WideCaps: a wide attention-based capsule network for image classification

S. J. Pawan · Rishi Sharma · Hemanth Reddy · M. Vani · Jeny Rajan

Abstract

The capsule network is a distinct and promising segment of the neural network family that has drawn attention due to its unique ability to maintain equivariance by preserving spatial relationships among the features. The capsule network has attained unprecedented success in image classification with datasets such as MNIST and affNIST by encoding the characteristic features into capsules and building a parse-tree structure. However, on datasets involving complex foreground and background regions, such as CIFAR-10 and CIFAR-100, the performance of the capsule network is suboptimal due to its naive data routing policy and incompetence in extracting complex features. This paper proposes a new design strategy for capsule network architectures for efficiently dealing with complex images. The proposed method incorporates the optimal placement of the novel wide bottleneck residual block and squeeze and excitation Attention Blocks into the capsule network upheld by the modified factorized machines routing algorithm to address the defined problem. This setup allows channel interdependencies at almost no computational cost, thereby enhancing the representation ability of capsules on complex images. We extensively evaluate the performance of the proposed model on the five publicly available datasets, namely the CIFAR-10, Fashion MNIST, Brain Tumor, SVHN, and the CIFAR-100 datasets. The proposed method outperformed the top-5 capsule network-based methods on Fashion MNIST, CIFAR-10, SVHN, Brain Tumor, and gave a highly competitive performance on the CIFAR-100 datasets.

Keywords Capsule network · Convolutional neural network · Image classification

1 Introduction

Since the rise of the convolutional neural network (CNN), it has fascinated researchers due to its working nature, inspired by the mechanism of mammalian visual cortex. Recently, CNNs have become more effective and have attained near-human-level performance on discriminative and generative tasks. We can observe the feat of CNNs from LeNet-5 [1], the first CNN-based neural network for recognizing handwritten characters. Later in 2012, Krizhevsky et al. introduced AlexNet [2], which won the annual ImageNet Large Scale Visual Recognition Challenge-2012 (ILSVRC) [3] by a large margin. Over the period, many other methods emerged, [4–9], attaining a state-of-the-art on the ILSVRC challenge.

Currently, CNNs are considered the gold standard in computer vision, natural language processing, speech processing, and other related disciplines.

Despite the massive success of CNN across multiple domains, there are several concerns with its information processing mechanism. It is worth noting that CNN’s neuronal design differs from that of the mammalian visual cortex; which prevents CNNs from extracting and encoding human-level information from data. CNNs are inept at coping with affine transformations as they cannot maintain spatial hierarchies among the features, resulting in poor generalization. The pooling operation, intended to curtail the spatial dimension of the feature space, comes at the cost of losing precise location and positional information, which affects CNN’s ability to represent the structural orientation of an entity. Furthermore, contrary to the human brain’s functionality, CNNs fail miserably to encode the information from the limited data samples. Moreover, it has been experimentally investigated and fortified that the patterns encoded by the CNNs are highly vulnerable and are prone to adversarial attacks [10–
image classification. The major contributions can be summarized as:

1. The capsule network architecture that utilizes wide bottleneck residual blocks along with the squeeze and excitation blocks (SE blocks) and Attention Capsules for accurately classifying complex images.

2. A modified FM routing algorithm to keep the range of prediction vectors bounded and the routing coefficients well distributed to improve the correlation between the parent and child capsules.

3. Extensively validate the performance of the proposed model on five publicly available datasets, namely Fashion MNIST, SVHN, CIFAR-10, CIFAR-100, and Brain Tumor dataset, to outperform the top-5 capsule network-based methods on CIFAR-10, Fashion MNIST, Brain Tumor, SVHN datasets with highly competitive performance on the CIFAR-100.

The remainder of the paper is structured as follows: Sect. 2 summarizes the recent studies that have significantly contributed to the capsule network literature of image classification. Section 3 provides a comprehensive overview of the proposed methodology. Section 4 narrates the data and experimental results. Section 5 concludes and provides a direction for future developments.

2 Literature survey

This section summarizes various capsule network-based classification methods proposed in the literature. In Table 1, we have listed the performance analysis of various methods against the benchmark datasets.

Many methods have been proposed in the literature by adopting structural modifications to improve the overall performance of the capsule network architectures. Xiang et al. [20] introduced a multi-scale capsule network (MS-CapsNet), to comprehend the hardware overhead and improve the feature representation capability. This method employs a multi-scale feature extraction block to achieve the objective. Further, MS-CapsNet incorporates a novel capsule dropout mechanism as a regularization technique. MS-CapsNet achieved an accuracy of 92.70%, 75.70% on Fashion MNIST and CIFAR-10 datasets, respectively. Phaye et al. [21] introduced two architectures, namely DCNet and DCNet++, to extract more complex features and thus form efficient and robust primary capsules. DCNet incorporates DenseNet as the backbone to extrapolate complex and diversified features. DCNet achieved an accuracy of 99.75% on the MNIST dataset. DCNet++ incorporates the primary capsules that carry scale information at multiple levels to diversify the patterns encoded by the capsules and improve the performance on complex datasets.

In 2011, Hinton et al. [14] introduced the theory of capsules. The capsules are the vectorized clusters of neurons whose activity vector represents the instantiation parameters of a visual entity, and the length of the vector depicts the probability of its presence. Inspired by Hinton’s doctrine of capsules, Sabour et al. [15] introduced a novel training mechanism called dynamic routing between the capsules for iteratively training the capsule network architecture, which bestowed tremendous potential compared to conventional CNNs on multiple datasets. The capsule network circumvents the pooling operation by carefully employing the routing mechanism to propagate elegant signals or features from lower to higher-level layers, making the capsule network translational equivariance. The activated capsules in one layer make pose predictions for the capsules in the next layer via transformation matrices. Then, the routing algorithm finds a center of mass of all the predictions using an iterative clustering algorithm. This approach ensures the propagation of relevant features to the subsequent layers, i.e., more prediction vectors agreeing to the parent capsules causing strong activation. This inherently results in acquiring more pertinent features representing a visual entity. Besides, the routing algorithm with capsules also benefits in encoding the vital and relevant information from limited samples, making it exceptional from conventional CNNs. As suboptimalthough capsule network possesses all the inclinations toward setting up a new standard in the neural networks discipline, it has some inherent limitations [15] such as (1) The performance of the capsule networks is below par compared to the conventional CNNs on complex datasets or images involving complex background and foreground regions. (2) Poor optimization due to the complex nature of capsule network architecture. (3) The capsule network tries to encode all the information from the cluttered or complex data, which causes a detrimental effect on the formation of informative capsules. This work proposes a novel capsule network-based architecture for efficiently dealing with complex images for image classification. The major contributions can be summarized below:

1. This paper presents a new approach for designing a capsule network architecture that utilizes wide bottleneck residual blocks along with the squeeze and excitation blocks (SE blocks) and Attention Capsules for accurately classifying complex images.

2. We introduce a modified FM routing algorithm to keep the range of prediction vectors bounded and the routing coefficients well distributed to improve the correlation between the parent and child capsules.
network or DE-CapsNet, which utilizes residual blocks and position-wise dot product operations to develop enhanced primary capsules that carry scale information at multiple levels for dealing with complex data. DE-CapsNet attained an accuracy of 94.25% and 92.96% on the Fashion MNIST and CIFAR-10 datasets, respectively. Sun et al. [24] introduced a dense capsule network called DenseCaps to improve the performance of capsules on complex datasets. DenseCaps allows feature-reuse and incorporates cross-capsule feature concatenation techniques to extract and encode salient features from complex data. DenseCaps achieved an accuracy of 99.70%, 94.93%, 95.99%, and 89.41% on MNIST, Fashion MNIST, SVHN, and CIFAR-10 datasets, respectively.

To mitigate the computation complexity and the massive amount of transformation matrices incurred by Sabour’s method, Hinton et al. [25] introduced a new routing method called the expectation maximization or EM routing algorithm. Further, the method introduced matrix capsules to encode the properties of an entity or feature. Capsules trained with the EM algorithm attained state-of-the-art results on the SmallNorb dataset [26]; however, the performance on CIFAR-10 remained sub-par. Deliege et al. [27] introduced HitNet, which employs a novel hit or miss (HoM) layer with the centripetal loss function by substituting the classification capsules and the margin loss function of Sabour’s method. Also, HitNet presented the concept of ghost capsules for classifying the mislabeled training data. HitNet achieved an accuracy of 83.03%, 94.50%, 73.30%, 92.30%, 99.62% on affNIST,1 SVHN, CIFAR-10, Fashion MNIST, and MNIST datasets, respectively. Wang et al. [28] addressed the routing algorithm as an optimization problem and introduced a novel routing technique similar to the agglomerative fuzzy k-means algorithm. Fuchs and Pernkopf [29] introduced the Wasserstein objective function to optimize the performance of the capsule network. The Wasserstein capsule network (W-CapsNet) facilitated scalability with relatively fewer trainable parameters to improve the performance on complex datasets. W-CapsNet achieved an accuracy of 70.39%, 93.43% on the CIFAR-100 and CIFAR-10 datasets, respectively. Zhao et al. [30] proposed a factorized machines (FM) routing algorithm to subdue the expensive computation complexity and minimize the hardware requirements. Unlike Sabour’s approach, where all the primary capsules contribute to predicting the output of the secondary-level capsules, in FM routing, the pair capsules will look for an agreement among themselves; if there is a strong agreement, these capsules approximate the secondary capsules output. However, the prediction vectors of FM routing were not bounded, causing the non-uniform distribution of routing coefficients. FM routing was evaluated on affNIST1, SVHN, CIFAR-10, Fashion MNIST, and MNIST datasets, and achieved an accuracy of 93.85%, 96.79%, 93.70%, and 94.70%, respectively. Rezwan et al. [31] introduced a simplistic data routing technique called iteration-free routing to minimize the number of routing iterations required for the convergence of capsule network architecture. Further, the method focuses on curtailing train-
ing time and resource overhead. This architecture uses a new class of capsules called a *single matrix* to achieve the objective. Iteration-free routing attained an accuracy of 75.85%, 94.72% on CIFAR-100 and CIFAR-10 datasets.

Rajasegaran et al. [32] proposed a new approach called DeepCaps by introducing a 3D convolution-based routing algorithm to expedite depth in the capsule network architecture by curtailing the number of trainable parameters. Further, DeepCaps incorporates a novel *class independent decoder* as a regularizer to aid the performance. DeepCaps attained an accuracy of 99.72%, 94.73%, 97.56%, 92.74% on MNIST, Fashion MNIST, SVHN, and CIFAR-10 datasets, respectively. A tensor-based capsule network called DeepTensorCaps was introduced by Sun et al. [33] to improve the performance of capsule networks on complex data. Unlike vector capsules, they claim that the tensor capsules are assertive in encoding complex features. DeepTensorCaps incorporates tensor dropout and multi-scale decoder networks as regularizers to aid the performance of the model. DeepTensorCaps archived an accuracy of 97.41%, 92.87%, and 95.50% on SVHN CIFAR-10 and Fashion MNIST datasets, respectively. Motivated by Sabour’s approach [15] of representing the features as *parse-tree* structure, Yang et al. [34] introduced RS-CapsNet to form *bigger-part with the help of intermediate-capsules* to improve the overall performance of the capsule network. RS-CapsNet achieved an accuracy of 97.08%, 91.32%, and 94.08% on SVHN, CIFAR-10, and Fashion MNIST datasets, respectively. In [35], the authors have presented a lightweight model named MobileCaps by fusing CNN and capsule network for classification followed by severity analysis of chest X-ray.

The attention mechanism has been introduced in the neural network literature focusing on paying attention to relevant features and suppressing the irrelevant ones. Li et al. [36] presented a feature pyramid network to induce attention from diverse pyramid scales followed by global average pooling guiding low-level features. In [37], Dong et al. presented a dual attention mechanism by combining channel and position attention modules for generating attention masks. Huang et al. [38] presented a reverse attention mechanism to facilitate a reverse-class response for identifying the regions outside the ROI. Dong et al. [39] presented a deep attention network coupled with a deep supervision mechanism to induce attention. Choi et al. [40] presented attention routing and capsule activation function by replacing naive dynamic routing with the squash activation function to preserve spatial information and improve representation. In [41], Huang et al. presented a dual attention framework by facilitating both conv-attention and caps-attention, improving the overall performance of the capsule network. Mazzia et al. [42] extended the idea of a capsule network by employing a self-attention mechanism and reducing the trainable parameters drastically to improve performance. Tsai et al. [43] introduced a new routing algorithm by employing the following design changes, (1) layer normalization, (2) adopting concurrent routing instead of iterative routing, and (3) routing with inverted dot product attention. Ahmed and Torresani introduced StarCaps [44] to constrain the training time associated with the capsule network. Further, StarCaps incorporates a novel straight-through attentive routing procedure by employing attention blocks. StarCaps enabled an efficient way to calculate the routing coefficient without any recurrence. In [45], the authors have conducted a detailed analysis of capsule networks. Moreover, the capsule network is extended to various domains, including image segmentation [46, 47], reconstruction [48], and generative models [49] illustrating its widespread application.

### 3 Proposed methodology

To improve the performance of the capsule network on complex data, one of the possible dimensions is fusing the functional units of CNNs into capsule networks by retaining the *capsulness* nature throughout the network. In this direction, we introduce a wide capsule network or WideCaps to address the inherent limitations of the capsule network architectures on complex datasets. Figure 1 depicts the block diagram of the proposed WideCaps architecture.

#### 3.1 The wide bottleneck residual and SE block

Inspired by the recent success of *normalizer-free* networks [50] (NF-Net), we strive to achieve the formulation of emphatic primary capsules and thus configure an efficient classification workflow. A wide capsule network architecture entails the convolutional stem, a novel wide bottleneck residual and SE block as the backbone, followed by the *Attention Capsules*. An input image is subjected to the stem consisting of four successive convolutional layers, and the resultant output is subjected to the subsequent backbone architecture to generate salient feature maps. The convolutional stem comprises four 2D successive convolutional layers with 16, 32, 64, and 128 kernels each of shape $(3 \times 3)$ having strides $(1 \times 1)$ with an interleaved ReLU activation function [51]. The output of the convolutional stem is then sent to the backbone architecture to make high-level feature maps. The backbone architecture contains three successive wide bottleneck residual and SE blocks in sequence. Each block has 4, 8, and 4 ResNet-SE layers, respectively.

We adopt a wide bottleneck residual block (all the bottleneck residual blocks are of projection version) over the standard bottleneck residual block; due to its efficient performance with fewer kernels, thereby curtailing the computation cost without affecting the performance. Figure 2 represents the schematic representation of the standard bottleneck residual block (a) and a wide bottleneck residual block (b). The
A wide attention-based capsule network for image classification

WideCaps: a wide attention-based capsule network for image classification

Fig. 1 An overview of the proposed Wide Capsule Network architecture

Fig. 2 Block diagram of a standard bottleneck residual block and b wide bottleneck residual block

bottleneck residual block includes three convolutional layers with filters \((1 \times 1), (3 \times 3),\) and \((1 \times 1)\), respectively interleaved with ReLU activation and batch normalization [52]. Additionally, it incorporates a \((1 \times 1)\) convolutional layer in the skip connections with appropriate filters to match the dimension. Finally, the feature maps from both ends are aggregated, and the resultant output is passed onto the ReLU activation, followed by the rest of the network. We modify the bottleneck residual block by halving the number of filters \(f\) of the bottleneck residual block to \(f/2\) in the first two convolutional layers, followed by multiplying it with 4 to regain the original number of filters to form a wide bottleneck residual block.

Further, we integrate wide bottleneck residual blocks with SE blocks to formulate more robust and appropriate feature maps. The output of the backbone architecture is reshaped to form primary capsules. A convolutional layer inside the primary capsule layer downsamples the tensor generated by the backbone architecture, followed by the batch normalization layer. This tensor is further converted into capsules by reshaping (e.g., a tensor with shape \(H \times W \times C\) is reshaped into \(H \times W \times C' \times K\), where \(H\), \(W\), and \(C\) denote height, width, and channel, respectively, and \(K\) is the number of neurons in a capsule), followed by a nonlinear squash activation function [15]. Further, the output capsules of the primary capsules are passed onto the attention capsules to generate output capsules.
3.2 Squeeze and excitation attention-based capsules

The capsule network has an inherent limitation of being responsive to noisy and irrelevant features causing significant performance degradation in complex images [15]. To alleviate this, we introduced Attention Capsules that focus on the most prominent parts of the image by diminishing the features that are unlikely to have any significance. Figure 3 represents the block diagram of SE attention capsules. The attention mechanism guides the capsules where to look for essential features in the image by employing the attention of the network using depth features captured by the capsules. The attention mechanism uses this correlation extracted by the capsules to generate attention maps. Further, it helps to improve the channel interdependencies, thereby enhancing the overall representation ability of the capsules.

The attention capsules are formulated by transforming the classification capsules into a vector represented by $F$, as shown in Fig. 3. In the subsequent stage, the resultant vector $F$ is propagated to the SE attention module. In the SE module, the features are subjected to a squeeze operation, wherein every channel of the input feature $F$ is squeezed to a single numeric value by applying an average pooling operation, resulting in $F_{avg}$, and then passed to the excitation operation. The excitation block consists of 2 fully connected layers activated by ReLU and sigmoid operation, resulting in $O_{ReLU}$ and $O_{sigmoid}$, respectively. Finally, the output of the SE attention model is multiplied by the input features $F$ to get the output features $F'$. Equations 1–4 [41] delineate the working of the SE attention model. Table 2 provides the architectural detail of the proposed method.

$$F_{avg} = \text{AvgPooling}(F)$$ (1)

$$O_{ReLU} = \text{ReLU}(W_1 \ast F_{avg} + b_1)$$ (2)

$$M_{SE} = \text{sigmoid}(W_2 \ast O_{ReLU} + b_2)$$ (3)

$$F' = F \ast M_{SE}$$ (4)

3.3 Modified FM routing algorithm

We introduce a modified FM routing algorithm to endure the range of bounded prediction vectors and distribute the routing coefficients well to improve the correlation between the parent and child capsules. We achieve this by adopting the softmax activation function over exp activation adopted by [30], which resulted in an uneven distribution of routing coefficients. Softmax enables the relevant features to get a high value and the counterpart. The working of the modified FM routing algorithm is presented below.

Every capsule in the layer $l - 1$ is denoted by $u_i$, and the prediction vector $\hat{u}_{ji}$ for every capsule in the layer $l$ is computed. Let the set of prediction vectors be $[\hat{u}_{j1}, \hat{u}_{j2}, \ldots , \hat{u}_{jn}]$, where the agreement between the capsules is formed through pairwise interactions among the capsules in the same layer. The pairwise product is established as $\hat{b}_{ji} = \hat{u}_{ji1} \circ \hat{u}_{ji2}$, where the sum of each element of $\hat{b}_{ji1}, \hat{b}_{ji2}$ gives the magnitude of agreement and also the orientation and pose of capsule $j$.

In general, the pairwise interaction of all capsules in layer $l - 1$ with the capsule $j$ of layer $l$ can be formulated as in Eq. 5 [30].

$$\hat{H}_j = \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \hat{u}_{ji1} \circ \hat{u}_{ji2}$$

$$= \frac{1}{2} \left( \sum_{i=1}^{n} \hat{u}_{ji} \circ \sum_{i=1}^{n} \hat{u}_{ji} - \sum_{i=1}^{n} \hat{u}_{jj1} \hat{u}_{jj2} \right)$$

where $\hat{u}_{ji} = [\hat{u}_{j1}, \hat{u}_{j2}, \ldots , \hat{u}_{jn}]$, $\hat{H}_j = [\hat{H}_{j1}, \hat{H}_{j2}, \ldots , \hat{H}_{jn}]$, and $n$ denote the total number of prediction vectors. Then, the output of capsule $j$ in layer $l$ is defined as in Eq. 6 [30].
Table 2  The architectural design of the proposed Wide Capsule Network architecture

| Block   | Type                  | Operation                           |
|---------|-----------------------|-------------------------------------|
| Stem    | Convolution           | Conv,[3 × 3]×16                      |
|         |                       | Conv,[3 × 3]×32                      |
|         |                       | Conv,[3 × 3]×64                      |
|         |                       | Conv,[3 × 3]×128                     |
| Backbone| Wide bottleneck residual SE| Conv,[1 × 1]×16         |
|         |                       | Conv,[3 × 3]×32                      |
|         |                       | Conv,[1 × 1]×64                      |
|         |                       | fc,[16 × 64]                          |
| Capsules| Wide bottleneck residual SE| Conv,[3 × 3]×16          |
|         |                       | Conv,[3 × 3]×128                     |
|         |                       | Conv,[3 × 3]×128                     |
|         |                       | fc,[16 × 128]                         |

Table 3  Performance analysis by subjecting proposed routing algorithm against the existing routing algorithm on CIFAR-10 dataset

| Methods               | Default | Proposed |
|-----------------------|---------|----------|
| Sabour et al. [15]    | 89.71   | 91.41    |
| Zhao et al. [30]      | 93.20   | 94.70    |
| Phaye et al. [21]     | 89.71   | 91.71    |

The bold letter depicts the superior performing method against the other competing methods.

\[
\hat{b}_j = \sum_{f=1}^{k} \hat{H}_{j,f} \tag{6}
\]

The coefficients of agreement are calculated by applying softmax function on the output as depicted in Eq. 7.

\[
\hat{x}_j = \text{softmax}(\hat{b}_j) \tag{7}
\]

The pose vector is defined as \( \hat{Q}_j = \frac{\hat{H}_j}{||\hat{H}_j||} \), where the direction of \( \hat{Q} \) determines the pose, orientation, size, rotation, etc., of an entity. As the summation operations present in Eqs. (5–6) could lead to the possible gradient explosion resulting in poor performance, the prediction vector \( \hat{u}_{ji} \) is scaled by dividing it with \( \sqrt{n} \) [30], which results in Eq. 8 [30]:

\[
\hat{H}_j = \frac{1}{2n} \left( \sum_{i=1}^{n} \hat{u}_{ji} \otimes \sum_{i=1}^{n} \hat{u}_{ji} - \sum_{i=1}^{n} \hat{u}_{ji} \otimes \hat{u}_{ji} \right) \tag{8}
\]

We use \( \hat{x}_j \) to determine which class is activated and by how much it has been activated. Asserting all the declarations and equations defined above, we conclude the above process in Algorithm 1 to be the modified version of FM routing [30]. In Table 3, we quantify the modification introduced by subjecting the modified FM routing algorithm to the following methods: [15, 21, 30] against their default routing algorithm on the CIFAR-10 dataset. We can observe that the proposed modification consistently outperforms the three methods considered for comparison with \( \geq 1.5\% \), which shows the efficiency of the proposed modification.

**Algorithm 1** Modified FM Routing

**Input**: Prediction vectors \( \hat{a}_j = (\hat{a}_{j1}, \hat{a}_{j2}, ..., \hat{a}_{jn}) \)

**Output**: \( \hat{Q}_j, \hat{x}_j \)

1. \( \hat{u}_{ji} \leftarrow L2\text{Normalize}(\hat{a}_{ji}) \)
2. \( \hat{H}_j \leftarrow \frac{1}{n} \sum_{j=1}^{n} \hat{u}_{ji} \otimes \sum_{i=1}^{n} \hat{u}_{ji} - \sum_{i=1}^{n} \hat{u}_{ji} \otimes \hat{u}_{ji} \)
3. \( \hat{Q}_j \leftarrow \frac{\hat{H}_j}{||\hat{H}_j||} \)
4. \( \hat{b}_j \leftarrow \sum_{j=1}^{N} \hat{H}_{j,f} \)
5. \( \hat{x}_j \leftarrow \text{softmax}(\hat{b}_j) \)
4 Experimental results and discussion

In this section, we brief on the datasets adopted in the study to evaluate the performance of the proposed model, followed by the hardware setup and training methodology. Further, we also compare and present the quantitative analysis of the proposed method with state-of-the-art methods.

4.1 Dataset description

1. Fashion MNIST dataset [16]: The Fashion MNIST dataset consists of 70,000 grayscale images, with 60,000 and 10,000 images for training and testing, respectively. The dataset is a collection of 10 fashion items, with 7000 samples per class. The dataset was introduced as a substitute to the MNIST dataset [54] with a slightly increased complexity level. Fashion MNIST is extensively used to evaluate the performance of capsule network models due to its higher complexity level than the naive MNIST dataset [54].

2. Street View House Numbers dataset [17]: SVHN is a real-time RGB dataset. It contains house numbers collected from Google Street View Images. SVHN is widely adopted for evaluating the performance analysis of machine learning models. The dataset consists of 70,000 images for training and 30,000 images for testing. The dataset includes images with diverse backgrounds, and hence, it is widely used to evaluate the performance of capsule network-based models.

3. Canadian Institute for Advanced Research-10 dataset [18]: CIFAR-10 is the most widely used tiny RGB dataset of 10 distinct classes acquired with diverse backgrounds and intensity levels. The dataset consists of 60,000 images for training and 10,000 images for testing. CIFAR-10 is the most complex dataset compared with the Fashion MNIST [16] and the SVHN [17]. As the performance of capsule network-based architectures is sub-par on complex data, CIFAR-10 is extensively used for evaluating the performance of capsule network-based architectures. Figure 5 shows the sample images from the above-discussed datasets.

4. Canadian Institute for Advanced Research-100 dataset [18]: CIFAR-100 is the higher-version CIFAR-10 dataset, consisting of 100 distinct classes. The dataset consists of 60,000 images, with 500 images per class for training, and 10,000 images, with 100 images per class for testing.

5. Brain Tumor classification dataset [19]: This dataset consists of 3064 samples (233 patients) of brain MRI images belonging to glioma (1426 images from 89 patients), pituitary (930 images from 62 patients), and meningioma (708 images from 82 patients), respectively.

4.2 Hardware setup and training methodology

All the experiments were conducted on a DGX workstation having 8 × NVIDIA Tesla V100 GPUs with 256 GB dedicated GPU memory on a Ubuntu 18.04 operating system, 64-bit Intel Xeon(R) Gold 5120 CPU @2.20 GHz × 28 processor, solid-state hard drive, 64 GB RAM. Further, the implementation was done in Python 3.6 with Keras [55] and TensorFlow 2 [56] as backend.

The initial weight matrix is initialized with HeNormal distribution [57]. We adopted an L2 regularizer to regulate the weight matrix. Further, we used stochastic gradient descent with momentum as the optimizer to train the network with a decay rate of 0.5 and momentum of 0.9. We adopted a drop-based learning rate scheduler strategy, with an initial learning rate (ILR) of 0.01 that halves the learning rate (LR) at every fixed number of epochs during the training process, as shown in Eq. 9 [55].

\[
LR = \text{ILR} \times \text{Drop Rate}^{(\text{Epoch}/\text{EpochDrop})}
\]  

Drop rate is the rate at which the learning rate is to be changed, and epoch drop is how frequently to change the learning rate. We empirically found that dropping the learning rate every 60 epochs would give better performance. We train our model from scratch with a batch size of 128 until convergence with the softmax categorical cross-entropy loss function [58] to better estimate the performance.

4.3 Results and discussion

We quantitatively evaluate and compare the performance of the proposed method with the other methods proposed in the literature by considering accuracy as the metric as defined in Eq. 10. In Table 4, we compare the efficacy of the proposed model with the top-5 capsule network-based methods on the Fashion MNIST, SVHN, CIFAR-10, and CIFAR-100 datasets, respectively. We can observe from Table 4 that the proposed model achieved a 7.14% improvement over the benchmark model [15] and a 1.71% improvement over the second-best method [43] on CIFAR-10 dataset, which is significant. However, it is worth noting that the result obtained for the benchmark model [15] was based on an ensemble of 7 models. Further, the proposed WideCaps also achieved a noticeable performance on the CIFAR-100 dataset with an accuracy of 77.80%, i.e., on par with [43]. Adding to it, the proposed method improved the performance on the Fashion MNIST dataset [16] by 0.10% over the best-performing model with highly competitive performance over the SVHN dataset [17]. Furthermore, the proposed method outperforms all the other studies in the literature that integrated the concept of attention into the capsule network over the CIFAR-10 dataset with highly
Table 4 Performance comparisons of the proposed method with top-5 capsule network-based methods on Fashion MNIST, SVHN, CIFAR-10, and CIFAR-100 datasets, respectively (Note: Unlike comparing with the same methods across the benchmark datasets, we compare the proposed method with the top-5 capsule network-based methods from Table 1 on the respective datasets)

| No | Fashion MNIST Models          | Accuracy | SVHN Models          | Accuracy |
|----|-------------------------------|----------|----------------------|----------|
| 1  | Sabour et al. [15]            | 94.65    | Phaye et al. [21]    | 95.58    |
| 2  | Phaye et al. [21]             | 94.65    | Sun et al. [33]      | 95.59    |
| 3  | Zhao et al. [30]              | 94.70    | Fuchs and Pernkopf [29] | 96.56    |
| 4  | Rajasegaran et al. [32]       | 94.73    | Yang et al. [34]     | 97.08    |
| 5  | Sun et al. [33]               | 95.50    | Sun et al. [33]      | 97.41    |
| 6  | Proposed                      | 95.98    | Proposed             | 98.10    |

The accuracy of the superior performing model is shown in bold.

Table 5 depicts the performance comparison of the proposed method with other methods integrating the notion of attention into the capsule network. We can observe that the proposed model achieved an improvement of 1.71% on CIFAR-10 with highly competitive performance on the CIFAR-10 datasets [18]. Also, in an interesting study, we compared the impact of the wider bottleneck residual block with that of the standard bottleneck residual block backbone on the proposed model against four public datasets. It is apparent from Fig. 6 that the wider bottleneck residual block consistently achieved the performance gain on all four datasets, depicting the significance.

Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)

The study centered on improving the performance of the proposed capsule network-based model on the CIFAR-10 dataset as it involves numerous challenges such as complex features, varying intensities, and distinct backgrounds with various noise levels. So, all the structural, algorithmic modifications followed by the hyperparameter tuning were performed based on the results obtained for the CIFAR-10 dataset. Also, it is more likely that if a model can perform well on CIFAR-10, then we can anticipate reasonably good performance on...
Table 5: Performance comparison of proposed method with other capsule network-based attention methods on CIFAR-10 and CIFAR-100 datasets

| Methods                        | CIFAR-10 | CIFAR-100 |
|-------------------------------|----------|-----------|
| Tsai et al. [43]              | 95.14    | 78.02     |
| Huang and Zhou [41]           | 85.47    | –         |
| Ahmed and Torresani [44]      | 91.23    | 67.66     |
| **Proposed**                  | **96.85**| **77.80** |

The bold letter depicts the superior performing method against the other competing methods.

The other datasets in general (Fig. 5 represents the learning curve obtained for the CIFAR-10 dataset). Further, to substantiate this, we evaluated the performance against the Brain Tumor dataset, and the results are depicted in Table 6. It is evident from Table 6 that the proposed model showed superiority by a significant improvement of 2% to the second-best model considered for the comparison. Further, we investigated the misclassified samples from the CIFAR-100 datasets (Fig. 7). It is apparent that capsule networks are prone to classification errors despite having parts to the whole learning mechanism. One of the possible reasons could be the larger extent of background pixels compared to foreground pixels (ROI), which could result in the indigent formation of the whole. In Table 7, we investigated the viewpoint invariance property of the proposed method on the SmallNORB dataset. It can be inferred that the proposed method has achieved comparable results, with Zhao et al. [30] demonstrating viewpoint invariance. We have also compared the runtime of the proposed method with Zhao et al. [30]. The proposed model takes 3.60 ms, 4.47 ms, 5.22 ms, and 6.21 ms per epoch on the Fashion MNIST, SVHN, CIFAR-10, and CIFAR-100 datasets, compared to 3.71 ms, 4.60 ms, 5.37 ms, and 6.53 ms of the proposed method. Though the proposed method is slightly more expensive than Zhao et al. [30], we defend the additional computation overhead considering the significant performance improvement.

As we discussed in the previous section, the benchmark method [15] uses a naive setup for forming the primary capsules and propagating the information downstream, which became a primary cause of performance degradation and the intricate information routing procedures on complex datasets. Many researchers adopted powerful CNN-based architectures as the backend to form more robust and informative capsules to tackle this. Adhering to this philosophy, we adopted the wide residual and SE blocks as the backend.

**Fig. 6** Impact of standard vs wider bottleneck residual block backbone on the proposed capsule network architecture.

**Fig. 7** Illustration of incorrectly classified samples of the proposed method on the CIFAR-100 dataset.

**Table 6** Performance comparison of the proposed method with other capsule network-based methods on the Brain Tumor classification dataset.

| Methods                  | Accuracy |
|--------------------------|----------|
| Sabour et al. [43]       | 80.90    |
| Phaye et al. [21]        | 82.77    |
| Zhao et al. [30]         | 90.10    |
| **Proposed**             | **92.30**|

The bold letter depicts the superior performing method against the other competing methods.

**Table 7** Performance comparison of the proposed method on the SmallNORB dataset.

| Methods      | Accuracy |
|--------------|----------|
| Sabour et al. [43] | 92.90 |
| Hinton et al. [25] | 92.10 |
| Zhao et al. [30]  | 93.60 |
| **Proposed**     | **93.45**|

The bold letter depicts the superior performing method against the other competing methods.
inspired by the recently introduced NF-Net [50] along with a modified FM routing algorithm, which yielded improvement with an accuracy of 95.35% on the CIFAR-10 dataset, which is an improvement of 5.64% over the benchmark model [15]. However, the wide residual–SE block was causing a significant increase in the trainable parameters. We adopted a wide bottleneck residual block to curtail this, which improved the accuracy to 95.71% from 95.35%, with an improvement of 0.57%.

4.4 Ablation study

Table 8 depicts the ablation study (architectural progression) toward the proposed WideCaps architecture. In the 1st two rows, we have listed the architectural details of Sabour [15] and Zhao [30], along with the performance achieved on the CIFAR-10 dataset. Unlike the methodology proposed by Zhao [30], we adopted Wide ResNet instead of ResNet v2 in WideCaps-v1 and achieved an accuracy of 93.20% with a significant improvement of 3.49%. In WideCaps-v2, we adopted a modified FM routing for the purpose discussed in Sect. 3.3. We achieved an accuracy of 94.66% with an improvement of 1.46% over [30], depicting the efficacy of the modified FM routing procedure. In WideCaps-v3, we embedded squeeze and excitation blocks and achieved a slight improvement of 0.20%. However, WideCaps-v3 resulted in a computation cost; to curtail this, we adopted a wide bottleneck residual connection with SE blocks in WideCaps-v4 and achieved an accuracy of 95.71% (+0.36% in comparison with WideCaps-v3). In WideCaps-v5, we adopted attention capsules to achieve superior performance, resulting in an improvement of 7.14% over [15], 3.65% over [30], and 1.14% over WideCaps-v4 with an accuracy of 96.85% on the CIFAR-10 dataset, and thus, we arrived at this configuration.

5 Conclusion

This study presents a novel capsule network-based architecture called WideCaps for efficiently dealing with complex images for image classification. The proposed model couples the capabilities of CNN and the capsules to achieve the defined objective. WideCaps uses a wide bottleneck residual connection with a squeeze and excitation attention block as the backbone network, followed by the attention capsules guided by the modified FM routing algorithm. The attention module expedited the flow of relevant features throughout the network by suppressing its counterpart. WideCaps achieved superior performance on CIFAR-10, Fashion MNIST, SVHN, and Brain Tumor with highly competitive performance on the CIFAR-100 dataset. Our future work includes investigating the performance of the WideCaps on more complex datasets such as ImageNet and the quantitative investigation of the equivariance property.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE 86(11), 2278–2324 (1998)
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Adv. Neural. Inf. Process. Syst. 25, 1097–1105 (2012)
3. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV) 115(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
4. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: integrated recognition, localization and detection using convolutional networks. In: 2nd international conference on learning representations, ICLR 2014; conference date: 14-04-2014 through 16-04-2014 (2014)
5. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9 (2015)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
7. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258 (2017)
8. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861
9. Huang, G., Liu, Z., Maaten, L. Van Der., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708 (2017)
10. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE Trans. Evol. Comput. 23(5), 828–841 (2019)
11. Moosavi-Dezfooli, S.-M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2574–2582 (2016)
12. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 427–436 (2015)
13. Liu, R., Lehman, J., Molino, P., Such, F. Petroski, Frank, E., Sergeev, A., Yosinski, J.: An intriguing failing of convolutional neural networks and the coodcoordsolution. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.) Advances in Neural Information Processing Systems, Vol. 31. Curran Associates, Inc (2017)
14. Hinton, G.E., Krizhevsky, A., Wang, S.D.: Transforming auto-encoders. In: International conference on artificial neural networks. Springer, pp. 44–51 (2011)
15. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett R. (Eds.), Advances in Neural Information Processing Systems, Vol. 30. Curran Associates, Inc (2017)
16. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747
17. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. NIPS Workshop on Deep Learn. Unsupervised Feature Learn. 2011 (2011)
18. Krizhevsky, A.: Learning multiple layers of features from tiny images. University of Toronto (2009)
19. Cheng, J., Yang, W., Huang, M., Huang, W., Jiang, J., Zhou, Y., Yang, R., Zhao, J., Feng, Y., Feng, Q., et al.: Retrieval of brain tumors by adaptive spatial pooling and fisher vector representation. PLoS ONE 11(6), e0157112 (2016)
20. Xiang, C., Zhang, L., Tang, Y., Zou, W., Xu, C.: Ms-capsnet: a novel multi-scale capsule network. IEEE Signal Process. Lett. 25(12), 1850–1854 (2018)
21. Phaye, S.S.R., Sikka, A., Dhall, A., Bathula, D.: Dense and diverse capsule networks: making the capsules learn better. arXiv:1805.04001
22. Hosgi, A., Wilcox, B., Gupta, Y., Rubin, D.L.: Self-attention capsule networks for object classification. arXiv:1904.12483
23. Jia, B., Huang, Q.: De-capsnet: a diverse enhanced capsule network with disperse dynamic routing. Appl. Sci. 10(3), 884 (2020)
24. Sun, G., Ding, S., Sun, T., Zhang, C., Du, W.: A novel dense capsule network based on dense capsule layers. Appl. Intell. 52, 3066–3076 (2021)
25. Hinton, G.E., Sabour, S., Frosst, N.: Matrix capsules with em routing. In: International conference on learning representations (2018)
26. LeCun, Y., Huang, F.J., Bottou, L.: Learning methods for generic object recognition with invariance to pose and lighting. In: Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004. CVPR 2004., Vol. 2, IEEE, pp. II–104 (2004)
27. Deliege, A., Cioppa, A., Droogenbroeck, M. Van.: Hitnet: a neural network with capsules embedded in a hit-or-miss layer, extended with hybrid data augmentation and ghost capsules. arXiv:1806.06519
28. Wang, D., Liu, Q.: An optimization view on dynamic routing between capsules
29. Fuchs, A., Pernkopf, F.: Wasserstein routed capsule networks. arXiv:2007.11465
30. Zhao, L., Wang, X., Huang, L.: An efficient agreement mechanism in capsnets by pairwise product. In: 24th European conference on artificial intelligence—ECAI 2020
31. Rezwan, I.M., Ahmed, M.B., Sourav, S., Quader, E., Hossain, A., Mohammed, N.: Mixcaps: capsules with iteration free routing. In: Digital image computing: techniques and applications (DICTA). IEEE 2020, 1–8 (2020)
32. Rajasegaran, J., Jayasundara, V., Jayasekara, S., Jayasekara, H., Seneviratne, S., Rodrigo, R.: Deepcaps: going deeper with capsule networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10725–10733 (2019)
33. Sun, K., Yuan, L., Xu, H., Wen, X.: Deep tensor capsule network. IEEE Access 8, 96920–96933 (2020)
34. Yang, S., Lee, F., Miao, R., Cai, J., Chen, L., Yao, W., Kotani, K., Chen, Q.: Rs-capsnet: an advanced capsule network. IEEE Access 8, 85907–85918 (2020)
35. Pawan, S., Sankar, R., Prabhudev, A.M., Mahesh, P., Prakashini, K., Das, S.K., Rajan, J.: Mobilecaps: a lightweight model for screening and severity analysis of covid-19 chest x-ray images. arXiv:2108.08775
36. Li, H., Xiong, P., An, J., Wang, L.: Pyramid attention network for semantic segmentation. arXiv:1805.10180
37. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3146–3154 (2019)
38. Huang, Q., Xia, C., Wu, C., Li, S., Wang, Y., Song, Y., Kuo, C.-C.J.: Semantic segmentation with reverse attention. arXiv:1707.06626
39. Dong, X., Lei, Y., Tian, S., Wang, T., Patel, P., Curran, W.J., Jani, A.B., Liu, T., Yang, X.: Synthetic mri-aided multi-organ segmentation on male pelvic ct using cycle consistent deep attention network. Radiother. Oncol. 141, 192–199 (2019)
40. Choi, J., Seo, H., Im, S., Kang, M.: Attention routing between capsules. In: Proceedings of the IEEE/CVF international conference on computer vision workshops, pp. 0–0 (2019)
41. Huang, W., Zhou, F.: Da-capsnet: dual attention mechanism capsule network. Sci. Rep. 10(1), 1–13 (2020)
42. Mazzia, V., Salvetti, F., Chiarberge, M.: Efficient-capsnet: capsule network with self-attention routing. Sci. Rep. 11(1), 1–13 (2021)
43. Tsai, Y.-H.H., Srivastava, N., Goh, H., Salakhutdinov, R.: Capsules with inverted dot-product attention routing. In: International conference on learning representations (ICLR) (2020)
44. Ahmed, K., Torresani, L.: Star-caps: capsule networks with attentions and graph convolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
45. Pawan, S., Rajan, J.: Capsule networks for image classification: a review. Neurocomputing 509, 102–120 (2022)
46. LaLonde, R., Xu, Z., Irmakci, I., Jain, S., Bagci, U.: Capsules for biomedical image segmentation. Med. Image Anal. 68, 101889 (2021)
47. Pawan, S., Sankar, R., Jain, A., Jain, M., Darshan, D., Anoop, B., Kothari, A.R., Venkatesan, M., Rajan, J.: Capsule network-based architectures for the segmentation of sub-retinal serous fluid in optical coherence tomography images of central serous choroidopathy. Med. Biol. Eng. Comput. 59(6), 1245–1259 (2021)

48. Zeng, T., So, H.K.-H., Lam, E.Y.: Redcap: residual encoder-decoder capsule network for holographic image reconstruction. Opt. Express 28(4), 4876–4887 (2020)

49. Jaiswal, A., AbdAlmageed, W., Wu, Y., Natarajan, P.: CapsuleGAN: generative adversarial capsule network. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, pp. 0–0 (2018)

50. Brock, A., De, S., Smith, S.L.: Characterizing signal propagation to close the performance gap in unnormalized resnets. In: 9th International Conference on Learning Representations, ICLR (2021)

51. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proceedings icml, Vol. 30, p. 3 (2013)

52. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, PMLR, pp. 448–456 (2015)

53. Jamel, T.M., Khammas, B.M.: Implementation of a sigmoid activation function for neural network using fpga. In: 13th Scientific Conference of Al-Ma’moon University College, Vol. 13 (2012)

54. LeCun, Y., Cortes, C., Burges, C.J.: The mnist database. http://yann.lecun.com/exdb/mnist

55. Chollet, F. et al.: Keras. https://keras.io (2015)

56. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: large-scale machine learning on heterogeneous systems. software available from tensorflow.org (2015). https://www.tensorflow.org/

57. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)

58. Bruch, S., Wang, X., Bendersky, M., Najork, M.: An analysis of the softmax cross entropy loss for learning-to-rank with binary relevance. In: Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, pp. 75–78 (2019)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

S.J. Pawan obtained his B.E and M.Tech degrees from VTU, Belgaum, India. He is currently working as a Senior Research Fellow and pursuing his Ph.D. in the Dept. of Computer Science and Engineering at the National Institute of Technology Karnataka (NITK), India. His research interest focuses on machine learning with deep neural networks for medical image processing, magnetic resonance imaging, and neuroscience.

Rishi Sharma obtained his B.Tech degree from National Institute of Technology Karnataka (NITK), India and currently works as a Software Engineer at Palo Alto Networks in Bangalore, India. His research interests lie at the intersection of computer science and artificial intelligence, with a focus on image processing, machine learning with deep artificial neural networks, and natural language processing.

Hemanth Reddy received his B.Tech degree from National Institute of Technology Karnataka (NITK), India in 2021. He is currently working as a Software Engineer at SAP Labs India. His primary research interests are in the fields of deep learning and image processing.

M. Vani has M.Tech degree in Computer Science and Engg from NITK, Surathkal. She is currently an Associate Professor at National Institute of Technology Karnataka, working in the department of Computer Science and Engineering. She has experience in computer science with emphasis on computer algorithms and machine learning.
Jeny Rajan received his Ph.D. from the University of Antwerp (Vision Lab), Belgium, in 2012. Currently, he is working as an Asst. Professor at the Dept. of Computer Science and Engineering at the National Institute of Technology Karnataka (NITK), India. His research interests include image processing, deep learning, and medical image analysis. He has published more than 75 research papers in reputed international journals and conference proceedings.