Accelerating Computation of Exemplar-SVM by Binary Approximation based on Matrix Decomposition

Takato Kurokawa[1]
kuro@vision.cs.chubu.ac.jp
Yuji Yamauchi[1]
yuu@vision.cs.chubu.ac.jp
Mitsuru Ambai[2]
manbai@d-itlab.co.jp
Takayoshi Yamashita[1]
yamashita@cs.chubu.ac.jp
Hironobu Fujiyoshi[1]
hf@cs.chubu.ac.jp

[1] Machine Perception And
Robotics Group,
Chubu University
Aichi, JPN

[2] Denso IT Laboratory, Inc.
Sibuya cross tower 28th Floor
2-15-1 Shibuya Shibuya-ku
Tokyo, JPN

Abstract

The Exemplar-SVM (E-SVM) is a learning method based on exemplar that uses only one positive sample and a substantial number of negative samples. In the detection stage, it is possible to detect the location of the target object and estimate the attribute by transferring the attribute of the nearest exemplar. The use of E-SVM classifiers leads to very high computational cost because it is necessary to compute the inner products of weight vectors for multiple classifiers and an input feature vector. For accelerating the computation of E-SVM, we propose binary approximation based on matrix decomposition. First, we stack the E-SVM's weight vectors as a matrix. Then, we decompose the matrix into common binary basis vectors and real-valued coefficient vectors for computing the approximated inner products by logical operation. We also introduce early rejection by cascade structure classifier into the proposed method. The evaluation experiments show that the computation time of the proposed method is lower by a factor of 200 than that of the conventional E-SVM.

1 Introduction

For robust object detection in the various appearances, Exemplar SVM [11] trains classifiers such that each instance sample (exemplar) is discriminated from the other samples. The E-SVM classifier has the capability of estimating the attributes of detected objects by transferring the attributes of the nearest exemplar. The E-SVM is also used for 3D object detection from point-cloud data in order to estimate 6D object pose [15]. Kobayashi proposed the optimization of the cost parameters for the highly accurate classification of the E-SVM [9]. Also, Modolo proposed an efficient optimization method for calibration by joint calibration [12]. In the training stage of E-SVM, it is necessary to construct classifiers individually

© 2017. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

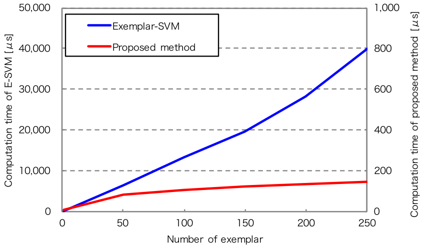
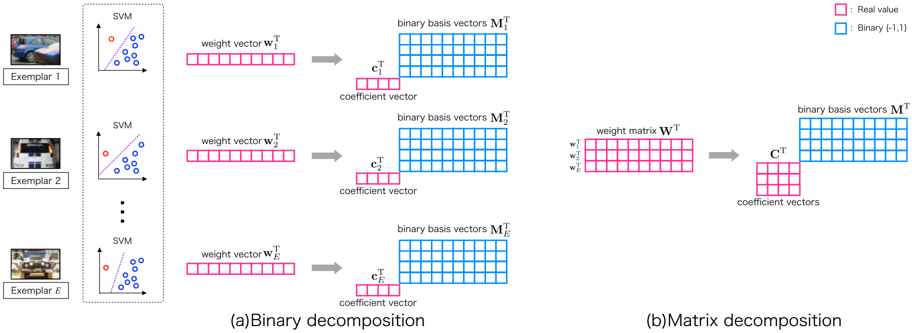


Figure 1: Relationship between the computation time and the number of exemplar.


Real value
Binary {-1,1}
Exemplar 1
Exemplar 2
Exemplar E
SVM
SVM
SVM
weight vector $\mathbf{w}_1^{\mathrm{T}}$
binary basis vectors $\mathbf{M}_1^{\mathrm{T}}$
$\mathbf{c}_1^{\mathrm{T}}$
coefficient vector
weight vector $\mathbf{w}_2^{\mathrm{T}}$
binary basis vectors $\mathbf{M}_2^{\mathrm{T}}$
$\mathbf{c}_2^{\mathrm{T}}$
coefficient vector
weight vector $\mathbf{w}_E^{\mathrm{T}}$
binary basis vectors $\mathbf{M}_E^{\mathrm{T}}$
$\mathbf{c}_E^{\mathrm{T}}$
coefficient vector
weight matrix $\mathbf{W}^{\mathrm{T}}$
$\mathbf{w}_1^{\mathrm{T}}$
$\mathbf{w}_2^{\mathrm{T}}$
$\mathbf{w}_E^{\mathrm{T}}$
binary basis vectors $\mathbf{M}^{\mathrm{T}}$
$\mathbf{C}^{\mathrm{T}}$
coefficient vectors

(a)Binary decomposition

(b)Matrix decomposition

Figure 2: Weight vector decomposition in the E-SVM classifier.

for each positive sample as exemplar. In the detection stage, we apply a raster scanning manner to all trained E-SVM classifiers to compute the classification score. This is why the E-SVM-based object detection needs high computational cost. The computational cost of the E-SVM are increases with increasing number of exemplars as shown in Fig. 1.

Our aim with this study is to accelerate the score computation of the E-SVM. Our work was inspired by recent researches on binary approximation such as binary hashing [7][17], binary descriptors [14][10][2] and vector decomposition [8][1][19]. We stack the weight vectors $\mathbf{w}_i$ $(i = 1, 2, \dots, E)$ of each linear SVM to obtain a weight matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_E)$ and then decompose the matrix into binary basis vectors that are common to all exemplars and scaling coefficient vectors, as shown in Fig. 2(b). If the feature vector is binary, it can be approximated by calculating of the inner product of the binary basis vectors and the real-valued coefficients. The inner product of the binary feature vector and binary basis vectors is computed by using logic operations and bit counting. This makes it possible to calculate the classification score in a very short time. Also, from the inner product with a coefficients that have a small number of elements, the error relative to the classification score can be reduced while maintaining accuracy. Furthermore, introducing early rejection into the classification process reduces the number of calculations and achieves high-speed E-SVM object detection. Our method can significantly reduce the computation time as shown in Fig. 1. The computation time of the proposed method is independent of the number of exemplars in the case of that exemplars have a high correlation to each weight vectors.

1.1 Related works

The methods proposed for increasing classification processing speed include two approaches: reducing the number of calculations and approximation. The first approach includes face detection with a cascade classifier structure, which was proposed by Viola *et al.* [16]. The other approach, approximation, includes binary hashing [5] and binary approximation [8][1][3][19]. Dean *et al.* have proposed a binary hashing-based method that restricts the classifiers to which extracted features are input [5]. Hare *et al.* proposed binary approximation of classifier calculations by decomposing the weight vector of the classifier to binary basis vectors and scaling coefficients [8]. This method approximates the inner product calculations in the classification process to achieve faster processing. Cheng *et al.* proposed high-speed object detection at 300 fps using binary approximation [3]. Efficient methods of vector decomposition for binary and ternary approximation have been proposed [19][1]. Yamauchi *et al.* proposed a method of vector decomposition by exhaustive search [19]. This achieves an even lower error in decomposition than that of Hare *et al.* [8]. Ambai *et al.* introduced a ternary approximation and data-dependent decomposition [1].

These vector decomposition methods described above are also easily applied to E-SVM as shown in Fig. 2(a). Since the vector decomposition is applied to each exemplar, the number of calculations of the inner product with the feature vector increases in proportion to the number of exemplars. The proposed method called matrix decomposition can reduce the number of calculations by using common binary basis vectors for E-SVM classification as shown in Fig. 2(b). Furthermore, we build cascade structure by successive matrix decomposition and apply early rejection by cascade structure in the raster scan manner at the detection stage.

1.2 Contributions

In this paper, we propose a method for decomposing a matrix for accelerating computation of Exemplar-SVM. The proposed method has three key features as summarized below.

1. **Matrix decomposition**: The weight matrix $\mathbf{W}$ of E-SVM as shown in Fig.2(b) is decomposed into a coefficient vectors and a binary basis vectors that are common to all exemplars.

2. **Grouping of highly correlated weight vectors**: Because the binary basis vectors obtained by matrix decomposition are common to all of the exemplars, there is little effect for reducing of permissible error when there is no correlation among the exemplars. Therefore, the highly correlated weight vectors are grouped together to achieve reduction of the permissible error by matrix decomposition. This makes it possible to reduce the number of calculations for the inner product of the binary basis vectors and input feature vector.

3. **Early rejection by cascade structure**: A classifier cascade is constructed by successive matrix decomposition. Introducing a rejection decision at each stage in the cascade reduces the computation for the non-target class in the detection processing.

2 Binary approximation based on vector decomposition

The classification score of linear SVM is computed by

$$F(\mathbf{x}) = \mathrm{sign}[\mathbf{w}^{\mathrm{T}}\mathbf{x} + b], \tag{1}$$

where $\mathbf{x} \in \{-1, 1\}$ is a binarized feature vector such as B-HOG, and $\mathbf{w} \in \mathbb{R}^D$ is a weight vector of the linear support vector machine (SVM) [4]. The B-HOG features are computed by thresholding real valued-HOG features. The papers[1, 18] reported that the B-HOG features have almost same classification performance as that of real-valued HOG features. Generally, the linear classifier classifies the detection object classes and the non-detection object classes by thresholding the output of the classification function $F(\mathbf{x})$. Linear SVM takes a lot of time to classify the classes because it needs to calculate the inner product between the input feature vector and weight vector. For such problems, binary decomposition that decomposes the weight vector into binary basis vectors and coefficient vector to accelerate classification by approximating the inner product has been proposed [8][19][1].

Binary approximation decomposes the weight vector $\mathbf{w}$ of classification function $F(\mathbf{x})$(Eq. (1)) into binary basis vectors $\mathbf{M} = (\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_k) \in \{-1,1\}^{D\times k}$ and coefficient vector $\mathbf{c} = (c_1, c_2, \cdots, c_k) \in \mathbb{R}^k$. The classification function $F(\mathbf{x})$ can be approximated by

$$\begin{aligned} F(\mathbf{x}) &\approx \operatorname{sign}[\mathbf{c}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{x} + b] & (2) \\ &:= \operatorname{sign}\left[\left(\sum_{i=1}^{k} c_i \mathbf{m}_i^{\mathrm{T}}\mathbf{x}\right) + b\right], & (3) \end{aligned}$$

where $\mathbf{x}$ is the binary vector $\mathbf{x} \in \{-1,1\}^D$. k is the basis number. Increasing k improves the approximation of the classification score $F(\mathbf{x})$. Because the calculation for the inner product of binary vectors can be replaced by calculation of the Hamming distance, high-speed calculation is possible.

Training for classifiers by Exemplar-SVM builds a weak classifier for the number of exemplars E. In the detection stage, we compute all scores by inputting the feature vector into each classifier. The exemplar of the classifier that has the highest score is then selected as the final result. If the scores of all of the classifiers are low, the non-target class is output as the final result. When the vector decomposition method is applied to E-SVM classifier, the binary base vectors $\mathbf{M}$ are calculated for each class (Fig. 2(a)). Accordingly, the number of times the inner product of feature vector $\mathbf{x}$ and $\mathbf{Mc}$ is calculated increases by a factor of the number of exemplars in E for E-SVM classification.

3 Proposed Method

If there is high correlation among the exemplar weight vectors $\mathbf{w}$, we can consider that the binary basis vectors, $\mathbf{M}$, are also similar. Therefore, we propose a matrix decomposition method in which the weight vectors of all the classes are represented as a single matrix, $\mathbf{W}$, and that matrix is decomposed into a binary basis vectors $\mathbf{M}$ that are common to all of the exemplars and coefficient vectors $\mathbf{C}$. By using the common binary basis vectors, it is possible to obtain the inner product of the input feature vector $\mathbf{x}$ and $\mathbf{M}$ in a single calculation, so high processing speed can be achieved. A further increase in processing speed is achieved by introducing early rejection based on a cascade E-SVM classifier architecture.

3.1 Matrix decomposition

First, a classifier for each class is trained by SVM to obtain a weight vector $\mathbf{w}$. Next, the weight vectors for all of the exemplars in E are stacked to construct a weight matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_E) \in \mathbb{R}^{D\times E}$ as shown in Fig. 2(b). The constructed weight matrix $\mathbf{W}$ is decomposed into binary basis vectors $\mathbf{M} = (\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_k) \in \{-1,1\}^{D\times k}$ and coefficient vectors $\mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_E) \in \mathbb{R}^{k\times E}$. We design a cost function P for the matrix decomposition. In the decomposition processing, $\mathbf{M}$ and $\mathbf{C}$ are optimized to minimize cost function P. Cost function P is the difference between the weight matrix $\mathbf{W}$ and $\mathbf{MC}$. The decomposition method is explained in detail below.

Matrix decomposition optimizes the binary basis vectors **M** and coefficient vectors **C** to minimize Eq. (4).

$$P = ||\mathbf{W} - \mathbf{M}\mathbf{C}||_F^2. \tag{4}$$

The matrix decomposition algorithm is shown in **Algorithm 1** which mutually optimizes **M** and **C** to minimize Eq. (4). **C** is optimized by applying Eq. (5), which is obtained by the least squares method, fixing **M** constant.

$$\mathbf{C} = (\mathbf{M}^{\mathrm{T}}\mathbf{M})^{-1}(\mathbf{M}^{\mathrm{T}}\mathbf{W}). \tag{5}$$

When **C** is fixed, the value of **M** is determined to minimize P that updated by testing binary pattern. Then, the number of combinations required for one iteration is $D \times 2^k$ because **M** can be optimized at each dimension. This procedure is repeated until there is convergence of the value of the cost function P. In this optimization, **M** and **C** are iterated at L times by changing the initial value of **M**, in order to avoid the dependency on the initial value of M.

Algorithm 1 Matrix decomposition.

```
function matrix_decomposition ($\mathbf{W}$, $L$, $k$)
for $l = 1$ to $L$ do
    Initialize $\mathbf{M}_l$ to a random choice from {-1,1}
    Initialize $\mathbf{C}_l$ to a random real number.
    repeat
        $\mathbf{C} = (\mathbf{M}^{\mathrm{T}}\mathbf{M})^{-1}(\mathbf{M}^{\mathrm{T}}\mathbf{W})$
        for $i = 1$ to $D$ do
            $\mathbf{m}_i = \underset{\mathbf{a} \in \{-1,1\}^k}{\arg\min} ||w_i - \mathbf{a}\mathbf{C}||_F^2$
        end for
        $P = ||\mathbf{W} - \mathbf{M}\mathbf{C}||_F^2$
    until $P$ converges
end for
From $\{\mathbf{M}_l\}$ and $\{\mathbf{C}_l\}$, the binary basis matrix and coefficient vectors for which the cost function $P$ is minimum are taken as $\hat{\mathbf{M}}$ and $\hat{\mathbf{C}}$
return $\hat{\mathbf{M}}$, $\hat{\mathbf{C}}$
```

3.2 Grouping of weight vectors in accordance with correlation

The matrix decomposition is most effective for weight vectors that are close to each other because all of the exemplars are represented by single binary basis vectors **M**. In practice, the vector distances for the weight matrix **W** that is to be decomposed are not necessarily close for all exemplars. Also, the distance of the weight vector **w** before decomposition may be close to the distance after decomposition. Therefore, the vectors that are to be decomposed are grouped in accordance with correlation as shown in Fig. 3 when decomposition is performed. Here, the proposed method is characterized by two different points between dictionary learning[13]. First in our method, the decomposed matrix M is a binary. Second, in our method, coefficient vectors **C** within each group are not sparse. The method for constructing the cascade by grouping and matrix decomposition is described in **Algorithm 2**, where the number of matrix decompositions equals to the number of basis k are performed for each group. A normalized value for the distance between vectors is used in the grouping. The normalization of the distance between vector $\mathbf{w}_i (i = 1, 2, \ldots, E)$ and vector $\mathbf{w}_j (j = 1, 2, \ldots, E)$ is shown in Eq. (6) and Eq. (7).

$$z = \max\{||\mathbf{w}_a||\ a = i, j\}. \tag{6}$$

$$d = -\left(\frac{||\mathbf{w}_i - \mathbf{w}_j||}{z} - 1\right). \tag{7}$$

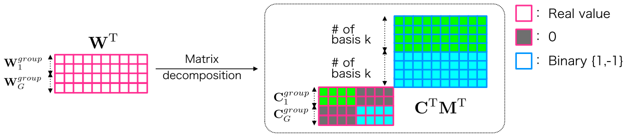


Figure 3: Matrix decomposition with grouping: Colors indicate grouping of elements in coefficient vectors.

Because the distance d is a real value of $-1 \sim 1$, the weight vectors $\mathbf{w}_i$ and $\mathbf{w}_j$ are the same when $d = 1$ and have the opposite directions when $d = -1$. In the grouping process, the threshold value th^{corre} and d are used to group the vectors such that vectors for which $|d| > th^{corre}$ are placed in the same group. The vectors in the same group g are stacked as matrix $\mathbf{W}_g^{group}$. So, weight matrix will be $\mathbf{W} = (\mathbf{W}_1^{group}, \mathbf{W}_2^{group}, \cdots, \mathbf{W}_G^{group}) \in \mathbb{R}^{D \times E}$. Therefore, all numbers of basis K are $G \times k$.

Algorithm 2 Matrix decomposition by grouping weight vectors in accordance with correlation.

```
function matrix_decomposition_with_grouping ($\mathbf{W}$, $L$, $k$)
Calculate the distances between the weight vectors in the weight matrix $\mathbf{W}$ with Eqs. (6) and (7).
Grouping weight vector
$\{\mathbf{W}_1^{group}, \mathbf{W}_2^{group}, \cdots, \mathbf{W}_G^{group}\} \leftarrow \mathbf{W}$
for $g = 1$ to $G$ do
    Optimize $\hat{\mathbf{M}}_g$ and $\hat{\mathbf{C}}_g$ by calling matrix_decomposition ($\mathbf{W}_g$, $L$, $k$)
end for
return $\{\hat{\mathbf{M}}\}$, $\{\hat{\mathbf{C}}\}$
```

3.3 Faster classification by approximated inner product calculation and early rejection

Our method uses the early rejection approach in the classification process to reduce the number of approximate inner product calculations needed for the non-target class. We therefore describe a method for constructing a cascade of successive matrix decompositions and an early rejection method using a cascade of classifiers in the following sections.

At the detection stage, we calculate the inner product of $\mathbf{x}$ and $\mathbf{MC}$ in the detection stage. Here, we must use all basis vectors for approximating the computation of inner product because $\mathbf{M}$ and $\mathbf{C}$ are optimized by the exhaustive algorithm. Therefor, the approximation calculation cannot be aborted by using each basis vectors. The algorithm is shown in **Algorithm** 3., where $\mathbf{M}_1$ and $\mathbf{C}_1$ are calculated by applying the matrix decomposition method to the weight matrix $\mathbf{W}$ in the first stage. In the second and subsequent stages, the matrix decomposition method is applied to the residual matrix $\mathbf{R}$. The residual matrix $\mathbf{R}$ calculate by difference between $\mathbf{M}_{n-1}\mathbf{C}_{n-1}$ of the previous stage and $\mathbf{W}$.

We performed exemplar classification with a cascaded classifier constructed with **Algorithm** 3. The cascaded classifier calculates the approximated classification score by using the sum of the inner product between a binary feature $\mathbf{x}$ and the $\mathbf{MC}$ at each stage as shown in Fig. 4. The approximation error can be reduced by increasing the number of stages N. However, for some of the calculations, it is possible to classify sufficiently determination of negative sample when cascade $n (n = 1, 2, \ldots, N)$ is less than N. For early rejection, we need to terminate the approximated score calculation at less than stage N. Rejection of non-target classes is decided by threshold processing of a score $\mathbf{s}_n$, that is calculated from the approx-

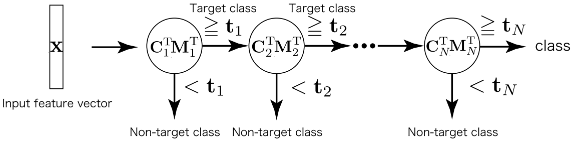


Figure 4: Early rejection in the cascade structure

imate inner product at each stage n. Here, the threshold $\mathbf{t} \in \mathbb{R}^E$ adopts a value that ensures the calculated score at the N stage will not be less than -1 because E-SVM is determined to be negative when the score is less than -1.

Algorithm 3 Algorithm for constructing a cascaded classifier by matrix decomposition method.

```
Require: $\mathbf{W}, k, N, L$
  $\mathbf{R} \leftarrow \mathbf{W}$
  for $n = 1$ to $N$ do
    Optimize $\hat{\mathbf{M}}_n$ and $\hat{\mathbf{C}}_n$ by calling matrix_decomposition_with_grouping ($\mathbf{R}, L, k$)
    $\mathbf{R} \leftarrow \mathbf{R} - \hat{\mathbf{M}}_n \hat{\mathbf{C}}_n$
  end for
  return $\{\hat{\mathbf{M}}\}, \{\hat{\mathbf{C}}\}$
```

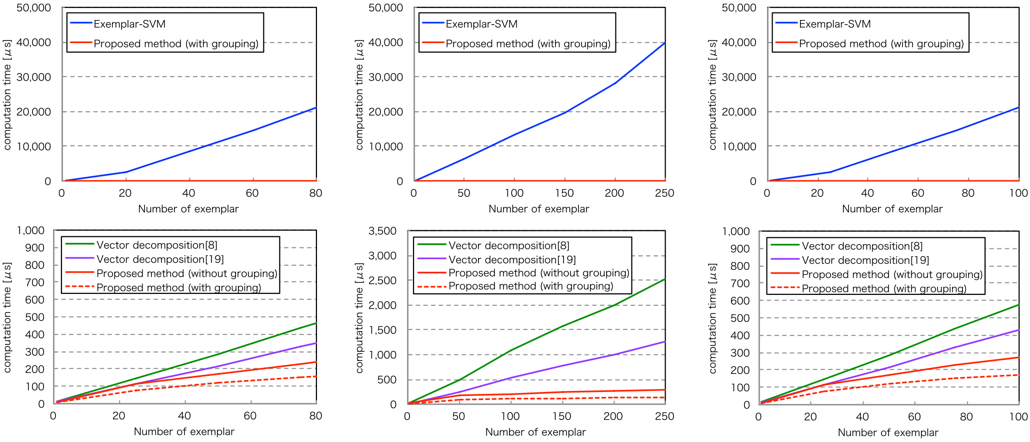
4 Evaluation experiments

To test the effectiveness of the proposed method, we compared it with the conventional vector decomposition methods [8][19].

4.1 PASCAL VOC 2007 evaluation

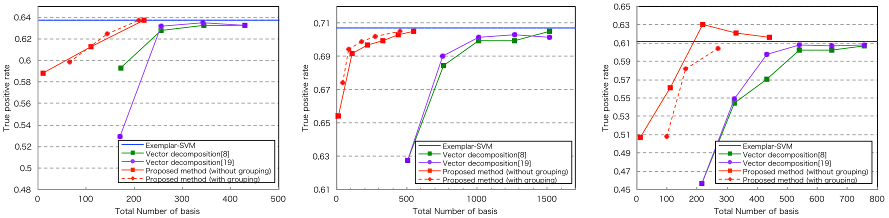
We evaluate computation time and classification accuracy for detection task with dataset in PASCAL VOC 2007[6] in these categories: bicycle, car, and train. For comparing the time required for classification, a personal computer equipped with an Intel Xeon CPU X7542 running at 2.67 GHz. We use the number of basis of the proposed method without grouping that is 11. The basis of the proposed method with grouping is 8.

We first investigate the relationship between the computation time and the number of exemplars. The details of the number of basis vectors and cascade stages we used in the experiments are as follows: bicycle: proposed method(W/O grouping) # of cascade stage=20, # of basis=11. proposed method (with grouping) # of cascade stage=25, # of basis=8. car: proposed method(W/O grouping) # of cascade stage=50, # of basis=11. proposed method (with grouping) # of cascade stage=50, # of basis=8. train: proposed method(W/O grouping) # of cascade stage=40, # of basis=11. proposed method (with grouping) # of cascade stage=32, # of basis=8. Fig. 5 shows the computation time for conventional E-SVM, vector decomposition [8][19], and the proposed method with changing the number of exemplars. The number of basis vectors and cascade stages was selected to maintain the original classification accuracy. We evaluate the computation time when the false positive rate is 0.01 with true positive rate is within 0.05. Fig. 5(a) on the top shows a comparison of E-SVM, and the proposed method, which is matrix decomposition with grouping, when category is bicycle. The proposed method is about 130 times faster than E-SVM when the number of exemplars is 80. Fig. 5(b) on the top shows a comparison of E-SVM and the proposed method, which



(a) bicycle (b) car (c) train

Figure 5: Result of number of exemplars vs. computation time: Figure on the top shows a comparison of E-SVM and the proposed method. Figure on the bottom shows a comparison of the conventional vector decomposition method and proposed method

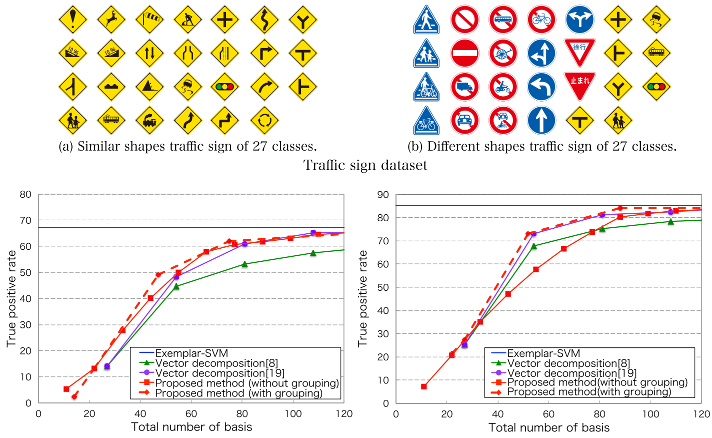


(a) bicycle (number of exemplar is 80) (b) car (number of exemplar is 250) (c) train (number of exemplar is 100)

Figure 6: Relationship between classification accuracy and number of basis: True positive rate were plot when the false positive rate = 0.01.

is matrix decomposition with grouping, when category is car. The proposed method is about 200 times faster than E-SVM when the number of exemplars is 250. Fig. 5(c) on the top shows a comparison of E-SVM, and the proposed method, which is matrix decomposition with grouping, when category is train. The proposed method is about 136 times faster than E-SVM when the number of exemplars is 100. Fig. 5 on bottom shows a comparison of the conventional vector decomposition methods [8][19] and the proposed method. The proposed method is about eight times faster than vector decomposition based on exhaustive algorithm [19]. From Fig. 5, the relationship between computation time and the number of exemplars in E-SVM and the vector decomposition method is linear. Therefore, it takes an immense amount of computation time for classification, when the number of exemplars is large. On the other hand, the processing time of the proposed method is independent of the number of exemplars because the proposed method can save the binary basis vectors. From this, we can say that the proposed method is effective when the number of exemplars is large.

We evaluate for classification accuracy and number of basis. Fig. 6 shows the relationship between classification accuracy and number of basis. From Fig. 6, the proposed method has an improvement in classification accuracy compared to vector decomposition when the number of basis is smaller. This means that the proposed method has a small computation cost and less memory usage for classification computation.



(a) Similar shapes traffic sign of 27 classes. (b) Different shapes traffic sign of 27 classes.

Traffic sign dataset

(c) Dataset1. (d) Dataset2.

Comparison of detection accuracy vs. number of basis: True positive rate were plot when the false positive rate = 0.01.

Figure 7: Data set for toy problem & Comparison of detection accuracy.

4.2 Toy problem evaluation

In this section, we discuss about the following two points: 1. effectiveness of matrix decomposition, and 2. effectiveness of grouping of weight vectors.

We varied the correlation in the matrix that is decomposed and compared the approximation errors of the conventional method and the proposed method (table 1). For a matrix of 1,158 dimensions and 27 classes, the correlation between classes was varied from 0.1 to 0.7. The mean square errors when the number of bases was 10 for each stage of the cascade in the proposed method, and the number of bases per class was 3 are presented in table 1. We can see that a decomposition accuracy that is the same or higher than that for the conventional method is possible for correlation values of 0.34 and above. Because the average correlation of the weight matrix **W** to be decomposed in the evaluation tests is 0.38, the approximation error is considered to be lower than that for the conventional method.

Table 1: Approximation errors.

Method	correlation between weight vectors				
	0.10	0.30	0.34	0.50	0.70
Vector decomposition [8]	7.16	7.10	7.17	7.18	7.92
Vector decomposition [19]	6.02	5.97	5.99	6.00	5.99
Proposed method	11.37	7.89	7.14	4.84	1.97

In this experiment, we investigate the effectiveness of grouping of weight vectors with two datasets. Dataset 1 deals with 27 traffic sign images that have similar shapes as shown in Fig. 7(a). Dataset 2 deals with 27 traffic sign images that include different shapes as shown in Fig. 7(b). For the positive training samples, we used the traffic sign images modified by changes in geometry and illumination, addition of noise, and compositing. For the samples, we used patch images taken from background images. The training was performed with 500 positive samples and 5,000 negative samples. In evaluating the classification accuracy, 5,000 positive samples and 200,000 negative samples from published data sets were used.

We compare the detection accuracy and number of basis of binary basis vectors as shown in Fig. 7(c) and (d). In dataset 1, the both the proposed methods have an improvement of classification accuracy compared to that of vector decomposition when the number of basis is smaller from Fig. 7(c). In dataset 2, the classification accuracy of matrix decomposition without grouping was degraded compared to vector decomposition when the number of basis is smaller from Fig. 7(d). On the other hand, the matrix decomposition with grouping has an improvement in classification accuracy compared to that of vector decomposition and matrix decomposition without grouping, when the number of basis is smaller from Fig. 7(d). From this, the matrix decomposition with grouping is effective when the weight vectors are not closed to each other like in dataset 2.

5 Conclusion

We confirmed that the proposed method is capable of accelerating Exemplar-SVM by matrix decomposition and early rejection by cascaded classifiers. For the problem of E-SVM classification, we have also shown that the proposed method is more effective when the similarity among classes is higher. In future work, we will aim for fast classification to make early decisions for target objects as well. The proposed approach can be extended to enable fast computation of deep convolutional neural network by applying multi-level quantization.

References

[1] Mitsuru Ambai and Ikuro Sato. Spade: Scalar product accelerator by integer decomposition for object detection. In *ECCV*, volume 8693, pages 267–281. Springer, 2014.

[2] Mitsuru Ambai and Yuichi Yoshida. Card: Compact and real-time descriptors. *ICCV*, pages 97–104, 2011.

[3] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014.

[4] C. Cortes and V. Vladimir. Support-Vector Networks. In *Machine Learning*, volume 20, pages 273–297, 1995.

[5] Thomas Dean, Mark A Ruzon, Mark Segal, Jonathon Shlens, Sudheendra Vijayanarasimhan, and Jay Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. *CVPR*, pages 1814–1821, 2013.

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[7] Yunchao Gong and Svetlana Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. *CVPR*, pages 817–824, 2011.

[8] Sam Hare, Amir Saffari, and Philip H.S. Torr. Efficient online structured output learning for keypoint-based object tracking. *CVPR*, pages 1894–1901, 2012.

[9] Takumi Kobayashi. Three viewpoints toward exemplar svm. *CVPR*, pages 2765–2773, 2015.

[10] Stefan Leutenegger, Margarita Chli, and Roland Yves Siegwart. Brisk: Binary robust invariant scalable keypoints. *ICCV*, pages 2548–2555, 2011.

[11] Tomasz Malisiewicz, Abhinav Gupta, and A. Alexei Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, pages 89–96, 2011.

[12] Davide Modolo, Alexander Vezhnevets, Olga Russakovsky, and Vittorio Ferrari. Joint calibration of ensemble of exemplar svms. *CVPR*, pages 3955–3963, 2015.

[13] Ron Rubinstein, Alfred M Bruckstein, and Michael Elad. Dictionaries for sparse representation modeling. *IEEE*, 98(6):1045–1057, 2010.

[14] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. *ICCV*, pages 2564–2571, 2011.

[15] Shuran Song and Jianxiong Xiao. Sliding shapes for 3d object detection in depth images. *ECCV*, pages 634–651, 2014.

[16] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *CVPR*, 1:511–518, 2001.

[17] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. *NIPS*, pages 1753–1760, 2009.

[18] Y. Yamauchi, C. Matsushima, T. Yamashita, and H. Fujiyoshi. Relational HOG Feature with Wild-Card for Object Detection. In *ICCV*, 2011.

[19] Yuji Yamauchi, Mitsuru Ambai, Ikuro Sato, Yuichi Yoshida, Hironobu Fujiyoshi, and Takayoshi Yamashita. Asymmetric feature representation for object recognition in client server system. *ACCV*, 2014.