RECONSTRUCTION OF FACIAL IMAGE

Prof. Drakshayini KB[1], Madhushree GM[2], Rakshitha K[3] , Rakshitha M[4], Tanuja HR[5]

[1]*Prof at Information Science and Engineering & Vidya Vikas Institute of Engineering and Technology, Mysuru.*
[2]*Information Science and Engineering & Vidya Vikas Institute of Engineering and Technology, Mysuru.*
[3]*Information Science and Engineering & Vidya Vikas Institute of Engineering and Technology, Mysuru.*
[4]*Information Science and Engineering & Vidya Vikas Institute of Engineering and Technology, Mysuru*
[5]*Information Science and Engineering & Vidya Vikas Institute of Engineering and Technology, Mysuru.*

---***---

Abstract - Human faces are devastating comparative in worldwide properties, including size, viewpoint proportion, and area of principle components, however, can differ extensively in subtle elements crosswise over people, gender, race, or because of countenance . We propose a completely unique technique for 3D face reconstruction of appearances that adventures the similitude of faces. Our technique uses as info a unique picture and uses an insignificant 3D reference model of an alternate individual's face. Traditional remaking techniques from single pictures, i.e. shape-from-shading, require learning of the reflectance properties and lighting and also profundity values for limit conditions. Late techniques evade these prerequisites by speaking to information confronts as mixes (of hundreds) of putting away 3D models. We propose instead to use the input image as a guide to "mold" one reference model to succeed in a reconstruction of the sought 3D shape.

***Key Words*: Deep learning, face allignment, face matching.**

1.INTRODUCTION

3D- shape, and reflectance give properties of articles that are invariant to the progressions brought on by the imaging procedure including perspective, enlightenment, and occlusion by other objects. Information of these properties can improve acknowledgment, permit expectation of appearance under novel seeing conditions, and help with an assortment of utilizations counting graphical activity, medicinal applications, and more. In this paper, considering the shading information alongside unpleasant earlier shape learning to 3-dimensional state of a completely unique face from an input image. In a worldwide sense, diverse countenances are exceptionally comparable . Appearances of changed people have an identical fundamental feature (eyes, nose, and mouth) in generally similar areas. Be that as it may, locally, confront shapes can fluctuate extensively crosswise over people, gender, race, or therefore of countenance . The worldwide similitude of appearances is abused, for instance, in face acknowledgment techniques, to evaluate the stance of novel faces by adjusting a face picture to a non-exclusive face demonstrate.

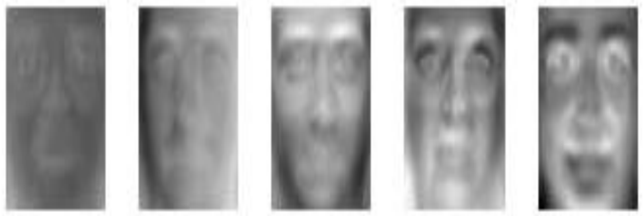
2.PROPOSED SYSTEM

We developing a method for shape recovery of a face from a single image that using reference 3D face model of either a different individual or a generic face. Intuitively, our method uses the input image as a guide to "mold" the reference model to reach the desired reconstruction. Here use the shading information to recover the 3D shape of a face while using the reference shape and extract information essential for the recovery process that's unknown a priori, such as lighting and pose.

3.METHODOLOGY

The project is implemented in modular approach. Each module is coded as per the requirements and tested and this process is iterated till the all the modules have been thoroughly implemented.

Geometry-based Method for feature extraction: Geometry-based Methods is used of specialized edge and contour detectors to find the location of a set of facial landmarks and to measure relative positions and distances between them. The accuracy of these early systems was demonstrated on very small databases of faces was used. In a geometry-based method similar to edge was compared with a method that represents face images as gradient images. The authors showed that differentiate gradient images provided better recognition accuracy than comparing geometry-based features. However, the geometry-based method was quicker and needed less memory. The feasibility of utilize facial landmarks and their geometry for face recognition was thoroughly studied. Specifically, they proposed a way supported measuring the Procrustes distance between two sets of facial landmarks and a way supported measuring ratios of distances between facial landmarks. The authors argued that albeit other methods that extract more information from the face (e.g. holistic methods) could achieve greater recognition accuracy, the proposed geometry-based methods were faster and will be utilized in combination with other methods to develop hybrid methods. Geometry-based methods have proven simpler in 3D face recognition due to the depth information encoded in 3D landmarks Geometry-based methods were crucial for face recognition and reconstruction research.



Top 5 eigenfaces computed using the ORL database of faces [31] sorted from most variance (left) to least variance (right).

Comparison Using CNN

Convolutional neural networks (CNNs) are the most common type of deep learning method for face recognition. The main advantage of deep learning methods is that they can be trained with large amounts of data to learn a face representation that is robust to the variations present in the training data. In this way, instead of designing specialised features that are robust to different types of intra-class variations (e.g. illumination, pose, countenance , age, etc.), CNNs can learn them from training data. The main drawback of deep learning methods is that they need to be trained with very large datasets that contain enough variations to generalise to unseen samples. Fortunately, several large-scale face datasets containing in-the-wild face images have recently been released into the public domain to train CNN models. Apart from learning discriminative features, neural networks can reduce dimensionality and be trained as classifiers or using metric learning approaches. CNNs are considered end-to-end trainable systems that do not need to be combined with any other specific methods. CNN models for face recognition are often trained using different approaches. One of them consists of treating the problem as a classification one, wherein each subject in the training set corresponds to a class. After training, the model can be used to recognize subjects that are not present in the training set by discarding the classification layer and using the features of the previous layer as the face representation. In the deep learning literature, these features are commonly mentioned as bottleneck features. Following this first training stage, the model can be further trained using other techniques to optimize the bottleneck features for the target application. Another common approach to learning face representation is to straight learn bottleneck features by optimizing a distance metric between pairs of faces or triplets of faces. An end-to-end face recognition CNN was proposed. The contrastive loss implements a metric learning procedure that aims to minimize the distance between pairs of feature vectors corresponding to the same subject while maximizing the distance between pairs of feature vectors corresponding to different subjects. The CNN architecture used in this method was also shallow and was trained with small datasets.

4.IMPLEMENTATION

Implementation is that the process of converting a replacement or a revised system design into an operational one. The objective is to place the new or revised system that has been tested into operation while holding costs, risks, and private irritation to the minimum. A critical aspect of the implementation process is to ensure that there will be no disrupting the functioning of the organization. The best method for capture while implanting any new system would be to use strategically test for testing all new programs. Before production files are used to test live data, text files must be created on the old system, copied over to the new system, and used for the initial test of every program.Another factor to be considered within the implementation phase is that the acquisition of the hardware and software. Once the software is developed for the system and testing is carried out, it is then the process of making the newly designed system fully operational and consistent in performance. Implementation is that the most vital stage in achieving a successful system and giving the user’s confidence that the new system is workable and effective. Implementation of a modified application to exchange an existing one. This type of conversation is comparatively easy to handle, provide there are not any major changes within the system.

SYSTEM IMPLEMENTATION

There are three major types of implementation are there but the following are proposed for the project. Following are the Algorithms used for this project:

Local Binary Patterns Histogram algorithm:

Local Binary Patterns Histogram algorithm was proposed in 2006. It is based on local binary operator. It is widely used in facial recognition due to its computational simplicity and discriminative power.

The steps involved to achieve this are:

- creating dataset
- face acquisition
- feature extraction
- classification

The LBPH algorithm is a part of opencv.

Steps

- Suppose we have an image having dimentions N x M.
- We divide it into regions of same height and width resulting in m x m dimension for every region.

Local binary operator is used for every region. The LBP operator is defined in window of 3x3.

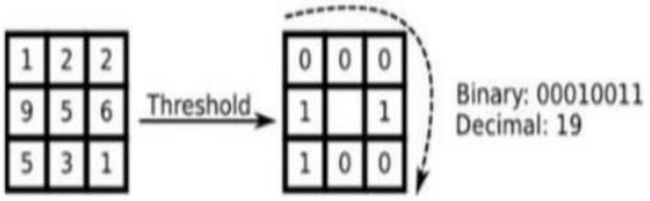
$$\mathrm{LBP}(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

here '(Xc,Yc)' is central pixel with intensity 'Ic'. And 'In' being the intensity of the neighbor pixel

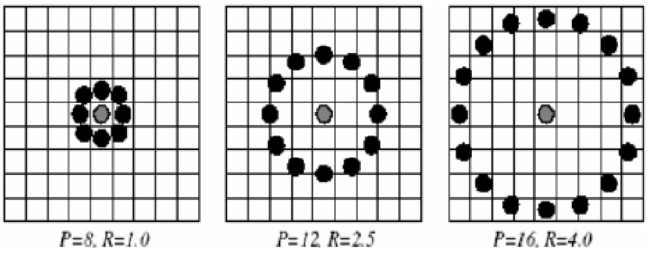
Using median pixel value as threshold, it compares a pixel to its 8 closest pixels using this function.

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

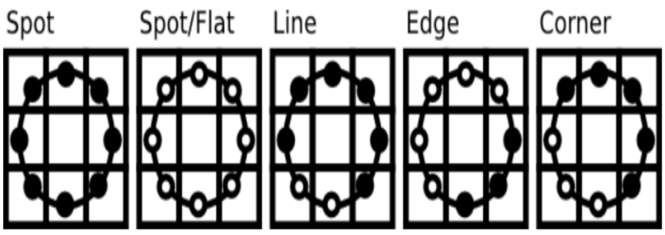
If the value of neighbor is greater than or equal to the central value it is set as 1 otherwise it is set as 0.Thus, we obtain a total of 8 binary values from the 8 neighbors.After combining these values we get a 8 bit binary number which is translated to decimal number for our convenience.This decimal number is called the pixel LBP value and its range is 0-255.



Later it was noted that a fixed neighborhood fails to encode details varying in scale .The algorithm was improved to use different number of radius and neighbors , now it was known as circular LBP.



The idea here is to align an arbitrary number of neighbors on a circle with a variable radius. This way the following neighborhoods are captured:



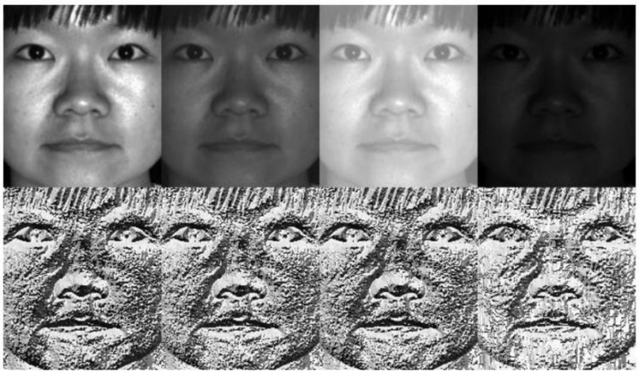
For a given point (Xc,Yc) the position of the neighbor (Xp,Yp), p belonging to P can be calculated by:

$$x_p = x_c + R\cos(\frac{2\pi p}{P})$$
$$y_p = y_c - R\sin(\frac{2\pi p}{P})$$

here R is radius of the circle and P is the number of sample points.If a points coordinate on the circle doesn't correspond to image coordinates, it get's interpolated generally by bilinear interpolation:

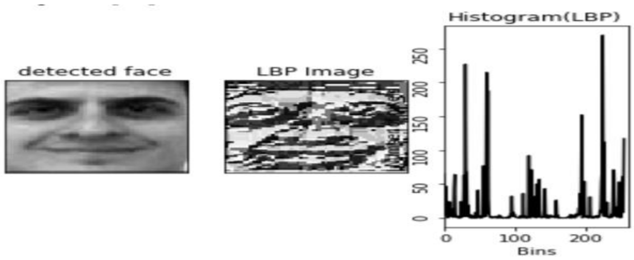
$$f(x,y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}$$

The LBP operator is strong against monotonic gray scale transformations.



After the generation of LBP value histogram of the region is created by counting the number of similar LBP values in the region.

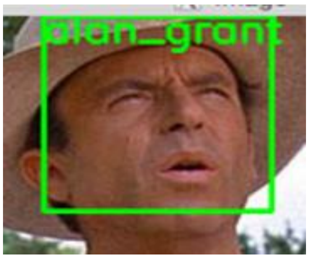
After creation of histogram for each region all the histograms are merged to form a single histogram and this is known as feature vector of the image.



Now we compare the histograms of the test image and the images in the database and then we return the image with the closest histogram.(This can be done using many techniques like Euclidean distance, chi-square, absolute value etc).The Euclidean distance is calculated by comparing the test image features with features stored in the dataset. The minimum distance between test and original image gives the matching rate.

$$d(a,b) = \sqrt{\sum_{i=1}^{n} |a_i - b_i|^2}$$

As an output we get an ID of the image from the database if the test image is recognized.



LBPH can recognize both side and front faces and it is not affected by illumination variations which means that it is more flexible.

PARALLEL CONVERSION TYPE OF IMPLEMENTATION: In this type of implementation both the current system and the proposed system run in parallel. This happens till the user gets the entire confidence on the proposed system and hence cuts of the present system.

4.1. PHASE-IN METHOD OF IMPLEMENTATION

In this sort of Implementation the proposed system is introduced phase-by-phase. This reduces the risk of uncertainty of proposed system.

Each program is tested individually at the time of development using the info and has verified that this program linked together within the way laid out in the programs specification, the computer system and its environment is tested to the satisfaction of the user. The system that has been developed is accepted and proved to be satisfactory for the user. And so the system goes to be implemented very soon. A simple procedure is included in order that the user can understand the various functions clearly and quickly.

Initially as a primary step the executable sort of the appliance is to be created and loaded within the common server machine which is accessible to the whole user and the server is to be connected to a network. The final stage is to document the whole system which provides components and therefore the operating procedures of the system. Implementation is that the stage of the project when the theoretical design is clothed into a working system. Thus it are often considered to be the foremost critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

5. TESTING

Testing is the major process involved in software **qualityassurance (QA).** It is iterative process. Here test data is prepared and is used to test the modules individually. System testing makes sure that all components of the system function properly as a unit by actually forcing the system to fail.

The test causes should be planned before testing begins. Then as the testing progresses, testing shifts focus in an attempt to find errors in integrated clusters of modules and in the entire system. The philosophy behind testing is to find errors. Actually testing is the estate of implementation that is aimed at ensuring that the system works actually and efficiently before implementation.

Testing is done for each module. After testing all the modules, the modules are integrated and testing of the ultimate system is completed with the test data, specially designed to point out that the system will operate successfully in all its aspects conditions. The procedure level testing is made first. By giving improper inputs, the errors occurred are noted and eliminated.

Thus the system testing may be a confirmation that each one is correct and a chance to point out the user that the system works. The final step involves Validation testing, which determines whether the software function because the user expected. The end-user rather than the system developer conduct this test most software developers as a process called "Alpha and Beta test" to uncover that only the highest user seems able to find.

This is the ultimate step in system life cycle. Here we implement the tested error-free system into real-life environment and make necessary changes, which runs in a web fashion. Here system maintenance is completed every months or year supported company policies, and is checked for errors like runtime errors, end of the day errors and other maintenances like table verification and reports.

During the need analysis and style , the output may be a document that's usually textual and non-executable. After the coding phase, computer programs are available which will be executed for testing purpose. This implies that testing not only possesses to uncover errors introduced during coding, but also errors introduced during the previous phases.

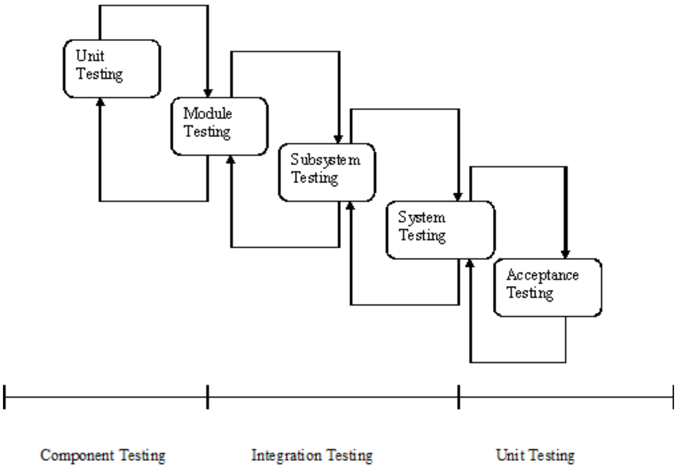


Fig. 7.1 The Testing process

The various sorts of testing done on the system are:

- Unit Testing
- Integration Testing
- Validation Testing
- System Testing
- Acceptance Testing

5.1 UNIT TESTING

Unit testing verification efforts on the littlest unit of software design, module. This is known as "Module Testing". The modules are tested separately. This testing is administered during programming stage itself. In these testing steps, each module is found to be working satisfactorily as reference to the expected output from the module.

5.2 INTEGRATION TESTING

Integration testing may be a systematic technique for constructing tests to uncover error associated within the interface. In the project, all the modules are combined then the whole programmer is tested as an entire . In the integration-testing step, all the error uncovered is corrected for subsequent testing steps.

5.3 VALIDATION TESTING

To uncover functional errors, that is, to see whether functional characteristics inform specification or not specified.

5.4 SYSTEM TESTING

Once individual module testing completed, modules are assembled to perform as a system. Then the top down testing, which begins from upper level to lower level module testing, has to be done to check whether the entire system is performing satisfactorily.
After unit and integration testing are over then the system as whole is tested. There are two general strategies for system testing.

They are:

- Code Testing
- Specification Testing

5.4.1 Code Testing

This strategy examines the logic of the program. A path is a specific combination of conditions handled by the program. Using this strategy, every path through the program is tested.

5.4.2 Specification Testing

This strategy examines the specifications stating what the program should do and how it should perform under various conditions. The test cases are developed for each condition of developed system and processed. It is found that the system developed perform according to its specified requirements. The system is used experimentally to ensure that the software will run according to tits specification and in the way user expect.

Specification Testing is done successfully by entering various types of end data. It is checked for both valid and invalid data and found System is working properly as per requirement.

5.5 ACCEPTANCE TESTING

When the system has no measure problem with its accuracy, the system passes through a final acceptance test. This test confirms that the system needs the original goal, Objective and requirements established during analysis. If the system fullfils all the requirements, it is finally acceptable and ready for operation.

5.6 TEST PLAN

A software project test plan may be a document that describes the objectives, scope approach and focus of a software testing effort. This process of preparing a test plan may be a useful thanks to think through the efforts needed to validate the acceptability of a software package . The completed document will help the people outside the test group understand 'Why and How' of production validation. Different test plans are used at different levels of testing.

5.6.1 Test Plans used in Unit Testing

Each module is tested for correctness whether it is meeting all the expected results. Condition loops in the code are properly terminated so that they don't enter into an infinite loop. Proper validations are done so as to avoid any errors related to data entry from user.

TEST CASES

TestCase Number	**Testing Scenario**	**Expected result**	**Result**
	Upload image		
TC -01	User upload image if face object found	Detect face	Pass
Tc – 02	If image not uploaded properly	Not detect	Pass
	If upload image other than face image like nature image etc..	Note detect	Pass
	Comparison		
TC – 01	Compare object with trained dataset	Detect fake face	Pass
	Compare object with trained dataset if object not found	Not detect	Pass

6. CONCLUSIONS

In view of problem of low accuracy of different in painting methods by using different algorithms and technologies we have introduced a technique based on deep learning using techniques like face detection, face alignment, face representation, face matching and validating our technique with real time dataset such asTufts and Olevit dataset to fill the missing ormasked regions in images with plausibly synthesized content. And it can able to generate realistic and semantically plausible images.

ACKNOWLEDGEMENT

The authors would like to express their gratitude to Mrs.Drakshayini KB,Assistant Professor, Department Of ISE, VVIET, for his valuable guidance and modifications to improve the quality of paper. And Dr. Madhu B K, Head of the Department, VVIET, for his constant support and suggestions. Also we acknowledge the efforts and hard work by the experts who have contributed towards development of different Vehicle Movement light Controller system.

REFERENCES

[1]Y. C. Feng, P. C. Yuen, and A. K. Jain, “A hybrid approach for generating secure and discriminating face template,” IEEE Transactions on Information Forensics and Security, 2017.

[2] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” in CVPR, 2018.

[3] T. Ahonen, A. Hadid, and M. Pietikainen, “Face description with local binary patterns: Application to face recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016