Facial micro-expression classification through an optimized convolutional neural network using genetic algorithm

Krishna Santosh Naidana, Yaswanth Yarra, Lakshmi Prasanna Divvela
Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India

Article Info

Article history:

Received Dec 19, 2023
Revised Sep 2, 2024
Accepted Sep 28, 2024

Keywords:

Convolutional neural network
Hyperparameters
Micro-expression
Optimization
Random search
Real-coded genetic algorithm

ABSTRACT

Computer vision facilitates machines to interpret the visual world using various computer aided detection (CAD)-based techniques. It plays a crucial role in micro-expression auto classification. A micro-expression is a brief facial movement which reveals a genuine emotion that a person tries to conceal, it usually lasts for a short duration and is imperceptible with normal vision. To reveal people's genuine emotions, an automatic micro-expression screening using convolutional neural network (CNN) is in great need. Traditional methods for micro-expression recognition (MER) suffer from low classification accuracy due to inadequate CNN hyperparameters selection. The proposed approach addresses these challenges by using an optimized CNN with adequate learning rate, batch size, epochs, and dropout rate. Real-coded genetic algorithm (RCGA) has been employed for the hyperparameter optimization. In this experimentation, features are extracted from the onset and apex frames of microexpression video clips of CASME II dataset. The proposed model's performance is measured using various metrics, including accuracy, precision, and recall. The proposed approach's performance is then compared with an optimized CNN using random search algorithm. The empirical investigation of existing CNN-based methods has proven efficacy of our proposed model.

Corresponding Author:

Yaswanth Yarra
Department of Computer Science and Engineering
Velagapudi Ramakrishna Siddhartha Engineering College
Kanuru, Vijayawada, Andhra Pradesh, 520007, India
Email: yaswanthyarra7112@gmail.com

1. INTRODUCTION

Facial expressions are vital in human interactions for non-verbal communication of emotions and intentions. Micro-expressions [1] are involuntary, subtle, and fleeting facial expressions that reveal the genuine emotions of an individual. Common applications of recognizing micro-expressions include deception detection in various scenarios, such as investigations, understanding patient emotions to improve medical care, evaluating interviewee loyalty and enhancing customer service interactions.

Automated facial micro-expression recognition (MER) presents a significant deep learning (DL) challenge due to its complexity and multi-dimensional nature. Prior studies have addressed this challenge through the integration of spatial and temporal features for effective micro-expression classification. These methods consider both visual cues such as facial appearance and muscular movements, as well as motion-related factors like speed and duration [2]-[5] but they often overlook the intricacies the challenges associated with processing complete video sequences. Usually, the computer aided detection (CAD)-based MER process comprises several crucial steps, including data preprocessing, pertinent feature extraction, and the application of emotion classification techniques. Our proposed method employs a convolutional neural network (CNN) architecture where onset and apex frames of micro-expression video clips are stacked along input channels to enhance spatial detail and sensitivity to rapid facial movements. Hyperparameters are optimized using a genetic algorithm (GA), improving convergence towards optimal settings and reducing computational complexity. The model's performance is evaluated on the CASME II dataset.

– Related works

In the literature, several notable techniques have been proposed to improve MER. Most of these techniques followed a phase-wise (i.e., preprocessing, feature extraction, and optimization) MER improvement approach. Thus, we have discussed the phase-wise state of the approaches to identify gaps.

Initially, an uncertainty-aware magnification-robust network (UAMRN) is proposed by Wei *et al.* [6] to address low intensity and imbalanced datasets. This approach secured an accuracy of 80.57%, 79.41%, and 78.05% on CASME II, SAMM, and SMIC-HS datasets respectively. Later, Zhao *et al.* [7] introduced MEPLAN that incorporated 3D residual prototypical networks and a local-wise attention module on SMIC and SAMM datasets. Next, Gasah *et al.* [8] used micro-expressions to examine children’s facial expressions during the use of a mobile learning application and the results showed that over 70% of participants exhibited positive emotions. Thereafter, Lee *et al.* [9] introduced N-step pre-training and Décalcomanie data augmentation and achieved a weighted average recall (WAR) of 0.7927. Subsequently, Wang *et al.* [10] presented a lightweight OF-PCANet+ technique for MER with an accuracy of 62.80% for SMIC and 53.25% for CASME2 datasets. Afterwards, Tang *et al.* [11] employed a dual-stream model with histogram equalization (HE) for MER with an accuracy of 73.09% on the CASME II dataset.

Chen *et al.* [12] used a 3D spatiotemporal CNN with convolutional block attention module (CBAM) for MER, validated on CASME II and SMIC datasets. Later, Wang *et al.* [13] proposed the haphazard cuboids (HC) feature extraction method with a three-stream convolutional neural network (TSCNN). Afterwards Abdulsattar and Hussain developed a CNN which uses histogram of oriented gradients (HOG) and local binary pattern (LBP) for facial recognition and achieved an accuracy of 97.56% with HOG+CNN model [14]. Next, Li *et al.* [15] proposed deep local-holistic network (DLHN), combining local features from hierarchical convolutional recurrent neural network (HCRNN) and sparse features from robust principal-component-analysis-based recurrent neural network (RPRNN), achieved a 60.31% mean accuracy. Subsequently, Thuseethan *et al.* [16] presented a Deep 3DCNN-ANN framework with substantial accuracy improvements. Thereafter, Pan *et al.* [17] introduced 3D CNN embedding in the transformer model (C3Dbed) to address the low-intensity issues.

Jin *et al.* [18] used GA for feature selection for performance improvement. Later, Cabada *et al.* [19] optimized CNN hyperparameters based on highest fitness value for emotion recognition. Next, Fouad *et al.* [20] employed CNN with hyperparameters optimized using particle swarm optimization and achieved an error of 0.87% on MNIST dataset. Thereafter, Thavasimani and Srinath [21] proposed a custom GA to auto-tune hyperparameters of a DL sequential model for classifying benign and malicious traffic from the internet of things-23 dataset, achieving 98.9% accuracy. Subsequently, Liu *et al.* [22] utilized GA for optimization, obtained an accuracy of 85.9%. Afterwards, Raji *et al.* [23] proposed simple deterministic GA (SDSGA) for hyperparameter tuning and achieved optimal results. Thereafter, WangPing *et al.* [24] proposed a feature selection operator based on genetic programming which can reduce complexity of high-dimensional features. Existing approaches in automated facial MER often struggle with effectively tuning hyperparameters for CNN architectures and grappling with the complexity inherent in processing entire video sequences. Despite the pivotal role hyperparameters play in CNN performance, many studies adopt default or arbitrary values, leading to suboptimal model performance and hindering convergence. Moreover, the complexity of handling entire video sequences poses additional challenges for MER techniques. Facial micro-expressions are inherently dynamic and fleeting, requiring the analysis of temporal patterns and subtle changes over time. However, processing entire video sequences introduces computational and memory constraints, especially when using DL models.

– Research contribution

Our study addresses gaps in prior research by examining the landscape of automated facial MER. In the proposed work the onset and apex frames are stacked along the input channels of a CNN to intricate spatial details, enhancing its ability to tackle tasks like video analysis and action recognition with a holistic perspective thus reducing the complexity of the model without compromising the required features. The CNN architecture is meticulously designed with three convolutional layers that facilitates the network's ability to hierarchically extract and analyze intricate spatial features. It also enhances CNNs sensitivity to the subtle and rapid facial movements inherent to microexpressions, ultimately resulting in improved classification performance for this challenging domain. In the process of hyperparameter optimization using a GA, the initial population comprises potential chromosomes that play a pivotal role in expediting convergence towards the most optimal hyperparameters. By initializing with promising candidates, the GA effectively kickstarts the search process, significantly reducing the number of generations required to identify the hyperparameters that yield the highest performance, thereby improving the efficiency of the optimization process. The rest of the paper is organized as follows: section 2 describes the phase-wise proposed CAD approach, section 3 performs a deep investigation of the obtained results, and section 4 concludes the article with future scope.

2. METHOD

In this approach, a CNN architecture has been proposed with a GA-based optimized hyperparameters. We found that the utilization of CNNs for image classification tasks, coupled with hyperparameter tuning using GA, correlates with enhanced accuracy. CNNs excel in automatically learning hierarchical representations of features from raw image data, effectively capturing local patterns and features through convolutional and pooling layers. GA automate the tedious process of hyperparameter optimization by iteratively evolving candidate solutions, efficiently searching the complex hyperparameter space of CNNs to maximize performance. This combined approach not only leverages the feature extraction capabilities of CNNs but also optimizes hyperparameters such as learning rates, batch sizes, and filter configurations, leading to improved accuracy and generalization on image classification tasks. The proposed method in this study tended to have an inordinately higher proportion of accuracy improvements as compared to traditional approaches. By integrating CNNs with GA-driven hyperparameter tuning, researchers can streamline model development, alleviate manual effort, and achieve superior performance in image classification tasks.

2.1. Preprocessing

The dataset used in our study is the widely recognized CASME II [25] dataset, which serves as a benchmark for spontaneous micro-expression analysis, providing a collection of 255 spontaneous micro-expression sequences from 26 participants all recorded using a high-speed camera operating at 200 frames per second. CASME II covers a range of seven micro-expressions, including happiness, surprise, disgust, repression, fear, sadness, and others. Each video sequence in the CASME II dataset is precisely annotated, providing frame-level information about the onset, apex, and offset frames of micro-expressions. The subsequent sub-phases: preprocessing, CNN model and its optimization for MER. The preprocessing pipeline involves extracting onset and apex frames from micro-expression sequences [26] from each micro-expression sequence. This is followed by image grayscale conversion and the contrast enhancement using HE approach. Then both the apex and onset frames are stacked along the CNN input channels for the expressions' pattern learning. Subsequently, data augmentation is employed to address the data size limitation. Various transformations are used to augment minority class stacked onset and apex frames. These transformations include rotation (up to 10 degrees), width/height shift (up to 10%), shear (up to 10% of width), zoom (up to 10%), horizontal flip (50% probability), and brightness adjustment (50%-150%). In Table 1 original and augmented sample counts have been given for various expressions. A stratified 80:20 split is employed by allocating 80% for training and 20% for evaluation. It ensures a consistent distribution of classes between training and testing to counter dataset imbalance.

Table 1. Number of original and augmented frames of CASME II dataset

Type	Happiness	Sadness	Repression	Disgust	Surprise	Fear	Others	Total
Original	32	7	27	63	25	2	99	255
Augmented	32	63	27	-	25	60	-	207
Total	64	70	54	63	50	62	99	462

2.2. Proposed convolutional neural network model

The proposed CNN architecture has three pairs of convolutional and MaxPooling layers as depicted in Figure 1. Convolutional layers are particularly effective at detecting patterns and features within image, while max pooling layers help to reduce the spatial dimensionality of the output from the convolutional layers, resulting in translation invariance. The first convolutional layer has 32 filters with a kernel size of 3×3 and utilized the ReLU activation function. The input shape of the images was specified as (64, 64, 2) where the first two dimensions represent the height and width of the input images (64 pixels each), and the last dimension represents the number of channels. The MaxPooling layer is applied with a pool size of 2×2. The next convolutional layer has 64, 3×3 filters with ReLU activation function. Another MaxPooling layer with a pool size of 2×2 followed this layer. Subsequently, a third convolutional layer with 128, 3×3 filters is added with a MaxPooling layer of pool size of 2×2. To prevent overfitting, dropout regularization is applied after the flatten layer. The model is then connected to a fully connected (Dense) layer with 128 units and a ReLU activation function. Finally, a SoftMax activation function is used at the output layer to predict one of the specified classes for an image. The Adam optimizer is utilized for training the model with the sparse categorical cross-entropy loss function.

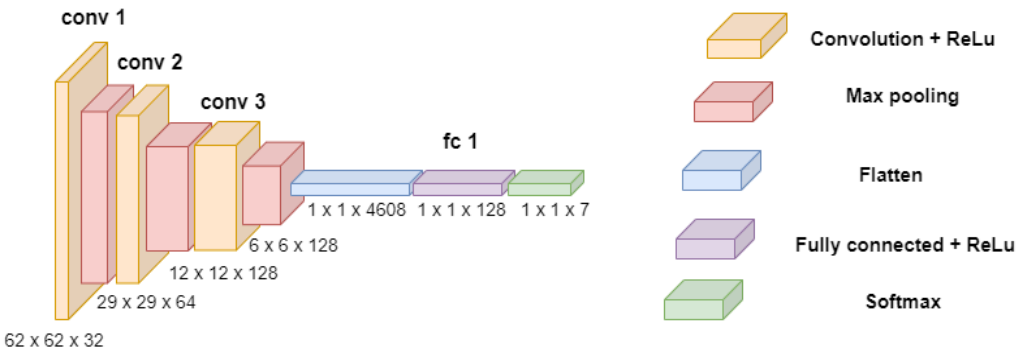

Figure 1. Architecture of proposed CNN

2.3. Optimization of proposed convolutional neural network model

Hyperparameter optimization is a widely employed approach to enhance CNN prediction accuracy. Specifically, in this optimization, four key hyperparameters: the learning rate (lr), batch size (bs), dropout rate (dp), and number of epochs (ep) have been considered. This choice has been made based on the results obtained from the existing optimization-based CNN experiments with significant outcomes. In this experimentation, we have empirically investigated the performances of GA and Random Search optimization methods in MER.

2.3.1. Real-coded genetic algorithm

A real-coded genetic algorithm (RCGA) is an optimization technique, based on principles of natural selection and genetics. It uses real-valued representations for individuals in the population. It allows more flexibility in handling of continuous variables during evolution. It achieves an optimal solution by evolving an individuals'(chromosomes') population through selection, mating, crossover, and mutation processes. In this study, the initial chromosome population is selectively generated. It is done through testing various hyperparameter combinations, creating a diverse and promising starting point for GA optimization. This approach ensures a faster GA convergence with relatively suitable population. In Table 2, an example of a chromosome encoded with values for four hyperparameters is given.

Table 2. Example of chromosome from the initial population

$F_{learning\_rate}$	F_{batch_size}	$F_{dropout\_rate}$	F_{epochs}
0.001	32	0.2	24

Each step of the Algorithm 1 is explained clearly as follows. In step two, the fitness function is used to evaluate the quality of each candidate solution (i.e., chromosome) in the GA. The GA tries to maximize this fitness value over generations. In step four, tournament selection is employed as the selection method for the GA optimization process. The fittest individuals among them are selected for mating. In step six of the algorithm, blend crossover is used as the crossover method in the GA optimization process with a crossover probability Pc=0.7. In blend crossover, a blending parameter (α) is used to control the degree of mixing between the parent individuals. In step six of the algorithm offspring1 and offspring2 are the two offsprings generated from the crossover, and parent1 and parent2 are the two parent individuals selected for reproduction. Mutation is a genetic operator that introduces diversity into the population and helps to explore new regions of the search space. Two mutation operators are employed in this study: Gaussian mutation and uniform mutation. Gaussian mutation is employed for lr and dp hyperparameters that take continuous values. In step seven $N(0, mutation\_rate * range)$ is a random value drawn from a normal distribution with mean 0 and standard deviation $mutation\_rate * range$. The range is the allowable range of values for the hyperparameter. $U(-mutation\_rate * range, mutation\_rate * range)$ is a random value drawn from a uniform distribution between $-mutation\_rate * range$ and $mutation\_rate * range$. The mutation probability Pm is set to 0.3. The flowchart for the proposed algorithm is depicted in Figure 2.

Algorithm 1. Genetic algorithm for hyperparameter tuning in CNN

Require: Initial population of chromosomes
Ensure: Optimized hyperparameters for CNN
1: Generate random population of chromosomes
2: Evaluate fitness of each chromosome based on

$$Test\ accuracy = \frac{Number\ of\ correctly\ classified\ instances}{Total\ number\ of\ instances} * 100\%$$

$$Fitness\ function = Test\ accuracy$$

3: **while** not converged do
4: Select individuals for tournament
5: Determine fittest individuals
6: Apply blend crossover using

$$offspring1 = (1-\alpha) * parent1 + \alpha * parent2$$

$$offspring2 = \alpha * \mathrm{parent1} + (1-\alpha) * parent2\ ,\alpha = 0.5$$

7: Apply mutation operators based on

$newValue = oldValue + N(0, mutationRate * range)$ (Gaussian Mutation)

$newValue = oldValue + U(-mutationRate * range, mutationRate * range)$ (Uniform Mutation)

8: Evaluate fitness of offspring using
9: **end while**
10: Return best-performing chromosome

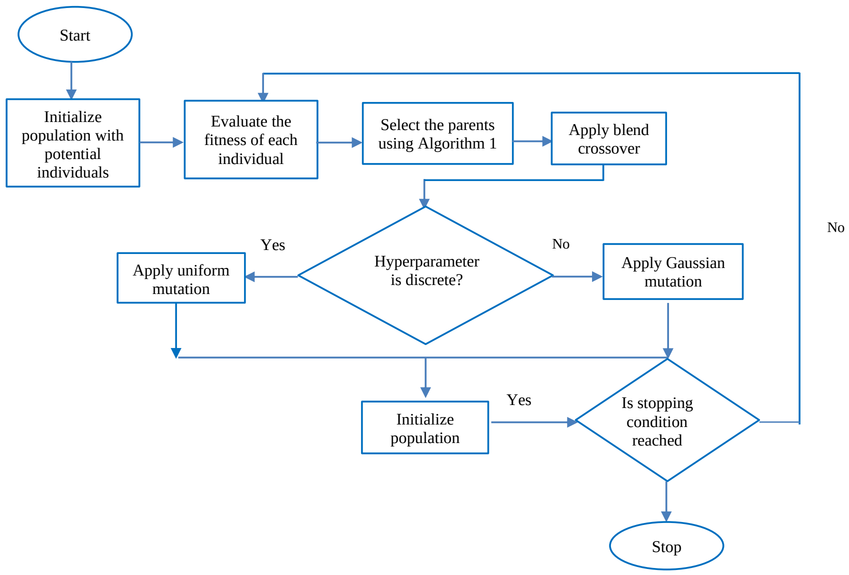

Figure 2. Flowchart of genetic algorithm

2.3.2. Random search

Random search is a popular approach for hyperparameter tuning, which randomly samples hyperparameters from a defined search space. In this study, a random search has been used to determine the set of optimal CNN hyperparameters. The search space is defined for various hyperparameters as follows: 0.001 to 0.0001 for lr, 0.2 to 0.4 for dp, 32 to 128 for bs, and 40 to 60 for ep. During each iteration, hyperparameters are randomly sampled from the search space and are employed to train and cross-validate the model. The hyperparameter set with the optimal performance is considered as the final set.

3. RESULTS AND DISCUSSION

This study investigated a comparative study between the performance of the two optimization techniques employed in our research and compared our work with state-of-the-art methods in the field of micro-expression classification. Through this analysis, we aimed to gain insights into the efficacy of our optimization approach and its competitiveness in the current landscape of micro-expression classification methodologies. Additionally, we sought to identify potential areas for improvement and future research directions within this domain.

3.1. Optimal hyperparameters

The RCGA and random search have been employed, starting with an initial population of five potential chromosomes. The individual with highest accuracy after running for five generations, has been considered for the optimal choice. The results were depicted in Table 3. The optimal hyperparameters determined by the proposed RCGA and random search are utilized in the construction of two separate CNN models. Further, a CNN model is built without any hyperparameter tuning for the baseline evaluation.

Table 3. Optimal hyperparameters determined by RGCA and random search

	Learning_rate	Batch_size	Dropout_rate	Number of epochs
CNN-GA	0.0006	17	0.29	52
CNN-random search	0.001	64	0.3	60

3.2. Classification performance

The performance of the three CNN models is evaluated on CASME II dataset using various metrics including accuracy, precision, recall, f1-score, and UAR. By employing multiple evaluation metrics, we aimed to gain a comprehensive understanding of the strengths and weaknesses of each model in accurately classifying micro-expressions within the given dataset. Table 4 presents the comparison of performance of three models.

Table 4. Comparison of classification performance of CNN, CNN optimized with RCGA, and CNN optimized with random search based on various metrics

Metrics	Standard CNN	RCGA optimized CNN	Random search optimized CNN
Accuracy	0.7634	0.8924	0.8602
Recall	0.7634	0.8924	0.8602
F1-Score	0.7359	0.8902	0.8585
Precision	0.7930	0.8905	0.8691
UAR	0.7516	0.9093	0.8646

The evaluation has revealed significant improvements in performance of the proposed RCGA optimized CNN framework. More precisely, it exhibited 12.9% higher classification accuracy when compared to the non-optimized CNN and 3.22% higher accuracy when compared to random search optimized CNN. The proposed framework achieved the highest UAR, with a value of 90.93%, surpassing the performance of other evaluated models. Similar trends were observed across other metrics as well. These results highlight the superior performance of the proposed framework. The variation of the prediction accuracies for all the three models used in the experiments are shown in Figure 3.

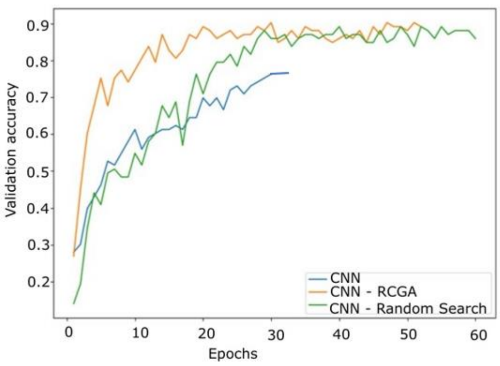

Figure 3. Variation of prediction accuracies for all three models

The graph in Figure 3 depicts the performance of three different models: standard CNN, CNN optimized with RCGA, and CNN optimized with random search. The standard CNN model initially achieves an accuracy of 30%, which gradually stabilizes around 76% after approximately 35 epochs. In contrast, the CNN optimized with random search exhibits more fluctuations, with accuracies ranging between 65% and 75% during the first 15-25 epochs, before reaching a peak accuracy of 86% around 60 epochs. Notably, the CNN optimized with RCGA demonstrates fewer fluctuations compared to the random search optimization, starting at around 28% accuracy and increases to approximately 89% over the course of approximately 52 epochs. This comparison highlights the effectiveness of optimization techniques such as RCGA in enhancing the performance and stability of CNN models, as evidenced by the superior accuracy achieved within a similar epoch range compared to both the standard CNN and random search-optimized CNN models. The corresponding confusion matrices obtained by optimizing hyperparameters of CNN through random search and RCGA are shown in Figures 4 and 5.

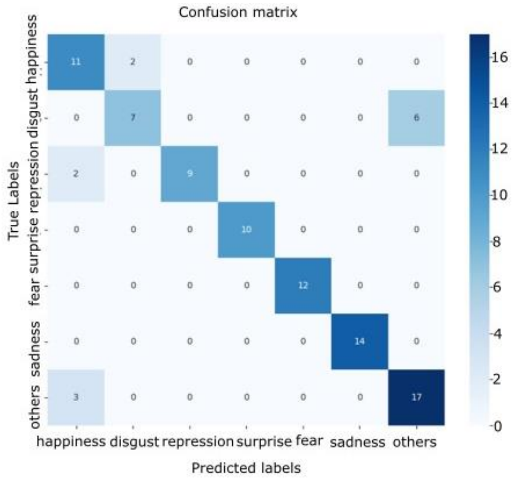

Figure 4. Confusion matrix using random search-CNN

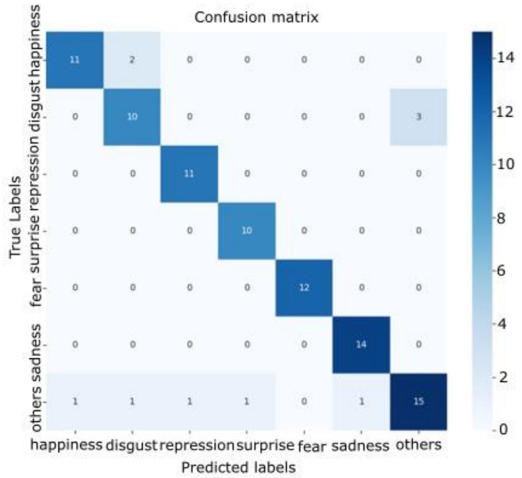

Figure 5. Confusion matrix for RCGA-CNN

Upon comparison, the second confusion matrix demonstrates a more balanced distribution of misclassifications across classes compared to the first matrix. It features a total of 4 false positives and 5 false negatives, whereas the first matrix has a total of 16 false positives and 17 false negatives. Notably, in the second matrix, the “Happy” class shows only 2 false negatives and 1 false positive, an improvement from the 3 false negatives and 2 false positives in the first matrix. Similarly, the “Fear” class exhibits better performance in the second matrix with only 1 false positive and 1 false negative, in contrast to the first matrix. Overall, the second matrix reflects superior accuracy and balance in class predictions. This empirical evidence indicates the effectiveness of the GA-based optimization approach in enhancing the model's accuracy across different micro-expression classes. Our proposed approach was solely tested on the CASME II dataset, without consideration of other datasets. Furthermore, our approach exclusively utilizes micro-expression video clips elicited under controlled laboratory conditions, precluding the direct handling of real-time data.

3.3. Comparative evaluation

Subsequently, a comprehensive comparison has been conducted between the proposed RCGA optimized CNN model and existing MER baselines that use CNN architectures and deep feature to evaluate overall performance of the proposed model, and the comparison is presented in Table 5. The proposed method may benefit from RCGA optimization without negatively affecting the overall performance of the CNN model. The proposed model secures 3.24% and 5.02% improvements in the recognition accuracy and F1-Score over the Deep3DCANN framework [16] and secures approximately 7% and 14% improvement in the recognition accuracy and F1-Score over the SE-DenseNet-T+EVM framework [2] on CASME II dataset. This shows considerable improvements in the performance over the other state of art methods for micro-expression classification.

Table 5. Comparison chart of proposed approach with state-of-art method

Reference and year	Method	Dataset	Acc	Rec	F1	Prec	UAR
[12] and 2020	CNN+CBAM	CASME II	0.6992	-	-	-	-
[22] and 2021	STSTNet+GA for feature selection	CASME II	-	-	-	-	0.882
[10] and 2022	Optical flow and PCANet+	CASME2	0.5325	0.5241	0.5493	-	-
[27] and 2022	Deep 3D CNN	CASME II	0.565	-	-	-	-
[28] and 2022	SE-DenseNet-T+EVM	CASME II	0.8275	-	0.7559	-	-
[6] and 2022	Décal-FSB (+rotation, resize, and flip)	SMIC	-	-	-	-	0.7947
[16] and 2023	Deep3DCANN	CASME II	0.86	-	0.84	-	-
[29] and 2023	DS-3DCNN with domain adaptation	SMIC	-	-	-	-	0.8061
		CASME II	0.8924	0.8924	0.8902	0.8905	0.9093

Note: Acc is accuracy, Rec is recall, F1 is F1 score, and Prec is precision.

4. CONCLUSION

In this study, we proposed a novel optimized RCGA-CNN model for spontaneous facial micro-expression classification, achieving a remarkable accuracy of 89.24%. Our research shows that the proposed optimized RCGA-CNN is more resilient than traditional CNN models without hyperparameter optimization. Recent observations indicate that dataset imbalance significantly affects micro-expression classification accuracy. Our results offer definitive proof that addressing dataset imbalance through data augmentation techniques leads to improved classification accuracy, rather than being caused by increased quantities of training data alone. The findings achieved not only advance the state-of-the-art in MER but also hold practical implications for real-world applications requiring accurate and reliable emotion detection. Future research may look into integrating generative adversarial networks (GANs) and exploring additional advanced techniques like attention mechanisms and transfer learning could further enhance the robustness and versatility of MER systems, paving the way for broader applications in security, psychology, human-computer interaction, and emotion recognition technology. Additionally, considering other datasets beyond CASME II and addressing the handling of real-time data are important areas for future scope and development.

ACKNOWLEDGMENT

The CASME II dataset used in this study was given by CASME Series database Managers. We would like to express our deep and sincere gratitude to Jingting Li for providing corresponding information and the dataset. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors do not have conflicts of interest with anyone.

REFERENCES

[1] L. A. Gottschalk, A. H. Auerbach, E. A. Haggard, and K. S. Isaacs, "Micro-momentary facial expressions as indicators of ego mechanisms in psychotherapy," *Methods of Research in Psychotherapy*, pp. 154-165, 1996, doi: 10.1007/978-1-4684-6045-2_14.
[2] S. L. Happy and A. Routray, "Fuzzy Histogram of Optical Flow Orientations for Micro-Expression Recognition," *IEEE Transactions on Affective Computing*, vol. 10, no. 3, pp. 394-406, 2019, doi: 10.1109/TAFFC.2017.2723386.
[3] D. H. Kim, W. J. Baddar, and Y. M. Ro, "Micro-expression recognition with expression-state constrained spatio-temporal feature representations," *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 382-386, doi: 10.1145/2964284.2967247.
[4] M. Peng, C. Wang, T. Chen, G. Liu, and X. Fu, "Dual temporal scale convolutional neural network for micro-expression recognition," *Frontiers in psychology*, vol. 8, pp. 1745, 2017, doi:10.3389/fpsyg.2017.01745.
[5] S. Thuseethan, S. Rajasegarar, and J. Yearwood, "Detecting micro-expression intensity changes from videos based on hybrid deep cnn, Pacific-Asia Conference on Knowledge Discovery and Data Mining," *Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part III 23*, *Springer International Publishing*, vol. 11441, pp. 387-399, 2019, doi: 10.1007/978-3-030-16142-2_30.
[6] M. Wei, Y. Zong, X. Jiang, C. Lu, and J. Liu, "Micro-expression recognition using uncertainty-aware magnification-robust networks," *Entropy*, vol. 24, no. 9, p. 1271, 2022, doi: 10.3390/e24091271.
[7] S. Zhao *et al*., "ME-PLAN: A deep prototypical learning with local attention network for dynamic micro-expression recognition," *Neural Networks*, vol. 153, pp. 427-443, 2022, doi: 10.1016/j.neunet.2022.06.024.
[8] M. Gasah, A. Baharum, N. H. M. Zain, S. Halamy, R. Hanapi, and N. A. M. Noor, "Evaluation of positive emotion in children mobile learning application," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 2, pp. 818-826, 2020, doi: 10.11591/eei.v9i2.2073.
[9] C. Lee, J. Hong, and H. Jung, "N-Step Pre-Training and Décalcomanie Data Augmentation for Micro-Expression Recognition," Sensors, vol. 22, no. 17, p. 6671, Sep. 2022, doi: 10.3390/s22176671.
[10] S. Wang, S. Guan, H. Lin, J. Huang, F. Long, and J. Yao, "Micro-Expression Recognition Based on Optical Flow and PCANet+," *Sensors*, vol. 22, no. 11, pp. 4296, 2022, doi: 10.3390/s22114296.
[11] J. Tang, L. Li, M. Tang, and J. Xie, "A novel micro-expression recognition algorithm using dual-stream combining optical flow and dynamic image convolutional neural networks," *Signal, Image and Video Processing*, vol. 17, pp. 769–776, 2023, doi: 10.1007/s11760-022-02286-0.
[12] B. Chen, Z. Zhang, N. Liu, Y. Tan, X. Liu, and T. Chen, "Spatiotemporal Convolutional Neural Network with Convolutional Block Attention Module for Micro-Expression Recognition," *Information*, vol. 11, no. 8, p. 380, 2020, doi: 10.3390/info11080380.
[13] G. Wang, S. Huang, and Z. Dong, "Haphazard Cuboids Feature Extraction for Micro-Expression Recognition," *IEEE Access*, vol. 10, pp. 110149-110162, 2022, doi: 10.1109/ACCESS.2022.3214808.
[14] R. S. Saeed and B. K. O. C. Alwawi, "A binary classification model of COVID-19 based on convolution neural network," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 3, pp. 1413-1417, 2023, doi: 10.11591/eei.v12i3.4832.
[15] J. Li, T. Wang, and S.-J. Wang, "Facial Micro-Expression Recognition Based on Deep Local-Holistic Network," *Applied Sciences*, vol. 12, no. 9, pp. 4643, 2022, doi: 10.3390/app12094643.
[16] S. Thuseethan, S. Rajasegarar, and J. Yearwood, "Deep3DCANN: A Deep 3DCNN-ANN framework for spontaneous micro-expression recognition," *Information Sciences*, vol. 630, pp. 341-355, 2023, doi: 10.1016/j.ins.2022.11.113.
[17] H. Pan, L. Xie, and Z. Wang,"C3DBed: Facial micro-expression recognition with three-dimensional convolutional neural network embedding in transformer model," *Engineering Applications of Artificial Intelligence*, vol. 123, pp. 106258, 2023, doi: 10.1016/j.engappai.2023.106258.
[18] Q. S. Jin, H. C. Xu, K. H. Liu, S. T. Liong, Y. S. Gan, and S. Su, "GA-APEXNET: genetic algorithm in apex frame network for

micro-expression recognition system," *Journal of Physics: Conference Series*, vol. 1544, no. 1, pp. 012149, 2020, doi: 10.1088/1742-6596/1544/1/012149.
[19] R. Z. Cabada, H. R. Rangel, M. L. B. Estrada, and H. M. C. Lopez, "Hyperparameter optimization in CNN for learning-centered emotion recognition for intelligent tutoring systems," *Soft Computing*, vol. 24, pp. 7593–7602, 2020, doi: 10.1007/s00500-019-04387-4.
[20] Z. Fouad, M. Alfonse, M. Roushdy, and A. B. M. Salem, "Hyper-parameter optimization of convolutional neural network based on particle swarm optimization algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 6, pp. 3377-3384, 2021, doi: 10.11591/eei.v10i6.3257.
[21] K. Thavasimani and N. K. Srinath, "Hyperparameter optimization using custom genetic algorithm for classification of benign and malicious traffic on internet of things-23 dataset," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 4, 2022, doi: 10.11591/ijece.v12i4.pp4031-4041.
[22] K.-H. Liu, Q.-S. Jin, H.-C. Xu, Y.-S. Gan, and S.-T. Liong, "Micro-expression recognition using advanced genetic algorithm," *Signal Processing: Image Communication*, vol. 93, pp. 1-10, 2021, doi: 10.1016/j.image.2021.116153.
[23] I. D. Raji, H. Bello-Salau, I. J. Umoh, A. J. Onumanyi, M. A. Adegboye, and A. T. Salawudeen, "Simple deterministic selection-based genetic algorithm for hyperparameter tuning of machine learning models," *Applied Sciences*, vol. 12, no. 3, pp. 1-19, 2022, doi: 10.3390/app12031186.
[24] Z. WangPing, J. Min, Y. JunFeng, L. KunHong, and W. QingQiang, "The design of evolutionary feature selection operator for the micro-expression recognition," *Memetic Computing*, vol 14, no. 1, pp. 61-76, 2022, doi: 10.1007/s12293-021-00350-9.
[25] W. J. Yan, X. Li, SJ Wang, G. Zhao, and Y. J Liu, "CASME II: An Improved Spontaneous Micro-Expression Database and the Baseline Evaluation," *PloS one*, vol. 9, no. 1, pp. 1-8, 2014, doi: 10.1371/journal.pone.0086041.
[26] S. T. Liong, J. See, K. Wong, and R. C. W. Phan, "Less is more: Micro-expression recognition from video using apex frame," *Signal Processing: Image Communication*, vol. 62, pp. 82-92, 2018, doi: 10.1016/j.image.2017.11.006.
[27] K. K. Talluri, M. A. Fiedler, and A. Al-Hamadi, "Deep 3D Convolutional Neural Network for Facial Micro-Expression Analysis from Video Images," *Applied Sciences*, vol. 12, no. 21, pp. 1-16, 2022, doi: 10.3390/app122111078.
[28] L. Cai, H. Li, W. Dong, and H. Fang, "Micro-expression recognition using 3D DenseNet fused Squeeze-and-Excitation Networks," *Applied Soft Computing*, vol. 119, p. 108594, 2022, doi: 10.1016/j.asoc.2022.108594.
[29] Z. Li, Y. Zhang, H. Xing, and K.-L. Chan, "Facial Micro-Expression Recognition Using Double-Stream 3D Convolutional Neural Network with Domain Adaptation," *Sensors*, vol. 23, no. 7, pp. 1-19, Mar. 2023, doi: 10.3390/s23073577.

BIOGRAPHIES OF AUTHORS

Dr. Krishna Santosh Naidana received his B.Tech. degree in computer science and engineering from the Jawaharlal Nehru Technological University, Hyderabad, and M.Tech. degree in software engineering from the GITAM University. He did his Ph.D. in VIT-AP University, Amaravati, Andhra Pradesh, India. His research area is medical image processing. He is currently working as Assistant Professor in the Department of Computer Scinece and Engineering at Velagapudi Ramakrishna Siddhartha Engineering College. He can be contacted at email: krishna.santosh10@gmail.com.

Yaswanth Yarra is currently pursuing B.Tech. degree in Computer Science and Engineering at Velagapudi Ramakrishna Siddhartha Engineering College. He contributed to the project "Dynamic Ranking of Personalized Recommendations using Cosine Similarity and User Ratings", which was presented at 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India. He can be contacted at email: yaswanthyarra7112@gmail.com.

Lakshmi Prasanna Divvela is currently pursuing B.Tech. degree in Computer Science and Engineering at Velagapudi Ramakrishna Siddhartha Engineering College. She contributed to the project "Raise of complaints on day-to-day issues by the public using Global Positioning System for finding the exact location of the problem", which was presented at 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), India. She can be contacted at email: lakshmiprasanna.divvela@gmail.com.