九州工業大学学術機関リポジトリ
Kyutacar
Kyushu Institute of Technology Academic Repository

Feature Transform Optimization for Pedestrian Classification

著者	Nakashima Yuuki, Tan Joo Kooi
journal or publication title	2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)
year	2018-09-11
URL	http://hdl.handle.net/10228/00007387

doi: http://dx.doi.org/10.23919/SICE.2018.8492633

Feature Transform Optimization for Pedestrian Classification

Yuuki Nakashima[1] and Joo Kooi Tan[2]

[1]Graduate School of Engineering, Department of Mechanical and Control Engineering, Kyushu Institute of Technology, Fukuoka, Japan
(E-mail: nakashima@ss10.cntl.kyutech.ac.jp)
[2]Faculty of Engineering, Department of Mechanical and Control Engineering, Kyushu Institute of Technology , Fukuoka, Japan
(E-mail: etheltan@ss10.cntl.kyutech.ac.jp)

Abstract: In this paper, we propose a FTOP (Feature Transform Optimization Problem) and its solution. We propose a method to optimize both parameters and processing order of feature transform simultaneously, not limited to convolution and pooling included in CNN (Convolutional Neural Network). In order to realize the optimization, we formulate it as a combinatorial optimization problem and solve it by meta-heuristics. The effectiveness of the proposed method is shown by applying the proposed method to pedestrian classification based on a benchmark data set.

Keywords: Image classification, feature transform, combinatorial optimization, meta-heuristics.

1. INTRODUCTION

Object detection for measuring the position and the size of an object from an image has been widely applied and researched by many researchers. Object detection consists of object classification and localization. For object classification at a specific position and size, it is judged whether it is a class given in advance or not. Localization estimates the position and the size of an object. Above all, object classification consists of feature transformation and machine learning. In the feature transformation, values stored in pixels are converted to a feature vector. Machine learning constructs classifiers by inputting feature vectors.

Various feature transformations have been proposed in the research of image classification. For pedestrian detection, HOG (Histograms of Oriented Gradients) [1], LBP (Local Binary Pattern) [2], Covariance [3], Texton [4] and ICF (Integrated Channel Features) [5] have been proposed. In addition, proposed feature transformations have been combined with another feature transformation like the combination of HOG and LBP, HOG and ICF, and expanded like EHOG (Extended HOG). These methods are expected to improve the correct rate of the object classification by increasing the variation of the feature vector obtained from the conventional feature transformation. As described above, feature transformation defined by researchers and methods to combine or extend them have improved the image classification.

On the other hand, CNN [8] achieves semi-automation of feature transformation by dividing tasks such that the structure of neural network is manual, whereas the weight of convolution integral is automatic. As in Texton, it is not necessary for researchers to give weights of convolution in advance. Given this advantage, researchers could concentrate on the architecture of the neural network and various research results were brought. One of them is the method using CNN [7] which performed well in the multi-class problem benchmark called ILSVRC (ImageNet Large Scale Visual Recognition Competition) [9]. Although ILSVRC is a multi-class classification problem, CNN is also used for pedestrian detection, which is a two-class classification [10-11]. The problem with CNN is that the weight of the convolution is automatically optimized, but the network structure of the neural network still needs to be given in advance by the researcher. This method does not consider the optimality on the construction of the feature transformation.

The purpose of this research is to acquire a method for automatically finding the optimal feature transformation in image classification. Therefore, the proposed framework can adjust the parameters of the feature transformation and optimize the processing order automatically, not limited to convolution and pooling included in CNN.

2. PROPOSED METHOD

In the proposed method, machine learning is performed by using optimized feature transformation (See Fig.1). Note that machine learning is used when optimizing the feature transformation. In order to optimize the feature transformation, the feature vector calculation, the machine learning and the evaluation of the trained model are iterated. In the following, we call the flow of feature transformation as FTN (Feature Transformation Network) firstly. We then define the flow of the machine learning process using the FTN. Secondly, the proposed optimization problem is defined by the above definition. Finally, we propose the method for solving a practical problem.

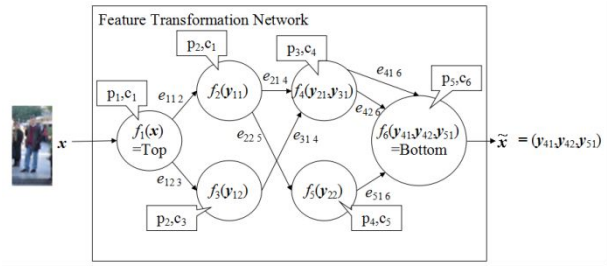
2.1 FTN (Feature Transformation Network)

We define the feature transformation equations, Eqs. (1-3), which describe the FTN.

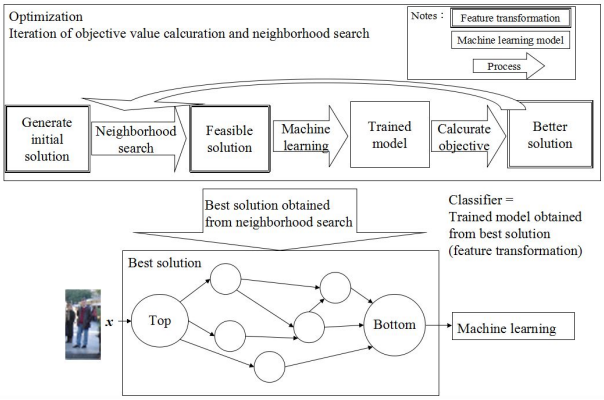
$$\boldsymbol{y}_1=f_1(\boldsymbol{x})=(x_{111},x_{121},\ldots,x_{1\,N_W 1},x_{211},\ldots,x_{N_H N_W 1},x_{112},\ldots,x_{N_H N_W N_C}),$$
$$\boldsymbol{x}\in\mathbf{R}^{N_H\times N_W\times N_C},\ N_H,\ N_W,N_C\in Z. \quad (1)$$

$$\boldsymbol{y}_j=f_j(\boldsymbol{s}_j),\ \boldsymbol{s}_j:=(\{y_{j'n_c}\}),$$
$$\boldsymbol{y}_j\in\mathbf{R}^{M_j},\ M_j\in Z,\ j=2,\ldots,J\text{-}1,j'\neq j,\ j'=1,2,\ldots,J\text{-}1,\ n_c\in Z. \quad (2)$$

$$\tilde{\boldsymbol{x}}=f_J(\boldsymbol{s}_J),\ \tilde{\boldsymbol{x}}\in\mathbf{R}^N,\ N\in Z \quad (3)$$



(a) Example of a FTN



(b) The proposed framework

Fig.1 The proposed method FTN and its optimization

In Eq. (1), let $\boldsymbol{x}$ be a tensor. Here, N_H, N_W and N_C denote height, width and channel, respectively. Here, let f_1 be an enumeration operator. Tensor $\boldsymbol{x}$ is enumerated into a vector by f_1.

In Eq. (2), let y_j be an output by function f_j. Function f_j is not limited to differentiable function with the parameters included in the function. In the case of CNN, convolution, activation function and pooling are applied to f. In the case of HOG, gradient, histogram and normalization are applied to f. A parameter included in a function is the weight of convolution or the number of bins for histogram, for example. The input to f_j is selected by channel n_c included in $y_{j'}$ as the following equation.

$$\boldsymbol{y}_{j'n_c} = (y_{j'11n_c}, y_{j'12n_c}, \ldots, y_{j'1N_W n_c}, y_{j'21n_c}, \ldots, y_{j'N_H N_W n_c}). \quad (4)$$

In Eq. (3), let $\tilde{\boldsymbol{x}}$ be an N-dimensional feature vector.

Here, for the following optimization, Eqs. (1-3) are defined as a graph G. Let G be a FTN. The graph G outputs a feature vector when pixel values are inputted. The graph G is a directed graph defined by Eqs. (5-6).

$$G := (\Phi, V, E). \quad (5)$$

$$\Phi: E \rightarrow V' \times V,\ V' = \{j'n_c\}, (j' \neq J),\ V = \{j = 2, .., J\},\ e_{j'n_c j} \in E. \quad (6)$$

Here, let Φ and E be an incidence function and a set of edges, respectively. Let $e_{j'n_c j}$ be an incidence from the output channel n_c obtained by function j' to function j. In the following section, f_1 and f_J are called 'Top' and 'Bottom', respectively. The f_j has cost c_j and the type of process $p_j \in P$ (P is a set of the type of process). The type of process p has an index $n_p \in N_{proc}$ (N_{proc} is an index set of p). cj and p_j are used for the optimization problem below. An example of the FTN containing above definitions is shown in Fig.1(a).

2.2 Machine learning

A machine learning process is explained in this subsection. Feature vector $\tilde{\boldsymbol{x}}$ is inputted to the training of the machine learning. The output $\boldsymbol{z}_{mv}$, obtained by a machine learning model F trained by $\tilde{\boldsymbol{x}}_{ml}$, $\forall m_l$, is written as follows;

$$\boldsymbol{z}_{mv} = F(\boldsymbol{y}_{mv}),$$
$$\forall m \in M \text{ (Index set of all samples)},$$
$$\forall m_l \in M_l \text{ (Index set of training samples)},$$
$$\forall m_v \in M_v \text{ (Index set of validation samples)},$$
$$M = M_l \cup M_v. \quad (7)$$

2.3 Definition of the optimization problem

In this subsection, a combinational optimization problem is defined using the above equations. The purpose of optimization is to find the optimum graph G from variables f_j. The proposed optimization problem is described in the following;

Objective:

$$\text{Minimize } L(\{\boldsymbol{z}_{mv}\}). \quad (8)$$

Subject to:

$$\forall e \in E'. \quad (9)$$

$$\sum_{j \in J'} c_j \leq C_{max}. \quad (10)$$

$$\sum_{\{j \in \text{index set of having p}\}} c_j \leq C_p, \forall p \in P. \quad (11)$$

$$\dim \tilde{\boldsymbol{x}} \leq D_{max}. \quad (12)$$

Here, let C_{max}, C_p and D_{max} be a total cost calculated from the graph G, total cost per the type of process and the maximum dimension of the feature vector, respectively.

The target of Eq. (8) is the improvement of generalization performance. Because of this, samples used for optimization are divided into training samples and validation samples (See Eq. (7)). Validation samples are used for calculating an objective function. Note that objective function L should be designed depending on application. We explain L in detail in section 3.

Incidences of the graph G are constrained by subject (9). When a computing resource is limited, reducing candidates of incidences is expected to be effective for searching solutions. The case that from p_1 to p_2 is enable and from p_2 to p_1 is unable, we set that $Q_{p1p2}=1$ and $Q_{p2p1}=0$, respectively. This needs not be one-to-one. To speak of extremes, when we set $Q_{pp'}=1$, $\forall p,p' \in P$, we can search with no limitation of incidences. Subject (9) is selective for application.

Subject (10) constrains that total cost of the graph G is less than C_{max}. A cost c_j is set depending on

application. Considering the processing time of feature transformation, c_j is set as the processing time measured in advance. Considering the total number of nodes of function, c_j is set 1, $\forall j \in J$.

Subject (11) constrains that the total cost per p is less than C_p.

Subject (12) constrains that the dimension of the feature vector is less than D_{max}. D_{max} is set depending on the computing resource.

We call objective (8) and subjects (9-12) as FTOP (Feature Transform Optimization Problem).

2.4 Search algorithm

In this subsection, we propose a method for solving the FTOP. A function *f* included in FTOP is not limited to differentiable function. Therefore, the method for solving continuous optimization problem is not applied to FTOP. We search the solution of FTOP using meta-heuristics. Neighborhood operations for the meta-heuristics are the followings;

N.I. Add a node j to G
N.II. Add an edge e to G
N.III. Delete a node j from G
N.IV. Delete an edge e from G
N.V. Swap an existent node j_{old} for a new node j_{new} in G

The best solution G^* is searched by repeating N.I-N.V. In this paper, we use local search for solving the FTOP. The search algorithm is shown in Fig.2.

3. EXPERIMENT

We compared the proposed method to previous methods experimentally using INRIA Person Dataset [14]. In terms of feature transformation, 2 design policies have been existed. One is designing equation and parameters manually (A), and the other is tuning parameters automatically (B).

For comparing (A), we selected the method using

Add edges and nodes operation
INPUT: G
OUTPUT: G
1:While
2: Create j_{add} at random
3: If (10) and (11) calculated with G and j_{add} are infeasible
4: Break;
5: End if
6: (N.I) Add j_{add} to G
7: Select $\boldsymbol{y}_{j_{add}\,n_c}$ from G at random
8: If $e_{j_{add}\,n_c\,j}$ is feasible
9: (N.II) Add $e_{j_{add}n_c j}$ to G , where $e_{j_{add}n_c j}$ join $\boldsymbol{y}_{j_{add}\,n}$ to j_{add}
10: End if
11:End while
12:Return G

Add edges to Bottom operation
INPUT: G
OUTPUT: G
1:While
2: Select $\boldsymbol{y}_{j'n_c}$ from G at random
3: If (9) or (12) is infeasible after joining $\boldsymbol{y}_{j'n_c}$ to Bottom
4: Break
5: End if
6: (N.II) Add $e_{j'n_c\ \mathrm{bottom}}$ to G, where $e_{j'n_c\ \mathrm{bottom}}$ join $\boldsymbol{y}_{j'n_c}$ to Bottom
7:End while
8:Return G

Generate initial solution
INPUT: -
OUTPUT: G
1:Add Top to G
2:G=Add edges and nodes operation(G)
3:Add Bottom to G
4:G=Add edges to Bottom operation(G)
5:Return G

Search Algorithm
INPUT: # of initial solutions, # of local search iteration for convergence, # of (N.V), edge elimination ratio
OUTPUT: G^*
1:For # of initial solutions
2: G=Generate initial solution
3: Calculate objective (8) with G
4: If Objective 3: < Best solution objective
5: Best solution $G^* = G$
6: End if
7:End for
8:G=copy(G^*)
9:While # of local search iteration for convergence > count
10: Generate j_{add} at random
11: Select j from G at random ($j \neq$Top and $j \neq$Bottom)
12: For # of (N.V)
13: If (9), (10), (11) are feasible after swapping j for j_{add}
14: (N.V) Swap j for j_{add}
15: End if
16: End for
17: (N.IV) Delete edges which join to Bottom depending edge elimination ratio
18: (N.III), (N.IV) Delete nodes having no edge recursively
19: G=Add edges and nodes operation(G)
20: G=Add edges to Bottom operation(G)
21: Calculate objective (8) with G
22: If Objective 21: < Best solution objective
23: G^*=copy(G)
24: count=0
25: Else
26: count++
27: G=copy(G^*)
28: End if
29:End while
30:Return G^*

Fig.2 Search algorithm for solving the FTOP

Covariance and ULBP [6]. The method is the best result

on INRIA Person Dataset in the feature transformation manually. Experimental parameters of ULBP and Covariance are set identical with those in [6].

For comparing (B), we selected the method using CNN [13] which adopts the Inception proposed by Google. The method is the best result on ILSVRC2014. We used GoogLeNet [15] which is an implementation of the neural network construction adopting Inception [13]. Input image resolution of the network construction [15] was larger than the experimental image resolution. For this reason, we changed the number of layer from 38 to 15. Iteration of GoogLeNet training was 100 epoch. Methods using CNN are reported with INRIA Person Dataset. The method [11] using CNN has better result than [6]. However, the method [11] predicts image classification and localization simultaneously. Because of this, we could not simply compare the proposed method to the method [11]. Alternatively, we selected the method [13] for the comparison.

Experimental image dataset is shown in Table 1. INRIA Person Dataset includes pedestrian images which are cropped and negative images uncropped. We cropped negative images at random. We set up the following conditions to FTOP. AdaBoost [12] was used for optimization and machine learning for designing a classifier. Parameters of AdaBoost are shown in Table 2.

We selected AUC (Area Under the Curve) for Objective (8) as an evaluation index. The method [6] has reported that AUC is effective for pedestrian classification. $\dim \tilde{\boldsymbol{x}} \leq D_{max}$ x D_{ratio} elements of a feature vector are selected at random when machine learning is run for calculating the objective value. $p \in P$ used in the experiment is shown in Table 3.

We describe how to optimize parameter of f_j. Range and stepping width of parameters included in f_j are set in advance. In the search algorithm, f_j is generated at random within the parameter range. Max filter has a filter size as a parameter, for example. In the case of convolution, the proposed method selects an algorithm and parameters of the algorithm at random. Here, the algorithm includes DoG (Difference of Gaussian), Gabor filter, and so on. The optimization parameters are shown in Table 4. For example, when "Convolution" is selected, "Gaussian" is chosen at random as the algorithm of "Convolution" among Gabor, Differential, Gaussian and DoG filters. The variance value "σ^2" of "Gaussian" is then selected at random.

The plan for searching the best solution efficiently was the followings: Firstly, the optimization problem was divided into sub optimization problems in this experiment. Secondly, solutions obtained from sub optimization problems were connected into one graph in parallel. In this way, the optimization problem could be solved in parallel. Note that there is no guarantee that it is optimal. Settings of sub optimization problems are shown in Table 5. Table 5 shows the followings: The type of process is transform functions. "CvtColor" is selected only once. The number of "Convolution" is not more than 15. The "Top" can be connected to "CvtColor", but "CvtColor" cannot be connected to "Top". The sub optimization problems (see Table 5) were solved three times by changing the random seed. Here, we tried the following two methods in this experiment. One is that only the best solution in sub optimization problems of the same setting is used for machine learning. As the result, 5 (5 settings x 1 best result) graphs are connected into final graph in parallel. This is called multi-start local search, and it prevents itself from falling into a local optimum. The other is that all the best solutions of the optimization problems are used for machine learning [16]. As the result, 15 (5 settings x 3 random seeds) graphs are connected into a final graph in parallel.

The experimental result is shown in Fig. 3 and Table 6. The result is provided by DET (Detection Error Trade-off) curves as shown in Fig. 3 and AUC (Area Under the DET Curve) as shown in Table 6. Example images of experimental results are shown in Fig. 4.

4. DISCUSSION

In Fig. 3, the closer the DET curve is to the origin of the coordinates, the more accurate the method is. In the same way, in Table 6, the smaller the AUC is, the higher the classification accuracy is. The DET in Fig. 3 and the AUC in Table 6 show that the proposed method has higher classification accuracy than previous methods in this experiment. When AdaBoost is used as in this experiment, effective features for classification are selected from many feature candidates by that those features that complement each other in the classification accuracy are selected. Therefore, in the case where, although the number of feature candidates is large, the number of effective features is small, it is considered that, although the long training time is required, the generalization performance is not improved. AUC of the proposed method (x1) is the smallest, although $\dim \tilde{\boldsymbol{x}}$ of the proposed method is the smallest compared with the previous methods. In other words, it is considered

Table 1 INRIA Person Dataset.

# of positive training samples	2416
# of negative training samples	11504
# of positive validation samples	1126
# of negative validation samples	11560
Resolution: width x height [pixels]	64x128

Table 2 INRIA Person Dataset,
(a) for optimization, (b) for classification.

(a)

# of weak classifier	100
type of weak classifier	Decision Tree (DT)
depth of DT	2
# of samples in a leaf	10

(b)

# of weak classifier	1000
type of weak classifier	Decision Tree (DT)
depth of DT	2
# of samples in a leaf	10

Table 3 List of the type of process.

Sign	Description
CvtColor	Convert from RGB to Gray scale, Lab, HSV, Luv, YUV.
Max	Max filter
Convolution	Gabor(σ^2:variance,θ:orientation,γ:aspect ratio,λ:wave length), differential filter, Gaussian(σ^2:variance), DoG(σ_1^2:variance,σ_2^2:variance).
Box	Sum up the brightness value of each pixel in the rectangular region.
*, +, /, -	Four arithmetic operation per elements
Sqrt	The square root of the sum of squares of the values of each element of the input channel.
Norm	Norm: L0, L1, L2, ∞
Orientation	Arctangent of the tensor element of a channel divided by that of the other channel.
Gradient	Gradient magnitude and orientation.
ULBP	ULBP.
Cov	Covarience.

Table 4 Optimization parameters.

# of channels that meet Dmax	10
Cmax	25
$\|M_I\|$	1392
$\|M_V\|$	12528
D_{ratio}	0.1
# of initial solutions	10
# of local search iteration for convergence	100
edge elimination ratio	0.2
c_j $\forall j$	1

Table 5 Parameter settings, (a) CNN base, (b) , (c)ULBP base, (d) Cov base, (e) HOG base.

(a)

the type of process	T:Top, p_1:CvtColor, p_2:Convolution, p_3:Max, B:Bottom					
C_{p1}						1
C_{p2}						15
Q		T	p_1	p_2	p_3	B
	T	0	1	0	0	0
	p_1	0	0	1	0	0
	p_2	0	0	1	1	0
	p_3	0	0	1	0	1
	B	0	0	0	0	0

(b)

the type of process	T:Top, p_1:CvtColor, p_2:Box, p_3:Max, p_4:*, p_5:Sqrt, p_6:Convolution, p_7:Norm, p_8:+, p_9:/, p_{10}:Orientation, p_{11}:Gradient, B:Bottom													
C_{p1}														1
Q		T	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	B
	T	0	0	0	0	0	0	0	0	0	0	0	0	0
	p_1	1	1	0	1	1	1	0	1	1	0	0	0	0
	p_2	0	1	0	0	0	0	0	1	1	0	0	1	0
	p_3	0	1	0	1	0	0	1	0	0	0	0	0	0
	p_4	0	1	1	0	0	0	0	1	0	0	0	0	0
	p_5	0	1	1	1	1	0	0	0	0	0	0	0	0
	p_6	0	0	0	0	0	0	0	1	1	1	0	1	0
	p_7	0	1	0	0	1	0	1	1	0	1	0	0	0
	p_8	0	0	0	0	0	1	1	1	1	0	0	1	0
	p_9	0	1	0	1	1	0	1	0	0	1	1	0	0
	p_{10}	0	0	0	0	0	1	0	0	1	0	1	0	0
	p_{11}	0	1	1	0	0	0	1	0	0	1	0	0	0
	B	0	0	0	1	0	0	0	0	0	0	0	0	0

(c)

the type of process	T:Top, p_1:CvtColor, p_2:Convolution, p_3:ULBP, p_4:Max, B:Bottom						
C_{p1}							1
C_{p2}							4
C_{p3}							1
Q		T	p_1	p_2	p_3	p_4	B
	T	0	0	0	0	0	0
	p_1	1	0	0	0	0	0
	p_2	0	1	1	0	0	0
	p_3	0	1	1	0	0	0
	p_4	0	0	0	0	1	0
	B	0	0	0	1	0	0

(d)

the type of process	T:Top, p_1:CvtColor, p_2:Convolution, p_3:Cov, p_4:Norm, p_5:Orientaion, p_6:Max, B:Bottom							
C_{p1}								1
C_{p3}								1
Q		T	p_1	p_2	p_3	p_4	p_5	p_6	B
	T	0	0	0	0	0	0	0	0
	p_1	1	0	0	0	0	0	0	0
	p_2	0	1	0	0	0	0	0	0
	p_3	0	0	1	0	1	1	0	0
	p_4	0	0	1	0	0	0	0	0
	p_5	0	0	1	0	0	0	0	0
	p_6	0	0	0	1	0	0	0	0
	B	0	0	0	0	0	0	1	0

(e)

the type of process	T:Top, p_1:CvtColor, p_2:HOG, B:Bottom				
C_{p1}					1
C_{p2}					3
Q		T	p_1	p_2	B
	T	0	1	0	0
	p_1	0	0	1	0
	p_2	0	0	0	1
	B	0	0	0	0

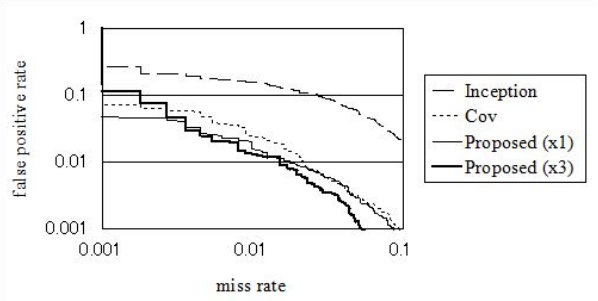


Fig. 3 DET

(a) (b) (c) (d)

Fig. 4 Example images of experimental results: (a) True positive, (b) false negative, (c) true negative, (d) false positive.

Table 6 Experimental results.

	Inception	ULBP+Cov	Proposed(x1)	Proposed(x3)
dim $\tilde{x}$	42368	66262	14733	48155
miss rate[%]	9.1474	3.3748	3.286	2.5755
false positive rate[%]	2.3616	0.5277	0.5882	0.4239
AUC	0.009437	0.0009959	0.0008758	0.0007069
Optimization [min]	4899	-	29809	29809
Training [min]	1183	2590	581	1896
classification [sec]	0.0804	1.154	0.2223	0.6053

that an effective feature for classification that supplement each other is calculated by the proposed method. Unfortunately, it is not a result obtained by fully automatic procedure. This is probably because various features were calculated due to the influence of giving different processing types to setting partial problems. Accordingly, comparing the proposed method (x1) with (x3), (x3) using all the best solutions of sub-optimization problems achieved smaller AUC.

Optimization time of each method is defined by training Neural Network for Inception, and by solving FTOP for the proposed method. (No use for ULBP+Cov.) For the optimization time in Table 6, the proposed method is the largest. This is because the machine learning is executed when calculating the objective value on searching solutions. By using machine learning faster than AdaBoost, or by decreasing $|M_l|$, optimization time can be reduced. On the other hand, the classification time is a calculation time for classifying one image having the size of 64x128 pixels into a certain category. With the classification time in Table 6, the proposed method is the largest. This is because it took longer time to calculate the feature vector than other methods. However, the classification time fluctuates depending on the difference in implementation of feature vector calculation.

4. CONCLUSION

In this paper, we proposed an optimization problem FTOP and its search algorithm to optimize the feature transformation in an object classification problem. In comparison with previous methods, the superiority of the proposed method was confirmed experimentally.

In the experiment, the optimization problem was divided into partial problems from the viewpoint of search time efficiency. In addition, the possibility of incidence of processing types was restricted. However, ideally, it is desirable to obtain a solution with a good objective value without these constraints. The proposed method therefore needs further improvement in order to remove the constraints in solving the FTOP in object recognition.

Further consideration on neighborhood operations and the search algorithm will be another way of refining the solution of the FTOP.

REFERENCES

[1] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection”, *Proc. of Computer Vision and Pattern Recognition* (*CVPR*), pp.886-893, 2005.

[2] T. Ojala, M. Pietikainen, and D. Harwood, “Performance evaluation of texture measures with classification based on Kullback discrimination of distributions”, *Proc. 12th International Conference on Pattern Recognition* (*ICPR*), Vol I, pp.582-585, 1994.

[3] O. Tuzel, F. Porikli and P. Meer, “Pedestrian detection via classification on Riemannian manifolds”, *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(10), pp.1713-1727, 2008.

[4] B. Julesz, “Textons, the elements of texture perception, and their interactions”, *Nature*, 290(5802), pp. 91-97, 1981.

[5] P. Dollár, Z. Tu, P. Perona and S. Belongie, “Integral channel features”, *Proc. of the British Machine Vision Conference*, pp.91.1-91.11, 2009.

[6] S. Paisitkriangkrai, C. Shen and A. Van Den Hengel, “Strengthening the effectiveness of pedestrian detection”, *Proc. of European Conference on Computer Vision*, part IV, pp.546-561, 2014.

[7] LSVRC, http://www.image-net.org/challenges/LSVRC/

[8] A. Krizhevsky, I. Sutskever and G. E. Hinton, “ImageNet classification with deep convolutional neural networks”, *Proc. of NIPS*, pp.1106–1114, 2012.

[9] Q. Z.Wu, Y. LeCun, L. D. Jackel and B. S. Jeng, “On-line recognition of limited vocabulary chinese character using multiple convolutional neural networks”, *Proc. of the 1993 IEEE International Symposium on Circuits and Systems*, vol. 4, pp.2435–2438, 1993.

[10] P. Sermanet, K. Kavukcuoglu, S. Chintala and Y. LeCun, “Pedestrian detection with unsupervised multi-stage feature learning”, *Proc. of Computer Vision and Pattern Recognition*, pp. 3626 - 3633, 2013.

[11] X. Du, M. El-Khamy, J. Lee and L. S. Davis “Fused DNN: A deep neural network fusion approach to fast and robust pedestrian detection”, *Proc. of IEEE Winter Conference on Applications of Computer Vision*, pp.953-961, 2017.

[12] Y. Freund and E. Schapire, “A decision theoretic generalization of on-line learning and an application to boosting”, *Journal of Computer and System Sciences*, Vol. 55, pp.119-139, 1997.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, “Going deeper with convolutions”, *Proc. of Computer Vision and Pattern Recognition* (*CVPR*), pp.1-9, 2015.

[14] INRIA Person Dataset, http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/

[15] Inception ver.3, http://dl.caffe.berkeleyvision.org/bvlc_googlenet.caffemodel

[16]Y. Nakashima, J. K. Tan, “Optimizing Feature Transform Structure for Object Classification”, J*ournal of Biomedical Fuzzy Systems Association* (*BMFSA*), 2018.