obtained using the fully-connected layers
- The accuracy increases with the depth of the layers, as the features obtained are more abstract.

Hands on ConvNet W5

Designing and training our own network. (EX3)

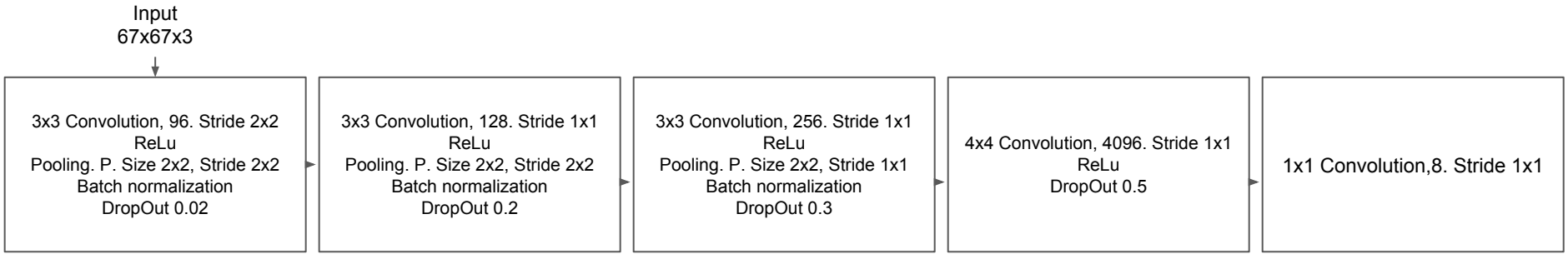

Figure 2.2 Designing our own network. The global parameters are: batchSize = 128, numEpochs = 40, LearningRate = 0.0001 and after 25 steps 0.01, weightDecay = 0.00002 and momentum = 0.9.

Hands on ConvNet W5

Effect of different initializations. DropOut vs Batch normalization approach, where do we put them. Momentum on gradient descent. (EX4)

	Test accuracy
Vanilla	0.7794
DropOut	0.7819
Batch normalization	0.8327
Dropout + Batch normalization	**0.8600**
Bag of words with last fc layer	0.8488

Table 2.2 Shows the difference in terms of accuracy between DropOut, BatchNormalization or combining both. The momentum is set to 0.9, it increases the system performance.

Layer order:

Convolution layer
ReLu
Pooling layer
Batch normalization
DropOut

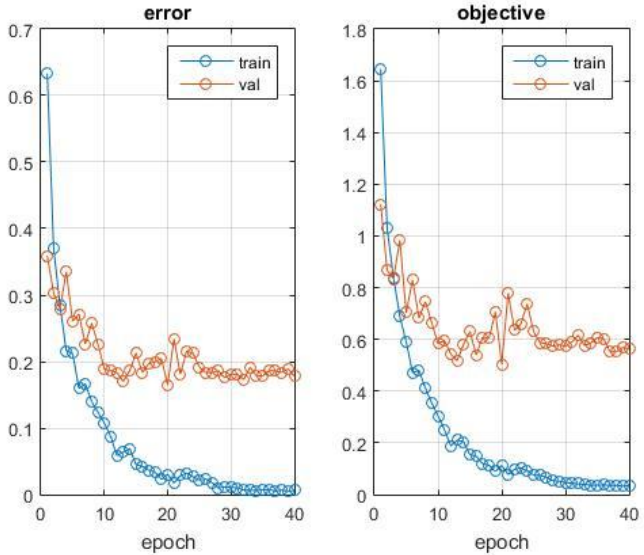

Figure 2.3 Error and objective plots (train/val) with Dropout + Batch normalization settings.

Summary

Hands on ConvNet

Conclusions

Conclusions

- Using output of layers in the network to feed our bag of words framework works better than any of the methods implemented during weeks 1-4
- Batch normalization accelerates deep network training
- Adding Batch normalization + DropOut to the system increases the performance
- Momentum helps to achieve a better accuracy
- It is really hard to avoid overfitting in a deep network
- The overall performance is good, although can be better if we use a stacking technique (i.e. combining the prediction of a deep network with a gradient boosting or random forest prediction)
- Comparing Convolutional Networks and conventional techniques implemented in the firsts weeks, we can conclude that the results in classification using BoW and features from the network are very similar (0.89 vs 0.95).
- CNN gives us encouraging results and has **only been trained and tested during a week**.

VISUAL RECOGNITION

Eric López, Gerard Martí, Sergio Sancho, Adriana Fernández

Master in Computer Vision, Barcelona
UAB, UOC, UPC, UPF

April 2015