GRAMMATICAL EVOLUTION FOR FEATURE EXTRACTION IN LOCAL THRESHOLDING PROBLEM

Go Frendi Gunawan, Sonny Christiano Gosaria, and Agus Zainal Arifin

Dept of Informatics, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember, Jalan Teknik Kimia,Gedung Teknik Informatika Kampus ITS Sukolilo, Surabaya, 60111, Indonesia

E-mail: gofrendiasgard@gmail.com

Abstract

The various lighting intensity in a document image causes diffculty to threshold the image. The conventional statistic approach is not robust to solve such a problem. There should be different threshold value for each part of the image. The threshold value of each image part can be looked as classifcation problem. In such a classifcation problem, it is needed to find the best features. This paper propose a new approach of how to use grammatical evolution to extract those features. In the proposed method, the goodness of each feature is calculated independently. The best features then used for classification task instead of original features. In our experiment, the usage of the new features produce a very good result, since there are only 5 miss-classification of 45 cases.

Keywords: *classification, extract feature, feature, grammatical evolution, local thresholding*

Abstrak

Variasi intensitas pencahayaan pada citra dokumen akan menyebabkan kesulitan dalam menentukan nilai *threshold* dari citra tersebut. Pendekatan statistik konvensional tidak cukup baik dalam memecahkan masalah ini. Dalam hal ini, diperlukan nilai *threshold* yang berbeda-beda untuk setiap bagian citra. Nilai *threshold* dari setiap bagian citra dapat dipandang sebagai masalah klasifikasi. Dalam permasalahan klasifikasi semacam ini, dibutuhkan pencarian fitur-fitur terbaik. Di sini diusulkan sebuah pendekatan baru untuk mengekstrak fitur-fitur tersebut dengan menggunakan *grammatical evolution*. Nilai kebaikan dari masing-masing fitur akan dihitung secara saling lepas. Dalam percobaan yang dilakukan, tampak bahwa penggunaan fitur-fitur baru tersebut menghasilkan hasil yang sangat baik. Hanya ditemukan 5 kesalahan pengklasifikasian dalam 45 kasus.

Kata Kunci: *ekstrak fitur, fitur, grammatical evolution, local thresholding, klasifikasi*

1. Introduction

Chou Et Al (2010) [1] has proposed a local thresholding method to solve the problem caused by various lighting intensity. Their method assembly Support Vector Machine to determine thresholding value of each local segment of the image. In their proposed method, there are four threshold value value, 0, 255, otsu threshold, and minimum otsu neighbor threshold. One of those four threshold value class will be chosen depending on otsu threshold value, minimum neighbor otsu threshold value, standard deviation and means.

Since there are only four threshold values provided for each segment, we can say that the chosen threshold value is the best among those four choices provided. But in our experiment, it is shown that in many case, the best threshold value is none of those four. Therefore, there is needed such a method to search the best threshold value which take care of as much possibility as possible.

The new approach can be done by simply using SVM with 256 output class. But it will also lead to high complexity and computation cost. In this case we should use only use the best features. The best features are not necessarily selected from the original features (otsu threshold value, minimum neighbor otsu threshold value, standard deviation, and means). The best features can also be constructed from the already exists features. There is no formula to generate those best features, but there are some ways to measure the goodness of those features.

Grammatical Evolution was introduced by Conor (1998) [2]. In our new approach, grammatical evolution will be used to generate and evaluate new features independently.

2. Methodology

Grammatical Evolution (GE) is an evolutionary algorithm which is empowered by context-free grammar. This algorithm is derived from Genetics Algorithm (GA). Unlike conventional GA, GE has two representation of individual. The first one is called genotype, which is similar to individual in GA, the second one is phenotype which is formed by combine genotype and grammatical rule.

The most important part of grammatical evolution is how to transform genotype into phenotype. Suppose we have grammar in table I. Every node can evolve based on provided production rule . In GE, we can classify the nodes as terminal (T), start (S), and non-terminal (N). For grammar in table I, <expr> is considered as S (Start Node), since there is no production rule that can produce <expr>. Every time change genotype into phenotype, we should use S (in this case <expr>) as starting node. Terminal set (T) is consists of +, -, *, /, x, y, and 1, since non of those nodes can produce any new node. Once the whole phenotype consists of terminal sets, the evolution process finished. N (Non Terminal Set) is consists of <expr>, <op>, <num> and <var>. Once thegrammar has been defined, we can start produce genotypes and transform them into phenotypes.

TABLE I
GRAMMAR EXAMPLE

Node Notation	Node	Production Rule	Rule Notation
(A)	<expr>	<expr><op><expr>	(A1)
		<num>	(A2)
		<var>	(A3)
(B)	<op>	+	(B1)
		-	(B2)
		*	(B3)
		/	(B4)
(C)	<var>	x	(C1)
		y	(C2)
(D)	<num>	1	(D1)

The transformation can be done by using modulo operation. In table 1, Node <expr> has 4 production rules. Therefore, we should take an integer number from the genotype, and calculate the result of that number mod 4. The result represented the chosen production rule that should replace the initial node. The process continues until the whole phenotype contains of terminal set. Suppose we have 11.01.00.10.01 as genotype and table I as production rule, the evolution process can be shown in table II. The evolution process will result 1+*y* as phenotype. We can then evaluate the goodness of the phenotype, by applying it to the fitness function provided.

Many paper proposed to construct a set of classifier rather than a single feature. Tsoulos et al, 2008 [3], evaluated the fitness of a set of features by using neural network. Gavrilis et al, 2008 [4], proposed a various classifier instead of just neural network. Rivero et al, 2010 [5] proposed the similar thing by using genetics programming instead of grammatical evolution.

Rather than using the classifier itself to define the goodness of feature set, we proposed another approach. In our method, GE is used to generate some individual features. The goodness of each feature is then, evaluated one by one. The classifier itself is not necessary to measure the goodness of each feature. A good feature should be able to represent the data in the form which is linearly separable. There should be no data with the same position hold the different class (overlapped). How to measure this goodness is described in the Fitness Function section.

TABLE II
EVOLUTION PROCESS

Before	Gene	Rule	After Transformation
<expr>	11 -> 3	<expr><op><expr>	<expr><op><expr>
<expr>	01 -> 1	<num>	<num><expr><op>
<num>	-	1	1<op><expr>
<op>	00 -> 0	+	1+<expr>
<expr>	10 -> 2	<var>	1+<var>
<var>	01 -> 1	y	1+y

After several generation, a population of features should be selected. The selection is not only based on the goodness of each feature, but also influenced by correlation of each feature. In our proposed method, Pearson correlation is used to serve this purpose.

The extracted features then used for classification by using linear Support Vector Machine. In general, our methodology is shown in figure 1. The grammar used in grammatical evolution is represented in the Backus Naur Form. In our proposed method, we use a special grammar. Each rule in grammar has such a probability value. The probability value sets from several experiments and assumptions. For example, we give division operator the lowest probability, because division by zero will lead to error. Also, the numeric feature is less important than dynamic feature (e.g: variables). The grammar used for our proposed method is shown in table III.

A simple 2 dimensional data will be used as an example in this section. Suppose we have such a data in figure 1. There are 2 features (*x* and *y*) and 3 classes (represented by circle, triangle and square symbol).

The data is not linearly separable. Actually a better feature can be extracted from original features. Consider that The data in figure 2 can be separated by two circles, we can extract a new better feature based on the formula of circle's radius.

In figure 3, we use $x^2 + y^2$ as a new feature. This feature can separate the data linearly. As we want to measure the goodness of each constructed feature, we need to conduct special fitness function. The fitness value of each features are influenced by some aspects: The count of overlapped point (a point with more than one class), The count of data with different class neighbor, and The complexity of the feature itself.

The first and the second aspects can be measured through looping the data, while the third aspect represented by how much production rule used to generate the respective feature.

$$\frac{1}{(1000*(0.001+0.5*a+0.01*b))} - 10^8 c \quad (1)$$

where a, b, and c are the first, second and third aspect respectively. As we want to deal with numeric dataset, the original images and groundtruths should be represented in some numeric values. For each values in range 0 to 255, we seek the optimum threshold which produce minimum mistake compared to the respective ground truth. The original images and groundtruths are shown in figure 4. Each images in figure 4 are divided into 3*3 segment. Therefore we have 45 total segment. For each segment, otsu threshold value, minimum neighbor otsu threshold value, standard deviation andmeans, and optimum threshold are calculated. This calculation produce a numeric dataset which is used as training and test set.

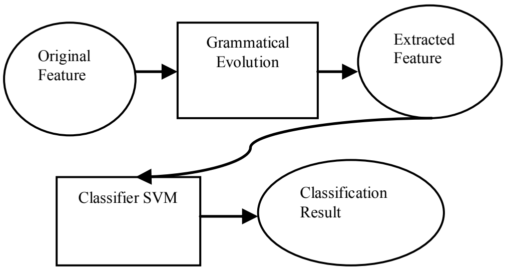

Figure 1. Methodology used in this research.

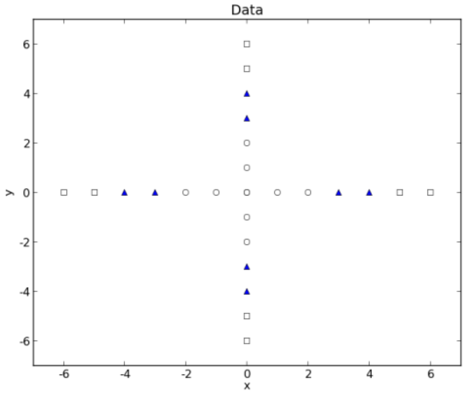

Figure 2. The data represented in original features.

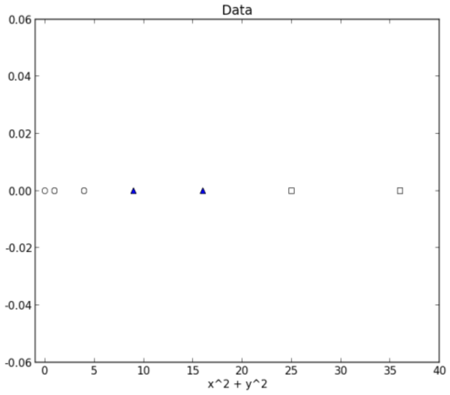

Figure 3. The data represented in newly constructed features.

TABLE III
GRAMMAR FOR FEATURE EXTRACTION

Node	Production Rule	Probability
<expr>	<expr><op><expr>	2
	<var>	8
	<num>	1
<op>	+	2
	-	2
	*	2
	/	2
	**	1
<var>	otsu	2
	stdev	2
	mean	2
	minOtsu	2
<num>	<digit>.<digit>	2
	<digit>	8
<digit>	<digit><digit>	1
	0	1
	1	1
	…	1
	8	1

3. Results and Analysis

Instead of original features (otsu, minOtsu, stdev and means), we find several most important features, as shown in table IV. The possible value of fitnessvalue is vary between 0 and 1.The best features then used for classification task. As explained in groundtruth section, we use 45 row of dataset. From the experiment, it is shown that the features generated are good enough, since there are only 5 miss-classification from 45 cases. This result is a bit better than using the original features itself which produce 6 miss-classification. However, the means square error value of using original features is lower than using extracted features. The complete experiment's result was presented in table V.

TABLE IV
THE BEST FEATURES CONDUCTED BY GE

Features	Fitness Value
(((stdev)*(stdev))-(minOtsu))*(minOtsu)	0.0099
((((stdev)/(otsu))-(mean))-(stdev))*(otsu)	0.0093
((stdev)/((stdev)+((otsu)*(minOtsu))))+(minOtsu)	0.009
(mean)/(otsu)	0.0082
((((((minOtsu)/((minOtsu)-(mean)))+(stdev))-(stdev))*(mean))-(minOtsu))*(minOtsu)	0.0076

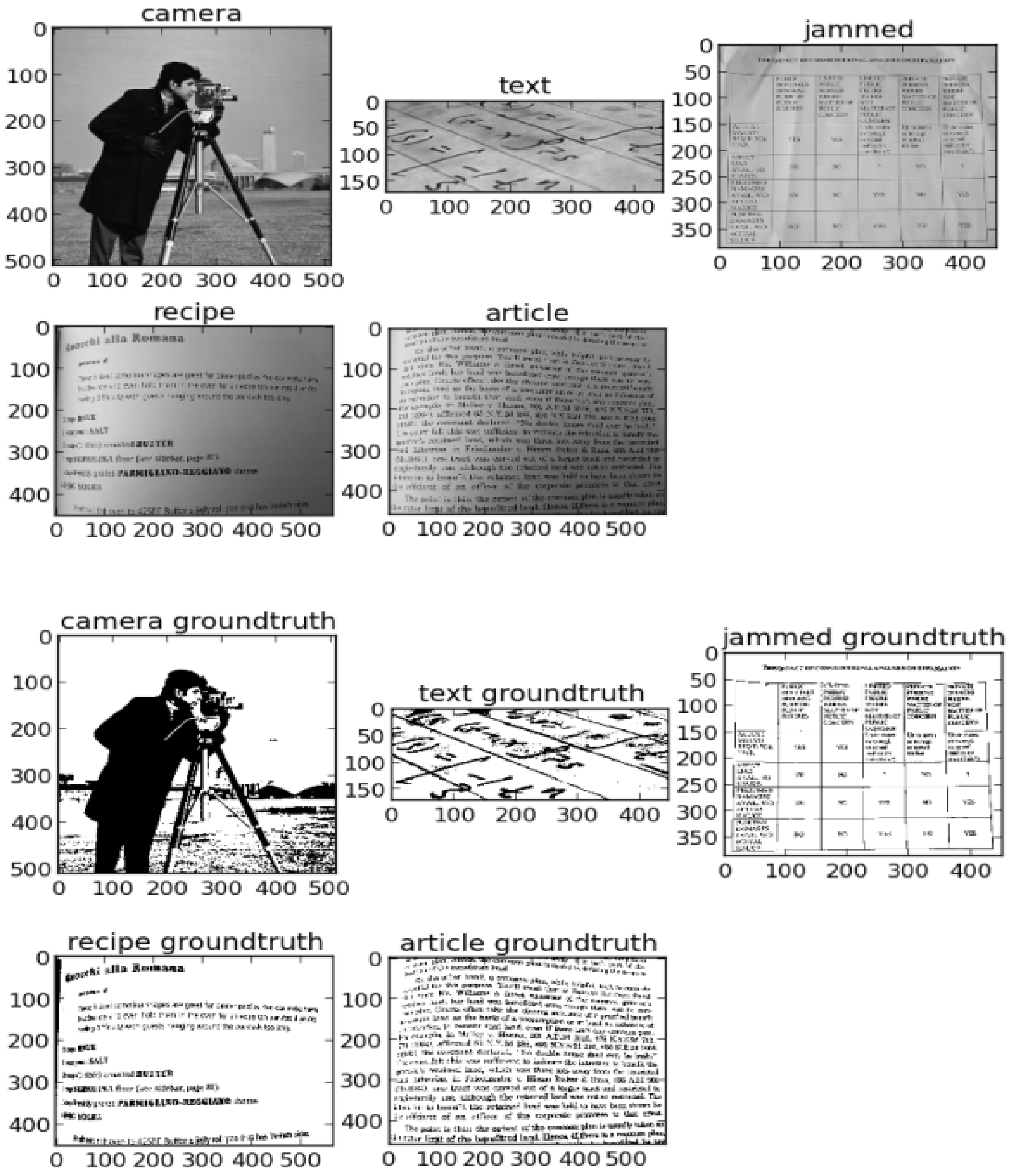

Figure 4. Original images and groundtruths.

TABLE V
THE EXPERIMENT'S RESULTS

Otsu	Stdev	Means	Min Otsu	Groundtruth Threshold	Extracted Feature Threshold	Miss-classification	Original Feature Threshold	Miss-classification
93	37.81	160.52	86	114	113	1	113	1
105	67.80	138.24	86	113	113		113	
160	8.31	161.26	92	0	0		0	
86	72.43	77.12	75	114	113	1	114	
92	74.29	80.18	75	113	113		114	1
112	29.30	151.15	79	113	113		113	
75	54.87	65.72	80	114	114		114	
80	45.99	115.96	75	113	114	1	113	
79	29.83	114.55	80	113	75	1	113	
105	17.81	117.06	97	111	111		111	
97	22.42	117.57	101	100	100		100	
115	15.77	132.06	97	124	124		126	1
101	24.29	126.57	97	110	110		111	1
101	26.51	121.22	97	111	111		111	
104	23.95	132.58	97	126	126		126	
111	23.48	136.35	101	126	126		126	
111	22.68	138.40	101	126	126		126	
122	10.26	140.68	101	126	126		126	
95	18.61	122.67	97	98	98		98	
97	21.94	120.43	95	104	104		106	1
96	20.12	119.52	97	106	106		106	
106	18.79	129.30	95	105	105		105	
108	19.08	129.61	95	105	105		105	
102	17.68	123.86	96	106	106		106	
103	20.97	127.46	106	102	102		102	
109	12.05	131.49	102	100	100		100	
102	11.72	124.12	102	104	104		104	
129	31.75	158.65	80	122	122		122	
96	23.98	130.29	70	106	106		106	
70	15.09	101.01	80	80	80		80	
125	29.09	156.76	63	118	118		118	
80	22.26	120.96	63	91	91		91	
87	12.26	86.68	63	66	66		66	
109	30.34	142.48	63	113	113		113	
63	28.84	96.88	69	75	75		75	
69	11.16	69.72	63	53	53		53	
128	40.98	166.01	78	132	132		132	
87	36.24	121.20	64	109	109		109	
78	28.97	109.10	64	86	86		86	
125	42.85	161.29	62	130	130		130	
78	37.76	106.04	50	82	82		82	
64	28.33	88.08	50	79	79		79	
115	39.40	149.58	62	114	113	1	113	1
62	31.53	88.02	50	73	73		73	
50	22.47	69.50	62	65	65		65	
					Miss-classification	5	Miss-classification	6
					MSE	0.846	MSE	0.077

4. Conclusion

Our proposed method can be considered as new approach for feature extraction. However, since we only use small amount of data, a more intensive study and experiment is needed to validate the robustness of our proposed method Some improvement are still possible. The grammatical evolution itself can be improved by using several optimization (e.g: Advance Grammatical Evolution by Kuroda et al). The feature generated can be mathematically redundant. Therefore, it is needed some symbolic simplication in order to serve the lower complexity.

Pearson Correlation to select the features is known to have several drawback. The usage of better correlation measurement (e.g Kendal Tau) might repair the quality of feature selection. By evaluating every single feature, there is a possibility that the good pair features will be eliminated (e.g: the features that become useful only if used together). A diferent fitness function should be developed in order to detect such a features. For everyone who are interested in develop a new research based on researchers proposed method, the source code in python-language is available online at:https://github.com/goFrendiAsgard/kakera-py/tree/bcdd29170d59e423e57544f2bdfcb7a3cc458312.

Acknowledgement

We would like to thanks Mr. Agus Zainal Arifin for the discussion. The discussion give us a fresh inspiration to conduct the fitness function used in this paper. We would also like to thanks Mrs. Zusana Pudyastuti for checking some grammatical error on this paper.

References

[1] C.H. Chou, W.H. Lin, & F. Chang, "Paper Title," *Pattern Recognition,* vol. 43, pp. 1518-1530, 2010.

[2] M. O'Neill & C. Ryan, "Grammatical Evolution: A Steady Approach," *In Proceeding of the Second International Workshop on Frontiers in Evolutionary Algorithms*, pp. 419-423, 1998.

[3] I.G. Tsoulos, D. Gavrilis, & E. Glavas, "Neural Network Construction and Training Using Grammatical Evolution," *Neurocomputing*, vol. 72, pp. 269-277, 2008.

[4] D. Gavrilis, I.G. Tsoulos, & E. Dermata, "Selecting and Constructing Features Using Grammatical Evolution," *Pattern Recognition Letters*, vol. 29, pp. 1358-1365, 2008.

[5] D. Rivero, J. Dorado, J. Rabunal, & A. Pazos, "Generation and Simplification of Artificial Neural Network by Means of Genetic Programing," *Neurocomputing*, vol. 73, pp. 3200-322, 2010.