Hybrid Saliency-SVM Method Implementation for Automatic Data Training Selection in Image Segmentation

Rully Soelaiman1,a, Chastine Fatichah1 and Aisha Yuliandari1

1 Department of Informatics Engineering, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia 60111

a rully@if.its.ac.id

Abstract. Image segmentation is one of the most important step in computer vision and image processing, which later will be used in image retrieval, object identifying and data classification. Image segmentation can be seen as classification problem, namely by marking each pixels according to certain characteristics. Support Vector Machine (SVM) is a classification method included in supervised learning. Supervised learning is a method which requires training and testing. Training sample used in training process isn’t always exist in few cases, especially in the image segmentation case. This particular research implemented SVM-based method which is Saliency-SVM for automatic data training selection in image segmentation. This method generates data training using SVM-based visual saliency detection where there is pre-segmentation step and trimap formation based on saliency information from visual saliency detection, HSV color space quantitation, histogram analysis and local homogeneity threshold. Data training produced is pixel belonging to positive (object) and negative (background). The step before segmentation done with SVM is feature extraction to create input vector in SVM. Object segmentation on the image is done by SVM based on SVM Trained Model. Test result from Saliency-SVM for this image segmentation has average accuracy value up to 94.84% compared to the image ground truth.

1. Introduction
Image segmentation is one of the most important step in computer vision and image processing, which later will be used in image retrieval, object identifying and data classification. Image segmentation can be seen as classification problem, namely by marking each pixels according to certain characteristics. Several classification methods have already successfully performed image processing. Support Vector Machine (SVM) is one of the method among others. Support Vector Machine (SVM) is a classification method included in supervised learning. Supervised learning is a method which requires training and testing. SVM used a few training sample which are marked for classification learning, meanwhile marked training sample doesn’t always exist in few cases, especially in image segmentation case [2]. In that case, this research will implement new approach which combine Visual Saliency Detection and SVM classifier for image segmentation. This method produced training data using SVM-based visual saliency where there are pre-segmentation and trimap formation steps based on saliency from visual saliency detection, HSV color space quantitation, histogram analysis and local homogeneity threshold. After training data formed, we extract the feature and classification with SVM.
The next part of this paper composed of Research Methodology in part 2, performance discussion from method for few scenarios in Testing and Analysis in part 3, and Conclusion in part 4.

2. Research Methods
This part will discuss about methods used in this research. This research composed of 3 main parts; training data selection which covers pre-segmentation process and trimap formation, HSV color space quantitation, histogram analysis and pixel training selection with local homogeneity threshold. The next part is features extraction which are color, saliency, spatial and texture features. The last part is pixel classification using SVM.

![General illustration of image segmentation using Saliency-SVM](image.png)

**Figure 1.** General illustration of image segmentation using Saliency-SVM

2.1. Training Data Selection
This particular pixel training selection with SVM-based visual saliency covers pre-segmentation process and trimap formation, where estimating object location and background based on saliency map resulted from visual saliency detection on input image. Trimap produced will be further identified with histogram analysis for dominant color selection which have been quantitized before in HSV color space, and later selected in pixel training with local homogeneity threshold. Those identifications were done to determine whether those pixels are within positive training set or negative training set.

2.1.1. Pre-segmentation and trimap formation
Pre-segmentation and trimap formation were done based on saliency information generated from visual saliency detection with spectral residual approach [3]. Visual saliency detection were used for generating saliency map, which then used for approximate object from an image. Spectral residual analyzes spectrum log from image, saliency map formed based on spectral residual from amplitude spectrum from fourier transform on the image. Spectrum fase on fourier transform is the key to calculate location from the most prominent part of it. Use the following equation to form saliency map with spectral residual:

\[
A(f) = \text{absolute}(F(l(x))) \tag{1}
\]

\[
P(f) = \text{angle}(F(l(x))) \tag{2}
\]

\[
L(f) = \log(A(f)) \tag{3}
\]

\[
R(f) = L(f) - h_n(f) * L(f) \tag{4}
\]

\[
S(f) = g(x) * ||F^{-1}[\exp R(f) + i * P(f)]||^2 \tag{5}
\]
Where $F^{-1}$ is Inverse Fourier transform, $P(f)$ is phase spectrum and $g(x)$ is 2D Gaussian Filter with $\sigma = 8$ for a better visual effect. To identify location of an object, do binary image with otsu method:

$$OM(x,y) = \begin{cases} 0 & \text{if } SM(x,y) < t \\ 1 & \text{if } SM(x,y) \geq t \end{cases} \quad (6)$$

Threshold from this binerization is determined by discrimination criteria $(\sigma_B^2, \sigma_W^2)$ from 2 class (salient object and background). For white part (object) valued 1 is $R_o$ whereas black pixel valued 0 is $R_b$.

To eliminate unnecessary pixel around boundary $R_o$, do morphology operation with:

$$M_o = R_o \ominus E_r \quad (7)$$

Boundary $R_o$ depreciated with erosion operation. $M_o$ is Mask Object and $\ominus E_r$ is erosion operator which shows depreciation $R_o$ on pixel $r_e$.

$R_o$ expanded with dilated operation. $M_b$ is Mask Background and $\oplus D_r$ dilation operator which shows $R_o$ expansion towards pixel $r_d$. Dilation result from $R_o$ minus by $R_o$ and combine the result with $R_b$.

$$M_b = ((R_o \oplus D_{r_d}) - R_o) \cup R_b \quad (8)$$

Structure element used in this morphology operation is square, with width of 70 pixels. $M_o$ and $M_b$ resulted from morphology operation serve as standard location in trimap formation. Where Trimap $(T_o, T_r, \text{dan } T_b)$ inside RGB color space no longer in binary. $T_o$ is a prominent part in the pre-segmentation process, $T_b$ is background and $T_r$ are vague pixels from remaining area of $T_o$ and $T_b$.

Figure 2 shows illustration of pre-segmentation process and trimap formation.

2.1.2. HSV color space quantitation

$T_o$ and $T_b$ in RGB color space converted to HSV color space and quantitated to generate 1-dimention vector. HSV (Hue:[0,360], Saturation:[0,1], and Value:[0,1]) is able to emphasize on human visual perception, which proved in having better result for image segmentation from RGB color space. To decrease computation complexity in dominant color determination $T_o$ and $T_b$ in histogram analysis, do HSV color space quantitation on $T_o$ and $T_b$.
Figure 3. Hue Channel Quantitation Scheme [2]

Figure 3 shows scheme result quantitated into 7 non-uniform bins represented from 0-6 and each shows main color with Equation 9.

\[
H = \begin{cases} 
0 & \text{if } h \in (342,16] \\
1 & \text{if } h \in (16,42] \\
2 & \text{if } h \in (42,64] \\
3 & \text{if } h \in (64,152] \\
4 & \text{if } h \in (152,195] \\
5 & \text{if } h \in (195,280] \\
6 & \text{if } h \in (280,342] 
\end{cases}
\tag{9}
\]

Figure 4. Quantitation scheme S&V channel [2]

\[
S = \begin{cases} 
0 & \text{if } s \in [0,0.3) \\
1 & \text{if } s \in [0.3,0.8) \\
2 & \text{if } s \in [0.8,1) 
\end{cases}
\tag{10}
\]

\[
V = \begin{cases} 
0 & \text{if } v \in [0,0.2) \\
1 & \text{if } v \in [0.2,0.7) \\
2 & \text{if } v \in [0.7,1) 
\end{cases}
\tag{11}
\]

When the value of S is large enough, for instance S better than 0.8, part III from figure 4 can be considered as pure red. When the value of V is small enough, for instance V smaller than 0.2 part I from figure 4 can be considered as pure black area. So, 3 non-uniform bins expressed as 0 until 2 are sufficient to represent value and saturation information.

According to the scheme above, vector 1-dimension feature built by 3 channel values as in Equation 12.

\[
L = Q_S Q_v H + Q_v S + V
\tag{12}
\]

As the often used quantization method, quantization coefficient determined as \(Q_S=Q_v=3\) then L can be calculated with Equation 13.
Thus, the 3 channels (hue, saturation and value) can be distributed in 1-dimension vector L and $\mathbf{L} \in \{0,1,\ldots,62\}$. Since quantization resulted in only 63 bins, then computation complexity will decrease significantly. Considering non-uniform characters on these 3 channels, quantization result is more similar to human visual method [2].

2.1.3. Histogram Analysis

Histogram analysis is the step for choosing histogram peaks in $T_o$ and $T_b$ that has been quantized in HSV color space.

The peak in histogram can be information from image’s color. For image with color, dominant color can be identified with peaks in global histogram. Peak selection in histogram was done to decrease color amount in an image which has been quantized in HSV color space and ignoring colors which appear not as often. Those colors are then dominant in $T_o$ and $T_b$. Assumption from [2] which has been tested in 1000 image, that no more than 3 or 4 significant dominant colors describe part $T_o$ and $T_b$. Steps in histogram analysis shown below:

1. Calculate global histogram from $T_o$ that has been quantized in HSV color space.

$$H^o = \frac{\text{Num}(f(x,y)=l_i)}{\text{Num}(T_o)}, (x, y) \in T_o, l_i \in \{0,1,\ldots,62\}$$

Where $\text{Num}(T_o) =$ pixel count in $T_o$. $\text{Num}(f(x,y)) =$ Pixel count with color level $l_i$ in $T_o$.

2. Identify all peaks. $P_{ho}: P_{l_1}, P_{l_2},\ldots,P_{l_{k'}}$. $l_i$ is index from peak $i$, and $l_1<l_2<\ldots<l_k$

3. Calculate maximum and minimum peak from $H^o$. Erase lowest peak based on $T_{ho}$. Calculate $T_{ho}$ with Equation 15 and a new peak will be formed $P_{ho} = P_{l_1}, P_{l_2},\ldots,P_{l_{k'}}$.

$$\mu_m = \frac{\mu_{max}+\mu_{min}}{2}$$
$$\sigma_m = \sqrt{\frac{\sum_{i=1}^{k} (P_{l_i}-\mu_{m})^2}{k}}$$
$$T_{ho} = \mu_m - \sigma_m$$

4. Erase few peaks according to threshold width $T_{w_0}$. $T_{w_0}=20$, assumed there are no more than 3 or 4 dominant colors on object $T_o$. For two adjacent peaks $P_{l_i}$ and $P_{l_j}$, if $(l_j-l_i)<T_{w_0}$ then save highest valued peak and remove other peaks from $P_{ho}$.

5. The result is peak sequence $P_{ho}$ and dominant color from $T_o$ determined as $C_o : l_1,l_2,\ldots,l_n$. Do $T_b$ as well with steps above. After dominant color chosen, then SVM training dataset consists of $T_{Sp}$ (positive training set) and $T_{Sn}$ (negative training set) can be formed with Equation 16.

$$T_{Sp} = \{(x,y)|f(x,y) = i, (x,y) \in T_o, i \in C_o\}$$
$$T_{Sn} = \{(x,y)|f(x,y) = i, (x,y) \in T_b, i \in C_b\}$$

2.1.4. Local Homogeneity Threshold

Generally, the amount of SVM training dataset is too large to be used as training directly. Neighborhood homogeneity threshold is used to choose pixels on $T_{Sp}$ and $T_{Sn}$ as training sample in SVM training. For pixel $p(i,j)$ in training set $T_{Sp}$, $T_{Sn}$ local homogeneity on $n \times n$ neighborhood calculated with Equation 17. $M_p$ is local homogeneity with $n \times n = 3$, $d(p,q)$ is Euclidean color

$$L = 9H + 3S + V$$

(13)
distance between pixels $p$ and $q$ on quantized HSV color space. If $M_p \leq T_{lh}$ then that pixel belongs to $TS_p$ and $TS_n$. The value of $T_{lh}$ is assigned by user, the better value of $T_{lh}$ inputted, the value of $TS_p$ and $TS_n$ generated will be more precise.

$$M_p = D_p^{n\times n} = \sum_{q \in n_p^{n\times n}} d(p, q)$$

(17)

2.2. Feature Extraction

The feature extraction is done by taking color value of r, g, and b on image in RGB color space, intensity (i) on an image quantized in HSV color space, saliency value on saliency map, spatial value (x,y) and texture value which is local energy and local gradient value [4] from an image with steerable filter applied.

2.3. Pixel Classification with Support Vector Machine

After feature value from each pixel extracted, to generate image segmentation, those values will be classified using Support Vector Machine. Data normalization, SVM training and SVM testing will be done in this stage.

2.3.1. Data Normalization

Normalization is a preprocessing technique to do data scaling and standardization. Normalization done here is by scale normalization, which scale data in a certain range where the common range is 0-1. This is done to avoid data range from being too distant.

2.3.2. Classification with SVM

SVM falls under supervised learning. Supervised learning is a method requiring training and testing. SVM method attempt to seek a separator function with decent generalization. Generalization is an ability of hypothesis to classify a group of training in the middle of two classes. Searching equivalent optimal hyperplane by maximizing margin or distance between boundary field on both class [5]. In the plane formed from data pattern, SVM will form hyperplane separating 2 classes. The simplest form of SVM is linear SVM. As a reference, training set $D$ can be denoted with Equation 18.

$$D = \{ (x_i, y_i) | x_i \in R^p, y_i \in \{-1,1\} \}_{i=1}^n$$

(18)

The value of $y_i \in \{-1,1\}$ represents class index or label valued 1 or -1 from dataset $x_i$. The best hyperplane separates data with $y_i = 1$ and $y_i = -1$. The hyperplane can be denoted in Equation 19.

$$w \cdot x - b = 0$$

(19)

Data point $x_i$ included in class 1 can be denoted as point which fulfills Equation 20 and data point $x_i$ included in class -1 can be denoted with Equation 21.

$$w \cdot x - b \geq 1, y_i = 1$$

(20)

$$w \cdot x - b \leq 1, y_i = -1$$

(21)

Ideal hyperplane has maximum margin in separating data classes. In accordance with application of distance calculation formula between point and line, distance between nearest data and central boundary of hyperplane is $\frac{1}{||w||}$. To maximize margin, value of $||w||$ needs to be minimized. Minimization of $||w||$
value means completing quadratic programming problem. This problem will search for minimum point from Equation 22 with regard to boundary from Equation 23.

\[
f: \frac{1}{2} ||w||^2
\]

(22)

\[y_i(x_i \cdot w + b) - 1 \geq 0, \forall i
\]

(23)

This quadratic programming problem can be solved with various computation techniques. Lagrange Multiplier is one of them. Lagrange Multiplier used variable \(\alpha\) to simplify calculation with the value of it being 0 or positive for Equation 24.

\[
L(w, b, \alpha) = \frac{1}{2} ||w||^2 - \sum_{i=1}^{i} \alpha_i(y_i(x_i \cdot w + b) - 1)
\]

(24)

Optimal value from Equation 26 can be calculated by minimizing L towards w and b, and maximizing L towards \(\alpha_i\). Equation 24 can also be modified so it will only contain \(\alpha_i\) by noticing the value of L = 0 on the optimum gradient point into Equation 25 with boundary from Equation 26. The data correlated with positive \(\alpha_i\) is the support vector.

\[
\sum_{i=1}^{i} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{i} \alpha_i \alpha_j y_i y_j x_i x_j
\]

(25)

\[
\alpha_i \geq 0 \text{ dan } \sum_{i=1}^{i} \alpha_i y_i = 0
\]

(26)

Basically SVM is a linear classification model, but it can be used to classify non-linear using kernel trick as in Equation 27.

\[
K(x_i, x_j) = \emptyset(x_i) \cdot \emptyset(x_j)
\]

(27)

Kernel trick in SVM is a method to classify non-linear data by transforming real data space into data space with higher dimension. Most used Kernels in SVM are shown in Table 1

| Kernel Name                      | Inner Product Kernel                        |
|---------------------------------|--------------------------------------------|
| Linear                          | \(x^T x\)                                  |
| Polynomial                      | \((x^T x_i + 1)^p\) with unrestricted p value, determined by user. |
| Radial-basis function (RBF/Gaussian) | \(\exp\left(-\frac{||x-x'||^2}{2\sigma^2}\right)\) with unrestricted \(\sigma\) value, determined by user. |

3. Testing and Result Analysis
The testing in this research compares segmentation yielded by application with ground truth, by calculating accuracy using Equation 28. Testing is done on 50 images taken from benchmark dataset in [6].

\[
ER = \frac{N_f + N_m}{N_t} \times 100%
\]

\[
Accuracy = 100 - ER
\]

(28)

Where variable \(N_f\) shows the amount of false-segmented on image pixels, \(N_m\) shows the amount of miss-segmented on image pixels and \(N_t\) is the total image pixels.
First test was done by comparing segmentation result with ground truth using initial parameter. Accuracy result for 50 images using initial parameter reached 93.69%, which means the segmentation outcome has been pretty good.

### Table 2. Parameter and Initial Methods

| No. | Parameter/Method                                                                 | Value/method name |
|-----|----------------------------------------------------------------------------------|-------------------|
| 1.  | Resize method on visual saliency detection                                       | ‘bilinear’        |
| 2.  | Value of n in average filter convolution matrix in visual saliency detection    | 3                 |
| 3.  | Pixel width on morphology operation in the pre-segmentation process and trimap formation | 70                |
| 4.  | Threshold on local homogeneity                                                   | 0                 |
| 5.  | Local window on local homogeneity                                               | 3 x 3             |
| 6.  | SVM kernel function on pixel classification process                             | Linear            |
| 7.  | Value of parameter C SVM on pixel classification process                         | 1                 |

The second test compares segmentation outcome with ground truth based on differences in resize method and n value on different average filter convolution matrix in visual saliency detection process. Resize methods comparison used in visual saliency detection are bilinear and bicubic. While n value in average filter convolution matrix is 3, as referenced in reference literature, 5 and 7. Table 3 shows the average accuracy for 50 images.

### Table 3. Test results with different parameters in visual saliency detection

| Resize Methods | Average Filter Convolution Matrix 3 | Average Filter Convolution Matrix 5 | Average Filter Convolution Matrix 7 |
|----------------|------------------------------------|------------------------------------|------------------------------------|
| Bilinear       | 93.69                              | 93.61                              | 93.54                              |
| Bicubic        | 93.65                              | 93.53                              | 93.52                              |

Third test was done by comparing segmentation outcome with ground truth based on pixel width differences on morphology operation in pre-segmentation process and different trimap formation. This was done on pixel width on morphology operation with the values of 70 and 75. Best accuracy resulted by pixel width of 75 with the average accuracy of 93.82%.

The next test was done by comparing segmentation outcome with ground truth based on threshold value differences in local homogeneity threshold process, where this threshold value can affect data training
Amount generated in pixel classification to produce image segmentation. This test will use threshold values of 0, 50 and 100. Table 4 shows average accuracy result from 50 images.

| Threshold value | 0  | 50 | 100 |
|----------------|----|----|-----|
| Accuracy       | 93.82 | 94.23 | 94.26 |

The test continues by comparing segmentation outcome with ground truth based on kernel function differences and parameter C value in pixel classification with SVM to create image segmentation. This test uses linear, polynomial and rbf kernel functions with parameter C value of 0.1 and 1. Average accuracy result from 50 images is shown in Table 5. Rbf kernel function has the best accuracy result with C value of 1 and average accuracy result of 94.84%.

| Kernel Function | Parameter | C value 0.1 | C value 1 |
|-----------------|-----------|-------------|-----------|
| Linear          |           | 94.45       | 94.26     |
| Polynomial      | $p = 3$   | 94.10       | 94.23     |
|                 | $p = 4$   | 94.15       | 94.20     |
| Rbf             | $\sigma = 0.25$ | 93.45       | 93.35     |
|                 | $\sigma = 0.5$ | 94.71       | 94.84     |

4. Conclusion
From previously done tests and test result analysis towards this system, it can be concluded that:
1) Saliency-SVM method can generate training data and execute color image segmentation well with average accuracy reaching 94.84%.
2) Image segmentation result with Saliency-SVM is influenced by resize method and $n$ value in average filter convolution matrix in visual saliency detection process, albeit with insignificant difference of accuracy. Resize bilinear method has the best accuracy result with the $n$ value of 3.
3) Image segmentation result with Saliency-SVM is influenced by pixel width on morphology operation in pre-segmentation process and trimap formation, which has the best accuracy with pixel width of 75.
4) Image segmentation result with Saliency-SVM is influenced by threshold value in pixel training selection with local homogeneity threshold, where data training amount generated will affect image segmentation result. Best accuracy was produced by threshold value of 100, which means the more pixel training formed, the better image segmentation generated.
5) Image segmentation result with Saliency-SVM is also affected by kernel function and parameter C on pixel classification process with SVM. Rbf kernel function has the best accuracy result with $\sigma$ value of 0.5 and C parameter of 1.

5. Acknowledgments
Author A.Y expresses the highest gratitude towards Allah SWT for all the grace and mercy, enabling author to finish this research swiftly. Author also expresses gratitude to Mr. Rully Soelaiman and Mrs. Chastine Fatichah who helped a lot in finishing this research.

References
[1] A Y W S M D A N S M Dinda Firly 2014 *Impelementasi Support Vector Machine dengan Optimasi Pruning pada Persoalan Klasifikasi Pola* (Surabaya: Institut Teknologi Sepuluh Nopember)

[2] W W Xuafei Bai 2014 *Saliency-SVM: An automatic approach for image segmentation* *Neurocomputing* **vol. 136** pp. 243-255

[3] X H a L Zhang 2007 *Saliency Detection: A Spectral Residual Approach*

[4] Q-Y W H-Y Y J B Xiang-Yang Wang 2011 *Color image segmentation using automatic pixel classification with support vector machine* *Neurocomputing* **vol. 74** pp 3898-3911

[5] D F S M R S S M Astris Dyah Perwita 2014 *Uji Kinerja Penggabungan Algoritma Support Vector Machine dan Simulated Annealing pada Permasalahan Klasifikasi Pola* (Surabaya: Institut Teknologi Sepuluh Nopember)

[6] J S N-N Z X T a H-Y Tie Liu [Online]. Available: http://research.microsoft.com/en-us/um/people/jiansun/SalientObject/salient_object.htm. [Accessed 03 June 2015].