



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 2, February 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423



9940 572 462



6381 907 438



ijirset@gmail.com



www.ijirset.com

Facial Recognition for Identical Twins using Image Segmentation

Mohana Priya M^{*1}, Dr.M.Kavitha^{*2}, Dr.A.Charles^{*3}, R.Manikumar^{*4}, T.Mohandoss^{*5}, V.Rajesh Kannan^{*6}

ME Student, Department of Electronics and Communication Engineering, Government College of Engineering

(Autonomous), Bargur, Krishnagiri, Tamil Nadu, India. ^{*1}

Associate Professor, Department of Electronics and Communication Engineering, Government College of Engineering

(Autonomous), Bargur, Krishnagiri, Tamil Nadu, India. ^{*2}

Assistant Professor, Department of Electronics and Communication Engineering, Government College of Engineering

(Autonomous), Bargur, Krishnagiri, Tamil Nadu, India. ^{*3,4,5,6}

ABSTRACT: This abstract discusses the increasing importance of face recognition in computer vision and biometrics, with a specific focus on "twin face recognition" dedicated to distinguishing between identical twins using advanced image processing and machine learning. The paper emphasizes the development of a twin face recognition system, employing sophisticated algorithms to accurately classify twin faces. The methodology addresses challenges in differentiating genetically identical individuals, making it applicable in security, access control, and forensics.

The study explores the difficulties of twin face recognition, including the need to overcome inherent similarities. Advanced machine learning models are utilized to capture nuanced differences, enhancing the system's robustness in real-world scenarios. The paper also discusses potential applications in security protocols and forensic investigations.

In conclusion, the proposed machine learning-based twin face recognition system shows promising results in distinguishing between identical twins, contributing to the broader fields of biometrics and computer vision. This specialized solution has applications in security, access control, and forensics, where accurate identification of identical twins is crucial. The research adds valuable insights to the nuanced process of classifying twin faces and underscores the significance of this technology in various domains.

KEYWORDS: Face recognition, facial marks, identical twins.

I. INTRODUCTION

This introduction delves into the intricate domain of twin face recognition, a sophisticated facet of facial recognition technology designed to differentiate between identical twins based on their facial features. Identical twins, sharing nearly identical genetic makeup, pose a unique challenge for traditional facial recognition systems due to their striking resemblances. The paper explores the innovative landscape of twin face recognition systems, employing cutting-edge algorithms and deep learning techniques to discern imperceptible facial differences.

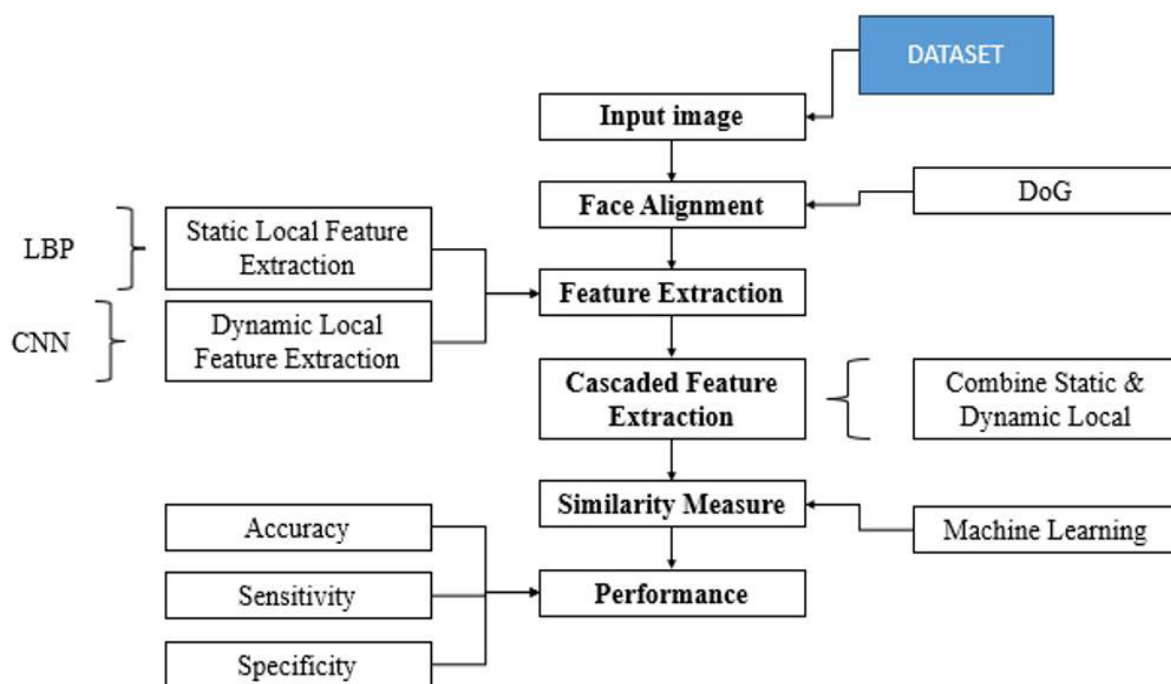
These systems analyze characteristics such as eye shape, freckle placement, and facial expressions to generate unique biometric profiles for each twin. The technology extends beyond security, authentication, and border control, offering insights into genetic research and facial variations among genetically identical individuals.

Twin face recognition has the potential to revolutionize industries like law enforcement and access control, providing an accurate means of distinguishing between identical twins. The introduction also touches upon the current landscape of facial expression recognition, emphasizing dynamic approaches in understanding natural facial events. The subsequent sections discuss limitations of existing models and introduce a novel Hidden Conditional Ordinal Random Field (H-CORF) model for improved prediction performance in real video classification problems.

II. RELATED WORK

In [1] Author demonstrates that modelling the joint constraints of codeword counts across the video is expressive enough to capture the stochastic structure of rather complex activities. In [2] Author's goal is to unify the classification and localization components into a single detection procedure. This means we can efficiently identify both the spatial and temporal region(s) within the sequence that best fit a learned activity model. In [3] Author seeks to learn what activity parts and their spatiotemporal relations should be captured to represent complex human activities, and how relevant they are for enabling efficient inference in realistic videos. In [4] Author approaches in this direction is to treat video frames as still images and apply CNNs to recognize actions at the individual frame level. In [5] Event detection algorithms operate by evaluating a classifier function at many different segments of the video and then predicting the event presence in segments with high-score. In [6] Author's Recent studies observe that it is quite time intensive to train deep learning algorithms to yield state of the art results. In [7] Author presents a hierarchical model for human activity recognition in entire multi-person scenes. In [8] Author presents a framework for the automatic recognition of complex multi-agent events in settings where structure is imposed by rules that agents must follow while performing activities. In [9] Author proposed a novel motion saliency detection method based on low-rank and sparse decomposition, which provides many video processing tasks, such as video segmentation and adaptive content delivery, with a powerful video pre-processing technique. In [10] Author analyses still images to produce primary visual features (including intensity, colour opponents, edge orientations and motion information), which are combined to form a saliency map that represents the relevance of visual attention.

III. SYSTEM ARCHITECTURE



The process of reading an image into the workspace using the `imread` command is described. The example reads a sample image, stores it in an array named `I`, and displays it using the `imshow` function. The Image Viewer app is introduced as a tool for image display and processing tasks.

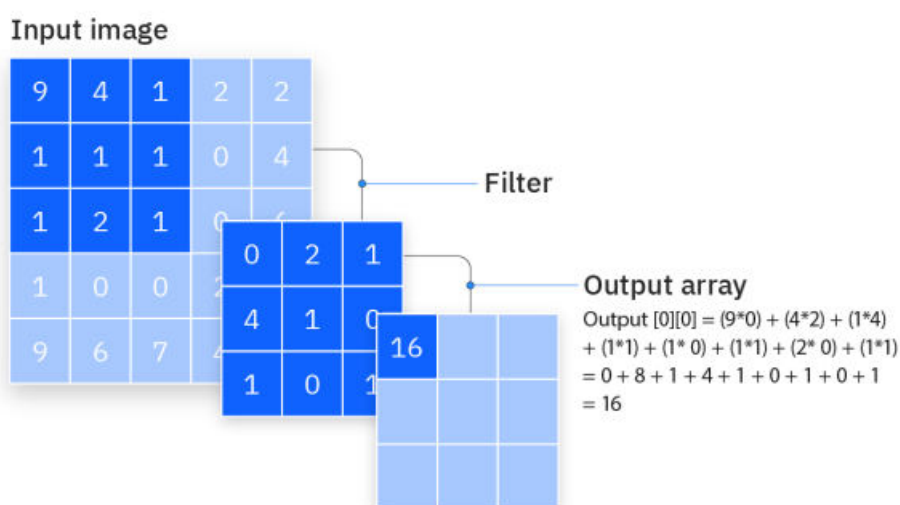
Preprocessing and emphasizing noise reduction in both analog and digital recording devices. It explains the types of noise, such as random and coherent noise, and highlights electronic devices' susceptibility to hiss caused by stray electrons.

Face detection technology used in various applications. It describes the Viola-Jones object detection framework, initially proposed for face detection in 2001, and mentions its adaptability for detecting other object classes. The

concept of feature extraction in pattern recognition and machine learning is discussed. Three feature extraction methods, LBP, CNN, and a method involving histogram computation, are explained. The LBP method involves dividing the examined window into cells, comparing pixel values with neighbors, creating a binary number, and generating a histogram for each cell, ultimately forming a feature vector for the entire window.

IV. PROPOSED ALGORITHM

CNN :Convolutional neural networks (CNNs) excel in processing image, speech, or audio signals with three main layers: Convolutional, Pooling, and Fully-connected (FC). The CNN's complexity grows layer by layer, initially focusing on simple features like colors and edges, progressively recognizing larger elements or shapes in the image. The Convolutional layer, the core of a CNN, performs computation using input data, filters, and feature maps. A feature detector, a 2D array of weights, moves across the image to perform convolution, generating a feature map. Parameters like filter size, stride, and zero-padding influence the output volume. Three hyperparameters—number of filters, stride, and zero-padding type—affect the output size. After convolution, a Rectified Linear Unit (ReLU) transformation introduces nonlinearity. The CNN structure involves additional convolution layers, creating a hierarchical feature hierarchy. Pooling layers reduce dimensionality, employing filters without weights. Two main pooling types are Max pooling (selecting the maximum value) and Average pooling (calculating the average value). While pooling loses some information, it enhances CNN efficiency and reduces complexity, limiting overfitting risk. The Fully-connected layer connects each node in the output layer directly to the previous layer, performing classification based on extracted features. ReLU functions are used in convolution and pooling layers, while FC layers typically employ a softmax activation function for probability output (0 to 1).



An additional convolutional layer in a CNN can follow the initial layer, forming a hierarchical structure as later layers perceive pixels within the receptive fields of prior layers. For instance, in identifying a bicycle within an image, the CNN views the bicycle as a composition of parts like frame, handlebars, wheels, and pedals. Each part represents a lower-level pattern, and their combination forms a higher-level pattern, establishing a feature hierarchy. This convolutional layer ultimately transforms the image into numerical values, enabling the neural network to interpret and extract relevant patterns.



Pooling layers, conducting dimensionality reduction in a CNN, employ a filter without weights to sweep across the input. The kernel applies an aggregation function within the receptive field, populating the output array. Two main pooling types are Max pooling, selecting the maximum value, and Average pooling, calculating the average value. While the pooling layer loses information, it offers benefits by reducing complexity, improving efficiency, and limiting overfitting risk. The Fully-connected layer, distinct from partially connected layers, directly connects each node in the output layer to the previous layer. This layer classifies inputs based on features extracted through prior layers and their filters. While ReLU functions are common in convolutional and pooling layers, FC layers usually utilize a softmax activation function, producing a probability from 0 to 1.

V. SOFTWARE USED

SOFTWARE REQUIREMENTS:

OS : WINDOWS 7,8,10,11

SOFTWARE : MATLAB 2023B

HARDWARE REQUIREMENTS:

PROCESSOR : INTEL PENTIUM.

RAM : 8 GB

VI. PSEUDO CODE

```
import numpy as np
```

```
class ConvolutionalLayer:
```

```
    def __init__(self, input_shape, num_filters, filter_size, stride, padding):
```

```
        self.input_shape = input_shape
```

```
        self.num_filters = num_filters
```

```
        self.filter_size = filter_size
```

```
        self.stride = stride
```

```
        self.padding = padding
```

```
        self.weights = np.random.randn(num_filters, filter_size, filter_size)
```

```
    def apply_convolution(self, input_data):
```

```
        # Implement convolution operation
```

```
        pass
```

```
class PoolingLayer:
```

```
    def __init__(self, pool_type, pool_size):
```

```
        self.pool_type = pool_type
```

```
        self.pool_size = pool_size
```

```
    def apply_pooling(self, input_data):
```

```

# Implement pooling operation
pass

class FullyConnectedLayer:
    def __init__(self, input_size, output_size):
        self.input_size = input_size
        self.output_size = output_size
        self.weights = np.random.randn(output_size, input_size)

    def apply_fully_connected(self, input_data):
        # Implement fully-connected operation
        pass

class CNN:
    def __init__(self):
        self.conv1 = ConvolutionalLayer(input_shape=(3, 64, 64), num_filters=32, filter_size=3, stride=1,
padding='same')
        self.pool1 = PoolingLayer(pool_type='max', pool_size=2)
        self.conv2 = ConvolutionalLayer(input_shape=(32, 32, 32), num_filters=64, filter_size=3, stride=1,
padding='same')
        self.pool2 = PoolingLayer(pool_type='max', pool_size=2)
        self.fc1 = FullyConnectedLayer(input_size=4096, output_size=128)
        self.fc2 = FullyConnectedLayer(input_size=128, output_size=10)

    def forward_pass(self, input_data):
        # Implement the forward pass through the network
        pass

# Example usage
cnn_model = CNN()
input_data = np.random.randn(3, 64, 64) # Example input image
output_probabilities = cnn_model.forward_pass(input_data)
print(output_probabilities)

```

VII. SIMULATION RESULTS

Convolutional neural networks (CNNs) stand out among other neural networks for their exceptional performance in handling image, speech, or audio signal inputs. Comprising three main layers – convolutional, pooling, and fully-connected (FC) layers – CNNs exhibit increasing complexity as they process data, identifying intricate details within an image. While the convolutional layer serves as the core building block, the final layer is the fully-connected layer.

In the convolutional layer, fundamental computations take place, involving input data, a filter, and a feature map. A feature detector (or kernel) moves across the image, performing a convolution process by calculating dot products between the filter and input pixels. This generates a feature map, representing detected features. The weights in the feature detector remain fixed during convolution, a concept known as parameter sharing.

Three crucial hyperparameters – the number of filters, stride, and zero-padding – influence the output volume. Filters determine the depth of the output, stride controls the kernel movement, and zero-padding adjusts output size. Padding types include valid (no padding), same (output matches input size), and full (increased output size).

Post-convolution, a Rectified Linear Unit (ReLU) transformation introduces nonlinearity to the model, contributing to its ability to capture complex relationships. This comprehensive overview elucidates the intricate workings of CNNs and their pivotal role in image recognition and signal processing tasks.

Pooling layers, a form of down sampling, reduce input parameters through a filter sweep, aggregating values within the receptive field. Two main pooling types are max pooling (selecting the maximum pixel value) and average pooling

(calculating the average value). While information loss occurs, pooling benefits include complexity reduction, enhanced efficiency, and prevention of overfitting in CNNs.

The fully-connected layer, in contrast, connects each node in the output layer directly to the previous layer's nodes. Responsible for classification based on extracted features, it utilizes soft max activation for proper input categorization, generating probabilities from 0 to 1. This layer contrasts with partially connected layers, contributing to a comprehensive understanding of CNN architecture and its role in efficient image processing and classification tasks.

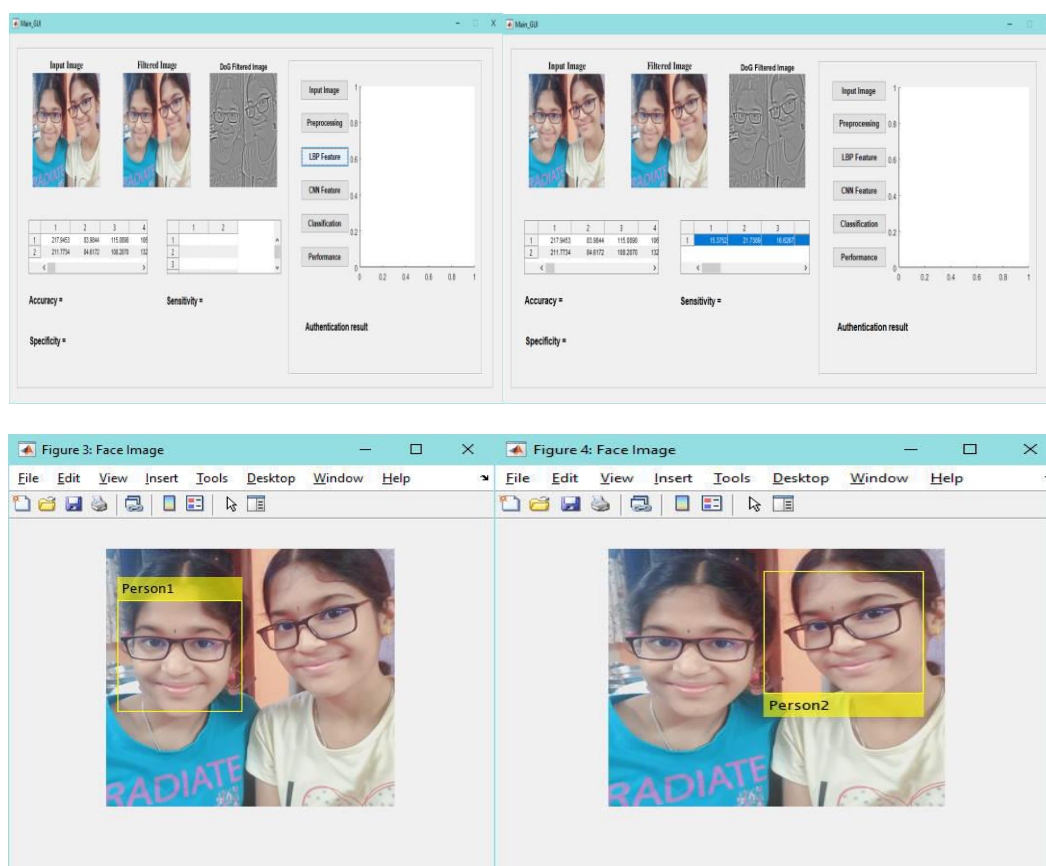


Figure 1: Simulation output

VIII. CONCLUSION AND FUTURE WORK

In this section we present the proposed approach for the task of texture classification. First, we introduce RLBP operator obtained by circularly shifting the weights of LBP operator. Further, the intrinsic structure of the patterns is utilized by incorporating the principle of uniform patterns to generate uniform RLBP (uRLBP). The dominant direction is defined as the index in the circular neighbourhood for which the difference is maximum. As an image undergoes a rotation the dominant direction in a neighbourhood also undergoes the rotation by the same angle.

REFERENCES

1. Hangyu Li, Nannan Wang , Member, IEEE, Xinpeng Ding , Xi Yang , Member, IEEE, and Xinbo Gao , Senior Member, IEEE. 'Adaptively Learning Facial Expression Representation via C-F Labels and Distillation' IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 30, 2021.
2. Jeffrey R. Paone, Patrick J. Flynn, Fellow, IEEE, P. Jonathon Philips, Fellow, IEEE, Kevin W. Bowyer, Fellow, IEEE, Richard W. Vorder Bruegge, Patrick J. Grother, George W. Quinn, Matthew T. Pruitt, and Jason M. Grant.



‘Double Trouble: Differentiating Identical Twins by Face Recognition’ IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 2, FEBRUARY 2014.

3. Delphine Poux, Benjamin Allaert, Nacim Ihaddadene, Ioan Marius Bilasco, Chaabane Djeraba, and Mohammed Bennamoun, Senior Member, IEEE. **‘Dynamic Facial Expression Recognition Under Partial Occlusion With Optical Flow Reconstruction’** IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 31, 2022.

4. Nisha Srinivas, Gaurav Aggarwal, Patrick J. Flynn, Fellow, IEEE, and Richard W. Vorder Bruegge.

‘Analysis of Facial Marks to Distinguish Between Identical Twins’ IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 7, NO. 5, OCTOBER 2012.

5. Zengqun Zhao, Graduate Student Member, IEEE, Qingshan Liu, Senior Member, IEEE, and Shanmin Wang. 57

‘Learning Deep Global Multi-Scale and Local Attention Features for Facial Expression Recognition in the Wild’ IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 30, 2021.

6. VIJAYALAKSHMI G. V. MAHESH, CHENGJI CHEN, VIJAYARAJAN RAJANGAM, (Member, IEEE), ALEX NOEL JOSEPH RAJ, (Member, IEEE), AND PALANI THANARAJ KRISHNAN, (Member, IEEE).

‘Shape and Texture Aware Facial Expression Recognition Using Spatial Pyramid Zernike Moments and Law’s Textures Feature Set’ IEEE ACCESS Received March 8, 2021, accepted March 21, 2021, date of publication March 30, 2021, date of current version April 12, 2021. Digital Object Identifier 10.1109/ACCESS.2021.3069881.

7. Chunlei Peng, Member, IEEE, Nannan Wang, Member, IEEE, Jie Li, and Xinbo Gao, Senior Member, IEEE.

‘Universal Face Photo-Sketch Style Transfer via Multiview Domain Translation’ IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 29, 2020.

8. Son Lam Phung, Abdesselam Bouzerdoum, and Douglas Chai **‘Skin Segmentation Using Color And Edge Information’** IEEE ISSPA ISBN: 0-7803-7946-2 2003.

9. Hwei-Jen Lin, Shu-Yi Wang, Shwu-Huey, and Yang -Ta-Kao **‘Face Detection Based on Skin Color Segmentation and Neural Network’** IEEE Transactions on, Volume: 2, pp1144- 1149, ISBN: 0-7803 9422-4, 2005.

10. R. M. Haralick, J. Shanmugam, and I. Dinstein, **‘Texture feature for image classification’** IEEE Transactions on Systems, Man and Cybernetics, 3, 610-621, 1973.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details