

Emotion Detection from Facial Expression using Support Vector Machine

Vanita Jain
Bharati Vidyapeeth's
College of Engineering,
New Delhi

Pratiksha Aggarwal
Bharati Vidyapeeth's
College of Engineering,
New Delhi

Tarun Kumar
Bharati Vidyapeeth's
College of Engineering,
New Delhi

Vaibhav Taneja
Bharati Vidyapeeth's
College of Engineering,
New Delhi

ABSTRACT

The objective of this paper is to apply Support Vector Machine to the problem of classifying emotion on images of human faces. This well defined problem is complicated by the natural variation in people's faces, requiring the classification algorithm to distinguish the small number of relevant features from the large pool of input features. Three different kernels i.e., linear kernel, polynomial kernel and RBF kernel are used to recognise eight facial expressions, anger, contempt, disgust, fear, happiness, neutral, sadness and surprise of human beings in still images. Accuracy of the three kernels is compared to judge the best kernel for facial expression recognition.

Keywords

Facial Expression, Support Vector Machine, Emotion Detection

1. INTRODUCTION

Facial expression is a natural way of conveying social information without words between humans. Facial Expression Recognition (FER) is an active research area in psychological study as well as computer-human interaction. A facial expression can be considered as a visible attestation of inner state of mind and hence gives idea about intention, interest, and psychology of that person. As a result of the huge information carried by facial expressions, they play vital role in human interaction with humans as well as machine [1]. Facial expression recognition can be used as a vital component in natural interface between human and machine. Smart Devices like computers or robots can recognize the expressions and hence emotional state of human. There are two main approaches for facial expression analysis. First approach makes use of the whole frontal face image and processes it in order to end up with the classifications of facial expression. In the second approach, instead of using the face images as a whole, it is divided into sub-sections and processed further. Human beings naturally and intuitively use facial expression as an important and powerful modality to communicate their emotions and to interact socially [2]. There has been continued research interest in enabling computer systems to recognise expressions and to use the emotive information embedded in them in human-machine interfaces [3].

In the recent years, due to its wide range of applications, FER has become a vital research field and efforts are still being taken for increasing FER accuracy. The differences distinguishing separate expressions such as 'anger' or 'disgust' and the large number of variations in a particular expression when performed by different subjects led us to the adoption of SVM as the classifier of choice.

2. SUPPORT VECTOR MACHINE

Support vector machine (SVM) is a set of supervised learning method used for classification, regression and outliers' detection. The goal of a support vector machine is to find the optimal separating hyperplane which maximizes the margin of the training data [4]. The SVM is useful in a number of pattern analysis tasks with the help of binary classifiers which differentiates effectively between various classifications of expression. SVM works by classifying data with the help of evaluation of an optimal hyper plane that separates one class's data points from those of the other [4].

Support Vector Machine aims to perform 2-class classification. It optimally separates the data by making use of an N-dimensional hyper plane. Therefore, SVM belongs to the family of linear classifiers and is an example of supervised learning. For Facial Expression Recognition (FER), SVM model divides the mapped data samples by clear gap as wide as possible. This division is performed by hyper planes. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

SVM exhibit good classification accuracy even when only a modest amount of training data is available, making them particularly suitable to a dynamic, interactive approach to expression recognition [5]. Support vector machine have previously been successfully employed in a variety of classification applications including identity and text recognition as well as DNA microarray data analysis [6].

3. FACIAL EXPRESSION RECOGNITION

FER is a type of pattern recognition problem which usually consists of two modules, Facial feature extraction and classification [1]. In the FER system, after feature extraction, the extracted features are classified. The feature extraction phase represents a key component of any pattern recognition system. It converts raw pixel data into a higher-level representation of shape, colour and texture.

The system is first trained to recognize basic expressions as anger, contempt, disgust, fear, happiness, neutral, sadness, surprise etc. and then is used for recognizing expression of unknown images. The flow of the system is shown in the Figure 1.

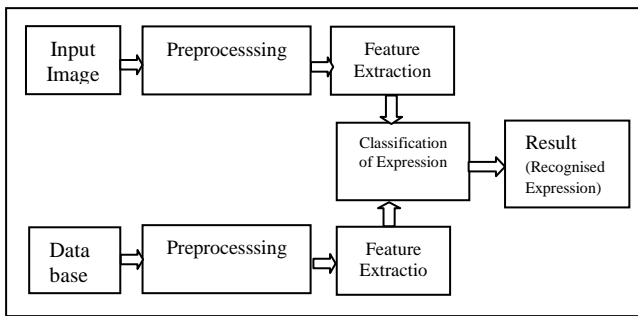


Figure 1: System Flow for FER Model

4. DESCRIPTION OF LIBRARIES USED

Support Vector Machine is implemented via Python for the dataset [13] for eight different facial expressions of different people. Six python libraries are used, Glob, CV2, Dlib, NumPy, Sklearn and SVC. The details of the libraries are:

(i) **Glob:** The glob module finds all the pathnames matching a specified pattern according to the rules used by the Unix shell, although results are returned in arbitrary order. No tilde expansion is done, but *, ?, and character ranges expressed with [] will be correctly matched. This is done by using the `os.listdir()` and `fnmatch.fnmatch()` functions in concert, and not by actually invoking a subshell. Note that unlike `fnmatch.fnmatch()`, glob treats filenames beginning with a dot (.) as special cases. (For tilde and shell variable expansion, use `os.path.expanduser()` and `os.path.expandvars()`.) For a literal match, wrap the meta-characters in brackets. For example, `[?]` matches the character '?' [7].

(ii) **OpenCV:** (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code [8].

(iii) **Dlib:** It is a general purpose cross-platform software library written in the programming language C++. Its design is heavily influenced by ideas from design by contract and component-based software engineering. Thus it is, first and foremost, a set of independent software components. It is open-source software released under a Boost Software License [9].

(iv) **NumPy:** It is the fundamental package for scientific computing with Python. It contains among other things [10]:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

(v) **Sklearn:** Scikit-learn (formerly scikits.learn) is a free software machine learning library for the python programming language. It features various classification, regression and clustering algorithms including support vector machine, random forests, gradient boosting, k-means and dbscan, and is designed to interoperate with the python numerical and scientific libraries numpy and scipy [11].

(vi) **SVC:** SVC (support vector classifier) is the function of SVM library that is used to implement multiclass classification. The implementation is based on libsvm. The fit time complexity is more than quadratic with the number of

samples which makes it hard to scale to dataset with more than a couple of 10000 samples. The multiclass support is handled according to a one-vs-one scheme [12].

5. METHODOLOGY

The data collected from [13], contains two main folders named sourceimages and sourceemotions. Sourceimages folder contains 123 subfolders. Each of which contains around 11 images ranging from neutral to particular expression. Sourceemotions folder contains 123 subfolders, each containing 8 subsubfolders containing text file which gives confidence of each expression. Each image is in RGB format with size 256x256. Figure 2 shows sample database.

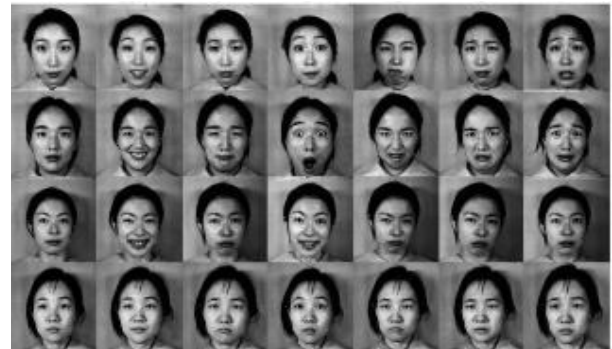


Figure 2: Sample Database

5.1 Preprocessing

The data is classified into eight folders named “anger”, “happy”, “sad”, “neutral”, “disgust”, “surprise”, “contempt”, “fear” according to the maximum confidence contained in text file of sourceemotion. Multiple images from neutral folder are removed manually.

Glob function of glob library is used to read file names of each folder and save it in a list. This list contains the names of the file of entire dataset for a particular emotion. The division of dataset is done by finding the length of the list, which contains the names of images, and dividing it into 80:20 ratio. The two lists are named as training data and testing data.

`imread()` function of opencv is used to read/open image. The read image is converted into greyscale from RGB using `cvtColor()` function of opencv.

The function named `getlandmarks()` is created in which two different functions of dlib library named `get_frontal_face_detector()` and `shape_predictor()` are called. The functions are used to detect face from the given image and the 68 landmarks respectively of each image in training data and testing data. These 68 features are in coordinate form which is saved in two different list named x-list and y-list.

Mean of both the lists (x and y) is calculated using `mean()` function of numpy library to determine the Centre of Gravity. Distance of each feature point from centre of gravity and angle with respect to x-axis is calculated and appended in a list named vectorised landmarks.

SVM is now trained using `fit()` function of Support Vector Classifier (SVC) module imported from `sklearn.svm` on training data. The `predict()` function of SVC module is used to test support vector machine. The results obtained are compared to the original confidence contained in sourceemotion folder to calculate accuracy. The new input images from which emotions are to be detected are saved in folder named prediction images and these images are read

through program for detection of face. The read image is cropped according to the face detected and converted into greyscale. These greyscale images are saved in a folder named classification set. Landmarks on these images are calculated as before and predicted using the trained SVM.

6. RESULT

The accuracy of Linear, Polynomial and RBF Kernel of SVM has been calculated for different training:testing ratio. Table 1 shows the accuracy of Linear Kernel for different ratios of training and testing. It has been observed that as the ratio of training and testing increases, the accuracy also increases.

Result of Linear Kernel is shown in Table 1.

Table 1: Accuracy of Linear Kernel for different ratios of training and testing.

(Training:Testing)	Percentage Accuracy of Linear Kernel
50-50	64%
60-40	67%
70-30	65%
80-20	73%
90-10	76%

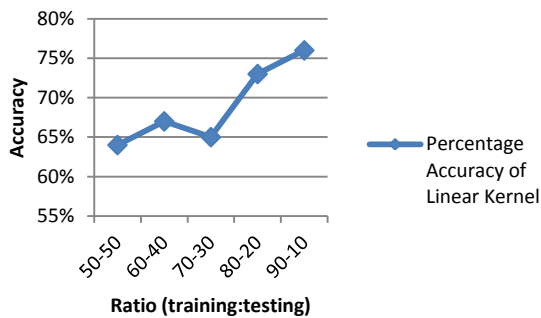


Figure 3: Accuracy vs Ratio graph for Linear Kernel

The result for Polynomial Kernel has also been obtained and it has been found that the efficiency is maximum when the training:testing ratio is 80:20 which has been shown in Table 2.

Table 2: Accuracy of Polynomial Kernel for different ratios of training and testing

Ratio (Training:Testing)	Accuracy of Polynomial Kernel
50-50	63.8%
60-40	65.3%
70-30	66.5%
80-20	70.2%
90-10	64.4%

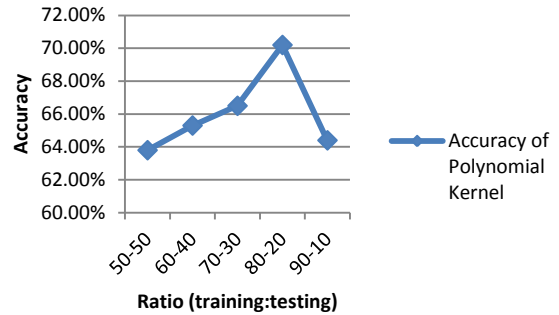


Figure 4: Accuracy vs Ratio graph for Polynomial Kernel

The accuracy RBF kernel also increases as in the case of is shown in Table 3.

Table 3: Accuracy of RBF Kernel for different ratios of training and testing

(Training:Testing)	Accuracy of RBF Kernel
50-50	22.9%
60-40	23%
70-30	22.8%
80-20	23%
90-10	23.3%

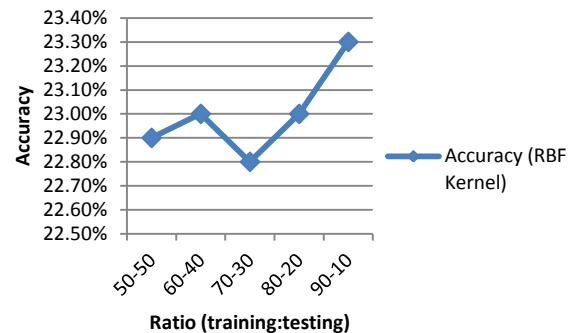


Figure 5: Accuracy vs Ratio graph for RBF Kernel

The comparison of the accuracies of the different kernels is shown in table 4 and figure 6.

Table 4: Comparison of accuracy of Linear Kernel, Polynomial Kernel and RBF Kernel for different ratios of training and testing.

(Training:Testing)	Percentage Accuracy		
	Linear Kernel	Polynomial Kernel	RBF Kernel
50-50	64%	63.8%	22.9%
60-40	67%	65.3%	23%
70-30	65%	66.5%	22.8%
80-20	73%	70.2%	23%
90-10	76%	64.4%	23.3%

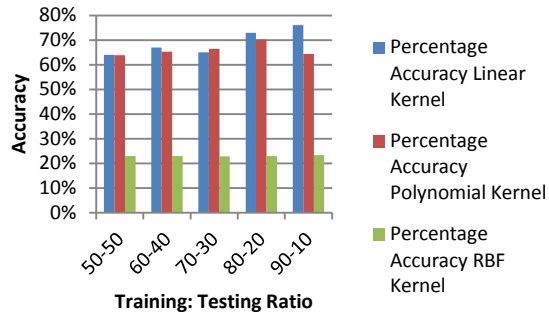


Figure 6: Accuracy vs Ratio graph for three different kernels

RBF kernel gives the lowest accuracy independent of training:testing ratio. However there is not much difference in the accuracy of linear and polynomial kernel.

7. CONCLUSION

Facial Expression Recognition is carried out using three different kernels of Support Vector Machine. The accuracy for each kernel is calculated which yields the result that linear kernel is the best for emotion recognition.

8. REFERENCES

- [1] K. R. Kulkarni and S. B. Bagal, "Facial Expression Recognition," *2015 Annual IEEE India Conference (INDICON)*, New Delhi, 2015, pp. 1-5.
- [2] Ekman, P. (1982) *Emotion in the Human Face*. Cambridge: Cambridge University Press.
- [3] Paul Ekman. *Pictures of facial affect*. 1976.
- [4] Emotional Expression Recognition using Support Vector Machine Melanie Dumas Department of Computer Science University of California, San Diego La Jolla, CA 92193-0114.
- [5] Vanita Jain, Renu Taneja, Anshul Goel, Megha Kumari, "Facial Features and Biomedical Signs: A Criterion for Pain Expression Recognition", *INDIACom-2017; IEEE Conference*, pp- 4482-4488, 2017
- [6] Facial Expression Recognition Using Support Vector Machine Philipp Michel and Rana El Kaliouby Computer Laboratory University of Cambridge Cambridge CB3 0FD, U.K.
- [7] Van Gent, P. (2016). *Emotion Recognition Using Facial Landmarks, Python, DLib and OpenCV. A tech blog about fun things with Python and embedded electronics.*
- [8] GUO Lei,WANG Qiu-guang, "Research of Face Detection Based on Adaboost Algorithm and OpenCV Implementation", *Journal of Harbin University of Science and Technology*, 2009.
- [9] Davis E. King, "Dlib-ml: A Machine Learning Toolkit", *Journal of Machine Learning Research* 10 (2009) 1755-1758.
- [10] Christian Bauckhage, "NumPy / SciPy Recipes for Data Science: Ordinary Least Squares Optimization", DOI: 10.13140/2.1.3370.3209/1
- [11] Matthieu Perrot, Édouard Duchesnay, "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, 2825-2830, 2011.
- [12] S. Zhou, X. Liang and C. Zhu, "Exploiting LCSVC Algorithm for Expression Recognition," *2008 The 9th International Conference for Young Computer Scientists, Hunan, 2008*, pp. 1760-1765. doi: 10.1109/ICYCS.2008.68
- [13] Cohn-Kanade (CK and CK+) database Download Site <http://www.consortium.ri.cmu.edu/ckagree/>
- [14] Vanita Jain, Shikha Rastogi, Tannishk Sharma, Aarushi Jain, "An empirical study of Classification Algorithms for the diagnosis of Hepatitis", *INDIACom-2017; IEEE Conference*, pp 4477-4481, 2017
- [15] M. Abdulrahman and A. Eleyan, "Facial expression recognition using Support Vector Machine," *2015 23rd Signal Processing and Communications Applications Conference (SIU)*, Malatya, 2015, pp. 276-279.
- [16] Liyuan Chen, Changjun Zhou, Liping Shen, "Facial Expression Recognition Based on SVM in E-learning", *IERI Procedia*, Volume 2, 2012, Pages 781-787.