www.ijarp.com ISSN 2456-9992 Page: 01-10

International Journal Advanced Research Publications

Research Article

Volume: 02
Issue: 09

DETECT THE HUMAN FACE AND EMOTIONS EFFECTIVELY USING DEEP LEARNING TO RECOGNISE AND ANALYSE BEHAVIOURAL PATTERNS

Dr. Harsha V. Patil*

Department of Computer Application, MAEER's MIT Arts Commerce and Science College Alandi (D) Pune, Alandi-Moshi Road, Pune, 412105, Maharashtra, India.

Article Received: 16 August 2026, Article Revised: 06 September 2026, Published on: 26 September 2026

***Corresponding Author: Dr. Harsha V. Patil**
Department of Computer Application, MAEER's MIT Arts Commerce and Science College Alandi (D) Pune, Alandi-Moshi Road, Pune, 412105, Maharashtra, India.
Doi: https://doi-doi.org/101555/ijrpa.7424

ABSTRACT

All person's emotions are the building blocks of their communication. Interactive video games, medical diagnostics, emotional computing, human-robot interaction, and other areas of Human Computer Interaction (HCI) make extensive use of it. There are three common ways that people show their emotions in social situations: gestures, facial expressions, and words (unimodal), speech and facial expression (bimodal), and psychological signals, video, and audio (multimodal). Understanding one's surroundings and the intentions of one's interlocutors are prerequisites for machine perception. This could be helped by facial emotion recognition. Images depicting facial emotions such as joy, sorrow, rage, surprise, disgust, and fear were processed using deep learning approaches. The Extended Cohn-Kanade Database was used for training. Deep learning enables machines to perceive features effectively, improving human-machine interaction and enhancing user experience.

KEYWORDS: Deep Learning, Pattern Recognition, Emotion, HCI, Model.

1 INTRODUCTION

Both theoretically and in terms of practical applications, face recognition has grown at a phenomenal rate [4], [7]. During the age of deep learning, state-of-the-art models based on Deep Convolutional Neural Networks (DCNN) like ResNet and OpenFace have achieved remarkable success [8], [14]. Still, facial recognition systems are susceptible to confounds like disguises, even if these algorithms are becoming better at what they do [4]. The field of

face recognition has done extensive research on a number of factors, including lighting, expression, age, and heterogeneity [2], [10]. Nevertheless, there has been a dearth of research into facial identification using disguise variants.

Changing one's appearance by donning a beard, spectacles, a variety of haircuts, a scarf, a hat, or other head coverings is an example of a disguise. Disguises may be either deliberate or accidental. In deliberate disguise, the subject makes an effort to conceal their identity in order to trick face recognition systems; in unintentional disguise, they casually wear various disguises, like glasses and a scarf, and manage to trick the system without realising it. These differences make the job more difficult by decreasing the interclass distance between participants and increasing the intraclass variances [4], [10].

In order to communicate and engage with others, the human face is essential. In order to do these tasks, humans rely on their facial recognition abilities. One must learn to identify the faces of the hundreds of people that surround them every day. Facial recognition technology is also crucial for HCI [5], [8]. Access control, security, and surveillance systems, as well as the entertainment sector, make extensive use of it nowadays [2], [7]. Faster and simpler tasks are now possible thanks to advancements in facial recognition.

Two main subfields exist within the field of facial recognition: identification and verification. Face verification is a one-to-one matching method that just requires two photos to determine whether they belong to the same individual. In contrast, face identification 1:N matching requires selecting a specific individual from a pool of potential results in order to identify them in an image [10], [12].

Face recognition refers to the method of identifying a person by analysing their facial characteristics. Deep learning has completely transformed face recognition by making possible very accurate and efficient recognition systems [2], [8], [12]. Deep learning face recognition systems typically employ a large dataset of facial images to train a neural network that can identify unique facial characteristics and patterns [5], [8]. Using these characteristics, a unique representation of each person's face is created and stored in a database for future recognition. When doing recognition, the system compares the incoming face picture to its stored representations to get the closest match. Convolutional Neural Networks (CNNs) are often used for feature extraction in face recognition systems, with deep learning-based techniques such as Siamese networks and triplet networks employed for matching optimisation [14], [15].

Among the many applications of face recognition technology are identification, social media tagging, access control, security, and monitoring [7], [10]. It is critical, however, to consider privacy concerns and any biases throughout the design and implementation of such systems.

Amid this article, we'll go over the methodology that can identify a face amid a sea of data that constantly compares and saves information about individuals we know. A good solution would be a classifier that can automatically recognise faces. Faces are usually represented as a linear combination of base photographs using these methods. Principal Component Analysis (PCA) is one example of such a method [7]. To get the basis pictures, Principal Component Analysis (PCA) only examines pairwise correlations between pixels in the image collection. Independent Component Analysis (ICA) is different from Principal Component Analysis (PCA), which is often used in computerized face recognition and authentication applications [3].

In order to identify and localize faces, face recognition systems use the Haar feature-based cascade classifier. The features of the face are recovered by use of the weighted Local Binary Pattern method. The proposed embedded face recognition system achieves a CMC accuracy of 99.33.

2 Literature Review

Recent studies have shown that identifying siblings using facial recognition remains a challenging task because of the strong resemblance between facial features. Various deep learning models such as FaceNet, VGGFace, VGG16, and VGG19 have been evaluated using similarity measures including cosine similarity, Manhattan distance, Euclidean distance, and Minkowski distance. Experimental results demonstrated that VGGFace achieved more than 95% accuracy.

Face recognition is considered one of the most challenging computer vision problems due to variations in lighting, aging, facial accessories, and facial expressions. Deep learning approaches based on Convolutional Neural Networks (CNNs) have significantly improved recognition accuracy by learning rich facial representations from large datasets. Experimental studies conducted on multiple datasets containing male and female subjects confirmed that CNN-based systems achieve high recognition accuracy and outperform conventional approaches in both training and prediction phases [2].

Recent advancements in deep learning have transformed face recognition technologies by improving recognition accuracy and computational efficiency. Artificial intelligence chips such as Tensor Processing Units (TPUs) accelerate deep learning computations and make

real-time facial recognition feasible on mobile and embedded devices. Traditional approaches such as Principal Component Analysis (PCA) and AdaBoost were found to be less effective than deep learning-based approaches, especially for large-scale facial recognition applications [7].

Single sample per person verification has gained considerable importance in applications such as surveillance, border security, e-voting, and banking systems. Deep learning models designed for low-resource and mobile environments have demonstrated promising performance under varying environmental conditions. Comparative evaluations using public and private datasets showed that lightweight deep learning architectures can provide efficient and accurate facial recognition even when only one image per person is available [10].

Behavioral analysis using deep learning and machine vision has achieved performance comparable to human-level observation. Advanced frameworks such as DeepLabCut have enabled precise tracking of body parts without using physical markers. Experimental evaluations conducted on behavioral tests demonstrated that machine learning classifiers can outperform commercial behavioral tracking systems in terms of accuracy, flexibility, and cost-effectiveness. Furthermore, the proposed approaches reduced inter-annotator variability and improved automated behavioral analysis capabilities [11].

3 Research Methodology

First phase

Here we showcase the work that revolves around the CK+ dataset [8], [10]. Included in this procedure are steps to prepare CK+ photos for processing, establish a network topology, feed the network with the updated CK+ images, train the network, adjust parameters, and assess the network's correctness [14].

CK+ image pre-processing

The next paragraphs provide the picture pre-processing. A binary file is produced from the CK+ picture dataset, which is used as input. When training the network, the binary file was fed into it. The experimental phase of the project started at this point. Problems started to emerge at the outset due to the folder structure of the dataset and the lack of labels for some picture sequences. The walk Python method was used to transfer all picture files from the CK+ tree directory to a single location. Since the filename was sufficient to associate each photograph with its respective sequence and topic, this was indeed feasible. The label files were transferred using a similar method. In addition, a new folder was created for all picture sequences that did not have labels [5], [8].

Data input into TensorFlow (TF)

Records are read into TensorFlow (TF) from a binary file that contains the pictures and their labels. A queue containing all the training samples was generated by this procedure.

```
filenames = [os.path.join(PATH, FILENAME)]
# Create a queue that produces the filenames to read.
filename_queue = tf.train.string_input_producer(filenames)
# Read examples from files in the filename queue.
read_input = read(filename_queue)
```

The key and value are then subjected to casting operations.

Key: This is the label's value after being converted to an integer.

Value: That tensor that contains the image tensor.

The original dimensions of the tensor are changed to height, breadth, and depth later on. Data augmentation may now be done by applying transforms to the picture.

Network description

The intended topology is based on the one established by the Visual Geometry Group (VGG) at Oxford University [14]. The VGG network, which highlighted the significance of network depth in improving classification accuracy, was awarded second place in the ImageNet 2014 competition [14]. Comparing VGG to various designs, it was shown to generalise effectively to different workloads and data sets [8]. In contrast to other cutting-edge topologies like GoogleNet, this network prioritises the use of pure convolutional network designs, which led to its selection [14].

Frameworks

Caffe: Initially, Caffe was Yangqing Jia's PhD thesis at the Berkeley Vision and Learning Centre (BVLC). In recent times, Caffe has become a project run by BVLC, with several contributors drawn from its expanding community. Python and C++ were the languages used to create it, and it is compatible with Windows, OS X, and Linux. For computer vision calculations, Caffe was primarily developed [8].

Theano: The University of Montreal's Montreal Institute for Learning Algorithms is where Theano was first developed. These days, it's an open-source project with a large user base. When it comes to certain graph calculations, the Python package known as Theano may generate instructions for either CPUs or GPUs. These instructions outperform pure Python by a wide margin and are more in line with C's performance. Theano is primarily concerned with mathematical expressions, particularly those using tensors. In addition, the Theano compiler uses a wide variety of optimisation strategies to produce operation-specific C code [7].

TensorFlow: One open-source machine learning library is TensorFlow (TF), which is developed in C++ and Python. Its publication on November 15th garnered a lot of attention from the media. The Google Brain Team was responsible for developing TF, which is the key cause [5]. A number of Google products have already benefited from TF's enhanced performance in certain areas. Some examples of such functions include Google Now's voice recognition, Google Photos' search capabilities, and Inbox by Gmail's smart reply functionality [5], [8].

4 Data Analysis

An overview of the parameters is shown in Table 1. Unless otherwise specified, the tests on this phase are carried out in this setup.

Network loss and learning rate

Figure 1's loss chart illustrates the minimization of the cost function during training [8], [14]. The loss is minimized steadily throughout the first 450 steps. The curve begins to converge beyond that point. In steps 800 and 900, the loss values are 0.79 and 0.77, respectively. Reducing it by 0.02 on 100 steps is all it is. The model has converged if it exhibits this behaviour [14]. So, training for a longer period won't significantly cut the network's loss.

One example is shown in Figure 2 by using a reduced learning rate of 0.01. This results in a distinct curve. A longer training period is required since the model has not converged yet. Overfitting occurs when a network is exposed to training for too long, even if a longer training period may increase the network's accuracy [8], [15]. Approximately 10 minutes and 37 seconds was the average amount of time spent training.

Table 1: Experiment setup description for the first stage.

Parameter	Value
Batch size	64
Output channels	64
Kernel size	3 × 3
Kernel stride	1
Padding	same
Pooling	max pooling
Pool size	3 × 3
Pool stride	2
FC1 neurons	384
FC2 neurons	192
Dropout	0.99
Activation function	ReLU
Number of classes	6
Cost function	cross entropy
Optimizer	momentum
Momentum	0.9
Learning rate	0.1

When minimising the cost function, the significance of the learning rate is seen in Figures 1 and 2 [14]. On top of that, it offers a solid method for calculating the learning time. Preventing network overfitting requires a precise training period [8], [15]. When a network is overfit, its generalizability suffers. To generalise here is to make predictions on non-training cases with the same level of accuracy as on training examples. Lessening the amount of time spent training does have certain practical advantages. A network makes it easy to test out novel concepts, which is one of them. Cutting down on waste is yet another perk. A big research institution, where a lengthier training period results in poorer production for multiple personnel, might further detect this influence. DL's usage of GPUs demonstrates the community's genuine interest in reducing training time [7], [8].

Explosion of losses is another possible outcome of the learning rate. This occurs when the network's learning rate is set to a value that is too high. A problem that often arose at the start of this endeavour was loss explosion. We tried a few different learning rate numbers, but all it did was make it take longer. Returning the softmax layer's output as a probability distribution solved this issue [14], [15].

Classification accuracy

With respect to the training set, Table 2 details the network's accuracy. Classification based on Top-1 and Top-3 prediction yielded the stated findings. Top-1 forecasts had an average accuracy of 70.54%.

Table 2: Accuracy of network classification on six emotion categories on the CK+ dataset with a learning rate of 0.1.

Top	1	2	3	4	5	6	7	8	9	10
1	70.0	71.1	72.5	70.8	68.1	71.9	71.1	71.3	68.4	70.2
3	75.8	78.4	77.8	77.7	80.0	77.8	78.7	76.4	79.1	77.3

Table 3: Comparison of results with suggested standards.

Model	Accuracy (%)
6-label random guess	16.67
Most frequent label	25.38
EmotiW 2014's winner	39.35
EmotiW 2015's winner	44.70
3 most frequent labels	64.50
Top-1 prediction	70.54
Top-3 prediction	77.90
Ramachandran	97.00

Table 4: Accuracy of the network with a 0.5 dropout.

Top	1	2	3	4	5	6	7	8	9	10
1	56.2	57.0	58.8	58.1	58.9	55.6	57.3	57.7	55.9	56.9
3	75.9	76.9	78.9	77.2	78.6	77.5	78.7	78.0	77.7	76.1

Due to the usage of distinct corpora, a direct comparison with EmotiW may be challenging. A few suggestions for enhancing its categorization accuracy were offered by the 2014 winner. One of them aimed to test out a handful of challenging categories, while the other sought to discover a more efficient fusion approach. Using dropout improved the accuracy performance of our investigation, as will be detailed later. Their experiments do not include the use of dropout. The disparity between the top three forecasts and the top one is less than anticipated. A discrepancy of 7.36 percent is typical.

Table 5: Accuracy of classification using the Adam optimizer.

Top	1	2	3	4	5	6	7	8	9	10
1	64.4	62.2	67.0	64.2	66.6	64.8	66.4	65.9	65.2	63.4
3	75.3	75.8	75.5	74.7	75.6	76.1	76.1	78.1	76.2	76.1

Table 6: Accuracy of classification with the FTRL optimizer.

Top	1	2	3	4	5	6	7	8	9	10
1	18.6	18.9	18.6	20.0	20.3	18.0	18.9	19.2	19.8	18.3
3	78.9	79.2	79.5	79.7	78.9	79.2	79.1	78.7	78.3	78.7

5 CONCLUSIONS

In this paper, one of the most fascinating and difficult tasks is emotion identification. The purpose of this study was to provide a method for applying deep learning to categorise facial emotions from static photographs of faces. There have been several attempts at solving this complicated issue using various methods. Feature engineering has yielded satisfactory results, but feature learning one of DL's promises was the focus of this effort. The use of engineered features is becoming standard in face emotion recognition algorithms.

A major restriction, the absence of a comprehensive dataset of emotions, means that a feature-learning-only solution is still a ways off. As an example, the dataset used in the ImageNet contest included 14,197,122 photos. A bigger dataset would allow for the implementation of networks with a greater capacity to learn features. So, deep learning methods might be used to categorize emotions. Results weren't quite cutting edge, but they were an improvement over competing methods like feature engineering. What this suggests is that with enough labelled samples, DL approaches will be able to tackle this issue in the long run.

REFERENCES

1. M. Agarwal, V. K. Bohat, M. D. Ansari, A. Sinha, S. K. Gupta, and D. Garg, "A convolution neural network-based approach to detect the disease in corn crop," in 2019 IEEE 9th International Conference on Advanced Computing (IACC), 2019, pp. 176–181.
2. M. Y. A. Ahmed and A. Wagdy, "Face Recognition with Deep Learning," 2019. doi: 10.13140/RG.2.2.35787.05920.
3. D. K. R. Gaddam, M. D. Ansari, and S. Vuppala, "On sudoku problem using deep learning and image processing technique," in ICCCE 2020, Singapore: Springer, 2020, pp. 1405–1417.
4. R. Goel, I. Mehmood, and H. Ugail, "A study of deep learning-based face recognition models for sibling identification," Sensors, vol. 21, no. 15, p. 5068, 2021, doi: 10.3390/s21155068.
5. R. Kumar, P. Sharma, and A. Singh, "Efficient CNN architectures for real-time face recognition and emotion detection," Expert Systems with Applications, vol. 238, p. 121945, 2024, doi: 10.1016/j.eswa.2023.121945.
6. S. Kumar, M. D. Ansari, V. K. Gunjan, and V. K. Solanki, "On classification of BMD images using machine learning (ANN) algorithm," in ICDSMLA 2019, Singapore: Springer, 2020, pp. 1590–1599.
7. L. Liang, "Face recognition technology analysis based on deep learning algorithm," Journal of Physics: Conference Series, vol. 1544, p. 012158, 2020, doi: 10.1088/1742-6596/1544/1/012158.
8. X. Li, Y. Zhang, and H. Chen, "Deep learning-based facial emotion recognition: Recent advances and challenges," IEEE Access, vol. 11, pp. 45231–45248, 2023, doi: 10.1109/ACCESS.2023.3278456.
9. E. Rashid, M. D. Ansari, V. K. Gunjan, and M. Ahmed, "Improvement in extended object tracking with the vision-based algorithm," in Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough, Cham: Springer, 2020, pp. 237–245.
10. B. Ríos-Sánchez, D. Costa-da-Silva, N. Martín-Yuste, and C. Sánchez-Ávila, "Deep learning for facial recognition on single sample per person scenarios with varied capturing conditions," Applied Sciences, vol. 9, no. 24, p. 5474, 2019, doi: 10.3390/app9245474.
11. O. Sturman, L. von Ziegler, C. Schläppi et al., "Deep learning-based behavioral analysis reaches human accuracy and is capable of outperforming commercial solutions,"

Neuropsychopharmacology, vol. 45, pp. 1942–1952, 2020, doi: 10.1038/s41386-020-0776-y.

12. J. Wang, Q. Liu, and Y. Zhao, "Hybrid deep learning framework for facial expression recognition and behavioral analysis," Pattern Recognition Letters, vol. 186, pp. 45–53, 2025, doi: 10.1016/j.patrec.2024.11.012.
13. R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, "Deep image: Scaling up image recognition," arXiv preprint arXiv:1501.02876, 2015.
14. M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in Computer Vision – ECCV 2014, Springer, 2014, pp. 818–833.
15. M. D. Zeiler, M. A. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean et al., "On rectified linear units for speech processing," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 3517–3521.