Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

A Hybrid Neuro-Fuzzy Approach for Heterogeneous Patch Encoding in ViTs Using Contrastive Embeddings & Deep Knowledge Dispersion

S Muhammad Ahmed Hassan Shah[1,2, *, †], Muhammad Qasim Khan[2], Yazeed Yasin Ghadi[3], Sana Ullah Jan[4], Olfa Mzoughi[5], Monia Hamdi[6]

[1]Medical Imaging and Diagnostics Laboratory (MIDL), National Centre of Artificial Intelligence (NCAI), COMSATS University Islamabad, Pakistan. syedmahmedhassan321@gmail.com

[2]Department of Computer Science, COMSATS University Islamabad Attock Campus, Pakistan. syedmahmedhassan321@gmail.com, qasimkhan@cuiatk.edu.pk

[3]Department of Computer Science, Al Ain University, UAE. yazeed.ghadi@aau.ac.ae

[4]School of Computing, Edinburgh Napier University, UK. s.jan@napier.ac.uk

[5]Department of Computer Science, College of Sciences and Humanities, Prince Sattam bin Abdulaziz University, Saudi Arabia. o.mzoughi@psau.edu.sa

[6] Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia. MSHamdi@pnu.edu.sa

† Author with most Technical Contribution

[*]Corresponding author: S Muhammad Ahmed Hassan Shah (syedmahmedhassan321@gmail.com)

This study is supported via funding from Prince Sattam bin Abdulaziz University project (PSAU/2023/R/1444) and Princess Nourah bint Abdulrahman University Researchers Supporting project (PNURSP2023R125), Riyadh, Saudi Arabia.

ABSTRACT Vision Transformers (ViT) are commonly utilized in image recognition and related applications. It delivers impressive results when it is pre-trained using massive volumes of data and then employed in mid-sized or small-scale image recognition evaluations such as ImageNet and CIFAR-100. Basically, it converts images into patches, and then the patch encoding is used to produce latent embeddings (linear projection and positional embedding). In this work, the patch encoding module is modified to produce heterogeneous embedding by using new types of weighted encoding. A traditional transformer uses two embeddings including linear projection and positional embedding. The proposed model replaces this with weighted combination of linear projection embedding, positional embedding and three additional embeddings called Spatial Gated, Fourier Token Mixing and Multi-layer perceptron Mixture embedding. Secondly, a Divergent Knowledge Dispersion (DKD) mechanism is proposed to propagate the previous latent information far in the transformer network. It ensures the latent knowledge to be used in multi headed attention for efficient patch encoding. Four benchmark datasets (MNIST, Fashion-MNIST, CIFAR-10 and CIFAR-100) are used for comparative performance evaluation. The proposed model is named as SWEKP-based ViT, where the term SWEKP stands for Stochastic Weighted Composition of Contrastive Embeddings & Divergent Knowledge Dispersion (DKD) for Heterogeneous Patch Encoding. The experimental results show that adding extra embeddings in transformer and integrating DKD mechanism increases performance for benchmark datasets. The ViT has been trained separately with combination of these embeddings for encoding. Conclusively, the spatial gated embedding with default embeddings outperforms Fourier Token Mixing and MLP-Mixture embeddings.

INDEX TERMS vision transformer, patch encoding, spatial gated unit, Fourier token mixing, MLP-mixture embedding, computer vision

I. INTRODUCTION

Deep learning has led to significant advancements in computer vision, such as object detection and image classification, but these models often rely on fixed architectures like convolutional neural networks which may not be suitable for tasks that require processing non-grid structures like graphs and sequences [1]. One of the key architectures used in deep learning for computer vision tasks is the CNN that are designed to process grid-like data, such as images, by applying a series of convolutional and pooling operations to the input data [2]. This allows the model to learn hierarchical features of the image, such as edges, textures, and shapes. CNNs have been used to achieve state-of-the-art performance on many image-classification benchmarks.

One of the key advantages of CNNs is their ability to learn spatial hierarchies of features [1]. As the data is passed through the layers, the model can learn more complex and abstract features from the lower layers, such as edges and textures, while still preserving the spatial information of the image. This allows the model to make more accurate classifications, especially when the features of the objects in the image are spatially related.

The transformer architecture, originally proposed for natural language processing tasks, was introduced by Google in the paper “Attention is All You Need” in 2017 [3]. The transformer architecture uses self-attention mechanisms to learn relationships between different parts of the input data, such as words in a sentence [4].

In 2020, Google researchers proposed a new method called “Patch-based Predictive Coding”, where they adapted the transformer architecture for computer vision tasks by breaking an image into a sequence of patches and treating them as tokens in a sentence [5]. This method was dubbed as Vision Transformer (ViT) and was introduced in the paper “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale” [6]. The authors proved that a simple ViT model can achieve good results on several image classification benchmarks.

Since the introduction of ViT, it has been adapted and improved upon by several researchers in the field. In 2021, a new variant of ViT was proposed called Distilled Vision Transformer (DeiT) which shows that a smaller and more efficient version of ViT can achieve similar performance to the original one [7]. Researchers at other institutions also proposed their own variations of ViT such as Patch-based Vision Transformer (PVT) and Residual Vision Transformer (R-ViT) which also show the good performance in various computer vision tasks.

The transformer architecture, originally designed for natural language processing, has been adapted for computer vision tasks and the ViT is a variant that processes images as a sequence of patches, allowing the model to learn relationships between regions of the image using self-attention [8][9]. These architectures use self-attention mechanisms to learn relationships between different regions of the input data, such as images.

One of the main advantages of ViTs is their ability to handle images of different sizes and aspect ratios without the need for cropping or resizing [10]. Additionally, the self-attention mechanism allows the model to learn global dependencies between the patches, which can be beneficial for image classification tasks.

In addition to their performance, ViT are also highly modular and can be easily adapted to different tasks and architectures. This makes them a versatile tool for researchers and practitioners in the field of computer vision and has led to a growing number of papers and studies being published on the use of vision transformers for a wide range of tasks [4], [11]. Another reason for the popularity of vision transformers is their ability to handle large-scale image datasets.

Traditional CNNs are often limited by the fixed architecture and the need to down sample images to fit within memory constraints. ViT, on the other hand, can handle much larger image sizes and can be trained end-to-end on large-scale datasets. This allows for more accurate and robust models to be trained, which can then be used for a wide range of computer vision applications.

The primary objective of this study is to develop a patch encoding mechanism that is efficient and effective in extracting meaningful information from data. Previous transformer models have relied solely on a simple projection and positional embedding to summarize all significant patch information into an *N*-dimensional vector. However, more robust transformations are needed to extract complex patterns from the data while minimizing information loss. To address this issue, we proposed the use of a SWEKP, which is a weighted and contrastive patch encoding method integrated in transformers. Following is the description of the main components of transformers and how they have been updated to propose the given mechanism.

ViT divide images into small non-overlapping patches, which are then processed through a transformer network. Patch encoding in ViT involves converting each patch into a vector representation using a linear projection layer, which maps each patch to a lower-dimensional embedding space. The output of this layer is concatenated with a positional encoding vector to provide the transformer network with spatial location information for each patch.

The resulting patch embeddings are fed into a multi-head self-attention mechanism to capture the relationships between different patches. Finally, the transformer network produces a fixed-length vector representation of the input image, which can be utilized for various downstream tasks, including image classification, object detection, and segmentation. Patch encoding and the encoder block of the transformer are crucial steps in the ViT architecture. In the patch encoding phase, the traditional approach of simple projection and positional embedding of patches is replaced with a weighted composition of five types of embeddings: projection, positional, mixture, Gated Multi-Layer Perceptron (GMLP), and Fourier. This change helps the patch encoding mechanism to extract more meaningful information from patches and fuse them together.

After patch encoding, the encoding is fed to the encoder block of the transformer for meaningful representation learning for downstream tasks. The use of multiple embeddings in the previous stage enables deep knowledge dispersion, which means that these embeddings are utilized further in the architecture.

Furthermore, the key, query, and value in the self-attention mechanism of the encoder block are replaced with mixture embedding, GMLP embedding, and Fourier embedding. By doing so, the previous knowledge gained in self-attention is utilized for more robust representations. Finally, a simple classifier takes the learned representations as input and predicts the target. The combination of these steps enhances the representation learning of ViT and makes it suitable for various downstream tasks such as image classification, object detection, and segmentation.

The proposed SWEKP-ViT model employs weights that control the flow of information extracted by the multiple patch-encoding approaches including mixture Multi-Layer Perceptron (MLP), Fourier MLP, and GMLP. By defining weights, the model can regulate the information flow from each encoding approach to the overall model output. This approach improves the performance of the model by allowing it to extract the most relevant information from the input data. The term hybrid describes the combination of different encoding techniques which are combined in a single mechanism, called SWEKP. By combining these techniques, the model can extract complex patterns and relationships from the input data.

The proposed architecture has been evaluated using several benchmark datasets, such as MNIST, Fashion-MNIST, CIFAR10, and CIFAR100. The objective is to specify that it be applied to a range of visual recognition tasks, including medical imaging, by evaluating the performance of the proposed model on the benchmark datasets. However, it can be applied in multiple applications beyond these datasets, such as medical imaging data to assist in the computer-aided diagnosis of different diseases, as well as in security or surveillance and remote sensing applications.

The main contributions of this work are summarized as follows.

A. CONTRIBUTION

1) A supervised SWEKP-based ViT is proposed which consist of heterogeneous patch encoding mechanism. In traditional ViT, linear projection and positional embedding are applied. However, the proposed SWEKP-based ViT model uses multiple types of latent embeddings including Mixture, Gated MLP and Fourier.

2) Divergent Knowledge Dispersion (DKD) is proposed for encoding process in ViT. It propagates all previous divergent knowledge to multi-headed attention mechanism at every iteration.

3) Lastly, a Mix-Up data augmentation technique is used with SWEKP transformer. It combines different features and their corresponding labels, to prevent a network from becoming too confident in the relationship between the features and labels.

II. LITERATURE REVIEW

The ViT has been a popular topic of research in recent years, particularly for its application in image classification tasks. In this section, we study different state of the art transformer methods developed in literature for benchmark dataset including MNIST, FASHION-MNIST, CUFAR10 and CIFAR100.

Traditional transformer, initially used for Natural Language Processing (NLP), is a deep neural network relying mainly on self-attention. Its robust representation capabilities have sparked interest in applying it to computer vision. Transformer-based models have shown comparable or superior results compared to convolutional and recurrent neural networks in various visual benchmarks [12]. Due to its high performance and reduced requirement for vision-related inductive bias, the computer vision community is increasingly exploring transformer. In this paper, authors classify vision transformer models based on task and evaluates their strengths and weaknesses.

The ViT architecture proposed has been applied to several image classification tasks, including the popular fashion-MNIST and MNIST datasets [6]. Before, that, in 2010, Zhai X et. al. introduced Multiscale Vision Transformers (MViT) for video and image recognition, combining the concept of multiscale feature hierarchies with transformer models [4][13]. It has multiple channel-resolution scale stages. Starting with a small channel dimension at the input resolution, the stages incrementally increase channel capacity while decreasing spatial resolution, resulting in a multiscale feature pyramid. Early layers handle simple low-level visual information at high spatial resolution, while deeper layers process complex high-dimensional features at coarser spatial resolution. It outperformed then state-of-the-art vision transformers that require large-scale pre-training and have 5-10 times higher computation and parameter cost when tested on various video recognition tasks.

Transformers with strong global relationship modeling capabilities have been applied to basic computer vision tasks recently, such as the ViT, which directly uses a pure transformer architecture for image classification by dividing images into fixed-length tokens and learning the relationships between them. However, this straightforward tokenization can harm object structures, assign grids to irrelevant regions like the background, and cause interference.

To address these issues, Yue X et al proposed an iterative and progressive sampling strategy that identifies distinctive regions [11]. The proposed Parallel Sequence Vision Transformer (PS-ViT) network includes a transformer encoder layer that receives embeddings from each iteration and predicts a set of sampling offsets to update the sampling locations for the next iteration. This approach of progressive sampling is differentiable and when combined with the ViT, it forms a powerful and efficient network called PS-ViT that can learn where to focus. PS-ViT has shown great effectiveness and efficiency.

R. Ranftl et al presented Dense Prediction Transformers, a model that uses ViT instead of convolutional networks as a backbone for dense prediction tasks [14]. The multi-resolution image representations are formed by combining tokens from various phases of the ViT, which are later merged into full-resolution predictions through a convolutional decoder. The transformer backbone is responsible for continuously processing high-resolution representations at each stage, having a global receptive field. As compared to fully convolutional networks, this results in finer-grained and more coherent predictions.

ViT have been highly successful in various vision tasks, but require significant computational resources, making them challenging to use on resource-limited devices. To overcome this, W. Li et. al., adopted the concept of depth-wise separable convolution to design the Separable Vision Transformer (SepViT) [15]. It uses depth-wise separable self-attention to interact with information within and between windows. It employs novel window token embedding and grouped self-attention to model the attention relationship between windows efficiently and capture long-range visual dependencies. Results from experiments on various benchmark tasks show that SepViT offers a balance of accuracy and latency and achieves state-of-the-art results.

P. Zhang et. al., introduces the Multi-Scale Vision Longformer, a new ViT architecture that improves encoding of high-resolution images [16]. The architecture uses two techniques: a multi-scale model structure for image encodings at multiple scales with efficient computation, and the Vision Longformer attention mechanism, a variant of Longformer originally developed for NLP, with linear complexity relative to input tokens. The proposed technique used for range of tasks like images classification, detection and segmentation.

There are multiple benchmark datasets available for image recognition such as CIFAR10 and Fashion MNIST dataset. O. Khanday et. al., investigate the impact of filter size on the accuracy of (CNNs) [17]. The model architecture is kept unchanged and only vary the filter size among different sizes (3x3, 5x5, and 7x7). The CIFAR10 and Fashion MNIST datasets are used in this study. Our results show that the accuracy decreases as the filter size increases, with 3x3 filters achieving an accuracy of 73.04% on CIFAR10 and 93.68% on Fashion MNIST.

In 2020. Kurt Ma et. al., proposed a Hilbert-Schmidt Independence Criterion (HSIC) bottleneck as a training method for deep neural networks [18]. Unlike the traditional cross-entropy loss and backpropagation, the HSIC bottleneck has numerous benefits. For example, it can solve the issues of exploding and vanishing gradients, enabling the training of deep networks without skip connections. Results show that the HSIC bottleneck performs similar to backpropagation with cross-entropy on MNIST, Fashion-MNIST, and CIFAR10 classification, even without the need to make the output look like the classification labels. The accuracy on the test set is reported as 98.8%, 88.3%, and 59.4% for the format-trained networks and 98.4%, 87.6%, and 56.5% for the backpropagation-trained networks for MNIST, Fashion-MNIST, and CIFAR10 datasets, respectively [19].

The Capsule Network (CapsNet) is a unique deep neural network structure that maps target instances to vectors and matrices instead of scalars, facilitated by the dynamic routing algorithm. This results in a more robust capacity with fewer parameters compared to traditional CNNs [20]. However, CapsNet has the drawback of considering everything in the image, which leads to poor performance when backgrounds are too diverse. In 2020, Chang S. et. al. proposed a Multi-Lane Capsule Network with Strict-Squash (MLSCN) addresses this issue [21]. A new Capsule network structure is introduced replacing the Squash function, and optimizing dropout. Experiments on MNIST, affNIST, and CIFAR10 datasets were conducted to validate MLSCN's modifications, and ablation experiments were conducted to analyze the contribution of each component. The results showed that MLSCN outperforms the original CapsNet in multiple benchmarks. They achieve the accuracy of 98.42, 76.79 on MNIST and CIFAR10 dataset against MLSCN achieving 73.472% accuracy.

III. MATERIALS AND METHODS

The ability of deep neural networks to learn complex representations from large amounts of data has led to significant improvements in a range of tasks, including image classification, object detection, semantic segmentation, and image generation [22]. With the increasing availability of large, annotated datasets and advancements in hardware, deep learning will continue to play a crucial role in advancing computer vision tasks.

A. PROPOSED METHODOLOGY

In this work, Stochastic Weighted Composition of Contrastive Embedding's & Divergent Knowledge

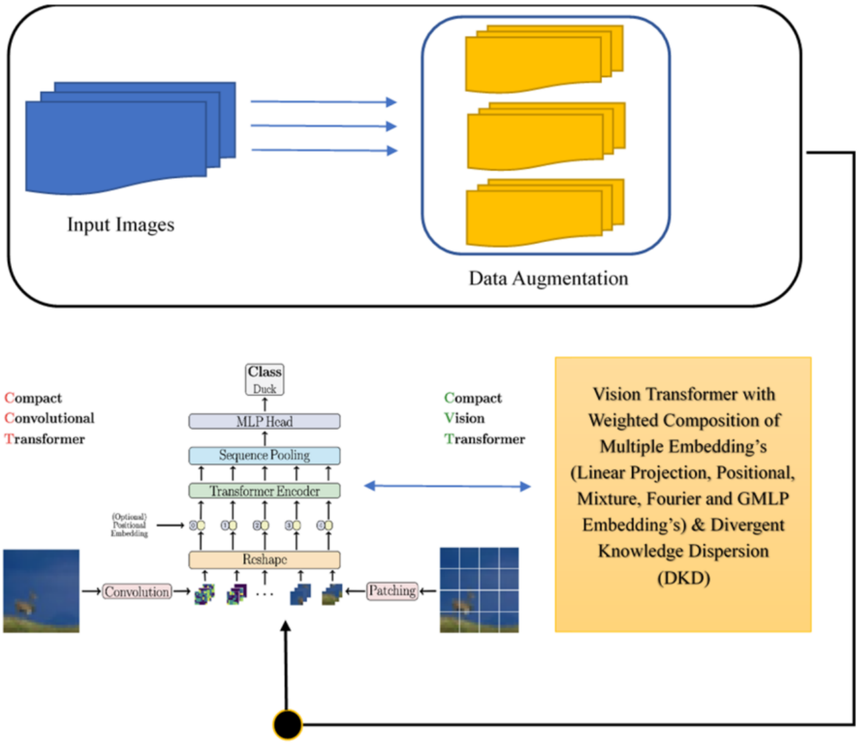


FIGURE 1. **Proposed Methodology**

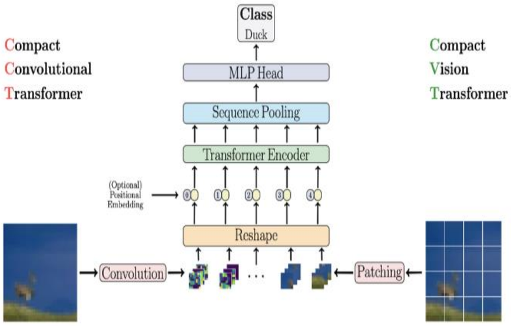


FIGURE 2. **Visual Diagram of Vision Transformer** [31]

Dispersion (SWEKP)-based ViT is proposed. This section presents detailed discussion about the different phases of proposed technique.

In proposed methodology, depicted visually in Figure 1, a Mix-Up augmentation is used for data augmentation process. After data augmentation, data is fed to patch encoder to convert images into patches. Next step is to transform patches into latent representations or embeddings and present it to transformer layers for more efficient patch encoding. Transformer use multiple attention blocks and MLPs for encoding purpose. After multiple blocks, a concise representation for each image is obtained. Finally, a weightage to representation is given and fed to classifier for class prediction.

B. DATA AUGMENTATION

Mix-Up data augmentation is type of image data augmentation that involves mixing up the data. The implementation of mix-up is straightforward and its purpose is to prevent overfitting in neural networks by combining different features and labels [23]. This technique is particularly useful when there is uncertainty in choosing augmentation techniques, such as in medical imaging datasets [24]–[26].

$$x_i = \lambda \, . \, x_i + (1 - \lambda) \, . \, x_j \qquad (1)$$

$$y_i = \lambda \, . \, y + (1 - \lambda) \, . \, y_j \qquad (2)$$

where lambda λ values are picked between 0-1, and sample from beta distribution. And x and y are data features and labels, respectively. Equations 1 and 2 represent the Mix-Up augmentation technique, where x_i and y_i correspond to the image and its label of the i^{th} sample, respectively. The parameter λ is a randomly drawn value from a beta distribution with alpha determining the degree of mixing between two images. The remaining portion of the image is formed by the second image, x_j, and its corresponding label y_j. The mixing ratio is controlled by the value of λ, which ranges from 0 to 1. A value of 0.5 corresponds to an equal mix of both images, whereas a value closer to 0 or 1 produces an image that closely resembles one of the original images. Mix-Up augmentation is an effective technique for reducing overfitting and improving the generalization performance of

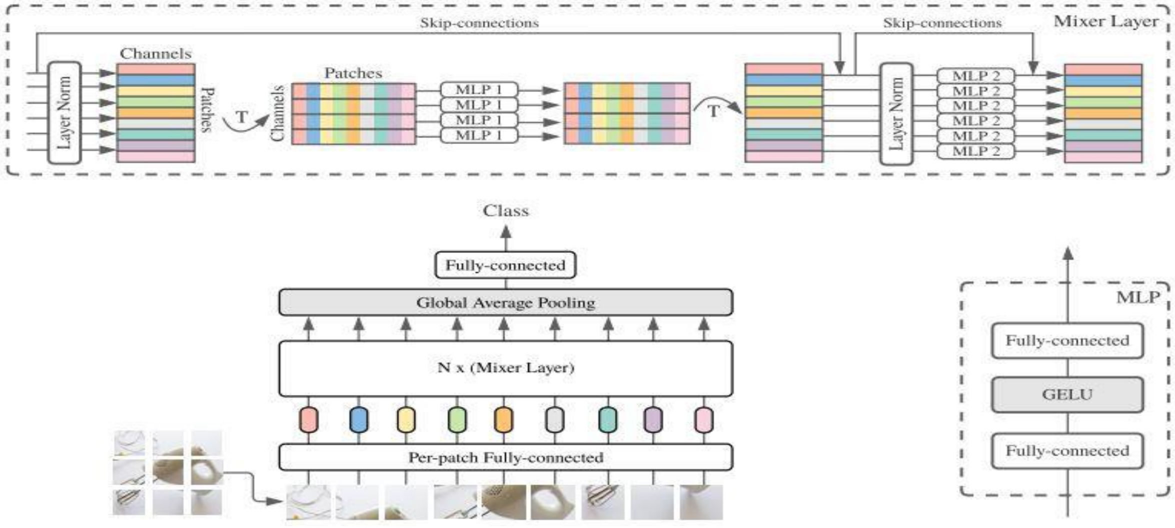


FIGURE 3. **Working diagram of mixer MLP** [31]

deep learning models. This is particularly useful when the appropriate set of augmentation transforms for a given dataset is uncertain, as in the case of medical imaging datasets. The Mix-Up technique can be applied to a wide range of data modalities, including computer vision, natural language processing, and speech.

The versatility of Mix-Up makes it applicable to a range of data modalities, including computer vision, NLP, speech, and others. The equation below shows the beta function in integral form [27].

$$\int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx = \frac{\Gamma(\alpha).\Gamma(\beta)}{\Gamma(\alpha+\beta)}, \qquad (3)$$

where α, β are the parameters in beta function.

C. PREPROCESSING

Scaling normalizes data by transforming each pixel value to fall between 0 and 1. This technique modifies the visual appearance of an image and adjusts the amount of information it contains. The scaled image is obtained by transforming the original image (I) with the minimum and maximum pixel values (I_{min} and I_{max}).

$$I_{Norm} = \frac{(I-I_{min})}{(I_{max}-I_{min})} \qquad (4)$$

Pixel Scaling is a normalization technique commonly used in image processing. It involves scaling the pixel values to a consistent range by dividing each value by the maximum value (255). This ensures standardized data and enhances model training. The overall visual appearance of the image remains unaffected as the relative differences in pixel intensities remain the same. The brightness, contrast and overall structure of the image are preserved. The only change is that pixel values are represented as floating-point numbers between 0 and 1 instead of integers between 0 and 255.

D. DIFFERENT TECHNIQUES IN PROPOSED SWEKP-ViT

ViT is a type of deep neural network architecture designed for computer vision tasks. In next sections we briefly discuss the working of proposed SWEKP-based ViT.

1) SIMPLE VISION TRANSFORMER

The main idea behind ViT is to treat an image as a sequence of patches instead of a 2D matrix of pixel values. Each patch is then represented as a fixed-length vector of values that is fed into the network. The network is composed of multiple layers of multi-head self-attention mechanisms, which are used to capture long-range dependencies in the image. The self-attention mechanism allows each patch to attend to other patches in the sequence, thereby capturing relationships between different parts of the image. Figure 2 presents a visual representation of a simple ViT.

After multiple layers of self-attention, the network applies a linear transformation to the sequence of patches to generate a higher-level representation of the image. This representation is then used to make predictions or perform other computer vision tasks, such as object recognition or segmentation. The use of self-attention mechanisms in ViT enables the network to learn about the relationships between the patches in an image, making it well-suited for tasks that require understanding of the entire image. The architecture of ViT has been shown to outperform traditional CNNs on a variety of computer vision benchmarks. Additionally, because the network is designed to process sequences of patches, it is more flexible than CNNs, which are limited by the structure of the convolutional filters.

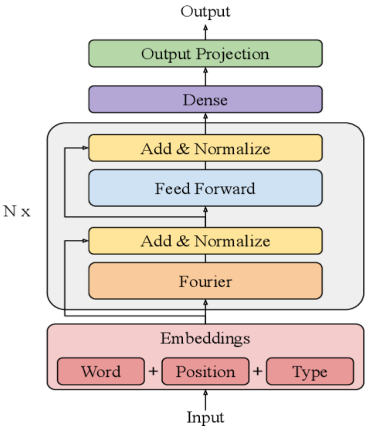


FIGURE 4. **Working Diagram of Fourier-Net** [32]

2) MIXTURE MLP

The MLP-Mixer architecture is built entirely with MLPs. It consists of two types of layers: one that applies MLPs independently to image patches (i.e., combining per-location features), and another that applies MLPs across patches (i.e., mixing spatial information) [27].

The mixer architecture is designed to distinctly differentiate the channel-mixing operations, 1) at each location from the token-mixing operations, and 2) across locations. Figure 3 illustrates the working of mixture MLP visually.

The mixer architecture inputs a sequence of S non-overlapping image patches, with each patch transformed to a desired hidden dimension C. This creates a two-dimensional real-valued input table X with dimensions $X \epsilon \mathbb{R}^{SXC}$. The number of patches is determined by the resolution of the original image and the desired patch size. The mixer consists of multiple layers of identical size, each with two MLP blocks. The first MLP block, also known as the token-mixing MLP, operates on columns of X and maps $\mathbb{R}^{S} \rightarrow \mathbb{R}^{S}$, while the second block, called the channel-mixing MLP, operates on rows of X and maps $\mathbb{R}^{C} \rightarrow \mathbb{R}^{C}$. Both MLP blocks have two fully connected layers and apply nonlinearity independently to each row of their input data tensor.

$$U_{*,i} = X_{*,i} + W_2\,\delta(W_1\,LayerNorm(X)_{*,i}), \quad i = 1,2,3 \ldots\ldots C \tag{5}$$

$$Y_{j,*} = U_{j,*} + W_4\,\delta(W_3\,LayerNorm(X)_{j,*}), \quad i = 1,2,3 \ldots\ldots S \tag{6}$$

where δ is element-wise non-linearity Gaussian Error Linear Unit (GeLU).

3) FOURIER MLP

The Fourier-Net (FNet) architecture is a transformer without attention, where each layer is composed of a Fourier mixing sublayer and a feed-forward sublayer [28]. This architecture replaces the self-attention sublayer of a typical Transformer encoder layer with a Fourier sublayer, which performs a 2D discrete Fourier transform on its input a 1-dimensional Discrete Fourier Transform (DFT) along the sequence dimension ($\mathcal{F}_{seq}$) and another 1D DFT along the hidden dimension ($\mathcal{F}_{h}$) [29]. The structure of the FNet can be seen in Figure 4.

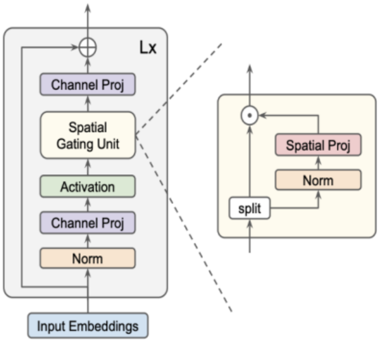


FIGURE 5. **Working Diagram of Spatial Gated MLP (GMLP)** [33]

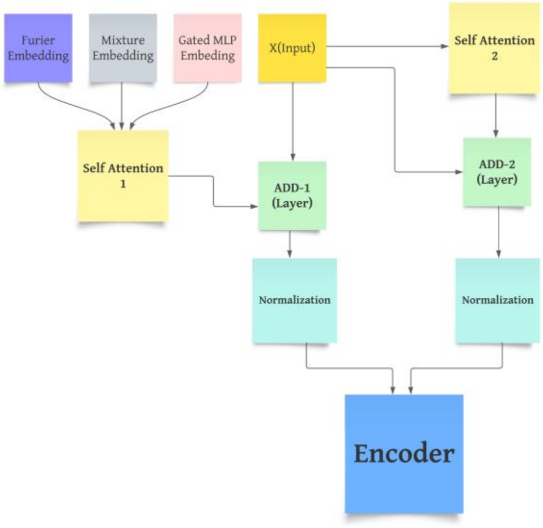


FIGURE 6. **Graph illustrating the process flow of Divergent Knowledge Dispersion (DKD)**

FNet use Fourier transform instead of self-attention, as shown in Figure 4, and a function is broken down into its component frequencies. The formula for the DFT is given a sequence x_n with n ϵ [0, N-1] is given as follows.

$$X_k = \sum_{n=0}^{N-1} x_n e^{(\frac{-2\pi i}{N})\,nk} \;, where\ 0 < k < N-1 \tag{7}$$

Let $\mathcal{F}_{\vartheta}$ represents the Fourier embedding, which is used in patch encoder, we will use this notation later in the paper.

4) SPATIAL GATED MLP

The Gated MLP (GMLP) model is made up of L blocks with the same structure and size. X represents the representation of tokens with a length of n and a dimension of d, and it exists in $X \epsilon \mathbb{R}^{nXd}$.

$$Z = \delta(XU) \tag{8}$$
$$Z^{\sim} = \delta(Z) \tag{9}$$
$$Y = ZV \tag{10}$$

The activation function $\boldsymbol{\delta}$, such as GeLU, is used in the formula. Linear projections along the channel dimension, U and V, are defined as in the feedforward networks (FFNs) of transformers, for example, their shapes are 768 x 3072 and 3072 x 768 for Bidirectional Encoder Representations from Transformer (BERT) base [30]. Shortcuts, normalizations, and biases are not shown for simplicity. Figure 5 depicts the schematic diagram of spatial GMLP.

The core component of the GMLP model is a function s $(\cdot)$ that captures spatial interactions among tokens in a sequence. When s is an identity mapping, the model reduces to a standard FFN, processing each token independently without cross-token communication. The goal is to design an effective s for capturing complex spatial interactions. The model's structure is influenced by inverted bottlenecks, where s $(\cdot)$ is defined as a spatial depth wise convolution. Unlike transformer models, GMLP does not require position embeddings as the spatial information is captured in s $(\cdot)$. Let Ω_δ represents the spatial gated embedding, we will use this notation later in the paper.

5) DIVERGENT KNOWLEDGE DISPERSION (DKD)

DKD is a knowledge propagation technique we used in ViT. When we use Mixer-MLP, Fourier and GMLP Embeddings for converting image patches into latent representations, these representations encode all information about image patches. In DKD mechanism we propagate this embedding far into the transformer network. Let Attention $(X_{patch}, X_{patch}, X_{patch})$ be the multi-headed attention with encoded patches as input where X_{patch} is the encoded patch before going into the multi-headed attention block. We know $\mathcal{M}_\sigma, \mathcal{F}_\vartheta, \Omega_\delta$ are three embeddings, and we use this representation for more efficient encoding. Instead of using single multi-headed attention which only take weighted mixture of all embeddings as input, we use two multi-headed attentions. One is simple default attention $(X_{patch}, X_{patch}, X_{patch})$ and second attention take these three embeddings as input.

$$\text{Attention}\ (\mathcal{M}_\sigma, \mathcal{F}_\vartheta, \Omega_\delta) = \text{SoftMax}\ \left(\frac{(\mathcal{M}_\sigma.\mathcal{W}^{\mathcal{M}}).(\Omega_\delta.\mathcal{W}^{\Omega})}{\sqrt{d_k}}\right).\ \mathcal{F}_\vartheta.\mathcal{W}^{\mathcal{F}} \tag{11}$$

In above equation $\mathcal{W}^{\mathcal{M}}, \mathcal{W}^{\Omega},$ and $\mathcal{W}^{\mathcal{F}}$represent the weight metrics. DKD use two attentions to propagate the previous latent knowledge in the next sections of vision transformer as shown in Figure 6. In DKD, initially, we incorporate four inputs: Fourier embedding, mixture embedding, GMLP embedding, and X (weighted patch encoding), which combines all the embeddings. The self-attention mechanism is implemented through Self-Attention-1 and Self-Attention-2 blocks, both of which utilize multi-headed attention. Following the self-attention phase, we employ addition layers that act as residual connections, combining the output of the attention mechanism with the original weighted input, x. After the addition operation, we apply normalization layers. In the last, these representations or embeddings are then passed to the encoder section, as denoted by "ENCODER" in Figure 6.

IV. EMBEDDINGS IN PROPOSED SWEKP-ViT ARCHITECTURE

In this paper, we proposed a novel ViT which use heterogeneous patch encoding mechanism. There are two main contributions of this paper. Firstly, a mixture of embeddings is used. Let say LP_∂ represents the linear projection and P_{Emb} is positional embedding, then we can write the image patches as latent representations as follows.

$$\mathcal{Z}_{\text{Default}} = (\text{LP}_\partial + \text{P}_{Emb}) \tag{12}$$

Different types of embeddings (Mixer, Fourier and GMLP) are applied to make patch latent representation more robust. Because, the three new embedding also use linear projection and positional embeddings, so they are represented as $(\text{LP}_\partial,\ \text{P}_{Emb})$. The proposed mixture function LP_∂ is written as GLP_∂ (Global Linear Projection) and P_{Emb} is written as GP_{Emb} (Global Positional Embedding) while $(\text{LP}_\partial,\ \text{P}_{Emb})$ are local embeddings inside in three advance embeddings. The proposed mixture function is written as,

$$\mathcal{Z}_{\Sigma,\phi} = \lambda\,.\,(\text{GLP}_\partial + \text{GP}_{Emb}) + [\omega_1.\mathcal{M}_\sigma + \omega_2.\mathcal{F}_\vartheta + \omega_3.\Omega_\delta] \tag{13}$$

Now $\text{GLP}_\partial + \text{GP}_{Emb}$ is default term, we add new terms $\mathcal{M}_\sigma\ (Mixer),\ \mathcal{F}_\vartheta(Fourier),\ \Omega_\delta(Spatial\ Gated)$ in embedding mixture function. Where λ, ω_i is the weights, therefore, called weighted composition. $\mathcal{Z}_{\Sigma,\phi}$ represents latent final representation and Σ shows the linear sum, and ϕ represents the parameters. The Equation 13 can be generalized for the 3 embeddings by modifying the $\mathcal{Z}_{\Sigma,\phi}$ as $\mathcal{Z}_{\odot,\phi}$ and given as

$$\mathcal{Z}_{\odot,\phi} = \lambda\,.\,(\text{GLP}_\partial + \text{GP}_{Emb}) \oplus [\psi(\mathcal{M}_\sigma, \mathcal{F}_\vartheta, \Omega_\delta) \odot \mathcal{W}(\omega_1, \omega_2, \omega_3)]. \tag{14}$$

Now, there is no linear summation between embeddings and weights. There is some other non-linear relation between them represented as $\odot$. Where $\psi(\mathcal{M}_\sigma, \mathcal{F}_\vartheta, \Omega_\delta)$ is some non-linear function applied on these embeddings. And $(\omega_1, \omega_2, \omega_3)$ shows the weight function. For example, weights are drawn from some probability distribution. But

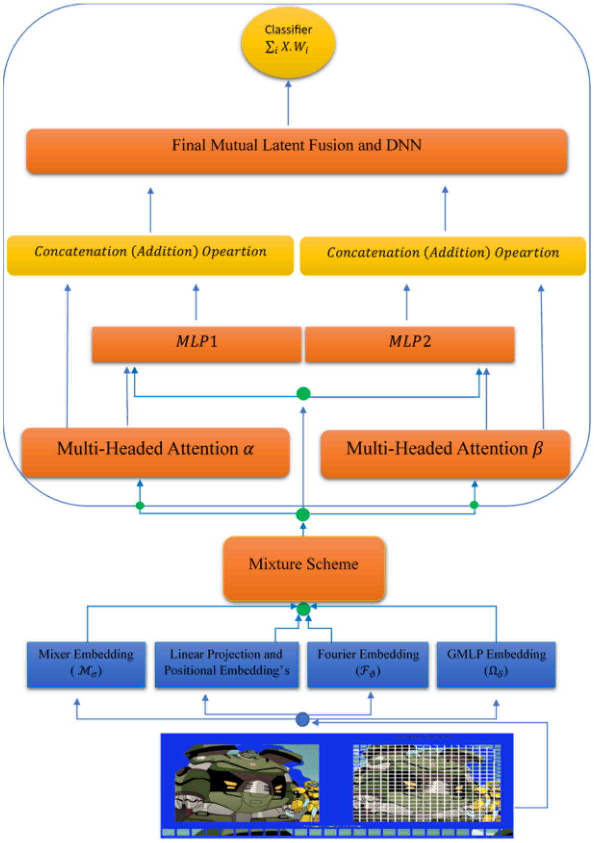

Classifier
$\sum_i X.W_i$
Final Mutual Latent Fusion and DNN
Concatenation (Addition) Opeartion
Concatenation (Addition) Opeartion
MLP1
MLP2
Multi-Headed Attention α
Multi-Headed Attention β
Mixture Scheme
Mixer Embedding ($\mathcal{M}_\sigma$)
Linear Projection and Positional Embedding's
Fourier Embedding ($\mathcal{F}_\vartheta$)
GMLP Embedding (Ω_δ)

FIGURE 7. Proposed SWEKP based ViT

this idea not implemented in this paper, we only implement $\mathcal{Z}_{\Sigma,\phi}$ simple linear summation of these representations and weights.

Now another important thing is that we can even more generalized the above equation. In above equation we use total five embedding's but if we have N embeddings. So, we can utilize 5 embeddings as

$$\mathcal{Z}_{\odot,\tau} = \mathrm{H}(\mathcal{E}_\tau) \odot \mathcal{W}(\omega_i) \tag{15}$$

Here H represents function of N embeddings, where $\mathcal{E}_\tau$ shows N embedding's $\mathcal{E}_1$, $\mathcal{E}_2 \ldots$ $\mathcal{E}_N$. And $\mathcal{W}(\omega_i)$ is weight function which generate weights for all N embeddings. We can also write our proposed scheme as $\mathrm{H}(\mathcal{E}_\tau) \odot \mathcal{W}(\omega_i)$

$$\mathrm{H}(\mathrm{GLP}_\partial, \mathrm{GP}_{Emb}, \mathcal{M}_\sigma, \mathcal{F}_\vartheta, \Omega_\delta) \odot \mathcal{W}(\omega_1, \omega_2, \omega_3, \omega_4, \omega_5) \tag{16}$$

Now in the above equation, if ⊙ is Σ then it is our case where we take linear combination of weights and embedding representations. But we can fuse weights and latent embeddings as we want, so ⊙ is any generalized operation. We use word stochastic in our title because we also take weight function as a probability distribution and drawn weights from it. After writing patches as latent representations, next we use multi-headed attention.

TABLE 1. Parameters for the Proposed SWEKP based ViT

Parameter	Value/Detail
Transformer Layers	8
Datasets	Fashion-MNIST, MNIST, CIFAR10, & CIFAR100
Attention Heads	4
Optimization Parameters	
Epochs	60
No. of Classes	10 & 100
Batch Size	32
Learning Rate	0.001
Optimization	Adam
Trainable Parameters	1,407,527

Multi-headed attention is a mechanism used in deep learning models to attend to multiple representations of input data simultaneously. In this technique, multiple attention mechanisms are applied to the input data in parallel and their outputs are concatenated and then fed into a fully connected layer to produce the final output. This allows the model to attend to different aspects of the input data and produce a more sophisticated representation compared to traditional single-headed attention models. This concept is widely used in state-of-the-art models for NLP tasks such as machine translation, text classification and question answering. In this work, SoftMax is used as non-linear function. First, we give encoded representations to attention and get.

$$\mathcal{A}_{\alpha}= \text{SoftMax}\ (\frac{(\mathcal{Z}_{\odot,\tau}.\mathcal{W}^{1}).(\mathcal{Z}_{\odot,\tau}.\mathcal{W}^{2})}{\sqrt{d_k}}).\ \mathcal{Z}_{\odot,\tau}.\,\mathcal{W}^{3} \qquad (17)$$

where $\mathcal{Z}_{\odot,\tau}$ is the encoded embedding and $\mathcal{W}^{\mathrm{i}}$, i from 1to 3 are wrights used in this attention block. We also propagate previous knowledge forward so another attention block is also used which take Mixer, Fourier and GMLP embedding as input.

$$\mathcal{A}_{\beta}=\text{SoftMax}\ (\frac{(\mathcal{M}_{\sigma}.\mathcal{W}^{\mathcal{M}}).(\Omega_{\delta}.\mathcal{W}^{\Omega})}{\sqrt{d_k}}).\ \mathcal{F}_{\vartheta}.\,\mathcal{W}^{\mathcal{F}} \qquad (18)$$

After attention two addition layer is used, first layer adds encoded patches $\mathcal{Z}_{\odot,\tau}$ with $\mathcal{A}_{\alpha}$ and second addition layer is used to add $\mathcal{Z}_{\odot,\tau}$ with $\mathcal{A}_{\beta}$. And after some normalizing technique, MLP is applied on each of addition layer outputs (MLP_{α} , MLP_{β}). Next step is to add the addition of $\boldsymbol{\mathcal{Z}_{\odot,\tau}}$ and $\boldsymbol{\mathcal{A}_{\alpha}}$ with $\boldsymbol{MLP_{\alpha}}$, and add the addition of $\boldsymbol{\mathcal{Z}_{\odot,\tau}}$ and $\boldsymbol{\mathcal{A}_{\beta}}$ with $\boldsymbol{MLP_{\beta}}$. And in the last we again add the two results come from these summations and give it to classifier for prediction. The working diagram shows each step in visual form. The Figure 7 shows the full working of proposed SWEKP-based ViT.

In the last paragraph, we provide a summary of the proposed transformer using mathematical equations. This paper presents a novel ViT (Vision Transformer) model that incorporates a heterogeneous patch encoding mechanism. The first phase of the proposed model involves data augmentation and preprocessing of the input, which includes resizing and scaling as described in equations 1, 2, and 4. Firstly, a combination of linear projection (LP$_{\partial}$) and positional embedding (P$_{Emb}$) is utilized through equation 12 to create more robust patch latent representations. To further enhance the representation power, different types of embeddings, including Mixer, Fourier, and GMLP embeddings, are incorporated addition with linear projection and positional embeddings. Equation 13 defines the proposed mixture function, which incorporates GLP$_{\partial}$ (Global Linear Projection) and GP$_{Emb}$ (Global Positional Embedding) as default terms. Additionally, new terms such as $\mathcal{M}_{\sigma}$ (Mixer), $\mathcal{F}_{\vartheta}$ (Fourier), and Ω_{δ} (Spatial Gated) are introduced into the embedding mixture function. To generalize the equation when considering N embeddings, equation 15 can be expressed as $\mathrm{H}(\mathcal{E}_{\tau}) \odot \mathcal{W}(\omega_{i})$, where H represents a function of N embeddings ($\mathcal{E}_{1}$, $\mathcal{E}_{2}\ldots$ $\mathcal{E}_{N}$), and $\mathcal{W}(\omega_{i})$ generates weights for all N embeddings (equation 16). The paper introduces new parallel mechanism DKD, which applies multiple attention mechanisms in parallel. The outputs are then concatenated and fed into a fully connected layer to obtain the final output. Attention block, depicted in equation 18, is employed to incorporate previous knowledge and takes input from Mixer, Fourier, and GMLP embeddings. The next steps involve using two addition layers to combine the encoded patches ($\mathcal{Z}_{\odot,\tau}$) with $\mathcal{A}_{\alpha}$ and $\mathcal{A}_{\beta}$, respectively. After normalization, MLP is applied to each output. The final stage includes adding $\mathcal{Z}_{\odot,\tau}$ and A_α with MLP_{α} and adding $\mathcal{Z}_{\odot,\tau}$ and $\mathcal{A}_{\alpha}$ with MLP_{β}. The resulting summations are then passed through the classifier for prediction. A visual representation of the proposed SWEKP-based ViT can be found in Figure 7, which provides a comprehensive overview of the entire working process.

V. RESULTS AND DISCUSSION

The primary objective of this research is to create an efficient vision transformer for visual recognition tasks. In this section, we conduct a comprehensive comparison between SWEKP based ViT and multiple ViTs with default and advanced embeddings, excluding the DKD mechanism. These architectures were trained on the same data, with equal epochs and learning rates. A brief discussion of each of these architectures can be found in the following sections. In order to conduct a fair comparison between the different models, the experiments were carried out with a fixed set of hyperparameters. Hyperparameters are predetermined values that control the behavior of the model during both training and testing phases. This approach ensures that all the experiments were conducted under the same conditions, thereby enabling a more precise analysis and comparison of the results. Additionally, employing the same hyperparameters prevents any variation in the model's behavior due to the use of different parameter settings, which might lead to differences in the obtained results.

A. Hyperparameters Configuration of Proposed Transformer

The details of our proposed ViT are presented in the table provided below. The experiment was carried out on four benchmark datasets, namely, Fashion-MNIST, MNIST, CIFAR10, and CIFAR100. The proposed ViT consists of a total of 8 transformer layers, and the number of parameters in the model is 1,407,527, as shown in Table 1. During the experiment, the proposed algorithm was trained for up to 60 epochs with a batch size of 32 and a learning rate of 0.001. One epoch refers to a single pass through the entire training dataset. The Batch Size refers to the number of training examples used in one iteration of the training process. In this case, the ViT model was trained using batches of 32 images at a time. The learning rate refers to the step size used to update the model's parameters during the training process. A lower learning rate typically results in slower but more stable convergence of the model during training. The four benchmark datasets utilized include Fashion-MNIST, which has 10 classes of shirts and pants; MNIST, which has 10 classes of numbers ranging from 0 to 9; CIFAR10, which has 10 classes of general images; and CIFAR100, which has 100 classes of general images. All the experiments were conducted using the same hyperparameter settings. In the subsequent sub-sections, we will perform a comparative analysis between SWEKP based ViT and other versions of

TABLE 2. Comparative Analysis on Fashion-MNIST data

Type of Embeddings	Accuracy	Recall	Precision	F1-Score	Loss	AUC
Linear Projection & Positional	90.56	89.24	92.02	90.59	0.2546	99.57
Mixture, GMLP & Fourier	90.01	88.68	91.46	90.03	0.2614	99.51
Linear Projection, Positional & Mixture	89.10	87.55	90.96	89.19	0.2888	99.43
Linear Projection, Positional & GMLP	90.65	89.31	92.29	90.77	0.2490	99.56
Linear Projection, Positional & Fourier	88.26	85.78	90.86	88.22	0.3158	99.35
Linear Projection, Positional, Mixture, GMLP & Fourier (All)	90.98	89.64	92.43	91.00	0.2425	99.54
Weighted Embedding	**93.57**	**91.67**	**92.76**	**92.21**	**0.2307**	**99.38**

TABLE 3. Comparative Analysis on Fashion-MNIST data with separate embeddings without local positional & projection embedding

Embeddings	Accuracy	Recall	Precision	F1-Score	Loss	AUC
Mixer Embedding	88.52	86.33	90.70	88.42	0.3110	99.38
GMLP Embedding	89.97	88.67	91.63	90.10	0.2635	99.53
Fourier Embedding	87.10	84.68	89.90	87.19	0.3365	99.25

SWEKP that have varying embedding settings for encoding purposes.

B. COMPARATIVE ANALYSIS ON FASHION-MNIST DATASET

Large amounts of data are crucial for deep learning models as they are highly dependent on data. To address this requirement, the Mix-up data augmentation technique is used to augment the data. The data was split into training and testing portions [80:20], with 50,000 images in the training set and 10,000 images in the testing set. After undergoing basic preprocessing, the algorithms were trained. Upon completion of training, the models were tested on the testing data to evaluate their performance on unseen data. The results clearly demonstrate the superiority of the proposed SWEKP based ViT over other architectures and are listed in Table 2.

The first column in the table represents the different types of embeddings used for encoding the patches in the transformer. Initially, in the simple ViT, there were two types of embeddings: Linear Projection and Positional Embedding. However, we introduced three additional types of latent embeddings: Mixture, GMLP, and Fourier. This brought the total number of embedding types to five, allowing us to conduct multiple experiments by changing the embeddings. For example, we first used the default Linear Projection and Positional Embedding and analyzed the results. Then, we added the Mixture Embedding and recorded the performance. Similar experiments were conducted for each combination of embeddings.

According to Table 2, we found that the SWEKP embedding performed exceptionally well in terms of accuracy, precision, f1-score, and recall. The accuracy achieved by the proposed weighted embeddings is 93.57%.

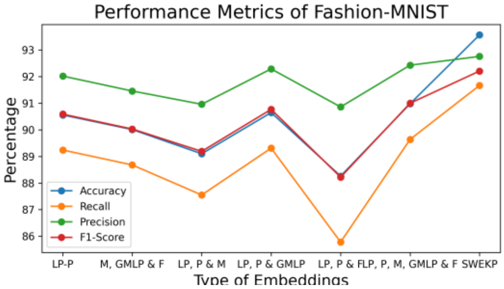


FIGURE 8. Comparative analysis of embedding on Fashion-MNIST dataset.

In all other sub-versions of SWEKP-based ViT, we used a linear sum of embeddings. However, in SWEKP Embedding, we used a weighted sum of embeddings, where each embedding was assigned a weight. For more information on SWEKP, refer to Section 3 (Materials and Methods). The performance of the embeddings are visually represented in Figure 8. By examining figure, which displays the accuracy, recall, precision, and F1-score for each embedding method, including LP, P, GMLP, F, and M, representing linear projection, positional embedding, GMLP embedding, Fourier embedding, and Mixture embedding, respectively, we can observe that the SWEKP embedding proposed in this study performs significantly better than other embeddings.

1) PERFORMANCE OF SINGLE EMBEDDING ON FASHION-MNIST

In this section, we evaluate the performance of each advanced embedding separately. Firstly, we use the GMLP embedding and report the results, then we use the Mixture embedding, and finally, we use the Fourier embedding. We conduct this experiment on each of the four datasets. In this

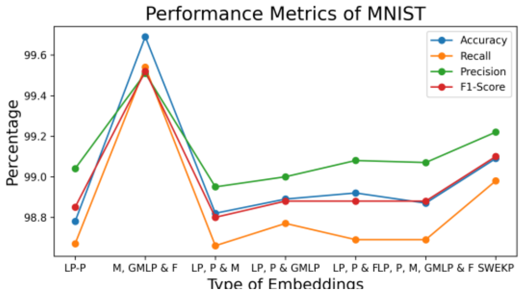


FIGURE 9 Comparative analysis of embedding on MNIST dataset.

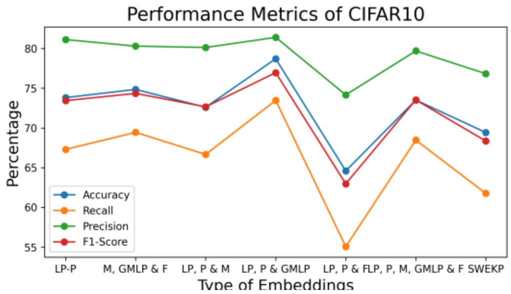


FIGURE 10 Comparative analysis of embedding on CIFAR10 dataset.

section, we compare the performance on the Fashion-MNIST data, given in Table 3. The performance results lead to a conclusion that the GMLP performs the best compared to the other two embeddings.

C. COMPARATIVE ANALYSIS ON MNIST DATASET

MNIST is a benchmark dataset consisting of numbers between 0 and 9. The data is split into training and testing parts, with 60000 images in the training set and 10000 images in the testing set. After basic preprocessing, the algorithms are trained. The models are then tested on the testing data to evaluate their performance on unseen data. The results clearly show that the ViT with only Mixture, GMLP, and Fourier embeddings, as well as the proposed SWEKP based ViT, perform very well. The results are listed in Table 4.

We have discussed the embeddings in detail in Section 3, and from now on, we will only analyze and discuss performance. As seen in Table 4, the ViT with only Mixture, GMLP, and Fourier embeddings and the SWEKP Embedding (with DKD) perform exceptionally well compared to others in terms of all performance measures. The combined GMLP, Fourier, and Mixture embeddings achieved the highest accuracy of 99.69%. It is important to note that the ViT with only Mixture, GMLP, and Fourier embeddings does not contain the DKD mechanism for knowledge dispersion. It can be concluded that the three new embeddings help the transformer to decode more efficiently compared to the default two embeddings (Linear projection and Positional). The performance of embeddings on MNIST data is visually presented in Figure 9.

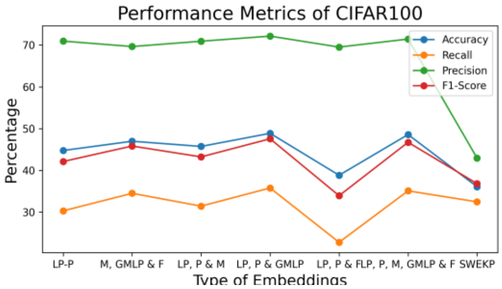


FIGURE 11 Comparative analysis of embedding on CIFAR100 dataset.

1) PERFORMANCE OF SINGLE EMBEDDING ON MNIST

In this section, we evaluate the performance of each advanced embedding separately. Firstly, we use the gMLP embedding and report the results, then we use the Mixture Embedding, and finally, we use the Fourier embedding. The results for the MNIST data are presented in Table 5 and the performance of each embedding can be observed. The GMLP performs the best compared to the other two embeddings.

D. COMPARATIVE ANALYSIS ON CIFAR10 DATASET

CIFAR10 is a benchmark dataset consisting of general images with 10 classes. The data is split into training and testing parts, with 50000 images in the training set and 10000 images in the testing set. After basic preprocessing, the algorithms are trained. The models are then tested on the testing data to evaluate their performance on unseen data. The results clearly demonstrate the superiority of the proposed SWEKP based ViT over other architectures and are listed in Table 7. As seen in the table, the ViT with Linear Projection, Positional, and GMLP embedding and the SWEKP Embedding (with DKD) perform exceptionally well compared to others in terms of all performance measures. The combined Linear Projection, Positional, and GMLP embeddings achieved the highest accuracy of 78.71%.

1) PERFORMANCE OF SINGLE EMBEDDING ON CIFAR10

In this section, we evaluate the performance of each advanced embedding separately. Firstly, we use the GMLP embedding and report the results, then we use the Mixture embedding, and finally, we use the Fourier embedding. The results for the CIFAR10 data are shown in Table 6. The visual performance of the embedding on CIFAR10 is illustrated in Figure 10.

TABLE 4 Comparative Analysis on MNIST data

Embeddings	Accuracy	Recall	Precision	F1-Score	Loss	AUC
Linear Projection & Positional	98.78	98.67	99.04	98.85	0.0342	99.98
Mixture, GMLP & Fourier	**99.69**	**99.54**	**99.51**	**99.52**	**0.0262**	**99.98**
Linear Projection, Positional & Mixture	98.82	98.66	98.95	98.80	0.0341	99.97
Linear Projection, Positional & GMLP	98.89	98.77	99.00	98.88	0.0305	99.98
Linear Projection, Positional & Fourier	98.92	98.69	99.08	98.88	0.0334	99.98
Linear Projection, Positional , Mixture, GMLP & Fourier (All)	98.87	98.69	99.07	98.88	0.0363	99.98
Weighted Embedding	**99.09**	**98.98**	**99.22**	**99.10**	**0.0297**	**99.96**

TABLE 5 Comparative Analysis on MNIST data with separate embeddings without local positional & projection embedding

Embeddings	Accuracy	Recall	Precision	F1-Score	Loss	AUC
Mixer Embedding	98.72	98.53	98.85	98.69	0.0383	99.99
GMLP Embedding	98.81	98.69	98.89	98.79	0.0380	99.93
Fourier Embedding	97.53	97.12	97.89	97.50	0.0728	99.93

TABLE 6 Comparative Analysis on CIFAR10 with separate embeddings without local positional & projection embedding

Embeddings	Accuracy	Recall	Precision	F1-Score	Loss	AUC
Mixer Embedding	70.62	63.24	79.21	70.17	0.8441	95.84
GMLP Embedding	74.80	69.60	80.60	74.57	0.7474	96.67
Fourier Embedding	59.77	50.02	70.08	58.14	1.1289	92.89

TABLE 7 Comparative Analysis on CIFAR10 Data

Embeddings	Accuracy	Recall	Precision	F1-Score	Loss	AUC
Linear Projection & Positional	73.82	67.30	81.13	73.44	0.7410	96.78
Mixture, GMLP & Fourier	74.85	69.44	80.30	74.35	0.7492	96.69
Linear Projection, Positional & Mixture	72.60	66.66	80.13	72.66	0.7888	96.35
Linear Projection, Positional & GMLP	**78.71**	**73.47**	**81.41**	**76.96**	**0.6915**	**98.54**
Linear Projection, Positional & Fourier	64.59	55.05	74.16	62.97	0.9999	94.34
Linear Projection, Positional, Mixture, GMLP & Fourier (All)	73.51	68.47	79.70	73.53	0.7957	96.29
weighted Embedding	69.42	61.75	76.83	68.35	0.9375	98.43

TABLE 8 Comparative Analysis on CIFAR100 Dataset

Embeddings	Accuracy	Recall	Precision	F1-Score	Loss	AUC
Linear Projection & Positional	44.76	30.32	70.94	42.13	2.1732	93.58
Mixture, GMLP & Fourier	46.98	34.52	69.64	45.83	2.0617	93.77
Linear Projection, Positional & Mixture	45.74	31.45	70.91	43.23	2.1164	93.79
Linear Projection, Positional & GMLP	**48.88**	**35.80**	**72.13**	**47.57**	**1.9936**	**94.35**
Linear Projection, Positional & Fourier	38.84	22.80	69.51	33.95	2.4206	92.36
Linear Projection, Positional , Mixture, GMLP & Fourier (All)	48.58	35.11	71.43	46.73	2.0108	94.17
weighted Embedding	36.10	32.48	42.95	36.83	3.8518	82.72

TABLE 9 Comparative analysis on CIFAR100 data with separate embeddings without local positional & projection embeddings

Embeddings	Accuracy	Recall	Precision	F1-Score	Loss	AUC
Mixer Embedding	45.13	31.61	69.98	43.18	2.1588	93.44
GMLP Embedding	48.81	36.10	71.04	47.60	1.9881	94.31
Fourier Embedding	36.90	20.80	69.10	31.62	2.5022	91.97

TABLE 10 Comparison with state-of-the-art architectures in term of accuracy

Reference	Model	CIFAR10	Fashion-MNIST	MNIST	CIFAR100
Khanday O et al [17]	Modified CNN	73.04%	93.68%	-	-
Kurt M a W et al [18]	HSIC (Hilbert-Schmidt Independence Criterion)	59.4%	88.3%	98.8%	-
Kurt M a W et al[18]	Simple CNN with back propagation	56.5%	87.6%	98.4%	-
Chang S et al[21]	MLSCN	76.79%	-	98.42%	-
Hassani et al.[34]	ViT-12/16	69.82%	-	99.63%	57.97%
Hassani et al.[34]	ViT-Lite-7/16	71.78%	-	-	52.87%
Hassani et al. [34]	ViT-Lite 6/16	78.12%	93.09%	99.66%	52.68%
Proposed	**SWEKP Embeddings**	**78.71**	**93.57%**	**99.69%**	**48.88%**

The Table 7 shows the performance of each architecture, and upon analysis, we conclude that the proposed SWEKP-ViT performs better than the other architectures.

E. COMPARATIVE ANALYSIS ON CIFAR100 DATASET

CIFAR100 is a benchmark dataset consisting of general images with 100 classes. The data is split into training and testing parts, with 50000 images in the training set and 10000 images in the testing set. After basic preprocessing, the algorithms are trained. The models are then tested on the testing data to evaluate their performance on unseen data. The results clearly demonstrate the superiority of the proposed SWEKP-based ViT over other architectures and are listed in Table 8.

Almost every model performs averagely on the CIFAR100 dataset. However, upon comparison of the performance of each model, it turns out that SWEKP based ViT, ViT with Linear Projection, Positional, and GMLP perform exceptionally well. The results are less than 50% due to the presence of 100 classes, some of which may have similar visual representations, causing confusion for the models. However, when compared to each other, SWEKP-ViT, ViT with Linear Projection, Positional, and GMLP obtain very good results compared to the others.

1) PERFORMANCE OF SINGLE EMBEDDING ON CIFAR100

In this section, we evaluate the performance of each advanced embedding separately. Firstly, we use the GMLP embedding and report the results. Then, we use the Mixture embedding and lastly, we have the results for the Fourier embedding. The results for the CIFAR100 data are shown in Table 9.

The table shows the performance of each embedding. Upon analysis, we conclude that the GMLP performs the best compared to the other two embeddings. The Figure 11 show the performance of embedding on CIFAR100 dataset.

F. COMPARISON OF PREVIOUSLY PROPOSED ARCHITECTURE ON BENCHMARK DATASET

In the last we compare our proposed architecture with previously proposed state of the art models. The Table 10 showcases some results from previously proposed algorithms on benchmark datasets. Our results are highly competitive and could potentially be improved with an increased number of epochs and optimization of hyperparameters. The primary objective of this paper is to propose a heterogeneous patch encoding-based ViT. We conduct a comprehensive comparative analysis and multiple experiments, demonstrating that SWEKP mixtures performs exceptionally well on benchmark datasets. Through rigorous experimentation, we have reached the conclusion that our proposed set of modifications applied to the Vision Transformer (ViT) model has led to notable performance improvements when evaluated on benchmark datasets. The results demonstrate the superiority of our enhanced approach compared to other existing state-of-the-art models.

G. COMPLEXITY ANALYSIS

In this study, the SWEKP patch encoding mechanism for transformers is introduced. With numerous transformer variations existing in the literature, we conduct a comprehensive complexity analysis of our proposed SWEKPT-based ViT in comparison to other state-of-the-art transformer models. MNIST, Fashion-MNIST, CIFAR10, and CIFAR100 are among the benchmark datasets that were utilized for evaluation. Table 11 presents the complexity analysis of various transformer-based architectures, focusing on their parameter characteristics. The performance metrics of these architectures, as outlined in Table 10.

The Table 11 clearly shows that state-of-the-art transformer models have a large number of trainable parameters. However, our proposed transformer architecture stands out with significantly fewer parameters, approximately 1.4 million. In terms of computational efficiency, our proposed architecture outperforms other state-of-the-art architectures.

VI. CONSLUSION

TABLE 11 **Complexity comparison on the basis of parameters**

No	Model	Parameters
1	ViT 12/16	85.63M
2	ViT 7/16	3.89M
3	ViT 6/16	3.36M
4	SWEKP-ViT	1.407M

Vision transformers are state of the art architectures for image recognition tasks. A ViT consist of two parts firstly we break image into patches and then encode the patches into latent representations, and for encoding we have two embedding's linear projection and positional embedding in simple ViT. In this research we proposed three new types of embeddings (Mixture, Spatial Gated and Fourier) for efficient patch encoding in vision transformer. The development of the proposed SWEKP-based ViT involves two distinct phases. First is weighted combination of previous two and new three embedding for patch encoding in transformer, and second is DKD for information propagation. Exploring different combinations of embeddings and conducting multiple experiments allows us to determine the optimal combination that gives the highest performance. We perform a massive comparative analysis on different combination of embedding. For example, firstly we use Mixture with default embedding's (linear projection and positional embedding) to see the effect on patch encoding mechanism. Then we use some different combination and record the results. We come up with a conclusion that two types of combination perform very well. First is weighted combination of all embedding's and second is GMLP with default embeddings (linear projection and positional embedding). We achieve higher results compared to other state of the art models present in literature. Based on our extensive testing and incorporating the suggested modifications, we observed a significant improvement in the performance of the enhanced ViT model.

ACKNOWLEDGMENT

This study is supported via funding from Prince Sattam bin Abdulaziz University, project number (PSAU/2023/R/1444), and Princess Nourah bint Abdulrahman University Researchers Support, Project number (PNURSP2023R125), Riyadh, Saudi Arabia.

REFERENCES

[1] J. Hoffmann, O. Navarro, K. Florian, B. Janßen, and H. Michael, “A Survey on CNN and RNN Implementations,” *Seventh Int. Conf. Performance, Saf. Robustness Complex Syst. Appl. Pesaro 2017*, 2017.

[2] D. Bhatt *et al.*, “Cnn variants for computer vision: History, architecture, application, challenges and future scope,” *Electronics (Switzerland)*, vol. 10, no. 20. 2021. doi: 10.3390/electronics10202470.

[3] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, vol. 2017-December.

[4] X. Zhai *et al.*, “Vision Transformer,” *arXiv:2010.11929*, 2021.

[5] H. Wu *et al.*, “CvT: Introducing Convolutions to Vision Transformers,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2021. doi: 10.1109/ICCV48922.2021.00009.

[6] M. Is, R. For, and E. At, “An image is worth 16x16 words: visual image transformer,” in *Iclr*, 2021.

[7] J. Hofmann, “Deit,” *Leb. Zeitung*, vol. 73, no. 26, pp. 50–50, 2021, doi: 10.51202/0947-7527-2021-26-050-1.

[8] K. Han *et al.*, “A survey on visual transformer,” *arXiv*. 2020.

[9] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in Vision: A Survey,” *ACM Comput. Surv.*, vol. 54, no. 10s, 2022, doi: 10.1145/3505244.

[10] D. Marin, J.-H. R. Chang, A. Ranjan, A. Prabhu, M. Rastegari, and O. Tuzel, “Token Pooling in Vision Transformers,” 2021.

[11] X. Yue *et al.*, “Vision Transformer with Progressive Sampling,” Feb. 2022, pp. 377–386. doi: 10.1109/iccv48922.2021.00044.

[12] K. Han *et al.*, “A Survey on Vision Transformer,” *IEEE Trans. Pattern Anal. Mach. Intell.*, Jan. 2022, doi: 10.1109/TPAMI.2022.3152247.

[13] H. Fan *et al.*, “Multiscale Vision Transformers,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 6804–6815. doi: 10.1109/ICCV48922.2021.00675.

[14] R. Ranftl, A. Bochkovskiy, and V. Koltun, “Vision Transformers for Dense Prediction,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 12159–12168. doi: 10.1109/ICCV48922.2021.01196.

[15] W. Li *et al.*, “SepViT: Separable Vision Transformer,” Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.15380

[16] P. Zhang *et al.*, “Multi-Scale Vision Longformer: A New Vision Transformer for High-Resolution Image Encoding,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 2978–2988. doi: 10.1109/ICCV48922.2021.00299.

[17] O. M. Khanday, S. Dadvandipour, and M. A. Lone, “Effect of filter sizes on image classification in CNN: A case study on CFIR10 and fashion-MNIST datasets,” *IAES Int. J. Artif. Intell.*, vol. 10, no. 4, pp. 872–878, Dec. 2021, doi: 10.11591/ijai.v10.i4.pp872-878.

[18] W. D. Kurt Ma, J. P. Lewis, and W. Bastiaan Kleijn, “The HSIC bottleneck: Deep learning without back-propagation,” in *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, 2020, pp.

5085–5092. doi: 10.1609/aaai.v34i04.5950.

[19] H. Li, H. Liu, X. Ji, G. Li, and L. Shi, "CIFAR10-DVS: An event-stream dataset for object classification," *Front. Neurosci.*, vol. 11, no. MAY, 2017, doi: 10.3389/fnins.2017.00309.

[20] G. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing," in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[21] S. Chang and J. Liu, "Multi-Lane Capsule Network for Classifying Images with Complex Background," *IEEE Access*, vol. 8, pp. 79876–79886, 2020, doi: 10.1109/ACCESS.2020.2990700.

[22] Z. He, "Deep Learning in Image Classification: A Survey Report," in *Proceedings - 2020 2nd International Conference on Information Technology and Computer Application, ITCA 2020*, 2020. doi: 10.1109/ITCA52113.2020.00043.

[23] C. Si *et al.*, "Better Robustness by More Coverage: Adversarial and Mixup Data Augmentation for Robust Finetuning," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021. doi: 10.18653/v1/2021.findings-acl.137.

[24] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[25] N. E. Khalifa, M. Loey, and S. Mirjalili, "A comprehensive survey of recent trends in deep learning for digital images augmentation," *Artif. Intell. Rev.*, vol. 55, no. 3, 2022, doi: 10.1007/s10462-021-10066-4.

[26] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, "A review of medical image data augmentation techniques for deep learning applications," *Journal of Medical Imaging and Radiation Oncology*, vol. 65, no. 5. 2021. doi: 10.1111/1754-9485.13261.

[27] T. Yu, X. Li, Y. Cai, M. Sun, and P. Li, "S2-MLP: Spatial-Shift MLP Architecture for Vision," in *Proceedings - 2022 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022*, 2022. doi: 10.1109/WACV51458.2022.00367.

[28] J. Tang, H. Kim, V. Guizilini, and S. Pillai, "Efficient Token Mixing for Transformers via Adaptive Fourier Neural Operators," *ICLR*, no. 2019, 2022.

[29] W. O. Saxton, "The discrete Fourier transform," in *Advances in Imaging and Electron Physics*, vol. 214, 2020. doi: 10.1016/bs.aiep.2020.04.002.

[30] M. Masala, S. Ruseti, and M. Dascalu, "RoBERT – A Romanian BERT Model," 2021. doi: 10.18653/v1/2020.coling-main.581.

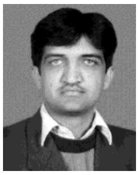
[31] I. Tolstikhin *et al.*, "MLP-Mixer: An all-MLP Architecture for Vision," in *Advances in Neural Information Processing Systems*, 2021, vol. 29.

[32] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontañón, "FNet: Mixing Tokens with Fourier Transforms," in *NAACL 2022 - 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 2022. doi: 10.18653/v1/2022.naacl-main.319.

[33] H. Liu, Z. Dai, D. R. So, and Q. V. Le, "Pay Attention to MLPs," in *Advances in Neural Information Processing Systems*, 2021, vol. 11.

[34] N. S. Ali Hassani, Steven Walton, A. Abuduweili, and H. S. , Jiachen Li, "Escaping the Big Data Paradigm with Compact Transformers," *arXiv*, vol. 1, p. 18, 2022, [Online]. Available: https://arxiv.org/pdf/2104.05704.pdf

Syed Muhammad Ahmed Hassan Shah was born in Kamra Village, District Attock, Pakistan in 2001. He received the B.S. degree in Computer Science from the COMSATS University Islamabad, Attock campus in 2022. Currently, he was a Research Assistant with the Medical Imaging and Diagnostic Lab, National Center of Artificial Intelligence, COMSATS Islamabad, Pakistan. His research interests include Computer Vision, Medical Imaging, Natural Language Processing, Computational Quantum Field Theory, Quantum Deep Learning, Graph Neural Networks, Evolutionary Optimization Techniques, Representation Theory, Generative Adversarial & Bayesian Learning.



MUHAMMAD QASIM KHAN received the master's degree in computer science from the University of Arid Agriculture, Rawalpindi, Pakistan, in 2011. He is currently pursuing the Ph.D. degree with the Department of Computer Science, university of engineering and technology Taxila, Pakistan. His current research interests include image processing and machine learning.

Yazeed Yasin Ghadi received his Ph.D. in Electrical and Computer Engineering from Queensland University. His dissertation on developing novel hybrid plasmonic photonic onchip biochemical sensors received the Sigma Xi best Ph.D. thesis award. He is currently an assistant professor of Software engineering at Al Ain University. He was a postdoc researcher at Queensland University before joining Al Ain. His current research is developing novel electro-acoustic-optic neural interfaces for large-scale high-resolution electrophysiology and distributed optogenetic stimulation. Yazeed has published over 80 peer-reviewed journal and conference papers and holds three pending patents. He is the recipient of several awards.

Sana Ullah Jan (Member, IEEE) received the B.S. degree in electronic engineering from International Islamic University, Islamabad, Pakistan, in 2012, and the combined M.S./Ph.D. degree from the University of Ulsan, Ulsan, South Korea. He is currently enrolled as Lecturer/Assistant Professor in Edinburgh Napier University, UK since September 2021. He was previously enrolled as Post-Doctoral Research Fellow at the Center of Affective and Human Computing for Smart Environment at the school of computing, engineering and physical sciences, University of the West of Scotland since September 2020 to August 2021. His research area is closely related to the Artificial Intelligence or Machine Learning-based cyber security and privacy in the Internet-of-Things, Cyber Physical Systems and eHealth.

Olfa Mzoughi received her Ph.D. degree in Computer Science (Signal and Image) from Telecom ParisTech, France, and the engineering and M.Sc. degrees in Telecommunications from the High School of Communications of Tunisia (SUPCOM). She is currently an Assistant Professor at the department of Computer Science, College of Sciences and Humanities, Prince Sattam bin Abdulaziz University. Her research interests include Computer vision, Image classification, Image processing, Image segmentation, Machine learning and Deep learning. Findings from her research have been published in many refereed publications.

Monia Hamdi is currently an associate professor in the College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, KSA. She received her B.Eng degree in Information Technology from Telecom SudParis, Paris-Saclay University, France, in 2008 and obtained in the same year the M.Sc. degree in Telecommunications and Networks from Institut National Polytechnique, Toulouse, France. She received the Ph.D. degree in Computer Science from the University Rennes 1, France in 2012. She worked from 2012 to 2017 as an assistant professor in the Higher Institute of Computer Science and Multimedia, Gabès University, Tunisia. From March to August 2015, she was a Visiting Researcher with the Department of Science and Technology, Linköping University, Sweden. Her research interests include Mobile communications, Wireless Sensor Networks, Edge computing, Internet of Things, and Artificial Intelligence.