ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 11, March 2017

DYNAMIC FACIAL EXPRESSION RECOGNITION WITH EXTREME SPARSE LEARNING

P.Vignesh[1], Mr. R.Yasir Abdullah M.E., [2]

[1]Department of Computer Science, Sri Subramanya College of Engineering and Technology, Palani

[2]Department of Computer Science, Sri Subramanya College of Engineering and Technology, Palani

[1]vignesh.p123@gmail.com

Abstract : Recognition of natural emotions from human faces is an interesting topic with a wide range of potential applications like human-computer interaction, automated tutoring systems, image and video retrieval, smart environments, and driver warning systems. Traditionally, facial emotion recognition systems have been evaluated on laboratory controlled data, which is not representative of the environment faced in real-world applications. A new dynamic facial expression recognition method is proposed. Dynamic facial expression recognition is formulated as a longitudinal group wise registration problem. The main contributions of this method lie in the following aspects: (1) subject-specific facial feature movements of different expressions are described by a diffeomorphic growth model; (2) salient longitudinal facial expression atlas is built for each expression by a sparse groupwise image registration method, which can describe the overall facial feature changes among the whole population and can suppress the bias due to large inter-subject facial variations; (3) both the image appearance information in spatial domain and topological evolution information in temporal domain are used to guide recognition by a sparse representation method.

1. Introduction

Automatic facial expression recognition (AFER) has essential real world applications. Its applications include, but are not limited to, human computer interaction (HCI), psychology and telecommunications. It remains a challenging problem and active research topic in computer vision, and many novel methods have been proposed to tackle the automatic facial expression recognition problem. Intensive studies have been carried out on AFER problem in static images during the last decade: Given a query facial image, estimate the correct facial expression type, such as anger, disgust, happiness, sadness, fear or surprise. It mainly consists of two steps: feature extraction and classifier design. For feature extraction, Gabor wavelet, local binary pattern (LBP), and geometric features such as active appearance model (AAM) are in common use. For classifier, support vector machine is frequently used. Joint alignment of facial images under unconstrained condition has also become an active research topic in AFER.

In recent years, dynamic facial expression recognition has become a new research topic and receives more and more attention. Different from the recognition problem in static images, the aim of dynamic facial expression recognition is to estimate facial expression type from an image sequence captured during physical facial expression process of a subject. The facial expression image sequence contains not only image appearance information in the spatial domain, but also evolution details in the temporal domain. The image appearance information together with the expression evolution information can further enhance recognition performance.

1. System Analysis

1.1 Existing System

The existing method uses a longitudinal atlas construction and diffeomorphic image registration problem. The system mainly consists of two stages, namely atlas construction stage and recognition stage. In the atlas construction stage, longitudinal atlas of different facial expressions are constructed based on sparse representation groupwise registration. The constructed atlas can capture overall facial appearance movements for a certain expression among the population. In the recognition stage, both the image appearance and temporal information are considered and integrated by diffeomorphic registration and sparse representation.

1.2 Proposed System

To recognize the emotions in the presence of self-occlusion and illumination variations, we combine the idea of sparse representation with Extreme Learning Machine (ELM) to learn a powerful classifier that can handle noisy and imperfect data. Sparse representation is a powerful tool for reconstruction, representation, and compression of high-dimensional noisy data (such as images/videos and features derived from them) due to its ability to uncover important information about signals from the base elements or dictionary atoms. While the sparse representation approach has the ability to enhance noisy data using a dictionary learned from clean data, it is not sufficient because our end goal is to correctly recognize the facial emotion. In a sparse-representation-based classification task, the desired dictionary should have both representational ability and discriminative power.

Since separating the classifier training from dictionary learning may cause the learned dictionary to be sub-optimal for the classification task, we propose to jointly learn a dictionary (which may not be necessarily over-completed) and a classification model. To the best of our knowledge, this is the first attempt in the literature to simultaneously learn the sparse representation of the signal and train a non-linear classifier based on sparse codes.

The key contributions of this system are as follows:

- A pose-invariant OF-based spatio-temporal descriptor, which is able to robustly represent facial emotions even when there are head movements while expressing an emotion. The proposed descriptor is capable of characterizing both the intensity and dynamics of facial emotions.
- A new classifier called Extreme Sparse Learning (ESL) is obtained by adding the ELM error term to the objective function of the conventional sparse representation to learn a dictionary that is both discriminative and reconstructive. This combined objective function (containing both linear and non-linear terms) is solved using a novel approach called Class Specific Matching Pursuit (CSMP). A kernel extension of the above framework called Kernel ESL (KESL) has also been developed.

our end goal is to correctly recognize the facial emotion. In a sparse-representation-based classification task, the desired dictionary should have both representational ability and discriminative power. Since separating the classifier training from dictionary learning may cause the learned dictionary to be sub-optimal for the classification task, we propose to jointly learn a dictionary (which may not be necessarily over-complete) and a classification model.

3. System Testing

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test. A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that “al gears mesh”, that is the internal operation of the product performs according to the specification and

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 11, March 2017

all internal components have been adequately exercised.

3.1 Types of Testing

3.1.1 Unit Testing

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as 'module testing'. The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

3.1.2 Integration Testing

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

i) Top-down integration testing.
ii) Bottom-up integration testing.

3.1.3 White Box Testing

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.

3.1.4 Black Box Testing

- Black box testing is done to find incorrect or missing function
- Interface error
- Errors in external database access
- Performance errors
- Initialization and termination errors

In 'functional testing', is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called 'black box testing'. It tests the external behavior of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

3.1.5 Validation Testing

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 11, March 2017

and final series of software validation tests begin validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

3.1.6 User Acceptance Testing

User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

3.1.7.Output Testing

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

4. System Implementation

4.1 FEATURE EXTRACTION

After frame conversion new face coordinate system on the image Plane is developed.Using Delaunay triangulation facial features like nose tip,left eye,right eye and mouth are detected. While the nose tip is considered as the origin of the face coordinate system, the reference vector connecting the nose tip to the midpoint between the centres of the two eyes is considered as the positive *y*-axis.

4.2 OPTICAL FLOW

In order to compute the dynamic features, we start by computing the Optical Flow of a given video based on both brightness and gradient constancy assumption. The local motion of facial components resulting from the act of expressing an emotion, global motion of the head is subtracted from the flow vector.

4.3 SPATIO-TEMPORAL FEATURES

Four pose-invariant features are proposed for encoding the motion information of facial components. The first feature is the *divergence* of the flow field, which measures the amount of local expansion or contraction of the facial muscles. The second feature captures the local spin around the axis perpendicular to the OF plane and is referred to as *Curl*. It is useful to measure the dynamics of the local circular motion of the facial components. The third feature is the scalar projection. This *Proj* feature captures the amount of expansion or contraction of each point with respect to the nose point. For example, the “happy” and “sad” emotions can be distinguished by this feature clearly.The fourth feature is the *Rot feature*. The *Rot* feature measures

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 11, March 2017

the amount of clockwise or anti-clockwise rotation of each facial point movement with respect to the position vector. this feature help in distinguishing between "happy" and "anger" emotions. The sign and magnitude of *Rot* feature are different for a sample lip point (the magnitude is exaggerated for better illustration) depending on the facial emotion. A spatio-temporal descriptor is obtained by concatenating the spatio-temporal features extracted at each local region in the video.

4.4 EXTREME SPARSE LEARNING

Sparse representation is that the images can be efficiently approximated by linear combination of a few elements called atoms. The dictionary can be obtained by either applying predefined transforms to the data or directly learning from training data. Extreme Learning machine(ELM) is the classification method used. Supervised sparse coding is performed on the dictionary $\boldsymbol{D}$ and the classification error based on the given ELM output weight vector β.Algorithm used in Supervised sparse coding is Class specific Matching pursuit. Supervised sparse coding, estimates a sparse code matrix $\boldsymbol{X}$ that simultaneously minimizes the reconstruction error based on the given dictionary $\boldsymbol{D}$ and the classification error based on the given ELM output weight vector β.

4.5 CLASS SPECIFIC MATCHING PURSUIT (CSMP)

The basic idea underlying the matching pursuit process is to sequentially find atoms in the dictionary and the corresponding sparse coefficients that minimize the objective function $\boldsymbol{\varepsilon}_{t}$..An updated sparse code matrix is obtained.

5 SYSTEM REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

Processor	:	Pentium IV 2.4 GHz
Hard Disk	:	320 GB
RAM	:	2GB

5.2SOFTWARE REQUIREMENTS

Operating System	:	windows 2010
Front End	:	MATLAB
Back End	:	SQL Server 2007

4.3 **MATLAB**

MATLAB (**mat**rix **lab**oratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 11, March 2017

symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

In 2004, MATLAB had around one million users across industry and academia. MATLAB users come from various backgrounds of engineering, science, and economics. MATLAB is widely used in academic and research institutions as well as industrial enterprises.

It is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB System

The MATLAB system consists of five main parts:

4.4 The MATLAB language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

The MATLAB working environment

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

Handle Graphics

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete Graphical User Interfaces on your MATLAB applications.

The MATLAB mathematical function library

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

The MATLAB Application Program Interface (API)

This is a library that allows you to write C and Fortran programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

4.6 FEASIBILITY STUDY

The feasibility study is carried out to test whether the proposed system is worth being implemented. The proposed system will be selected if it is best enough in meeting the performance requirements.

The feasibility carried out mainly in three sections namely.

- Economic Feasibility
- Technical Feasibility
- Behavioral Feasibility

Economic Feasibility

Economic analysis is the most frequently used method for evaluating effectiveness of the proposed system. More commonly known as cost benefit analysis. This procedure determines the benefits and saving that are expected from the system of the proposed system. The hardware in system department if sufficient for system development.

Technical Feasibility

This study center around the system's department hardware, software and to what extend it can support the proposed system department is

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 11, March 2017

having the required hardware and software there is no question of increasing the cost of implementing the proposed system. The criteria, the proposed system is technically feasible and the proposed system can be developed with the existing facility.

Behavioral Feasibility

People are inherently resistant to change and need sufficient amount of training, which would result in lot of expenditure for the organization. The proposed system can generate reports with day-to-day information immediately at the user's request, instead of getting a report, which doesn't contain much detail.

5 System Implementation

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. The people are not sure that the software is meant to make their job easier.

- The active user must be aware of the benefits of using the system
- Their confidence in the software built up
- Proper guidance is impaired to the user so that he is comfortable in using the application

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not running on the server, the actual processes will not take place.

5.1 User Training

To achieve the objectives and benefits expected from the proposed system it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for education and training is more and more important. Education is complementary to training. It brings life to formal training by explaining the background to the resources for them. Education involves creating the right atmosphere and motivating user staff. Education information can make training more interesting and more understandable.

5.2 Training on the Application Software

After providing the necessary basic training on the computer awareness, the users will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

5.3 Operational Documentation

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. A documentation providing the whole operations of the system is being developed. Useful tips and guidance is given inside the application itself to the user. The system is developed user friendly so that the user can work the system from the tips given in the application itself.

6 Conclusion

In this system, we propose a new way to tackle the dynamic facial expression recognition problem. It is formulated as a longitudinal atlas construction and diffeomorphic image registration problem. Our method mainly consists of two stages, namely atlas construction stage and recognition stage. In the atlas construction stage, longitudinal atlas of different facial expressions is constructed based on sparse representation groupwise registration. The constructed atlas can capture overall facial appearance movements for a certain expression among the population. In the recognition stage, both the image appearance and temporal information are considered and integrated by diffeomorphic registration and sparse representation. Our method has been extensively evaluated on five dynamic facial expression recognition databases. The experimental results show that this method consistently achieves higher recognition rates than other compared methods. One limitation of the proposed method is that it is still not robust enough to overcome challenges of strong illumination changes. The main reason is that the LDDMM registration algorithm used in this paper may not compensate strong illumination changes. One possible solution is to use complex image matching metrics in the LDDMM framework, such as localized correlation coefficient and localized mutual information which have some degrees of robustness against illumination changes. This is one possible future direction for this study.

7 References

1. M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, "Recognizing facial expression: machine learning and application to spontaneous behavior," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 568–573.

2. M. Beg, M. Miller, A. Trouve, and L. Younes, "Computing large deformation metric mappings via geodesic flows of diffeomorphisms," International Journal of Computer Vision, vol. 61, pp. 139–157, 2005.

3. Y. Chang, C. Hu, R. Feris, and M. Turk, "Manifold based analysis of facial

expression," Image and Vision Computing, vol. 24, no. 6, pp. 605–614, 2006

4. L. Cohen, N. Sebe, A. Garg, L. Chen, and T. Huang, "Facial expression recognition from video sequences: temporal and static modeling," Computer Vision and Image Understanding, vol. 91, pp. 160–187, 2003.

5. B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," Pattern Recognition, vol. 36, pp. 259–275, 2003.

6. H. Fang, N. Parthalain, A. Aubrey, G. Tam, R. Borgo,P. Rosin,P. Grant, D. Marshall, and M. Chen, "Facial expression recognition in dynamic sequences: An integrated approach," Pattern Recognition, vol. 47, no. 3, pp. 1271–1281, 2014.

7. Y. Guo, G. Zhao, and M. ¨Pietikainen, "Dynamic facial expression recognition using longitudinal facial expression atlases," in European Conference on Computer Vision, 2012, pp. 631–644.

8. S. Koelstra, M. Pantic, and I. Patras," A dynamic texture-based approach to recognition of facial actions and their temporal models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, pp. 1940–1954, 2010.

9. S. Lucey, A. Ashraf, and J. Cohn, "Investigating spontaneous facial action recognition through aam representations of the face," in Face Recognition Book, 2007, pp. 275–286.

10. E. Learned-Miller, "Data driven image models through continuous joint alignment," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 2, pp. 236–250, 2006.

11. T. Ojala, M.¨Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 971–987, 2002.

12. M. Pantic and I. Patras, "Dynamics of facial expression: Recognition of facial actions and their temporal segments form face profile image sequences," IEEE Transactions on Systems, Man, and Cybernetics Part B, vol. 36, no. 2, pp. 433–449, 2006

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 11, March 2017

13. A. Ramirez and O. Chae, "Spatiotemporal directional number transitional graph for dynamic texture recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, p. In Press, 2015.

14. Z. Wang, S. Wang, and Q. Ji, "Capturing complex spatio-temporal relations among facial muscles for facial expression recognition," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2013, pp. 3422–3429.

15. S. Yousefi, P. Minh, N. Kehtarnavaz, and C. Yan, "Facial expression recognition based on diffeomorphic matching," in International Conference on Image Processing, 2010, pp. 4549–4552.

16.Z. Zeng, M. Pantic, G. Roisman, and T. Huang, " A survey of affect recognition methods: Audio, visual, and spontaneous expressions," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, pp. 39–58, 2009.

17. G. Zhao and M.¨Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, pp. 915–928, 2007.

18. Y. Zhang and Q. Ji, "Active and dynamic information fusion for facial expression understanding from image sequences," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, pp. 699–714, 2005