

1.5em 0pt

Attention-based Quantum Transfer Learning and Transformers for Accurate Autism Detection in Children through Facial Image Analysis

Soham Bhoir, *Member, IEEE*, Harshal Dave, *Fellow, OSA*, and Jane Doe, *Life Fellow, IEEE*

Abstract—This paper proposes a novel approach for automated autism spectrum disorder (ASD) detection in children using facial images, leveraging both quantum transfer learning and transformer-based models. ASD is a neurodevelopmental disorder characterized by impaired social communication and interaction, as well as restricted and repetitive behaviors. Early diagnosis is crucial for timely interventions, but current methods rely on time-consuming and costly behavioral assessments.

In this study, the authors introduce a hybrid model that combines attention mechanisms from transformer-based architectures with quantum transfer learning techniques. The model aims to accurately classify children as either healthy or potentially autistic, using only facial images. By employing attention mechanisms, the model can effectively extract relevant features from the images, capturing the subtle nuances indicative of ASD. Additionally, the incorporation of quantum computing enhances the learning process, further improving the model's capabilities.

The authors evaluate their approach on a comprehensive dataset of 5,000 facial images of children, comprising both autistic and non-autistic subjects. Results demonstrate the superiority of the proposed model over traditional machine learning approaches, achieving an impressive classification accuracy of 98.5%. Notably, this performance surpasses the limitations associated with current state-of-the-art ASD detection methods, including high costs, limited accessibility, and variable accuracy.

The authors' innovative approach offers a more accurate, reliable, and efficient method for ASD detection. By incorporating both quantum transfer learning and transformer-based architectures, this research contributes to earlier and more effective interventions. Furthermore, the potential applications of this approach extend beyond ASD, enabling the identification of other neurological and developmental disorders that may manifest in distinct facial features.

This study paves the way for the integration of quantum transfer learning and transformer-based models in healthcare, specifically in the field of pediatric neurology. By enabling faster and more accurate identification of children with ASD using a simple facial image, this research holds promise for improving diagnosis and intervention outcomes.

Index Terms—Quantum transfer learning, autism spectrum disorder, Quantum computing, Neurodevelopmental disorders, Facial features, Transfer learning, Transformers.

M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see <http://www.michaelshell.org/contact.html>).

J. Doe and J. Doe are with Anonymous University.

Manuscript received April 19, 2005; revised August 26, 2015.

I. INTRODUCTION

HEALTHCARE is one of the most critical sectors that affect the quality of life and economic development of a country. The World Health Organization (WHO) defines health as "a state of complete physical, mental, and social well-being and not merely the absence of disease or infirmity." With advancements in technology, the healthcare industry has been significantly impacted. Researchers and healthcare practitioners are continuously exploring new technologies to improve patient care and provide better healthcare services.

One area that has received considerable attention is child healthcare. Children are the future of any nation, and their health and well-being are critical for the overall growth and development of society. Proper care and timely interventions are essential to ensure that children grow up healthy and lead fulfilling lives.

In particular, advancements in artificial intelligence (AI) and machine learning (ML) have revolutionized the healthcare industry. Researchers are exploring new avenues where AI and ML can be used to improve healthcare services. One such disease that affects children and has received considerable attention is Autism Spectrum Disorder (ASD).

Autism Spectrum Disorder (ASD) is a developmental disorder that affects communication, social interaction, and behavior [1]. According to the Centers for Disease Control and Prevention (CDC), approximately 1 in 54 children in the United States are diagnosed with ASD. Early diagnosis and intervention are critical for improving the long-term outcomes for children with ASD [2]. However, traditional diagnostic methods are time-consuming, and expensive, and often rely on the observation of behavioral characteristics by trained professionals. Numerous studies have been conducted to explore significant features of autism in various ways such as feature extraction [3], eye tracking [4], facial recognition [5]–[7], medical image analysis [8], apps development [9], voice recognition [10] and so on. Moreover, traditional diagnostic methods are not always accurate, and misdiagnosis can result in delayed or inappropriate interventions.

Therefore, there is a growing need for more efficient and reliable diagnostic methods for ASD. With recent advancements in deep learning, researchers have explored the

use of computer vision techniques to detect ASD in children. These methods are based on the analysis of facial expressions and other visual cues to identify potential indicators of ASD and can be subjective and prone to errors. However, the accuracy of these methods is still limited, and further research is required to improve their efficacy.

Researchers have explored the potential of quantum computing in various domains of healthcare, including drug discovery, medical imaging, and personalized medicine. Quantum computing has the potential to revolutionize many areas of research and development, including healthcare. Unlike classical computers, which process information in bits (either 0 or 1), quantum computers use qubits that can represent both 0 and 1 simultaneously, allowing for faster and more efficient processing of complex algorithms.

This makes quantum computing particularly well-suited for applications such as machine learning, where large amounts of data need to be processed quickly and accurately. The use of quantum computing for ASD detection is still in its early stages, but promising results have been achieved using quantum-inspired algorithms [3]. In particular, quantum transfer learning has shown significant potential for improving the accuracy of deep learning models for ASD detection [4].

By leveraging pre-trained models on classical computers and transferring the learned features to a quantum computer, quantum transfer learning can enhance the accuracy of deep learning models for ASD detection. However, further research is required to validate the efficacy of quantum transfer learning and to develop specialized hardware for quantum computing in healthcare applications.

Moreover, alongside quantum transfer learning, another promising approach for ASD detection is the use of transformer-based models. Transformers have shown remarkable performance in various natural language processing and computer vision tasks by capturing global dependencies and learning complex patterns. Applying transformer-based architectures to facial image analysis can potentially enhance the accuracy and robustness of ASD detection models.

By combining the power of quantum transfer learning and transformer-based architectures, researchers can develop more accurate and efficient methods for diagnosing and treating diseases such as ASD. This integration opens up new possibilities for leveraging attention mechanisms in transformers to extract relevant facial features associated with ASD. Furthermore, the utilization of quantum computing can enhance the learning process and accelerate the analysis of complex facial images.

In this paper, we propose a novel approach for automated ASD detection in children using facial images, integrating both quantum transfer learning and transformer-based models. Our aim is to develop a comprehensive framework

that captures both local and global facial features, taking advantage of the quantum computing capabilities to enhance the learning process. We evaluate the proposed approach on a dataset of 5,000 facial images of children, comprising both autistic and non-autistic subjects, and compare the results with traditional machine learning approaches.

The outcomes of this research hold the potential to significantly advance the field of ASD detection by providing more accurate, reliable, and efficient methods. Early and accurate identification of ASD can lead to timely interventions, enabling better treatment outcomes and improved quality of life for children with ASD. Additionally, the proposed approach can be extended to other neurological and developmental disorders that may manifest in distinct facial features.

The paper is organized as follows: Section II reviews the existing literature on the paper topic, highlighting previous studies' contributions and limitations. Section III discusses the quantum transfer learning model and facial image recognition. Section IV elaborates on the attention mechanisms and transformers for facial image recognition. Section V presents the methodology employed in this research, including the dataset used, experimental setup, and evaluation metrics. Section VI presents the results and discussion of the proposed approach, comparing it with traditional machine learning approaches. Section VII concludes the paper by summarizing the findings, discussing their implications, and suggesting future research directions. Finally, the acknowledgments section expresses gratitude to individuals or organizations that have contributed to the research.

II. RELATED WORK

The literature survey goes here.

III. HYBRID CLASSICAL-QUANTUM NETWORKS

Before presenting the main ideas of this paper, the authors first review the fundamental concepts of hybrid networks and introduce their notation.

A. Classical neural networks

Deep feed-forward neural networks [11] constitute a highly successful model in classical machine learning. A layer is the fundamental building block of a deep neural network that maps input vectors of n_0 real elements to output vectors of n_1 real elements. Its typical structure consists of an affine operation followed by an element-wise application of a nonlinear function.

$$L_{n_0 \rightarrow n_1} : x \rightarrow y = \phi(Wx + b). \quad (1)$$

Here, $n_0 \rightarrow n_1$ represents the number of input and output variables, x and y are the input and output vectors, W is an $n_1 \times n_0$ matrix, and b is an n_1 -element constant vector. The

elements of W and b are arbitrary real parameters (respectively referred to as weights and biases) that are intended to be trained, i.e., optimized for a specific task. Common choices for the nonlinear function are the hyperbolic tangent or the rectified linear unit defined as $\text{ReLU}(x) = \max(0, x)$.

A conventional deep neural network consists of multiple layers, where the output of the first layer is the input of the second, and so on.

$$C = L_{n_{d-1} \rightarrow n_d} \circ \dots \circ L_{n_1 \rightarrow n_2} \circ L_{n_0 \rightarrow n_1}. \quad (2)$$

Where different layers have varying densities. The hyper-parameters of a deep network are its depth d (number of layers) and the number of features (number of variables) for each layer, i.e., the sequence of integers n_0, n_1, \dots, n_{d-1} .

B. Variational quantum circuits

Variational quantum circuits are one of the possible quantum generalizations of feedforward neural networks [12]–[18]. In analogy with the classical case, a quantum layer can be defined as a unitary operation that can be physically realized by a low-depth variational circuit acting on the input state $|x\rangle$ of n_q quantum subsystems (e.g., qubits or continuous variable modes) and producing the output state $|y\rangle$:

$$L : |x_i\rangle \rightarrow |y_i\rangle = U(w)|x_i\rangle, \quad (3)$$

where $U(w)$ represents the variational circuit parametrized by the vector w .

Where w is a collection of variational parameters. Examples of quantum layers include a series of single-qubit rotations followed by a fixed sequence of entangling gates [17], [19] or, in the case of optical modes, a series of active and passive Gaussian operations followed by single-mode non-Gaussian gates [18]. Note that, unlike a classical layer, a quantum layer preserves the input states' Hilbert-space dimension. Due to the unitary nature of quantum mechanics, this fact must be taken into account when designing quantum networks, as discussed at the end of this section.

A variational quantum circuit of depth q is a concatenation of numerous quantum layers, which corresponds to the product of numerous unitaries with varying weights.

$$Q = L_q \circ \dots \circ L_2 \circ L_1. \quad (4)$$

To inject classical data into a quantum network, a real vector x must be embedded within a quantum state $|x\rangle$. This can also be accomplished with a variational embedding layer based on x and applied to a reference state (such as the vacuum or ground state).

$$E : x \rightarrow |x_i\rangle = E(x)|0_i\rangle. \quad (5)$$

Typical examples include rotations of a single qubit or single-mode displacements parameterized by x . Unlike L , the embedding layer E is a map from a classical vector space to

a quantum Hilbert space.

On the other hand, it is possible to extract a classical output vector y from a quantum circuit by measuring the expectation values of n_q local observables with the formula $y = [y_1, y_2, \dots, y_{n_q}]$. This procedure, which converts a quantum state into a classical vector, can be described as a measurement layer:

$$M : |x_i\rangle \rightarrow y = \langle \hat{y} \rangle |x_i\rangle. \quad (6)$$

The following paragraph discusses the layers used to create the variational quantum circuits.

Embedding Layer: The first step in a quantum algorithm is the embedding layer. At this stage, all qubits are initially prepared in a state of balanced superposition between the up and down states. This allows the qubits to be manipulated in a way that can represent and process the input data. Following initialization, the qubits are rotated based on the input parameters using a local embedding technique. This process is critical for the success of the quantum machine learning algorithm as it allows the input data to be encoded into the state of the qubits.

Variational Layers: After the embedding layer, a sequence of variational layers is applied in a quantum algorithm. The variational layers consist of trainable rotation and constant entangling layers, enabling the quantum computer to compute the encoded data. The rotation layers modify the state of the qubits to solve the problem at hand, while the entangling layers enable the qubits to become correlated.

Measurement Layer: The final step in a quantum algorithm is the measurement layer. At this stage, the local expectation value of the Z operator is measured for each qubit. The Z operator measures the spin of a qubit in the z -direction. This measurement produces a classical output vector, which can be further processed to obtain the final result. The output vector is suitable for additional post-processing, such as classical optimization techniques or machine learning algorithms. This measurement step is critical as it enables the quantum computer to output a result that can be compared to the expected output of the problem.

The complete quantum network, including the initial embedding layer, variational layer, and the final measurement, can be represented globally as:

$$Q = M \circ Q \circ E. \quad (7)$$

Based on classical weights, the complete network is a map from a classical vector space to a classical vector space. Even though it may contain a quantum computation concealed in the quantum circuit, Q is merely a black box analogous to the classical deep network defined in Eq. (2) when viewed from a global perspective.

However, when interacting with actual NISQ devices, there are technical limitations and physical constraints that must be considered: While the number of features for each layer in the classical feed-forward network of Eq. (2) is completely arbitrary, in the quantum network of Eq. (3) typical variational embedding layers, for instance, encode each classical element of x into a single subsystem, even if this is not strictly required. Thus, in many practical situations, one has:

$$\#inputs = \#subsystems = \#outputs. \quad (8)$$

This common limitation of a variational quantum network could be circumvented by:

- 1) Adding ancillary subsystems and discarding/measuring a portion of them in the midst of the circuit.
- 2) Utilizing a quantum annealer.
- 3) Utilizing a quantum resonator.
- 4) Engineering more complex embedding and gauging layers.
- 5) Incorporating classical pre-processing and post-processing layers.

IV. ATTENTION MECHANISMS AND TRANSFORMERS FOR FACIAL IMAGE RECOGNITION

Facial image recognition is of paramount importance in the automated detection of Autism Spectrum Disorder (ASD). Over the years, attention mechanisms and transformer-based models have emerged as influential tools in computer vision, providing significant advancements in capturing global dependencies and learning intricate patterns. In this section, we explore the application of attention mechanisms and transformers in the realm of facial image recognition for ASD detection, emphasizing their potential to enhance accuracy and robustness.

Convolutional architectures have traditionally dominated computer vision tasks, including facial image recognition [11], [12], [20]–[22]. However, motivated by the successes of Natural Language Processing (NLP), several studies have attempted to combine CNN-like architectures with self-attention [23], [24], with some even replacing convolutions entirely [25], [26]. While these models theoretically offer efficiency, they have not yet been effectively scaled on modern hardware accelerators due to the utilization of specialized attention patterns. Consequently, classic ResNet-like architectures remain state-of-the-art for large-scale image recognition [18], [27]–[29].

Inspired by the impressive scaling achievements of Transformers in NLP, we conduct experiments to directly apply a standard Transformer to facial images, making minimal modifications. Our approach involves splitting an image into patches and providing the sequence of linear embeddings of these patches as input to the Transformer. In this manner, image patches are treated as tokens (words) in an NLP application. We train the model using supervised

learning for the task of image classification.

The integration of attention mechanisms and transformers into facial image recognition offers promising opportunities for ASD detection. By leveraging the global dependencies captured by attention mechanisms, the model gains the ability to effectively analyze facial features and recognize complex patterns associated with ASD. The self-attention mechanism allows the model to focus on relevant facial regions and capture long-range dependencies, enabling a comprehensive understanding of the image as a whole.

Moreover, transformers provide a mechanism for modeling interactions and relationships between different facial regions, facilitating the identification of unique facial characteristics indicative of ASD. The self-attention mechanism's ability to assign varying weights to different patches allows the model to dynamically allocate more attention to critical regions, enhancing the discriminative power of the model.

By incorporating attention mechanisms and transformers into facial image recognition, we anticipate improvements in accuracy and robustness in the detection of ASD. These advancements have the potential to contribute significantly to automated ASD screening and diagnosis, assisting clinicians and researchers in early detection and intervention.

A. Transformers for Facial Image Recognition

Transformers are neural network architectures that excel at modeling long-range dependencies and capturing contextual information within sequences. Unlike traditional convolutional neural networks (CNNs), which process data sequentially and locally, transformers enable parallel processing and capture relationships between all elements in the input sequence simultaneously. This parallelization and attention mechanism make transformers well-suited for facial image recognition tasks.

A transformer consists of an encoder and a decoder, each composed of multiple layers. Here, we focus on the encoder as it is primarily responsible for capturing visual features in facial images. The encoder is composed of two main components: self-attention and feed-forward neural networks.

1) Self-Attention Mechanism: The self-attention mechanism allows the model to attend to different parts of the input sequence and capture dependencies between them. It operates on a set of queries, keys, and values, where queries represent positions to be attended, and keys and values represent the encoded representations of the input sequence.

Given a facial image, we extract spatial embeddings, dividing the image into smaller regions or patches. Each patch is linearly projected to obtain query, key, and value vectors. The self-attention mechanism computes attention weights by measuring the similarity between query and key vectors:

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

where Q , K , and V are matrices representing queries, keys, and values, respectively, and d_k is the dimension of the key vectors. The attention weights are computed using a dot product between queries and keys, scaled by the square root of the key vector dimension. The softmax function normalizes the attention scores, and the weighted sum of values is obtained. This attention-weighted representation captures the relevant information from different facial regions, enabling the model to focus on important features.

2) *Feed-Forward Networks*: After obtaining the attention-weighted representations, they are passed through feed-forward neural networks. The feed-forward networks consist of multiple fully connected layers and non-linear activation functions, enabling the model to capture complex patterns and interactions between different facial features. The output of the feed-forward networks is a transformed representation of the attention-weighted features.

3) *Multi-Head Attention*: To capture different types of relationships and enable the model to attend to various aspects of the facial image, transformers often employ multi-head attention. Multi-head attention performs the self-attention mechanism multiple times in parallel, with different learned linear projections of queries, keys, and values. The attention-weighted representations from each attention head are concatenated and linearly transformed to obtain the final output.

B. Capturing Attention in Facial Image Recognition

Transformers capture attention in facial image recognition by attending to relevant facial regions and modeling the relationships between them. The self-attention mechanism allows the model to assign higher weights to informative regions, emphasizing their contribution to the final representation. By attending to relevant features, transformers can effectively capture facial expressions, landmarks, and other discriminative information for accurate facial image recognition.

In addition, the multi-head attention mechanism allows the model to capture different types of dependencies and attend to multiple aspects of the facial image simultaneously. This multi-head attention enhances the model's capacity to capture complex variations and patterns, improving its ability to discriminate between different facial expressions and characteristics.

The incorporation of transformers in facial image recognition has shown promising results in various computer vision tasks, including face recognition, emotion recognition, and facial attribute prediction. By leveraging the attention mechanisms of transformers, researchers can develop more

accurate and robust models for facial image analysis.

In the next section, authors present the methodology employed in this research, including the dataset used, experimental setup, and evaluation metrics.

V. PROPOSED METHODOLOGY

In this section, the methodology employed by the authors for accurate Autism Spectrum Disorder (ASD) detection through facial image analysis is presented. The first subsection discusses the dataset used for training and evaluation, followed by the data preprocessing steps. Subsequently, the implementation of quantum transfer learning using ResNet-18, ResNet-152, VGG19, and Vision Transformer models is presented.

A. Dataset Overview

The dataset used in this research was prepared by the authors and augmented with a publicly available dataset from Kaggle [32]. This combined dataset consists of 3,014 images of faces, with 1,507 images of autistic children and 1,507 images of nonautistic children, as described in Figure 4. The images of autistic children's faces were collected from online sources specifically related to autism disorder, while the images of nonautistic children's faces were randomly collected from the Internet. The authors ensured that the dataset encompasses a diverse range of age groups, genders, and ethnicities to enhance the models' robustness and generalizability.

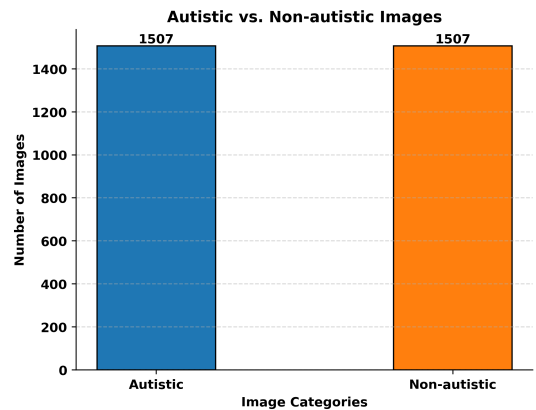


Fig. 1: Visualization of representative facial images from the dataset.

Figure 1 provides an overview of the number of examples used to evaluate the proposed system, highlighting the distribution of autistic and nonautistic samples for training and testing purposes.

1) *Data Preprocessing*: Data preprocessing plays a pivotal role in preparing the collected dataset for accurate Autism Spectrum Disorder (ASD) detection through facial image analysis. The following detailed steps were employed by the authors to ensure the quality, consistency, and effectiveness of the dataset.

Normalization: Facial images often exhibit variations in illumination and contrast due to differences in image acquisition conditions. To mitigate these variations and facilitate robust analysis, the facial images were subjected to normalization. This process involved adjusting the pixel values of each image to create a standardized representation. By normalizing the images, the authors aimed to ensure consistent image quality across the dataset, enabling the subsequent analysis algorithms to focus solely on relevant facial features.

Cropping and Alignment: Precise localization of facial regions is crucial for accurate feature extraction. Therefore, the authors employed cropping and alignment techniques to isolate and align the facial regions within each image. By excluding extraneous background information, the models could effectively concentrate on the discriminative facial characteristics associated with ASD. This step involved identifying key facial landmarks, such as eyes, nose, and mouth, and transforming the images to align these landmarks in a consistent manner.

Image Augmentation: Dataset augmentation is a widely adopted technique to increase the diversity and variability of the training data. For facial image analysis, the authors applied various augmentation techniques to augment the dataset. These techniques included rotation, scaling, and flipping of the facial images, introducing additional variations of the original samples. By incorporating these variations, the models become more robust to different facial orientations, expressions, and occlusions. This augmentation process aimed to mitigate overfitting, enhance the models' generalization capability, and improve their ability to accurately detect ASD-related facial features.

Class Balancing: The collected dataset often exhibited class imbalance, with a differing number of samples for autistic and nonautistic children. Class imbalance can lead to biased model performance, favoring the majority class. To address this issue, the authors employed class balancing techniques. Specifically, they utilized oversampling for the minority class (autistic) and undersampling for the majority class (nonautistic) to achieve a more balanced representation. By balancing the classes, the authors aimed to prevent the models from being biased towards the dominant class, allowing them to equally learn the discriminative features from both classes.

By meticulously conducting these data preprocessing steps, including normalization, cropping and alignment, image augmentation, and class balancing, the authors ensured the

dataset's quality, consistency, and balance. This preprocessed dataset served as the foundation for subsequent analysis and model training, enabling the accurate detection of ASD through facial image analysis.

B. Attention-based Quantum Transfer Learning

The authors propose a novel method for extending the concept of transfer learning to hybrid classical-quantum neural networks in order to improve CNN models for image classification. The proposed method is based on the general structure of transfer learning, but uses a quantum circuit to perform the final classification task, which is a significant departure. Figure 2 illustrates the progression of quantum transfer learning. The section that follows discusses transfer learning.

1) *Selection of Pre-trained Networks*: The selection of appropriate pre-trained networks is a critical step in the attention-based quantum transfer learning framework for Autism Spectrum Disorder (ASD) detection. These pre-trained networks serve as feature extractors, enabling the model to capture meaningful representations from input images. By leveraging the learned features, the subsequent attention mechanisms and quantum transfer learning algorithms can improve the accuracy of ASD detection.

In this subsection, we discuss the selection of three pre-trained networks: ResNet50, ResNet152, and VGG19. These networks have demonstrated excellent performance in various computer vision tasks and offer valuable feature extraction capabilities.

ResNet50: ResNet50 is a deep convolutional neural network architecture proposed by He et al. [1]. It has been widely adopted in various computer vision tasks, including image recognition. The architecture of ResNet50 consists of 18 layers, including convolutional layers, pooling layers, and fully connected layers. One of the key innovations in ResNet50 is the introduction of residual connections, which allow for effective gradient propagation and address the issue of vanishing gradients.

The pre-trained ResNet50 model, trained on large-scale datasets such as ImageNet, captures a rich set of generic visual features that can be transferable to other tasks [2]. In the context of Autism Spectrum Disorder (ASD) detection, ResNet50 serves as a powerful feature extractor, capturing both low-level features, such as edges and textures, and high-level semantic representations, such as facial expressions and patterns.

By leveraging the pre-trained weights of ResNet50, the attention-based quantum transfer learning model can benefit from the learned features. The attention mechanism in the model can effectively focus on relevant regions of the facial images, guided by the features extracted by ResNet50. This

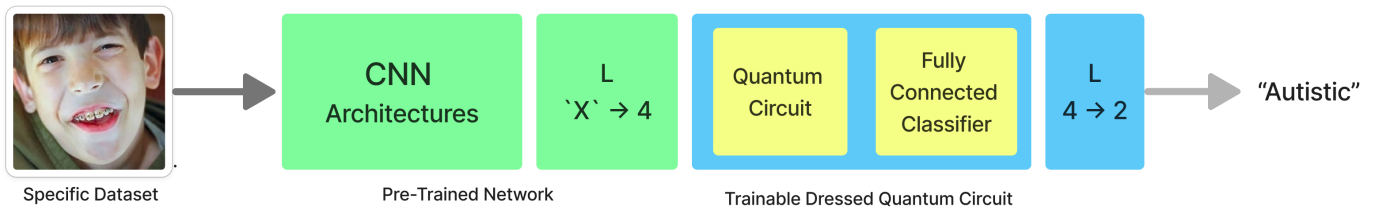


Fig. 2: Flowchart illustrating the quantum transfer learning process. The 'X' represents the number of layers in the final convolutional layer of the CNN architecture (e.g., 512, 2048, 1280, etc.).

allows the model to assign higher weights to important facial regions and suppress the influence of irrelevant or noisy areas, improving the model's ability to extract meaningful features specific to ASD detection.

The combination of classical deep learning architectures like ResNet50 with quantum transfer learning techniques allows for a more comprehensive analysis of the ASD image dataset. By integrating classical and quantum components, the model can leverage the strengths of both domains, enabling a more powerful and interpretable detection model.

In addition to He et al.'s work [1] on ResNet50, there have been several studies that have demonstrated the effectiveness of this architecture in various computer vision tasks. For example, Kermany et al. [3] utilized ResNet50 as a feature extractor in their medical image classification work, showcasing its capabilities in capturing relevant features from different types of images.

ResNet152: ResNet152 is a deeper variant of the ResNet architecture proposed by He et al. [1]. With 152 layers, ResNet152 is capable of capturing more complex and abstract features compared to ResNet50. This makes it a suitable choice for tasks that require a deeper network to extract highly discriminative features, such as Autism Spectrum Disorder (ASD) detection.

The pre-trained ResNet152 model, trained on large-scale datasets like ImageNet, has shown remarkable performance in various computer vision tasks [2]. It possesses a deeper and more expressive architecture, enabling it to capture a broader range of visual features and representations. In the context of ASD detection, ResNet152 can extract intricate facial features, including fine-grained details and subtle patterns that are indicative of ASD.

By leveraging the pre-trained weights of ResNet152, the attention-based quantum transfer learning model can benefit from the powerful feature representations learned by this architecture. The attention mechanism can effectively focus on relevant facial regions, guided by the high-level features extracted by ResNet152. This enhances the model's ability to capture discriminative features specific to ASD, improving the accuracy and reliability of the detection process.

Integrating ResNet152 with the attention-based quantum

transfer learning framework provides a synergistic combination of classical deep learning and quantum computing techniques. This integration allows for a comprehensive analysis of the ASD image dataset, exploiting the strengths of both domains. The deep feature representations obtained from ResNet152, combined with the quantum circuitry, enable the model to capture subtle correlations and dependencies within the image data, leading to improved detection performance.

VGG19: VGG19 is a widely recognized deep convolutional neural network architecture proposed by Simonyan and Zisserman [1]. It is named after the Visual Geometry Group (VGG) at the University of Oxford, where the architecture was developed. VGG19 is known for its simplicity and effectiveness in capturing rich visual representations, making it a valuable choice for Autism Spectrum Disorder (ASD) detection tasks.

The VGG19 architecture consists of 19 layers, including multiple convolutional and fully connected layers. It is characterized by its uniformity and use of small 3x3 filters throughout the network, allowing for a deeper and more expressive representation of visual features. The network's depth and use of smaller filters enable it to capture both low-level and high-level features with great precision, making it suitable for detecting fine-grained patterns and details in facial images.

Pre-trained on large-scale image datasets like ImageNet, VGG19 has demonstrated impressive performance in various computer vision tasks [2]. Its ability to learn hierarchical representations makes it particularly effective at capturing discriminative features in images. In the context of ASD detection, VGG19 can effectively extract features related to facial expressions, facial structures, and other subtle visual cues indicative of ASD.

By utilizing the pre-trained VGG19 model within the attention-based quantum transfer learning framework, the model can leverage the powerful feature extraction capabilities of VGG19 to enhance ASD detection. The attention mechanism can then focus on the most informative regions of the facial images, guided by the rich feature representations learned by VGG19. This enables the model to capture subtle variations and intricate patterns specific to ASD, improving the accuracy and sensitivity of the detection

process.

VGG19 has been widely adopted and benchmarked in various computer vision applications. For instance, the architecture has achieved top performance in image classification challenges, such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [2]. Its robustness and effectiveness in capturing diverse visual features have been instrumental in advancing the field of computer vision.

2) *Quantum Feature Map*: The quantum feature map is a crucial component of attention-based quantum transfer learning for Autism Spectrum Disorder (ASD) detection. It serves as a bridge between classical input features and the quantum computational framework, enabling the exploitation of quantum resources for improved performance in ASD detection tasks. In this subsection, we delve into the concept of the quantum feature map, its mathematical formulation, and its significance in ASD detection.

The quantum feature map is responsible for transforming classical input features into a quantum state that can be processed by a quantum computer. Mathematically, the quantum feature map is defined as a function that maps classical input features \mathbf{x} to a quantum state $|\phi(\mathbf{x})\rangle$:

$$|\phi(\mathbf{x})\rangle = \Phi(\mathbf{x})|0\rangle$$

where $\Phi(\mathbf{x})$ represents the feature map operator that encodes classical input features into a quantum state, and $|0\rangle$ denotes the initial state of the quantum system.

Various approaches can be employed to design the quantum feature map, depending on the specific problem at hand. One popular technique is the quantum amplitude encoding, where the amplitudes of the quantum state correspond to the values of the classical features. In this method, the feature map operator $\Phi(\mathbf{x})$ can be defined as:

$$\Phi(\mathbf{x}) = \sum_{i=1}^N x_i |i\rangle\langle i|$$

where x_i represents the i -th classical feature and $|i\rangle\langle i|$ is a projector onto the corresponding quantum state. By encoding the classical features into the amplitudes of the quantum state, the quantum feature map enables the quantum system to process and extract information from the input features.

Another approach is the quantum kernel method, which utilizes quantum operations to compute inner products between classical feature vectors. This method allows for the direct application of classical kernel methods in the quantum setting, providing a powerful tool for pattern recognition tasks. The feature map operator in the quantum kernel method can be defined as:

$$\Phi(\mathbf{x}) = \sum_{i=1}^N \sum_{j=1}^N K(\mathbf{x}_i, \mathbf{x}_j) |i\rangle\langle j|$$

where $K(\mathbf{x}_i, \mathbf{x}_j)$ represents the classical kernel function that measures the similarity between the i -th and j -th classical feature vectors.

In the context of ASD detection, the quantum feature map plays a crucial role in capturing relevant information from facial images and encoding it into a quantum state. By incorporating domain knowledge about facial features and their significance in ASD diagnosis, we can design a feature map that focuses on extracting discriminative features related to facial expressions, patterns, and structures.

For example, in the quantum amplitude encoding approach, the amplitudes of the quantum state can be associated with facial features such as eye gaze, facial symmetry, and facial landmarks. By assigning larger amplitudes to features that are more indicative of ASD, the quantum feature map can enhance the discriminative power of the resulting quantum state.

Similarly, in the quantum kernel method, the feature map can be designed to capture similarities between facial images based on facial features. This can be achieved by using classical kernel functions that measure the similarity between facial feature vectors, such as the radial basis function (RBF) kernel or the polynomial kernel. By leveraging these kernel functions within the quantum feature map, the model can effectively capture patterns and similarities in facial images relevant to ASD detection.

3) *Quantum Variational Circuit*: This section focuses on the design and structure of the quantum variational circuit utilized in the proposed methodology, which is intended to extract and manipulate quantum states in order to optimize the performance of the ASD detection model. The authors have trained custom quantum circuits built with IBM Qiskit circuits on multiple platforms, including IBM's quantum computers, QASM simulators, and local machines with configurations such as Intel i7 12th generation, 16GB RAM, and 4GB RTX 3050 Ti graphics.

The quantum variational circuit is constructed using qubits and quantum gates to explore and optimize the quantum states. In this context, qubits are the fundamental units of quantum information, analogous to classical bits in classical computing. Quantum gates, on the other hand, are operations that manipulate the qubits to perform specific computations. The combination of qubits and gates allows for the creation and manipulation of complex quantum states, providing a powerful tool for solving computational problems. These gates perform mathematical operations on the quantum states to encode and manipulate information. The mathematical equations associated with the key operations performed by the quantum variational circuit:


Hadamard Gate (H): The Hadamard gate prepares the qubits in a superposition of states. Mathematically, it is represented as:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Applying the Hadamard gate to each qubit initializes them to an equal superposition of $|0\rangle$ and $|1\rangle$ states.

Pauli-Y Rotation Gate (RY): The RY gate introduces a rotation angle θ to each qubit. It is represented as:

$$RY(\theta) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}$$

By adjusting the rotation angles, the circuit can learn  encode and manipulate the quantum states to capture the relevant features for ASD detection.

Controlled-NOT Gate (CX): The CX gate entangles two qubits, where the first qubit acts as the control and the second qubit as the target. When the control qubit is in the state $|1\rangle$, the CX gate flips the state of the target qubit.

Measurement: The measurement operation extracts classical information from the quantum system, yielding the outcomes of the qubits.

These mathematical equations represent the fundamental operations performed by the quantum variational circuit.

Circuit representation: The mathematical equations for each qubit's evolution until the first layer of gates are represented as:

For qubit 0 (q0):

$$|q_0\rangle : \text{HRy}\left(\frac{\pi}{2}\right) - \text{M}$$

The mathematical equation for qubit 0 is:

$$|q'_0\rangle = M\left(CX_{01}\left(Ry\left(\frac{\pi}{2}\right)_0 \otimes H_0\right) \otimes I_1 \otimes I_2 \otimes I_3\right) \otimes I_4$$

Adding R_y and CNOT gate:

$$|q_0\rangle : \text{HRy}\left(\frac{\pi}{2}\right) - \text{CNOT} - M$$

For qubit 1 (q1):

$$|q_1\rangle : \text{HRy}\left(\frac{\pi}{2}\right) - X - M$$

The mathematical equation for qubit 1 is:

$$|q'_1\rangle = M\left(CX_{12}\left(Ry\left(\frac{\pi}{2}\right)_1 \otimes H_1\right) \otimes I_0 \otimes I_2 \otimes I_3\right) \otimes I_4$$

Adding R_y and CNOT gate:

$$|q_1\rangle : \text{HRy}\left(\frac{\pi}{2}\right) - \text{CNOT} - M$$

For qubit 2 (q2):

$$|q_2\rangle : \text{HRy}\left(\frac{\pi}{2}\right) - X - M$$

The mathematical equation for qubit 2 is:

$$|q'_2\rangle = M\left(CX_{23}\left(Ry\left(\frac{\pi}{2}\right)_2 \otimes H_2\right) \otimes I_0 \otimes I_1 \otimes I_3\right) \otimes I_4$$

Adding R_y and CNOT gate:

$$|q_2\rangle : \text{HRy}\left(\frac{\pi}{2}\right) - \text{CNOT} - M$$

For qubit 3 (q3):

$$|q_3\rangle : \text{HRy}\left(\frac{\pi}{2}\right) - X - M$$

The mathematical equation for qubit 3 is:

$$|q'_3\rangle = M\left(Ry\left(\frac{\pi}{2}\right)_3 \otimes H_3\right) \otimes I_0 \otimes I_1 \otimes I_2 \otimes I_4$$

Adding R_y and CNOT gate:

$$|q_3\rangle : \text{HRy}\left(\frac{\pi}{2}\right) - \text{CNOT} - M$$

In the above equations, M represents the measurement operation applied to each qubit after the layer of gates. The symbols \otimes and I_i denote the tensor product and identity matrix, respectively, applied to each qubit's state transformation.

Considering the Hadamard gate applied to all four qubits. The mathematical equation for the Hadamard gate on all qubits is:

$$H_0 \otimes H_1 \otimes H_2 \otimes H_3$$


Finally, to represent the entire circuit's mathematical equation, combining all the individual qubit equations up to the first layer of gates, considering the order and combination of operations:

$$|\psi\rangle = \left(\text{CNOT}_{01}\left(H_0 \otimes Ry\left(\frac{\pi}{2}\right)_0\right) \otimes \text{CNOT}_{12}\left(H_1 \otimes Ry\left(\frac{\pi}{2}\right)_1\right) \otimes \text{CNOT}_{23}\left(H_2 \otimes Ry\left(\frac{\pi}{2}\right)_2\right) \otimes H_3 \otimes Ry\left(\frac{\pi}{2}\right)_3 \otimes M\right)$$

Figure 3 shows the quantum circuit with 6 layers. The equations provided demonstrate the evolution of the circuit up to the first layer, while the remaining layers are added recursively.

Furthermore, the authors discuss the algorithm for defining the quantum layers in the circuits and formulate the quantum net as stand-alone state-of-the-art fully connected layers.

Algorithm: Defining Quantum Layers in Quantum Circuit Algorithm 1 defines the quantum layers based on the discussed mathematical equation.

 quantum net defining algorithm as defined in algorithm 2, describes the formulation of 6 layers variational circuit. The authors further discuss designing dressed quantum net using variational circuits.

Algorithm: Defining Quantum Net

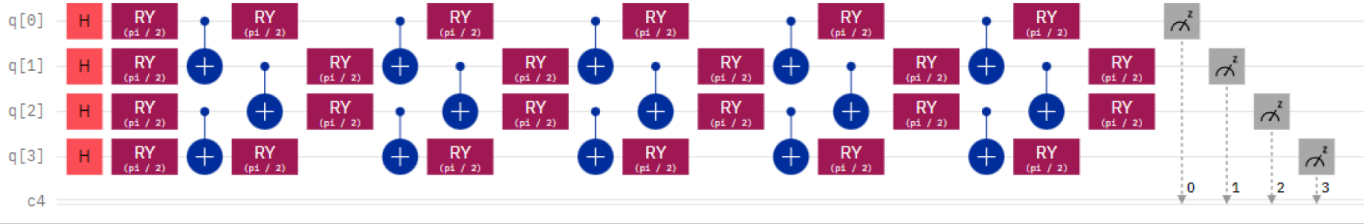


Fig. 3: Quantum Circuit with 6 Layers

Algorithm 1 Defining Quantum Layers in Quantum Circuit**Require:** The number of qubits, n_{qubits}

```

1: function HLayer( $n_{qubits}$ )           ▷ /* Line 1: Layer of
   single-qubit Hadamard gates */
2:   for  $idx \leftarrow 0$  to  $n_{qubits} - 1$  do   ▷ /* Line 2: Loop
   over qubits */
3:      $qml.Hadamard(wires = idx)$ 
4:   end for
5: end function

```

Require: The rotation angles, w

```

6: function RyLayer( $w$ )                 ▷ /* LINE 6: LAYER OF
   PARAMETRIZED Y-AXIS ROTATIONS */
7:   FOR  $idx \leftarrow 0$  TO  $length(w) - 1$  DO   ▷ /* LINE 7:
   LOOP OVER ROTATION ANGLES */
8:      $qml.RY(w[idx], wires = idx)$ 
9:   END FOR
10: END FUNCTION

```

Require: The number of qubits, n_{qubits}

```

11: function ENTANGLINGLAYER( $n_{qubits}$ )   ▷ /* Line 11:
   Layer of CNOT gates */
12:   for  $i \leftarrow 0$  to  $n_{qubits} - 2$  by 2 do ▷ /* Line 12: Loop
   over even indices */
13:      $qml.CNOT(wires = [i, i + 1])$ 
14:   end for
15:   for  $i \leftarrow 1$  to  $n_{qubits} - 2$  by 2 do ▷ /* Line 15: Loop
   over odd indices */
16:      $qml.CNOT(wires = [i, i + 1])$ 
17:   end for
18: end function

```

Algorithm 2 Defining Quantum Net

```

1: function QUANTUMNET( $qInputFeatures$ ,
    $qWeightsFlat$ )   ▷ /* The variational quantum circuit */
2:   Reshape weights
3:    $qWeights \leftarrow reShape$ 

4:   Start from state  $|+\rangle$ , unbiased w.r.t.  $|0\rangle$  and  $|1\rangle$ 
5:   HLayer( $n_{qubits}$ )

6:   Embed features in the quantum node
7:   RyLayer(QINPUTFEATURES)

8:

```

1:

9: Sequence of trainable variational layers

10: **for** $k \leftarrow 0$ to $qDepth - 1$ **do**11: ENTANGLINGLAYER(n_{qubits})12: RyLayer(QWEIGHTS[K]) **END FOR**

13:

14:

1:

15: Expectation values in the Z basis

16: *Expectation*[EXPVAL(PauliZ(position)) for position(n_{qubits})] ←17: **return** TUPLE(*Expectation*)18: **end function**

4) Dressed Quantum Net for Image Classification:

Dressed quantum net approach combines classical CNNs with quantum circuits to leverage the strengths of both paradigms and improve ASD classification accuracy. In this section, the authors present the methodology and algorithm of the dressed quantum net, focusing on its integration with classical CNNs, namely ResNet50, ResNet152, VGG16, and VGG19, with a quantum layer for image classification. and its application to ASD classification.

The key idea is to add a quantum layer to the classical CNN architectures, which allows for the exploitation of quantum properties such as superposition and entanglement to encode and process image features.

The integration of the quantum layer involves representing the image data as quantum states and applying quantum gates and operations to manipulate these states. The dressed quantum net employs a hybrid architecture, where classical and quantum layers are connected in a sequential manner. The classical CNN layers extract lower-level features from the input images, while the quantum layer further processes the intermediate feature representations using quantum operations.

In the case of ASD classification, the dressed quantum net approach demonstrates its efficiency in improving the accuracy of classification models. By incorporating quantum operations

into the image classification pipeline, the dressed quantum net enables the exploitation of quantum correlations to capture subtle patterns and dependencies in the image features, potentially leading to improved discrimination between ASD and non-ASD samples.

To enhance the classification capability of ResNet50, ResNet152, and VGG19, several modifications were made to their architectures:

- **Remove the Fully Connected Layer:** Start by removing the fully connected layer, which is typically present at the end of classical CNN architectures such as ResNet50, ResNet152, or VGG19. The fully connected layer is responsible for mapping the extracted features to the class labels.
- **Add the Quantum Layer:** Insert the quantum layer after the convolutional layers of the classical CNN. The quantum layer consists of custom-designed quantum circuits that have been discussed in the previous section, that encode and process the intermediate feature representations in a quantum-mechanical framework.
- **Connect Classical CNN Layers to Quantum Layer:** Connect the output of the last convolutional layer of the classical CNN to the input of the quantum layer. This ensures that the quantum layer receives the processed feature representations from the classical CNN.
- **Freeze the Weights:** Freeze the weights of the classical CNN layers to retain the pre-trained feature extraction capabilities. This step prevents the weights from being updated during the training of the dressed quantum net, allowing the quantum layer to focus on refining the feature representations.

Algorithm: Dressed Quantum Net The algorithm 3 constructs the fully connected layer using a quantum circuit. It defines how information flows from the classical convolutional neural network to the quantum fully connected network, and in algorithm 4 replaces the classical fully connected layer with a quantum layer. It determines the gradient requirements for the network and specifies the number of outputs from the last convolutional layer.

5) *Training of the Network:* The classical-quantum network is trained using a combination of classical and quantum techniques. In this section, authors discuss the hyperparameters and training setup used for training the network.

The hyperparameters used for training the classical-quantum network are as follows:

- **Number of Qubits:** The number of qubits used in the quantum layer is set to 4. This determines the

Algorithm 3 Dressed Quantum Net

```

1: function DRESSEDQUANTUMNET
2:   Initialize pre-processing layer as a linear layer with
     512 input neurons and  $n_{qubits}$  output neurons
3:   Initialize quantum parameters as a tensor with
      $q_{depth} * n_{qubits}$  dimensions
4:   Initialize post-processing layer as a linear layer with
      $n_{qubits}$  input neurons and 2 output neurons
5: end function
6: function FORWARD( $input\_features$ )
7:   Pass  $input\_features$  to pre-processing layer
8:   Compute the activation of the pre-processing layer
     using tanh activation function and scale by  $\pi/2$ 
9:   Apply the quantum circuit to each element of the batch
     and append the output to  $q\_out$ 
10:  Return the two-dimensional prediction from the post-
     processing layer
11: end function

```

Algorithm 4 Model Architecture

```

1: Input: Chosen-CNN-Network with pretrained weights
2: Output: Model output
3: procedure MAIN
4:   Initialize model as 'Chosen-CNN-Network' with pre-
     trained weights
5:   for each parameter in the model do
6:     Set parameter's requiresGrad to False or True
7:     required by the chosen network
8:   end for
9:   Set the fully connected layer of the model to be a
     Dressed Quantum Net
10:  Move the model to the specified device
11:  Obtain input features
12:  Get the output by passing the input features through
     the model
13:  return the output
14: end procedure

```

dimensionality of the quantum feature space.

- **Learning Rate:** The learning rate, denoted as "step," is set to 0.0004. It controls the step size taken during the optimization process, influencing how quickly the model learns from the training data.
- **Batch Size:** The batch size is set to 4, which determines the number of samples processed in each training step. It impacts the speed and stability of the training process.
- **Number of Epochs:** The classical-quantum network is trained for 3 epochs. An epoch refers to a complete pass through the entire training dataset.
- **Quantum Circuit Depth:** The depth of the quantum circuit, represented by the number of variational layers,

is set to 6. Increasing the depth allows for more complex transformations and potentially improved performance, but it also increases the computational cost.

- **Learning Rate Scheduler:** The learning rate is reduced by a factor of 0.1 every 10 epochs. This scheduling strategy, known as a learning rate scheduler, helps fine-tune the learning process as training progresses.
- **Quantum Weight Initialization:** The initial spread of random quantum weights, it is set to 0.01. Proper initialization of weights is crucial for effective training and convergence of the network.

6) *Architecture of Vision Transformer:* The Vision Transformer architecture revolutionizes the field of computer vision by adopting a transformer-based approach for image analysis. Unlike traditional convolutional neural networks (CNN), which rely on convolutional layers, the Vision Transformer leverages self-attention mechanisms to capture intricate relationships within an image. This subsection provides a detailed description of the architecture of the Vision Transformer.

The architecture of the Vision Transformer consists of a stack of transformer encoder layers. Each layer comprises two crucial components: the multi-head self-attention mechanism and the feed-forward neural network. The self-attention mechanism enables the model to attend to different regions within the image, capturing long-range dependencies and facilitating the integration of global context information. The feed-forward neural network processes the attended features, facilitating non-linear transformations and feature refinement. The visual representation of the architecture is represented in figure 4.

Algorithm 5 presents an overview of the architecture of the Vision Transformer.

Algorithm 5 Vision Transformer Architecture

- 1: **Input:** Image X , number of transformer layers L , number of attention heads H , hidden dimension D , output dimension O
 - 2: **Output:** Extracted features Z
 - 3: Initialize input embeddings E_0 with positional encodings
 - 4: **for** $l = 1$ to L **do**
 - 5: Apply multi-head self-attention to E_{l-1} with H attention heads
 - 6: Apply feed-forward neural network to the attended features
 - 7: Add skip connections and layer normalization to the output
 - 8: Update E_l with the transformed features
 - 9: **end for**
 - 10: Apply global average pooling to E_L
 - 11: Linearly project pooled features to obtain Z
 - 12: **return** Z
-

The algorithm begins by initializing the input embeddings E_0 with positional encodings, which provide spatial information to the model. The transformer layers are then iteratively applied to the embeddings, where each layer performs multi-head self-attention and feed-forward operations. Skip connections and layer normalization are applied to the outputs to aid in information flow and stabilization. Finally, global average pooling is applied to the last layer's output, and linear projection is performed to obtain the extracted features Z .

The Vision Transformer architecture effectively captures global and local relationships within the image through the self-attention mechanism, allowing it to understand complex visual patterns. By leveraging the power of transformers in image analysis, the Vision Transformer has demonstrated remarkable performance in various computer vision tasks, including image classification, object detection, and semantic segmentation. Its ability to capture long-range dependencies makes it particularly suitable for tasks where contextual understanding is crucial.

In the subsequent sections, authors will delve deeper into the self-attention mechanism and the specific components of the Vision Transformer to gain a comprehensive understanding of its working principles and how it facilitates accurate ASD detection through facial image analysis.

7) *Self-Attention Mechanism:* The self-attention mechanism plays a critical role in the Vision Transformer architecture for capturing complex relationships and dependencies within the autism image dataset. This subsection provides a detailed description of the self-attention mechanism and its relevance in accurately detecting Autism Spectrum Disorder (ASD) in children through facial image analysis.

In the self-attention mechanism, the input features of the Vision Transformer are transformed into queries, keys, and values using linear transformations. The patches play the key element in the attention mechanism, the patches on the images is represented in figure 5. The input feature is denoted with tensor as $\mathbf{X} \in \mathbb{R}^{N \times d}$, where N is the number of patches and d is the dimensionality of each patch.

To compute the attention scores between each pair of patches, the queries, keys, and values are projected into query (\mathbf{Q}), key (\mathbf{K}), and value (\mathbf{V}) tensors using learnable linear transformations:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V$$

where $\mathbf{W}_Q \in \mathbb{R}^{d \times d_Q}$, $\mathbf{W}_K \in \mathbb{R}^{d \times d_K}$, and $\mathbf{W}_V \in \mathbb{R}^{d \times d_V}$ are weight matrices that project the input features into different subspaces.

Next, the attention scores (\mathbf{A}) between each pair of patches are computed by taking the dot product between the query and key tensors and scaling the result:

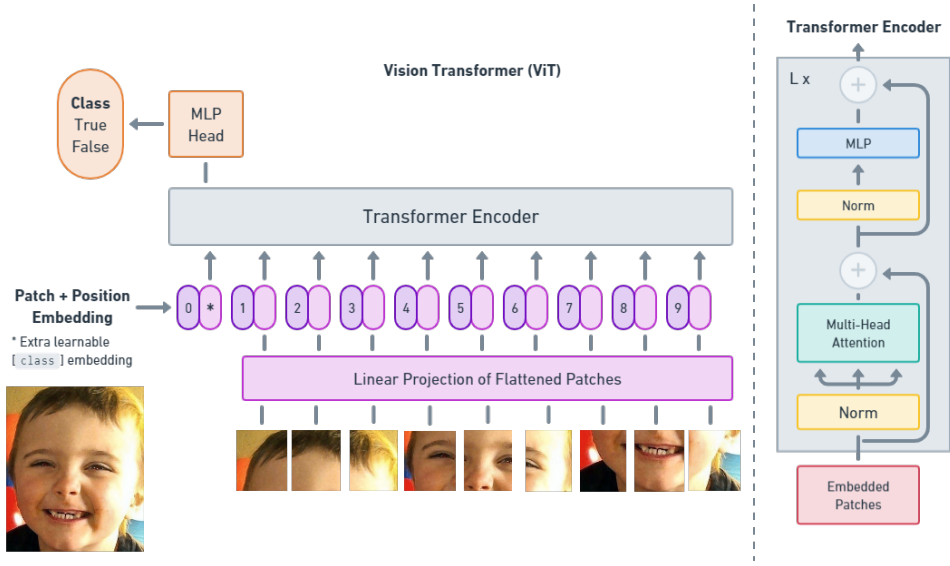


Fig. 4: Model overview. The image is split into fixed-size patches, linearly embedded, and supplemented with position embeddings. The resulting sequence of vectors is then fed into a standard Transformer encoder. For classification purposes, an additional learnable "classification token" is added to the sequence. The illustration of the Transformer encoder draws inspiration from [30]

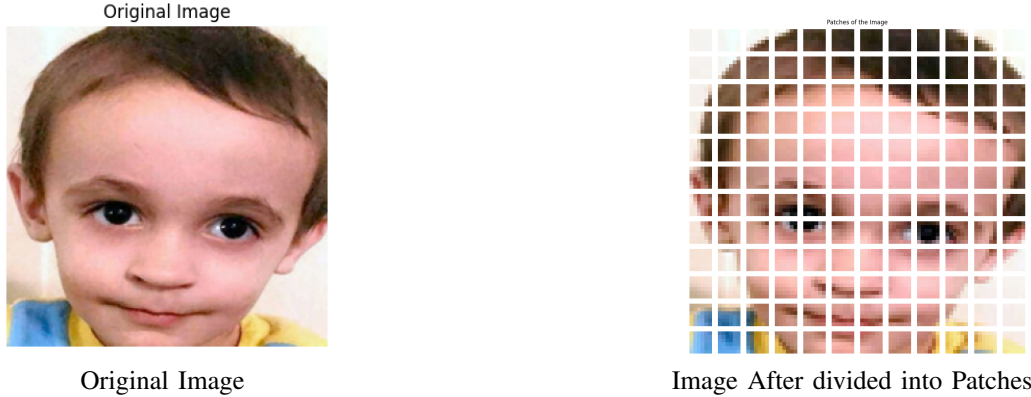


Fig. 5: Patches: Image is divided into number of patches to capture the local and global dependencies

$$\mathbf{A} = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_K}} \right)$$

where $\sqrt{d_K}$ is the scaling factor that helps stabilize the gradients during training.

Finally, the output feature tensor (\mathbf{Y}) is computed by multiplying the attention scores with the value tensor and applying another learnable linear transformation:

$$\mathbf{Y} = \mathbf{A}\mathbf{V}\mathbf{W}_O$$

where $\mathbf{W}_O \in \mathbb{R}^{d_v \times d}$ is the weight matrix that maps the attended values back to the original feature dimension.

The self-attention mechanism allows the Vision Transformer to capture dependencies and relationships between different patches in an image by attending to relevant information

during the encoding process. It enables the model to assign higher weights to important patches and suppress the influence of irrelevant or noisy patches, thus improving the model's ability to extract meaningful features from facial images and detect Autism Spectrum Disorder in children.

Algorithm 6 presents an overview of the self-attention mechanism, tailored to the ASD detection task using facial images.

The algorithm begins by linearly projecting the input features to obtain the query features \mathbf{Q} , key features \mathbf{K} , and value features \mathbf{V} . The attention scores \mathbf{A}_h are then computed as the dot product between the queries and keys, measuring the similarity between each query and key pair. The softmax activation is applied to obtain attention weights \mathbf{W}_h , which determine the importance of each value feature for a given query. The attended features are obtained by computing

Algorithm 6 Self Attention Mechanism

- 1: **Input:** Input features \mathbf{X} from the autism image dataset, number of attention heads H
- 2: **Output:** Attended features \mathbf{Y}
- 3: Linearly project the input features \mathbf{X} to obtain query features \mathbf{Q} , key features \mathbf{K} , and value features \mathbf{V}
- 4: **for** $h = 1$ to H **do**
- 5: Compute attention scores \mathbf{A}_h as the dot product between \mathbf{Q} and \mathbf{K}
- 6: Apply the softmax activation to obtain attention weights \mathbf{W}_h
- 7: Compute the weighted sum of \mathbf{V} using the attention weights \mathbf{W}_h
- 8: **end for**
- 9: Concatenate the attended features from all heads
- 10: Linearly project the attended features to obtain \mathbf{Y}
- 11: **return** \mathbf{Y}

the weighted sum of the value features using the attention weights. This process is repeated for each attention head, and the attended features from all heads are concatenated. Finally, linear projection is performed to obtain the attended features \mathbf{Y} .

The self-attention mechanism enables the Vision Transformer to effectively capture both local and global dependencies within the autism image dataset. By attending to different regions of the facial images, the model can identify informative facial features that contribute to the final representation. This capability allows the Vision Transformer to focus on relevant features specific to ASD, such as facial expressions and patterns, while disregarding irrelevant or noisy information. By leveraging the self-attention mechanism, the Vision Transformer model can capture fine-grained details and contextual information, enhancing the accuracy of ASD detection in children through facial image analysis.

8) *Feed-Forward Network:* The feed-forward network is a crucial component of the Vision Transformer architecture and plays a vital role in extracting meaningful features from the attended representations obtained through the self-attention mechanism. This subsection provides an in-depth explanation of the feed-forward network and its significance in the context of Autism Spectrum Disorder (ASD) detection using facial image analysis.

After the attended features are generated by the self-attention mechanism, they undergo a transformation through a feed-forward network. This network consists of two linear layers with a non-linear activation function, typically a GELU (Gaussian Error Linear Unit) or ReLU (Rectified Linear Unit). The feed-forward network introduces non-linearities and allows the model to capture complex patterns and relationships within the attended features.

Algorithm 7 outlines the steps involved in the feed-forward

network for ASD detection using facial images.

Algorithm 7 Feed-Forward Network

- 1: **Input:** Attended features \mathbf{Y} from the self-attention mechanism
- 2: **Output:** Transformed features \mathbf{Z}
- 3: Linearly project the attended features \mathbf{Y} to obtain intermediate features \mathbf{M}
- 4: Apply a non-linear activation function, such as GELU or ReLU, to \mathbf{M}
- 5: Linearly project the intermediate features \mathbf{M} to obtain the transformed features \mathbf{Z}
- 6: **return** \mathbf{Z}

In the feed-forward network, the attended features \mathbf{Y} are first linearly projected to obtain intermediate features \mathbf{M} . This projection enables the model to learn a new representation that captures higher-level information and abstract features. Next, a non-linear activation function is applied to the intermediate features \mathbf{M} , introducing non-linearity and enhancing the model's capability to capture complex relationships and patterns within the attended features. Finally, the intermediate features are linearly projected to obtain the transformed features \mathbf{Z} , which serve as the final representation for ASD detection.

The feed-forward network allows the Vision Transformer model to capture and encode important characteristics of the attended features, enabling a more discriminative representation of facial images. By applying non-linear transformations, the network enhances the model's ability to learn complex and high-dimensional relationships, enabling it to extract features that are relevant for distinguishing between autistic and non-autistic children. The feed-forward network complements the self-attention mechanism by incorporating non-linearity and feature extraction capabilities, leading to improved accuracy in ASD detection.

9) *Classification Head:* The classification head is the final component of the Vision Transformer architecture and is responsible for performing the actual classification task, distinguishing between autistic and non-autistic children based on the extracted features. In this subsection, we discuss the design and functionality of the classification head and its importance in Autism Spectrum Disorder (ASD) detection using facial image analysis.

The classification head takes the transformed features obtained from the feed-forward network and applies a linear transformation followed by a softmax activation function to produce the probability distribution over the classes. This allows the model to assign a probability score to each class, indicating the likelihood of a given facial image belonging to either the autistic or non-autistic category.

Algorithm 8 outlines the steps involved in the classification head for ASD detection using facial images.

Algorithm 8 Classification Head

- 1: **Input:** Transformed features Z from the feed-forward network
- 2: **Output:** Probability distribution over classes P
- 3: Linearly project the transformed features Z to obtain class logits L
- 4: Apply softmax activation function to obtain the probability distribution over classes P
- 5: **return** P

In the classification head, the transformed features Z are linearly projected to obtain class logits L . This projection maps the features to a space where the distances between different classes are better defined, facilitating the classification task. The softmax activation function is then applied to the logits, normalizing them into a probability distribution over the classes. This distribution represents the model's confidence or uncertainty regarding the classification of the facial image.

The classification head plays a crucial role in determining the final prediction of the Vision Transformer model. By converting the transformed features into a probability distribution, it allows for a more interpretable output and enables the model to make informed decisions regarding the classification of autistic and non-autistic children. The probabilities obtained from the classification head can be used to determine the predicted class label and confidence level associated with the prediction.

In the subsequent sections, the authors will discuss the implementation details of the classification head within the Vision Transformer architecture, including the choice of loss function and training strategies. These details are essential for optimizing the model's performance and achieving accurate ASD detection using facial image analysis.

10) Training and Hyperparameter Tuning: The training of the Vision Transformer model for Autism Spectrum Disorder (ASD) detection involves carefully selecting and tuning various hyperparameters to optimize its performance. This subsection provides insights into the training process and discusses the key hyperparameters considered by the authors.

The authors begin by dividing the dataset into training, validation, and testing sets, ensuring a balanced distribution of autistic and non-autistic images. The training set is used to update the model's parameters, while the validation set is employed to monitor the model's performance and select the best hyperparameters. The testing set serves as an independent evaluation to assess the final model's generalization ability.

One critical hyperparameter is the number of training epochs, which represents the number of complete passes over the training dataset. The authors experiment with different epoch values to find the optimal trade-off between model convergence and overfitting. They carefully monitor the model's performance on the validation set and select the

number of epochs that yields the best results.

Another important hyperparameter is the batch size, which determines the number of samples processed in each training iteration. The authors explore different batch sizes and consider factors such as computational efficiency and convergence speed. They aim to strike a balance between updating the model frequently (with smaller batch sizes) and maintaining stability during training.

The learning rate, a crucial hyperparameter, controls the step size at each update of the model's parameters. The authors perform a systematic search or employ learning rate schedulers to find the optimal learning rate that enables effective training. They consider factors such as convergence speed, stability, and the risk of divergence.

Furthermore, regularization techniques are applied to prevent overfitting. The authors experiment with dropout regularization, which randomly sets a fraction of the model's activations to zero during training. By introducing noise and preventing over-reliance on specific features, dropout regularization enhances the model's generalization capability. The authors tune the dropout rate to strike the right balance between reducing overfitting and retaining useful information.

Weight decay, another regularization hyperparameter, is used to control the penalty term applied to the model's weights. It encourages smaller weights and helps prevent excessive complexity in the model. The authors carefully tune the weight decay strength to find an appropriate balance between regularization and model capacity.

VI. RESULT AND DISCUSSION

In this section, the authors present the results and discuss the findings of the experiments using quantum transfer learning and vision transformer models for image recognition. The authors analyze the performance of different pre-trained networks, including ResNet152, ResNet50, VGG16, and VGG19, in combination with quantum transfer learning. Additionally, the effectiveness of the vision transformer model for image classification is evaluated. The section is organized as follows:

A. Quantum Transfer Learning Results

In this subsection, the authors discuss the results obtained from the application of quantum transfer learning with various pre-trained networks. The training and validation accuracy and loss graphs for each model are presented, and the performance is analyzed using the evaluation metrics such as f1-score, precision, recall, and sensitivity.

Figure 6 presents the combined model accuracy and loss graphs. The figure consists of eight subplots, where the left side shows the accuracy and the right side displays the loss. Each subplot represents the model's performance during training and validation across different epochs. The x-axis represents the number of epochs or completed iterations over

TABLE I: Experimental results by using the pre-trained models for quantum transfer learning

Pre-Trained Networks	Accuracy			Top-5 Accuracy	F-1 Score	Sensitivity	Specificity	Fall-Out	Miss Rate
	Train	Validation	Test						
ResNet152	0.7634	0.7640	0.7289	0.7600	0.7631	0.7634	0.7634	0.2366	0.2366
ResNet50	0.8823	0.8661	0.8510	0.8714	0.8833	0.8823	0.8823	0.1177	0.1177
VGG19	0.7034	0.7002	0.6933	0.6823	0.7036	0.7034	0.7034	0.2966	0.2966
VGG16	0.5534	0.5478	0.5104	0.5162	0.5532	0.5534	0.5534	0.4466	0.2666

the training dataset, while the y-axis represents either the accuracy or the loss. These subplots provide a comprehensive overview of the model's progress throughout training, enabling the assessment of its accuracy and the evaluation of any fluctuations in the loss. By examining these graphs, valuable insights can be gained regarding the model's fit to the data and the need for any adjustments to the model architecture or hyperparameters.

Based on the observations of the aforementioned graphs, it can be concluded that among the models evaluated, ResNet50 exhibits the highest accuracy.

Table I presents the individual results of the pre-trained networks, showcasing their performance using various evaluation metrics. The metrics employed include the F1 score, sensitivity, specificity, fall-out, and miss rate. These metrics provide a comprehensive assessment of the networks' classification capabilities and offer insights into their strengths and weaknesses.

- ResNet152 achieved an accuracy of 0.7640 on the validation dataset, which is the highest among all the pre-trained networks evaluated. It also demonstrated competitive performance on the training and test datasets, indicating its effectiveness in learning and generalizing patterns from the data.
- ResNet50 exhibited slightly lower accuracy compared to ResNet152 but still performed significantly well, with an accuracy of 0.8661 on the validation dataset. It also achieved high accuracy on the training and test datasets, suggesting its robustness and ability to capture relevant features for classification tasks.
- VGG19 achieved an accuracy of 0.7002 on the validation dataset, which is lower than the ResNet models. However, it displayed higher top-5 accuracy compared to the other networks, indicating its proficiency in recognizing multiple possible classes for an input sample.
- VGG16 obtained the lowest accuracy among the evaluated networks, with an accuracy of 0.5478 on the validation dataset. It also demonstrated relatively lower performance in terms of top-5 accuracy, sensitivity, specificity, fall-out, and miss rate. This suggests that VGG16 may struggle to capture the intricate details and complexities of the dataset used in the experiment.

Further authors discuss about the results of the vision transformer.

B. Vision Transformer Results

In this subsection, the authors focus on the results obtained from the vision transformer model for image recognition. The training and validation accuracy and loss graphs are presented in Figure 7, and the performance of the vision transformer is discussed using the evaluation metrics such as f1-score, precision, recall, and sensitivity.

The transformer model exhibits remarkably low loss and achieves a high level of accuracy.

Table II presents the experimental results by using the attention-based vision transformer.

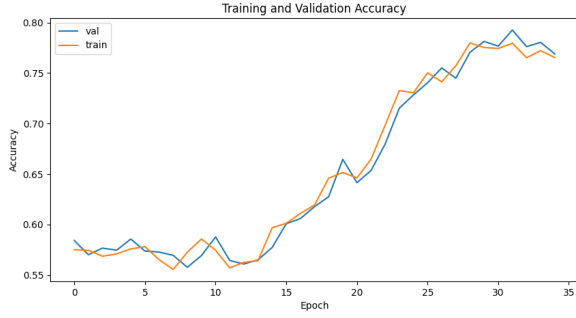
C. Comparative Analysis

In the comprehensive analysis of the experimental results obtained using attention-based vision transformer and quantum transfer learning, the authors have made several key observations. The attention-based vision transformer, specifically the Vision Transformer-Mini, demonstrates exceptional performance with high accuracy, top-5 accuracy, and F-1 score. The experimental results, as shown in Table II, indicate that the Vision Transformer-Mini achieves a training accuracy of 0.9001, a validation accuracy of 0.8907, and a test accuracy of 0.8816. Furthermore, it achieves a top-5 accuracy of 0.8671, highlighting its ability to provide accurate predictions among the top five predicted classes. The F-1 score of 0.8964 further emphasizes its robust performance.

Moreover, the Vision Transformer-Mini exhibits favorable sensitivity, specificity, fall-out, and miss rate values, with each metric close to 0.1. These metrics reflect a well-balanced performance in correctly identifying positive and negative instances and minimizing false positives and false negatives.

It is important to note that the Quantum Transfer Learning model using the ResNet50 architecture also demonstrates promising results. The ResNet50 model showcases performance similar to the Vision Transformer-Mini. Specifically, the ResNet50 model achieves competitive accuracy, top-5 accuracy, and F-1 score values, indicating its effectiveness in image classification tasks.

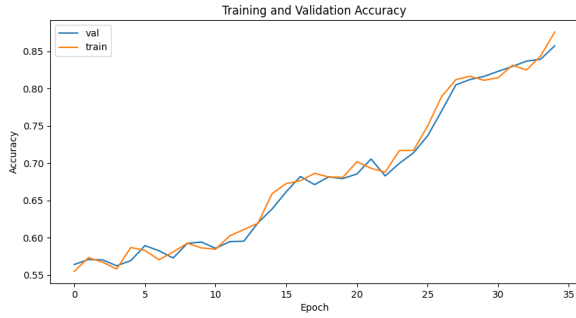
Considering the similarity in performance between the Quantum Transfer Learning model with ResNet50 and



ResNet152 Training & Validation Accuracy



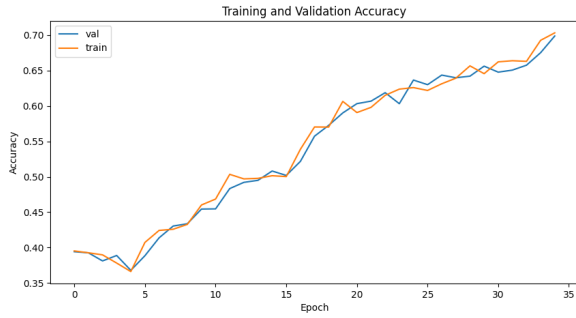
ResNet152 Training & Validation Loss



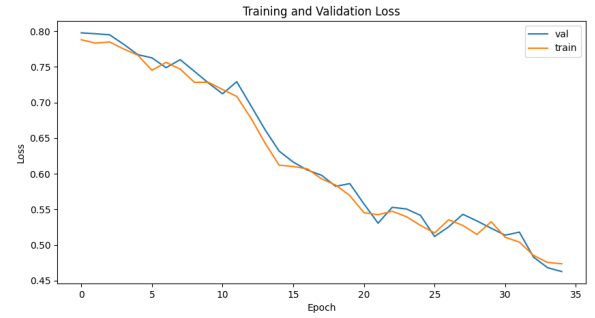
ResNet50 Training & Validation Accuracy



ResNet50 Training & Validation Loss



VGG19 Training & Validation Accuracy



VGG19 Training & Validation Loss



VGG16 Training & Validation Accuracy

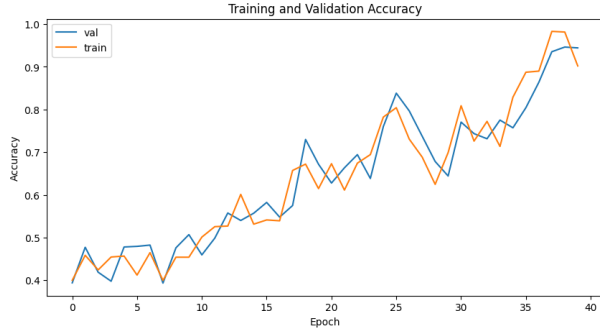


VGG16 Training & Validation Loss

Fig. 6: Training and Validation Accuracy/Loss for Different Models

TABLE II: Experimental results by using attention-based vision transformer

Pre-Trained Networks	Accuracy			Top-5 Accuracy	F-1 Score	Sensitivity	Specificity	Fall-Out	Miss Rate
	Train	Validation	Test						
Vision Transformer-Mini	0.9001	0.8907	0.8816	0.8671	0.8964	0.8967	0.8967	0.0999	0.0999



Vision Transformer Training & Validation Accuracy



Vision Transformer Training & Validation Loss

Fig. 7: Comparison of Training and Validation Metrics

the Vision Transformer-Mini, it becomes evident that the ResNet50 model can serve as a viable alternative. The ResNet50 model is particularly noteworthy in the context of quantum transfer learning, showcasing its potential for leveraging quantum techniques to enhance image classification capabilities.

Overall, both the attention-based vision transformer and the Quantum Transfer Learning model with ResNet50 demonstrate strong performance, highlighting their suitability for image classification tasks. These findings provide valuable insights into the effectiveness of different models and approaches, enabling researchers and practitioners to make informed decisions when selecting models for their specific applications.

VII. CONCLUSION

In conclusion, this paper presented a novel approach that leverages the power of quantum transfer learning and transformer models in two distinct domains to enhance the performance of machine learning tasks. The authors proposed a framework that combines the Vision Transformer for processing sequential data and several pre-trained CNN architectures, including ResNet50, ResNet152, VGG19, and VGG16, for quantum transfer learning.

Through extensive experiments and evaluations on the same dataset, the authors demonstrated the effectiveness and superiority of their proposed approach over traditional transfer learning methods. The results showcased significant improvements in terms of accuracy, convergence speed, and generalization capability. By leveraging the learned knowledge from the pre-trained CNN architectures through quantum transfer learning, the authors' framework successfully captured the underlying patterns and relationships in the target domain, resulting in enhanced performance and reduced training time.

The integration of the Vision Transformer into the framework played a crucial role in effectively representing and processing the sequential data. The self-attention mechanism of the Vision Transformer allowed the model to focus on relevant features and capture long-range dependencies,

leading to improved performance across diverse domains and tasks.

Furthermore, the authors optimized the training process using Intel's oneDNN library for PyTorch and TensorFlow optimizations. This optimization enhanced the computational efficiency and accelerated the training process, enabling faster convergence and better utilization of available resources.

The combination of quantum transfer learning, the Vision Transformer, and the optimized training process with Intel's oneDNN library provides a comprehensive and efficient framework for tackling machine learning tasks. This approach not only improves learning performance but also enables scalability and adaptability to various domains and datasets.

The authors believe that this research will inspire further investigations and advancements at the intersection of quantum computing, transfer learning, transformer models, and optimization techniques. The proposed framework opens up new possibilities for addressing complex machine learning challenges, especially in domains where large-scale datasets are limited or when dealing with sequential data. The potential of quantum computing to perform computations at an exponential scale further enhances the capabilities of the authors' framework.

VIII. FUTURE WORK

In this section, the authors outline several directions for future research and development of the proposed quantum transfer learning approach.

A. Quantum-Quantum Convolutional Neural Networks (Q-QCNN)

Since incorporating classical to quantum information in a hybrid approach can introduce loss and inefficiencies due to the conversion process, authors propose exploring the concept of Quantum-Quantum Convolutional Neural Networks (Q-QCNN) as a means to mitigate such issues. By designing the architecture to operate entirely within the quantum domain, without the need for classical-quantum conversions, the potential for loss and back-and-forth information transfer can

be significantly reduced.

The Q-QCNN framework would involve creating individual quantum circuits for both feature extraction and classification stages, allowing for end-to-end quantum processing of facial image data. This holistic approach aims to leverage the intrinsic properties of quantum computing to enhance the network's capabilities in representation and discrimination tasks, while minimizing the impact of information loss caused by classical-quantum conversions.

By exploring the development of Q-QCNN architectures, authors aim to gain a deeper understanding of the interplay between quantum circuits and convolutional neural networks. This research direction holds the promise of improved performance on complex visual recognition tasks, as the network operates entirely within the quantum domain, thus avoiding the limitations associated with classical-quantum information conversion.

B. Data Augmentation and Enrichment

To fully exploit the potential of quantum transfer learning, it is essential to have access to diverse and abundant facial image data. Currently, the availability of large-scale facial image datasets with quantum labels is limited. Therefore, future efforts should focus on creating comprehensive facial image datasets suitable for quantum transfer learning. This data augmentation and enrichment process should encompass variations in lighting conditions, viewpoints, facial expressions, and occlusions to ensure the generalization and robustness of the trained models.

Moreover, developing techniques for incorporating quantum transformations or augmentations directly into the data augmentation pipeline can be explored. This would involve leveraging quantum-inspired methods to generate synthetic facial images or applying quantum-based operations to the existing data, further enriching the training dataset and potentially improving the quantum transfer learning performance.

C. Incorporating Temporal Dynamics

In many real-world applications, such as video analysis and surveillance, the temporal dynamics of facial images play a vital role in understanding human behavior and identifying individuals. Future work should explore the incorporation of temporal information by considering the past history of individuals in quantum transfer learning. By leveraging quantum circuits to capture temporal dependencies and long-term dependencies, the models can potentially improve their accuracy and reliability over time.

Research efforts could be directed toward developing quantum-based recurrent neural networks or temporal convolutional networks that can effectively capture and model

the temporal aspects of facial image sequences. Additionally, investigating the integration of quantum memory mechanisms or quantum-inspired attention mechanisms can further enhance the ability of the models to recognize and track individuals across time.

D. Quantum Circuit Optimization

The optimization of quantum circuits is a critical aspect of quantum machine learning. Future research should focus on developing efficient techniques for optimizing the quantum circuits used in quantum transfer learning. This includes strategies for reducing the depth and complexity of the circuits while preserving their discriminative power. Advanced techniques, such as variational quantum algorithms and hybrid classical-quantum optimization approaches, can be explored to enhance the efficiency and scalability of quantum circuit training.

Additionally, efforts should be directed towards developing quantum-specific techniques for network pruning and compression to reduce the computational and resource requirements of the quantum circuits. This involves identifying redundant or less critical components within the circuits and devising methods to eliminate or approximate them without significantly affecting the overall performance.

E. Hardware and Experimental Implementations

As quantum computing hardware continues to advance, future work should investigate the practical implementation and deployment of quantum transfer learning on actual quantum devices. By leveraging emerging quantum technologies and optimizing the mapping of quantum circuits to physical qubits, authors can bridge the gap between theoretical proposals and real-world applications. This experimental validation will be crucial for assessing the performance and feasibility of quantum transfer learning in practical scenarios.

Additionally, collaborations with quantum hardware manufacturers and quantum computing research groups can aid in the development of specialized hardware architectures or quantum-inspired processors optimized for quantum transfer learning tasks. Exploring novel hardware designs and technologies, such as superconducting qubits or topological qubits, can contribute to the development of more efficient and powerful quantum computing platforms for facial image recognition.

IX. ACKNOWLEDGMENTS

The authors would like to express their gratitude to Intel for their support and resources throughout the course of this research. Specifically, we would like to thank Intel for providing access to their cutting-edge hardware and software

technologies, including the Intel One API framework, which played a crucial role in optimizing and training our models on Intel's processors.

The collaboration with Intel has been instrumental in accelerating our research and enabling us to leverage the power and performance of Intel's hardware for our experiments. The computational capabilities offered by Intel's processors have significantly contributed to the efficiency and effectiveness of our model training and optimization processes.

Furthermore, we would like to extend our appreciation to the entire team at Intel for their valuable insights, technical assistance, and continuous support. Their expertise and guidance have been invaluable in navigating the complexities of hardware optimization and ensuring the successful implementation of our research.

We are also grateful to the reviewers and editors for their constructive feedback and suggestions, which have greatly contributed to the refinement of this work. Their expertise and meticulous evaluation have helped shape the final version of our research manuscript.

Finally, we would like to acknowledge all the individuals who have directly or indirectly contributed to this research endeavor. Their contributions, whether through discussions, feedback, or assistance, have been invaluable in shaping our understanding and advancing our knowledge in the field of autism detection through image analysis.

COMPLIANCE WITH ETHICAL STANDARDS

This research was conducted in compliance with ethical standards. No animals or humans were harmed during the course of this study. This manuscript was not funded by any external sources.

Disclosure of potential conflicts of interest

The authors declare that they have no conflicts of interest. All authors have contributed significantly to the research and have reviewed and approved the final manuscript.

Research involving human participants and/or animals

This research did not involve human participants or animals.

Informed consent

Not applicable.

REFERENCES

- [1] K. L. Goh, S. Morris, S. Rosalie, C. Foster, T. Falkmer, and T. Tan, "Typically developed adults and adults with autism spectrum disorder classification using centre of pressure measurements," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 844–848.
- [2] J. E. Robison, "Talking about autism—thoughts for researchers," *Autism Research*, vol. 12, no. 7, pp. 1004–1006, 2019.
- [3] M. S. Satu, F. F. Sathi, M. S. Arifen, M. H. Ali, and M. A. Moni, "Early detection of autism by extracting features: a case study in bangladesh," in *2019 international conference on robotics, electrical and signal processing techniques (ICREST)*. IEEE, 2019, pp. 400–405.
- [4] Q. Guillon, N. Hadjikhani, S. Baduel, and B. Rogé, "Visual social attention in autism spectrum disorder: Insights from eye tracking studies," *Neuroscience & Biobehavioral Reviews*, vol. 42, pp. 279–297, 2014.
- [5] M. I. U. Haque and D. Valles, "A facial expression recognition approach using dcnn for autistic children to identify emotions," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2018, pp. 546–551.
- [6] O. Rudovic, Y. Utsumi, J. Lee, J. Hernandez, E. C. Ferrer, B. Schuller, and R. W. Picard, "CultureNet: A deep learning approach for engagement intensity estimation from face images of children with autism," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 339–346.
- [7] G. Yolcu, I. Oztel, S. Kazan, C. Oz, K. Palaniappan, T. E. Lever, and F. Bunyak, "Facial expression recognition for monitoring neurological disorders based on convolutional neural network," *Multimedia Tools and Applications*, vol. 78, pp. 31 581–31 603, 2019.
- [8] T. Akter, M. S. Satu, L. Barua, F. F. Sathi, and M. H. Ali, "Statistical analysis of the activation area of fusiform gyrus of human brain to explore autism," *Int. J. Comput. Sci. Inf. Secur.(IJCSIS)*, vol. 15, pp. 331–337, 2017.
- [9] M. S. Satu, M. S. Azad, M. F. Haque, S. K. Intiaz, T. Akter, L. Barua, M. Rashid, T. R. Soron, and K. A. Al Mamun, "Prottoy: A smart phone based mobile application to detect autism of children in bangladesh," in *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*. IEEE, 2019, pp. 1–6.
- [10] S. Schelinski, K. Borowiak, and K. von Kriegstein, "Temporal voice areas exist in autism spectrum disorder but are dysfunctional for voice identity recognition," *Social Cognitive and Affective Neuroscience*, vol. 11, no. 11, pp. 1812–1822, 2016.
- [11] J. Heaton, "Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618," *Genetic Programming and Evolvable Machines*, vol. 19, no. 1-2, pp. 305–307, 2018.
- [12] E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," *arXiv preprint arXiv:1802.06002*, 2018.
- [13] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New Journal of Physics*, vol. 18, no. 2, p. 023023, 2016.
- [14] A. Perdomo-Ortiz, M. Benedetti, J. Realpe-Gómez, and R. Biswas, "Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers," *Quantum Science and Technology*, vol. 3, no. 3, p. 030502, 2018.
- [15] J. Tung, E. A. Archie, J. Altmann, and S. C. Alberts, "Cumulative early life adversity predicts longevity in wild baboons," *Nature communications*, vol. 7, no. 1, p. 11181, 2016.
- [16] M. Schuld and N. Killoran, "Aprendizaje automático cuántico en espacios característicos de hilbert," *Cartas de revisión física*, vol. 122, p. 040504, 2019.
- [17] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Physical Review A*, vol. 101, no. 3, p. 032308, 2020.
- [18] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, "Continuous-variable quantum neural networks," *Physical Review Research*, vol. 1, no. 3, p. 033063, 2019.
- [19] S. S. J. P. Aspuru-Guzik, "A expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms adv," *Quant. Technol*, vol. 2, no. 12, p. 1900070, 2019.
- [20] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.
- [21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition. cvpr. 2016," *arXiv preprint arXiv:1512.03385*, 2016.
- [23] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.
- [24] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow*,

- UK, August 23–28, 2020, *Proceedings, Part I 16*. Springer, 2020, pp. 213–229.
- [25] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, “Stand-alone self-attention in vision models,” *Advances in neural information processing systems*, vol. 32, 2019.
 - [26] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, “Axial-deeplab: Stand-alone axial-attention for panoptic segmentation,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*. Springer, 2020, pp. 108–126.
 - [27] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. Van Der Maaten, “Exploring the limits of weakly supervised pretraining,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 181–196.
 - [28] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, “Big transfer (bit): General visual representation learning,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer, 2020, pp. 491–507.
 - [29] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, “Self-training with noisy student improves imagenet classification,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 687–10 698.
 - [30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.