SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

A Robust Explainable Quantum Ensemble Deep Learning Architecture with Optimized Feature Selection For 3d Face Recognition

Aswathy.R[1], Dr. A. Sherin[2]

[1]Research Scholar, Department of Computer Science, Nehru Arts and Science College (NASC), Thirumalayampalayam, Coimbatore - 641 105, Tamil Nadu. Email id: aswathyravi998@gmail.com
[2]Assistant Professor, Department of Computer Science, Nehru Arts and Science College (NASC), Thirumalayampalayam, Coimbatore - 641 105, Tamil Nadu. Email id: sherinfaizalrahiman@gmail.com

Abstract
In computer vision, face recognition is regarded as one of the most significant applications. and AI owing to its broad usage in security systems, surveillance systems, biometric recognition systems, and even smart human-computer interactions. However, factors such as changes in illumination, facial expression, pose, occlusion, and image quality still create huge barriers in performing accurate face recognition. In order to overcome the above issues, this paper proposes an Explainable Stacking Ensemble Quantum Deep Learning (ESEQDL) along with Intelligent Feature Selection scheme for face recognition using graph neural network (GNN) landmarks that create three-dimensional (3D) face meshes from two-dimensional (2D) facial images from different sources. Preprocessing makes use of Restormer to improve image quality, followed by segmentation through the deep learning approach using the SegFormer framework that helps in separating the foreground from the background and thus making the detection more accurate and noise-free. Feature Extraction in the Vision Transformer (ViT) architecture. Additionally, a feature selection method is implemented to help select the most important facial features and exclude unnecessary information. For the purpose of this study, Optimized Recursive Feature Elimination (ORFE) technique is used for feature selection. The above proposed framework uses several deep learning algorithms into one stacking ensemble quantum algorithm for better feature extraction and performance of classification tasks. In this study, the different types of explainable ensemble quantum deep neural networks used include SwinFace, S-ViT, InceptionNet QCNN, Grad CAM and SHAP for feature extraction of diverse and discriminating facial representations from the input face image. The meta-classifier uses the output of all different basis models. This approach increases the resistance and generalization capacity of the face recognition system through the use of the strengths of multiple architectures of deep learning. Evaluation of the proposed approach is done on several face images datasets in various environmental conditions. The results of the experiment readily demonstrate that the stacking standard deep quantum learning approach coupled with intelligent feature selection performs better than the traditional systems.

Keywords: Facial images, graph neural network (GNN), Restormer, SegFormer architectures, Vision Transformer (ViT), SwinFace, S-ViT, InceptionNet QCNN, Grad CAM and SHAP.

1. Introduction

Facial recognition has developed to become one of the most widely employed biometrics due to its non-intrusive nature, convenience, and vast applications in security, surveillance, access control, forensic science, and Human Computer Interaction (HCI) systems [1]. Recent advancements of Artificial Intelligence (AI), machine learning (ML), and deep learning (DL), facial recognition systems have witnessed impressive gains in recognition accuracy and reliability [2]. It is still difficult to recognize individuals in case of lighting variations, change in poses, expression changes, occlusion, age factor, low resolution, and many other problems. The task of identifying individuals based on their faces having almost identical facial features is known as identical face recognition, which is one branch of biometrics identification [3]. However, unlike traditional face recognition approaches, where the main task is distinguishing the identity of different faces, identical face recognition poses a much greater problem because of the low intra-class differences among faces [4]. This becomes especially clear when dealing with twins, brothers, sisters, or lookalikes, or large databases of similar faces.

Conventional methods for face recognition are mainly based on handcrafted feature extraction approaches, which usually do not represent the complexity of facial structures and can be affected by changes in environments [5]. However, recent DL models such as Deep Belief Networks (DBNs), Vision Transformers (ViTs), Convolutional Neural Networks (CNNs), and Residual Networks (ResNets) have been shown to be more effective in learning discriminative facial features using large amounts of data [6]. DL models individually might have drawbacks such as overfitting, even with all these advancements, high

computational cost, and poor generalization performance when faced with varying facial image distributions [7]. In order to counteract these problems, ensemble learning has been given great attention due to its potential in enhancing classification efficiency through combining various learning models' strengths. As there are different ensemble methods available, stacking ensemble learning method combines the predictions by the base learners using meta-learner to provide improved accuracy, stability, and generalization performance [8]. In contrast, feature extraction performance and selection of face features play an important role in the success of stacking ensembles.

Feature selection is an important aspect of face recognition as it facilitates the reduction of data dimensionality, removes redundant data, and increases computational efficiency [9]. Smart algorithms for feature selection help in determining the best possible features of faces, which not only increase accuracy but also decrease computational time. There has been a lot of attention recently on quantum computing approaches due to their ability to work in difficult optimization problems. The field of quantum DL integrates the capabilities of quantum algorithms and deep learning algorithms, resulting in better feature extraction, fast convergence, and more advanced learning capabilities than traditional deep learning techniques [10]. Besides obtaining high recognition rates, modern biometric systems need to be transparent and understandable to be trustworthy and reliable in practical applications. Explanation of decision making in DL models can be performed by using the techniques of Explainable Artificial Intelligence (XAI) through identification of those areas on the face which are important for recognition results [11]. The application of explanation techniques increases transparency of the model and helps to validate the system.

Based on these challenges, the research paper suggests the use of Explainable Stacking Ensemble Quantum Deep Learning and Intelligent Feature Selection Approach for Face Recognition. The suggested approach involves the implementation of several quantum-based deep learning algorithms in a stacking ensemble structure along with using intelligent methods of feature selection. Additionally, methods for explainable AI are used to improve the understandability of decision-making. With the use of intelligent feature optimization, quantum-based learning, ensemble classifier, and explainable methods, the proposed approach aims to provide recognition performance which is better in terms of accuracy, robustness, computation cost, and transparency.

The rest of the paper has the following structure: An overview of some of the most recent methods for facial image identification is given in Section 2. Section 3 explains the proposed approach's procedure. Section 4 discusses the results and comments. Section 5 concludes and discusses future tasks.

2. Literature Review

Section 2 examines some of the most existing techniques for employing sophisticated machine learning algorithms to recognize faces in images.

Author and year	Methods	Advantages	Disadvantages
Wang et al [2022]	deep learning	Enables real-time identification and verification with minimal human intervention.	Advanced deep learning-based recognition systems often require significant processing power and memory resources.
Alagarsamy et al [2020]	Tensor Flow framework	Recognizes individuals without physical contact, making it more convenient than fingerprint or iris scanning.	Recognition performance may vary across different demographic groups if training datasets are not sufficiently diverse.
Reddy et al [2026]	Convolutional neural network (CNN)	Eliminates the need for passwords, ID cards, or tokens, simplifying authentication procedures.	Systems may be vulnerable to presentation attacks using photographs, videos, or deepfake images if anti-spoofing measures are not implemented.
Hangaragi et al [2023]	deep neural network	Can efficiently manage and search large databases containing millions of facial images.	Changes in lighting conditions can significantly affect recognition accuracy.
Sivanagireddy et al [2024]	DenseNet 169	Allows recognition from images and video streams captured by cameras at a distance.	Different head orientations and viewing angles may reduce identification performance.

Sandhya et al [2023]	Deep Neural Network (DNN)	Provides reliable biometric-based identity verification for access control	Smiling, frowning, and other facial expressions can alter facial characteristics.
Gupta et al [2022]	Flickr-Faces-HQ Dataset (FFHQ)	Enables real-time identification and verification with minimal human intervention.	Continuous monitoring and data collection may raise ethical and privacy issues.
Tela et al [2022]	Convolutional neural network (CNN)	accuracy of 76.19%	Image quality, camera resolution, background clutter, and motion blur can negatively impact recognition accuracy.
Chaves et al [2020]	deep-learning-based face detectors	Can efficiently manage and search large databases containing millions of facial images.	Systems may be vulnerable to presentation attacks using photographs, videos, or deepfake images
Nida et al [2021]	VGG models	Assists in criminal identification, missing person detection, and public safety monitoring.	Masks, sunglasses, hats, and facial coverings can obstruct important facial features.
Mohamed et al [2020]	Convolution Neural Networks (CNN)	proposed method achieves 99.67% accuracy.	Facial appearance changes over time, making long-term recognition more difficult.

Despite these advancements, existing methods still face several challenges. Traditional methods exhibit limited adaptability to complex real-world scenarios, large annotated datasets and substantial computing resources are often needed for DL models. Furthermore, many approaches lack interpretability, making the logic underlying recognition decisions hard to comprehend. Recognition accuracy may also decrease when dealing with identical faces, twins, occlusions, aging effects, low-resolution images, and unconstrained environments. Therefore, there remains a need for more robust, explainable, and computationally efficient face recognition frameworks that integrate intelligent feature selection, ensemble learning, quantum-inspired optimization, and explainable artificial intelligence techniques.

3. Proposed Methodology

This study proposes an Explainable stacking ensemble quantum deep learning and intelligent feature selection method for reliable face recognition that uses graph neural network (GNN) landmarks to create three-dimensional (3D) face meshes from two-dimensional (2D) facial images obtained from divezrse origins. The preprocessing pipeline incorporates Restormer to enhance image quality, the foreground (face) is then successfully separated from the background using deep learning-based segmentation using Mask R-CNN architectures, which reduces noise and increases detection accuracy. Feature extraction are extracted by using the Vision Transformer (ViT). In addition, an intelligent feature selection mechanism is incorporated to identify the most relevant facial features and eliminate redundant or irrelevant information. In this work the Optimized Recursive Feature Elimination (ORFE) based feature selection process reduces computational complexity, improves model efficiency, and enhances recognition accuracy. The proposed framework integrates multiple deep learning models within a stacking ensemble quantum architecture to improve feature learning and classification performance. In this work the different explainable ensemble quantum deep neural networks are employed such as Resnet, DenseNet, InceptionNet QCNN, Grad CAM and SHAP to extract diverse and discriminative facial representations from input face images. A meta-classifier is used to aggregate the results of various underlying models, improving prediction accuracy and lowering classification error rates. The ensemble framework enhances the face recognition system's resiliency and generalization ability by using the advantages of many deep learning architectures. The proposed system is evaluated using standard face image datasets under different environmental conditions.

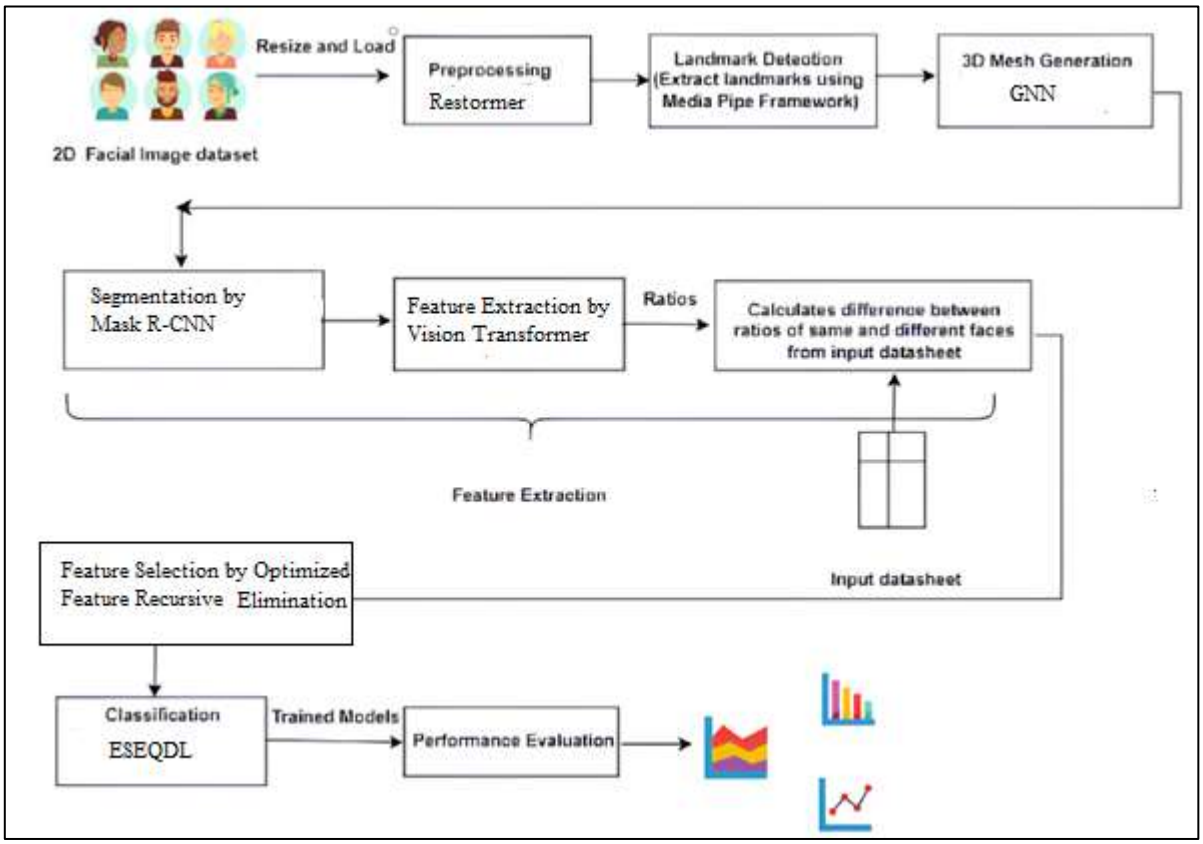

Figure 1. The proposed methodology's overall procedure

3.1. Data Collection

The Labelled Faces in the Wild (LFW) dataset is a widely used benchmark for evaluating face recognition under unconstrained real-world conditions. Developed by researchers at the University of Massachusetts Amherst, it includes 5,749 individual 13,233 facial images, collected from the web and detected and aligned using the Viola–Jones face detector. Among these subjects, 1,680 individuals have at least two distinct images in the dataset. LFW provides four original image sets along with three forms of aligned images. Among these variants, deep-funneled images have generally demonstrated better performance in face-verification algorithms than the other alignment types.

3.2. Pre-processing

Two-dimensional facial images were collected to obtain clear and standardized samples for face detection and normalization. In the proposed method, each input image undergoes preprocessing steps, including cropping and resizing, to construct the facial image dataset. An Adaptive Median Filter (AMF) is then applied to suppress unwanted noise in the FERC images. This filtering technique enhances image quality and supports more reliable model performance, particularly when the input images contain substantial noise. AMFs employ spatial algorithms for identifying image pixels that are noisy [23]. They compare pixels with their neighbours for these determinations where both comparison criteria and neighbourhood sizes can be varied [24]. Noisy pixels can be defined as pixels that differ from majority of their neighbours and are not physically aligned with their comparable pixels. Median pixel values of nearby pixels that pass noise labelling tests are then used to replace noisy pixels. Therefore, they are applied to lower noises in FERC images and enhance image qualities.

- **Restormer**

Restormer is a transformer-based neural network that has been designed for image restoration purposes. Global dependencies are not captured by traditional convolutional neural networks in images due to their local receptive field nature, attention-based mechanisms enable restormer to restore image details.

Restormer is a modern model for image restoration tasks that use a transformer architecture. Restormer improves upon the conventional transformer paradigm by using an encoder-decoder structure by refining the components of self-attention and feed-forward networks [25]. During training, a progressive learning scheme is used to boost the performance of the image restoration process. In contrast to windowed transformer models, Restormer is able to capture multi-scale features from both local and global scales without breaking the image into small windows. This architecture maintains contextual information, thus

ensuring the effectiveness of image restoration process. This Restormer network architecture mainly has two components: Gated Dconv Feed-Forward Network (GDFN) and Multi-Dconv Head Transposed Attention (MDTA). Because it generates transposed attention maps based on cross-covariance of feature channels, the MDTA serves as a substitute for the traditional Multi-Head Self-Attention. The GDFN model builds on the conventional feed-forward network by using gated operations and depth-wise convolution, which enhance feature representation and improve feature extraction efficiency. Restormer also uses progressive learning in the training process, whereby the network is initially trained with small patches of images that gradually increase in size during training. The proposed approach will make it easier for the model to identify the image's overall characteristics and context. Training progressively has also contributed to good performance when tested. With efficient global dependencies modeling and progressive learning, Restormer has provided an effective solution to the problem of image denoising.

3.3. Face Detection

The method of identifying faces in an image is known as face detection through the knowledge gained about facial features. It could be difficult due to the following reasons: detection accuracy is affected by background, lighting, motion of the object being detected, pose of the face, and differences in facial expressions. Efficient tools for face detection and face landmark localization have been made available in the MediaPipe library [26]. In the proposed framework, the MediaPipe Face Detection model is used to detect faces.

3.3.1. Landmark Detection

The technique allows for precise and reliable localization of facial landmarks based on the face images. The suggested technique automatically detects and extracts all necessary facial features. In this study, 468 facial landmarks have been detected by employing the MediaPipe framework in lieu of the standard 68 points representation, providing a more detailed facial geometry for improved accuracy. The MediaPipe framework can simultaneously detect multiple faces and represent their landmarks in three-dimensional space. Its face geometry solution supports real-time landmark estimation, including on resource-constrained devices such as mobile platforms.

3.3.2. 3D Face Mesh Generation using Graph Neural Network (GNN)

The facial landmarks extracted from the two-dimensional input image are mapped onto a three-dimensional representation to construct a facial mesh. For each detected face, the Face Mesh API generates a 3D mesh comprising 468 landmark points, together with the corresponding edges and triangular connections. In the proposed method, a Graph Neural Network (GNN) is incorporated to support 3D facial mesh generation. GNNs have recently attracted considerable attention in a variety of application sectors because of their ability to model relationships between interconnected data elements. Graph Neural Networks (GNNs) have demonstrated strong potential for graph-based analysis by effectively learning the relationships and dependencies among interconnected nodes. A GNN is a neural network architecture that operates directly on graph-structured data [27]. One of its common applications is node classification, where each node has a class label connected to it, and the model learns to infer the labels of nodes for which the ground-truth class is unavailable.

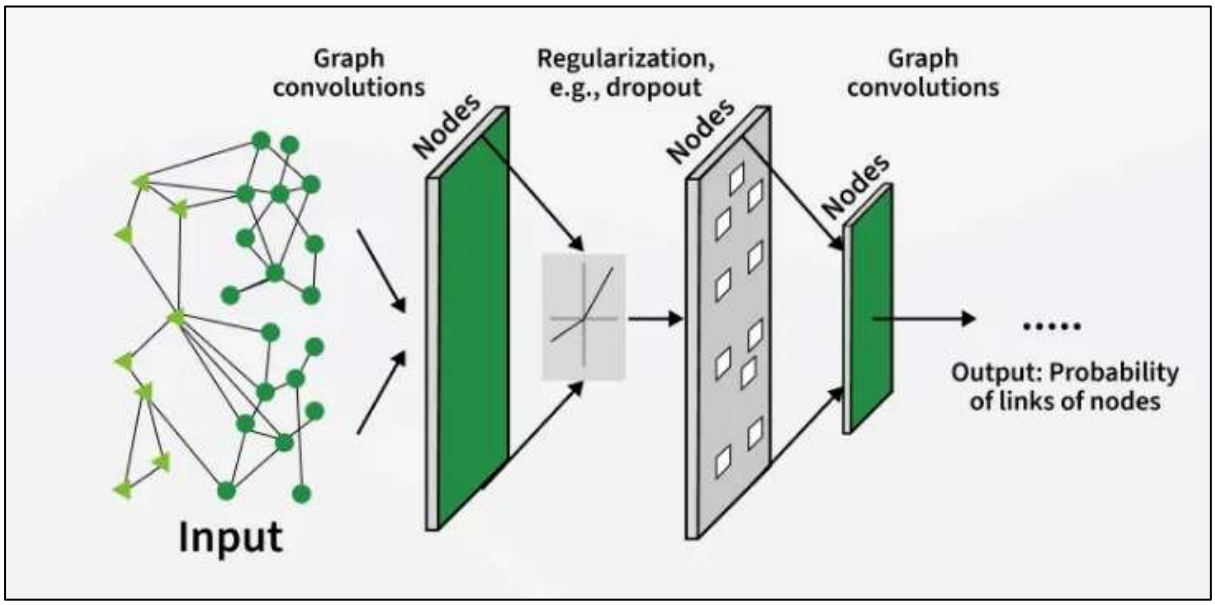

Fig. 2. The architecture of graph neural network-based model

Figure 2 illustrates the architecture of the proposed framework. An estimate is provided of the probability that a patient p will have a disease m by evaluating the similarity between their respective embeddings. First, a graph encoder is introduced to transform each node in an arbitrary graph into a compact, low-dimensional embedding $z \in Rd$, which is subsequently used for disease prediction. Recent developments in Graph Convolutional Networks (GCNs), node embeddings are generated by aggregating the feature information of directly connected neighboring nodes. The aggregation process applies a common transformation function uniformly across the graph while considering into account each node's first-order neighborhood, enabling the learned representation to capture local structural relationships. Accordingly, each graph node has its own local computational neighborhood, while the same set of learned parameters is shared across nodes to control information aggregation and propagation. This parameter-sharing mechanism promotes efficient utilization of structural information throughout the graph and enables the model to generate embeddings for previously unseen nodes, such as a newly added patient in the dataset.

Initially, to be concise, uniformly describe a disease, symptom, or patient node as $v \in G$ for a graph $G = \{C,P\}$. The embedding $\mathbf{h}l\ v$ of node v is consequently determined at the l-th information transmission layer as follows:

$$h_{N(v)}^{l} = AGGREGATE\left(\left\{h_{v'}^{l-1}, \forall v' \in N(v)\right\}\right) \quad (1)$$

$$h_{v}^{l} = \sigma\left(W^{l}.\left[h_{v}^{l-1};\ h_{N(v)}^{l}\right]\right) \quad (2)$$

where W^l is the weight matrix to be learned at the l-th layer and h_v^{l-1} is the embedding of node v at the previous layer, and L is the total layer size. The concatenation of two vectors is represented by [.], and the collection of equally sampled neighbor nodes of v is denoted by $N(v)$. It should be noted that the node embedding h0 $v \in$ Rd for l = 0 is initialized using either side information from the data (depending on availability) or randmonized values. For example, h0v will be started as a real-valued dense feature vector provided a node with the available demographics and face picture profiles in the data. The observed value of a feature dimension (such as age) is indicated by each number in $h0v$. At the (l−1)-th layer, the aggregation function aggregates the embeddings of node v's neighbors, produces the synergic representation hl $N(v)$. The aggregator may be mean, max pooling, RNNs, etc., and σ is a non-linear activation function (such as tanh). Use $mean(\cdot)$ by default to aggregate information in our model. Before arriving at the final layer L's embedding for every node, we next do a normalization step:

$$h_v = \frac{h_v^L}{\|h_v^L\|_2}, \qquad \forall_v \in g \quad (3)$$

In the proposed method, the representations of the patient record graph P and the medical concept graph C both have symptom nodes p are learned within a common embedding space. Thus, the representation of a given symptom remains consistent across both graphs, providing a meaningful link between the face-image information and the corresponding nodes in the two graph structures. To further establish contextual alignment, all node embeddings are projected into a shared space and then processed using a nonlinear activation function:

$$z_v = \sigma(W h_v), \qquad \forall_v \in g \quad (4)$$

where zv is node v's final embedding and W is the learnable projection weight.

3.4. Segmentation using SegFormer architectures

Each instance (presence of each item contained in the image) is identified using segmentation, which also colors them with distinct pixels. In essence, it creates the segmentation mask for each object in the image by classifying each pixel's position [28]. Because it maintains the items' safety while identifying them, this method provides additional details about the things in the image.

SegFormer is a state-of-the-art transformer network that is utilized for semantic segmentation by employing a light-weighted multilayer perceptron (MLP) decoder along with a Transformer encoder to perform precise and efficient segmentation of images. In contrast to traditional segmentation networks that are based mainly on convolution operations, SegFormer keeps track of both the local and global context information using the self-attention mechanism. SegFormer has been used for the task of face image segmentation where the objective is to separate the facial region from the rest of the image and detect the important features of faces like the eyes, nose, mouth, eyebrows, hair, skin, etc. This is because SegFormer has an encoder-decoder architecture where the Multi-Scale Vision Transformer (MiT) encoder generates multi-scale feature representations for the input image and the MLP decoder fuses these features into a single segmentation mask. SegFormer architecture is illustrated in figure 3.

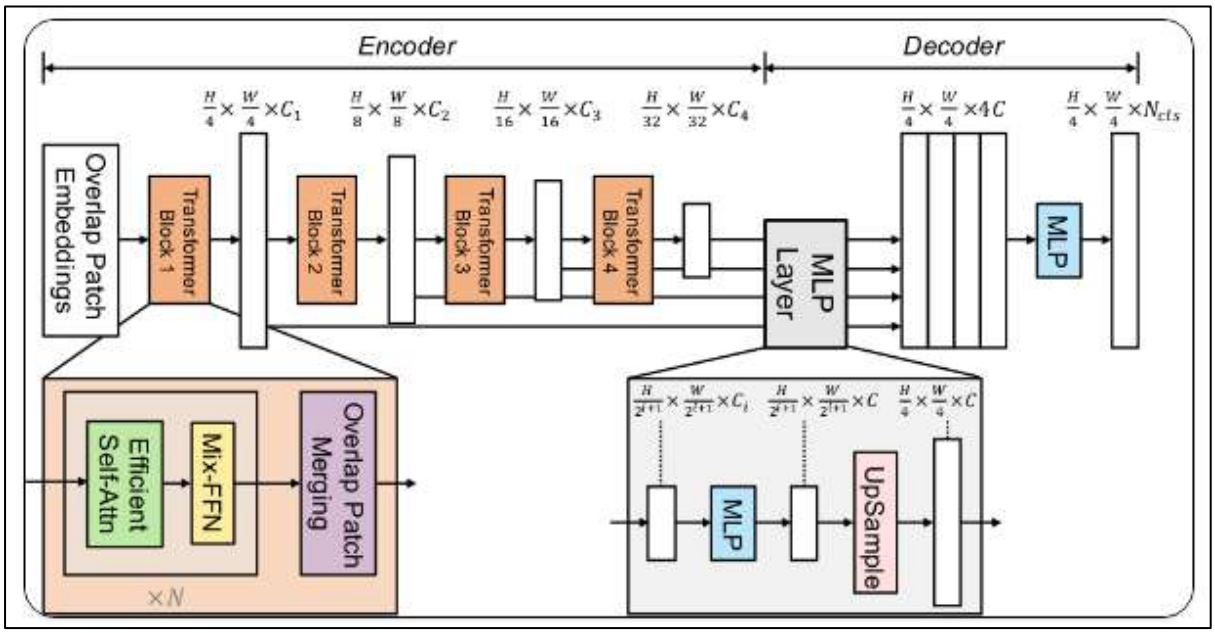

Figure.3. Architecture of SegFormer

The process starts with image pre-processing and embedding patches, whereby facial images are partitioned into patches and embedded into feature tokens. The next stage involves processing the feature tokens using the MiT encoder with several transformer layers in order to learn the hierarchical facial representations. The decoder merges multi-level features and generates segmentation maps at the pixel level that define the facial regions precisely. The segmentation masks that are generated can further be applied in various applications like face recognition, facial attribute analysis, expression recognition, and face biometrics. There are several strengths of SegFormer that make it an efficient algorithm for segmentation of face images: high accuracy of segmentation, resistance to illumination change, variations in pose, occlusions, and complicated background. In addition, it uses lightweight decoder that ensures reduction in computational complexity and memory consumption. This way, it becomes possible to use the SegFormer for real-time applications related to face processing.

3.5. Feature Extraction using Vision Transformer (ViT)

Vision Transformer is a deep learning model that can be used for Computer Vision and Deep Learning image processing tasks. It works on the concept of breaking down the image into smaller patches and then applying transformers to analyze its features [29]. The Vision Transformer algorithm performs image recognition by:

- Splitting images into patches
- Encoding patches into embeddings
- Learning global relationships using transformer attention
- Producing predictions using the CLS token representation

The ViT employs an attention-based mechanism to capture relationships among both spatially adjacent and distant image regions. To facilitate this process, the input image is partitioned into a sequence of smaller patches. This operation is conceptually similar to applying a convolutional kernel, where the resulting representation can be viewed as a four-dimensional tensor organized according to the batch dimension and the spatial row, column, and feature-depth dimensions. As a result, the image I $\in \mathbb{R}H\times W\times C$ is reshaped into PP $\in \mathbb{R}N\times P2\times C$, where H and W stand for the width and height of the image and the number of channels. N, on the other hand, denotes the quantity of patches identified as,

$$N = \frac{H\times W}{p^2} \qquad (5)$$

P is the patch size in this case. The patch is 6 by 6, and the image measures 78 by 78. The number of patches is computed using the image and patch size: $(H\times WP2)$=78×78(6)2=169. Following patch partition, a 2D matrix, PP, is produced from the raw image (I), The 1D embedding vector, $PPLinear\_Projection$, having a size of 64, is then linearly projected:

$$PP_{Linear\_Projection} = [[\, I_1^1\ I_1^2\ I_1^{64}]\ [I_2^1\ I_2^2\ I_2^{64}]\ [I_{169}^1\ I_{169}^2\ I_{169}^{64}\,] \qquad (6)$$

To the high computational cost of Transformer models, positional embeddings are assigned to image patches to retain their spatial information. The patches are organized into smaller groups and subsequently processed at higher image resolutions. The positional embedding E_{POS} is generated by combining sine and

cosine functions with different frequencies. For odd positions, the cosine function is applied, whereas even positions use the sine function for positional encoding. Here, pos denotes the patch position, (i) represents the embedding dimension, and (d) indicates the maximum length of the patch sequence. The resulting positional information is encoded across different positions using sinusoidal functions. Finally, the linearly projected patch representation is combined with its corresponding positional embedding to obtain the final embedded patch.

$$E_{POS} = \{sin\left(\frac{pos}{1000^{\frac{2t}{d}}}\right), \quad t\ is\ even\ cos\left(\frac{pos}{1000^{\frac{2t}{d}}}\right), \quad t\ is\ odd \tag{7}$$

$$EP = concatenate(PP_{Linear\ projection}, E_{POS}) \tag{8}$$

Following positional encoding and linear projection, the resulting patch embeddings are fed into the encoder. The encoder contains eight repeated blocks, each comprising normalization, multi-head self-attention (MSA), and MLP layers. The input (EP) is combined with the MLP for processing and normalization following transmitting the MSA output through a residual connection. The MLP incorporates a dense layer and dropout to enhance feature transformation and reduce overfitting. An input skip connection is concatenated with this attention output, increasing position effect since the following layer receives the original embedded patch. Following equations produce three embedding matrices, key K, query Q, and value V, which are used to calculate attention:

$$Query,\ Q = EP.WQ \tag{9}$$

$$Key,\ K = EP.WK \tag{10}$$

$$Value,\ V = EP.WV \tag{11}$$

Weight matrices and embedded patches are in W_Q, W_K, $W_V \in \mathbb{R}dmodel \times dk$ are all included. The MHA layer uses the single attention function referred to as "head," which is generated by equation (12), several times in parallel. In this instance, the dot-scaled product d_k prevents the attention value from increasing. This attention value shows the connection between two image patches by functioning as a scoring function. With four heads in the proposed design, the multi-head attention (MHA) is shown as follows:

$$Attention, (Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{12}$$

$$MHA = Attention, (Q, K, V)_{\times 4} \tag{13}$$

In the subsequent layers, the multi-layer perceptron uses a subsequent layer with Gaussian error linear unit (GELU) activation to form non-linearity. The Gaussian distribution's cumulative distribution is denoted by ϕ. $Transformer_{feature_{shape}}$, the layer's output, is finally extracted and then compressed using the following equations:

$$GELU(x) = xP(X, x) = x\phi(x) \tag{14}$$

$$Transformer_{feature_{shape}} = GELU(MHA) \tag{15}$$

$$Y_{transformer} = flatten(Transformer_{feature\ \_shape}) \tag{16}$$

The deep neural network classifier is then given the $Y_{transformer}$. Additionally, the ViT network's transformer feature extractor's ideal parameters. The validation dataset, the model's performance was taken into consideration while modifying the parameters, which were initially predicted based on the size and complexity of the dataset.

3.6. Feature Selection using Optimized Recursive Feature Elimination (ORFE)

The ORFE is an advanced technique for feature selection where the aim is to identify the most crucial and discriminating features and discard the unnecessary or redundant features. ORFE enhances the traditional Recursive Feature Elimination (RFE) technique through intelligent optimization techniques to enhance the effectiveness of feature selection, accuracy, and efficiency [30]. In the iterative process, the algorithm trains the learning model and discards the less important features recursively until an optimum feature set is achieved. Using only the most informative features helps ORFE to reduce dimensionality, avoid overfitting, speed up the learning process, and increase the accuracy of classification. The use of Optimized Recursive Feature Elimination makes it possible to choose the most important facial features for face recognition tasks.

This work introduces ORFE, an enhanced recursive feature elimination method that integrates RFE with a collision factor. Unlike conventional RFE, which eliminates features without considering their contribution to subsequent states, ORFE evaluates feature relevance at each stage. It can therefore retain weak or redundant features when their combination with other features contributes to improved model performance.

The discriminant function in a linearly separable problem is $g(x_i) = w.x_i + b$, where x_i represents training data $x_i \epsilon R^n$ i =1,...,m, b is a bias factor, and w is a weight vector. Slack variables ξi, which measure a data point's departure from the optimal hyperplane, may be introduced for linearly non-separable cases. The design of SVMs minimizes [8].

$$\phi(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \quad \xi_i \qquad (17)$$

subject to: $y_i(w.x_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0$

where y_i corresponds to the intended, $y_i = \{\pm 1\}, i = 1, \ldots, m$.

A dual problem is used to address the optimization issue:

$$W(\alpha) = \sum_{i=1}^{n} \quad \alpha_i - \frac{1}{2}\sum_{j=1}^{m} \quad y_i y_j \alpha_i \alpha_j (x_i.x_j) \qquad (18)$$

subject to: 1) $0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, m$

2) $\sum_{i=1}^{m} \quad \alpha_i y_i$

where the Lagrange coefficients are represented by α_i.

RFE retains the least significant characteristics, which are associated with the smallest weight value in w. A weight vector for linear SVMs is computed using:

$$w = \sum_{k \in SV} \quad y_k \alpha_k x_k \qquad (19)$$

- **Collision factor**

The collision factor is designed to use a convex function during the initial iterations, enabling particles to explore a broad search space and identify promising solutions. As the iterations progress, a concave function is adopted to gradually reduce the collision factor toward its minimum, thereby promoting local exploitation. This adaptive behavior supports algorithm convergence. Based on this principle, the collision factor is formulated using a cosine function, as given below (20):

$$K = \frac{cos((\frac{\pi}{G_{max})\times T)} + 2.5}{4} \qquad (20)$$

where the number of iterations is denoted by T. When G_{max} = 40 was set, a shifting curve with value K emerged. At initially, the curve of K is a convex function, though it develops into a concave function. Formula (20) produces formula (21) when the value K is replaced. Formula (21) is explained as follows:

$$v_{id} = \left(\frac{cos((\frac{\pi \times T}{G_{max}))\times 2.5}}{4}\right) \times [v_{id} + 2 \times rand() \times (p_{id} - x_{id}) + 2 \times Rand() \times (p_{gd} - x_{id})] \qquad (21)$$

RFE iteratively removes features and re-ranks the remaining ones by retraining the SVM at each stage. Although weak features are typically discarded, some may provide valuable information when combined with other features. Therefore, eliminating weak or redundant features without considering feature interactions can reduce classification performance. The significance of a potentially weak feature is determined by evaluating the classifier's performance after its removal, with reference to the corresponding w_i value. If eliminating the feature causes a reduction in classification performance, the feature is retained even when it has the lowest w_i value. The feature with the second-lowest w_i value is then evaluated in the same manner. This procedure continues until removing a feature does not reduce the classification performance. Since the features are initially ranked according to their w_i values, the proposed approach is termed ORFE.

3.7. Classification using explainable ensemble quantum deep neural networks

The proposed framework integrates multiple deep learning models within a stacking ensemble quantum architecture to improve feature learning and classification performance. In this work the different explainable ensemble quantum deep neural networks are employed such as SwinFace,S-ViT, InceptionNet QCNN, Grad CAM and SHAP to extract diverse and discriminative facial representations from input face images. A meta-classifier that improves prediction accuracy and reduces classification errors combines the outputs of various basis models. The ensemble framework enhances the face recognition system's resilience and ability for generalization by using the advantages of many deep learning architectures.

3.7.1. SwinFace

SwinFace is an advanced face recognition framework based on the Swin Transformer architecture, which leverages hierarchical vision transformers and shifted window self-attention mechanisms to learn highly discriminative facial representations [31]. Unlike typical Convolutional Neural Networks (CNNs), SwinFace is able to capture both the local information about faces as well as the global context, allowing for face recognition despite various challenges such as different face poses, illumination, facial expressions, occlusions, and low resolution faces. The core component of SwinFace is the Swin Transformer, which creates non-overlapping patches from an input face image and processes them through a hierarchical transformer structure. The shifted window attention mechanism efficiently models long-range dependencies while maintaining computational efficiency. By progressively merging image patches and extracting multi-scale features, SwinFace learns rich facial representations that are highly effective for identity recognition and verification tasks.

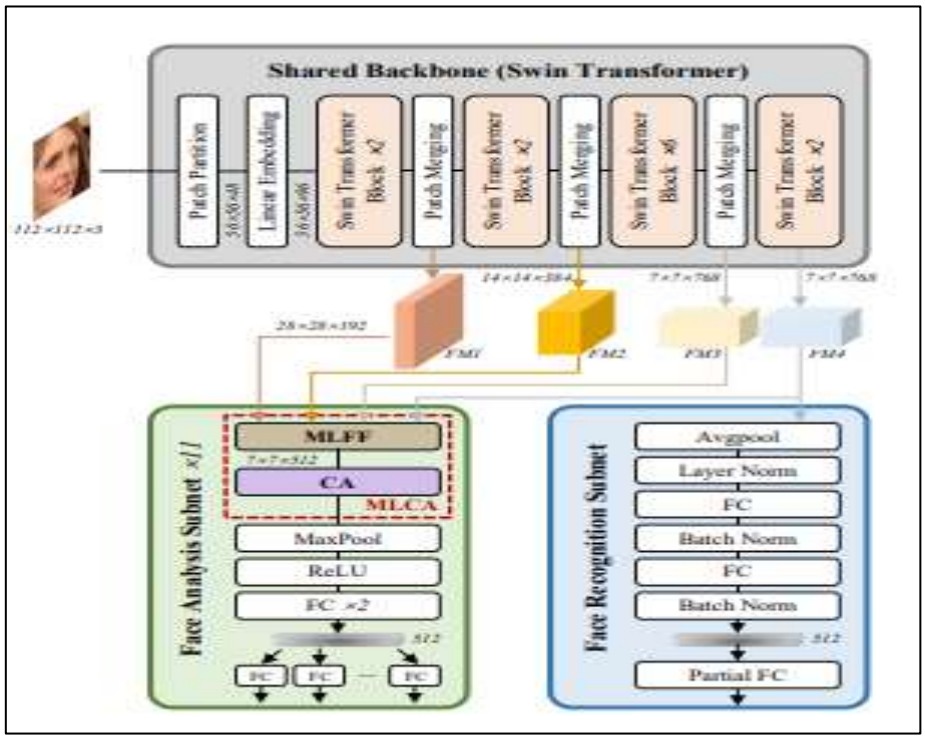

Fig. 4. SwinFace architecture

Fig. 4 shows an overview of the SwinFace architecture. In this study, shared feature maps at several levels are extracted using a single Swin Transformer. We further carry out multi-task learning using 11 face analysis subnets and a face recognition subnet based on shared feature maps.

1) Shared Backbone: The shared Swin Transformer backbone generates hierarchical feature representations. A cropped face image of size 112 × 112 × 3 is the patch partition module was initially utilized to divide the data into non-overlapping patches. With a patch size of 2 × 2, this produces 56 × 56 tokens with 48-dimensional features. Feature project into 96 dimensions is done using a linear embedding layer. We then alternate patch-merging layers with Swin Transformer blocks to extract hierarchical representations. Patch merging layers reduce token count by 4x but double our feature space. Subsequently, Swin Transformer blocks further process the features without altering their spatial dimensions. The model generates four hierarchical feature maps, named as FM1-FM4, with dimensions of 28 × 28 × 192, 14 × 14 × 384, 7 × 7 × 768, and 7 × 7 × 768, respectively.

2) Face Recognition Subnet: Face recognition demands stable features under local changes. For this reason, only the last level feature map, FM4, is fed into the face recognition sub-network. Inspired by ArcFace architecture, batch normalization (BN) is used to generate the final 512 dimensional embedding.

3) Face Analysis Subnets: This proposed network can perform 42 tasks related to face analysis, which have been classified into 11 categories, depending on the similarity of the tasks. All the tasks included in one category use one single face analysis subnetwork to minimize computation costs. These subnetworks include Multi-Level Channel Attention (MLCA) block, max pooling layer, ReLU, fully connected layers (2), and other fully connected layers, depending upon the task to be performed. MLCA allows each subnetwork to focus on specific features.

` In the face recognition pipeline, the facial images undergo preprocessing techniques such as face detection, face alignment, face normalization, and face augmentation. After the pre-processing step, the images undergo feature extraction using the Swin Transformer backbone. The extracted features are then projected into the discriminative feature space through various metric learning methods including ArcFace, CosFace, and Triplet Loss. Finally, the similarity metric is utilized to ascertain whether the two facial images are from the same person. SwinFace has been able to show better results when benchmarked against other face recognition systems due to its ability to capture fine-grained and global features of the face. The model uses transformer technology that allows for better generalization capabilities, robustness, and recognition accuracy.

3.7.2. Sparse Vision Transformer (S-ViT)

Sparse-ViT is an advanced architecture based on the classical ViT model that lowers computational costs through attending only to the key parts of images instead of all patches of the input images equally [32]. Classical Vision Transformers use global self-attention operations for all patches of images, which leads to large computation and memory costs, especially when working with high-resolution faces. Sparse ViT

solves this problem through the implementation of the mechanism of sparse attention that focuses only on key patches. The Sparse ViT algorithm is able to identify the discriminative facial features including eyes, nose, mouth, face boundary and facial texture pattern while ignoring the insignificant patches in face recognition. First, patch embeddings are formed by splitting the input face image into patches. Then, using sparse self-attention process, only the most relevant facial patches are attended, which avoids unnecessary computational burden. The Sparse ViT algorithm performs hierarchical feature extraction and transformer encoding, resulting in very powerful facial representations for face verification and identification tasks.

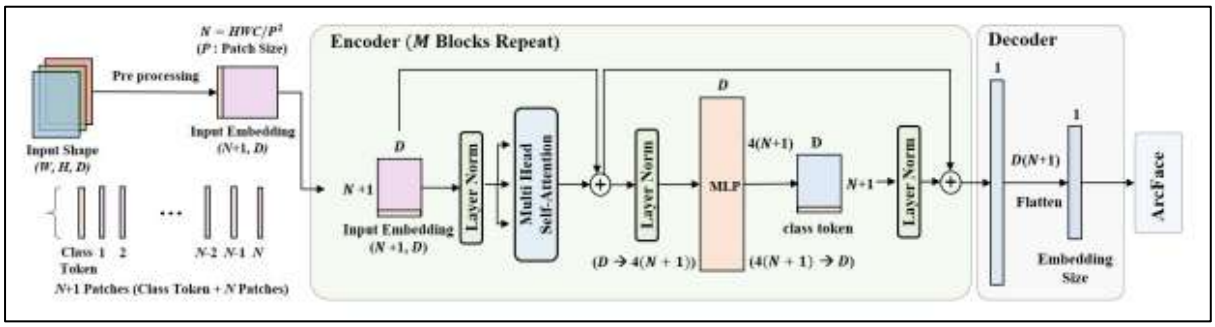

Figure 5. Overall architecture of S-ViT

A) Architecture of S-ViT

As shown in Figure 5, S-ViT is a model with a pre-processing phase that uses the same hybrid technique as ViT and a comparable architecture. S-ViT varies from ViT in these ways:

1) In the conventional ViT, absolute positional encoding is introduced during preprocessing. In S-ViT, this step is omitted, and relative positional encoding is incorporated directly into the multi-head self-attention (MSA) layer. As it provides positional information based on the connections between searches, keys, and values, image Relative Positional Encoding (iRPE) is specifically used. Bias and contextual are the two modes that iRPE provides. Below is the fundamental formulation of the bias mode:

$$e_{ij} = \frac{(x_i W^Q)(x_j W^k)^T}{\sqrt{d_x}} + b_{ij}, \quad b_{ij} = PE_{ij} \qquad (22)$$

The conventional relative positional encoding in Eq. (22), the proposed method introduces the positional information as a bias term after the dot-product operation. In the bias mode, positional encoding is independent of the input features, whereas the contextual mode establishes an interaction between them through a dot product of the input representation and the positional encoding matrix. For example, the appropriate bias term is expressed as follows when the query interacts with the positional encoding:

$$b_{ij} = (x_i W^Q)(PE_{ij})^T \qquad (23)$$

Eq. (23) shows how changing the bias affects the query and key interaction with positional encoding. Positional encoding and query only interact. It describes bias as follows:

$$b_{ij} = (x_i W^Q)(PE_{ij}{}^k)^T + (x_i W^k)(PE_{ij}{}^Q)^T \qquad (24)$$

Among the two iRPE modes, S-ViT specifically used contextual mode, and Eq. (24) is utilized to apply the bias.

2) Second, S-ViT uses all token embeddings in the decoder rather than relying only on the class token. While the original ViT decoder uses the information aggregated by a single class token, S-ViT directly incorporates all token representations. Although this substantially increases the decoder input dimension and results in more than 90% of the model parameters being concentrated in the decoder, it also provides improved performance and a favorable accuracy–computational cost trade-off, as demonstrated by the experimental results.

3) Third, avoid using the softmax loss function and instead use ArcFace. We believe ArcFace would be better suited for our objective as it is often used to enhance CNN-based facial recognition performance.

B) Exploring Sparisity in ViT

S-ViT exhibits a sparser weight distribution than the conventional ViT. Figure 5 compares the weight distributions of both models trained on the AI Hub and CASIA datasets, shown in Figures 5(a) and 5(b), respectively. The top, middle, and bottom rows represent the distributions for the complete model, encoder blocks, and decoder block, respectively. Although sparse weights can facilitate model pruning and reduce computational requirements, excessive sparsity may adversely affect model performance. Experimental results indicate that S-ViT achieves higher baseline performance and greater resilience to pruning than the conventional ViT. This increased sparsity primarily results from using all token representations as decoder inputs, redundant parameters get near-zero weights throughout training, producing the distribution illustrated in Figure 5. Despite not directly causing sparsity, iRPE, it improves the overall model performance.

C) Pruning for S-ViT

In CNNs, pruning can be performed at the level of individual convolutional filters. Likewise, the rows of an MLP weight matrix can be treated as analogous structural units and removed as a group. Accordingly, the MLP layers of S-ViT are pruned using a row-wise strategy. The rows selected for removal are identified according to either the sum or the mean of their weight L1 norms. The corresponding L1-norm summation is defined as follows:

$$Weight\ sum = \sum_{N=i}^{M} |W_{ki}|, \quad (k < M) \qquad (25)$$

where $\boldsymbol{k}$ is an integer less than $\boldsymbol{M}$ that designates a particular row for the weights sum computation, and $\boldsymbol{M}$ and $\boldsymbol{N}$ stand for number of MLP layer weight matrix rows and columns, respectively.

3.7.3. Inspection v2

Inception-v2 is the second-generation Inception CNN architecture and is distinguished by the integration of batch normalization. This modification improves training stability and allows the model to omit dropout and local response normalization. In contrast, Inception-v1 employs relatively large convolutional filters, such as 5×5, which can substantially reduce spatial information and may result in a loss of classification accuracy [33]. A substantial reduction in input dimensions can cause information loss and negatively affect network performance. Inception-v2 also reduces computational complexity through convolution factorization. A 3×3 convolution can be replaced by sequential 1×3 and 3×1 convolutions, providing an equivalent receptive field with approximately 33% lower computational cost. However, this factorization is less effective in the early layers where feature maps are large and is more suitable when the input size is approximately m×m, with (m) ranging from 12 to 20. In addition, the auxiliary classifier used in Inception-v1 can improve network convergence by supplying useful gradient signals to earlier layers, thereby mitigating the vanishing-gradient problem in deep architectures.

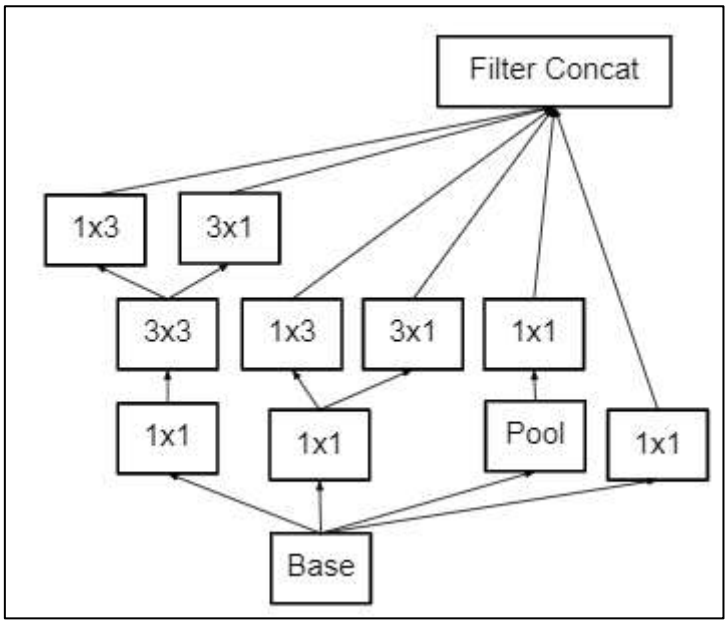

Figure.6. Inspection V2 architecture

In the Inception V2 architecture as shown in figure 6. Inception-v2 replaces a 5×5 convolution with two successive 3×3 convolutions, reducing computational cost and improving processing efficiency. Since a 5×5 convolution requires approximately 2.78 times more computation than a 3×3 convolution, this replacement provides a more efficient architecture while maintaining an effective receptive field. The model also factorizes *nXn* convolutions into sequential 1×n and n×1 operations. For instance, splitting up a 3×3 convolution operation to be performed via 1×3 and 3×1 would save computational resources by about 33%. In order to mitigate any problems with representation bottlenecking, the unit enlarges the feature banks instead of building additional network layers.

3.7.4. QCNN

A Quantum Convolutional Neural Network (QCNN) is the result of combining quantum computing with convolutional neural networks. The concept of QCNN combines quantum mechanical operations like superposition, entanglement, and quantum measurement with the design of conventional convolutional neural networks.

However, in quantum computing, qubits can be used for information representation in such a way that conventional ideas about CNN can be applied to the quantum environment [34]. This chapter describes the

architecture of the suggested QCNN. As can be seen from Fig. 7, the QCNN utilizes quantum versions of two basic CNN operations: convolution and pooling. The overall procedure is described as follows:

1) The convolution circuit ascertains the hidden state by applying several qubit gates between adjacent qubits.

2) Pooling circuits minimize quantum system size by monitoring qubit proportions or employing 2-qubit gates like CNOT gates.

3) Repeat the pooling and convolution circuits described in (1)–(2).

4) A fully connected quantum circuit may be utilized to anticipate the classification result when the quantum system is relatively small. The Multi-scale Entanglement Renormalization Ansatz (MERA) is commonly employed to realize this structure because it efficiently represents many-body quantum states. In its conventional form, MERA progressively expands the quantum system by introducing qubits in the (|0\rangle) state at each depth. QCNN applies this concept in the other manner, where the quantum system's size is gradually reduced from the input representation by the reversed MERA, making it suitable for QCNN architectures. The QCNN framework proposed by Cong further enhances this structure by incorporating Quantum Error Correction (QEC) [11]. In the MERA framework, each class label is associated with a representative quantum state |ψ〉 . Since QCNN operates using the reverse form of MERA, providing |ψ〉 as the input enables the corresponding class label to be determined. However, when an input state |ψ〉 that cannot be represented by MERA is encountered, the QCNN may fail to produce a definite classification. Incorporating QEC addresses this limitation by introducing additional degrees of freedom, enhancing the model's ability to manage certain input conditions.

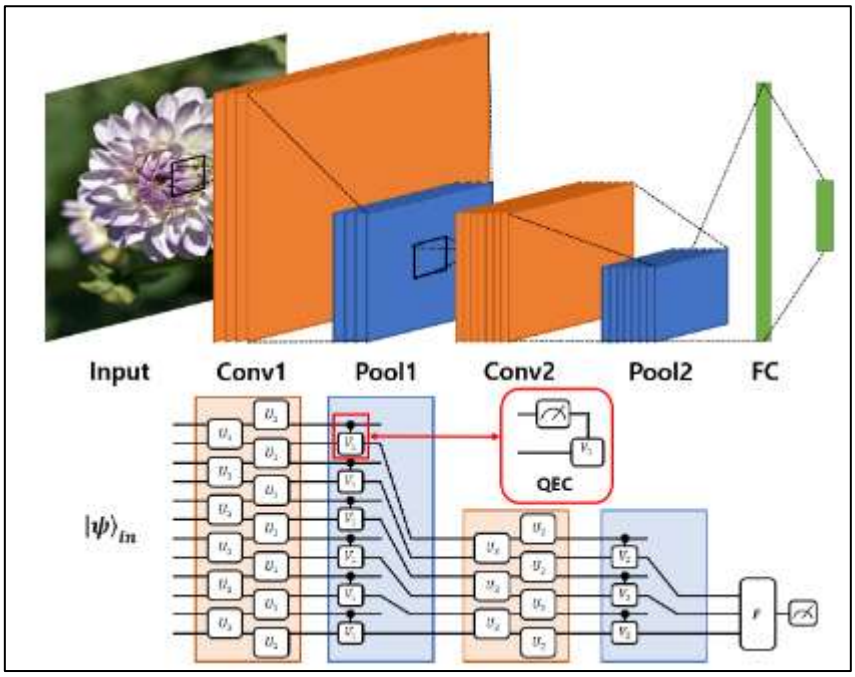

Fig. 7. Simple example of CNN and QCNN.

In the pooling layer, the result should be |0〉 , the same as the newly provided state in MERA, when the data given to QCNN is |ψ〉 . However, it is possible that the measured result will be |1〉 if the input data is |ψ′〉 , which MERA cannot produce. They are used to adjust the result if |1〉 is measured by adding gate to the surrounding qubits. Thru more predictable measurement results, the approach may provide improved performance. However, the training of QCNN models is difficult because optimization of quantum circuits can suffer from barren plateaus, where gradients become extremely small and learning slows down. Using parameter initialization introduced in this work for improving the performance of the QCNN.

- **Parameter initialization**

Assuming $Pa(\Psi i)$ shows how often the classifier makes accurate predictions Ψi, and $itter(\psi i)$ indicates a number of times that Ψi has spent in the network. The weight ω(Ψi) of the classifier is expressed as follows:

$P_a(\Psi_i) > P_a^{\pi}\ then\ \omega(\Psi_i) = P_a(Psi_i)$ (26)

else $\omega(\Psi_i) = \frac{P_a(\Psi_i)}{\sqrt{itter(\Psi_i)}}$ (27)

where P_a^{π} represents the hybrid model's classifiers' average accuracy π. The model Ψ's final prediction is as follows:

$\Psi(x) = i \quad if \qquad (28)$

$\sum_{t=1}^{T_{max}} \omega(\Psi_t) F_t^{(i)}(x) = \frac{max}{j\epsilon\{1,,2,.....J\}} \sum_{t=1}^{T_{max}} \omega(\Psi_t) F_t^{(i)}(x) \quad (29)$

The proposed hybrid approach assigns weights to individual classifiers according to their predictive accuracy and computational time. Classifiers whose weights fall below a predefined threshold are excluded during the initial processing stage. Using classification accuracy as the optimization objective enables the hybrid framework to select an effective combination of classifiers and improve overall performance.

3.7.5. Grad CAM

Gradient Weighted Class Activation Mapping (Grad-CAM) [35] is one of the widely recognized approaches in the field of XAI, used for explaining predictions made by deep neural networks, such as CNNs. It focuses on visualizing the image areas that are most influential for a certain classification output by considering the gradients of the target class in the final convolutional layer. The gradients are utilized for computing the importance weights of the feature maps, which are later combined to produce a discriminative localization map referred to as heatmap.

The heatmap created explains the influential parts of the image affecting the predictions made by the model, making it easier for the scientists and engineers to understand how the classification is performed. Grad-CAM is useful in face recognition tasks because it helps to show the important facial features like eyes, nose, mouth, etc. This interpretable approach makes it possible to improve the transparency of the model, make an error analysis, gain trust in the automated decision-making systems, and validate the deep learning models in the critical security environment. Therefore, Grad-CAM has become an indispensable part of developing trustworthy face recognition models.

Grad-CAM Formula

For feature map k, the important weight is determined as follows:

$\alpha_k^c = \frac{1}{Z} \frac{\partial y^c}{A_{ij}^k} \qquad (30)$

where:

- α_k^c = feature map k's significance weight for class c
- y^c = score of class c
- A_{ij}^k= activation in feature map k at location (i,j)
- Z = the total pixel value of the feature map

The computation of the Grad-CAM heatmap is:

$L_{Grad-CAM}^c = ReLU \alpha_k^c A^k \qquad (31)$

where:

- $L_{Grad-CAM}^c$ is the localization map for class c
- ReLU retains only positive contributions to the prediction.

3.7.6. SHAP

SHAP (SHapley Additive exPlanations) is an advanced form of Explainable Artificial Intelligence (XAI) that can be used to interpret and explain predictions based on deep learning models. Based on the principles of cooperative game theory, SHAP calculates a value of importance, which is also referred to as the shapley value, using each attribute to evaluate its role in the model's prediction [36]. For face recognition, SHAP is used for explaining the influence of facial features extracted using deep learning models like CNN, DenseNet, ResNet, ViT, and ensembles. Using the analysis of feature importance scores, SHAP helps in identifying the facial features that play an important role in recognition. These include eyes, nose, mouth, facial contours, and textures. The visual explanations generated through this technique help researchers identify whether the model focuses more on facial features instead of unnecessary background details.

SHAP improves transparency and credibility in face recognition systems through interpretable explanations for how the model works. This includes the identification of any bias in the models, debugging of classification errors, verification of the feature selection process, and improvement in model accuracy. Additionally, SHAP can be used in combination with intelligent feature selection to evaluate the importance of selected features. In security-critical applications such as biometric authentication, surveillance, and access control, SHAP has been found to be essential in providing transparency and explainability of recognition decisions while meeting the ethical requirements of AI. Through the integration of SHAP with deep learning-based face recognition techniques, scholars can be able to realize not only high accuracy in face recognition but also increased interpretability.

SHAP Formula

For feature iii, the Shapley value is determined as follows:

$\phi_i = \frac{|S|!(|F|-|S|-1)!}{|F|!} [f(S \cup \{i\} - f(S)] \qquad (32)$

Where:

- ϕi= the feature's SHAP value

- F = entire feature set
- S = excluding i features
- f(S) = subset S model prediction

4. Results and Discussion

The proposed facial expression recognition is implemented in Matlab. The proposed methodology is tested on https://www.kaggle.com/datasets/jessicali9530/lfw-dataset. The LFW dataset is a benchmark collection of facial photographs developed for evaluating face recognition under unconstrained conditions. Created by researchers at the University of Massachusetts Amherst, the dataset contains 13,233 images representing 5,749 individuals. The Viola-Jones face detector was used to localize and center the faces in the images that were collected from the internet. Among the subjects, 1,680 individuals are represented by at least two distinct photographs. The original LFW collection includes four image sets and three different forms of aligned face images. With regard to accuracy, precision, recall, and f-measure, the proposed Improved Convolutional Neural Network based face recognition outperforms previous adaptive boosting (AB) and Random Forest (RF).

1. Accuracy

The weighed percentage of facial expressions is correctly identified by the measurement accuracy. It is represented as,

$$\text{Accuracy}=\frac{TP+TN}{TP+FP+TN+FN} \quad (33)$$

Where,

TP-True Positive
TN-True negative
FP-False Positive
FN-False Negative

2. Precision

The proportion of properly anticipated positive results to all expected observations is known as precision.

$$\text{Precision}=\frac{TP}{TP+FP} \quad (34)$$

3. Recall

The proportion of effectively anticipated positive results to the total number of observations in the initial class is defined as recall.

$$\text{Recall}=\frac{TP}{TP+FN} \quad (35)$$

4. F-Measure

The weighed mean of accuracy and recall is used to get the F-measure.

$$F=2\frac{P\times R}{P+R} \quad (36)$$

Table 1. Table of comparisons between proposed and existing approaches

Method	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
AB	85.12	81	84	83
RF	93	90.09	92.2	91.12
ICNN	94.54	95.7	93.3	94.49
FN-QCNN	97.78	98.1	97.45	97.7
ESQDL	98.32	98.89	98.67	98.23

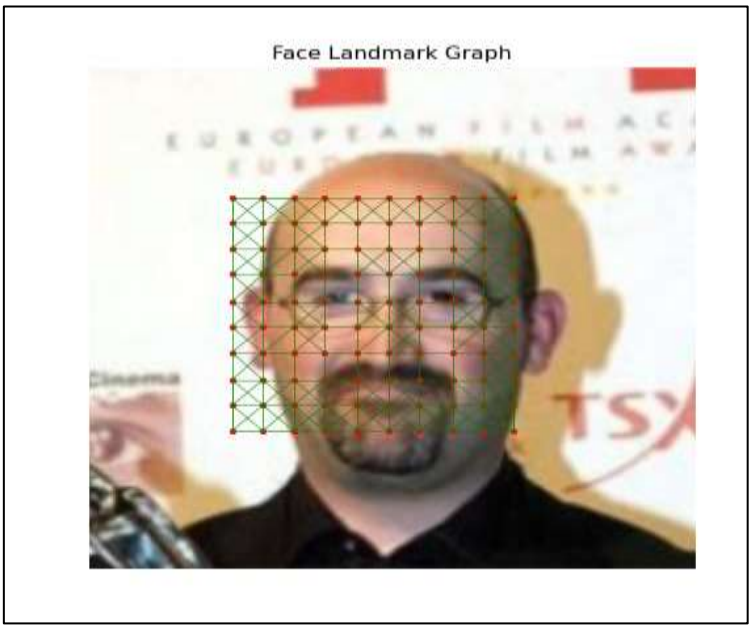

Figure .8. Face land mark graph

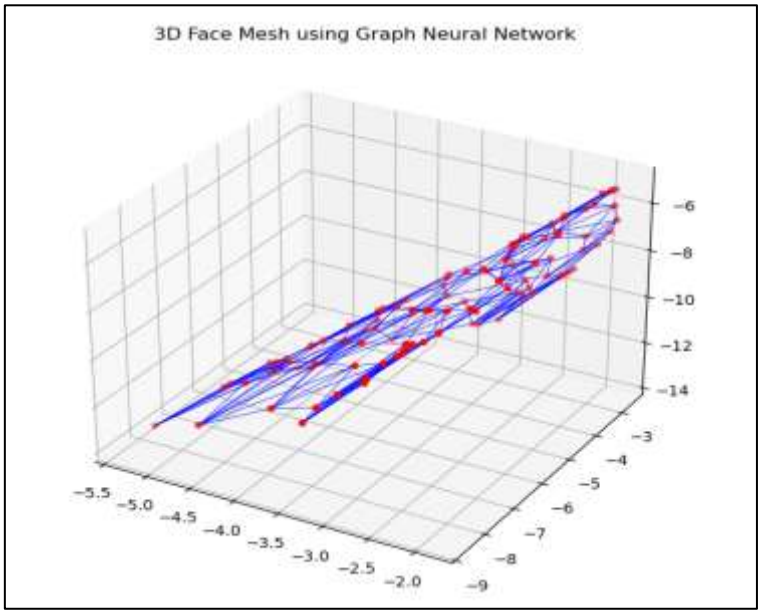

Figure .9. 3D face mesh using graph neural network

Figure .10. Segmentation image

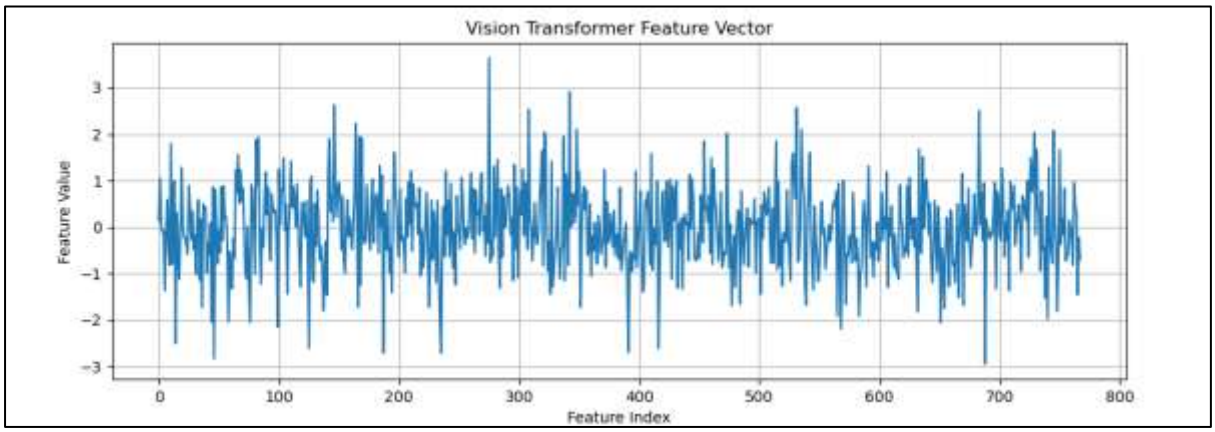

Figure .11. Vision transformer feature vector

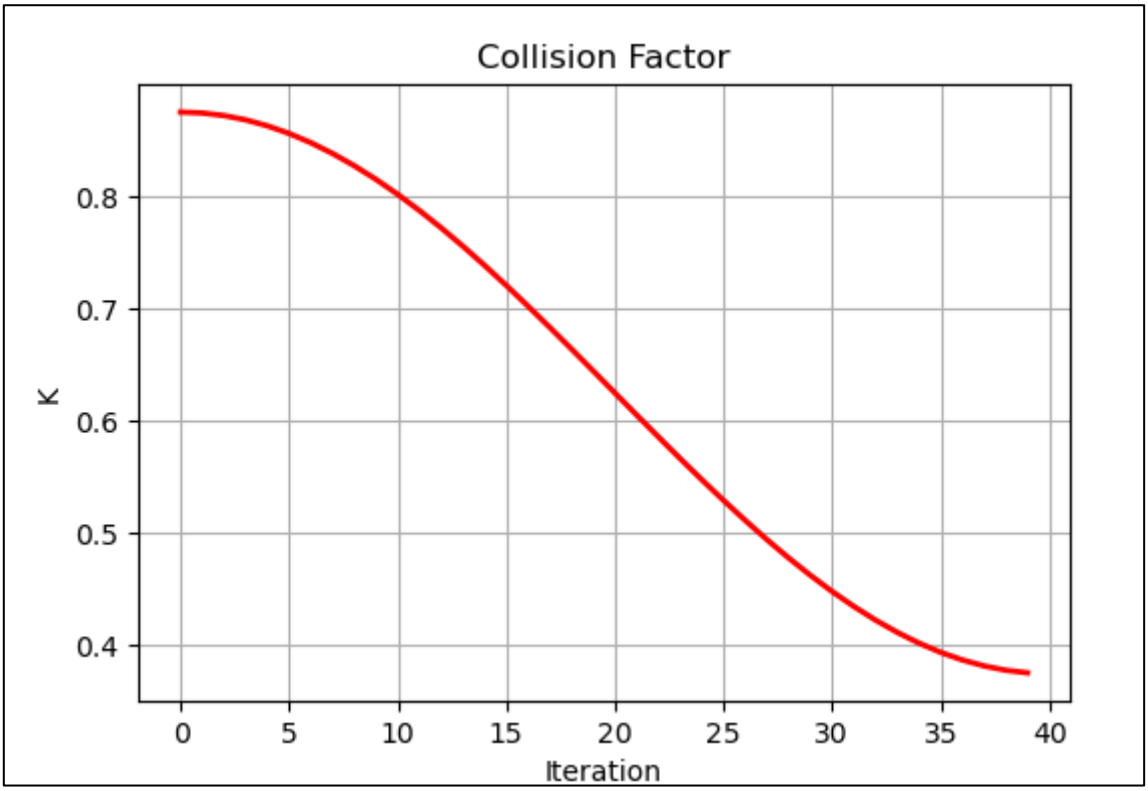

Figure .12. Result of collision factor

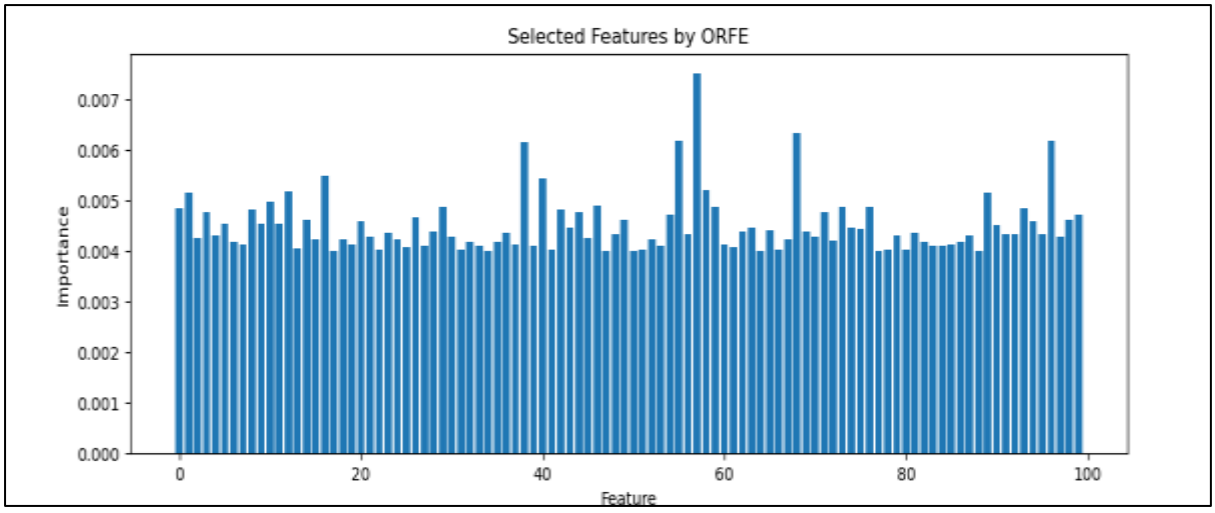

Figure 13. Selected features by ORFE

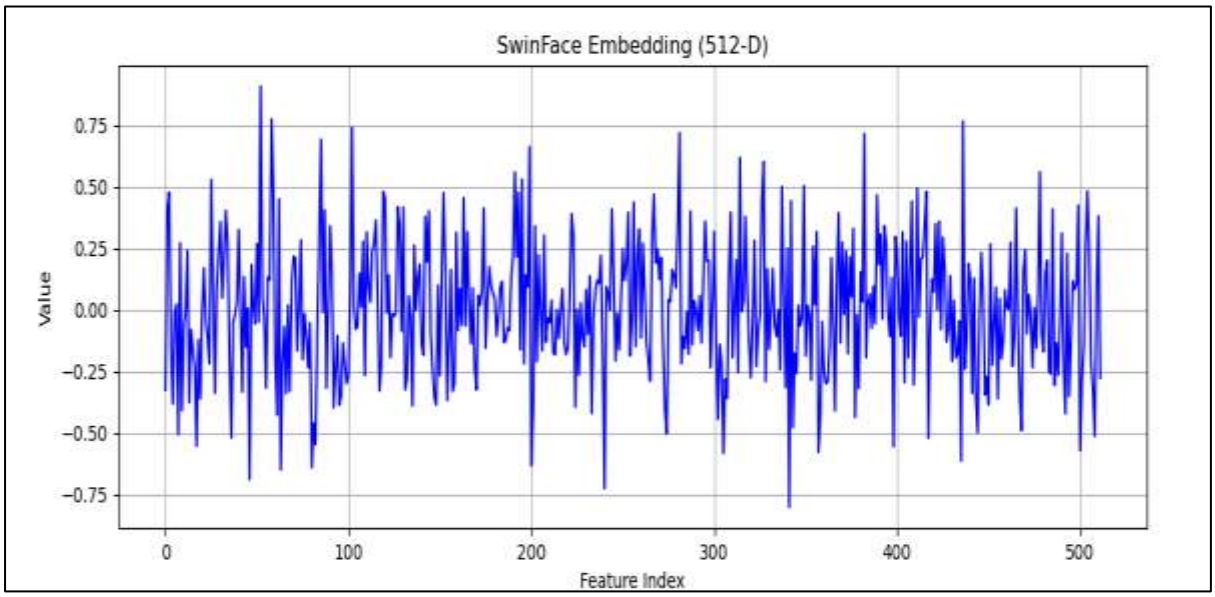

figure 14. swin face embedding

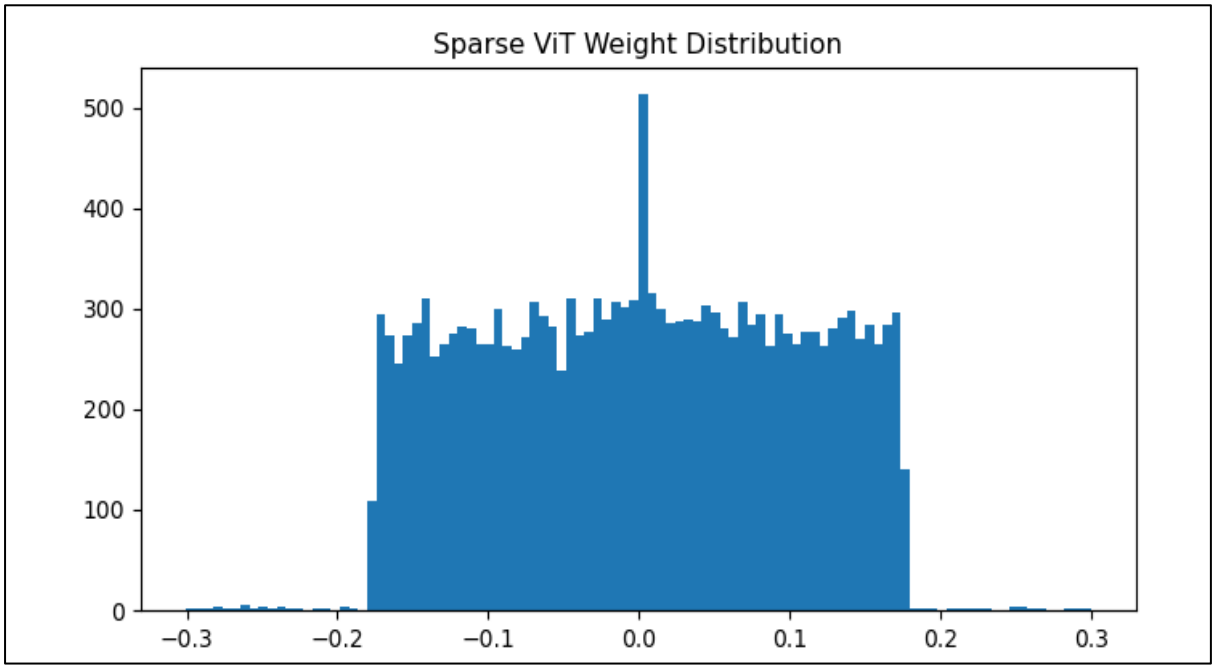

Figure 15. sparse ViT weight distribution

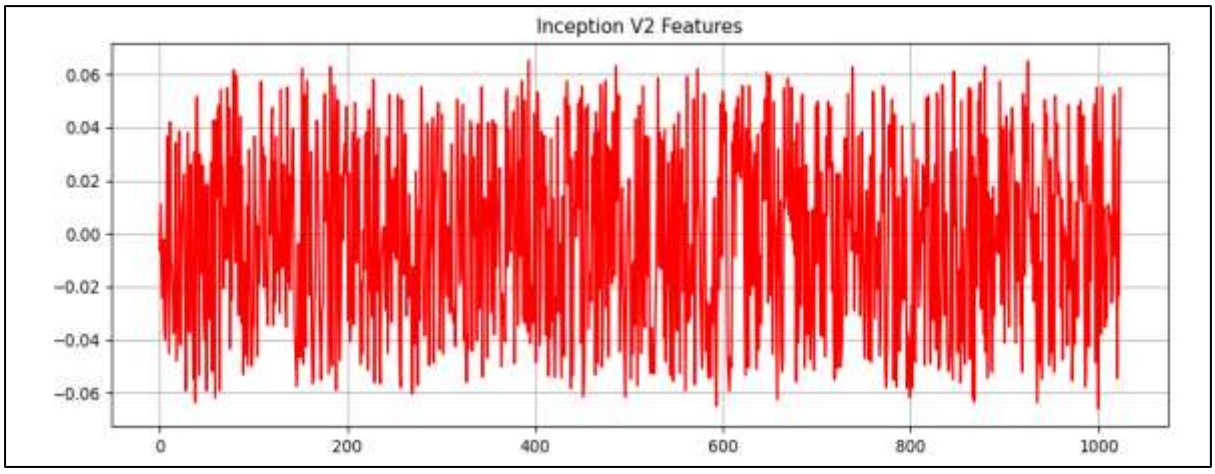

Figure .16. Inception V2 features

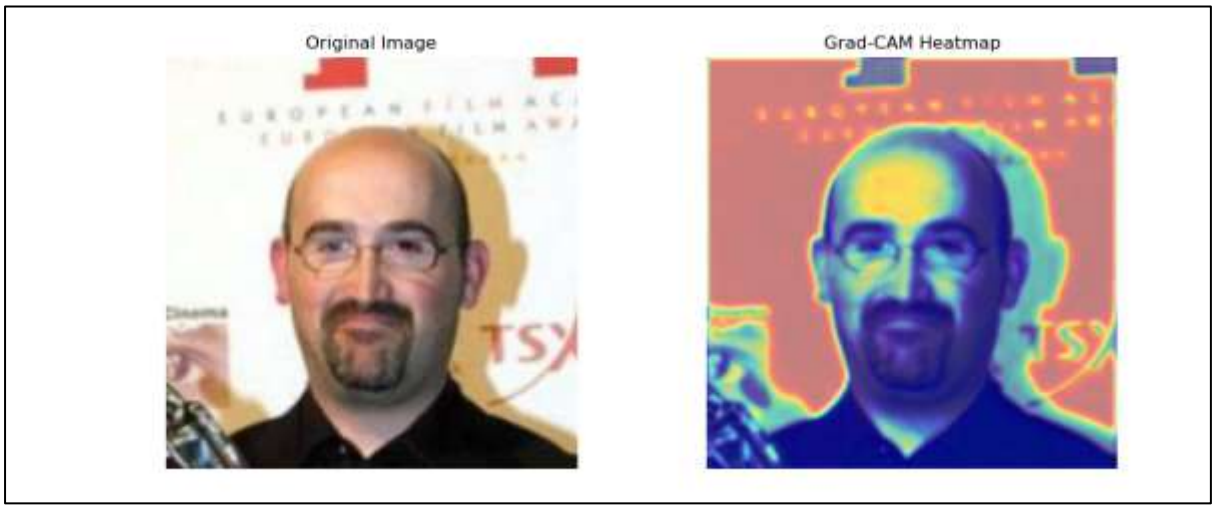

Figure 17. output image of heatmap

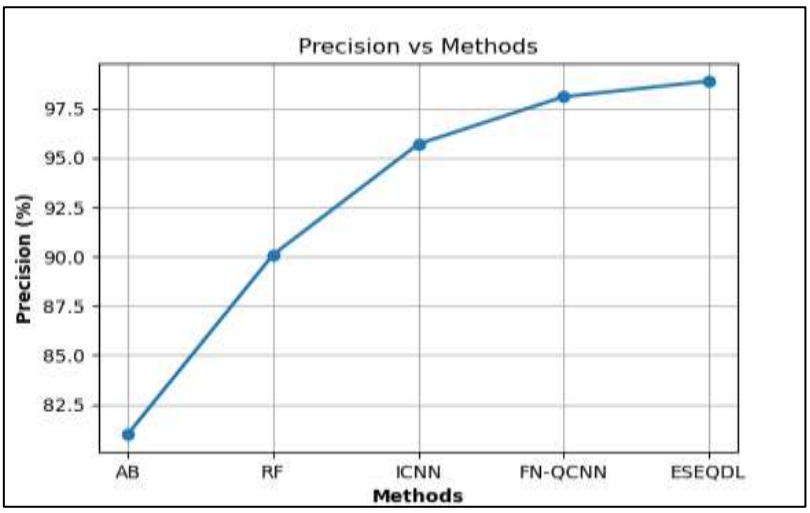

Fig.18. Precision comparison of the proposed ESEQDL

Figure 18 presents a precision-based comparison of the proposed ESEQDL model with existing methods for face image recognition. The ESEQDL model achieves higher precision across the evaluated datasets than the selected machine-learning approaches. These findings are consistent with the previously observed error rates and may be attributed to the non-redundant rule sets generated by the proposed classifier. From the results it concludes that the proposed ESEQDL technique has high precision results compare to the existing classification techniques.

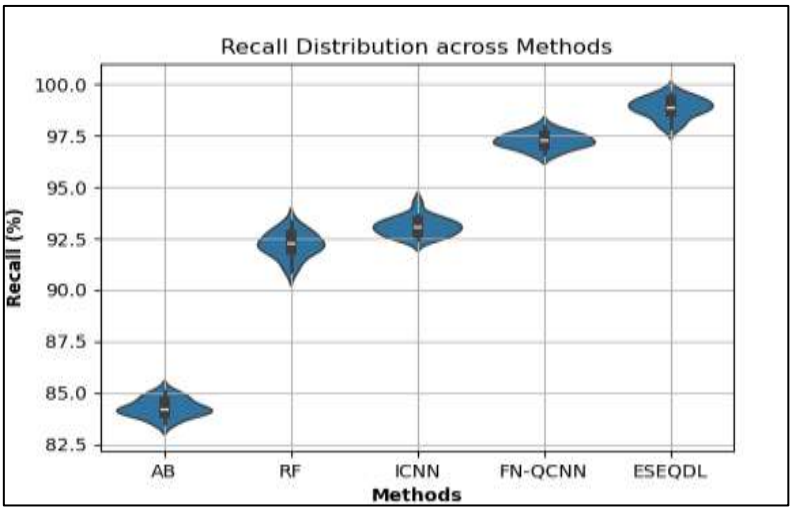

Fig.19. Recall comparison of the proposed ESEQDL

Figure 19 compares the recall performance of the proposed ESEQDL model with existing face recognition methods. The model is trained with a large number of images for each candidate using the ESEQDL approach, resulting in an extensive training dataset that improves recognition performance. The results also indicate that illumination conditions affect recognition accuracy, with lower light intensity increasing the likelihood of misclassification. This limitation can be reduced by incorporating more low-light facial images into the training data used to construct the face classifier. The parameter optimization using improved whale optimization increases the recall ratio.

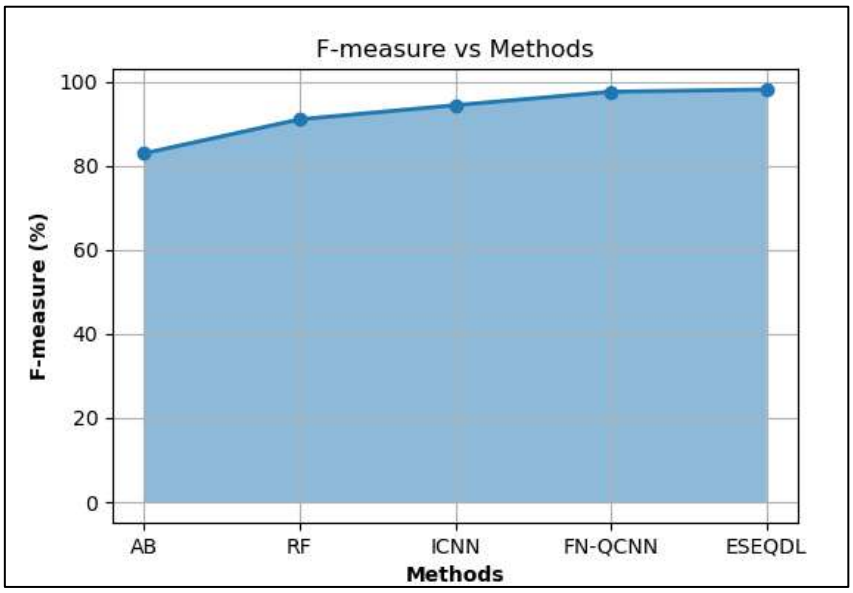

Fig.20. F-measure comparison of the proposed ESEQDL

Figure 20 presents the F-measure comparison of the proposed ESEQDL model with existing approaches for face image recognition. The evaluation of the feature extraction and classification methods indicates that the LFW dataset provides the strongest correlation between its variables and the target class. A comparison of machine-learning algorithms shows that proposed model performs effectively. In particular, the LFW dataset produces the best F-measure compared with the other datasets. From the graph it is noted that the proposed ESEQDL model has high f-measure results than the existing methods.

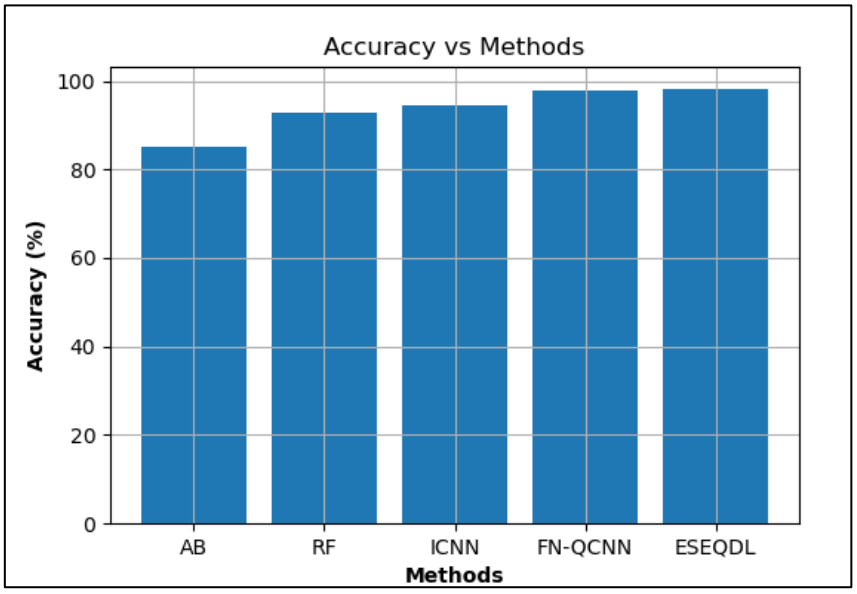

Fig.21. Accuracy comparison of the proposed ESEQDL model

Figure 21 compares the accuracy of the proposed method with existing approaches for face image recognition. An effective deep learning model should correctly identify the target class while maintaining its ability to generalize to previously unseen samples. Model validation is commonly assessed using accuracy, which is evaluated through measures such as sensitivity and specificity. From the results it

concludes that the proposed FN-QCNN technique has high accuracy results compare to the existing classification techniques. The proposed FN-QCNN method achieves higher precision than the existing approaches considered in the comparison. The adopted preprocessing strategy also contributes to improved recognition performance, thereby increasing the overall effectiveness of the facial recognition system. The proposed ESEQDL yields improved accuracy results of 98.32% whereas FN-QCNN yields 97.78% ICNN technique yields is 94.54% RF technique yields is 93% and AB technique provides 85.12%.

5. Conclusion

An Explainable Stacking Ensemble Quantum Deep Learning and Intelligent Feature Selection Approach for Face Recognition has been introduced by the research work to address the challenges associated with face recognition due to its complexity in a real-world setting. This framework involves the use of quantum-inspired deep learning techniques, intelligent feature selection, and stacking ensemble learning to increase the discriminative power of face representation. Moreover, explainable AI algorithms have been used in order to increase the transparency of the recognition process. The intelligent feature selection algorithm was able to select those facial features which were most informative, hence lowering the computational complexity. The stacking-based ensemble model proved to be effective in using the capabilities of different deep learning algorithms. Further, the use of the quantum inspired learning algorithm also helped enhance the optimization capabilities, the rate of convergence, and the effectiveness of feature representation learning. The explainability module gave useful information regarding the decision making process through indicating the regions and features of the faces that played an important role in the recognition process, thus enhancing the trustworthiness of the system. It was found through the experimental evaluation that the proposed technique performed better than traditional face recognition techniques. The results were consistent with the fact that the fusion of quantum deep learning, intelligent feature selection, ensemble classification, and explainable AI greatly improved face recognition accuracy under varying lighting, pose, expression, occlusion, and other difficult situations. In general, the suggested Explainable Stacking Ensemble Quantum Deep Learning and Intelligent Feature Selection Methodology is highly effective, precise, and explainable, making it an ideal method for developing next-generation face recognition systems used in various security applications. Future research can focus on the development of lightweight and computationally efficient quantum-inspired deep learning architectures suitable for real-time face recognition applications on edge devices, mobile platforms, and resource-constrained environments.

References

1. Li, L., Mu, X., Li, S., & Peng, H. (2020). A review of face recognition technology. *IEEE access*, *8*, 139110-139120.
2. Adjabi, I., Ouahabi, A., Benzaoui, A., & Taleb-Ahmed, A. (2020). Past, present, and future of face recognition: A review. *Electronics*, *9*(8), 1188.
3. Kortli, Y., Jridi, M., Al Falou, A., & Atri, M. (2020). Face recognition systems: A survey. *Sensors*, *20*(2), 342.
4. Tomar, V., Kumar, N., & Srivastava, A. R. (2023). Single sample face recognition using deep learning: a survey. *Artificial Intelligence Review*, *56*, 1063.
5. Prasad, P. S., Pathak, R., Gunjan, V. K., & Ramana Rao, H. V. (2019, August). Deep learning based representation for face recognition. In *ICCCE 2019: Proceedings of the 2nd International Conference on Communications and Cyber Physical Engineering* (pp. 419-424). Singapore: Springer Singapore.
6. Ratnaparkhi, S. T., Tandasi, A., & Saraswat, S. (2021, January). Face detection and recognition for criminal identification system. In *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 773-777). IEEE.
7. Jafri, S., Chawan, S., & Khan, A. (2020, February). Face Recognition using Deep Neural Network with" LivenessNet". In *2020 International Conference on Inventive Computation Technologies (ICICT)* (pp. 145-148). IEEE.
8. Khan, S., Ahmed, E., Javed, M. H., Shah, S. A., & Ali, S. U. (2019, July). Transfer learning of a neural network using deep learning to perform face recognition. In *2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)* (pp. 1-5). IEEE.
9. Hussain, S. A., & Salim Abdallah Al Balushi, A. (2020, January). A real time face emotion classification and recognition using deep learning model. In *Journal of physics: Conference series* (Vol. 1432, No. 1, p. 012087). IOP Publishing.
10. Manna, S., Ghildiyal, S., & Bhimani, K. (2020, June). Face recognition from video using deep learning. In *2020 5th International conference on communication and electronics systems (ICCES)* (pp. 1101-1106). IEEE.

11. Teoh, K. H., Ismail, R. C., Naziri, S. Z. M., Hussin, R., Isa, M. N. M., & Basir, M. S. S. M. (2021, February). Face recognition and identification using deep learning approach. In *Journal of Physics: Conference Series* (Vol. 1755, No. 1, p. 012006). IOP Publishing.
12. Wang, H., & Yan, W. Q. (2022). Face detection and recognition from distance based on deep learning. In *Aiding Forensic Investigation Through Deep Learning and Machine Learning Frameworks* (pp. 1-17). IGI Global Scientific Publishing.
13. Alagarsamy, S., Govindaraj, V., Irfan, M., Swami, R., & Kumar, N. M. (2020). Smart recognition of real time face using convolution neural network (CNN) Technique. *vo*, *83*, 23406-23411.
14. Reddy, V. N., Rao, P. H. S., Singh, N. S., Kumar, V. S. S., Reddy, Y. B., & Dharnasi, P. (2026). Face recognition using criminal identification system. *International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)*, *9*(2), 520-527.
15. Hangaragi, S., & Singh, T. (2023). Face detection and recognition using face mesh and deep neural network. *Procedia Computer Science*, *218*, 741-749.
16. Sivanagireddy, K., Jagadeesh, S., & Narmada, A. (2024). Identification of criminal & non-criminal faces using deep learning and optimization of image processing. *Multimedia Tools and Applications*, *83*(16), 47373-47395.
17. Sandhya, S., Balasundaram, A., & Shaik, A. (2023). Deep learning based face detection and identification of criminal suspects. *Computers, Materials, & Continua*, *74*(2), 2331.
18. Gupta, A., Punj, D., & Pillai, A. (2022). Face Recognition system based on convolutional neural network (CNN) for criminal identification. In *Mobile Radio Communications and 5G Networks: Proceedings of Second MRCN 2021* (pp. 21-33). Singapore: Springer Nature Singapore.
19. Tela, G., Hiwarale, S., Dhave, S., & Rathi, D. (2022). Cnn based criminal identification. *database*, *2*(2).
20. Chaves, D., Fidalgo, E., Alegre, E., Alaiz-Rodríguez, R., Jáñez-Martino, F., & Azzopardi, G. (2020). Assessment and estimation of face detection performance based on deep learning for forensic applications. *Sensors*, *20*(16), 4491.
21. Nida, N., Irtaza, A., & Ilyas, N. (2021, January). Forged face detection using ELA and deep learning techniques. In *2021 international Bhurban conference on applied sciences and technologies (IBCAST)* (pp. 271-275). IEEE.
22. Mohamed, S. S., Mohamed, W. A., Khalil, A. T., & Mohra, A. S. (2020). Deep learning face detection and recognition. *International Journal of Advanced Science and Technology*, *29*(2), 1-6.
23. Verma, Kesari, Bikesh Kumar Singh, and A. S. Thoke. "An enhancement in adaptive median filter for edge preservation." *Procedia Computer Science* 48 (2015): 29-36.
24. Ibrahim, Haidi, Nicholas Sia Pik Kong, and Theam Foo Ng. "Simple adaptive median filter for the removal of impulse noise from highly corrupted images." *IEEE Transactions on Consumer Electronics* 54.4 (2008): 1920-1927
25. Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., & Yang, M. H. (2022). Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5728-5739).
26. M. Zubair. (Jan. 6, 2022). *Face Detection With Mediapipe| Towards Data Science*. [Online]. Available: https://towardsdatascience.com/write-a-fewlines- of-code-and-detect-faces-draw-landmarks-from-complex-imagesmediapipe-932f07566d11.
27. Corso, G., Stark, H., Jegelka, S., Jaakkola, T., & Barzilay, R. (2024). Graph neural networks. *Nature Reviews Methods Primers*, *4*(1), 17.
28. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, *34*, 12077-12090.
29. Dutta, P., Sathi, K. A., Hossain, M. A., & Dewan, M. A. A. (2023). Conv-ViT: a convolution and vision transformer-based hybrid feature extraction method for retinal disease detection. *Journal of Imaging*, *9*(7), 140.
30. Awad, M., & Fraihat, S. (2023). Recursive feature elimination with cross-validation with decision tree: Feature selection method for machine learning-based intrusion detection systems. *Journal of Sensor and Actuator Networks*, *12*(5), 67.
31. Qin, L., Wang, M., Deng, C., Wang, K., Chen, X., Hu, J., & Deng, W. (2023). SwinFace: A multi-task transformer for face recognition, expression recognition, age estimation and attribute estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, *34*(4), 2223-2234.
32. Li, L., Thorsley, D., & Hassoun, J. (2022). Sait: Sparse vision transformers through adaptive token pruning. *arXiv preprint arXiv:2210.05832*.
33. Iqbal, H., Chawla, H., Varma, A., Brouns, T., Badar, A., Arani, E., & Zonooz, B. (2022). AI-driven road maintenance inspection v2: reducing data dependency & quantifying road damage. *arXiv preprint arXiv:2210.03570*.

34. Oh, S., Choi, J., & Kim, J. (2020, October). A tutorial on quantum convolutional neural networks (QCNN). In *2020 International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 236-239). IEEE
35. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-CAM: visual explanations from deep networks via gradient-based localization. *International journal of computer vision*, *128*(2), 336-359.
36. Mosca, E., Szigeti, F., Tragianni, S., Gallagher, D., & Groh, G. (2022, October). SHAP-based explanation methods: a review for NLP interpretability. In *Proceedings of the 29th international conference on computational linguistics* (pp. 4593-4603).