Feature Selection in Support Vector Machines for Cars and Pedestrian Detection

V. Moser, A. Chebira

An application for cars and pedestrian detection has been developed to run on CSEM's vision-in-package (VIP), the world's smallest vision system. The detector is based on support vector machines (SVMs) and uses Haar features. Only thirty training images are needed to build a robust detector. To be embedded on the VIP, the speed of the detector and the required memory size had to be optimized. Hence, a feature selection mechanism has been designed, which also increases the efficiency of the detector. This application can be used to manage traffic lights at crossroads.

If a system for cars and pedestrian detection is to be deployed at crossroads, it must be cheap, low power, easily trainable and operate in real time. CSEM's VIP-429 fulfils these expectations in terms of cost and power consumption. Many algorithms exist for cars and pedestrian detection. The one we chose for this platform had to be able to make the decision in one second the most. Moreover, it should not require a large amount of labeled data for its training.

Even when used in a simple application, convolutional neural networks[1] and Adaboost-based classifiers[1] need a few hundreds of labeled training data to avoid overfitting. Moreover, the training must be done off-line, meaning that the classifier must be completely retrained if additional data become available. Deep networks[1] can be trained from few images, but require a lot of computing power and are not suitable for small platforms. SVM based algorithms[1] are likely to be less robust than the above-mentioned algorithm for complicated tasks, however they require less training data, and allow new data to be easily integrated into the detector later. The number of support vectors (SVs) and the type of features used can be parameterized, allowing us to tune the computing needs when used on an embedded platform. For our application, Haar features have been chosen for their well-known efficiency in object classification and their ease of computation.

The developed application detects, each second, from a single image, the presence of pedestrians and cars at specific positions. For example, a pedestrian waiting to cross the road or a car stopped at traffic lights. If the scanning zone is 10x10 pixels for each position, the usage of Haar features and SVM does not allow us to have more than 60 SVs for each detector.

The number of features is also a key point in the development of the application. From the observation that not all features are relevant to a specific SV, we decided to select, for each SV, only a subset of features that are specific to that particular SV. For example, as depicted on Figure 1, if the result of a feature on a positive SV (green image on Figure 1) is close to the result of the same feature on the negative SVs (red images on Figure 1), the feature is assumed not to be representative of the SV and can thus be removed.

The SVM training algorithm is configured to use 192 different Haar features for each input and a maximum of 60 SVs. The training database contains 30 labeled images and the testing database contains 100 labeled images. Tests have been performed with a different number of features for each SV. To assess the performance of the detection algorithm, we computed the number of misclassifications, and a "separability" metric. The "separability" metric is computed by dividing the difference of the minimum value of the positive class $min(pos)$ and the maximum value of the negative class $max(neg)$, by the difference between the medians of both classes: $\frac{min(pos)-max(neg)}{med(pos)-med(neg)}$. This metric represents how well the two classes are separated given their SVs. A negative separability implies a non-zero misclassification.

For pedestrian detection, the results are the following:

- 192 features: 3 misclassifications, separability: -1.05
- 60 features: no misclassification, separability: 0.24
- 30 features: 1 misclassification, separability: -0.06

For car detection, the results are the following:

- 192 features: no misclassification, separability: 0.46
- 50 features: no misclassification, separability: 0.64
- 20 features: no misclassification, separability: 0.60

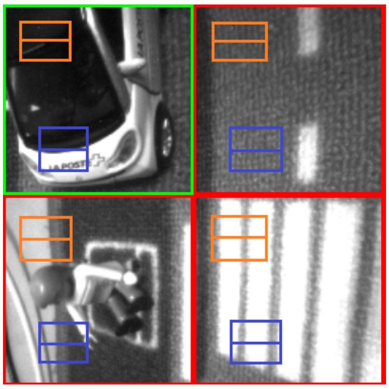

Figure 1: Example of feature selection for car detection: the green image represents positives while the red ones represent negatives. The blue feature is one that is representative of the positive while orange feature is not.

It is interesting to note that in the case where only 20 features are selected by each SV in the car detector, there are 13 features from the 192 initial ones that are never chosen by any of the 60 SVs. Those 13 features do not need to be computed, allowing an improvement in speed.

Due to the memory saved (SVs of size 50 instead of 192), this feature selection allows us to embed the detector on the VIP. In addition, our results show an improvement in the efficiency of the algorithm.

In a future work, the effect of integrating this feature selection during the training of the SVM could be studied.

[1] C. Bishop, "Pattern recognition and machine learning", Springer ISBN 0-387-31073-8, (2006)