Vol. 34, No. 3, 2025

Machine GRAPHICS & VISION

International Journal

Published by
The Institute of Information Technology
Warsaw University of Life Sciences – SGGW
Nowoursynowska 159, 02-776 Warsaw, Poland
in cooperation with
The Association for Image Processing, Poland – TPO

Optimization of VR Human-computer Game Interaction Based on Improved PIFPAF Algorithm and Binocular Vision

Hong Zhu[1] and Bo Chen[2,*]
[1] *School of Experimental Art, Hubei Institute of Fine Arts, Wuhan, China*
[2] *The School of Arts, Hubei University of Education, Wuhan, China*
[*] *Corresponding author: Bo Chen (chenbo1565@163.com)*

Submitted: 12 Feb 2025 **Accepted:** 7 Apr 2025 **Published:** 26 Jul 2025

Abstract To make virtual reality human-computer games more accurate and provide users with an immersive gaming experience, the study combines the method of improved part intensity field and part association field (PIFPAF) with binocular vision to optimize the interaction of VR human-computer games. The experimental results indicated that the PIFPAF algorithms performed relatively well with number of errors and target keypoint correlation of 0.22 and 0.97, respectively. In terms of processing speed, the algorithm performed faster in both 640×480 and 320×240 resolutions, with 13 fps and 19 fps, respectively. Among the five predefined gestures, the "pointing" gesture was recognized correctly the largest number of times in 30 test sessions, with 29 successful identifications. In contrast, the "clenched fist" gesture had the fewest correct recognitions, totaling 26. The success of the suggested approach is confirmed by the experimental findings, which show that the optimized human-computer interaction system has high accuracy and processing speed. This study offers a fresh approach to the advancement of human-computer interaction technology and encourages technological integration innovation in the realm of virtual reality human-computer gaming.

Keywords: virtual reality; PIFPAF algorithm; binocular stereo vision; keypoint detection algorithm; dimensioning algorithm.

1. Introduction

As virtual reality (VR) technology advances quickly, it has progressively made its way into a variety of industries, including gaming, education, healthcare, design, and more, as a new interactive experience. The naturalness and intuitiveness of human-computer interaction (HCI) are also the key factors to enhance user experience [5, 7]. Traditional VR interaction methods, such as joysticks and keyboards, although satisfy users' needs to a certain extent, have certain limitations in simulating real-world interaction. It limits the user's immersion and interaction experience in the VR environment [17]. Although some deep learning-based stereo matching methods have made progress, they still face challenges such as high computational complexity, large hardware requirements, and poor adaptability to dynamic scenes in real-time applications. Optimizing the network structure, introducing lightweight models, utilizing parallel computing, and adaptive feature extraction techniques can improve their efficiency and real-time performance.

The part intensity field and part association field (PIFPAF) algorithm, originally proposed by Kreiss et al. [9], aims to solve the keypoint association problem in multi-person pose estimation. This algorithm can more accurately detect human body structure and

associate keypoints by predicting local keypoints and spatial relationships between keypoints, especially suitable for complex interactive scenes. The development of PIFPAF is inspired by earlier work such as OpenPose, proposed by Cao et al. [3], which pioneered bottom-up keypoint detection in multi-person pose estimation.

In current HCI technology, traditional methods have many key problems in VR interaction scenarios, including chaotic keypoint matching during multi-person interaction, which leads to a decrease in tracking accuracy. In the case of occlusion, the loss of keypoint information is severe, which affects the accuracy of recognition. The high computational complexity affects real-time interaction performance. PIFPAF enhances local feature extraction by deep neural networks and optimizes the skeleton keypoint matching strategy, enabling it to infer the pose of the occluded part well even in occluded environments while maintaining computational efficiency. These features make it an ideal choice for optimizing VR HCI systems, enhancing immersive experiences and real-time performance, and promoting the development of intelligent interaction systems. Binocular vision can provide depth information to more accurately localize the user's body parts in complex environments [1, 4]. Therefore, to improve the naturalness and accuracy of VR human-computer game interaction (HCGI), this study investigates VR HCGI based on the improved PIFPAF algorithm with binocular vision.

The innovativeness of the research lies in the improvement of the existing PIFPAF algorithm in order to increase the accuracy and real-time performance of human pose estimation. It also combines binocular vision technology with the improved PIFPAF algorithm, thus proposing a new depth-aware interaction. The contribution of this research is to improve the accuracy of keypoint detection and the robustness of limb association through this algorithm. Its branch accurately locates keypoints using Gaussian heat maps and establishes human skeleton connections using vector fields, which is more resistant to occlusion compared to traditional methods. In addition, PIFPAF optimized the feature extraction network, adopted an efficient ResNet backbone network, and used adaptive scale inference to improve the ability to detect different human postures while reducing computational redundancy. These enhancements significantly improve real-time performance and enable efficient and accurate pose estimation in complex interactive scenarios, resulting in a smoother and more intuitive interactive experience.

The research will be carried out in four sections. The Section 2 is a review of the current research status of binocular vision and VR HCI. The Section 3 is the optimization study of human keypoint detection and HCI system. The Section 4 contains the experimental analysis of the research algorithms and system performance. In the Section 5 the results of experiments with the methodology proposed in this paper are discussed. The last Section 6 is a summary of the research.

2. Related works

With the advancement of computer graphics and HCI technology, VR technology has gradually matured, providing users with unprecedented immersive experiences. Optimization of HCI experience is also a hot research topic at present. Lyu aimed to explore the current state of HCI in the metaverse, and research results showed that key technologies such as 5G, blockchain, and HCI supported the development of the metaverse. In the future, humanized somatosensory connections in HCI could become a trend [14]. Ramadoss proposed an optimized non-invasive human-machine interaction model to improve the accuracy of human motion recognition in HCI. The research results showed that this method had significant effects on human motion and target recognition, reducing noise by 7.2% and improving accuracy to 97.2% [15]. Li proposed an interaction design model that combined artificial intelligence (AI) and voice information to enhance the HCI experience in VR environments. Research showed that this model promoted the application and development of VR technology in multiple fields such as gaming, fitness, and education by optimizing the HCI design [11].

Keypoint detection algorithms and stereo binocular vision can effectively detect and track human posture and motion. Read proposed a research method that comprehensively examined the use of binocular vision and stereoscopic vision in order to explore the advantages and disadvantages of binocular vision and its mechanisms. The research results indicated that although binocular vision reduced the overall field of view, it enhanced obstacle avoidance and contrast sensitivity [16]. Bonnen et al. proposed a research method that combined eye and body tracking to explore the role of binocular vision in complex terrain walking. The research results indicated that binocular vision was crucial for locating a foothold, and its absence could systematically affect gaze strategies, increasing perceptual uncertainty and making the gaze more inclined towards a nearby foothold [2]. Lin et al. proposed a recognition method based on improved ResNet and skeleton keypoints to improve the accuracy of single image human action recognition, and constructed a multi task network. The research results showed that this method could accurately recognize human movements under different human motion, background light, and occlusion conditions. Compared with the original network and main recognition algorithms, it had an advantage in accuracy and balances network parameters, solving the problems of large network and slow operation [12]. Zhang proposed a method that combined efficient network structure, training strategy, and post-processing techniques to address the challenge of human keypoint detection in a single image. The research results indicated that this method effectively improved the detection accuracy and outperforms the latest technology on the benchmark of keypoint detection [20]. To improve the accuracy and practicality of the fall detection system, Inturi proposed a new visual based fall detection scheme. The research results indicated

that the system could effectively detect five types of falls and six types of daily activities, and performed well on the UP-FALL dataset [6].

VR and HCI technology have made significant progress in recent years. They generate highly realistic 3D virtual environments through computers, allowing users to interact with the virtual world in an immersive way. They are widely used in various fields such as gaming, education, healthcare, and design. In the VR field, major manufacturers have introduced several innovative devices. In 2023, Sony released the PlayStation VR2, which featured internal and external tracking, eye tracking, a high-definition display, and a controller with adaptive triggering and haptic feedback to enhance the gaming experience. In 2024, Apple released the Apple Vision Pro, a fully enclosed mixed reality headset that emphasizes video perspective functionality. Although it lacks the external controller of traditional VR headsets, it is described as a spatial computer. In terms of HCI, with the advancement of technologies such as computer graphics and AI, HCI is gradually shifting from traditional keyboard- and mouse-based interaction modes to more natural and intelligent interaction methods. For example, interaction methods based on gesture recognition, speech recognition, eye tracking, and other technologies are gradually emerging. The integration of eye tracking technology enables the system to optimize rendering based on the user's point of view, improving performance and immersion. In addition, the development of hand tracking and gesture recognition technology allows users to interact with virtual environments in a more natural way, reducing reliance on traditional controllers. Together, these advances are driving the evolution of VR and HCI technologies, providing users with a more intuitive and immersive experience.

To summarize, many scholars have researched on HCI technology. Moreover, there are more studies on the acquisition of information about human motion and posture using binocular vision technology or human keypoint detection algorithm, and certain results have been achieved. However, most of the scholars only use a single algorithm model and do not improve the model's deficiencies. Most researchers focus on action recognition, trajectory prediction, and interactive feedback when researching VR interactive technology. However, these studies have certain limitations when faced with complex actions and multi-person interaction scenarios. First, current mainstream methods often rely on deep learning-based motion capture or pose estimation algorithms, such as convolutional neural networks, recurrent neural networks, and their variants, when dealing with complex actions. Although these methods can achieve good recognition results in simple single-person interaction scenarios, there are bottlenecks in the recognition and prediction of complex actions, mainly due to the difficulty of the model to accurately capture high-speed and nonlinear motion trajectories, especially when multiple joints are involved in coordinated motion. Existing methods have weak temporal modeling capabilities and are difficult to accurately predict subsequent actions. Furthermore, in multi-person interaction scenarios, traditional methods typically use data fusion based on visual or inertial sensors to analyze user interaction behavior. However, these methods

are susceptible to data noise and environmental disturbances when faced with multiple occlusions, dynamic background changes, or synchronized interactions. This can result in interaction delays, increased error rates in action recognition, and ultimately reducing the immersion and real-time feedback effects of the game. On the other hand, traditional optimization algorithms typically rely on rule-based or reinforcement learning frameworks, such as Markov decision processes and reinforcement learning, when dealing with path planning and action generation problems in VR HCI. However, these methods have a high computational overhead in high-dimensional state spaces, making it difficult to respond to the complex action needs of users in real time. In addition, traditional reinforcement learning models are unable to efficiently model the dynamic interaction relationships between different users in multiuser collaborative interactions, making it difficult for the system to adapt to changing interaction patterns. It can be concluded that in response to the complexity and real-time requirements of VR HCGI scenarios, the existing research has the limitation of balancing computational efficiency, interaction accuracy, and real-time response capability, which has become a key challenge to further enhance the VR interaction experience.

For the above reasons, in this research the PIFPAF algorithm is improved by combining it with the enhanced binocular vision technology to locate the user in 3D, so as to optimize the VR HCGI system.

3. Optimization of VR interpersonal game interaction

3.1. Keypoint dimensional enhancement algorithm based on improved binocular vision technique

VR technology can use computer-generated 3D images and sounds to simulate the real sensory experience of humans [10, 22]. However, in VR human-computer games, users' hand movements and body movements have a high degree of complexity and diversity [8]. The keypoint detection algorithm can improve the accuracy of human motion detection in VR games by accurately locating human joints. These algorithms can capture player poses in real time, reducing errors caused by occlusion or complex movements, making interactions smoother. Combined with deep learning models, keypoint detection can optimize limb tracking, enabling the system to more accurately understand the player's intent. It can also improve the accuracy of physical feedback, improve action matching, avoid delays, enhance immersion, and provide strong technical support for motion interaction and prediction in VR games. Based on this, the study adopts the human body keypoint detection algorithm for keypoint positioning of human body images. When designing keypoint detection algorithms to accurately capture various complex user actions in VR environments, the following key factors need to be considered. Firstly, the algorithm should have high robustness to cope with challenges such as occlusion, lighting changes, and complex backgrounds. Secondly, it is necessary to ensure real-time

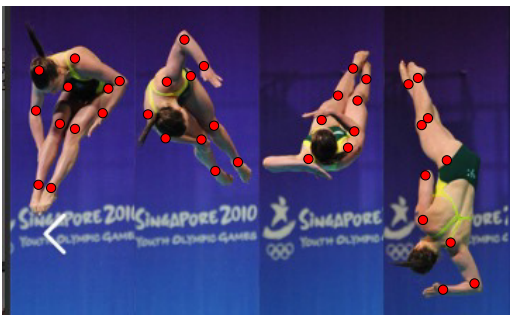


Fig. 1. Image of human keypoint positioning

performance to meet the low latency requirements of VR interaction. In addition, the algorithm should be able to adapt to users of different body types and action patterns, and have good generalization ability. Finally, it is necessary to optimize computational efficiency to achieve efficient operation with limited hardware resources.

Figure 1 illustrates the precise keypoint positioning. In this Figure, the keypoint detection algorithm for human keypoint positioning is mainly distributed in the joints of face and limb joints and torso. Facial keypoints are mainly used in VR applications for facial expression recognition, user authentication, and immersive interactive experiences. By accurately capturing facial movements, real-time facial expression mapping of virtual characters can be achieved, enhancing the authenticity of social interactions. In addition, facial keypoints can optimize voice synchronization and improve character performance. In security, they can be used for identity recognition, ensuring personalized settings and data security. For immersive control, the combination of eye tracking can provide a more natural way of visual interaction, improving the responsiveness and user experience of VR systems [13, 19]. In dynamic VR environments, facial keypoint detection faces challenges such as high real-time requirements, insufficient robustness in complex scenes, and limited computing resources. To address these obstacles, the following strategies can be adopted: firstly, optimize the algorithm structure and introduce lightweight CNN to reduce computational complexity and improve processing speed; Secondly, by combining multimodal information such as depth information and optical flow information, the robustness of facial keypoint detection is enhanced; Finally, parallel computing technology is utilized to further enhance the real-time performance of the algorithm. In addition, an adaptive feature extraction method attention mechanism is adopted to dynamically focus on key facial regions, improving detection accuracy. In order to effectively improve the accuracy and real-time performance of facial keypoint detection in dynamic VR environments with limited computing resources.

The data interaction function of the VR system is shown in Figure 2. In this Figure,

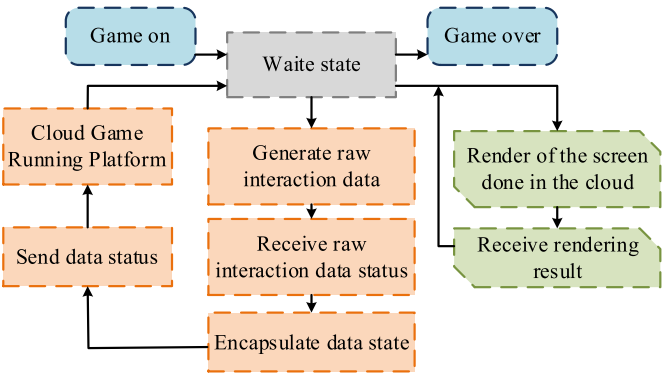


Fig. 2. Interactive activity diagram of the interaction controller

the interactive controller has five states, starting from the initial waiting state. After the game starts, it enters the state of receiving raw interactive data and recording player operations. Then it encapsulates the data and sends the data status, processes and uploads the interactive data to the cloud. Next, it enters the state of receiving rendering results and receives rendered images from the cloud. After the game ends, the controller returns to the waiting state and prepares for the next interaction [5]. In VR games, interactive controllers must manage different states, including idle, active, interactive, and feedback, to ensure a smooth user experience. Real-time state switching determines response speed, such as the accuracy of gesture recognition, physical collision detection, and environmental feedback. Accurate state management can reduce latency, improve immersion, optimize the allocation of computing resources, and prevent lag. The combination of intelligent predictive algorithms and adaptive control strategies can enhance real-time interaction capabilities, making player interaction in virtual environments more natural and fluid. Moreover, its combination of intelligent prediction algorithms and adaptive control strategies can enhance real-time interaction capabilities, making players' operations in virtual environments more natural and smooth.

The real-time state switching of interactive controllers is extremely important for the accuracy of gesture recognition and physical collision detection. It reduces the delay between user actions and system feedback, improving the real-time response. In addition, dynamic computing resource allocation optimizes processing efficiency, prioritizing critical interaction tasks such as gesture recognition or collision detection. Real time state switching also enhances the naturalness of interaction, allowing the system to smoothly transition between different interaction modes based on user intent, improving the user experience. It also optimizes error handling, allowing the system to quickly adjust strategies to address recognition errors and reduce the negative impact on the experience. Ultimately, real-time state switching enables the controller to adapt

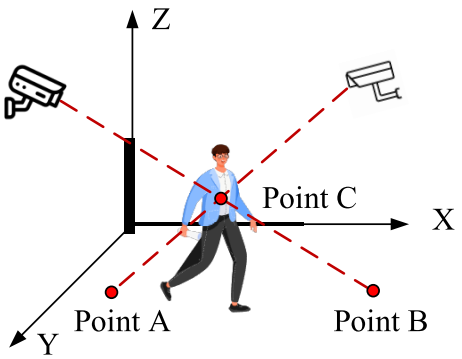


Fig. 3. Binocular positioning principle.

to complex scenarios, handle concurrent operations, ensure the accuracy and timeliness of interactive operations, and significantly improve the interaction quality of VR systems. However, in this study a universal 2D human keypoint definition methodwas used. This method lacks depth information and is difficult to accurately recover human posture, especially in occluded or complex motion scenes. Second, changes in perspective can cause keypoint positions to shift, which affects the stability of posture estimation. In addition, 2D methods are difficult to capture the 3D rotation information of human joints, which limits the accuracy of VR interaction. Therefore, research is needed to increase the dimensionality of human keypoint localization, combined with 3D keypoint detection or deep learning models, to improve the accuracy of human pose recognition in VR environments.

The ability to perceive depth, as well as the position of objects in three dimensions, is crucial for HCI in VR. This is achieved through the use of binocular vision, which enables the calculation of disparity, thereby enhancing both immersion and spatial perception capabilities. In comparison with monocular vision, binocular vision has been demonstrated to facilitate more precise distance measurement. In contrast to technologies that rely on LiDAR or depth cameras, binocular vision offers several advantages, including cost efficiency, broader applicability, and enhanced performance under variable lighting conditions, thereby mitigating recognition failure. The principle of the method is shown in Figure 3. The method uses dual lenses to detect the point simultaneously. Binocular vision provides depth information for HCI in VR by simulating the stereoscopic imaging mechanism of the human eye, significantly enhancing immersion and spatial perception. It achieves three-dimensional spatial reconstruction through disparity calculation, optimizing users' spatial positioning and interactive experience in virtual environments. Binocular vision can capture user posture and gestures in real time, and achieve natural and smooth interaction with the help of keypoint detection technology, especially

performing well in complex motion and occlusion scenes. In addition, binocular vision supports seamless integration of virtual and real environments, enhancing the interactive effects of augmented reality scenes. Its depth information can also optimize rendering performance by dynamically adjusting rendering resources, improving visual effects, and reducing computational resource waste. Binocular vision has strong adaptability and can work stably in different lighting and complex backgrounds, expanding the application scope of VR technology. However, the method is not adapted to more complex game scenes and is affected by light. The keypoint dimension enhancement algorithm for improving binocular vision technology can alleviate the problem of human occlusion in VR scenes by integrating deep learning with traditional visual geometry modeling methods. Its theoretical basis mainly comes from core technologies such as stereo matching, multi-view geometry, keypoint extraction, and dimension enhancement mapping. First, the algorithm relies on the disparity information of binocular vision by constructing the epipolar geometric relationship between the left and right cameras and combining it with a deep learning-based keypoint detection network to achieve accurate extraction of human joint points. Compared to monocular vision methods, binocular systems provide richer depth information, allowing the estimation of 3D positions based on unobstructed perspectives even when certain areas are obstructed. In addition, the keypoint dimensionality enhancement algorithm effectively completes missing keypoints caused by occlusion by high-dimensional mapping of low-dimensional 2D keypoint information, combined with spatiotemporal constraints and data-driven optimization strategies, such as Transformer based sequence modeling methods, and improves global consistency. The advantage of this method is that even if some joint points are occluded, the system can still infer their reasonable positions based on known joint topology relationships, thus reducing interaction errors caused by occlusion. Therefore, in this study the binocular stereo vision methodwill be improved. The the 2D pixel position by coordinate transformationwill be uplifted. Firstly, the calculation of converting the world coordinate system (CS) to the camera CS is shown in Equation (1).

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = W \otimes \begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} + T\,, \tag{1}$$

where (X_E, Y_E, Z_E) is the world CS, (X_C, Y_C, Z_C) is the camera CS, W is the rotation matrix, and T is the translation vector. Then the camera CS is converted to the image CS. The specific calculation is shown in Equation (2).

$$Z_C \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix}, \tag{2}$$

where (x, y, z) is the position of the same point in 2D image coordinates, a is the camera focal length, and Z_C is the depth coordinate. The conversion from image CS to pixel CS is performed. The specific calculation is shown in Equation (3).

$$\begin{bmatrix} m \\ n \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\mathrm{d}x} & 0 & m_0 \\ 0 & \frac{1}{\mathrm{d}y} & n_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{3}$$

where $\begin{bmatrix} m \\ n \\ 1 \end{bmatrix}$ is the transformed chi-square coordinates, $\frac{1}{\mathrm{d}x}$ and $\frac{1}{\mathrm{d}y}$ are the scaling factors, and m_0 and n_0 are the translations. In conclusion, it is possible to determine the transformation relationship between the global CS and the pixel CS. Equation (4) illustrates this particular computation.

$$Z_C \begin{bmatrix} m \\ n \\ 1 \end{bmatrix} = \lambda \otimes \begin{bmatrix} W & T \\ \vec{0} & 1 \end{bmatrix} \otimes \begin{bmatrix} X_E \\ Y_E \\ Z_E \\ 1 \end{bmatrix}, \tag{4}$$

where λ is the camera internal reference matrix. Camera calibration is critical for accurate 3D reconstruction, as it can eliminate lens distortion and provide internal and external parameters of the camera to improve reconstruction accuracy. The key steps include: image acquisition using a calibration board to obtain multi-angle images, feature point detection to extract corner or marker points, parameter estimation to compute internal parameters (focal length and principal points) and external parameters (position and rotation), aberration correction to correct for lens aberrations, and optimization and tuning to use nonlinear optimization to improve calibration accuracy. These steps ensure the accuracy of the camera model during the 3D reconstruction process and enhance the authenticity of spatial point cloud data. Among them, the internal reference calibration is calculated by the classical Zhang calibration method [21], which can find the distortion coefficient of the camera. The specific calculation is shown in Equation (5).

$$\mathrm{dist} = [\theta_1, \theta_2, \theta_3, \varphi_1, \varphi_2], \tag{5}$$

where θ is the radial distortion coefficient, and φ is the tangential aberration coefficient. Camera external parameter calibration can be carried out by changing the camera position and updating the position and attitude of the camera in the world CS. The specific flow of the keypoint dimensional enhancement algorithm is shown in Figure 4. In this Figure it can be seen that the keypoint dimensional enhancement algorithm consists of two parts: determining the internal and external parameters of the camera and increasing the dimension calculation of the data. Among them, the camera calibration stage

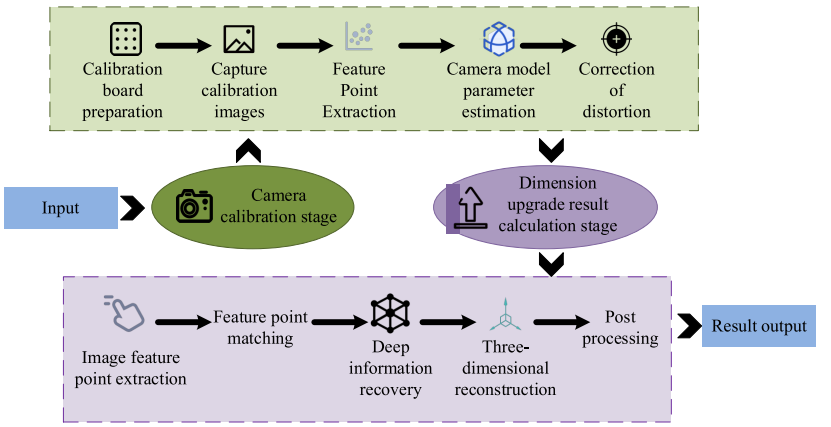


Fig. 4. The overall flow chart of the dimensional enhancement algorithm.

is the preparatory stage of the algorithm, which needs to be performed only once. The stage of calculating the result of increasing dimension needs to extract the feature points in the 2D image to be processed and match them with the feature points of the known 3D structure. The geometric structure of the 3D scene is reconstructed based on the position of each feature point in the 3D space and the recovered depth information. In addition to this, the reconstructed 3D model is optimized and smoothed to improve the accuracy and visual effect. Finally, the upscaled results such as depth map are output. The dimensionality enhancement algorithm for feature point extraction and depth recovery can effectively improve 3D scene reconstruction. First, key feature points can be extracted by deep learning or traditional methods to improve matching accuracy. Then, binocular disparity estimation or deep neural network can be combined to recover depth information. Next, dimensionality boosting algorithms are used to optimize point cloud distribution, enhance geometric details of sparse regions, and improve reconstruction accuracy. Finally, by integrating multi-perspective information and correcting errors, the 3D model becomes more accurate and coherent, resulting in higher quality virtual environment reconstruction. The advantages of the keypoint dimensionality enhancement algorithm based on improved binocular vision technology in terms of speed and accuracy are mainly reflected in efficient stereo matching, optimized depth estimation, and keypoint reconstruction strategies. Compared with traditional methods, this algorithm improves the efficiency of stereo matching by introducing an adaptive disparity optimization strategy and a multi-scale feature fusion mechanism, making depth computation more stable and reliable, while reducing computational overhead and improving real-time performance. In addition, this method combines sparse point cloud completion technology in the keypoint reconstruction process, resulting in higher human keypoint

reconstruction accuracy, especially in complex interactive scenes, which can provide more accurate motion capture. The theoretical basis for dealing with human occlusion in VR scenes lies in the disparity redundancy and depth compensation properties of binocular vision. Specifically, in the binocular imaging process, different camera angles can provide redundant information, allowing partially occluded keypoints to be inferred from unobstructed angles. This process alleviates the problem of keypoint loss caused by occlusion in monocular methods. Furthermore, the algorithm constructs spatial topological constraints based on graph neural networks, thereby enabling mutual constraints between detected keypoints and inferring the position information of partially occluded areas. This enables a more complete reconstruction of human body structure. Compared with traditional methods, this improved algorithm can handle human keypoint detection in complex scenes more stably. Even in occlusion situations, it can improve the prediction accuracy of keypoints through multi-view feature compensation and spatial relationship inference. At the same time, combined with optimized computation processes, it reduces computational complexity, making it faster and more accurate in interactive VR scenes.

3.2. Optimization study of PIFPAF algorithm

The study adopts an improved binocular vision technique for human posture keypoint positioning in VR human-computer games. However, to realize user action recognition and animation simulation in game interaction systems, the study needs to further improve the applicability and detection effect of the algorithm in different game scenarios. PIFPAF is an advanced human pose estimation method, which is especially suitable for multi-person pose detection in low-resolution and crowded scenes [18]. Its network structure is shown in Figure 5.

The key to the PIFPAF algorithm in human pose estimation lies in the synergistic effect of the two branches, part intensity field (PIF) and part association field (PAF). The PIF branch is mainly used to detect the location information of human keypoints, improve the accuracy of keypoint detection by predicting the density distribution of each joint, and combine Gaussian distribution to enhance local features, so that the model can accurately locate keypoints even when dealing with complex backgrounds and occlusion situations. As a high-precision positioning mechanism for each keypoint, it not only regresses the continuous spatial coordinates of keypoints, but also effectively enhances the robustness of the model to occlusion, attitude distortion, and low resolution keypoints through the collaborative modeling of heatmaps and displacement vectors. Especially, in human-computer interaction scenarios such as VR and virtual reality, the PIF branch can provide more accurate responses to local human features with higher density pixel level supervision, thereby significantly improving the system's perception ability. Gaussian distribution is used in the keypoint regression process to model the position distribution of each predicted keypoint. By generating a two-dimensional Gaussian heatmap centered on the keypoint on the feature map, accurate weighting of local

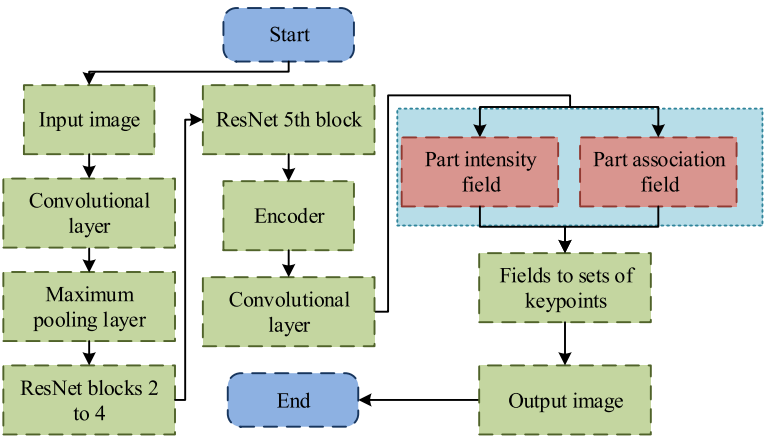


Fig. 5. Structural diagram of the PIPAF network

areas is achieved, thereby improving the accuracy of keypoint localization. In complex environments such as occlusion, lighting changes, or multi person interactions, Gaussian distribution can highlight the saliency of key areas, effectively suppress background interference, and enable the model to extract keypoint information more stably and accurately. This mechanism significantly enhances the robustness and detection performance of the PIFPAF model under high noise conditions. The PAF branch is responsible for learning the correlation information between different joints in the human body, using vector fields to represent the topological relationships between different joints, thereby maintaining structural consistency in multi-person interaction scenarios and effectively reducing the keypoint confusion problem. The combination of the two branches enables PIFPAF to achieve higher robustness in posture estimation. The PIF branch ensures accurate detection of keypoints, while the PAF branch ensures the rationality of the human body structure, especially in challenging scenarios such as occlusion, complex movements, and multi-person interaction. PAF can effectively utilize joint connection relationships for posture correction.

In addition, compared to traditional regression-based methods, PIFPAF's end-to-end optimization strategy allows the network to globally optimize posture estimation in terms of the entire structure, achieving a better balance between speed and accuracy. The network first receives raw image data and inputs it into the PIFPAF model, extracts features through convolutional layers, and downsamples at the max pooling layer. The encoder consists of multiple residual blocks to process features in depth. Then, the two branches separately generate keypoint field predictions. Finally, the decoder converts the feature map into a set of keypoints. Further research is conducted to optimize the

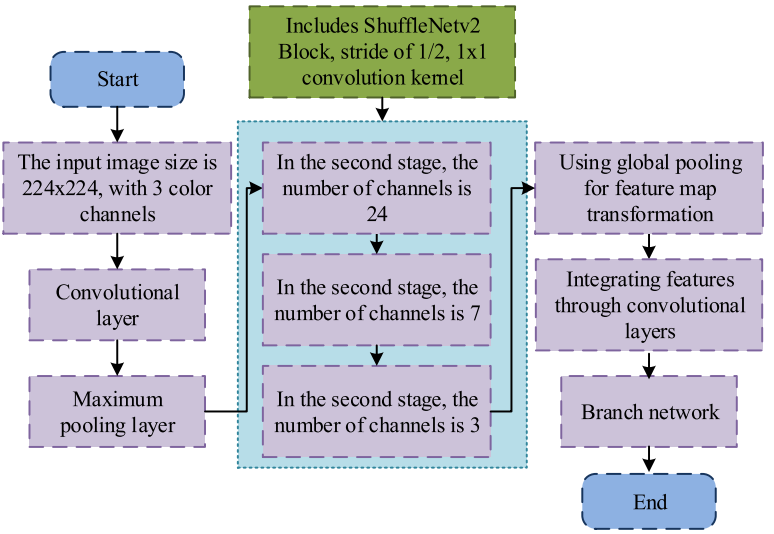


Fig. 6. Network structure in the improved PIFPAF feature extraction stage.

ResNet Block feature extraction network structure in the algorithm, in order to propose an improved PIFPAF algorithm. Its structure is shown in Figure 6.

ResNet has a large number of parameters, making training difficult. As shown in Figure 6, the convolutional layer is responsible for extracting local features of the image, which is crucial for identifying keypoints as it can capture key visual patterns in the image. Residual blocks alleviate the gradient vanishing problem in deep network training by introducing shortcut connections, allowing the network to train deeper layers more effectively and extract richer feature representations. These deep level features are particularly important for precise keypoint detection in complex environments, as they provide more contextual information and details. In addition, replacing ResNet Block with ShuffleNetv2 Block is to improve processing speed while maintaining accuracy. The design of ShuffleNetv2 Block is more lightweight and suitable for real-time applications, which is crucial for fast response and smooth interaction experience in VR environments. Feature extraction is performed on the original data after the replacement network, and the results are fed into the two-branch network for regression. The two-branch network's PIF branch is utilized to locate the important human body components and forecast each one's size, position, and confidence. Its output parameter set is calculated as shown in Equation (6).

$$P^{ij} = \{p_a^{ij}, p_x^{ij}, p_y^{ij}, p_o^{ij}, p_\tau^{ij}\}, \tag{6}$$

where i and j are the coordinates of the network output, p_a is the confidence map of the

pixel, p_o^{ij} is the correction parameter for computing the loss function, p_τ^{ij} is the Gaussian smoothing parameter, and p_x^{ij} and p_y^{ij} are the components of the offset vectors in the x and y directions of the keypoints closest to the pixel, respectively. The computation of the Gaussian function is specifically shown in Equation (7).

$$G(x,y) = \frac{1}{2\pi\tau^2} e^{-\frac{x^2+y^2}{2\tau^2}}, \tag{7}$$

where τ is the 2D form of the Gaussian function. Its bandwidth is positively correlated with the influence range of the function. Based on the calculation of the parameters and the function, the prediction results of the keypoint location can be obtained. Its calculation is specified in Equation (8).

$$F(x,y) = \sum_{ij} p_a^{ij} G(x,y|p_x^{ij}, p_y^{ij}, p_\tau^{ij}), \tag{8}$$

where $F(x,y)$ is the keypoint position prediction function. PAF, on the other hand, is used to connect the detected body parts through the association information to form a complete human posture. Its output parameter set is calculated as shown in Equation (9).

$$A^{ij} = \{a_a^{ij}, a_{x1}^{ij}, a_{y1}^{ij}, a_{o1}^{ij}, a_{x2}^{ij}, a_{y2}^{ij}, a_{o2}^{ij}\}, \tag{9}$$

where a_{x1}^{ij}, a_{y1}^{ij}, a_{x2}^{ij}, and a_{y2}^{ij} are the components of the offset vector on the horizontal axis x and vertical axis y, respectively, and a_{o1}^{ij} and a_{o2}^{ij} are correction functions. The result of the output of the network structure includes three types of outputs, and the loss values of the three types of outputs are calculated and summed to obtain the total output of the network. The specific calculation is shown in Equation (10).

$$\text{LOSSES} = \text{BCELoss} + \text{SCALELoss} + \text{REGLoss}, \tag{10}$$

where BCELoss is the confidence correlation output, REGLoss is the offset vector correlation output, and SCALELoss is the target scale related output.

In this way, in this study an optimized HCGI system is constructed. The local game interaction system and the cloud game running platform make up the majority of the system. Among them, the operation process of the game interaction system is as follows. First, the images are captured by two GB cameras to obtain the raw images of the current frame. Then, the AI performs algorithm calculations such as keypoint detection, keypoint uplift, gesture recognition, etc. on the captured images to generate the composed raw data. Then the data generated by the AI module is encapsulated to form a JSON file and sent to the cloud via SOCKET communication. The operation process of the cloud game platform first requires preliminary data processing, including data reception, parsing, and operation. According to the processed data, the game is

rendered, i.e. the processed data is used to generate JPG images. Then the rendered image data is sent back to the local machine via SOCKET communication, and the rendering effect is played in the local display module.

4. Performance analysis of optimized VR HCGI system

4.1. Experimental environment and data sources

In the experiment an improved binocular vision technology's keypoint dimensionality enhancement algorithm is used to evaluate the ability of the system to handle human occlusion and interactive performance in VR scenes. The AR game HCI experimental verification between HCI system and cloud game platform can be conducted. The software development environment for the experiment is Windows 10 operating system, PyCharm Community development tool, and PyTorch GPU deep learning environment. The hardware environment is RTX2060 6 G GPU and 16 G memory. In addition, the experimental environment also includes a high-performance GPU computing platform, and uses the Unity 3D engine and OptiTrack optical motion capture system to build a high-precision interactive VR test environment. The data acquisition of the experiment adopts a binocular stereo camera array to capture keypoint information under different occlusion conditions, and optimizes keypoint dimensionality and pose estimation through deep learning networks. The experimental setup includes several scenarios such as single person, multiple people, partial occlusion, and heavy occlusion to test the adaptability and robustness of the algorithm in different complex environments. Specific evaluation metrics include spatial accuracy indicators such as keypoint prediction accuracy, mean joint error, and posture estimation accuracy. The inference speed and computational complexity of the algorithm are measured simultaneously to evaluate its real-time performance. In addition, the experiment uses trajectory smoothness and latency indicators to verify the smoothness of the interaction, ensuring that the algorithm remains efficient and stable in complex interaction processes. The specific experimental scene is shown in Figure 7.

The layout and environmental characteristics of the experimental scenes have a significant impact on the performance of keypoint dimensionality enhancement algorithms in binocular stereo vision technology. As shown in Figure 7, the experiment is conducted in a specially designed VR interactive space with uniform and controllable lighting conditions to reduce the impact of lighting changes on depth estimation. The lighting equipment adopts a multi-angle light source arrangement at the top and side to ensure sufficient illumination in different directions while avoiding strong shadows or overexposure, thereby improving the image quality obtained by the binocular camera. The experimental space needs to ensure that participants have sufficient activity space to simulate real-world VR application scenarios. In the experimental environment, some obstacles

Fig. 7. Example of the experimental scene – a specially designed VR interactive space.

such as tables and chairs, simulated walls, or virtual interactive devices are appropriately placed to test the performance of the algorithm in complex occlusion situations. The presence of these obstacles can obscure keypoints of the human body, increasing the difficulty of inferring depth information. In addition, the scene may contain dynamically moving objects, such as other test subjects or virtual interactive elements, which may affect the stability of the stereo matching algorithms. By introducing different types of occlusion, such as partial occlusion and global occlusion, the adaptability of the algorithm in environments of varying complexity are evaluated. Environmental factors have a significant impact on the experimental results. First, lighting conditions can affect the quality of binocular matching. Too dark or high contrast environments can lead to errors in disparity calculation, thereby reducing the accuracy of keypoint dimensionality enhancement. Second, the arrangement of obstacles affects the occlusion pattern. If the occlusion is large or has strong reflective properties, it may introduce additional depth estimation noise. In addition, the background texture characteristics of experimental scenes can also affect the robustness of binocular matching. In complex backgrounds, false matches may increase. Therefore, it is necessary to optimize the background appropriately, such as using low-texture backgrounds to reduce interference. Finally, it is also necessary to consider the installation position and angle of the camera to ensure that the obtained binocular disparity information can fully cover important parts of the human body while avoiding depth distortion caused by viewing angle deviation.

The experiment faces several challenges and limitations during implementation and testing. First, human occlusion is complex and highly unpredictable, especially in multi-person interactions, where the uncertainty of the occlusion region affects the accuracy of keypoint dimensionality enhancement. Second, binocular stereo vision relies on high-quality image matching, but depth estimation can be subject to errors under changing lighting, dynamic backgrounds, or reflections. In addition, improving the algorithm has a high computational complexity, and optimizing computational efficiency while ensuring real-time performance has become a key issue. During the experimental process, it is

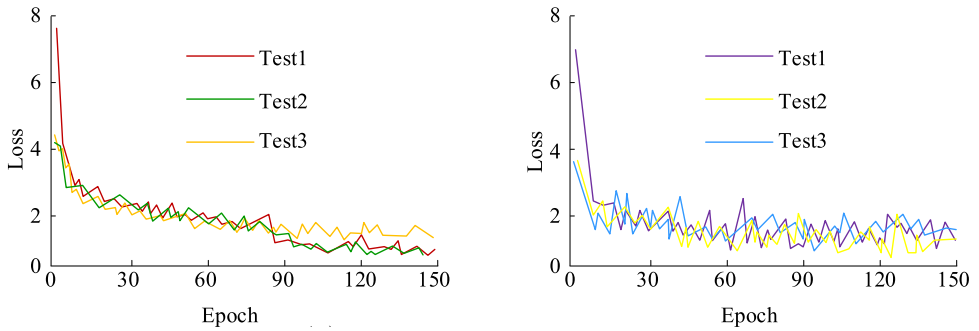
necessary to balance the accuracy of data annotation with the size of the data, ensuring that the occlusion data used for training is sufficiently rich to improve the generalization ability of the model. Hardware limitations are also a factor. Although high-performance GPUs are used for inference, the computational cost of the model still needs to be controlled to avoid delays that affect the VR interactive experience. Finally, due to the involvement of multiple device synchronizations in VR interaction, such as motion capture systems, VR headsets, and binocular cameras, time synchronization errors can affect the overall experimental results, requiring additional calibration steps to improve system consistency.

A total of 100 participants were recruited for the study, including 50 males and 50 females. The age distribution of the selected subjects includes children, adolescents, middle-aged, and elderly groups. Body types include lean, normal, and overweight. Moreover, all the participants are without any motor dysfunction.

4.2. Performance analysis of the keypoint dimensional enhancement algorithm with improved PIFPAF algorithm

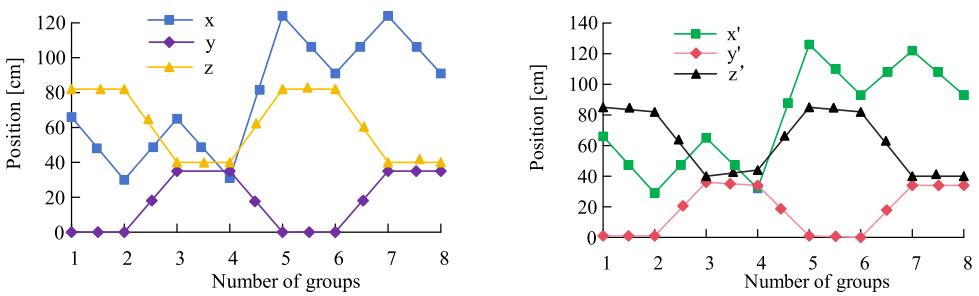
To investigate the effect of different training strategies of the upscaled human keypoint detection algorithm on the loss function of the dataset, the study uses Basenet and Headsnet to train the dataset, respectively. Basenet uses pre-trained model initialization and fine tuning on multi-scale feature maps. During the training process, random data augmentation is used to improve generalization ability, while the Adam optimizer is used to dynamically adjust the learning rate to avoid gradient oscillations. Headsnet uses a multi-task loss function combined with keypoint heatmaps and depth information monitoring to improve its ability to recover occluded areas. A total of 120 rounds of experiments were conducted. There were three groups of experiments. Test 1 and Test 3 were all trained with Basenet and Headsnet, respectively. Test 2 contained 50 rounds of each of the two types of training. The loss function variation curves obtained from the experiments are shown in Figure 8.

In Figure 8a, the loss functions of all three groups on the training set decrease with the number of training rounds, and decrease rapidly at the beginning and then stabilize. Among them, the starting value of the loss function of Test 1 is much higher than that of the other two groups, which is 8. The starting values of Test 2 and Test 3 are 4.2 and 4.3, respectively. The loss functions of Test 2 and Test 3 have a close trend in the early stage. However, in the later stage when Test 2 and Test 1 are stabilized, the loss function changes are closer to each other and both of them are roughly stabilized at about 1.5. Test 3 stabilizes with a slightly higher loss value, fluctuating within a range around 2. In Figure 8b, the loss function value of the three experimental training sets on the validation set decreases with the increase of training rounds. It decreases drastically in a short period of time, after which the change decreases. However, the volatility is relatively large, and all of them fluctuate in the range of 0.5 to 2.5. This may be



a Training set loss curve. **b** Verification set loss curve.

Fig. 8. Loss function of the experiment on the dataset.



a World coordinates of real points. **b** World coordinates of predicted points.

Fig. 9. Test results of the dimensional enhancement algorithm on the target position.

due to the diversity of data in the validation set or the difference in the generalization ability of the model on different data. It can be observed that Test 2 has more rounds in the validation set in which the loss function value achieves the minimum value.The experimental results show that the experimental group Test 2, which combines two network training strategies, has the best training effect. To further verify the feasibility of this keypoint dimensional enhancement algorithm, the experiment is to localize the target object by this algorithm and compare the error between the predicted and actual position. The world coordinates of the actual point are (x, y, z) and the world coordinates of the predicted point are (x', y', z'). A total of 8 experiments are conducted and the specific results are shown in Figure 9.

In Figure 9a, the position change range of the target object on the x-axis is large and fluctuates in the range of $[30, 120]$ cm. The position change curves of y-axis and z-axis are almost symmetrical to each other, and their change ranges are $[0, 35]$ cm and

Tab. 1. Results of performance comparison of different algorithms.

Algorithm name	Improved PIFPAF algorithm	OpenPose	ResNet50 + YoloV3
NE [%]	0.22	0.25	0.19
OKS	0.97	0.96	0.98
640×480 [fps]	13	4.2	0.75
320×240 [fps]	19	15	0.80
Extendibility	High	Middle	Middle
CPU utilization ratio [%]	15.4	23.1	30.5

$[40, 82]$ cm, respectively. In Figure 9b, the variation ranges of the target object in x-axis, y-axis and z-axis are $[29, 126]$, $[1, 36]$, $[40, 85]$ cm. Comparing the coordinate change curves of the target in Figure 9a and 9b, it can be observed that the coordinates of the predicted object position and the actual position using the keypoint dimensional enhancement algorithm are very similar to each other, and the total average absolute error is 2.11 cm. The experimental findings demonstrate that the keypoint dimensional enhancement method is capable of precisely capturing the target's shifting location in space. It has good robustness, and can adapt to different environments and changes in conditions. To investigate the performance of the improved PIFPAF algorithm, OpenPose, and ResNet50 + YoloV3 algorithms are compared. Two metrics, number of errors (NE) and object keypoint similarity (OKS) are calculated. Moreover, the speed of the algorithms is compared for different resolution images. OpenPose is a multi-stage CNN-based pose estimation algorithm that uses a bottom-up approach to detect human keypoints and analyzes limb structure through keypoint correlation. It is suitable for multi-person pose estimation. ResNet50 + YoloV3 combines deep residual networks with object detection algorithms, using ResNet50 to extract human features and YoloV3 for efficient object detection and localization, ensuring the accuracy and real-time performance of keypoint detection. The performance comparison between the two in VR HCI scenarios can help analyze the accuracy and speed advantages of keypoint detection. Table 1 displays the individual experimental outcomes.

In Table 1, the ResNet50 + YoloV3 algorithm performs best on the NE and OKS metrics with 0.19% and 0.98, respectively, with the lowest error rate and the highest keypoint similarity. OpenPose has the worst performance on both metrics, which may be related to the bottom-up approach adopted by OpenPose. The improved PIFPAF algorithm, on the other hand, performs in the middle, with NE and OKS of 0.22 and 0.97, respectively. However, in terms of processing speed, the improved PIFPAF algorithm performs faster in both 640×480 and 320×240 resolutions, with 13 fps and 19 fps, respectively. OpenPose's processing speed at 640×480 resolution is somewhere in between at 4.2 fps. However, the processing speed at 320×240 resolution is 15 fps, which is not much different from the improved PIFPAF algorithm. The processing speed of ResNet50 + YoloV3 is significantly lower than the other two algorithms, which may be

Tab. 2. Statistics of gesture recognition efficiency in 30 tests for each gesture.

Gesture	Gesture 1 *open palm*	Gesture 2 *clench fist*	Gesture 3 *thumbs up*	Gesture 4 *victory symbol*	Gesture 5 *pointing*
Correct identification in 1st run	28	26	29	27	28
Correct identification in 2nd run	1	3	0	1	1
Correct identification in 3rd run	0	1	1	1	1
Correct identification in 4th run	1	0	0	1	0
Correct identification in 5th run	0	0	0	0	0

due to the fact that the algorithm has sacrificed some of its speed in order to obtain higher accuracy. Due to the improvement of the algorithm structure and the use of parallel processing techniques, the testing findings demonstrate that the revised PIFPAF algorithm greatly boosts processing speed while maintaining higher accuracy.

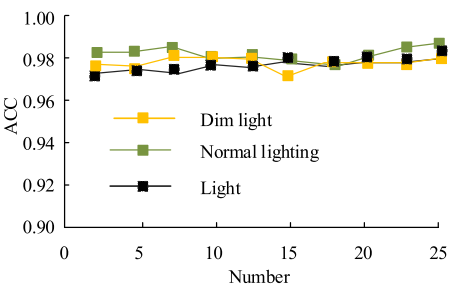
4.3. Performance analysis of optimized VR HCGI system

To further verify the recognition accuracy of the VR human-computer gaming system using the improved PIFPAF algorithm and binocular vision optimization for the actual user actions, in the experiment five static gestures. Moreover, 30 sets of tests were conducted to recognize the specified gestures in the interaction actions.

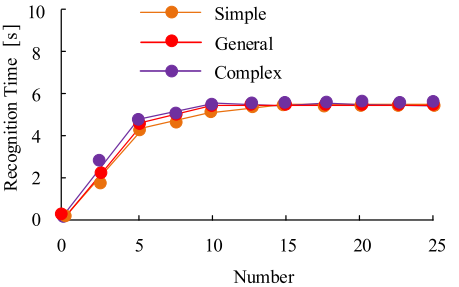
In order to standardize the evaluation of gesture recognition accuracy in interactive systems, five commonly used static gestures were defined and assigned identification numbers. Gesture 1 is *open palm*, Gesture 2 is *clench fist*, Gesture 3 is *thumbs up*, Gesture 4 is *victory symbol*, and Gesture 5 is *pointing*. These gestures were selected due to their clear and recognizable visual features, and were repeatedly tested throughout the entire recognition experiment to maintain consistent gesture identifiers.

The total number of times each gesture was correctly recognized in the repeated test is shown in Table 2. It can be seen that almost all of the five gestures were recognized by the interactive system in one recognition run. Among them, Gesture 3 is recognized in one recognition the largest number of times: 29, and Gesture 2 is recognized in one recognition the smallest number of times: 26. Moreover, almost most of the gestures are fully recognized in the first three runs. The results of the experiment demonstrate that the interaction system can meet the needs of the player by accurately identifying the user's gesture movements while they are playing.

The experiment further investigates the recognition of the optimized interactive system under different light or environment complexity conditions. Figure 10 displays the specific outcomes. The graphs in this Figure show the trend of the recognition accuracy of the system with the number of tests under different lighting environments. Figure 10b shows the variation of the recognition time with the number of tests under different environmental complexities. The recognition accuracy under the three lighting conditions



a The recognition accuracy of the system in different lighting environments.

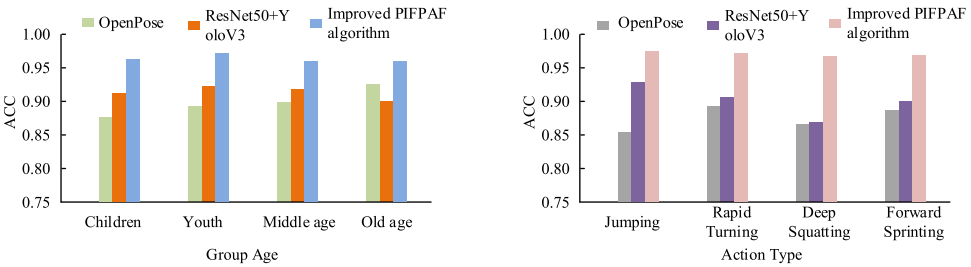


b The recognition time of the system in environments of different levels of complexity.

Fig. 10. System identification performance analysis under different environmental conditions.

shown in Figure 10a is relatively close, with small fluctuations around 0.98. This indicates that the system can maintain stable recognition performance under different lighting conditions. This may be due to the system's strong lighting robustness in the feature extraction and recognition algorithms, which can effectively adapt to different lighting environments. As shown in Figure 10b, there is a slight difference in the recognition time among the three environments in the initial stage. When the number of recognition is 5, the recognition times for simple, normal, and complex environments are approximately 4.1 s, 4.2 s, and 4.3 s, respectively. As the number of tests increases, the detection time of the three environments gradually stabilizes around 5.2 s. It can be concluded that the complexity of the environment has a relatively small effect on the recognition time of the system, and the system can achieve consistent and stable processing efficiency in environments of different complexity after adapting to the environment.

Further experiments are conducted to compare the accuracy of posture recognition among different user groups and dynamic scenarios. Among them, the experiment selects four movement postures for recognition: jumping, fast turning, deep squatting, and forward sprinting. The results are shown in Figure 11. The graphs in Figures 11a and 11b show the comparison of pose recognition accuracy for different age groups and motion poses of each algorithm. In Figure 11a, the improved PIFPAF algorithm performs best in all age groups, with an accuracy rate higher than 0.95. OpenPose performs poorly in all age groups, especially in children and middle-aged populations, with accuracy rates below 0.90. ResNet50 + YOLOv3 performs well in young and middle-aged populations. This may be due to the large deviation between the body types of children and the elderly and the standard dataset, which affects the accuracy of the model's keypoint detection. As shown in Figure 11b, the improved PIFPAF algorithm performs best in all motion types, with an accuracy rate around 0.97. The recognition accuracy range of OpenPose is $[0.85, 0.89]$. The performance of ResNet50 + YOLOv3 is in between, with a maximum recognition accuracy of about 0.93 for children. The unstable recognition accuracy of



a The system's posture recognition accuracy for users of different ages.

b The recognition accuracy of different algorithms for different dynamic scenes.

Fig. 11. Comparison of recognition effects of various algorithms on different groups and motion postures.

Tab. 3. Statistical analysis of keypoint detection algorithm performance.

Index	OpenPose	ResNet50 + YoloV3	Improved PIFPAF algorithm	Standard deviation	p
Key-point detection accuracy [%]	85.2	89.5	94.3	3.2	$p < 0.05$
Interaction response time [ms]	120.4	98.7	85.6	8.5	$p < 0.01$
Block recovery accuracy [%]	72.8	81.2	92.5	4.1	$p < 0.01$
False detection rate [%]	14.6	10.2	5.8	2.8	$p < 0.05$

OpenPose and ResNet50 + YOLOv3 may be attributed to the fact that activities such as jumping and rapid turning result in brief losses of body keypoints. In contrast, activities like deep squatting and sprinting forward cause significant displacement, making it challenging for single frame-based posture recognition algorithms to track reliably. The improved PIFPAF optimizes keypoint matching through binocular vision, maintaining high detection accuracy even under various intense movements.

To verify the feasibility of the proposed method, the experiment further conducts a significance test on the optimized VR HCGI system. The specific results obtained are shown in Table 3. According to this Table, the optimized VR human-machine game interaction system performs better in several key indicators. Among them, the accuracy of keypoint detection reached 94.3%, which is significantly improved compared to OpenPose's 85.2% and ResNet50 + YoloV3's 89.5%, with $p < 0.05$. This improvement is mainly due to the keypoint dimension enhancement algorithm of binocular vision technology, which effectively enhances the ability to capture spatial information and improves the accuracy of keypoint recovery under occlusion. In terms of interaction response time, the average processing time of the improved algorithm is only 85.6 ms, which is significantly optimized compared to the other two algorithms, with a $p < 0.01$. This optimization is due to the lightweight design of the algorithm structure, which

Tab. 4. Results of key point detection and 3D attitude estimation accuracy.

	Index	Experimental group 1	Experimental group 2	Control group 1	Control group 2
Key point detection accuracy	Average error of pixel standard	1.2	1.5	2	2.5
	Match success rate [%]	95.2	93.7	89.5	88.7
3D pose estimation accuracy	Average joint error [cm]	1.5	1.8	2.2	2.8
	Pose estimation accuracy [%]	94.3	92.5	88.7	85.6

makes the inference process more efficient and reduces the computational overhead. In terms of occlusion restoration accuracy, the improved algorithm reaches 92.5%, $p < 0.01$, The false detection rate of the proposed method is reduced to 5.8% ($p < 0.05$), further verifying the robustness of the improved algorithm.To further verify the effectiveness of epipolar geometry in improving the accuracy of keypoint detection in VR systems based on binocular vision, as well as the influence of coordinate transformation on the accuracy of 3D pose estimation, two experimental groups were set up: high calibration accuracy + epipolar geometry and high calibration accuracy + epipolar geometry, and two control groups: low calibration accuracy + epipolar geometry and low calibration accuracy + epipolar geometry for comparative experiments.

The results obtained are shown in Table 4. In terms of keypoint detection accuracy, the average pixel error of experimental group 1 is 1.2, and the matching success rate is 95.2%, both of which are better than the average error of 1.5 and the success rate of 93.7% in experimental group 2. The performance of the control group was poor, with an average error of 2 and 2.5 for control group 1 and control group 2, respectively, and a success rate of 89.5% and 88.7%, respectively. In terms of 3D pose estimation accuracy, the average joint error of experimental group 1 is 1.5 cm, and the pose estimation accuracy is 94.3%, which is also better than experimental group 2. The control group had larger errors of 2.2 cm and 2.8 cm, respectively, and lower accuracy rates of 88.7% and 85.6%. In summary, the experimental group outperformed the control group in keypoint detection and 3D pose estimation, indicating that high-precision camera calibration and coordinate transformation methods can significantly improve the accuracy of 3D pose estimation, reduce average joint error, and improve the accuracy of pose estimation. Experimental group 1 performed the best, indicating that the algorithm using epipolar geometry constraints significantly outperformed the algorithm without epipolar geometry in terms of keypoint detection accuracy and 3D pose estimation accuracy.

5. Discussion

The experimental results showed that the optimized VR HCGI system outperformed traditional methods in terms of keypoint detection accuracy, interaction response speed, and occlusion recovery ability, thus improving the real-time interaction experience in virtual environments. This optimization rendered VR devices more adaptable to complex action recognition, multi-user interaction, and occlusion environments, and it could be widely applied in immersive gaming, remote collaboration, rehabilitation training, and other fields. For example, in sports VR games, the system must accurately recognize large movements such as running and jumping to provide real feedback. In rehabilitation training, optimizing action recognition for different ages and physical conditions ensures safety and effectiveness. It provided a new solution for the development of VR interaction technology and laid the foundation for optimizing future intelligent HCI systems. This advantage was mainly due to the application of binocular vision technology combined with the keypoint dimensionality enhancement algorithm, which could more accurately restore occluded keypoints and improve the stability of detection. In terms of keypoint detection accuracy, experimental results indicated that the improved algorithm achieved 94.3%, which was significantly improved compared to OpenPose (85.2%) and ResNet50 + YoloV3 (89.5%). This advantage was mainly due to the deep information fusion of binocular vision, which allowed the system to exploit multi-view features and reduce the error of monocular methods in occluded scenes. In addition, the keypoint dimensionality enhancement algorithm enhanced local features and optimized globally, making keypoint localization more accurate. In terms of interaction response time, the optimized algorithm had an average processing time of 85.6 ms, which was nearly 30% less than OpenPose's 120.4 ms. This improvement was mainly due to the improved network structure, which used a lightweight CNN for feature extraction and reduced computational complexity by optimizing feature matching strategies, making inference faster. Compared to traditional deep learning methods, this algorithm was more suitable for real-time interactive applications and improved the user experience. In terms of occlusion restoration accuracy, the improved algorithm achieved 92.5%, a 20% improvement over OpenPose's 72.8%. This improvement was due to the introduction of binocular depth estimation, which allowed the system to make reasonable inferences based on the spatial information of other keypoints even when some keypoints were occluded. The accuracy was higher compared to methods based on monocular RGB images. In addition, by combining the Transformer structure for global feature modeling, the system could infer missing parts from the full pose distribution, further improving robustness. Compared with other studies, some existing research used long short-term memory networks or gated recurrent units for temporal modeling. However, their computational complexity was large and difficult to meet real-time interaction requirements. The improved

model used in this study achieved a better balance between computational complexity and accuracy, and was suitable for efficient HCI systems in VR scenes.

In the process of VR interaction, synchronizing facial keypoint data with voice input to improve character performance and realism faces many challenges. Firstly, the synchronization of data collection is a crucial issue. The capture of facial keypoints relies on visual sensors, while voice input relies on audio devices, and there are differences in sampling rate and processing speed between the two, resulting in difficulties in aligning data on the timeline. Secondly, the real-time requirements are extremely high. The VR environment requires a low latency interactive experience, and the synchronization processing of facial expressions and speech needs to be completed in a very short time, which puts extremely high demands on the efficiency of algorithms and hardware performance. In addition, robustness in complex scenarios is also a challenge. In environments with multiple interactions or noisy backgrounds, the accuracy of facial keypoint detection and speech recognition can be affected, which in turn affects the synchronization effect. Finally, the difference in personalized expression is also a problem. There are significant differences in facial expressions and voice tones among different users. How to preserve these personalized features during synchronization while achieving natural and smooth interaction is a direction that needs further research in the future.

6. Conclusion

To capture and analyze the user's gesture in real time to ensure real-time performance in VR environment so as to provide a smoother and intuitive interaction experience, in this study the PIFPAF algorithm was improved. It was also combined with binocular vision technology to optimize the VR HCGI operation. The experimental results indicated that the loss functions of all three tested groups decreased with the increase of training rounds and then stabilized. Among them, Test 2 and Test 1 were closer to each other in terms of the variation of the loss function as they stabilized on the training set, both roughly stabilizing around 1.5. Test 3 had slightly higher loss values as it stabilized, fluctuating in the range around 2. The loss function values of the three sets of experimental training on the validation set fluctuated in the range of 0.5 to 2.5 in the later stages, and Test 2 had the highest number of rounds in which the loss function value achieved the minimum in the validation set. The overall results of the performance verification of the keypoint dimensional enhancement algorithm revealed that before and after using this algorithm, the predicted object positions were very similar to the coordinates of the actual positions, and the total average absolute error was 2.11 cm. The experimental results indicated that the experimental group combining the two network training strategies had the best training effect, and the keypoint dimensional enhancement algorithm could accurately capture the moving position of the target in space with good feasibility. The study

demonstrated that the proposed method exhibited favorable applicability and accuracy in gaming scenarios.

However, the research is still limited by experimental conditions in specific environments, and the robustness under complex lighting and extreme occlusion conditions still needs to be improved. Therefore, future research will optimize the generalization ability of the model and combine it with deep learning to improve interaction accuracy. This can improve the real-time and accuracy of VR interaction systems and provide new ideas for the development of intelligent HCI technology.

Authors' declarations

Conflict of interest

The authors have no conflict of interest to report.

Data availability

The information on data are included in the manuscript.

References

[1] M. Akram, S. Siddique, and M. G. Alharbi. Clustering algorithm with strength of connectedness for m-polar fuzzy network models. *Mathematical Biosciences and Engineering* 19(1):420–455, 2022. doi:10.3934/mbe.2022021.

[2] K. Bonnen, J. S. Matthis, A. Gibaldi, M. S. Banks, D. M. Levi, et al. Binocular vision and the control of foot placement during walking in natural terrain. *Scientific reports* 11(1):20881, 2021. doi:10.1038/s41598-021-99846-0.

[3] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43(1):172–186, 2021. doi:10.1109/TPAMI.2019.2929257.

[4] B. M. S. Hasan and A. M. Abdulazeez. A review of principal component analysis algorithm for dimensionality reduction. *Journal of Soft Computing and Data Mining* 2(1):20–30, 2021. doi:10.30880/jscdm.2021.02.01.003.

[5] K. Head-Marsden, J. Flick, C. J. Ciccarino, and P. Narang. Quantum information and algorithms for correlated quantum matter. *Chemical Reviews* 121(5):3061–3120, 2021. doi:10.1021/acs.chemrev.0c00620.

[6] A. R. Inturi, V. M. Manikandan, and V. Garrapally. A novel vision-based fall detection scheme using keypoints of human skeleton with long short-term memory network. *Arabian Journal for Science and Engineering* 48(2):1143–1155, 2023. doi:10.1007/s13369-022-06684-x.

[7] J. Katona. A review of human–computer interaction and virtual reality research fields in cognitive infocommunications. *Applied Sciences* 11(6):2646, 2021. doi:10.3390/app11062646.

[8] T. Kosch, R. Welsch, L. Chuang, and A. Schmidt. The placebo effect of artificial intelligence in human–computer interaction. *ACM Transactions on Computer-Human Interaction* 29(6):1–32, 2023. doi:10.1145/3529225.

[9] S. Kreiss, L. Bertoni, and A. Alahi. PifPaf: Composite fields for human pose estimation. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11969–11978. IEEE Computer Society, 2019. doi:10.1109/CVPR.2019.01225.

[10] Z. Q. Lan, J. X. Wang, and L. Q. Wang. Multi-view line matching based on multi-view stereo vision and leiden graph clustering. *Journal of Geo-Information Science* 26(7):1629–1645, 2024. doi:10.12082/dqxxkx.2024.240080.

[11] K. Li and X. Li. AI driven human–computer interaction design framework of virtual environment based on comprehensive semantic data analysis with feature extraction. *International Journal of Speech Technology* 25(4):863–877, 2022. doi:10.1007/s10772-021-09954-5.

[12] Y. Lin, W. Chi, W. Sun, S. Liu, and D. Fan. Human action recognition algorithm based on improved resnet and skeletal keypoints in single image. *Mathematical Problems in Engineering* 2020(1):6954174, 2020. doi:10.1155/2020/6954174.

[13] N. S. Logan, H. Radhakrishnan, F. E. Cruickshank, P. M. Allen, P. K. Bandela, et al. Imi accommodation and binocular vision in myopia development and progression. *Investigative Ophthalmology & Visual Science* 62(5):4, 2021. doi:10.1167/iovs.62.5.4.

[14] Z. Lyu. State-of-the-art human-computer-interaction in metaverse. *International Journal of Human–Computer Interaction* 40(21):6690–6708, 2024. doi:10.1080/10447318.2023.2248833.

[15] J. Ramadoss, J. Venkatesh, S. Joshi, P. K. Shukla, S. S. Jamal, et al. Computer vision for human-computer interaction using noninvasive technology. *Scientific Programming* 2021(1):3902030, 2021. doi:10.1155/2021/3902030.

[16] J. C. A. Read. Binocular vision and stereopsis across the animal kingdom. *Annual Review of Vision Science* 7(1):389–415, 2021. doi:10.1146/annurev-vision-093019-113212.

[17] G. R. E. Said. Metaverse-based learning opportunities and challenges: a phenomenological metaverse human–computer interaction study. *Electronics* 12(6):1379, 2023. doi:10.3390/electronics12061379.

[18] S. Seinfeld, T. Feuchtner, A. Maselli, and J. Müller. User representations in human-computer interaction. *Human–Computer Interaction* 36(5-6):400–438, 2021. doi:10.1080/07370024.2020.1724790.

[19] W. Xu, B. Chen, Y. Hu, and J. Li. A novel wide-band directional music algorithm using the strength proportion. *Sensors* 23(9):4562, 2023. doi:10.3390/s23094562.

[20] J. Zhang, Z. Chen, and D. Tao. Towards high performance human keypoint detection. *International Journal of Computer Vision* 129(9):2639–2662, 2021. doi:10.1007/s11263-021-01482-8.

[21] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In: *Seventh IEEE International Conference on Computer Vision*, vol. 1, pp. 666–673, 1999. doi:10.1109/ICCV.1999.791289.

[22] Z. Zimiao, X. Kai, W. Yanan, Z. Shihai, and Q. Yang. A simple and precise calibration method for binocular vision. *Measurement Science and Technology* 33(6):065016, 2022. doi:10.1088/1361-6501/ac4ce5.

A review: Machine Learning Techniques of Brain Tumor Classification and Segmentation

Iliass Zine-dine*, Jamal Riffi, Khalid El Fazazy, Ismail El Batteoui, Mohamed Adnane Mahraz and Hamid Tairi

Laboratory of Informatics, Signals, Automatics and Cognitivism (LISAC), Faculty of Sciences Dhar El Mehraz (FSDM), Sidi Mohamed Ben Abdellah University (USMBA), Fez, Morocco

**Corresponding author: Iliass Zine-dine (zinedine.iliass@gmail.com)*

Submitted: 11 Mar 2025 **Accepted:** 13 May 2025 **Published:** 01 Sep 2025

Abstract Classifying brain tumors in magnetic resonance images (MRI) is a critical endeavor in medical image processing, given the challenging nature of automated tumor recognition. The variability and complexity in the location, size, shape, and texture of these lesions, coupled with the intensity similarities between brain lesions and normal tissues, pose significant hurdles. This study focuses on the importance of brain tumor detection and its challenges within the context of medical image processing. Presently, researchers have devised various interventions aimed at developing models for brain tumor classification to mitigate human involvement. However, there are limitations on time and cost for this task, as well as some other challenges that can identify tumor tissues. This study reviews many publications that classify brain tumors. Mostly employed supervised machine learning algorithms like support vector machine (SVM), random forest (RF), Gaussian Naive Bayes (GNB), k-Nearest Neighbors (K-NN), and k-means and some researchers employed convolutional neural network methods, transfer learning, deep learning, and ensemble learning. Every classification algorithm aims to provide an accurate and effective system, allowing for the fastest and most precise tumor detection possible. Usually, a pre-processing approach is employed to assess the system's accuracy; other techniques, such as the Gabor discrete wavelet transform (DWT), Local Binary Pattern (LBP), Gray Level Co-occurrence Matrix (GLCM), Principal Component Analysis (PCA), Scale-Invariant Feature Transform (SIFT) and the descriptor histogram of oriented gradients (HOG). In this study, we examine prior research on feature extraction techniques, discussing various classification methods and highlighting their respective advantages, providing statistical analysis on their performance.

Keywords: brain tumor, feature extraction, machine learning, deep learning.

1. Introduction

In today's society, health issues are more common than ever, and people's lifestyles are also getting more and more unhealthy [18]. In the human body, brain is the most complex organ; it is composed of nerve cells and tissues that regulate the most fundamental bodily functions, such as muscle movement, breathing, and the senses. Brain tumors are one of the most feared diseases in medical science because they are a type of tumor that affects the central nervous system [37]. According to 2016 cancer statistics provided by the World Health Organization (WHO), brain tumors are treated as the leading cause of cancer. The challenge of manually classifying brain tumor MR images with comparable structures or appearances is demanding and complicated. Classification of brain tumor

MR images with similar structures or appearances is a difficult and challenging task, to solve this issue, automated classification might be used to categorize MR images of brain tumors with the least amount of radiologists' involvement.

In recent years, medical image processing has emerged as a crucial tool for the early detection of brain cancer, attracting significant attention from researchers worldwide [54]. Efforts are focused on developing models to assist specialists in accurately predicting the presence of tumors [19]. Despite the challenges faced by developers, such as variations in image composition, dimensions, and pixel quality, artificial intelligence—particularly computer vision—plays a pivotal role in advancing the digitalization of medical diagnostics and enhancing active research in this field [41]. Deep learning (DL), a subset of machine learning, enables computers to discover data representations, anticipate future outcomes, and draw conclusions based on factual information. These techniques are considered among the most significant computational intelligence strategies and are widely applied in medical image classification [30]. However, without a pre-processing phase and effective feature extraction methods, many of these strategies fail to deliver their expected benefits [7]. Recently, machine learning (ML) and DL algorithms have gained prominence as powerful tools for medical image classification, with transformers and auto-encoders playing a critical role in addressing various challenges in the field.

Convolutional neural networks (CNNs) and vision transformers (ViTs), in capturing complex patterns and semantic details from medical images, thereby improving classification performance [3]. Autoencoders, commonly utilized in unsupervised learning, are instrumental in deriving meaningful representations from raw image data, aiding in feature identification and dimensionality reduction [2]. Moreover, Generative Adversarial Networks (GANs) offer the distinct ability to produce synthetic medical images, enhancing data augmentation and increasing the diversity of training datasets, which contributes to the creation of more robust classification models for medical imaging applications.

The accuracy of brain tumor data classification is influenced by various factors, including the type and complexity of the data, such as image composition, dimensions, and pixel quality. It also depends on the methods employed, the techniques used for feature extraction, and the parameters of the algorithms implemented in the approach [45].

The structure of this article is as follows. In Section 2 the search strategy is outlined. In Section 3 the existing literature is analysed in detail. Finally, in Section 4 the conclusions of the study and the proposed directions for future research are presented.

2. Search strategy

In our study numerous significant manuscripts employing various methods and techniques for brain tumor classification were studied. These articles were sourced from

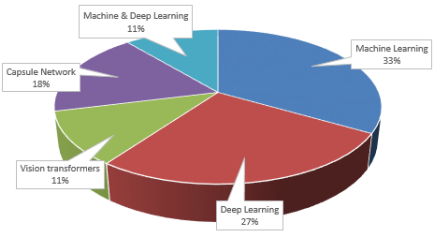


Fig. 1. The percentage of articles reviewed in this study.

platforms such as Google Scholar [58] and ScienceDirect [59]. Medical image classification approaches often leverage diverse machine learning algorithms and convolutional neural network architectures, including VGG, ResNet, AlexNet, and others. These methods incorporate distinct feature extraction techniques, such as descriptors, filters, and Gabor transforms. Additionally, advanced techniques like vision transformers and autoencoders have gained prominence, offering the ability to extract meaningful representations from image data and significantly improving image analysis and classification [52]. These approaches are complemented by standard preprocessing techniques, including resizing, normalization, data augmentation, and center cropping, which are commonly applied in the initial stages of image analysis workflows.

In this review, the referenced studies were systematically categorized according to the primary methodology employed: traditional Machine Learning, Deep Learning, Capsule Networks, and Vision Transformers. Approximately 33% of the cited articles focused on classical ML approaches, leveraging algorithms such as Support Vector Machines, Random Forests, and k-Nearest Neighbors. These methods often relied on handcrafted feature extraction techniques including Local Binary Patterns (LBP), Discrete Wavelet Transform (DWT), and Gray Level Co-occurrence Matrix (GLCM). Deep Learning-based studies accounted for around 27% of the references, with CNNs being the dominant architecture. These approaches demonstrated improved performance through automatic feature extraction and were frequently trained and evaluated on publicly available datasets such as BraTS [56], ISLES [55], and Figshare [57]. In addition to the individual contributions of Machine Learning (33%) and Deep Learning (27%) approaches, a notable 11% of the cited studies employed a hybrid ML & DL classification methodology, combining handcrafted features with deep feature representations to enhance classification accuracy. Capsule Networks were examined in roughly 18% of the cited work, offering robust spatial feature representation and enhanced interpretability, particularly in scenarios involving affine transformations. Vision Transformers, representing about 11% of the corpus, are an emerging trend, providing state-of-the-art performance by modeling global image context through self-attention mechanisms. Figure 1 illustrates the percentage of articles reviewed in this study.

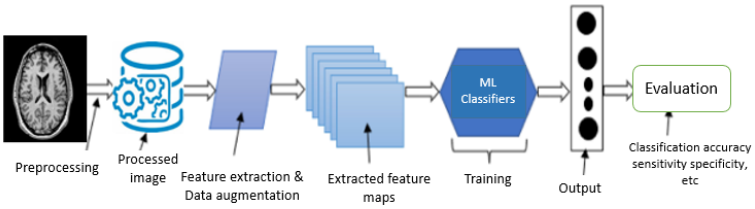


Fig. 2. Overview of the essential modules in a conventional ML-based brain tumor classification.

3. Analysis of the literature

The classification and segmentation of brain tumors remain an active area of research. Many researchers are exploring this topic, utilizing various techniques mentioned earlier to develop approaches with improved performance. The tables 1, 3, 4, 5, 6 below summarize the methods used in this field, including classification techniques, feature extraction methods, and the datasets employed.

3.1. Machine learning methods

Machine learning algorithms are among the most widely used methods for brain tumor classification, renowned for their effective detection capabilities. A key objective in many studies is to improve classification performance, which can be achieved through various methods and techniques applied at different stages. Enhancements may occur during dataset preprocessing, where traditional image processing techniques are implemented, or during the feature extraction phase, leveraging descriptors and neural network architectures. Furthermore, optimization during the classification phase, such as fine-tuning the algorithm's parameters, plays a crucial role in achieving superior results. Together, these efforts contribute significantly to improving the accuracy of classification outcomes. Figure 2 presents an overview of the essential modules in a conventional ML-based brain tumor classification.

Table 1 presents a comparison of studies that utilize different machine learning models, various feature extraction techniques, and diverse datasets to predict the classification accuracy of brain tumors.

Based on the findings presented in Tab. 1, it is evident that multiple factors play a role in enhancing the efficacy of brain tumor classification. Each approach employs specific methods and techniques tailored to its primary objective, encompassing various phases to achieve optimal results:

The standard data pre-processing stage is deemed crucial in the machine learning workflow, as it ensures that the data is appropriately configured for the application of

Tab. 1. Comparison of Machine Learning Models, Feature Extraction Methods, and Datasets for Brain Tumor Classification Accuracy.

Ref	Classification Method	Feature Extraction	Dataset	Accuracy
[27]	Machine Learning Methods Classifier	Crop, Resize, Augmentation, Transfer Learning	253 MRI, 3000 MRI, 3064 MRI	90%, 97%, 90%
[23]	LSTM	LBP, CNN	154 MRI	98%
[28]	Machine Learning	LBP	3064 MRI	95%
[33]	SVM, KNN, SRC, NSC, and the k-means	Wavelet, Statistical features	BraTS 2017	96%
[1]	Random Forest	Gray Level, LBP, HOG	BraTS 2013	93%
[12]	Random Forest Classifier	RGB to Gray, Resize, LBP, HOG, SFTA, GWF	BraTS 2012, BraTS 2014, BraTS 2015, BraTS 2017	90%, 89%, 94%, 91%
[38]	SVM Classifier, AC-CLS Segmentation	RGB to Graylevel Histogram Equalization, KMFCM	41 MRI	99%
[29]	LSTM	CNN, DWT	3064 MRI	98%
[14]	Support Vector Machine, K Nearest Neighbors, Neural Network, ELM	Resize, Watershed segmentation, morphological process, Wavelet	16 MRI	96%
[21]	Decision Tree, Multi-Layer Perceptron	Sigma Filter, Adaptive threshold, Region Detection, Binary Object Feature	174 MRI	95%, 91%
[11]	Machine Learning Methods Classifier	Weiner filter, Potential Field clustering, threshold, morphological dilation, LBP, GWT	86 MRI, BraTS 2013, BraTS 2015	93%, 93%, 97%
[42]	MLP Naïve bayes	RGB to Grey (Binarization), Median Filter (Noise Remove), edge detection, watershed, GLCM	212 MRI	98%, 91%
[51]	Machine Learning, Ensemble Learning	Crop, Resize, Augmentation, DWT, HOG	253 MRI	92%
[43]	Support vector machine	DTI analysis, Perfusion analysis, segmentation, normalization	141 MRI	97%
[39]	Support vector machine	Contrast Stretching, Augmentation, Transfer learning AlexNet, GoogLeNet, VggNet	3064 MRI	98%

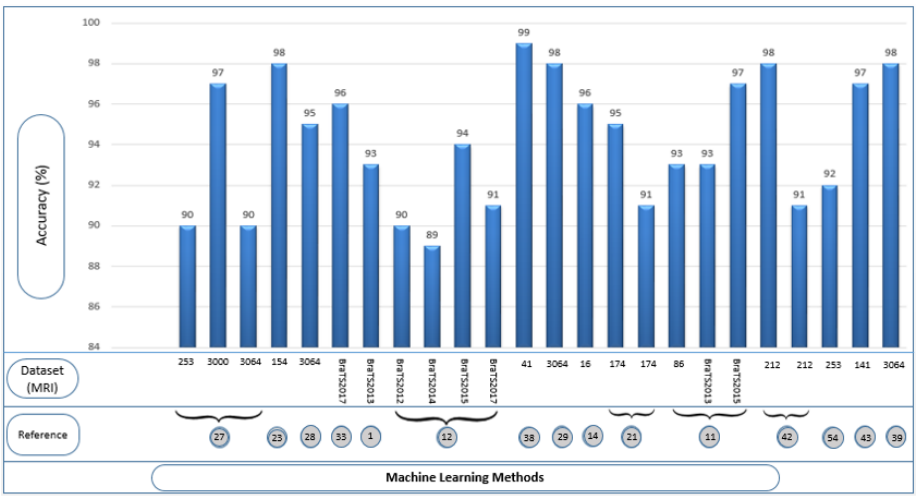


Fig. 3. The classification accuracy reported in each brain tumor study, based on machine learning algorithms and their respective datasets. Labels given in the row "Reference" are related to literature references according to Tab. 2, p. 37.

learning algorithms, thereby enhancing the quality, convergence, and performance of resultant models. This phase encompasses techniques such as data cleaning, normalization, scaling, and augmentation, all of which are recommended for thorough examination.

The feature extraction phase plays a pivotal role in enhancing data representation and reducing dimensionality for improved interpretability and comprehension. Various techniques, including CNN layers, LBP, DWT, HOG, GLCM, dilation, and filters, are commonly employed in this phase, each serving a specific purpose. Making the right choice of technique can significantly enhance classification accuracy. In the final phase, known as the classification or decision-making phase, the selection of parameters for the classification algorithm significantly impacts the effectiveness of the approach.

Figure 3 illustrates the highest accuracy rates achieved for brain tumor classification across different datasets. These accuracies were obtained through the application of various machine learning methods, highlighting the effectiveness of the employed classification techniques. Notably, the preprocessing and feature extraction methods played a crucial role in enhancing the model performance. By refining the input data, reducing noise, and selecting the most relevant features, these techniques contributed significantly to the high accuracy observed in the figure. This evaluation underscores the importance of carefully designing preprocessing pipelines and feature extraction strategies to optimize classification performance in brain tumor diagnosis.

Traditional machine learning algorithms, while effective in numerous classification

Tab. 2. Relations of labels given in Figs. 3, 5, 6, 8, 10 in the row 'References' to the literature references denoted here as 'Ref.'.

Label	Ref.	Label	Ref.	Label	Ref.	Label	Ref.	Label	Ref.
①	[1]	④	[4]	⑤	[5]	⑥	[6]	⑦	[7]
⑧	[8]	⑨	[9]	⑩	[10]	⑪	[11]	⑫	[12]
⑬	[13]	⑭	[14]	⑮	[15]	⑯	[16]	⑰	[17]
⑳	[20]	㉑	[21]	㉓	[23]	㉔	[24]	㉕	[25]
㉖	[26]	㉗	[27]	㉘	[28]	㉙	[29]	㉛	[31]
㉜	[32]	㉝	[33]	㉞	[34]	㊱	[36]	㊳	[38]
㊴	[39]	㊵	[40]	㊷	[42]	㊸	[43]	㊹	[44]
㊻	[46]	㊼	[47]	㊽	[18]	㊾	[48]	㊿	[49]
㉜	[50]	㊿④	[51]	㊿⑥	[53]				

tasks, exhibit several limitations when applied to complex medical imaging scenarios. One of the primary challenges lies in their reliance on handcrafted feature extraction, which often demands significant domain expertise and may fail to capture the full intricacies of high-dimensional medical data such as MRI scans. This manual process can lead to suboptimal performance, particularly in cases where subtle spatial patterns are critical for accurate tumor classification or segmentation. Furthermore, traditional ML models typically struggle with generalization when applied to diverse datasets or varying imaging conditions. To address these shortcomings, deep learning techniques—especially convolutional neural networks—have emerged as a powerful alternative. These models are capable of automatically learning hierarchical features directly from raw data, reducing the dependency on manual intervention and enhancing model robustness. By capturing both low-level and high-level features through stacked layers, deep learning architectures offer improved performance and scalability, making them more suitable for complex brain tumor analysis tasks. As a result, the shift from traditional ML to DL represents a significant advancement in the development of more accurate and automated diagnostic tools.

3.2. Deep learning methods

Convolutional Neural Networks are a type of multi-layer feedforward artificial neural network, initially inspired by the visual cortex [22]. CNNs play a pivotal role in deep learning and have emerged as one of the most commonly used architectures in recent

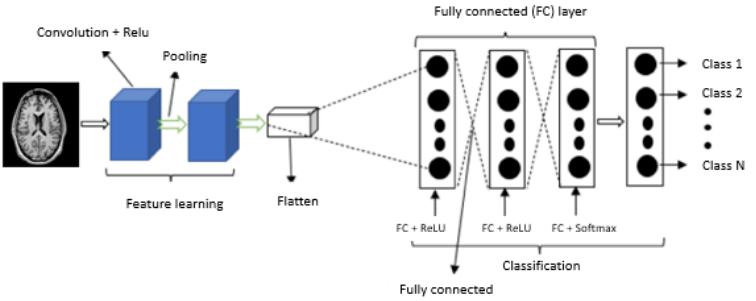


Fig. 4. Illustration showing the fundamental layers of a Convolutional Neural Network.

years, particularly for image recognition tasks. They excel in performing complex operations through convolution filters, which enable effective feature extraction. The convolutional layers in CNNs progressively learn intricate visual patterns from raw input data by applying filters to detect features such as edges, textures, and patterns in images. This hierarchical representation of data not only facilitates a deeper understanding of the inherent structures within the data but also significantly enhances classification performance. The initial layer in a Convolutional Neural Network serves to introduce the input image into the model, initiating the processing sequence through subsequent layers. As the data progresses, convolutional operations, pooling layers, and activation functions work collaboratively to extract meaningful and abstract features from the input. These features are then passed to one or more fully connected layers, which play a crucial role in tasks such as classification, segmentation, or detection of objects within the image. Ultimately, the final output is produced by the output layer, which delivers the network's prediction or decision. A typical CNN structure is depicted in Figure 4.

Table 3 presents a comparison of studies that utilize different deep learning architectures, various feature extraction techniques, and diverse datasets to predict the classification accuracy of brain tumors.

The findings in the table underscore critical factors contributing to the optimization of brain tumor classification methods, with each approach utilizing specific methods and techniques across various phases:

- Data Preprocessing: This phase is vital for preparing data for learning algorithms, which enhances model quality, convergence, and overall performance. Techniques such as data cleaning, normalization, scaling, and augmentation play an essential role in ensuring the data is well-suited for analysis.
- Feature Extraction: A key step in improving data representation and reducing dimensionality, feature extraction enhances interpretability and contributes significantly to classification accuracy. Methods like CNNs layers, local binary patterns (LBP), discrete wavelet transforms (DWT), histograms of oriented gradients (HOG), gray-level

Tab. 3. Comparison of Deep Learning Architectures, Feature Extraction Methods, and Datasets for Brain Tumor Classification Accuracy.

Ref.	Classification Method	Feature Extraction	Dataset	Accuracy
[31]	CNN Classifier	RGB to Grayscale, Edge detection, Morphological operation, watersheld	500 MRI	72%
[40]	CNN Classifier	histogram equalization technique, Gaussian filter	3064 MRI	93%
[46]	CNN Classifier	Resize, Augmentation, Grayscale, regularization techniques	3064 MRI, 516 MRI	96%, 98%
[32]	DNN	Fuzzy C-means, DWT, PCA	66 MRI	97%
[16]	CNN Classifier	Resize, Augmentation	3064 MRI	97%
[44]	CNN Classifier	MidResBlock	3064 MRI	96%
[10]	DNN Classifier	Resize, Crop Lesion, Uncropped Lesion, segment Lesion	3064 MRI	98%
[47]	CNN Classifier	MidResBlock	3064 MRI	94%
[13]	DNN	Resize, CNN, Segmentation	BraTS 2012, BraTS 2013, BraTS 2014, BraTS 2015, ISLES 2016, ISLES 2017	98%, 99%, 100%, 93%, 95%, 98%
[48]	Ensemble of ViTs	optimization of transformer parameters	3064 MRI	98.7%
[9]	Hybrid transformer enhanced convolutional neural network (TECNN)	CNN, Attention mechanism	BraTS 2018, Figshare datasets	96.75%, 99.1%

co-occurrence matrices (GLCM), dilation, and various filters provide specialized benefits in this regard.

- Classification: The selection of parameters in this phase has a profound impact on the effectiveness of the approach. Careful and informed parameter choices are essential to maximize performance and achieve optimal results.

Figure 5 presents a graphical representation of the highest accuracy rates achieved for brain tumor classification across different datasets using deep learning methods, particularly Convolutional Neural Networks. The remarkable performance observed can be attributed to the effectiveness of CNNs in automatically extracting relevant features

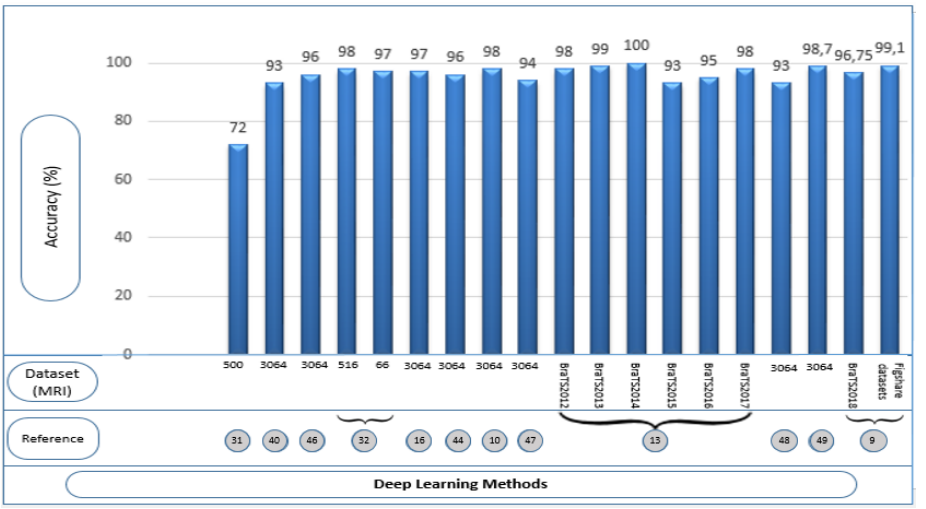


Fig. 5. The classification performance achieved in various brain tumor studies utilizing deep learning techniques across different datasets. Labels given in the row "Reference" are related to literature references according to Tab. 2, p. 37.

from medical images. Furthermore, preprocessing techniques such as image normalization, augmentation, and noise reduction have played a key role in enhancing the quality of input data, ultimately improving model accuracy. The combination of well-structured preprocessing pipelines and robust feature extraction capabilities of CNNs has significantly contributed to achieving high classification performance, demonstrating the potential of deep learning in brain tumor diagnosis.

Despite the considerable advancements brought by deep learning in medical image analysis, several limitations continue to hinder its full potential in clinical applications. Deep learning models, especially convolutional neural networks, demand extensive computational power and access to large, well-annotated datasets to achieve high performance. In practice, such datasets are often scarce, particularly in specialized medical domains like brain tumor diagnosis. Furthermore, these models are prone to overfitting, especially when trained on limited data, and their "black-box" nature makes their decision processes difficult to interpret. Additionally, deep learning algorithms may struggle to generalize effectively when applied across different clinical settings or imaging devices. To address these issues, recent research has explored hybrid approaches that integrate the strengths of both traditional machine learning and deep learning techniques. These combined frameworks often use deep learning for automated feature extraction, followed by classical ML algorithms—such as SVM or Random Forest—for final classification. This strategy not only reduces dependency on large labeled datasets but also enhances

model interpretability and robustness. By leveraging the complementary advantages of both paradigms, these integrated systems aim to improve diagnostic accuracy and reliability in complex imaging tasks.

3.3. ML and CNN

Recently, numerous approaches have employed convolutional neural networks in combination with machine learning algorithms to enhance classification performance. This research focuses on integrating CNN techniques with various machine learning algorithms to optimize performance in image classification tasks. By harnessing the feature extraction capabilities of CNNs alongside the adaptability of machine learning algorithms for classification, these approaches aim to achieve significant improvements in classification accuracy. This integration contributes to advancements in computer vision and pattern recognition, paving the way for more effective solutions in the field. Table 4 presents a comparison of studies that utilize different machine learning models and deep learning architectures, various feature extraction techniques, and diverse datasets to predict the classification accuracy of brain tumors.

Based on the results presented in Tab. 4, we observe the significant advancements in CNN techniques and machine learning algorithms for extracting intricate features from complex datasets, particularly in the field of image classification. By harnessing the

Tab. 4. Comparison of machine learning models and deep learning architectures, Feature Extraction Methods, and Datasets for Brain Tumor Classification Accuracy.

Ref.	Classification Method	Feature Extraction	Dataset	Accuracy
[35]	SVM, DNN	Fuzzy C-Means (FCM), CNN	BraTS 2015	97%
[20]	SVM, KNN, transfer learned, deep network	GoogLeNet, CNN	3064 MRI	97%, 98%, 92%
[34]	artificial neural network, Parzen window, k-Nearest Neighbors	Wavelets, PCA	166 MRI	98%, 99%, 99%
[53]	Machine Learning Methods Classifier, VGG16	Resize, Augmentation, Crop, Transfer Lerning	253 MRI	88%, 98%
[17]	SVM, Decision Tree, Random Forest, CNN, ResNet 50, AlexNet, Google Lenet, hybrid DCNN-LUNET	Resize, Laplace Gaussian (LOG) filtering and contrast-limited adaptive histogram smoothing, VGG-16, ROI Segmentation, FCM-GMM	260 MRI	97%, 96%, 97%, 98.82%

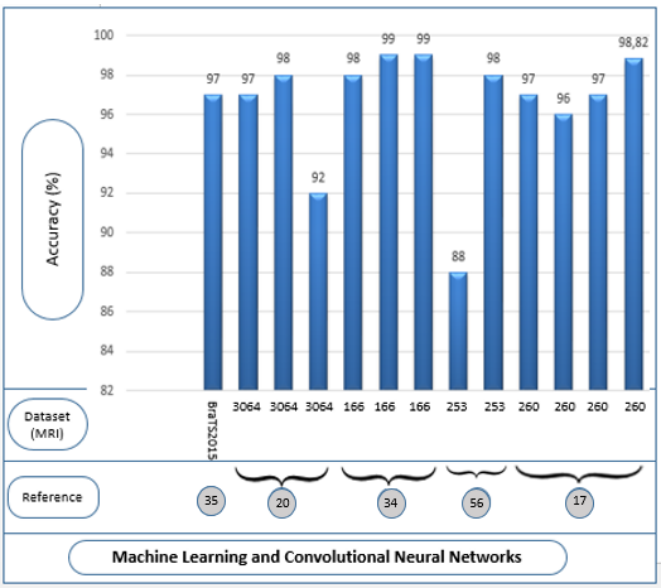


Fig. 6. The classification outcomes reported in several brain tumor studies that employed Machine and Deep Learning approaches on diverse datasets. Labels given in the row "Reference" are related to literature references according to Tab. 2, p. 37.

hierarchical feature extraction capabilities of CNNs alongside the discriminative power of machine learning algorithms, these approaches strive to substantially enhance classification performance. This integration aims to achieve higher accuracy and robustness in classifying diverse image datasets, thereby contributing to progress in computer vision and pattern recognition research.

Figure 6 illustrates the highest accuracy rates achieved for brain tumor classification across various datasets using both traditional machine learning techniques and Convolutional Neural Networks. The superior performance is largely influenced by the effectiveness of feature extraction methods, which play a crucial role in distinguishing tumor types. Preprocessing steps, including contrast enhancement, noise reduction, and data augmentation, further refine the input images, ensuring better model generalization. The combination of handcrafted feature extraction in machine learning and automatic feature learning in CNNs has led to significant improvements in classification accuracy, highlighting the importance of data quality and of the preprocessing steps in achieving optimal results.

Traditional machine learning techniques face notable limitations, particularly in the

context of complex medical imaging tasks such as brain tumor classification. These methods often depend on handcrafted feature extraction, which requires substantial domain knowledge and may overlook critical spatial or contextual information embedded in the images. Although deep learning has emerged as a powerful alternative—capable of learning hierarchical features directly from raw data—it also presents significant challenges. These include the necessity for large annotated datasets, high computational requirements, risk of overfitting, limited transparency in decision-making, and reduced adaptability across heterogeneous clinical settings. In light of these issues, Capsule Networks have been proposed as a promising new approach. Unlike conventional CNNs, Capsule Networks are designed to preserve spatial hierarchies and relationships between features, making them more robust to affine transformations and better suited for modeling complex structures in medical images. Moreover, their architecture allows for enhanced interpretability and potentially better generalization from smaller datasets, offering a compelling direction for overcoming some of the critical shortcomings observed in both traditional ML and standard deep learning models.

3.4. Capsule network architectures

While convolutional neural networks have been extensively utilized for feature extraction in image processing tasks, they exhibit limitations in capturing spatial relationships among features. Capsule Networks address this limitation by preserving the spatial hierarchy of features more effectively. CapsNets introduce the concept of capsules, which encapsulate spatial information more efficiently than traditional CNNs. Furthermore, CapsNets offer significant advantages, including improved generalization, robustness to affine transformations, and enhanced interpretability. These qualities make them a compelling alternative for tasks requiring accurate spatial feature extraction and classification in medical imaging. The table below provides a detailed overview of various methodologies that employ capsule networks for brain tumor classification. Figure 7 illustrates the standard pipeline employed in brain tumor segmentation approaches utilizing Capsule Networks (CapsNet).

Table 5 presents a comparison of studies that utilize capsules networks architectures, various feature extraction techniques, and diverse datasets to predict the classification accuracy of brain tumors.

Currently, much research in classification highlights the limitations of traditional CNNs in effectively extracting spatial features, largely due to their reliance on pooling operations, which can result in the loss of critical spatial information. To overcome these challenges, recent studies have explored the use of capsule networks as a promising alternative. Capsule networks are specifically designed to capture hierarchical spatial relationships within images more effectively than CNNs, potentially improving feature extraction and classification accuracy. Additionally, capsule networks provide several advantages, including better handling of spatial hierarchies, increased robustness

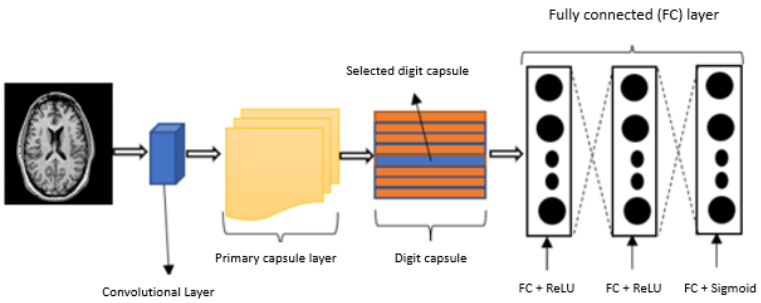


Fig. 7. Illustration of a typical segmentation workflow leveraging Capsule Networks.

Tab. 5. Comparison of capsules networks architectures, Feature Extraction Methods, and Datasets for Brain Tumor Classification Accuracy.

Ref.	Feature Extraction	Dataset	Accuracy
[6]	Hyperparameter optimization	3064 MRI	90%
[7]	T-distributed Stochastic Neighbor Embedding (TSNE)	3064 MRI	86%
[8]	Boosting approach	3064 MRI	92%
[49]	Rotation and patch extraction	3064 MRI	94%
[4]	activation function	3264 MRI	96.7%
[5]	CapsNet, dilation convolution	3064 MRI	95.54%
[15]	SegCaps–Capsule network, brain tumor segmentation	BraTS 2020	87.96%

to affine transformations, and enhanced interpretability of learned features. This innovative approach addresses the shortcomings of CNNs in spatial feature extraction, offering significant advancements in image classification for medical applications.

The strong performance of these models can be attributed to their ability to capture spatial hierarchies and maintain spatial relationships between features, unlike traditional CNNs. The effectiveness of the model is further enhanced by preprocessing techniques such as normalization, noise reduction, and data augmentation, which improve the quality of input data. Additionally, robust feature extraction methods contribute to the model's capacity to distinguish complex patterns within brain tumor images, ultimately leading to superior classification accuracy. The chart in Fig. 8 illustrates the classification outcomes reported in several brain tumor studies that employed Capsule Network approaches on diverse datasets.

While Capsule Networks have demonstrated significant potential in preserving spatial hierarchies and improving robustness to affine transformations, they still face several practical limitations that hinder their widespread adoption in medical imaging tasks.

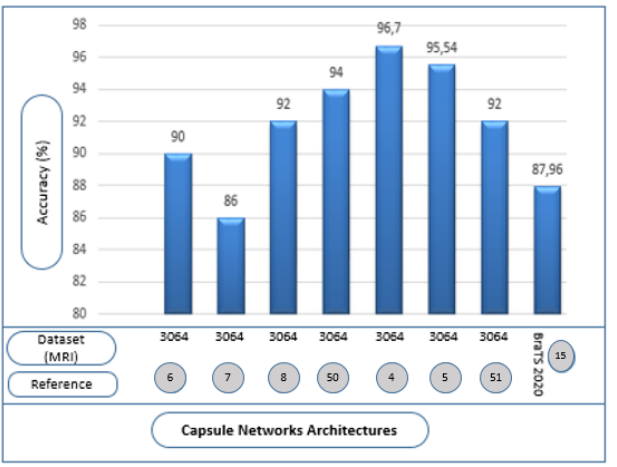


Fig. 8. The classification performance achieved in various brain tumor studies utilizing capsule networks techniques across different datasets. Labels given in the row "Reference" are related to literature references according to Tab. 2, p. 37.

One of the main challenges lies in their computational inefficiency; the dynamic routing mechanism, which is central to Capsule Networks, is resource-intensive and leads to slower training and inference times. Additionally, these networks are relatively sensitive to hyperparameter tuning and lack standardized architectures, making their implementation and optimization more complex compared to traditional deep learning models. In response to these shortcomings, Vision Transformers have emerged as a compelling alternative. Unlike Capsule Networks, ViTs leverage self-attention mechanisms to model global dependencies within an image, allowing for more efficient capture of contextual information across the entire visual field. Moreover, Vision Transformers demonstrate greater scalability and adaptability, showing strong performance even when trained on relatively limited data through techniques such as transfer learning and data augmentation. As research in this area progresses, ViTs are increasingly being considered as a powerful tool for medical image classification and segmentation, potentially overcoming the architectural and computational limitations associated with Capsule Networks.

3.5. Vision Transformers

Recent advances in image classification have drawn attention to the inherent limitations of conventional Convolutional Neural Networks, particularly in capturing long-range dependencies and global contextual information within medical images. These limitations stem mainly from the localized nature of convolution operations and the use of pooling layers, which can lead to the loss of important spatial relationships. To address these

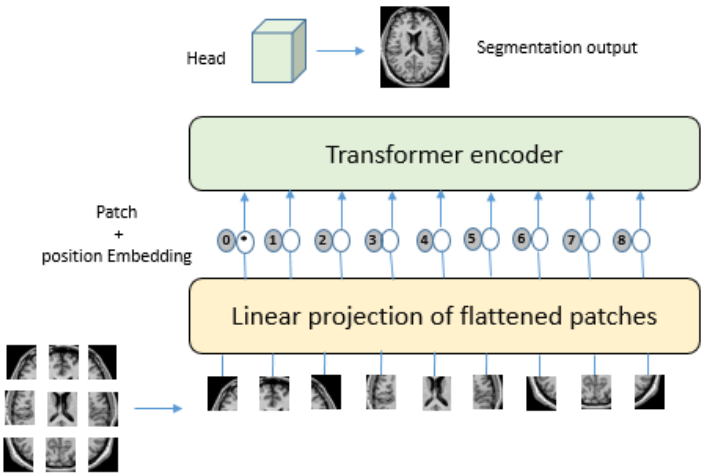


Fig. 9. Overview of the Vision Transformers model.

issues, researchers have increasingly explored Vision Transformers as a powerful alternative. Unlike CNNs, Vision Transformers leverage self-attention mechanisms to model global interactions across the entire image, allowing for more comprehensive and context-aware feature representation. This enables ViTs to retain critical spatial and semantic details, enhancing classification performance. Furthermore, ViTs offer advantages such as scalability, better generalization in complex datasets, and improved interpretability due to their attention maps, which highlight key regions influencing decision-making. This modern architecture represents a promising direction for improving image classification in brain tumor analysis and other medical imaging tasks.

The high performance of these models can be credited to their ability to analyze images holistically, maintaining spatial coherence while focusing on the most informative regions through self-attention. Unlike CNNs, which process image patches locally, ViTs treat the entire image as a sequence of patches, enabling the network to recognize complex global patterns that are essential in medical image analysis. This performance is further strengthened by preprocessing strategies such as image normalization, denoising, and data augmentation, which enhance input consistency and variability. Additionally, the integration of advanced feature extraction pipelines allows the model to effectively distinguish between subtle differences in tumor structures, leading to highly accurate and reliable classification outcomes. These capabilities make Vision Transformers a compelling choice for future developments in AI-assisted medical diagnostics. Figure 9 illustrates the standard pipeline employed in brain tumor segmentation approaches utilizing vision transformers.

Table 6 presents a comparison of studies that utilize vision transformers architectures, various feature extraction techniques, and diverse datasets to predict the classification accuracy of brain tumors.

Based on the analysis shown in Tab. 6, it becomes clear that various components contribute significantly to improving the performance of brain tumor classification systems. Each method integrates specific techniques aligned with its core objective, progressing through several essential stages to achieve optimal accuracy. The data preprocessing phase remains fundamental in Vision Transformer-based workflows, as it prepares the input for optimal attention-based modeling. Techniques such as normalization, image denoising, patch embedding, resizing, and data augmentation are critical in ensuring consistency, reducing artifacts, and enhancing generalization. These operations help the model interpret input images more effectively during training and inference.

The feature representation and encoding stage is particularly crucial in Vision Transformers. Instead of relying on handcrafted features or convolutional layers, ViTs divide images into fixed-size patches and transform them into sequences of embeddings, which are processed through self-attention layers. This enables the model to capture both local and global dependencies across the entire image, significantly enriching the representation of complex patterns in brain tumor regions. Additionally, position embeddings are integrated to retain spatial information, further improving interpretability.

Finally, during the classification stage, the transformer encoder's output is used to make predictions through fully connected layers. The effectiveness of this stage is influenced by the architecture's depth, the number of attention heads, and the choice of loss functions and optimization strategies. The Figure 10 highlights the top classification accuracies achieved across multiple datasets using Vision Transformer-based models. These impressive results are largely attributed to the robust preprocessing procedures and the ViTs' superior ability to model long-range spatial relationships. The evaluation reaffirms the importance of designing effective preprocessing workflows and utilizing advanced attention mechanisms to optimize classification performance in brain tumor diagnostics.

Tab. 6. Comparison of Vision Transformers model, Feature Extraction Methods, and Datasets for Brain Tumor Classification Accuracy.

Reference	Feature Extraction	Dataset	Accuracy
[50]	Transformers and 3D CNN	BraTS 2019, BraTS 2020	90.09%
[24]	Swin transformers and CNN	BraTS 2021	93.3%
[25]	Transformers and CNN	MSD dataset	78.9%
[26]	Transformers and 3D CNN	BraTS 2021	90.8%
[36]	Transformers and 3D CNN "U-Net shaped encoder-decoder"	BraTS 2021	91.2%

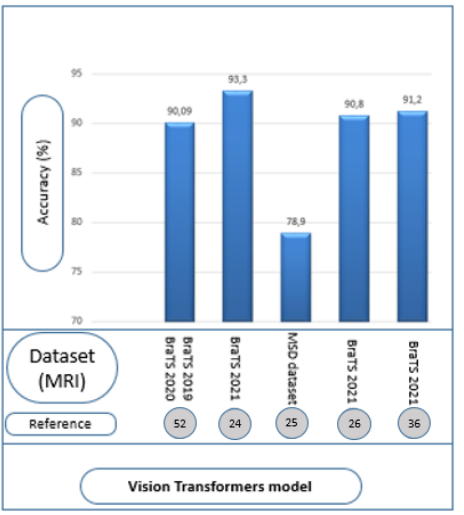


Fig. 10. The classification accuracy of each study on brain tumors, based on vision transformers and the corresponding datasets used. Labels given in the row "Reference" are related to literature references according to Tab. 2, p. 37.

The figure below presents the classification results obtained from multiple brain tumor studies that adopted Vision Transformer-based methods across various datasets.

Although Vision Transformers have gained traction for their ability to model long-range dependencies and capture global image context more effectively than traditional convolutional approaches, they are not without limitations. One of the primary challenges associated with ViTs is their need for extensive training data to perform optimally, which can be a significant constraint in the medical imaging field where labeled datasets are often limited. Additionally, their architecture tends to be computationally demanding, both in terms of memory usage and training time, which can limit their accessibility in resource-constrained clinical environments. ViTs also exhibit sensitivity to hyperparameter selection and are often less interpretable compared to some traditional machine learning models. These constraints have sparked a wave of innovation among researchers who are actively exploring novel hybrid models, architectural optimizations, and lightweight transformer variants tailored to medical contexts. The current trend involves designing more efficient classification algorithms that combine the strengths of ViTs with other paradigms, such as convolutional modules or attention-enhanced ML models, to achieve better accuracy, generalizability, and scalability. This competitive research environment is fostering the development of next-generation models that aim to balance performance, efficiency, and interpretability for robust brain tumor classification and other critical diagnostic tasks.

3.6. Discussion

In conclusion, the classification of brain tumors using Machine Learning, Deep Learning, Capsule Network architectures, and Vision Transformers model has demonstrated significant advancements in accuracy and robustness. DL approaches, particularly Convolutional Neural Networks, have surpassed ML techniques by automatically learning hierarchical features, improving generalization. More recently, Capsule Networks have further enhanced classification performance by preserving spatial relationships between features, addressing limitations of CNNs in detecting complex structures. The effectiveness of these models is strongly influenced by preprocessing techniques such as normalization, noise reduction, and data augmentation, which enhance input quality. Additionally, feature extraction methods play a crucial role in identifying relevant tumor characteristics, leading to improved classification accuracy. The integration of advanced architectures with optimized preprocessing and feature extraction strategies paves the way for more reliable and precise brain tumor diagnosis, contributing to enhanced decision-making in medical imaging.

A critical challenge in deploying ML, DL, Capsnet and Vit models for brain tumor analysis lies in their limited ability to generalize across diverse clinical settings. Variations in MRI acquisition protocols, scanner types, and patient populations often lead to distributional shifts that can significantly impact model performance. Models trained on a specific dataset may not perform reliably when applied to external data due to differences in resolution, contrast, noise levels, and anatomical variability. Addressing this issue requires the integration of domain adaptation techniques, robust data augmentation, and cross-institutional validation to ensure that AI models remain accurate, consistent, and clinically applicable across a wide range of imaging environments.

4. Conclusion and future scope

In this review, we provided an in-depth examination of recent advances in brain tumor classification and segmentation, focusing on notable research studies that implement a variety of machine learning, deep learning, Capsule Networks, and Vision Transformers techniques. These studies have contributed significantly to the improvement of classification performance through enhanced feature extraction, preprocessing, and the careful selection of classification algorithms. The analysis underscores the importance of each stage in the diagnostic pipeline—from data preparation through normalization and augmentation, to robust feature extraction using methods like CNNs, Gabor filters, DWT, LBP, and GLCM, and finally to accurate classification through optimized models.

While the reviewed models demonstrate impressive performance, this study also acknowledges key limitations that remain a challenge in clinical applications. For instance, traditional ML approaches rely heavily on handcrafted features, which often limit their

performance in complex imaging contexts. DL models, although more effective in learning features automatically, face challenges such as high computational demands, the need for large annotated datasets, interpretability issues, and limited generalizability across diverse clinical environments.

To address these challenges, emerging research is exploring hybrid models that combine ML and DL to leverage the strengths of both paradigms. Additionally, recent developments in Capsule Networks and Vision Transformers present promising alternatives by offering improved spatial awareness and better feature representation. However, these models also face issues such as high training complexity, stability concerns, and a lack of standardized benchmarks.

This area is in the urgent need for models that generalize well across different MRI acquisition protocols and scanner types, as well as the development of computationally efficient architectures suitable for real-time clinical deployment. Furthermore, advancing techniques such as transfer learning, semi-supervised learning, and explainable AI are critical to overcoming current limitations.

Finally, while our review primarily focuses on brain tumor classification, the discussed techniques have broader applications, including the diagnosis of other neurological diseases such as Alzheimer's and Parkinson's. As the field evolves, our future research aims to develop versatile, interpretable, and clinically adaptable AI tools to support early and accurate diagnosis across a wide range of brain pathologies.

References

[1] S. Abbasi and F. Tajeripour. Detection of brain tumor in 3D MRI images using local binary patterns and histogram orientation gradient. *Neurocomputing* 219:526–535, 2017. doi:10.1016/j.neucom.2016.09.051.

[2] I. Aboussaleh, J. Riffi, K. el Fazazy, A. M. Mahraz, and H. Tairi. 3DUV-NetR+: A 3D hybrid semantic architecture using transformers for brain tumor segmentation with multimodal MR images. *Results in Engineering* 21:101892, 2024. doi:10.1016/j.rineng.2024.101892.

[3] I. Aboussaleh, J. Riffi, K. E. Fazazy, A. M. Mahraz, and H. Tairi. STCPU-Net: advanced U-shaped deep learning architecture based on Swin transformers and capsule neural network for brain tumor segmentation. *Neural Computing and Applications* 36(30):18549–18565, 2024. doi:10.1007/s00521-024-10144-y.

[4] K. Adu, Y. Yu, J. Cai, I. Asare, and J. Quahin. The influence of the activation function in a capsule network for brain tumor type classification. *International Journal of Imaging Systems and Technology* 32(1):123–143, 2022. doi:10.1002/ima.22638.

[5] K. Adu, Y. Yu, J. Cai, and N. Tashi. Dilated capsule network for brain tumor type classification via MRI segmented tumor region. In: *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 942–947. IEEE, 2019. doi:10.1109/ROBIO49542.2019.8961610.

[6] P. Afshar, A. Mohammadi, and K. N. Plataniotis. BayesCap: A Bayesian approach to brain tumor classification using capsule networks. *IEEE Signal Processing Letters* 27:2024–2028, 2020. doi:10.1109/LSP.2020.3034858.

[7] P. Afshar, K. N. Plataniotis, and A. Mohammadi. Capsule networks for brain tumor classification based on MRI images and coarse tumor boundaries. In: *ICASSP 2019 – 2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 1368–1372. IEEE, 2019. doi:10.1109/ICASSP.2019.8683759.

[8] P. Afshar, K. N. Plataniotis, and A. Mohammadi. BoostCaps: a boosted capsule network for brain tumor classification. In: *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 1075–1079. IEEE, 2020. doi:10.1109/EMBC44109.2020.9175922.

[9] M. Aloraini, A. Khan, S. Aladhadh, S. Habib, M. F. Alsharekh, et al. Combining the transformer and convolution for effective brain tumor classification using MRI images. *Applied Sciences* 13(6):3680, 2023. doi:10.3390/app13063680.

[10] A. M. Alqudah, H. Alquraan, I. A. Qasmieh, A. Alqudah, and W. Al-Sharu. Brain tumor classification using deep learning technique – A comparison between cropped, uncropped, and segmented lesion images with different sizes. arXiv, arXiv:2001.08844, 2020. doi:10.48550/arXiv.2001.08844.

[11] J. Amin, M. Sharif, M. Raza, T. Saba, and M. A. Anjum. Brain tumor detection using statistical and machine learning method. *Computer Methods and Programs in Biomedicine* 177:69–79, 2019. doi:10.1016/j.cmpb.2019.05.015.

[12] J. Amin, M. Sharif, M. Raza, and M. Yasmin. Detection of brain tumor based on features fusion and machine learning. *Journal of Ambient Intelligence and Humanized Computing* 15:983–999, 2024. doi:10.1007/s12652-018-1092-9.

[13] J. Amin, M. Sharif, M. Yasmin, and S. L. Fernandes. Big data analysis for brain tumor detection: Deep convolutional neural networks. *Future Generation Computer Systems* 87:290–297, 2018. doi:10.1016/j.future.2018.04.065.

[14] A. Ari and D. Hanbay. Deep learning based brain tumor classification and detection system. *Turkish Journal of Electrical Engineering and Computer Sciences* 26(5):2275–2286, 2018. doi:10.3906/elk-1801-8.

[15] M. J. Aziz, A. A. T. Zade, P. Farnia, M. Alimohamadi, B. Makkiabadi, et al. Accurate automatic glioma segmentation in brain MRI images based on CapsNet. In: *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 3882–3885. IEEE, 2021. doi:10.1109/EMBC46164.2021.9630324.

[16] M. M. Badža and M. Č. Barjaktarović. Classification of brain tumors from MRI images using a convolutional neural network. *Applied Sciences* 10(6):1999, 2020. doi:10.3390/app10061999.

[17] T. Balamurugan and E. Gnanamanoharan. Brain tumor segmentation and classification using hybrid deep CNN with LuNetClassifier. *Neural Computing and Applications* 35:4739–4753, 2023. doi:10.1007/s00521-022-07934-7.

[18] J. Chen and M. J. Berry. Selenium and selenoproteins in the brain and brain diseases. *Journal of Neurochemistry* 86(1):1–12, 2003. doi:10.1046/j.1471-4159.2003.01854.x.

[19] S. Deepak and P. M. Ameer. Brain tumor classification using deep CNN features via transfer learning. *Computers in Biology and Medicine* 111:103345, 2019. doi:10.1016/j.compbiomed.2019.103345.

[20] S. Deepak and P. M. Ameer. Brain tumor categorization from imbalanced MRI dataset using weighted loss and deep feature fusion. *Neurocomputing* 520:94–102, 2023. doi:10.1016/j.neucom.2022.11.039.

[21] D. N. George, H. B. Jehlol, and A. S. A. Oleiwi. Brain tumor detection using shape features and machine learning algorithms. *International Journal of Scientific and Engineering Research* 6(12):454–459, 2015.

[22] X. Han and Y. Li. The application of convolution neural networks in handwritten numeral recognition. *International Journal of Database Theory and Application* 8(3):367–376, 2015. doi:10.14257/ijdta.2015.8.3.32.

[23] A. M. Hasan, H. A. Jalab, R. W. Ibrahim, F. Meziane, A. R. AL-Shamasneh, et al. MRI brain classification using the quantum entropy LBP and deep-learning-based features. *Entropy* 22(9):1033, 2020. doi:10.3390/e22091033.

[24] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, et al. Swin UNETR: Swin transformers for semantic segmentation of brain tumors in MRI images. In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. 7th International Workshop, BrainLes 2021. Held in Conjunction with MICCAI 2021*, vol. 12962 of *Lecture Notes in Computer Science*, pp. 272–284. Springer, 2021. doi:10.1007/978-3-031-08999-2_22.

[25] A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, et al. UNETR: Transformers for 3D medical image segmentation. In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1748–1758, 2022. doi:10.1109/WACV51458.2022.00181.

[26] Q. Jia and H. Shu. BiTr-Unet: A CNN-transformer combined network for MRI brain tumor segmentation. In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. 7th International Workshop, BrainLes 2021. Held in Conjunction with MICCAI 2021*, vol. 12963 of *Lecture Notes in Computer Science*, pp. 3–14. Springer, 2021. doi:10.1007/978-3-031-09002-8_1.

[27] J. Kang, Z. Ullah, and J. Gwak. MRI-based brain tumor classification using ensemble of deep features and machine learning classifiers. *Sensors* 21(6):2222, 2021. doi:10.3390/s21062222.

[28] K. Kaplan, Y. Kaya, M. Kuncan, and H. M. Ertunç. Brain tumor classification using modified local binary patterns (LBP) feature extraction methods. *Medical Hypotheses* 139:109696, 2020. doi:10.1016/j.mehy.2020.109696.

[29] H. Kutlu and E. Avcı. A novel method for classifying liver and brain tumors using convolutional neural networks, discrete wavelet transform and long short-term memory networks. *Sensors* 19(9):1992, 2019. doi:10.3390/s19091992.

[30] K. Maharana, S. Mondal, and B. Nemade. A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings* 3(1):91–99, 2022.

[31] H. C. Megha. Evaluation of brain tumor mri imaging test detection and classification. *International Journal for Research in Applied Science and Engineering Technology* 8(6):124–131, 2020. doi:10.22214/ijraset.2020.6019.

[32] H. Mohsen, E.-S. A. El-Dahshan, E.-S. M. El-Horbaty, and A.-B. M. Salem. Classification using deep learning neural networks for brain tumors. *Future Computing and Informatics Journal* 3(1):68–71, 2018. doi:10.1016/j.fcij.2017.12.001.

[33] N. Nabizadeh and M. Kubat. Brain tumors detection and segmentation in MR images: Gabor wavelet vs. statistical features. *Computers & Electrical Engineering* 45:286–301, 2015. doi:10.1016/j.compeleceng.2015.02.007.

[34] S. Najafi, M. C. Amirani, and Z. Sedghi. A new approach to mri brain images classification. In: *2011 19th Iranian Conference on Electrical Engineering*, pp. 1–5. IEEE, 2011.

[35] K. Pathak, M. Pavthawala, N. Patel, D. Malek, V. Shah, et al. Classification of brain tumor using convolutional neural network. In: *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 128–132. IEEE, 2019. doi:10.1109/ICECA.2019.8821931.

[36] H. Peiris, M. Hayat, Z. Chen, G. Egan, and M. Harandi. A robust volumetric transformer for accurate 3D tumor segmentation. In: *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2022)*, vol. 13435 of *Lecture Notes in Computer Science*, pp. 162–172. Springer, 2022. doi:10.1007/978-3-031-16443-9_16.

[37] H. M. Rai and K. Chatterjee. Detection of brain abnormality by a novel Lu-Net deep neural CNN model from MR images. *Machine Learning with Applications* 2:100004, 2020. doi:10.1016/j.mlwa.2020.100004.

[38] P. G. Rajan and C. Sundar. Brain tumor detection and segmentation by intensity adjustment. *Journal of Medical Systems* 43(8):282, 2019. doi:10.1007/s10916-019-1368-4.

[39] A. Rehman, S. Naz, M. I. Razzak, F. Akram, and M. Imran. A deep learning-based framework for automatic brain tumors classification using transfer learning. *Circuits, Systems, and Signal Processing* 39(2):757–775, 2020. doi:10.1007/s00034-019-01246-3.

[40] J. Seetha and S. S. Raja. Brain tumor classification using convolutional neural networks. *Biomedical & Pharmacology Journal* 11(3):1457–1461, 2018. doi:10.13005/bpj/1511.

[41] J. L. Semmlow. *Biosignal and medical image processing*. CRC press, Boca Raton, 2008. doi:10.1201/9780203024058.

[42] K. Sharma, A. Kaur, and S. Gujral. Brain tumor detection based on machine learning algorithms. *International Journal of Computer Applications* 103(1):7–11, 2014. doi:10.5120/18036-6883.

[43] S. Shrot, M. Salhov, N. Dvorski, E. Konen, A. Averbuch, et al. Application of MR morphologic, diffusion tensor, and perfusion imaging in the classification of brain tumors using machine learning scheme. *Neuroradiology* 61:757–765, 2019. doi:10.1007/s00234-019-02195-z.

[44] Z. Sobhaninia, N. Karimi, P. Khadivi, R. Roshandel, and S. Samavi. Brain tumor classification using medial residual encoder layers. arXiv, arXiv:2011.00628, 2020. doi:10.48550/arXiv.2011.00628.

[45] D. Stamate, R. Smith, R. Tsygancov, R. Vorobev, J. Langham, et al. Applying deep learning to predicting dementia and mild cognitive impairment. In: *Artificial Intelligence Applications and Innovations: Proceedings of the 16th IFIP WG 12.5 International Conference (AIAI 2020), Part II*, vol. 584 of *IFIP Advances in Information and Communication Technology*, pp. 308–319. Springer, 2020. doi:10.1007/978-3-030-49186-4_26.

[46] H. H. Sultan, N. M. Salem, and W. Al-Atabany. Multi-classification of brain tumor images using deep neural network. *IEEE Access* 7:69215–69225, 2019. doi:10.1109/ACCESS.2019.2919122.

[47] Z. N. K. Swati, Q. Zhao, M. Kabir, F. Ali, Z. Ali, et al. Brain tumor classification for mr images using transfer learning and fine-tuning. *Computerized Medical Imaging and Graphics* 75:34–46, 2019. doi:10.1016/j.compmedimag.2019.05.001.

[48] S. Tummala, S. Kadry, S. A. C. Bukhari, and H. T. Rauf. Classification of brain tumor from magnetic resonance imaging using vision transformers ensembling. *Current Oncology* 29(10):7498–7511, 2022. doi:10.3390/curroncol29100590.

[49] R. Vimal Kurup, V. Sowmya, and K. P. Soman. Effect of data pre-processing on brain tumor classification using Capsulenet. In: *Proceedings of ICICCT 2019 – System Reliability, Quality Control, Safety, Maintenance and Management*, pp. 110–119. Springer, 2020. doi:10.1007/978-981-13-8461-5_13.

[50] W. Wang, C. Chen, M. Ding, H. Yu, S. Zha, et al. TransBTS: Multimodal brain tumor segmentation using transformer. In: *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2021)*, vol. 12901 of *Lecture Notes in Computer Science*, pp. 109–119. Springer, 2021. doi:10.1007/978-3-030-87193-2_11.

[51] I. Zine-dine, A. Fahfouh, J. Riffi, K. El Fazazy, I. El Batteoui, et al. Brain tumor classification using feature extraction and ensemble learning. *Machine Graphics and Vision* 33(3/4):3–28, 2024. doi:10.22630/MGV.2024.33.3.1.

[52] I. Zine-dine, J. Riffi, K. El Fazazy, I. El Batteoui, M. A. Mahraz, et al. A hybrid model for Alzheimer's disease classification based on neural network architectures enhanced

by GAN model. *International Journal of Online and Biomedical Engineering* 21(8):23, 2025. doi:10.3991/ijoe.v21i08.54363.

[53] I. Zine-dine, J. Riffi, K. El Fazazy, M. A. Mahraz, and H. Tairi. Brain tumor classification using machine and transfer learning. In: *Proceedings of the 2nd International Conference on Big Data, Modelling and Machine Learning (BML)*, vol. 1, pp. 566–571, 2021. doi:10.5220/0010762800003101.

[54] I. Zine-dine, J. Riffi, K. El Fazziki, M. Mohamed Adnane, and T. Hamid. Alzheimer's disease classification using histogram of oriented gradient, transfer learning, and capsules network. *International Journal of Intelligent Systems and Applications in Engineering* 12(4):5335–5350, Jun 2025. https://ijisae.org/index.php/IJISAE/article/view/7325.

[55] E. de la Rosa, J. Kirschke, B. Wiestler, B. Menze, M. Reyes, et al. ISLES Challenge 2022. Ischemic Stroke Lesion Segmentation, 2022. https://www.isles-challenge.org/.

[56] S. Bakas, U. Baid, C. Carr, E. Calabrese, E. Colak, et al. RSNA-ASNR-MICCAI Brain Tumor Segmentation (BraTS) Challenge 2021, 2021. http://braintumorsegmentation.org/.

[57] Figshare LLP. fig**share**. A Digital Science Solution. https://figshare.com.

[58] Alphabet. Google Scholar. https://scholar.google.co.in.

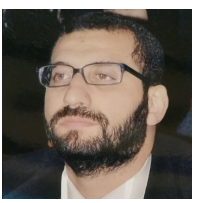
[59] Elsevier. ScienceDirect. https://www.sciencedirect.com.

Iliass Zine-dine: PhD in Computer Science at the Laboratory of Computer Science, Signals, Automation, and Cognitivism (LISAC), Faculty of Science, Dhar El Mahraz. Sidi Mohamed Ben Abdellah University (U.S.M.B.A), Fez, Morocco.

Jamal Riffi: Doctor of Computer Science at the Laboratory of Computer Science, Signals, Automation, and Cognitivism (LISAC), Faculty of Science, Dhar El Mahraz. Sidi Mohamed Ben Abdellah University (U.S.M.B.A), Fez, Morocco.

Khalid El Fazazy: Doctor of Computer Science at the Laboratory of Computer Science, Signals, Automation, and Cognitivism (LISAC), Faculty of Science, Dhar El Mahraz. Sidi Mohamed Ben Abdellah University (U.S.M.B.A), Fez, Morocco.

Ismail El Batteoui: Doctor of Computer Science at the Laboratory of Computer Science, Signals, Automation, and Cognitivism (LISAC), Faculty of Science, Dhar El Mahraz. Sidi Mohamed Ben Abdellah University (U.S.M.B.A), Fez, Morocco.



Mohamed Adnane Mahraz: Doctor of Computer Science at the Laboratory of Computer Science, Signals, Automation, and Cognitivism (LISAC), Faculty of Science, Dhar El Mahraz. Sidi Mohamed Ben Abdellah University (U.S.M.B.A), Fez, Morocco.

Hamid Tairi: Doctor of Computer Science at the Laboratory of Computer Science, Signals, Automation, and Cognitivism (LISAC), Faculty of Science, Dhar El Mahraz. Sidi Mohamed Ben Abdellah University (U.S.M.B.A), Fez, Morocco.

Advancing Plant Disease Detection: A Comparative Analysis of Deep Learning and Hybrid Machine Learning Models

Rajesh Kumar* and Vikram Singh
Department of Computer Science & Engineering, Chaudhary Devi Lal University, Sirsa, Haryana, India
*Corresponding author: Rajesh Kumar (rajesh31070@gmail.com)

Submitted: 21 Feb 2025 **Accepted:** 28 Apr 2025 **Published:** 24 Sep 2025

Abstract Detecting plant diseases is essential for precision agriculture, as it enhances crop production and ensures the security of the food supply. We adopted two methods for this research: a method based on deep learning, through Convolutional Neural Networks (CNN), and a hybrid model using classical machine learning. The dataset comprised images of plant leaves from Kirtan village in Hisar, Haryana, which were annotated by plant pathologists. The CNN model, which autonomously extracts hierarchical spatial features, achieved an accuracy of 97.57%, making it ideal for large datasets. Conversely, the Hybrid model utilizing handcrafted GLCM and LBP features and SVM classifiers achieved 91.73% accuracy while providing interpretability and computational efficiency in resource limited setups. The performance of the models was measured in terms of accuracy, precision, recall and F1-score. Applications range from on-line monitoring with drones to diagnostic equipment for the farmer.

Keywords: plant disease detection, hybrid model, machine learning, deep learning, GLCM, LBP, CNNs, SVMs, feature extraction, data augmentation, classification models, evaluation metrics, precision agriculture.

1. Introduction

Detecting plant diseases is essential in contemporary agriculture, with considerable implications for global food security, crop management, and environmental sustainability. Early and accurate recognition of plant diseases empowers farmers with the means to make respective treatments. Traditional detection methods reliant on manual inspection and expert knowledge are characterized by high labor intensity, significant time consumption, and susceptibility to human error.

Recent developments in machine learning and computer vision have helped to develop automated systems to diagnose plant diseases. Early methodologies utilized classical machine learning techniques that incorporated hand-crafted features, including the Grey-Level Co-occurrence Matrix (GLCM) and Local Binary Patterns (LBP), in conjunction with classifiers such as Support Vector Machines (SVM). These models exhibit interpretability and demonstrate strong performance on small datasets; however, they are constrained by their reliance on manual feature engineering and exhibit limited scalability.

The advent of deep learning, in particular Convolutional Neural Networks (CNNs), has transformed the area with the ability to learn features directly from the image data.

This leads to enhanced accuracy and scalability. CNNs require significant computational resources and extensive labeled datasets, rendering them less appropriate for low-resource agricultural settings.

This research compares between CNN and hybrid-based machine learning models for plant disease classification, which fills the gap between these two approaches in plant disease detection. We measure the performance of each model on a locally acquired and annotated dataset by applying accuracy, precision, recall, and F1-score, and assessing their respective applicability in on-site agricultural practices. The results are intended to aid in the design of practical and scalable disease detection solutions in precision agriculture. To contextualize these contributions, the following literature review examines prior studies on machine learning and deep learning approaches, highlighting their strengths, limitations, and areas where this research advances the field.

The remaining parts of this paper are as follows. After the literature review in Section 2, the proposed methodology for plant disease detection will be presented in Section 3. The results of classifications will be shown in Section 4. The comparison between the CNN and the hybrid-based machine learning models will be made in Section 5. Finally, the paper will be concluded in Section 6.

2. Literature Review

Machine learning-based (ML) techniques for plant disease classification have been explored in several studies. Iniyan et al. [8] used Support Vector Machines (SVM) and Artificial Neural Networks (ANN) for detecting plant diseases, showing the benefits of classical ML algorithms. Similarly, Saleem et al. [14] notably found that ML models lagged behind the deep learning-based classifiers in this regard, since it was a comparative analysis. Dixit et al. [5] presented an ML-based model for wheat crop disease detection, resulting in more efficient monitoring of agriculture.

One promising solution for plant disease detection is deep tissue learning. Bakr et al. [1] used transfer learning with DenseNet to classify tomato diseases and concluded with a high accuracy in tomato disease classification. Balafas et al. [2] presented a detailed comparison between ML and DL models; it was shown to achieve better results in CNN when applied to tasks in detecting plant diseases. Khalid and Karan [11] discussed the effectiveness of deep learning and proposed an end-to-end automatic disease classification model. Sharma and Guleria [16] explored and compared numerous deep learning models for image classification, indicating that CNNs performed remarkably well with regard to generalization across datasets. Yu [20] conducted a review of deep learning methods used for image-based classification tasks and confirmed their effectiveness.

To achieve higher classification accuracy, hybrid and attention-based models were explored. Shao et al. [15] developed a hybrid ViT-CNN model by combining Vision Transformers (ViT) and CNN to achieve fine-grained classification. Kalim et al. [10]

developed a hybrid CNN-RF model that combines the initial feature extraction and machine learning classifiers for the detection of diseases in citrus leaves. Bera et al. [3] introduced an attention-based deep network, where n features selection is optimized for plant disease classification. Gao et al. [7] propose a two-branch channel attention-based model, which improved the crop disease identification accuracy.

Comparative analyses of different classification models have been performed in some studies. Wang et al. [19] conducted a systematic comparison between conventional ML and DL methods in the context of image classification, thus further confirming the superiority of deep learning. Lorente et al. [12] referenced several classical and modern deep learning methods for classification of plant diseases. Deep learning methods for image classification have been extensively reviewed, with Wu et al. [13] providing a detailed summary of key advancements in CNN, ViT, and Hybrid models.

However, despite all this progress, there are still problems when it comes to real-world use. Shoaib et al. [17] provided an overview of recent advances in deep learning-based approaches for plant disease detection and underscored the need for better model generalization and larger datasets. To enhance the robustness of the model, data augmentation techniques are essential, as highlighted by Furqan et al. [6]. Kabir et al. [9] explored differences in Gray-Level Co-occurrence Matrix (GLCMs) and advised GLCM optimization to enhance feature extraction for deep learning models. Chen et al. [4] addressed the problems of real-world occlusions and illuminations while showing robustness via detection techniques. A survey on smart agriculture applications by combining ML and DL techniques for crop disease prediction was provided by Subbarayudu and Kubendiran [18].

A review of the literature (Table 1) shows a clear trajectory of moving from traditional machine learning approaches to recently used deep learning-based methods for detecting plant diseases. Although CNN is a popular architecture, hybrid and attention-based models have led to greater accuracy. Future research should focus on overcoming dataset limitations, improving model interpretability, and deploying AI-driven solutions in real-world agricultural situations should gain more attention.

Tab. 1. Comparative analysis of different studies.

Reference	**Methodology used**	**Limitations**
Iniyan et al. [8]	SVM, ANN for plant disease detection.	Limited dataset, potential overfitting
Saleem et al. [14]	Comparison of ML vs DL models	DL models require high computational power
Dixit et al. [5]	Machine Learning for wheat crop disease	ML models lack scalability for large datasets

to be continued in the next page

Tab. 1. Comparative analysis of different studies (continued).

Reference	**Methodology used**	**Limitations**
Bakr et al. [1]	DenseNet with Transfer Learning.	Transfer learning effectiveness depends on pre-trained model
Balafas et al. [2]	Comparison of ML and DL models	ML models underperform compared to DL
Khalid & Karan [11]	Deep Learning-based model for classification	End-to-end DL model needs large labeled datasets
Sharma & Guleria [16]	Comparison of CNN-based image classification models	CNNs require large amounts of training data
Yu [20]	Evaluation of deep learning architectures	Performance varies across architectures
Shao et al. [15]	Hybrid ViT-CNN for fine-grained classification	Hybrid model complexity may hinder real-time deployment
Kalim et al. [10]	Hybrid CNN-RF model for citrus leaf disease	CNN-RF may struggle with unseen datasets
Bera et al. [3]	Attention-based deep learning network	Attention-based models increase inference time
Gao et al. [7]	Dual-branch channel attention model	Attention mechanism increases computational cost
Wang et al. [19]	Comparative analysis of ML and DL approaches	Deep learning models need extensive training
Lorente et al. [12]	Comparison of classical and deep learning techniques	Classic methods less accurate than DL
Wu et al. [13]	Comprehensive review of DL for image classification	Review lacks implementation details
Shoaib et al. [17]	Review of advanced deep learning models	Advanced DL models require extensive resources
Furqan et al. [6]	Deep learning for plant disease classification	Limited generalization on diverse plant datasets
Kabir et al. [9]	GLCM feature extraction analysis	GLCM feature extraction is computationally expensive
Chen et al. [4]	Handling occlusion and illumination in fruit detection	Occlusion handling reduces detection speed
Subbarayudu & Kubendiran [18]	Survey on ML, DL techniques for smart agriculture	Survey lacks experimental validation

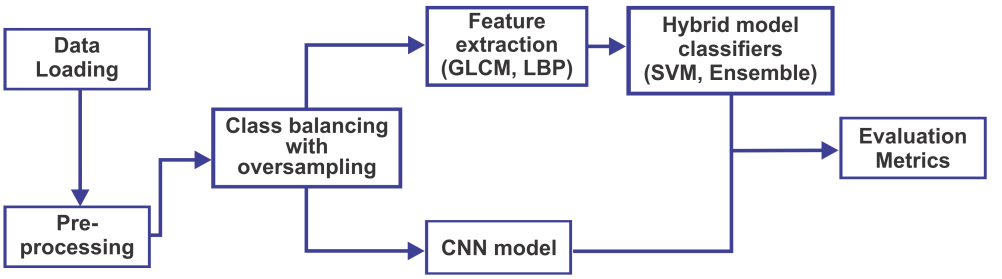


Fig. 1. Proposed system model demonstrating the CNN model pathway and the Hybrid model pathway.

3. Methodology

The methodology for plant disease detection (Figure 1) highlights two separate approaches: a CNN-based model and a hybrid traditional machine learning model. The process begins with data loading, where the dataset is prepared and structured for analysis, followed by preprocessing, which includes resizing, normalization, and augmentation to ensure consistency and diversity in the data.

To address class imbalances, class balancing with oversampling was employed to ensure the equal representation of all classes in the dataset. The workflow then branched into two distinct pathways.

CNN Model Pathway: The balanced dataset was directly input into a CNN model, which automatically extracted hierarchical spatial features and performed end-to-end classification.

Hybrid model pathway: The balanced dataset undergoes feature extraction using GLCM and LBP to capture texture-based information. These features are then used to train traditional classifiers such as SVMs and ensemble models.

The two pipelines meet at the evaluation phase where they are evaluated independently using performance metrics like accuracy, precision, recall, and F-1 score. This approach allowed a comparison of the two approaches and highlighted the advantages and application cases of both the scenarios.

3.1. Data collection and preprocessing

The dataset images were obtained directly from agricultural fields in the nearby village of Kirtan, Hisar, Haryana (India). The acquired images mimic the field environment and are therefore practical. An expert plant pathologist labelled multiple images of a

Class	Shuffled sample 1	Shuffled sample 2
Healthy		
Diseased		

Fig. 2. Randomly shuffled samples from the dataset.

particular plant disease to create high-quality annotated data suitable for the effective training and evaluation of models.

3.2. Data loading

The dataset, which contained labeled images of plant leaves, was organized into structured directories. Using TensorFlow, the data were loaded and automatically split into training, validation, and testing sets. This ensured a clean separation for model training and performance evaluation.

Representative samples from the dataset (Figure 2) illustrate various plant leaf conditions utilized for training and evaluation purposes. This encompasses healthy and diseased leaves across various classifications. Healthy leaf samples demonstrate uniform

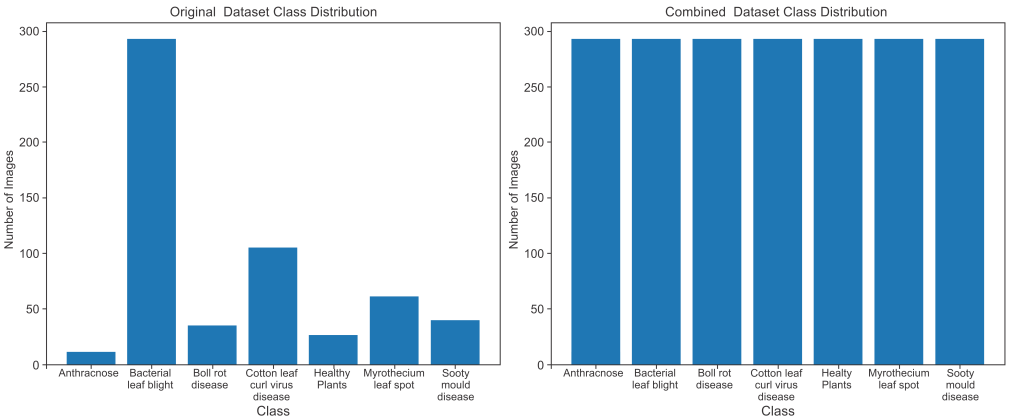


Fig. 3. Exploratory Data Analysis of the data spread before and after oversampling.

color and texture, while diseased samples show distinct symptoms, including spots, discoloration, or fungal patches. The images were obtained under natural field conditions, ensuring that the dataset accurately reflects real-world agricultural environments.

3.2.1. Image resizing and normalisation

All images were resized to 224×224 pixels to ensure the image size was compatible with the deep learning architectures. The pixel values were normalized between 0 and 1.

$$I_{\text{norm}}(x, y) = \frac{I(x, y)}{255}\,. \tag{1}$$

3.2.2. Class balancing with oversampling

In order to rectify the imbalance in classes, oversampling techniques were used. Figure 3 shows how oversampling affects the distribution of the dataset, showing better balance among plant disease classes. This guarantees equal contributions from each class during training, which is essential for avoiding biased predictions and enhancing model generalization in practical situations.

3.2.3. Dataset augmentation

Training used to be performed on data with augmented examples for improved generalization. The augmentation techniques included the following:

- rotation (±20 degrees),
- horizontal and vertical flipping,
- scaling (±10%).

This process artificially increases the dataset diversity, reduces overfitting, and improves the robustness to real-world variations.

3.3. Hybrid model

The Hybrid model combines traditional feature extraction techniques with machine-learning classifiers.

3.3.1. Feature extraction

Grey-level co-occurrence matrix (GLCM): Textural features, such as contrast, energy, homogeneity, and correlation, were derived. The formulas for contrast and energy are as follows:

$$\text{Contrast} = \sum_{k,l} Z(k,l) \cdot (k-l)^2 \,, \tag{2}$$

$$\text{Energy} = \sum_{k,l} Z(k,l)^2 \,. \tag{3}$$

Local Binary Patterns (LBP): Local texture patterns were encoded as binary values by comparing each pixel with its neighbors:

$$\text{LBP}(k_a, l_a) = \sum_{c=0}^{c-1} 2^c \cdot S(I_c - I_k) \,. \tag{4}$$

The GLCM was chosen for its robust capability in characterizing texture by summarizing the spatial relationships of pixel intensities, which is vital for differentiating plant diseases. We employed Local Binary Patterns (LBP), which are robust against illumination variations with sufficient precision, to analyze plant images collected in a natural field environment. Both methods were selected for their previous successes in similar agricultural studies.

3.3.2. Classifier models

Traditional classifiers

The features were then used to train classical classifiers like support vector machines or gradient boosting classifiers.

SVM with RBF kernel

We adopt the SVM model with RBF kernel, in which the kernel function is given as:

$$K(k,m) = \exp\left(-\gamma \|k-m\|^2\right) \,, \tag{5}$$

where γ is a parameter that balances the contribution of a single training example.

Ensemble classifiers

Stacking Multiple classifier outputs like SVM with RBF kernel, K-nearest Neighbors (KNN), Random Forest (RF) and Gradient Boosting (GB) were combined to form a stacked ensemble. The output of ensemble classifier can be expressed as:

$$\hat{y}_{\text{final}} = \sum_{i=1}^{n} w_i \cdot \hat{y}_i \,, \tag{6}$$

where $\hat{y}_i$ is the predicted output from the i-th classifier, and w_i are the corresponding weights of each classifier.

3.4. CNN-based deep learning model

The CNN model was developed as a standalone solution for plant disease classification.

3.4.1. Convolutional layers

These layers extracted spatial features from the images using filters. The feature maps produced represent patterns such as edges and textures. The convolution operation is defined as:

$$f(m,n) = (I * K)(m,n) = \sum_{i,j} I(m+i, n+j) \cdot K(i,j) \,, \tag{7}$$

where K denotes the convolutional kernel.

3.4.2. Pooling layers

By contrast, max-pooling stacked down the spatial dimension of feature maps by rejecting unimportant features.

$$P(m,n) = \max \left\{ f(o,q) \right\}_{(o,q)\in \text{window}} \,. \tag{8}$$

3.4.3. Dropout Layers

These layers randomly deactivate neurons during training to prevent overfitting.

3.4.4. Fully connected layers

The last fully connected layers transform the retrieved characteristics to the class probability with the softmax function. where the softmax is given by:

$$\hat{y}_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \,, \tag{9}$$

where z_i represents the input to the i-th output unit.

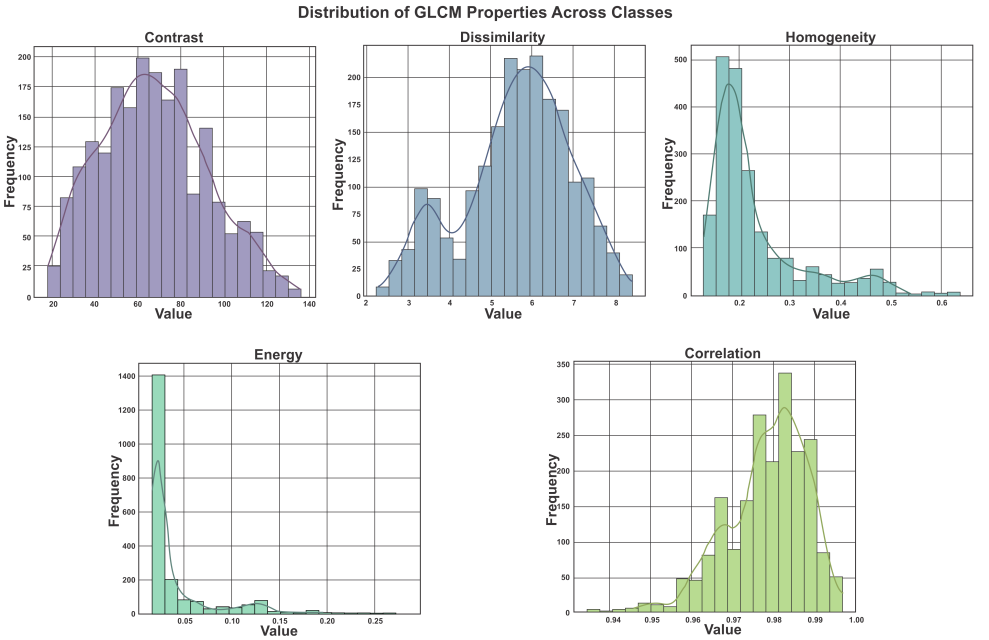


Fig. 4. Distribution of GLCM properties across classes.

3.4.5. Training process

The model was trained with Adam optimizer and categorical cross-entropy loss. We used the following loss function:

$$L = -\sum_{i} y_i \log(\hat{y}_i)\,. \tag{10}$$

We use early stopping to stop training as soon as the validation performance did not improve by more than a threshold used to control resource usage.

The algorithms applied in the Hybrid and the CNN models are shown as Algorithm 1, page 67, and Algorithm 2, page 68, respectively.

4. Results

Visualizations in Figure 4 illustrate the distribution of various GLCM properties—contrast, dissimilarity, homogeneity, energy, and correlation—across different classes within the dataset.

Algorithm 1 Plant disease detection using the Hybrid model

Input: Dataset of plant images D resized to 224×224 pixels.
Output: Class predictions for test images.
Data Loading and Preprocessing:
Load dataset D and split into D_{train}, D_{val}, and D_{test}.
Normalize the pixel values and apply Gaussian blurring for noise reduction.
Feature Extraction:
Compute GLCM features: contrast, energy, homogeneity, correlation.
Compute LBP features:

$$\text{LBP}(k_a, l_a) = \sum_{c=0}^{c-1} 2^c \cdot S(I_c - I_k)\,.$$

Model Construction:
Train individual classifiers on the extracted features.
SVM with RBF kernel:

$$K(k, m) = \exp\left(-\gamma \|k - m\|^2\right)\,.$$

Train Random Forest and Gradient Boosting classifiers.
Ensemble Model:
Combine predictions from classifiers using stacking:

$$\hat{y}_{\text{final}} = \sum_{i=1}^{n} w_i \cdot \hat{y}_i\,.$$

Model Evaluation:
Evaluate the ensemble model on D_{test}.
Compute evaluation metrics: accuracy, precision, recall, and F1-score.
Class Prediction:
For an input test image, predict the disease class using the ensemble model.

1. **Contrast and Dissimilarity** exhibited a roughly symmetrical distribution, indicating that the texture variation in pixel intensity across classes follows a predictable pattern, which aids in distinguishing fine-grained details among diseases.
2. **Homogeneity** showed a skewed distribution toward lower values, suggesting that most images have less uniform textures, which are characteristic of diseased plant surfaces.
3. **Energy** has a highly skewed distribution, where most images exhibit low energy

Algorithm 2 Plant disease detection using CNN-based deep learning

Input: Dataset of plant images D, resized to 224×224 pixels.
Output: Class predictions for test images.
Data Loading and Preprocessing:
Load the dataset D and split it into training D_{train}, validation D_{val}, and test D_{test} sets.
Normalize pixel values:

$$I_{\text{norm}}(x, y) = \frac{I(x, y)}{255}\,.$$

Apply data augmentation (rotation, flipping, and scaling).
Model Construction:
Define the CNN architecture:
Convolutional layers for feature extraction:

$$f(x, y) = (I * K)(x, y)\,.$$

Pooling layers for dimensionality reduction:

$$P(x, y) = \max\{f(i, j)\}_{(i,j)\in\text{window}}\,.$$

Fully connected layers for classification.
Model Training:
Compile the model using categorical cross-entropy loss:

$$L = -\sum_{i} y_i \log(\hat{y}_i)\,.$$

Use the Adam optimizer and train the model on D_{train} with validation on D_{val}.
Model Evaluation:
Test the trained model on D_{test}.
Compute evaluation metrics: accuracy, precision, recall, and F1-score.
Class Prediction:
For an input test image, predict the disease class using the trained CNN.

values, reflecting lower uniformity or regularity in pixel patterns across most diseased samples.

4. **Correlation** demonstrates a bell-shaped curve, highlighting consistent relationships between neighboring pixel intensities, which can be leveraged to identify patterns specific to certain diseases.

Overall, these distributions emphasise the importance of GLCM features in capturing

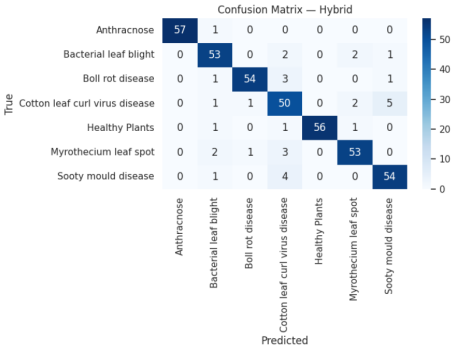


Fig. 5. Confusion matrix of Hybrid model.

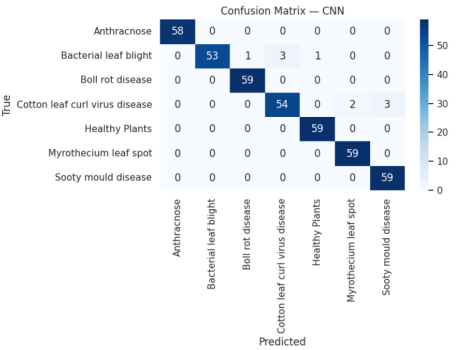


Fig. 6. Confusion matrix of CNN model.

critical textural details across various plant disease classes, making them essential inputs for traditional machine-learning classifiers.

The two confusion matrices provide a comparative evaluation of the Hybrid model (Figure 5) and the CNN model (Figure 6) for plant disease detection across multiple classes. The diagonal entries indicate the correct classifications, whereas the off-diagonal entries represent misclassifications.

1. The Hybrid model achieved strong accuracy across most classes (Figure 5). Anthracnose was classified almost perfectly (57/58 correct), while Boll Rot Disease (54/59) and Myrothecium Leaf Spot (53/59) also showed high reliability. Bacterial Leaf Blight had 53/58 correct, with minor confusion spread across Anthracnose, Myrothecium, and Cotton Leaf Curl Virus Disease. Healthy Plants were identified accurately in 56/59 cases, with a few errors toward Cotton Leaf Curl Virus Disease. The most challenging category was Cotton Leaf Curl Virus Disease, with 50/59 correct and misclassifications distributed into multiple related classes, including Sooty Mould Disease and Boll Rot Disease. Sooty Mould Disease itself was well distinguished (54/59), though again some overlap occurred with Cotton Leaf Curl Virus Disease. Overall, the Hybrid model performed robustly, though diseases with similar visual symptoms remained a source of confusion.

2. The CNN model demonstrated excellent performance (Figure 6), with near-perfect alignment between predicted and true labels. Anthracnose, Healthy Plants, and Myrothecium Leaf Spot were classified with 100% accuracy, while Boll Rot Disease and Sooty Mould Disease also achieved very high recognition rates (59/59 correct). Minor misclassifications appeared in Bacterial Leaf Blight (53/58 correct) and Cotton Leaf Curl Virus Disease (54/59 correct), though these errors were few and distributed

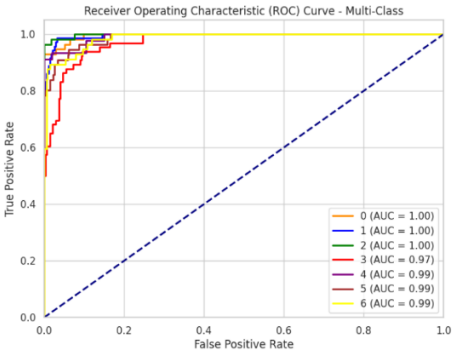


Fig. 7. ROC curves for the Hybrid model.

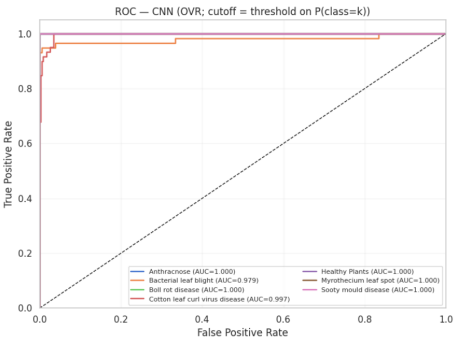


Fig. 8. ROC curves for the CNN model.

across related categories. Overall, the CNN model outperformed the Hybrid approach, achieving higher accuracy across all classes and showing stronger reliability in distinguishing visually similar diseases.

The ROC curves in Figures 7 and 8 illustrate the classification performance of the Hybrid and CNN models, respectively, across multiple plant disease categories. In both cases, the curves lie close to the top-left corner, indicating a very high true positive rate with minimal false positives. The Hybrid model (Figure 7) achieves near-perfect discrimination, with AUC values ranging from 0.97 to 1.00 across all seven classes, suggesting consistent reliability even in multi-class settings. Similarly, the CNN model (Figure 8) demonstrates excellent performance, with most classes such as Anthracnose, Healthy Plants, and Sooty mould disease achieving an AUC of 1.000, while a few, like Bacterial leaf blight (AUC = 0.979), are marginally lower but still highly accurate. Overall, both models show outstanding classification ability, though the Hybrid model provides more uniform performance across classes, whereas the CNN excels in certain categories with perfect separation.

The cutoff parameter used in the construction of the ROC curves was the decision threshold on the predicted probability $P(y = k \mid x)$, varied over the full interval $[0, 1]$.

In summary, the CNN model outperforms the Hybrid model by achieving near-perfect classification in most categories, highlighting its effectiveness in handling complex patterns and large datasets, whereas the Hybrid model exhibits moderate misclassifications, particularly in challenging classes like "Cotton Leaf Curl Virus Disease." Building on these findings, the next section provides a direct comparison of both models across key evaluation metrics, offering a deeper understanding of their relative strengths, weaknesses, and suitability for practical deployment.

The scores achieved by the two methods are compared in the Table 2.

Tab. 2. Scores of the Hybrid model and CNN model

Class	Hybrid model			CNN		
	precision	**recall**	**F1-score**	**precision**	**recall**	**F1-score**
Anthracnose	1.00	0.98	0.99	1.00	1.00	1.00
Bacterial leaf blight	0.89	0.91	0.90	1.00	0.91	0.95
Boll rot disease	0.96	0.92	0.94	0.98	1.00	0.99
Cotton leaf curl virus disease	0.79	0.85	0.82	0.95	0.92	0.93
Healthy Plants	1.00	0.95	0.97	0.98	1.00	0.99
Myrothecium leaf spot	0.91	0.90	0.91	0.97	1.00	0.98
Sooty mould disease	0.89	0.92	0.90	0.95	1.00	.98
Accuracy		0.9173			0.9757	
Macro Average	0.92	0.92	0.92	0.98	0.98	0.98
Weighted Average	0.92	0.92	0.92	0.98	0.98	0.98
Computational time [s]		27.14			64.66	

5. Comparison

The comparative performance of the Hybrid and CNN models (Figure 9) is shown on major evaluation measures: accuracy, precision, recall, and F1-score. CNN model performed consistently better than the Hybrid model with the accuracy of 97.57%, precision of 98%, recall 98%, F1-score 97%, which shows the excellent ability to deal with complex data and generalize appropriately. By contrast, our Hybrid model performed at 91.73% in all metrics. Despite slightly worse performance of the Hybrid model, it does qualify as a strong competitor in situations where computational efficiency and interpretability is crucial, especially when resources are limited. This comparison highlights that the CNN

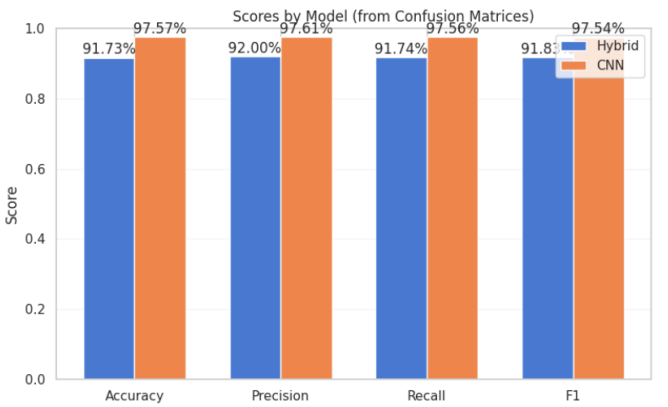


Fig. 9. Comparison charts of Hybrid and CNN models of the system pathway.

Tab. 3. Comparative analysis of Hybrid model and CNN model

Aspect	CNN-based model	Hybrid model
Accuracy	High, particularly on large and diverse datasets.	Moderate, depends on the quality of handcrafted features.
Scalability	Highly scalable to large datasets.	Limited scalability due to reliance on manual feature engineering.
Interpretability	Low, functions as a "black-box" model.	High, with explicit and interpretable features from GLCM and LBP.
Computational Efficiency	Computationally intensive; requires GPUs.	Efficient, suitable for environments with limited computational resources.
Generalization	Strong generalization on unseen data.	Moderate, struggles with variability in new datasets.
Suitability for Small Datasets	Limited; prone to overfitting without sufficient data.	High; performs well on small, structured datasets.
Ease of Implementation	Relatively straightforward with end-to-end learning.	Requires domain expertise for feature extraction.
Real-World Applications	Best for large-scale, automated systems.	Ideal for resource-constrained or interpretable decision-making scenarios.

model is well suited for large and diverse datasets, yet, the Hybrid model offers a powerful and interpretable alternative for simple or small scale applications. The comparison of the models in several aspects is shown in Table 3.

Our dataset suffers from potential bias as we have limited the dataset, based on a specific geographical location (Kirtan village, Hisar, India) and the diseases we select and can introduce a challenge to generalize the results to other types of diseases. Models primarily trained on local data might not generalize to other agricultural environments. In order to mitigate these biases and improve the robustness of the model, future studies should incorporate data from different regions and crop species.

6. Conclusion

Early diagnosis of plant disease is imperative for advanced Precision Agriculture and sustainable food production. In this paper, a comparison between a deep learning model using CNNs and a Hybrid model that integrates standard feature extraction methods and

machine learning classifiers is conducted. Evaluation of both the models was performed using a custom dataset of plant leaf pictures collected locally with evaluation metrics with accuracy, precision, recall, and F1-score.

CNN model showed improved classification accuracy and scalability, so that it is fit for large-scale, unmanned agricultural applications. Unfortunately, it is computationally intensive and it is not applicable in a low-resource environment. They also demonstrated that the Hybrid (which was only slightly less accurate) was highly interpretable and computationally expedient. These features increase their real-world applicability (especially in pampered planet circumstance or when transparency of decision-making is critical). Finally, class imbalance was addressed by the use of data augmentation and oversampling methods to guarantee fair representation during training of the model. By visual and statistical analyses, we demonstrated the effectiveness of GLCM and LBP texture information for disease classification.

Notwithstanding these strengths, the study presents limitations. The dataset was collected solely from Kirtan village in Hisar, Haryana (India), potentially limiting the generalizability of the models to other agricultural contexts characterized by different environmental conditions and crop varieties. Future research should prioritize the curation of diverse datasets from various regions, the development of lightweight deep learning architectures suitable for on-field deployment, and the standardization of feature extraction techniques for Hybrid models.

In summary, both CNN and Hybrid models demonstrate considerable potential for facilitating intelligent, real-time detection of plant diseases. Their incorporation into agricultural systems, including drone surveillance and mobile diagnostic tools, provides farmers with actionable insights, enhances crop yield, and promotes sustainable farming practices globally.

Author's declarations

Conflict of interest

The authors have no conflict of interest to report.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgement

The authors sincerely acknowledge Dr. Man Mohan, Assistant Scientist at CCS Haryana Agricultural University, Hisar, for his invaluable assistance in labeling the plant disease dataset.

References

[1] M. Bakr, S. Abdel-Gaber, M. Nasr, and M. Hazman. Tomato disease detection model based on densenet and transfer learning. *Applied Computer Science* 18(2):56–70, 2022. doi:10.35784/acs-2022-13.

[2] V. Balafas, E. Karantoumanis, M. Louta, and N. Ploskas. Machine learning and deep learning for plant disease classification and detection. *IEEE Access* 11:114352–114377, 2023. doi:10.1109/ACCESS.2023.3324722.

[3] A. Bera, D. Bhattacharjee, and O. Krejcar. An attention-based deep network for plant disease classification. *Machine Graphics and Vision* 33(1):47–67, 2024. doi:10.22630/MGV.2024.33.1.3.

[4] J. Chen, J. Wu, Z. Wang, H. Qiang, G. Cai, et al. Detecting ripe fruits under natural occlusion and illumination conditions. *Computers and Electronics in Agriculture* 190:106450, 2021. doi:10.1016/J.COMPAG.2021.106450.

[5] N. Dixit, R. Arora, and D. Gupta. Wheat crop disease detection and classification using machine learning. In: *Infrastructure Possibilities and Human-Centered Approaches with Industry 5.0*, pp. 267–280. IGI Global Scientific Publishing, 2024. doi:10.4018/979-8-3693-0782-3.ch016.

[6] M. Furqan, S. Rizvi, and P. Singh. Plant disease diagnosis and classification using deep learning. *International Journal of Pathology and Drugs* 01(1):1–8, 2023. doi:10.54060/pd.2023.1. https://pd.a2zjournals.com/index.php/pd/article/view/1.

[7] R. Gao, R. Wang, L. Feng, Q. Li, and H. Wu. Dual-branch, efficient, channel attention-based crop disease identification. *Computers and Electronics in Agriculture* 190:106410, 2021. doi:10.1016/j.compag.2021.106410.

[8] S. Iniyan, R. Jebakumar, P. Mangalraj, M. Mohit, and A. Nanda. Plant disease identification and detection using support vector machines and artificial neural networks. In: *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, pp. 15–27. Springer, Singapore, 2020. doi:10.1007/978-981-15-0199-9_2.

[9] M. Kabir, F. Unal, T. C. Akinci, A. A. Martinez-Morales, and S. Ekici. Revealing GLCM metric variations across a plant disease dataset: A comprehensive examination and future prospects for enhanced deep learning applications. *Electronics* 13(12):2299, 2024. doi:10.3390/electronics13122299.

[10] H. Kalim, A. Chug, and A. P. Singh. Citrus leaf disease detection using hybrid CNN-RF model. In: *2022 4th International Conference on Artificial Intelligence and Speech Technology (AIST)*, pp. 1–4, 2022. doi:10.1109/AIST55798.2022.10065093.

[11] M. Khalid and O. Karan. Deep learning for plant disease detection: Deep learning for plant. *International Journal of Mathematics, Statistics, and Computer Science* 2:75–84, 2023. doi:10.59543/ijmscs.v2i.8343.

[12] Ò. Lorente, I. Riera, and A. Rana. Image classification with classic and deep learning techniques. arXiv, arXiv:2105.04895, 2021. doi:10.48550/arXiv.2105.04895.

[13] W. Meng, J. Zhou, P. Y., W. S., and Z. Y. Deep learning for image classification: A review. In: S. Ruidan, Y.-D. Zhang, and A. F. Frangi (Eds.), *Proceedings of 2023 International Conference on Medical Imaging and Computer-Aided Diagnosis (MICAD 2023)*, pp. 352–362. Springer Nature Singapore, 2024. doi:10.1007/978-981-97-1335-6_31.

[14] M. H. Saleem, J. Potgieter, and K. M. Arif. Plant disease detection and classification by deep learning. *Plants* 8(11):468, 2019. doi:10.3390/plants8110468.

[15] R. Shao, X.-J. Bi, and Z. Chen. Hybrid ViT-CNN network for fine-grained image classification. *IEEE Signal Processing Letters* 31:1109–1113, 2024. doi:10.1109/LSP.2024.3386112.

[16] S. Sharma and K. Guleria. Deep learning models for image classification: Comparison and applications. In: *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 1733–1738, 2022. doi:10.1109/ICACITE53722.2022.9823516.

[17] M. Shoaib et al. An advanced deep learning models-based plant disease detection: A review of recent research. *Frontiers in Plant Science* 14:1158933, 2023. doi:10.3389/fpls.2023.1158933.

[18] C. Subbarayudu and M. Kubendiran. A comprehensive survey on machine learning and deep learning techniques for crop disease prediction in smart agriculture. *Nature Environment and Pollution Technology* 23(2):619–632, 2024. doi:10.46488/NEPT.2024.v23i02.003.

[19] P. Wang, E. Fan, and P. Wang. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters* 141:61–67, 2021. doi:10.1016/J.PATREC.2020.07.042.

[20] Y. Yu. Deep learning approaches for image classification. In: *Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering (EITCE '22)*, ACM International Conference Proceeding Series, pp. 1494–1498, 2022. doi:10.1145/3573428.3573691.

Vikram Singh received his Ph.D. degree in Computer Science from Kurukshetra University, Kurukshetra, Haryana, India, in 2004. Presently he is working as Professor in the Department of Computer Science and Engineering at Chaudhary Devi Lal University, Sirsa, Haryana, India.

Rajesh Kumar is pursuing a doctoral degree in Artificial Intelligence under the guidance of Professor Vikram Singh at the Department of Computer Science and Engineering, Chaudhary Devi Lal University, Sirsa, Haryana, India. His research focuses on applying machine learning to agricultural automation and precision farming. He currently serves as a D.T.P. Operator at Chaudhary Charan Singh Haryana Agricultural University, Hisar, Haryana, India, a role he has held since 1994.

Fine-Tuning Stable Diffusion for Generating 2D Floor Plans Using Prompt Templates

Ahmed Mostafa, Omar Amir, Ali M. Mohamed and Marwa O. Al Enany*
Higher Institute of Computer Science and Information Systems,
October 6 University, Culture and Science City, Egypt
Corresponding author: Marwa O. Al Enany (marwaanny33@gmail.com)

Submitted: 02 Feb 2025 **Accepted:** 20 May 2025 **Published:** 30 Sep 2025

Abstract Automated generation of 2D floor plans is crucial for architectural design, requiring models to balance precision and adaptability to user-defined specifications. Diffusion models, like Stable Diffusion, excel at generating high-quality images but lack an intrinsic understanding of structured layouts such as floor plans. Conversely, Graph Neural Networks (GNNs) are adept at encoding relational data, representing floor plan objects as nodes and their connections as edges, but they are not generative or capable of processing textual inputs. In this work, we fine-tune Stable Diffusion 1.5 on a custom dataset of floor plans, leveraging structured prompt templates to constrain the model's creativity and guide it toward generating concise, error-tolerant outputs. This research suggests integrating the generative capabilities of diffusion models with the representational strengths of GNNs to overcome inherent challenges in diffusion models, like their inability to explicitly encode spatial relationships. This integration could expand the capabilities of these models, empowering them to comprehend and produce structured layouts more effectively. While computational constraints limited our exploration of this hybrid architecture, our results demonstrate that prompt engineering and dataset preprocessing significantly improve the output quality. This study highlights the potential for generative models in architectural tasks and lays the groundwork for integrating logical reasoning into diffusion-based architectures.

Keywords: graph neural networks (GNNs), diffusion model, latent diffusion, floorplan representation.

1. Introduction

Human culture deeply intertwines with the history of architecture and architectural drawings, reflecting the ability to conceptualize and design living spaces. Architectural drawings, meticulously crafted by hand and archived on paper or as scanned raster images, continued until the second half of the 20th century. Even with the widespread adoption of computer-aided design (CAD) software [7], most architectural drawings remain primarily distributed in image formats, limiting the depth of information that can be extracted and analyzed.

The challenge of generating floor plans that meet specific user requirements has long been a complex task for architects and designers [33]. Traditional methods rely heavily on manual design processes, requiring extensive expertise and time-consuming iterations. This process is inherently complex and requires meticulous attention to detail. Architects and engineers must consider a myriad of factors, including spatial relationships, structural integrity, building codes, and client preferences. To ensure both functionality

and compliance with regulatory standards, architects and engineers must precisely configure each element, from the placement of walls, doors, and windows to the allocation of rooms and incorporation of utilities. Moreover, the manual approach to floor plan creation poses challenges in efficiency and productivity. Engineers must painstakingly adjust dimensions, realign components, and ensure consistency across different sections of the plan. This meticulous work not only prolongs project timelines but also redirects valuable resources towards less innovative design aspects. The pressure to deliver accurate and high-quality floor plans under tight deadlines further exacerbates the strain on professionals in the industry.

In light of these challenges, there is a growing need for automated solutions [35] that can streamline the floor plan generation process. An effective solution would reduce the manual workload, minimize errors, and accelerate the design phase, allowing engineers and architects to focus on creativity and innovation. Automation can also enhance the ability to rapidly explore multiple design alternatives, providing clients with a broader range of options and facilitating more informed decision-making.

Recent advancements in artificial intelligence and machine learning have opened new possibilities for automated floor plan generation [2, 9, 25, 32]. Unlike traditional design methods that rely solely on human expertise, AI-driven approaches can rapidly generate multiple design iterations based on input parameters.

Diverse machine learning methodologies have been employed in floor plan analysis. Convolutional Neural Network-based methodologies have been predominantly utilized due to their applicability to many sorts of floor-plan photographs [5]. CNN-based methodologies necessitate only fundamental image pre-processing techniques and exhibit robustness to floor plan noise. Furthermore, they can be utilized across many drawing styles without necessitating modification, rendering them quick and adaptable.

Nevertheless, due to the pixel-level segmentation employed by these approaches, accurately capturing the precise contours of indoor features is challenging. To address this issue, some methodologies have integrated supplementary post-processing processes that refine the neural network's output. However, this results in the loss of features inherent to the original indoor elements, such as the representation of polygons as line vectors [6, 18]. For instance, walls must possess distinct thickness and area, but, when the shapes become indistinct during the convolution layers, the walls are ultimately represented as line vectors by the post-processing techniques.

For certain user applications, such as representing navigable areas in IndoorGML format [31], abstracting a floor plan layout using machine learning models may be crucial. However, the inherent flexibility and deformability of vector data allows for the adaptation of vector outputs that preserve the original floor plan's form into various objects based on user intent.

This research contributes to developing a novel model that generates 2D floor plans

automatically by leveraging user-specified inputs such as the number of bedrooms, desired room types, and other key architectural constraints. The proposed model aims to address several key challenges in automated floor plan generation:

- translating abstract user requirements into precise spatial configurations,
- ensuring functional and logical room relationships,
- maintaining architectural design principles,
- providing rapid, customizable design solutions.

By utilizing stable diffusion, a generative AI technique, the research demonstrates the potential of AI to streamline the initial stages of architectural design. The model leverages the diffusion model's ability to generate complex, contextually coherent images by progressively denoising latent representations, enabling the creation of floor plans that transform user inputs into detailed spatial layouts. Another contribution is exploring the capabilities and limitations of diffusion models in generating reliable 2D floor plans while proposing hybrid approaches that combine generative and graph-based methods to further push the boundaries of the field.

The remainder of this paper is organized as follows. Section 2 reviews related work in floor plan generation. Section 3 presents our proposed framework. Section 4 describes the dataset design. Section 5 details the Stable Diffusion model structure. Section 6 covers the implementation details. Section 7 presents the evaluation results. Section 8 discusses the findings, and Section 9 concludes the paper.

2. Related work

The generation of 2D floor plans has advanced significantly, leveraging various generative models to address challenges in structured design and adaptability to user-defined parameters. This section highlights key contributions that inform and contextualize this work, focusing on diffusion models, text-conditioned generation, and data-structure-driven approaches.

2.1. Diffusion-based models for floor plan generation

The author in [11] recommend using a diffusion-based approach to create realistic floorplan images that include room types, furniture specifications, and fenestration details which were often omitted in earlier models, like HouseGAN++ [21]. This method surpasses current results and generates helpful floorplans. However, the direct pixel-input method limits scalability. Two solutions, Cascade Diffusion models and Latent Diffusion models [28], have been introduced to address this issue. Cascade diffusion models incrementally improve image resolution, while Latent Diffusion models map high-dimensional inputs into a more manageable latent space.

The authors in [30] introduced a diffusion-based method that directly predicts a list of polygons for each room, utilizing a transformer architecture. This approach employs a denoising process on 2D coordinates of room and door corners, integrating both discrete and continuous denoising steps to establish geometric relationships such as parallelism and orthogonality. Evaluations on the RPLAN dataset [36, 37] demonstrated significant improvements over state-of-the-art methods, with the capability to generate non-Manhattan structures and control the exact number of corners per room.

A novel approach called HouseCrafter is proposed in [22] that transforms 2D floorplans into complete 3D indoor scenes. By adapting a 2D diffusion model trained on web-scale images, the method generates consistent multi-view RGB-D images across different locations of the scene. These images are generated autoregressively, guided by the floorplan, ensuring consistency and enabling high-quality 3D scene reconstruction. Experiments on the 3D-Front dataset demonstrated the effectiveness of HouseCrafter in generating house-scale 3D scenes.

Further on, in [10] a shear wall layout generation method based on a diffusion process is proposed. Compared with the StructGAN method, the diffusion-based approach demonstrates improved performance in generating realistic and efficient shear wall layouts, contributing to advancements in structural design automation. According to enhancing diffusion models, in [26] the diffusion models in the context of computational design are assessed, particularly on floor plans. A method for refining diffusion models via semantic encoding is suggested. The semantic encoding proposed in this paper enhanced the validity of produced floor plans to 90%. Nevertheless, the article also highlights deficiencies in existing diffusion models, primarily because to an absence of semantic comprehension.

2.2. Text-Conditioned Floor Plan Generation

The research of [17] introduced a pioneering dataset called Tell2Design of over 80 000 floor plans paired with natural language descriptions. This work explored the use of Sequence-to-Sequence models for translating textual input into spatially coherent layouts. By addressing architectural constraints through text-conditioned generation, the authors opened avenues for leveraging natural language as a design interface, making floor plan generation accessible and user-friendly. However, experimental results show that current text-conditional picture generation methodologies fail to address the design creation problem, highlighting the difficulty in understanding ambiguous information and the characteristics of design diversity in the task.

The study in [39] proposed a two-phase method for text-to-floorplan generation, leveraging large language models (LLMs) to create initial layouts from textual descriptions. The approach integrates LLMs to interpret and generate spatial configurations, enhancing the alignment between user requirements and generated designs. The collaboration between LLMs and visual generative models in [8] generates LayoutGPT which is

a method that converts a big language model into a visual planner via in-context learning and CSS style prompts. LayoutGPT can generate credible visual configurations in both image space and three-dimensional indoor environments besides improving image compositions by producing accurate layouts and obtaining performance in interior scene synthesis that is comparable to supervised methods.

Another study in [14] presented a method to automatically render 2D floor plan images from natural language descriptions. This work represents an early attempt to synthesize floor plans directly from textual inputs, bridging the gap between language and visual design in the architectural domain.

2.3. Data Structure-Driven Approaches

For data structure-driven approaches, in [19] a framework that focuses on numerical attributes of floor plans is proposed, including room dimensions and intermediate representations, to ensure adherence to constraints and enhance functional accuracy. New datasets and evaluation metrics are introduced, providing insights into integrating data structures for improved generative modeling of architectural layouts. The study fine-tunes a large language model (LLM), Llama3, but finds it flawed in accurately producing rooms with areas corresponding to computed polygons.

The technique proposed in [1] attempts to present floorplans using numerical vectors that encode design semantics and human behavioral characteristics. The framework comprises two components. The initial component features an automated program that transforms floorplan photos into attributed graphs. The features consist of design semantics and human behavioral characteristics produced by simulation. In the second component, it introduced an innovative LSTM Variational Autoencoder for the purposes of embedding and producing floorplans. The qualitative, quantitative, and expert assessments indicate that this embedding system generates significant and precise vector representations for floorplans, demonstrating its capacity for creating new floorplans.

Authors in [15] developed GenFloor, an interactive design system that generates optimized spatial layouts based on geometrical, topological, and performance constraints. The system introduces novel permutation methods for existing space layout graph representations, such as O-Tree and B*-Tree, enabling the generation of diverse floor plans that meet specified design criteria. GenFloor facilitates designers in their generative design workflow by providing a user-friendly interface and evaluation functionalities.

Research on extracting structured data from image floor plans has also informed generative tasks. A notable study [23] introduced techniques for creating vector and raster representations of floor plans to improve localization accuracy in indoor positioning systems. Though not directly focused on generation, this work highlights the importance of accurate preprocessing and representation, essential for downstream applications. The study proposes a computer vision method for automated map annotation, which significantly reduces the processing time from 40 minutes to 5 minutes. Despite the method's

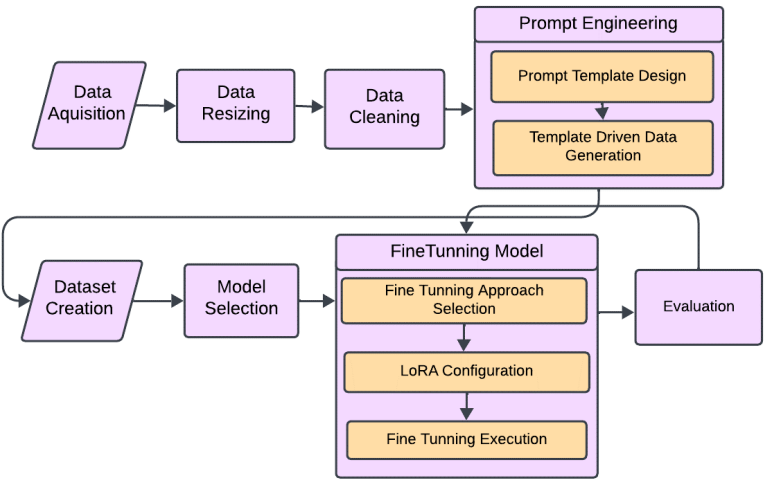


Fig. 1. The stages involved in the process of fine-tuning Stable Diffusion for floor plan generation.

limitations, users can consistently achieve the map model with minimal user modifications.

Most of the previously mentioned studies collectively illustrate the evolving landscape of generative modeling for architectural design. From enhancing realism and accuracy through diffusion-based techniques to leveraging structured prompts and intermediate representations, these advancements underscore the potential for integrating diverse methodologies.

3. The proposed framework

Our proposed framework for fine-tuning Stable Diffusion for floor plan generation consists of several key stages, as illustrated in Figure 1.

3.1. Data acquisition and preparation

This is the initial step in gathering information to train the model. The data used in this study consisted of a curated collection of 300 floorplan images. These images are obtained from various publicly available architectural and design repositories to ensure diversity in design styles and layouts. The primary goal is to create a dataset suitable for training a Stable Diffusion model capable of understanding architectural floorplans.

3.2. Data resizing

Processing the acquired data to match the required dimensions or specifications. To ensure consistency across the dataset and compatibility with the neural network architecture, all floorplan images are resized to 256×256 pixels. This resolution has been selected as it strikes a balance between computational efficiency and preserving sufficient detail in the architectural features. The resizing process employs bicubic interpolation to minimize distortion and preserve the original proportions of the designs.

3.3. Data cleaning

Removing noise, inconsistencies, and irrelevant data from the dataset to ensure high-quality input. The raw floorplan images contained extraneous elements such as annotations, text labels, and other metadata that were not essential for the intended application. To address this, all images underwent a manual and automated cleaning process. This step involves removing textual and graphical artifacts while retaining the structural integrity of the floorplans. Open-source image editing tools and Python-based libraries, such as OpenCV and PIL, are used to streamline the cleaning process.

3.4. Prompt engineering

To enhance task-specific understanding by creating well-structured and targeted inputs. Two steps are included:

- **Prompt Template Design**: Design templates for input prompts to ensure the model understands tasks effectively.
- **Template-Driven Data Generation**: Use the designed templates to generate additional synthetic data or reformat existing data to align with the task requirements.

3.5. Dataset creation

Combining processed and cleaned data to create a final dataset suitable for fine-tuning. This might include integrating real and synthetic data. Following preprocessing, the cleaned and resized images are organized into a structured dataset. Each image is saved in a standardized format (e.g., PNG) to maintain quality and reduce potential issues arising from compression artifacts. An accompanying CSV file stores metadata associated with each image, including source information and preprocessing steps, to improve reproducibility.

3.6. Model selection

Choose the base model to fine-tune. This could involve selecting a pre-trained model that aligns with the task domain.

3.7. Model fine-tuning

- **Fine-Tuning Approach Selection**: Decide on the strategy for fine-tuning (e.g., full model fine-tuning, Low-Rank Adaptation (LoRA), or adapters).
- **LoRA Configuration**: If using LoRA, configure its parameters to efficiently fine-tune large models with minimal resources.
- **Fine-Tuning Execution**: Perform the fine-tuning process using the prepared dataset and configurations.

3.8. Evaluation

Assessing the fine-tuned model's performance against predefined metrics or benchmarks. The results inform whether further adjustments are needed. The process includes feedback loops where insights from the evaluation stage or fine-tuning process may inform modifications in earlier stages, such as dataset creation, prompt design, or model configurations.

4. Dataset designing and characteristics

The main objective is to produce realistic 2D floor plan designs that adhere to a set of linguistic instructions detailing the basic parts of the floor plan. Each data sample consists of a collection of prompts that outline the essential elements of the corresponding floor plan design, which encompass: Semantics that defines the type of the described rooms, Geometry which defines the size and shape of each room, and Topology that illustrates the relationships between various rooms. It can be classified into three categories: relative location, connectedness, and inclusion. The objective is a systematic interior arrangement that conforms to the provided linguistic directives.

- **Volume**: The dataset comprises 313 images, providing a moderately sized collection for training and validation purposes.
- **Diversity**: The dataset encompasses a wide range of architectural styles, including residential, commercial, and mixed-use layouts, ensuring broad applicability of the trained model.
- **Quality Control**: Each image was reviewed post-preprocessing to verify that all unnecessary elements were successfully removed and that the structural details were preserved.

This dataset forms the foundation for the subsequent stages of model training and evaluation, ensuring high-quality inputs and consistency throughout the study. Figure 2 shows samples from our dataset.

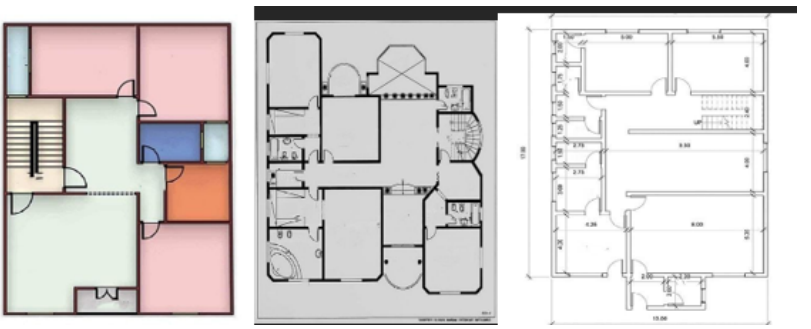


Fig. 2. Samples from the designed dataset showing variety in floor plan designs.

5. Stable Diffusion model structure

Stable Diffusion v1.5 [28, 29] is an advanced text-to-image synthesis model that leverages the principles of diffusion processes within a latent space to generate high-quality images conditioned on textual input. It builds upon the foundational work in diffusion models and latent variable models, integrating them to produce a scalable and efficient framework for image generation tasks [27]. At its core, Stable Diffusion v1.5 is a type of Latent Diffusion Model (LDM) that operates in a compressed, lower-dimensional latent space rather than directly in the high-dimensional pixel space. This approach significantly reduces computational overhead while preserving the ability to generate detailed and coherent images [32]. The decision to utilize Stable Diffusion v1.5 rather than more recent versions such as v2.0 or SDXL was both strategic and deliberate. Version 1.5 is widely regarded for its balance between quality, control, and compatibility with a broad ecosystem of tools and community-created resources. Unlike later models that introduced significant architectural changes—such as a new VAE, deeper prompt sensitivity, and more restrictive output filtering—v1.5 offers a more consistent and interpretable output across a variety of prompts, which is essential for projects requiring reproducibility and detailed prompt engineering. Additionally, the abundance of pretrained models, custom LoRAs, and fine-tuned checkpoints built on v1.5 significantly enhance flexibility and creativity without the computational overhead of retraining. Using v1.5 thus ensures that the work remains accessible, adaptable, and efficient, with outputs that are not only high in visual fidelity but also aligned with the project's specific creative and technical needs [20]. The main architecture of Stable Diffusion v1.5 model consists of the elements described in the following Subsections 5.1, 5.2, 5.3 and 5.4.

5.1. Text encoder

The model utilizes a pre-trained text encoder, typically derived from the CLIP (Contrastive Language-Image Pretraining) architecture developed by OpenAI, which aims to align text and image embeddings in a shared latent space. The CLIP Text Encoder is typically a variant of the Transformer architecture, processing the input text and outputs a fixed-length embedding vector.

The text encoder transforms input textual prompts into [4] a high-dimensional vector representation. It extracts meaningful features from the input text and transforms it into a compact, high-dimensional vector representation, providing semantic guidance to the diffusion model.

The encoded text vector is used as a conditioning input for the U-Net in the diffusion model, which uses cross-attention mechanisms to ensure the generated images align with the textual description. The encoded text vector is pretrained to generalize well across different prompts and concepts, captures nuanced relationships between words, and can work with multiple languages or dialects with fine-tuning.

However, the CLIP Text Encoder may struggle with highly abstract or nonsensical prompts or cultural or domain-specific nuances not covered during training. By leveraging the robust CLIP Text Encoder, Stable Diffusion v1.5 achieves an effective translation of textual descriptions into high-quality images, balancing semantic richness with computational efficiency.

5.2. Latent space representation

Stable Diffusion v1.5 utilizes latent space representation, a compact, high-level mathematical representation of data, to encode and manipulate data in a compressed, lower-dimensional space. This lower-dimensional representation is used in conjunction with a Variational Autoencoder (VAE) and a U-Net architecture to reduce computational complexity and optimize the model's operation. The VAE encoder compresses high-dimensional image data into a latent space and reconstructs it back into pixel space, while the U-Net operates on this latent tensor during the diffusion process. The model learns to denoise latent representations in reverse diffusion steps, optimizing for a perceptual loss to maintain high fidelity to the original data. At inference time, the trained model refines a noisy latent tensor into a meaningful latent representation, which is then reconstructed by the VAE decoder. This approach allows Stable Diffusion to work efficiently with large datasets and generate high-quality images with less computational overhead. Overall, latent space representation is a core component of Stable Diffusion v1.5, enabling efficient processing and high-quality image generation [24].

5.3. Diffusion process

The diffusion model is a U-Net architecture which is neural network responsible for predicting noise at each timestep and is adapted for denoising tasks in the latent space. The forward diffusion process involves gradually adding Gaussian noise to the latent variables over several time steps, This process, also called the noising process, gradually adds Gaussian noise to an input image over a fixed number of steps. On the other hands, the reverse diffusion process entails learning to denoise these latent variables to recover the original data distribution. This process, also called the denoising process, starts from pure noise and attempts to reconstruct the original image. This is where the model learns to "undo" the noise added during forward diffusion. Stable Diffusion performs this process in a latent space, rather than pixel space, using a LDM for efficiency [16,38]. Latent diffusion offers several advantages, including efficiency, scalability, and quality of generated images. By operating in the latent space, the model reduces data dimensionality, resulting in lower computational requirements and faster training times. The compact latent space representation also allows for high-resolution image generation, ensuring high-fidelity, fine-detail images.

5.4. Conditioning mechanism

The model incorporates a conditioning mechanism that integrates the textual embeddings from the text encoder into the diffusion model at each time step. This alignment ensures that the denoising process is guided by the semantic content of the input text, enabling coherent text-to-image synthesis [3]. Stable Diffusion incorporates conditioning to guide the reverse diffusion process toward generating specific outputs. Using a text encoder, textual information is embedded into a high-dimensional space and injected into the U-Net as cross-attention layers, or other Inputs: The process can also be conditioned on images, masks, or other inputs, enabling tasks like inpainting or image-to-image generation.

6. Implementation

To realize the proposed solution of generating precise 2D floor plans using Stable Diffusion v1.5, we implemented a fine-tuning process utilizing Hugging Face's Low-Rank Adaptation (LoRA) framework [12, 13]. This approach allowed us to adapt the pretrained diffusion model to our specific task without the need for extensive computational resources or retraining from scratch. Leveraging LoRA, we injected trainable rank decomposition matrices into the attention layers of the Transformer architecture within Stable Diffusion v1.5. This method effectively reduced the number of trainable parameters, making the fine-tuning process more efficient while maintaining the model's

capacity to learn task-specific representations. The implementation process proceeded as follows:

- The Hugging Face Transformers library was set up, ensuring compatibility with the LoRA integration. The pre-trained Stable Diffusion v1.5 model was loaded as the base model for fine-tuning. We configured the LoRA parameters to insert low-rank adaptation matrices into the attention layers, specifying the desired rank to balance between computational efficiency and model expressiveness.
- The fine-tuning dataset, comprising pairs of constrained prompts and corresponding floor plan images, was prepared to align with the requirements of LoRA training. Each prompt is structured consistently, varying only in numerical parameters, as previously described. Although the details of data preparation are discussed elsewhere, it is important to note that the dataset was formatted to be compatible with the Hugging Face dataset utilities, enabling seamless integration into the training pipeline.
- During the training process, the standard optimization techniques were utilized. The optimizer was set to AdamW with a learning rate carefully selected to ensure stable convergence without overfitting. We adopted a learning rate scheduler to adjust the learning rate dynamically based on the training progress.
- The loss function was configured to emphasize the reconstruction accuracy of the floor plans. While the primary objective remained the minimization of the denoising score matching loss inherent in diffusion models, we integrated additional components to focus on the structural aspects of the floor plans. Specifically, edge-aware loss functions that penalized discrepancies in the line structures between the generated and ground truth images was incorporated. This helped the model prioritize the preservation of architectural details crucial for floor plan accuracy.
- Training was conducted on hardware equipped with GPUs capable of handling the computational demands of the model. The use of LoRA significantly reduced memory usage, allowing the fine-tuning process to be executed on standard GPU setups without requiring distributed training or specialized hardware.
- The training progress was monitored by evaluating intermediate outputs and loss convergence. Visual inspections of generated floor plans were performed to ensure that the model was learning to produce outputs that adhered to the specified parameters in the prompts. Any signs of the model reverting to overly creative outputs were addressed by adjusting training hyperparameters, such as the learning rate or weight decay.
- Upon completion of the fine-tuning, the adapted model was saved using Hugging Face's model saving utilities. This enabled easy deployment and sharing of the model for inference tasks.

The final model was capable of generating precise 2D floor plans that accurately reflected the numerical specifications in input prompts.

7. Evaluation and Results

To validate the effectiveness of this implementation, evaluations are conducted using a set of test prompts with varying parameters. The generated floor plans are assessed for accuracy in room counts, spatial arrangements, and adherence to architectural conventions.

The performance of the fine-tuned Stable Diffusion model in generating accurate 2D floor plans is done by employing the Structural Similarity Index Measure (SSIM) as an evaluation metric. The initial proposal for the Structural Similarity Index was made in [34] by Wang, Bovik et al. in 2004. The two images being compared must be appropriately sized and aligned in order to compare them point by point. A sliding $N \times N$ (usually 11×11) Gaussian weighted window is used for the computations. Luminance, contrast, and structure are the three similarity functions that are computed on the windowed image data. The general form of the SSIM index is then created by combining the three mentioned similarity functions as:

$$\mathrm{SSIM}(x, y) = [l(x, y)] \cdot [c(x, y)] \cdot [s(x, y)] \tag{1}$$

where l, c, and s compare luminance, contrast and structure, respectively.

The ability of this index to mimic human subjectivity is its strongest attractive point. Specifically, changes in the spatial arrangement of image brightness have a significant impact on the Human Visual System (HVS) and the SSIM Index. SSIM is particularly suited for this implementation as it assesses the structural resemblance between two images, focusing on spatial configurations and line structures essential in architectural designs. The evaluation involves generating floor plan images based on prompts from a test set using both the base Stable Diffusion model and our fine-tuned model. Each generated image was compared to its corresponding ground truth floor plan using SSIM.

For implementation and testing purposes, the Fine-tuned model was tested with two different prompts. The first prompt was unusual or rarely spread in design as

- **Prompt 1**: "floor plan of house having two living rooms, one bedrooms, three bathroom, two kitchen, one garage, three store, one entrance."

The SSIM Scores of Diffusion model versus fine-tuned Diffusion model for the mentioned prompt are summarized in Table 1, numbers 1 and 2, while 10 generated images from the fine tuned Stable Diffusion model are presented in Figure 3.

The SSIM index generally ranges from -1 to 1, with 1 signifying perfect similarity, 0 denoting no similarity, and -1 representing perfect anti-correlation. Our fine-tuned model achieved an average SSIM score of 0.3426, surpassing the base model's average SSIM score of 0.3191. The higher SSIM score indicates that the fine-tuned model produces floor plans that are structurally more similar to the ground truth images, demonstrating enhanced accuracy in capturing the architectural details specified in the prompts. The results demonstrated that this fine-tuned model successfully generated

Tab. 1. SSIM scores for base Diffusion model and fine-tuned Diffusion model.

No.	Model	SSIM score
1	Base Model	0.3191
2	Fine-Tuned Model (Prompt 1)	0.3426
3	Fine-Tuned Model (Prompt 2)	0.4254

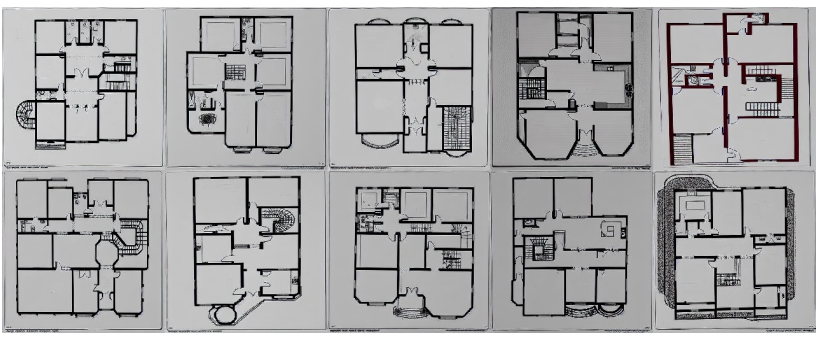


Fig. 3. Generated images from Prompt 1: "floor plan of house having two living rooms, one bedrooms, three bathroom, two kitchen, one garage, three store, one entrance."

floor plans that met the specified criteria, confirming the efficacy of our implementation strategy using Hugging Face's LoRA framework.

The second tested prompt was more popular and familiar in most home designs:

- **Prompt 2**: "floor plan of house having one living room, two bedrooms, one bathroom, one kitchen, one hall, one entrance."

The proposed model has generated 10 designs that matches the mentioned prompt and give more varieties for the designer from this prompt. The SSIM Scores of Diffusion model versus fine-tuned Diffusion model for the mentioned prompt are summarized in Table 1, number 3, while 10 generated images from the fine tuned Stable Diffusion model are presented in Figure 4.

The improvement can be attributed to this approach of constraining the input prompts during fine-tuning. By limiting prompts to a static structure with only key numerical parameters varying—such as the number of bedrooms or dining rooms—we guided the model to focus on these critical elements. This constraint reduces unnecessary creativity, enabling the model to generate floor plans that more precisely reflect the specified requirements.

Although the numerical increase in SSIM is modest, it signifies meaningful enhancements in the context of architectural design, where precision is paramount. Even small

Fig. 4. Generated images from Prompt 2: "floor plan of house having one living room, two bedrooms, one bathroom, one kitchen, one hall, one entrance."

improvements in SSIM correspond to better alignment of walls, rooms, and spatial relationships, which are crucial for the practical usability of floor plans. The fine-tuned model's outputs exhibit greater fidelity to the intended layouts, suggesting that our method effectively bridges the gap between creative image generation and the need for exactitude in architectural applications. These results validate that constraining the model's input prompts and fine-tuning it on specialized data can improve its performance in generating accurate floor plans. The fine-tuned model demonstrates a better understanding of the correlation between the specified numerical parameters and the spatial configurations required, making it a more reliable tool for architectural design tasks that demand high levels of precision.

8. Discussion

To assess the performance of Stable Diffusion v1.5 in producing architectural design outputs, two floor plan prompts were evaluated using expert judgment based on design clarity, feature correctness, and alignment with the prompt. A panel of 3 evaluators with experience in architecture, interior design, and AI image generation rated the outputs using the following criteria:

- **Relevance to Prompt**: Correct inclusion and quantity of specified rooms and features.
- **Layout Clarity**: Logical and readable floor plan arrangement.
- **Design Aesthetics**: Overall visual structure and neatness.
- **Spatial Realism**: Realistic spatial proportions and plausible connectivity between rooms.
- **Prompt Sensitivity**: Ability of the model to reflect prompt changes between two variants.

Tab. 2. Expert Evaluation of Generated Floor Plans

Prompt	Relevance	Clarity	Aesthetics	Spatial Realism	Average
Prompt 1	4	3	4	3	3.8
Prompt 2	5	4	4	4	4.4

Table 2 displays the results.

Prompt 1
The model captured most of the required rooms but struggled with accurate quantity differentiation, especially for "three stores" and "two kitchens", which were either combined or misrepresented. The overall spatial layout lacked architectural realism (e.g., bathrooms sometimes placed without adjacent bedrooms or hall access), although room labeling was reasonably intuitive.

Prompt 2
The model performed better with this simpler prompt. All rooms were represented clearly, and the spatial layout was more plausible and visually coherent. Room positioning followed a logical flow, and the floor plan adhered closely to modern residential design principles. The evaluation indicates that Stable Diffusion v1.5 is effective for generating conceptual and visually descriptive floor plans from textual prompts, especially when the prompts are concise and moderately complex. However, for prompts with a high number of room types or quantities, the model's spatial reasoning and ability to differentiate repeating elements (e.g., multiple stores or kitchens) become more limited. These findings suggest the model is well-suited for early-stage ideation and visual storytelling, but not for technical architectural planning.

9. Conclusion

This study demonstrated the effective fine-tuning of Stable Diffusion v1.5 for the generation of precise 2D floor plans. By constraining the model with structured prompts that varied only in specific numerical parameters, we guided it to focus on accuracy and the nuanced spatial configurations essential in architectural designs. This approach successfully reduced unnecessary creativity inherent in diffusion models, resulting in outputs that more closely adhered to the specified requirements. The improved performance, reflected in higher SSIM scores compared to the base model, highlights the potential of combining prompt engineering with fine-tuning to adapt generative models for tasks demanding exactitude. Our findings indicate that diffusion models can be tailored to produce functionally accurate and detailed images in domains where precision is

paramount, expanding their applicability beyond creative image synthesis to practical, precision-oriented applications.

Acknowledgement

This work was supported by the Higher Institute of Computer Science and Information Systems, Culture and Science City, October 6 University, Egypt.

References

[1] V. Azizi, M. Usman, H. Zhou, P. Faloutsos, and M. Kapadia. Graph-based generative representation learning of semantically and behaviorally augmented floorplans. *The Visual Computer* 38(8):2785–2800, 2022. doi:10.1007/s00371-021-02155-w.

[2] S. K. Baduge, S. Thilakarathna, J. S. Perera, M. Arashpour, P. Sharafi, et al. Artificial intelligence and smart vision for building and construction 4.0: Machine and deep learning methods and applications. *Automation in Construction* 141:104440, 2022. doi:10.1016/j.autcon.2022.104440.

[3] T. Berrada, P. Astolfi, M. Hall, R. Askari-Hemmat, Y. Benchetrit, et al. On improved conditioning mechanisms and pre-training strategies for diffusion models. In: *Advances in Neural Information Processing Systems*, vol. 37, pp. 13321–13348. Curran Associates, Inc., 2024. https://proceedings.neurips.cc/paper_files/paper/2024/hash/18023809c155d6bbed27e443043cdebf-Abstract-Conference.html.

[4] D. Bolya and J. Hoffman. Token merging for fast stable diffusion. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4599–4603, 2023. doi:10.1109/CVPRW59228.2023.00484.

[5] L.-P. de las Heras, S. Ahmed, M. Liwicki, E. Valveny, and G. Sánchez. Statistical segmentation and structural recognition for floor plan interpretation: Notation invariant structural element recognition. *International Journal on Document Analysis and Recognition (IJDAR)* 17(3):221–237, 2014. doi:10.1007/s10032-013-0215-2.

[6] S. Dodge, J. Xu, and B. Stenger. Parsing floor plan images. In: *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pp. 358–361, 2017. doi:10.23919/MVA.2017.7986875.

[7] J. Encarnação, R. Lindner, and E. G. Schlechtendahl. *Computer Aided Design: Fundamentals and System Architectures*. 2nd edn. Springer-Verlag, Berlin, Heidelberg, 2012. doi:10.1007/978-3-642-84054-8.

[8] W. Feng, W. Zhu, T.-J. Fu, V. Jampani, A. Akula, et al. LayoutGPT: Compositional visual planning and generation with large language models. In: *Advances in Neural Information Processing Systems*, vol. 36, pp. 18225–18250. Curran Associates, Inc., 2023. https://proceedings.neurips.cc/paper_files/paper/2023/hash/3a7f9e485845dac27423375c934cb4db-Abstract.html.

[9] G. Goodman. *A machine learning approach to artificial floorplan generation*. Master's thesis, University of Kentucky, 2019. https://uknowledge.uky.edu/cs_etds/89.

[10] Y. Gu, Y. Huang, W. Liao, and X. Lu. Intelligent design of shear wall layout based on diffusion models. *Computer-Aided Civil and Infrastructure Engineering* 39(23):3610–3625, 2024. doi:10.1111/mice.13236.

[11] L. Hahkio. *Generation of realistic floorplans using diffusion-based models*. Master's thesis, University of Helsinki, 2023. https://urn.fi/URN:NBN:fi:aalto-202310156363.

[12] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, et al. LoRA. Hugging Face. https://huggingface.co/docs/diffusers/main/en/training/lora.

[13] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, et al. LoRA: Low-rank adaptation of large language models. arXiv, arXiv.2106.09685, 2021. doi:10.48550/arXiv.2106.09685.

[14] M. Jain, A. Sanyal, S. Goyal, C. Chattopadhyay, and G. Bhatnagar. Automatic rendering of building floor plan images from textual descriptions in English. arXiv, arXiv.1811.11938, 2018. doi:10.48550/arXiv.1811.11938.

[15] M. Keshavarzi and M. Rahmani-Asl. GenFloor: Interactive generative space layout system via encoded tree graphs. *Frontiers of Architectural Research* 10(4):771–786, 2021. doi:10.1016/j.foar.2021.07.003.

[16] C. Kupferschmidt, A. D. Binns, K. L. Kupferschmidt, and G. W. Taylor. Stable rivers: A case study in the application of text-to-image generative models for Earth sciences. *Earth Surface Processes and Landforms* 49(13):4213–4232, 2024. doi:10.1002/esp.5961.

[17] S. Leng, Y. Zhou, M. H. Dupty, W. S. Lee, S. C. Joyce, et al. Tell2Design: A dataset for language-guided floor plan generation. arXiv, arXiv.2311.15941, 2023. doi:10.48550/arXiv.2311.15941.

[18] C. Liu, J. Wu, P. Kohli, and Y. Furukawa. Raster-to-vector: Revisiting floorplan transformation. In: *Proc. 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2214–2222, 2017. doi:10.1109/ICCV.2017.241.

[19] Z. H. Luo, L. Lara, G. Y. Luo, F. Golemo, C. Beckham, et al. DStruct2Design: Data and benchmarks for data structure driven generative floor plan design. arXiv, arXiv.2407.15723, 2024. doi:10.48550/arXiv.2407.15723.

[20] Z. Ma, Y. Zhang, G. Jia, L. Zhao, Y. Ma, et al. Efficient diffusion models: A comprehensive survey from principles to practices. arXiv, arXiv.2410.11795, 2024. doi:10.48550/arXiv.2410.11795.

[21] N. Nauata, S. Hosseini, K.-H. Chang, H. Chu, C.-Y. Cheng, et al. House-GAN++: Generative adversarial layout refinement networks. arXiv, arXiv.2103.02574, 2021. doi:10.48550/arXiv.2103.02574.

[22] H. T. Nguyen, Y. Chen, V. Voleti, V. Jampani, and H. Jiang. HouseCrafter: Lifting floorplans to 3D scenes with 2D diffusion model. arXiv, arXiv.2406.20077, 2024. doi:10.48550/arXiv.2406.20077.

[23] M. Opiela, M. Hrehová, and F. Galčík. Map model extraction from image floor plans. In: *Proceedings of the Work-in-Progress Papers at the 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN-WiP 2023)*, vol. 3581 of *CEUR-WS.org: IAOA Series*. Nuremberg, Germany, 2023. https://ceur-ws.org/Vol-3581/194_WiP.pdf.

[24] L. Papa, L. Faiella, L. Corvitto, L. Maiano, and I. Amerini. On the use of stable diffusion for creating realistic faces: from generation to detection. In: *2023 11th International Workshop on Biometrics and Forensics (IWBF)*, pp. 1–6, 2023. doi:10.1109/IWBF57495.2023.10156981.

[25] P. N. Pizarro, N. Hitschfeld, I. Sipiran, and J. M. Saavedra. Automatic floor plan analysis and recognition. *Automation in Construction* 140:104348, 2022. doi:10.1016/j.autcon.2022.104348.

[26] J. Ploennigs and M. Berger. Automating computational design with generative AI. arXiv, arXiv.2307.02511, 2024. doi:10.48550/arXiv.2307.02511.

[27] A. Razzhigaev, A. Shakhmatov, A. Maltseva, V. Arkhipkin, I. Pavlov, et al. Kandinsky: an improved text-to-image synthesis with image prior and latent diffusion. arXiv, arXiv.2310.03502, 2023. doi:10.48550/arXiv.2310.03502.

[28] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In: *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022)*, pp. 10684–10695, 2022. doi:10.1109/CVPR52688.2022.01042.

[29] R. Rombach and P. Esser. SD v1.5. Hugging Face, 2024. https://huggingface.co/stable-diffusion-v1-5.

[30] M. A. Shabani, S. Hosseini, and Y. Furukawa. HouseDiffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. In: *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2023)*, pp. 5466–5475, 2023. doi:10.1109/CVPR52729.2023.00529.

[31] S. Srivastava, N. Maheshwari, and K. S. Rajan. Towards generating semantically-rich IndoorGML data from architectural plans. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 42:591–595, 2018. doi:10.5194/isprs-archives-XLII-4-591-2018.

[32] L. Wang, J. Liu, Y. Zeng, G. Cheng, H. Hu, et al. Automated building layout generation using deep learning and graph algorithms. *Automation in Construction* 154:105036, 2023. doi:10.1016/j.autcon.2023.105036.

[33] X.-Y. Wang, Y. Yang, and K. Zhang. Customization and generation of floor plans based on graph transformations. *Automation in Construction* 94:405–416, 2018. doi:110.1016/j.autcon.2018.07.017.

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13(4):600–612, 2004. doi:10.1109/TIP.2003.819861.

[35] R. E. Weber, C. Mueller, and C. Reinhart. Automated floorplan generation in architectural design: A review of methods and applications. *Automation in Construction* 140:104385, 2022. doi:10.1016/j.autcon.2022.104385.

[36] W. Wu, L. Fan, L. Liu, and P. Wonka. MIQP-based layout design for building interiors. *ACM Transactions on Graphics (SIGGRAPH Asia)* 38(6):1–12, 2019.

[37] W. Wu, X.-M. Fu, R. Tang, Y. Wang, Y.-H. Qi, et al. Data-driven interior plan generation for residential buildings. Wenming Wu's Homepage, 2019. https://wutomwu.github.io/particulars.html?id=1. RPLAN project page.

[38] J. Yang, Z. Cheng, Y. Duan, P. Ji, and H. Li. ConsistNet: Enforcing 3D consistency for multi-view images diffusion. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7079–7088, 2024. doi:10.1109/CVPR52733.2024.00676.

[39] Z. Zong, Z. Zhan, and G. Tan. HouseLLM: LLM-assisted two-phase text-to-floorplan generation. arXiv, arXiv.2411.12279v3, 2024. doi:10.48550/arXiv.2411.12279.