Face Recognition to Identify Look-Alike Faces using Support Vector Machine

Z Rustam\textsuperscript{1,2} and R Faradina\textsuperscript{1,3}\textsuperscript{*}

\textsuperscript{1}Department of Mathematics, Universitas Indonesia, Depok, 16424, Indonesia
\textsuperscript{*}Corresponding Author. \textsuperscript{2}rustam@ui.ac.id; \textsuperscript{3}ridhani.faradina@sci.ui.ac.id

Abstract. Face recognition has been one of the most interesting technology to study for many researchers. It allows a huge number of face images to be recognized in just a short amount of times, rather than recognizing each image individually through a normal human’s eyes. Its idea is generally based on the assumption that each individual has a unique identity that can be distinguished between one another. However, in the real world, there are individuals who have similar faces to each other. They are referred as “look-alike” faces. This research was conducted to recognize look alike faces. Each image is represented by the features it contains. The classification method used in this research was Support Vector Machine, with the implementation of kernel. Two types of kernel used in this research were the Radial Basis Function kernel and the polynomial kernel. Results showed that SVM with Radial Basis Function kernel obtained higher accuracies.

1. Introduction
Humans have the ability to recognize faces. However, relying on only human’s ability to recognize faces when dealing with large data such as hundreds to thousands and maybe even millions of facial images might cause errors. Thus, researchers have been trying continuously to construct face recognition algorithms that can produce high accuracy for various issues related to face recognition [1,3].

Face recognition is basically done on the assumption that each individual has a unique identity that can be distinguished from one individual to another. However, in the real world, there are individuals who have similar faces to each other. They are referred to as “look-alike” individuals [4].

In general, human faces look similar because of three reasons namely biological reasons such as twins and siblings, certain situations such as lighting, pose, expression or makeup, and because of plastic surgery that can turn a human face similar to other human faces in particular parts such as mouth, nose, chin or lips [4-5].

This research was conducted to recognize look alike faces. By doing so, real-world applications such as searching for missing person, criminals or fugitives can become easier. By recognizing look alike faces, the running process will save much more time because unqualified faces can be removed as they are not needed to be used for further processes. Because the data on the actual problem are usually very large, the reduction of such data can minimize the cost. A database for look-alike faces has already been made by Lamba et al. in 2011 [4], of which it became the database used in this research.

Classification method used in this research is Support Vector Machine (SVM). SVM is one of the machine learning methods introduced by Cortes and Vapnik in 1995 [6]. SVM is often used for the
classification phase in facial recognition because of its high accuracy in general [1]. Initially, SVM is used for problems with only two classes. However, SVM has been developed for multiclass classification.

SVM is a great method to classify data that can be linearly separated. However, linearly separable data are rarely found in reality. So, the data need to be processed first in order for it to be linearly separable. This is where the kernel trick comes in. This research used the implementation of kernel in SVM with two types of kernel, which are the Radial Basis Function kernel and the polynomial kernel.

This paper is organized as follows: In Section 2, the methods used in this research are explained including the basics of Support Vector Machine and the implementation of kernel in SVM. Section 3 explains the data used in this research. Experiment results are portrayed in Section 4 while Section 5 conveyed the conclusions.

2. Method

In this section, the methods used for this research will be explained. First, the concepts and basic formula for Support Vector Machine are first explained. After that, this section explains about the implementation of kernel in SVM.

2.1. Support Vector Machine

Support Vector Machine (SVM) is one of the most widely used methods of supervised learning machine to analyse and identify patterns [7]. For classification cases, SVM is also believed to produce high accuracy. SVM was first introduced to the world by classifying two classes. However, along with the development of researches over years, SVM has now been developed to be able to perform multiclass classification [2, 8-9]. Few examples of the application of SVM are insolvency prediction in insurance companies [10], policy holders satisfactorily classify automobile insurance [11], and intrusion detection system classification [12].

Basically, the Support Vector Machine algorithm constructs a hyper plane to separate two different classes. The hyper plane chosen by the algorithm is the one that maximizes the minimum distance between the hyper plane and the training data. The minimum distance is commonly referred to as margin [6].

The hyper plane formed by SVM is defined as follows [13]:

$$f(x) = w^T \cdot x + b$$

(1)

where $w$ is a set of weights and $b$ is the bias.

The optimization problem of SVM can be summarized as:

$$\min \frac{1}{2} \|w\|^2$$

subject to:

$$y_i(w^T x_i + b) \geq 1, \quad \forall i = 1, 2, ..., N$$

(3)

The above optimization problem in equation (2) with the constraint in equation (3) was considered as a quadratic programming problem. Therefore, one way to solve this was to find the dual form of the equation. We then introduced Lagrange multiplier $\lambda_i, i = 1, 2, ..., N$. Then, Lagrange function was formed as follows:

$$L(w, b, \lambda) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^{N} \lambda_i (y_i(w^T x_i + b) - 1)$$

(4)

By finding the derivatives of equation (4) with respect of $w$ and $b$ equal to zero and then substituting the results back to the equation produced the dual form of equation (2), stated as:

$$\max \left( \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j (x_i \cdot x_j) \right)$$

(5)

subject to:
\[
\sum_{i=1}^{N} \lambda_i y_i = 0, \quad \lambda_i \geq 0
\]  \quad (6)

By solving equation (5) with constraints expressed in equation (6), we could find the optimal parameters and therefore the optimal hyper plane could be formed.

The above SVM discussion was accurate for linearly separable data. However, it was rare or even impossible to find such perfectly linearly separable data in this world. To overcome this, we could map the data to a higher space such that they could be separated linearly in the new space. This was called the kernel trick.

2.2. Implementation of kernel in SVM.

In general, SVM could produce high accuracy for classification with data that could be separated linearly. However, in reality, such data were rarely found. So we needed an additional method that could overcome the problem of non-linear data when using SVM. One way that could be used was by implementing kernel in SVM.

The kernel function could act as a link between non-linear data and the SVM algorithms. The idea of kernel function was to map data in the input space to a new feature space that had a higher dimension such that the data could be separated linearly in the new feature space.

Suppose \(x_1, x_2, \ldots, x_n \in \mathbb{R}^n\) was a set of the original data in \(\mathbb{R}^n\). There was a function \(\phi\) that mapped the data in \(\mathbb{R}^n\) to a new feature space with a higher dimension \(\mathcal{F}\).

\[
\phi: \mathbb{R}^n \rightarrow \mathcal{F}
\]

The kernel function was defined as follows [13]:

\[
K(x, y) = \phi(x) \cdot \phi(y)
\]  \quad (7)

and the distance between two kernelled data were defined as:

\[
d(x, y) = \|\phi(x) - \phi(y)\| = \sqrt{K(x, x) - 2K(x, y) + K(y, y)}
\]  \quad (8)

There were several types of kernel, but this research used two types that had been commonly used for SVM, which were the Radial Basis Function (RBF) kernel and the polynomial kernel. The RBF kernel was defined as:

\[
K(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)
\]  \quad (9)

and the polynomial kernel was defined as [14]:

\[
K(x, y) = (x \cdot y + 1)^d
\]  \quad (10)

The idea of kernelled SVM remained the same as the original SVM, but the work was done when all the data had been mapped into a new feature space such that the data became linearly separable. So, the hyper plane was now defined as [6]:

\[
f(x) = w^T \phi(x_i) + b
\]  \quad (11)

The optimization problem of kernelled SVM was:

\[
\min \frac{1}{2} \|w\|^2
\]  \quad (12)

subject to:

\[
y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i, \quad \forall i = 1, 2, \ldots, N
\]  \quad (13)

By using Lagrange multiplier and forming the Lagrange function, we could find the dual form for the optimization of the kernelled SVM as:

\[
\max \left(\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j K(x_i, x_j) \right)
\]  \quad (14)
subject to:

$$\sum_{i=1}^{N} \lambda_i y_i = 0, \quad \lambda_i \geq 0$$ (15)

By solving equation (14) with constraints expressed in equation (15), we could find the optimal parameters for the optimal hyper plane. Suppose $\lambda^*$ was the solution, then we could find that

$$w = \sum_{i=1}^{N} \lambda_i^* y_i \phi(x_i)$$ (16)

Substituting $w$ in equation (16) to $f(x)$ in equation (11) could result finding $b$ as follows:

$$b = \frac{1}{n_s} \sum_{i \in S} y_i - \sum_{i \in S} \lambda_i^* y_i K(x_i, x_j)$$ (17)

3. Data

The data used in this research were collected from the Look-Alike Faces Database (LAF) which was created by Lamba et al. in 2011 [4]. From the database, there were 50 subjects or people who had 5 images of their genuine face and 5 look-alike images. So in total, 500 face images were used in this research.

The images from the database had already been separated into two different categories by Lamba et al. [4], which were the genuine face images and the look-alike face images. The genuine face images consisted of the faces of well-known or famous people such as the celebrities, ranging from Western, Eastern, and Asian origins. While the look-alike face images consisted of ordinary people from all over the world. The genuine face images were used for the training data set, while the look-alike images were used for the testing dataset. So, the ratio of the training data set and the testing dataset was 5:5 for each subject.

Images in the Look-Alike Faces Database were varied in colour and size. Some images were in colour form (RGB) and some were in grayscale form. All the images were converted into grayscale form for this research. The reason of doing that was particularly because computational time needed would be much reduced.

4. Experiment Results

Experiment for face recognition to identify look-alike faces as proposed in this research was done and the results were summarised in table 1 and table 2 below. Each table contained the results of face recognition to identify look-alike faces using Support Vector Machine. Table 1 shows the result when Radial Basis Function kernels with various $\sigma$ were used, whereas table 2 shows the result when polynomial kernels with various $d$ were used.

As seen in table 1, the experiments done using Support Vector Machine with implementation of RBF kernel resulted in similar accuracies which were 94% except for $\sigma = 0.00001$ which obtained 74.4%. Therefore, the next step was to analyse the results in the amount of time needed which was the running time. The lowest running time for this case was 91.75 seconds with $\sigma = 1$. So, the best result for face recognition to identify look-alike faces using Support Vector Machine with RBF kernel in this research was obtained when $\sigma = 1$ was used.

For the results of Support Vector Machine with polynomial kernel for the face recognition to identify look-alike faces, the highest accuracy was also 94%. The highest accuracy while using Support Vector Machine with polynomial kernel was obtained four times in this experiment, which were for $d = 7$ to $d = 10$. Further analysis is to see the running time and choose the lowest one. The lowest running time for this case is 74.796875 seconds with $d = 7$. So, the best result for face recognition to identify look-alike faces using Support Vector Machine with polynomial kernel in this research was obtained when $d = 7$ was used. It could be seen that for the ones that obtained the highest accuracy, which were from $d = 7$ to $d = 10$, the time needed became higher as the parameter increased.
Table 1. Results Obtained Using Support Vector Machine with $\Sigma$ as the Parameter of the Radial Basis Function Kernel

| $\sigma$ | Accuracy (%) | Running time (seconds) |
|----------|--------------|------------------------|
| 0.00001  | 74.4         | 126.390625             |
| 0.001    | 94           | 126.671875             |
| 0.01     | 94           | 126.65625              |
| 0.1      | 94           | 126.859375             |
| 0.5      | 94           | 126.328125             |
| 1        | 94           | 91.75                  |
| 10       | 94           | 314.5625               |
| 100      | 94           | 126.265625             |
| 1000     | 94           | 126.0625               |

Table 2. Results Obtained Using Support Vector Machine With $d$ As The Parameter Of The Polynomial Kernel

| $d$ | Accuracy (%) | Running time (seconds) |
|-----|--------------|------------------------|
| 1   | 74.4         | 92.109375              |
| 2   | 2            | 160.96875              |
| 3   | 2            | 127.09375              |
| 4   | 2            | 141.421875             |
| 5   | 2            | 134.140625             |
| 6   | 0.4          | 1113.828125            |
| 7   | 94           | 74.796875              |
| 8   | 94           | 82.296875              |
| 9   | 94           | 92.921875              |
| 10  | 94           | 312.671875             |

5. Conclusion
This research proposes face recognition to identify look-alike faces using Support Vector Machine. Two types of kernel function have been implemented, which are the Radial Basis Function (RBF) kernels and the polynomial kernels. The data used are 500 face images taken from the Look-Alike Face database (LAF).

Based on the results obtained, the highest accuracy using Support Vector Machine with RBF kernel is 94%, with 91.75 seconds as the lowest running time. For the polynomial kernel, the highest accuracy is also 94% with 74.796875 seconds needed. For comparison, the highest accuracy obtained in this research by using two different kernels are the same, but the lowest running time needed for the face recognition to finish by using polynomial kernel is lower than the running time needed when RBF kernel was used. In conclusion, for face recognition to identity look-alike faces, SVM has successfully produced high accuracies.
6. References

[1] Guo G, Li S and Chan K 2000 Face Recognition by Support Vector Machines *Proc. 4th. IEEE International Conference on Automatic Face and Gesture Recognition* pp 196–201

[2] Li J, Zhao B, Zhang H and Jiao J 2009 Face Recognition System Using SVM Classifier and Feature Extraction by PCA and LDA Combination *Int. Conf. on Computational Intelligence and Software Engineering* pp 1–4

[3] Wei J, Jian-qí Z and Xiang Z 2011 Face Recognition Method Based On Support Vector Machine And Particle Swarm Optimization *Expert Systems with Applications* 38 pp 4390–4393

[4] Lamba H, Sarkar A, Vatsa M, Singh R and Noore A 2011 Face Recognition For Look-Alikes: A Preliminary Study *Int. Joint Conf. on Biometrics (IJCB)* pp 1–6

[5] Moeini A, Faez K, Moeini H and Safai A M 2017 Open-Set Face Recognition Across Look-Alike Faces in Real-World Scenarios *Image and Vision Computing* 57 pp 1–14

[6] Cortes C and Vapnik V 1995 Support-Vector Networks *Machine Learning* 20 pp 273–297

[7] Vapnik V N 1999 An Overview of Statistical Learning Theory *IEEE Transactions on Neural Networks* 10(5) pp 988–999

[8] Ko J and Byun H 2003 Multi-Class Support Vector Machines with Case-Based Combination for Face Recognition *Int. Conf. on Computer Analysis of Images and Patterns* pp 623–629

[9] Prakash J S, Vignesh K A, Ashok C and Adithyan R 2012 Multi Class Support Vector Machines Classifier for Machine Vision Application *Int. Conf. on Machine Vision and Image Processing (MVIP)* pp 197–199

[10] Rustam Z and Yaurita F 2018 Insolvency Prediction in Insurance Companies Using Support Vector Machines And Fuzzy Kernel C-Means *Journal of Physics: Conference Series* 1028(012118)

[11] Rustam Z and Ariantri N P A A 2018 Support Vector Machines for Classifying Policyholders Satisfactorily in Automobile Insurance *Journal of Physics: Conference Series* 1028(012005)

[12] Rustam Z and Zahras D 2018 Comparison Between Support Vector Machine and Fuzzy C-Means as Classifier for Intrusion Detection System *Journal of Physics: Conf. Series* 1028(012227)

[13] Bishop C M 2006 Pattern Recognition and Machine Learning (New York: Springer)

[14] Liu L, Shen B and Wang X 2014 Research On Kernel Function of Support Vector Machine *Advanced Technologies, Embedded and Multimedia for Human-centric Computing* pp 827–834