Face Recognition with Support Vector Machines (Heisele, Ho, Poggio) Computationally Efficient Face Detection (Romdhani, Torr, Schölkopf, Blake)

Nicolas Höning - M5

July 13, 2006

This report summarizes two papers from the field of Face Recognition. They were presented during the seminar "Information Processing in Machine Learning and Computational Linguistics" at the University Of Osnabrück in the summer term 2006.
Face recognizers became quite efficient through the last years. They are being used in more and more applications like airport passport controls. However, there is still a lot of room for improvement. Both of the papers propose methods for such improvements, each in a distinct step of the process.

Face Recognition With Support Vector Machines
This paper by Heisele, Ho and Poggio from the MIT [Heisele et al, 2001] compares two approaches to face recognition: the global approach and the component-based approach, favouring the latter.

Face recognizers still rely heavily on stable conditions like pose or illumination. For example, the european passport requires that you take your picture in a pose that is exactly determined so that today's face recognizers can deal with it.
Their experiment used a training set of 8,593 grey face images of five subjects (1,383 frontal views). The picture sizes ranged between 80x80 to 130x130 pixels and the pictures were rotated up to 40^o. The test data was 974 pictures with different illumination and background.

It becomes quite clear (when you look at the training data) that the authors focused on the problem of pose invariance to compare the two approaches. Every approach has a face detection stage and a face recognition stage and both make use of Support Vector Machine classifiers (SVM).
Before we introduce the face recognition approaches: in addition to what was introduced in class, it might be of interest that there are two ways of differenciating between several classes (here: persons) with SVMs. The "classic" SVMs perfom a binary decision between two classes. By using a kernel function we

can map non-linear separable data to a high-dimensional "feature" space and thereby make it separable. But what if we have more than two (say: q) classes (faces) ?
The "one-vs-all" approach to SVMs would be to let each SVM separate a single class from all others. In contrast, the "pairwise approach" would be to each SVM separates between a pair of classes. They get organized in a tree structure (each node is a SVM). Since there is (yet) no known performance difference, the one-vs-all approach is favored by the authors, since the pairwise approach requires q^2 SVMs to be trained.

The global approach is the classic way of doing face recognition. It technically means that a single vector represents the whole face image. That way, the global features of the face are mapped. The face detection stage has as output a normalized picture with only the face part in it. It assures brightness and scale invariance. Scale invariance is reached by an algorithm moving a window over the picture to find the face. Grey values were normalized between 0 and 1 to reach brightness invariance.
For the face recognition stage, the authors used one-vs-all SVMs to recognize faces. But those were not very robust against pose change, so they tried a second method: they performed a divise clustering stage (on all pictures of one person) and trained the SVMs on classifying between the clusters. That lead to a view-point specific tree of one person's face images where average faces are the nodes.

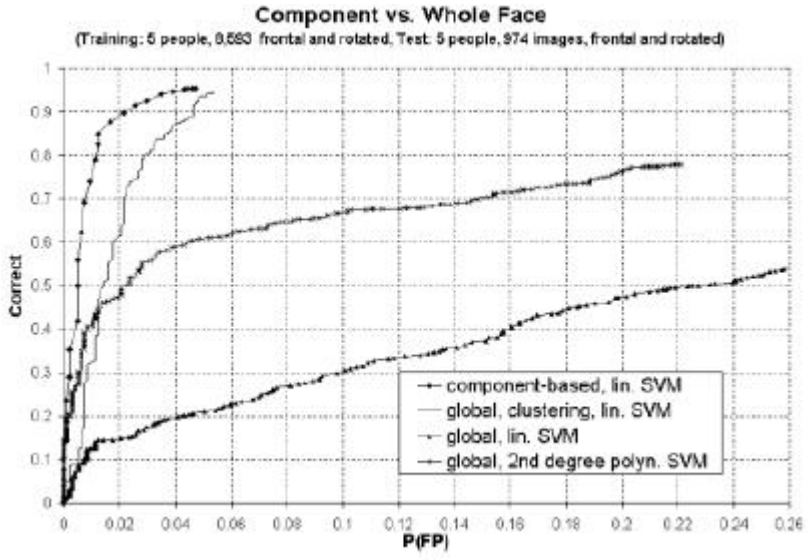

Figure 1: ROC curves when trained and tested on frontal and rotated faces

The component-based approach only learns parts of the faces (components). The idea is that when the face rotates, the changes within those components are small compared to the global features of the whole face.
The face detection stage normalized picture as it was done for the global

approaches. Then a first step detected facial components like eyes, mouth and son on (there were 14 components used). A second step combined the result of the first one. The result of this stage represents a configuration of the components. In the face recognition stage components got normalized (size and grey-values) and then one-vs-all SVMs (one person = one class) classified the pictures.

In the experiments (see Figure 1 on page 2), the authors not only compared the three methods described above (two global, one component-based). They also used non-linear (polynomial) SVMs with the simple global approach to test for improvements due to the power of the SVMs themselves. All other SVMs were linear.
The experimental results show that the component system was always the best method even though it had less powerful SVMs. The Clustering in the global approach led to a significant improvement, even over non-linear SVMs. Rotation is too complicated for linear global classifiers.

Computationally Efficient Face Detection
This paper [Romdhani et al, 2001] by Romdhani, Torr, Schölkopf and Blake deals with the stage of face detection (scanning a lot of patches in a picture for contained faces) and proposes an idea to be far more efficient with that.
Despite of being very useful for classification tasks, SVMs generally are slow classifiers. Their performance is proportional to the number of support vectors (i.e. training examples - in face recognition there are quite a lot needed). The idea of the authors is: can we compute a small set of vectors out of the set of support vectors so that the classification works almost as well? There is already a theory called "Reduced Set Vectors" out there [C.J.C. Burges, 1996] which the authors used to describe an efficient algorithm for face detection.
In SVM tasks, we have a decision surface like this: $\Psi = \sum_{i=1}^{N_x} \alpha_i \Phi x_i$. This term describes Ψ as linear combination of the support vectors. $\Phi(z)$ is the (usually nonlinear) map of the kernel function k, mapping a vector into the feature space.
In our case, we want something like this: $\Psi' = \sum_{i=1}^{N_z} \beta_i \Phi z_i$ - where N_z is much smaller than N_x (a lot less vectors involved) and $\Psi - \Psi'$ (the introduced error) gets minimized.
Now, minimizing $\Psi - \Psi'$ would work as follows: The first reduced vector z would have a span (lineare Hülle) of $\Phi(z)$. We want to minimize the orthogonal projection (the distance) of Ψ to $\Phi(z)$. That problem can be reduced and then transformed w.r.t. k (the kernel function). Generally, when we have Ψ' with m reduced vectors, the (m+1)th vector can be computed from that to yield Ψ''. Ψ'' minimizes the distance to Ψ a bit further (but takes more computing time to classify, of course)
Now the authors are ready to propose their Sequential Reduced Set Machine (SRSM) algorithm:

- 1. start with the first of the reduced set vectors (m=1)

- 2. evaluate the given patch
- 3. if the result is smaller zero, we can reject the patch and stop
  if not, increment m, try step 2 again
- 4. if all of our reduced set vectors have been used and the result is still $>= 0$, try with the whole SVM.

The idea behind it is: There are many, many patches that could be a face (from a pixel to the whole image). The huge majority of them can be thrown out by very few reduced set vectors (that have been calculated in advance). That way, we lose some accuracy, but we gain a lot of speed in the process.

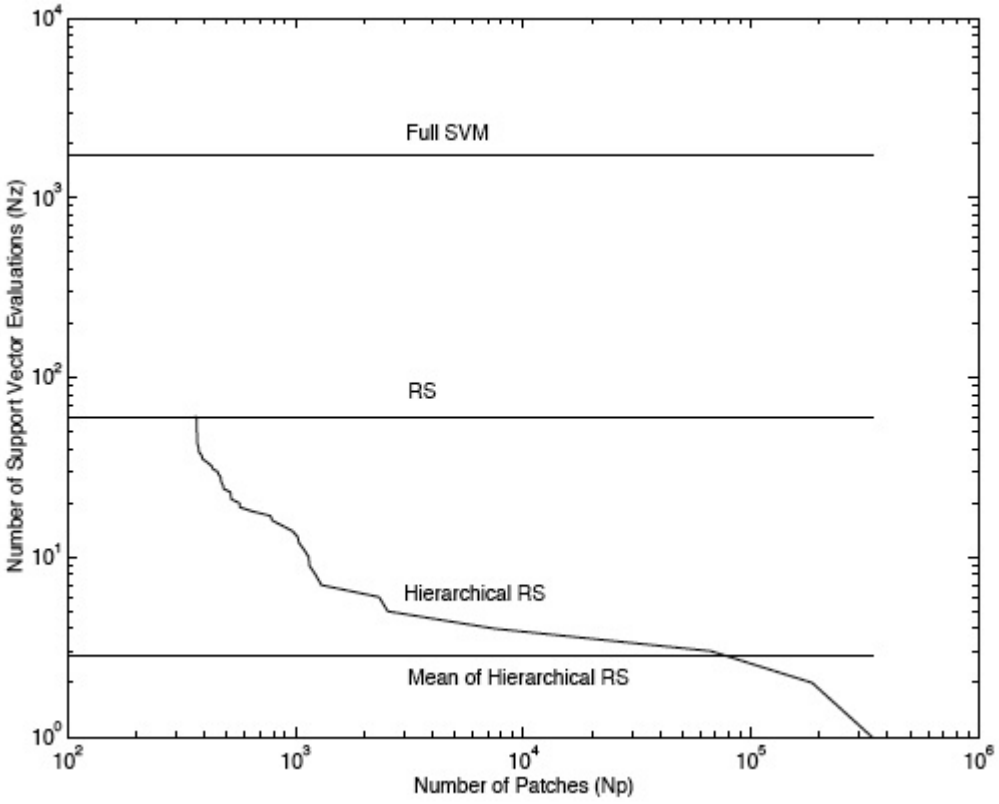

Figure 2: Number of Reduced Set Vectors used per patch for the full SVM (8291 support vectors), Reduced Set SVM and Sequential Reduced Set SVM (both at 100 reduced set vectors)

In their experiment (see Figure 2 on page 4), the authors compared a full SVM approach, a classical Reduced Set (RS) approach (that always uses all 100 Reduced Set Vectors that approximated the full SVMs to a certain degree) and their SRSM algorithm using the same Reduced Set Vectors.
While the full SVM approach was of course by far the slowest method, the researchers also found the speed improvement they were looking for: the SRSM system was 30 times faster than the RSM system. Concerning Accuracy: the

researchers used a test set that was being used by other researchers and did slightly worse (comparison is still hard because they did no preprocessing in this experiment).

To conclude, the authors propose some ideas for further research:

- They used the Gaussian Kernel as distance metric, something else might even be more suitable
- If we already found a face of size x, shouldn't we then prefer patches of sizes similar to x?
- This method can also be applied to other problems than face detection

References

[Heisele et al, 2001] 'Face Recognition With Support Vector Machines: Global versus Component-Based Approach', Center for Biological and Computational Learning Cambridge, 2001

[Romdhani et al, 2001] 'Computationally Efficient Face Detection', Microsoft Research Ltd Cambridge, 2001

[C.J.C. Burges, 1996] 'Simplified support Vector decision rules', *13th Intl. Conference on Machine Learning pages 71-77*, 1996

List of Figures