Advanced Gesture Recognition in Gaming: Implementing EfficientNetV2-B1 for "Rock, Paper, Scissors"

Chander Prabha
Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India
prabhanice@gmail.com

Retinderdeep Singh
Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India
retinderdeepsingh@gmail.com

Meena Malik
Chandigarh University, Mohali, Punjab, India
meenamlk@gmail.com

Manas Ranjan Pradhan
School of Computing, Skyline University College, Sharjah, United Arab Emirates
manaspradhan@yahoo.com

Biswaranjan Acharya
Department of Computer Engineering – AI & BDA, Marwadi University, Rajkot, Gujarat, India
biswaacharya@ieee.org (corresponding author)

Received: 29 January 2025 | Revised: 27 February 2025 | Accepted: 6 March 2025

 DOI: https://doi.org/10.48084/etasr.10373

ABSTRACT

The study introduces a gesture recognition system for the classic "Rock, Paper, Scissors" game, based on a modified EfficientNetV2-B1 architecture. The dataset comprises 2,700 images, evenly divided among the three classes: "Rock", "Paper", and "Scissors". Leveraging the efficiency and accuracy of the EfficientNetV2-B1 model in image recognition tasks, the system was trained to classify these gestures effectively, and after fine-tuning, it achieved an accuracy of 98.89% and an Area Under the Curve (AUC) of ~1.0, indicating near-perfect classification across all classes. This performance highlights the potential of EfficientNetV2-B1 for real-time gesture recognition, with applications in interactive gaming and other gesture-based user interfaces. The proposed system also offers a foundation for further research and development in gesture recognition technologies.

Keywords-rock-paper-scissors; deep learning; EfficientNetV2-B1; gesture recognition

I. INTRODUCTION

Communication between humans and computers increasingly relies on gesture recognition technology, which plays a vital role in gaming systems, Augmented Reality (AR), Virtual Reality (VR), robotics, and assistive technologies. At the same time, the COVID-19 pandemic accelerated the demand for such contactless interfaces, positioning gesture-based systems as intuitive, hygienic alternatives to traditional input methods [1]. Early hand gesture recognition systems relied on vision-based techniques, including edge detection and motion tracking, rather than skin color segmentation. However, these classical methods were highly sensitive to changes in lighting, hand orientation, and background conditions, leading to inconsistent results. The introduction of deep learning and machine learning significantly advanced gesture recognition in Human-Computer Interaction (HCI), improving both accuracy and efficiency [2]. In particular, Convolutional Neural Networks (CNNs) have proven highly effective, as they automatically extract robust image features and learn hierarchical representations from raw visual input [3, 4]. Pioneering CNN architectures like LeNet and AlexNet significantly boosted detection accuracy [5], with AlexNet's

success catalyzing widespread adoption of deep learning across domains, including gesture recognition [6]. For dynamic gestures, traditional machine learning approaches—like Hidden Markov Models (HMMs) and Support Vector Machines (SVMs) have also been applied to reduce variability in gesture execution [7].

More recently, the EfficientNet family of CNNs has garnered attention for achieving high accuracy with relatively low computational cost. In particular, the EfficientNetV2-B1 variant stands out for real-time applications due to its optimized architecture, which balances speed and accuracy [8, 9]. It improves upon its predecessor with compound scaling, progressive image resizing, and integrated MBConv layers, offering better performance with fewer parameters and faster training [10-12]. These characteristics make EfficientNetV2-B1 especially well-suited for real-time gesture recognition tasks. Its lightweight and efficient design enables deployment in resource-constrained environments, while maintaining low latency and high classification accuracy [13].

This study presents a gesture recognition system based on EfficientNetV2-B1 for classifying the three gestures in the "Rock, Paper, Scissors" game, which is a simple yet widely recognized gesture-based game that serves as an effective testbed for developing and evaluating gesture detection systems. The model is trained on a dataset of 2,700 images, equally divided among the three gesture classes. The dataset was curated to include a wide range of hand positions, lighting conditions, and backgrounds to ensure robustness and applicability in real-world scenarios.

II. PROPOSED METHODOLOGY

The overall architecture of the proposed gesture classification system is illustrated in Figure 1.

A. Dataset

The dataset employed in this study comprises 2,700 images, equally distributed among the three gesture classes: "Rock", "Paper", and "Scissors" [14]. With 900 images per class, the dataset ensures a balanced class representation. From the dataset, 2,160 (80%) images were used for training, 270 (10%) images were used for validation, and 270 (10%) images were used for testing.

The dataset incorporates a wide range of variations in skin tone, hand size, hand orientation, image resolution, lighting conditions, and backgrounds. Each image was carefully annotated with its corresponding gesture label to support supervised learning. Figure 2 presents representative examples from the three gesture classes.

The gestures used in the "Rock-Paper-Scissors" game have practical implications beyond gameplay, particularly in the design of gesture-based interfaces:

- Rock (Clenched Fist): Commonly associated with selection, confirmation, or "click" actions in contactless interfaces and gaming systems.
- Paper (Open Palm): A widely recognized gesture used for stopping actions, pausing operations, or requesting attention in AR/VR environments and interactive platforms.
- Scissors (Two Extended Fingers): Often used in directional commands, gesture-based navigation, and even sign language recognition, making it valuable for assistive technologies and touchless user interfaces.

Proper recognition of these gestures is critical in a range of applications, including gaming, AR, VR, sign language translation, and HCI systems [15].

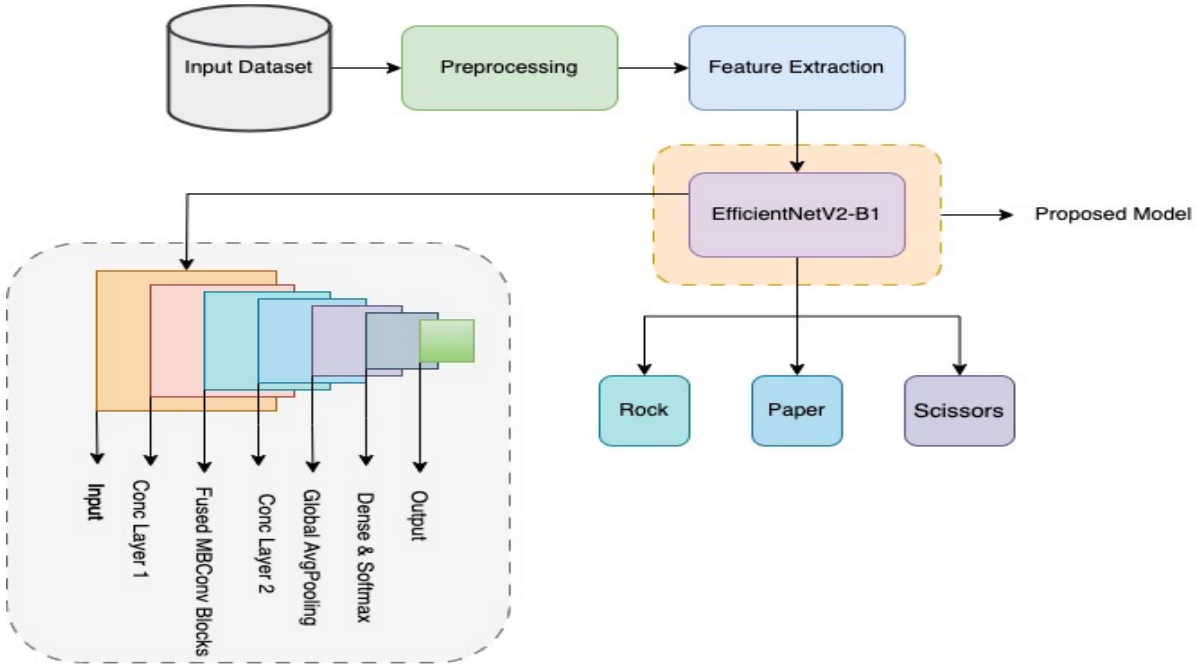

Fig. 1. Proposed methodology.

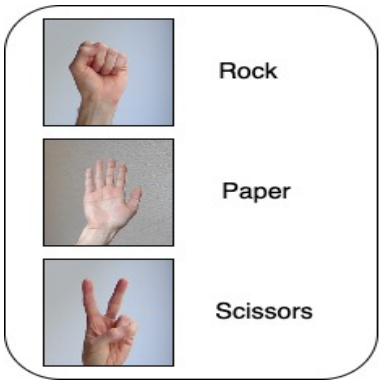

Fig. 2. Dataset classes.

B. Preprocessing

Preprocessing plays a critical role in preparing the dataset for effective training of the gesture recognition model. The following steps were applied to standardize and optimize the input data:

- Image Resizing: All images were resized to 224 × 224 pixels, matching the input dimension requirements of the EfficientNetV2-B1 model. This resizing ensures consistent input size across all images and allows for efficient batch processing.
- Normalization: To scale pixel intensity values to a consistent range, all pixel values were normalized to the range (0, 1) by dividing each value by 255. This transformation accelerates convergence during training and improves model stability by keeping the input distribution within a standard range.
- Label Encoding: The categorical gesture labels—Rock, Paper, and Scissors—were numerically encoded as 0, 1, and 2, respectively. This encoding format aligns with the requirements of the model's output layer for multi-class classification [16].

Given the limited size of the dataset (2,700 images), data augmentation was applied to artificially increase variability and enhance generalization capabilities. Augmentation not only simulates real-world diversity in gestures but also reduces the risk of overfitting. The augmentation techniques applied are summarized in Table I and include transformations such as random rotations, horizontal flipping, zooming, and brightness adjustments. These techniques simulate real-world conditions like different hand orientations, backgrounds, and lighting conditions, further strengthening the model's real-time applicability.

TABLE I. AUGMENTATION TECHNIQUES

Augmentation Technique	Purpose
Random Rotation (± 15°)	Accounts for different hand orientations.
Horizontal & Vertical Flip	Ensures robustness in mirrored gestures.
Zooming (0.8x–1.2x)	Simulates variations in hand distance from the camera.
Color Jittering (Brightness, Contrast, Saturation)	Helps the model adapt to different lighting conditions.
Gaussian Noise Injection	Reduces overfitting and enhances feature learning.

C. Model

Key features of the EfficientNetV2-B1 design are:

- Compound Scaling [17]: EfficientNetV2 employs a compound scaling method that uniformly scales the model's depth, width, and input resolution. This coordinated scaling delivers better performance compared to the independent scaling of these factors. The B1 variant was selected specifically for its balance of precision and efficiency, making it ideal for lightweight applications like gesture recognition on edge devices.
- Fused-MBConv Layers [18]: The model leverages Fused-MBConv layers, which integrate squeeze-and-excitation blocks with depthwise separable convolutions. These fused operations reduce computational overhead while maintaining high model accuracy, resulting in faster inference speeds without sacrificing performance.
- Progressive Training: EfficientNetV2 adopts a progressive image scaling strategy during training. Initially, the model is trained in lower-resolution images and then fine-tuned with higher-resolution ones. This process improves generalization while significantly reducing training time [19].

The architecture consists of 8 stages, each comprising multiple MBConv blocks optimized for efficient inference, presented in Table II. Furthermore, the key hyperparameters used for training the model are detailed in Table III.

TABLE II. MODEL ARCHITECTURE

Layer Type	Output Size	Layers & Details
Input Layer	224 × 224 × 3	RGB Input Image
Stem Conv Layer	112 × 112 × 24	3×3 Conv, ReLU6
MBConv1	112 × 112 × 24	1×1, 3×3 Depthwise, 1×1 SE
MBConv2	56 × 56 × 48	1×1, 3×3 Depthwise, 1×1 SE
MBConv3	28 × 28 × 64	1×1, 5×5 Depthwise, 1×1 SE
MBConv4	14 × 14 × 128	1×1, 7×7 Depthwise, 1×1 SE
MBConv5	7 × 7 × 160	1×1, 7×7 Depthwise, 1×1 SE
Global Average Pooling	1 × 1 × 160	Feature Compression
Fully Connected Layer	3 (Softmax)	Classification Output

TABLE III. HYPERPARAMETER OF THE MODEL

Hyperparameter	Value	Justification
Batch Size	32	Balances memory usage and training stability.
Optimizer	Adam	Adaptive learning rate for better convergence.
Learning Rate	0.001 (reduced on plateau)	Prevents overshooting and ensures smooth training.
Weight Decay (L2 Regularization)	0.0001	Reduces overfitting.
Dropout Rate	0.3	Prevents overfitting by randomly deactivating neurons.
Epochs	10	Achieved convergence within 10 epochs.
Loss Function	Categorical Cross-Entropy	Suitable for multi-class classification.

D. Feature Extraction

Feature extraction in deep learning involves isolating the most informative aspects of input data to make accurate predictions. In the EfficientNetV2-B1 model, feature extraction is performed through a series of convolutional and fused layers that learn hierarchical representations of hand gestures from image inputs.

At the initial stages of the network, convolutional layers extract low-level features such as edges, textures, and simple geometric shapes. These early feature maps act as the foundation for understanding the structure of the image and are particularly effective in distinguishing basic visual patterns relevant to hand positioning and orientation. As the data propagates deeper into the network, the Fused-MBConv layers capture more complex and abstract features, such as the contours and contextual shapes of specific hand gestures. Before the last classification layer, a Global Average Pooling (GAP) layer is applied to reduce each feature map to a single value. This not only compacts the learned spatial features but also reduces the risk of overfitting and lowers the number of trainable parameters. Following this, the final dense layer equipped with a softmax activation function outputs the probability for each gesture class. The class with the highest probability is selected as the predicted label, ensuring an interpretable and confident final decision [20].

To visualize how the model processes and identifies gestures, feature maps from both the first and last layers of the EfficientNetV2-B1 model were generated:

- First Layer Feature Maps (Figure 3): These maps emphasize simple low-level features such as borders, edges, and textures, capturing the primary motion cues of the hand. These form the foundational components for gesture recognition.
- Final Layer Feature Maps (Figure 4): These advanced representations combine information learned across all prior layers, enabling the network to discriminate between gesture categories effectively. The composite of high-level features in these maps shows how the network interprets semantic meaning and subtle gesture differences.

These visualizations demonstrate how the model successfully leverages deep feature extraction to robustly classify gestures in real-time.

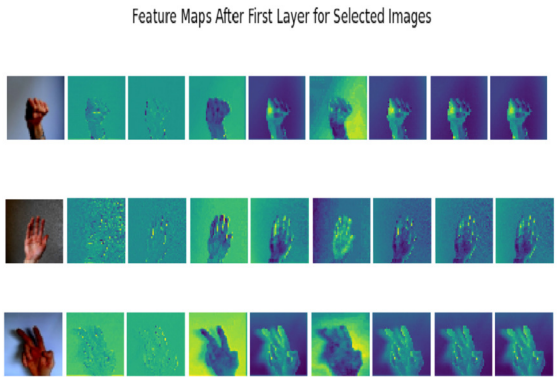

Fig. 3. Features after the first layer.

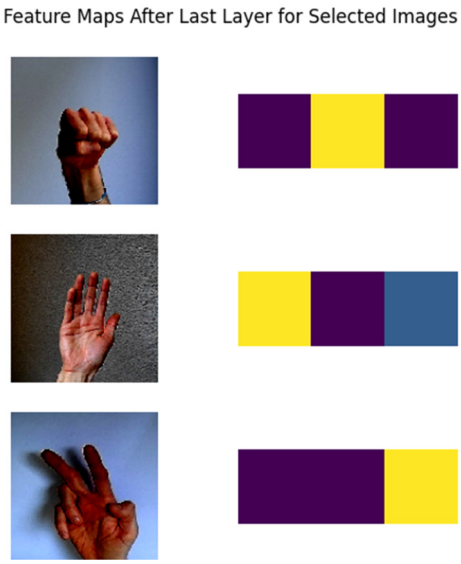

Fig. 4. Features after the last layer.

III. RESULTS

The EfficientNetV2-B1 model, optimized using the "Rock, Paper, Scissors" dataset, was comprehensively evaluated to assess its effectiveness in gesture recognition. Evaluation metrics, including accuracy, precision, recall, F1-score, confusion matrix, and the Area Under the Curve (AUC), were employed to measure performance across the training, validation, and testing phases.

A. Training and Validation

Performance metrics, including loss and accuracy for both the training and validation sets, were monitored over 10 epochs (Figures 5 and 6).

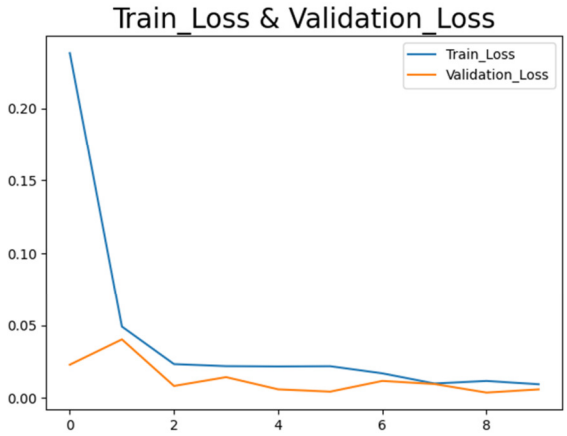

Fig. 5. Training and validation loss graph.

The training loss consistently decreased from an initial value of 0.2372 to 0.0095 by the final epoch. Correspondingly, the training accuracy improved steadily from 90.64% in the first epoch to 99.69% in the last, indicating effective and progressive learning throughout the training phase. Similarly,

the validation loss showed a clear downward trend, starting at 0.0229 and decreasing to 0.0038 by the end of training. The validation accuracy started at 99.59% and ended at approximately the same percentage by the final epoch. These results underscore the model's strong generalization capability and its ability to perform exceptionally well on unseen data.

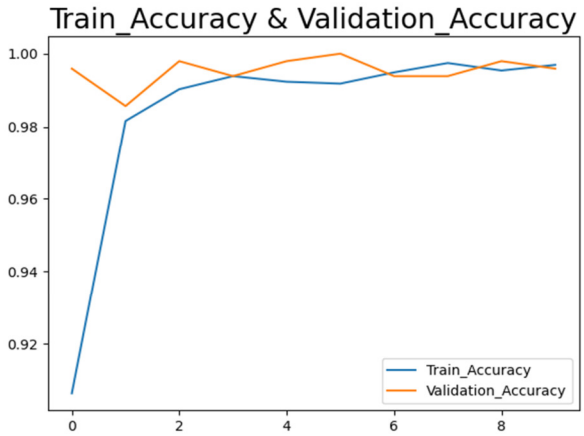

Fig. 6. Training and validation accuracy graph.

B. *Testing*

The model's performance on the test set is shown in Table IV. Across the majority of the metrics and for all three gestures, the proposed model achieves a score of 99%

TABLE IV. PERFORMANCE MATRIX

Classes	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
Rock	99	100	99	99
Paper	99	99	99	99
Scissors	99	98	98	99

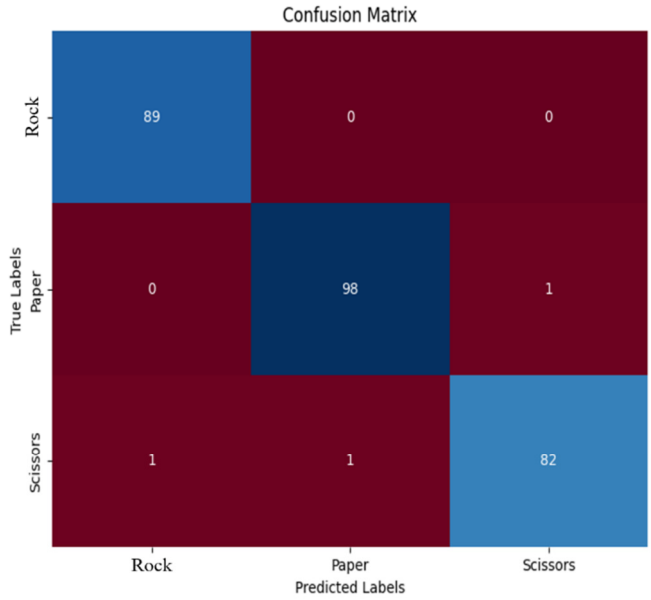

Fig. 7. Confusion matrix.

The confusion matrix presents the counts of true positives, false positives, true negatives, and false negatives, highlighting potential areas of model weakness. The test set confusion matrix is shown in Figure 7. The model demonstrates strong performance, with high accuracy across all classes and a very low misclassification rate. Specifically, there were three misclassifications: one "Scissors" predicted as "Rock", one "Scissors" as "Paper", and one "Paper" as "Scissors".

Misclassification Rate (MR) is calculated as (1):

$$MR = \frac{\text{Number of Misclassified Samples}}{\text{Total Samples}} \cdot 100 \quad (1)$$

With only 1.1% of predictions incorrect, the model exhibits high accuracy and reliability in gesture recognition.

Additionally, the AUC was computed for each gesture category to evaluate the classification performance, shown in Figure 8. The AUC values were 0.9993 for "Rock", 0.9987 for "Paper", and 0.9975 for "Scissors", indicating excellent class separability and model accuracy.

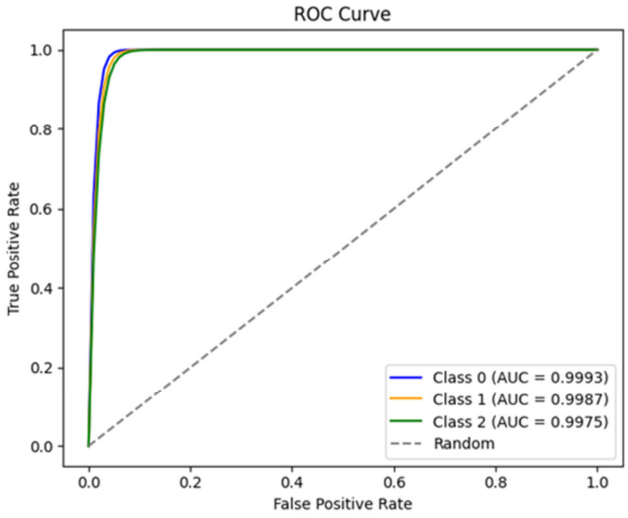

Fig. 8. ROC curve.

C. *Comparative Analysis*

Authors in [21] employed transfer learning using AlexNet and achieved a validation accuracy of 99.54% on a dataset comprising 2,188 images. In contrast, the proposed approach achieved a validation accuracy of 99.79% on a dataset of approximately 2,700 images. Both studies highlight the effectiveness of deep learning models in gesture recognition through transfer learning, even with relatively small datasets. While AlexNet and EfficientNetV2-B1 each have their strengths, the latter offers a more modern architecture with improved scalability and adaptability, making it better suited for a wide range of gesture recognition tasks [22-24].

Additionally, the proposed approach uses compound scaling, which means that depth, width, and resolution are scaled together, unlike traditional CNNs that scale them independently. This balanced approach prevents overfitting and ensures high accuracy while reducing computational overhead.

Compared to AlexNet [21], VGG [25], and ResNet [26], this results in fewer parameters while also achieving higher accuracy, as shown in Table V.

TABLE V. STATE OF THE ART SUMMARY

Model	Parameters (M)	Accuracy (%)	Computational Cost
AlexNet [21]	61 M	99.54%	High
VGG-16 [25]	138 M	98.5%	Very High
ResNet-50 [26]	25.6 M	99.1%	Moderate
EfficientNetV2-B1	8.1 M	99.79%	Low (Best Trade-off)

IV. CONCLUSION

This study developed a gesture recognition system for the "Rock, Paper, Scissors" game using the EfficientNetV2-B1 variant. The system achieved a validation accuracy of 99% on a dataset of approximately 2,700 images across three classes ("Rock", "Paper", "Scissors") and an Area Under the Curve (AUC) value of ~1.0. These results demonstrate the model's strong and precise motion-based classification capabilities, which combine computational efficiency with high accuracy.

The findings suggest that EfficientNetV2-B1 is well-suited for applications in virtual reality, gaming, and other human-computer interaction interfaces where reliable and fast gesture detection is critical. Highlighting the potential of powerful deep learning models for real-time gesture recognition, this work lays the foundation for future research aimed at expanding the range of recognized gestures and integrating such systems into more interactive technologies.

REFERENCES

[1] S. S. Rautaray and A. Agrawal, "Interaction with virtual game through hand gesture recognition," in *2011 International Conference on Multimedia, Signal Processing and Communication Technologies*, Aligarh, India, Dec. 2011, pp. 244–247, https://doi.org/10.1109/MSPCT.2011.6150485.

[2] H. G. Doan and N. T. Nguyen, "Fusion Machine Learning Strategies for Multi-modal Sensor-based Hand Gesture Recognition," *Engineering, Technology & Applied Science Research*, vol. 12, no. 3, pp. 8628–8633, Jun. 2022, https://doi.org/10.48084/etasr.4913.

[3] A. Mohanty, S. S. Rambhatla, and R. R. Sahay, "Deep Gesture: Static Hand Gesture Recognition Using CNN," in *Proceedings of International Conference on Computer Vision and Image Processing*, vol. 460, Singapore, 2017, pp. 449–461.

[4] A. Dey, A. Anand, S. Samanta, B. K. Sah, and S. Biswas, "Attention-Based AdaptSepCX Network for Effective Student Action Recognition in Online Learning," *Procedia Computer Science*, vol. 233, pp. 164–174, 2024, https://doi.org/10.1016/j.procs.2024.03.206.

[5] B. Karsh, R. H. Laskar, and R. K. Karsh, "mIV3Net: modified inception V3 network for hand gesture recognition," *Multimedia Tools and Applications*, vol. 83, no. 4, pp. 10587–10613, Jan. 2024, https://doi.org/10.1007/s11042-023-15865-1.

[6] C. Griffin, L. Feng, and R. Wu, "Spatial Dynamics of Higher Order Rock-Paper-Scissors and Generalisations." *arXiv*, 2023, https://doi.org/10.48550/ARXIV.2312.16722.

[7] A. S. M. Miah, Md. A. M. Hasan, Y. Tomioka, and J. Shin, "Hand Gesture Recognition for Multi-Culture Sign Language Using Graph and General Deep Learning Network," *IEEE Open Journal of the Computer Society*, vol. 5, pp. 144–155, 2024, https://doi.org/10.1109/OJCS.2024.3370971.

[8] B. Kim and S. Seo, "EfficientNetV2-based dynamic gesture recognition using transformed scalogram from triaxial acceleration signal," *Journal of Computational Design and Engineering*, vol. 10, no. 4, pp. 1694–1706, Jul. 2023, https://doi.org/10.1093/jcde/qwad068.

[9] J. Qi, L. Ma, Z. Cui, and Y. Yu, "Computer vision-based hand gesture recognition for human-robot interaction: a review," *Complex & Intelligent Systems*, vol. 10, no. 1, pp. 1581–1606, Feb. 2024, https://doi.org/10.1007/s40747-023-01173-6.

[10] A. Amrutesh, K. P. Asha Rani, A. Amruthamsh, S. Gowrishankar, and C. G. Gowtham Bhat, "Quantitative Study on Variation of Glaucoma Eye Images Using Various EfficientNetV2 Models," in *AI-Centric Modeling and Analytics*, 1st ed., Boca Raton: CRC Press, 2023, pp. 173–197.

[11] Z. Mohammadi, A. Akhavanpour, R. Rastgoo, and M. Sabokrou, "Diverse hand gesture recognition dataset," *Multimedia Tools and Applications*, vol. 83, no. 17, pp. 50245–50267, Nov. 2023, https://doi.org/10.1007/s11042-023-17268-8.

[12] M. N. Ichsan, N. Armita, A. E. Minarno, F. D. S. Sumadi, and H. Hariyady, "Increased Accuracy on Image Classification of Game Rock Paper Scissors using CNN," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 4, pp. 606–611, Aug. 2022, https://doi.org/10.29207/resti.v6i4.4222.

[13] E. N. H. Kırğıl and Ç. B. Erdaş, "Enhancing Skin Disease Diagnosis Through Deep Learning: A Comprehensive Study on Dermoscopic Image Preprocessing and Classification," *International Journal of Imaging Systems and Technology*, vol. 34, no. 4, Jul. 2024, Art. no. e23148, https://doi.org/10.1002/ima.23148.

[14] *Rock-Paper-Scissors-Dataset*. (2024), A. Donciu-Julin. [Online]. Available: https://github.com/alexdjulin/RockPaperScissorsCNN.

[15] A. Dey, S. Biswas, and D.-N. Le, "Recognition of Wh-Question Sign Gestures in Video Streams using an Attention Driven C3D-BiLSTM Network," *Procedia Computer Science*, vol. 235, pp. 2920–2931, 2024, https://doi.org/10.1016/j.procs.2024.04.276.

[16] A. B. S. Salamh and H. I. Akyüz, "A Novel Feature Extraction Descriptor for Face Recognition," *Engineering, Technology & Applied Science Research*, vol. 12, no. 1, pp. 8033–8038, Feb. 2022, https://doi.org/10.48084/etasr.4624.

[17] A. Kumar and A. Mantri, "Gesture-Based Model of Mixed Reality Human-Computer Interface," in *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*, Moradabad, India, Dec. 2020, pp. 226–230, https://doi.org/10.1109/SMART50582.2020.9337130.

[18] M. Bhalekar and M. Bedekar, "D-CNN: A New model for Generating Image Captions with Text Extraction Using Deep Learning for Visually Challenged Individuals," *Engineering, Technology & Applied Science Research*, vol. 12, no. 2, pp. 8366–8373, Apr. 2022, https://doi.org/10.48084/etasr.4772.

[19] E. K. Goyal and A. Singh, "Indian Sign Language Recognition System for Differently-able People," *Journal on Today's Ideas - Tomorrow's Technologies*, vol. 2, no. 2, pp. 145–151, Dec. 2014, https://doi.org/10.15415/jotitt.2014.22011.

[20] D. Ye *et al.*, "Towards Playing Full MOBA Games with Deep Reinforcement Learning." *arXiv*, 2020, https://doi.org/10.48550/ARXIV.2011.12692.

[21] F. Ahmed, W. A. Khan, M. Iqbal, A. R. Ahmad Abazeed, H. Alrababah, and M. F. Khan, "Rock-Paper-Scissors Image Classification Using Transfer Learning," in *2023 International Conference on Business Analytics for Technology and Security (ICBATS)*, Dubai, United Arab Emirates, Mar. 2023, pp. 1–6, https://doi.org/10.1109/ICBATS57792.2023.10111433.

[22] L. Jiao and J. Zhao, "A Survey on the New Generation of Deep Learning in Image Processing," *IEEE Access*, vol. 7, pp. 172231–172263, 2019, https://doi.org/10.1109/ACCESS.2019.2956508.

[23] C. Tian, Y. Xu, L. Fei, and K. Yan, "Deep Learning for Image Denoising: A Survey," in *Genetic and Evolutionary Computing*, vol. 834, J.-S. Pan, J. C.-W. Lin, B. Sui, and S.-P. Tseng, Eds. Singapore: Springer Singapore, 2019, pp. 563–572.

[24] O. Koller, S. Zargaran, H. Ney, and R. Bowden, "Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition," in *Proceedings of the British Machine Vision Conference 2016*, York, UK, 2016, https://doi.org/10.5244/C.30.136.

[25] S. Sharma and S. Singh, "Vision-based hand gesture recognition using deep learning for the interpretation of sign language," *Expert Systems with Applications*, vol. 182, Nov. 2021, Art. no. 115657, https://doi.org/10.1016/j.eswa.2021.115657.

[26] A. Alnuaim, M. Zakariah, W. A. Hatamleh, H. Tarazi, V. Tripathi, and E. T. Amoatey, "Human-Computer Interaction with Hand Gesture Recognition Using ResNet and MobileNet," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–16, Mar. 2022, https://doi.org/10.1155/2022/8777355.