Analog Integrated Circuits and Signal Processing (2023) 114:315–324
https://doi.org/10.1007/s10470-022-02108-w

Deep and shallow fast embedded capsule networks: going faster with capsules

Mohammed Abo-Zahhad[1,2] · Islam Eldifrawi[1] · Moataz Abdelwahab[1] · Ahmed H. Abd El-Malek[1]

Received: 4 December 2021 / Revised: 4 December 2021 / Accepted: 28 September 2022 / Published online: 2 November 2022
© The Author(s) 2022

Abstract

Capsule Networks (CapsNets) is a great approach for understanding data in the field of computer vision. CapsNets allow a deeper understanding of images compared to the traditional Convolutional Neural Networks. The first test for CapsNet was in digits recognition on the 'MNIST' dataset, where it successfully achieved high accuracy. CapsNets are reliable at deciphering overlapping digits. Deep Capsule Networks achieved state-of-the-art accuracy in CIFAR10 which isn't achieved by shallow capsule networks. Despite all these accomplishments, Deep Capsule Networks are very slow due to the 'Dynamic Routing' algorithm. In this paper, Fast Embedded Capsule Network and Deep Fast Embedded Capsule Network are introduced, representing novel capsule network architectures that uses 1D convolution based dynamic routing with a fast element-wise multiplication transformation process. These architectures not only compete with the state-of-the-art methods in terms of accuracy in the capsule domain, but also excels in terms of speed, and reduced complexity. This is shown by the 58% reduction in the number of trainable parameters and 64% reduction in the average epoch time in the training process. Experimental results shows excellent and verified properties.

Keywords 1D convolutional kernels · CapsNets · Fashion MNIST · CIFAR10 · Facial expressions recognition · MNIST · CK+

1 Introduction

Many computer vision tasks had their feature engineering done by various feature extraction methods such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and Scale Invariant Feature Transform (SIFT) [1]. After the introduction of CNNs in many applications, accuracy was improved [2, 3]. The performance of CNNs is enhanced by using as much training data as possible, or by increasing the depth and the width of the network (e.g., the number of levels of the network and the number of units at each level).

Despite all the accomplishments of CNNs, they lack the ability to understand spatial relationship between features and there is spatial invariance caused by pooling. Capsule Networks (CapsNets) don't have these limitations. MNIST digit recognition dataset [4] was the first dataset used for testing CapsNets [5]. CapsNets provided high accuracy and was used in other applications like Facial Expressions Recognition Capsules (FERCaps) [6] for emotion detection and many others. Despite the success and high accuracy observed for such datasets, yet CapsNets' true potential was not fully realized, providing sub-par performance on several complex datasets like CIFAR10. DeepCaps [7] achieved state-of-the-art performance when used with CIFAR10, but still DeepCaps and CapsNets are slow in comparison with CNNs due to the clustering and routing algorithms. To increase the speed, new simpler architectures with reduced complexity and simpler algorithms are needed.

✉ Islam Eldifrawi
islam.eldifrawi@ejust.edu.eg

Mohammed Abo-Zahhad
mohammed.zahhad@ejust.edu.eg

Moataz Abdelwahab
moataz.abdelwahab@ejust.edu.eg

Ahmed H. Abd El-Malek
ahmed.abdelmalek@ejust.edu.eg

[1] School of Electronics, Communications & Computer Engineering, Egypt-Japan University of Science & Technology (E-JUST), New Borg El Arab City, Alexandria 21934, Egypt

[2] Electrical Engineering Department, Faculty of Engineering, Assiut University, Assiut 71515, Egypt

In this paper, new architectures Fast Embedded Capsule Network (FECapsNet) and Deep Fast Embedded Capsule Network (Deep-FECapsNet) are proposed to make CapsNets and DeepCaps faster while maintaining accuracy. Both architectures rely on two pillars; which are a new 1D convolution based dynamic routing that reduces the complexity and the number of capsules inputted in the routing process. In addition, a new transformation process using element-wise multiplication for faster processing to predict parent capsules from the children capsules is adopted. Experimental results shows remarkable reduction in the number of trainable parameters, as well as a high speed gain while persevering state-of-the-art accuracy in the capsule domain.

2 Related work

In this section, 1D convolutions in CNNs are mentioned as they will be incorporated later with a new functionality in the capsule domain in the new routing algorithm. In addition, CapsNets are briefly mentioned as they are the basic block of all the baseline architectures mentioned. FERCaps [6] are then mentioned as an application on CapsNets. ResNets [8] are also introduced because when they are combined with CapsNets they represent the building concept of DeepCaps that is later compared to the proposed architecture. DeepCaps are also introduced as they will represent the benchmark and the evaluation criteria for the performance of the proposed method. Enhanced Capsule Network (DE-CapsNet) [9] is introduced as it's a competing architecture to DeepCaps and it will be compared with the proposed method.

2.1 1D convolutions in CNNs

In many architectures, 1D convolutions are used for features extraction in 1D signals. As shown in Fig. 1, one 1D convolution kernel can compress a 3D feature map to 2D. To produce a 3D feature map, the depth of the resulting feature map is controlled by the number of 1D filters. 1D convolutions usage with CapsNets as in [10] and [11] was limited to feature extraction, as the input data type in these papers was one dimensional data e.g speech, text and ECG signals.

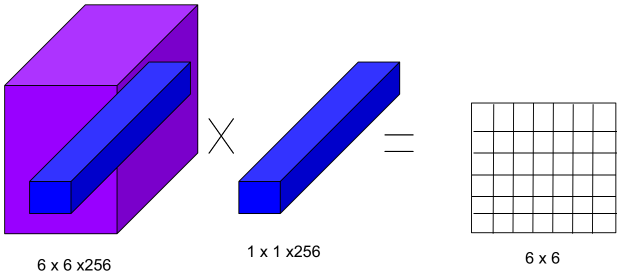

Fig. 1 Example on 1D convolutional kernels used in the convolution of a 3D feature map

In this paper 1D convolutions aren't used for feature extraction, but they are rather used in the routing process to give a compressed embedded representation for the relationships between the features of an object forming the capsules.

2.2 Brief on ResNets, CapsNets, FERCaps, DeepCaps and DE-CapsNet "baseline architectures":

CNNs and deep learning, where features are automatically extracted and the patterns are learned from the data, has become the next evolutionary step and many breakthroughs have been achieved in various tasks. Experts aren't needed for designing the kernels to extract features, but a large amount of data is needed as well as deep architectures for complex features extraction. Despite the success of CNNs, CNNs don't capture the spatial relationships between different features, causing them to be susceptible to adversarial attacks and to the incapability to detect misplacement of the features or deformations.

CapsNets [5] came as a nearer step for the simulation of human vision, and to solve this deficiencies of CNN by keeping the spatial relationship between the features and dealing with the features as vectors instead of scalars. The dynamic routing in CapsNets is used to route the data to the correct capsules in a similar way to how human vision happens. The network has two convolutional layers and one fully connected classifier as shown in Fig. 2. CapsNets are resistant to adverserial attacks and can detect misplacement of features and they work well with simple datasets like MNIST. If CapsNets are used with complex datasets like CIFAR10, they don't compete with CNNs in terms of accuracy as they aren't deep enough.

Facial expressions Recognition Capsules (FERCaps) represent a modified CapsNets architecture used to classify basic facial expressions. Facial expressions play an important role in the recognition of emotions. The focus is on the seven basic emotional states mentioned in the CK+ dataset, which are: neutral, happiness, sadness, surprise, disgust, fear, and anger [15]. The baseline architecture is

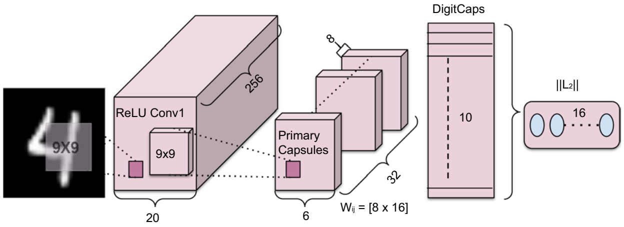

Fig. 2 CapsNets architecture of the system for MNIST [5]

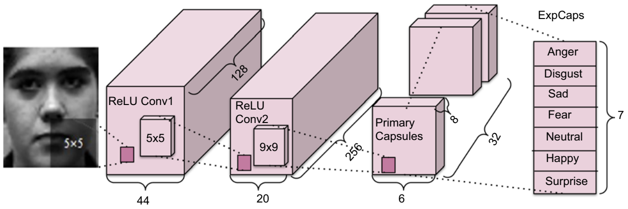

Fig. 3 FERCaps architecture [6]

shown in Fig. 3 and the classification layer is for emotion detection “Emotion caps”.

Using the concept of residual learning [8] that mainly solve the issue of vanishing gradients in deep neural networks using skip connections, DeepCaps [7] then appeared. In its core, DeepCaps is a combination of residual learning with CapsNets and dynamic routing based on 3D convolution, to produced a deep architecture that achieve state-of-the-art accuracy for CIFAR10, but the number of parameters is very high and the speed is low.

DE-CapsNet appeared to represent a simpler architecture based on residual learning and disperse dynamic routing with using small kernel sizes and also a deep network but not as deep as DeepCaps. DE-CapsNets achieved higher accuracy with higher speed and less number of parameters but they are still slow in comparison with CNNs due to the routing/clustering algorithms.

The need for a fast dynamic routing algorithm that allow the usage of deep architectures within an acceptable limit for the number of trainable parameters should be thought of as the next step.

3 The proposed deep fast embedded CapsNet (Deep FECapsNet) and its shallow version fast embedded CapsNet (FECapsNet)

In the beginning of this section, Deep FECapsNet architecture is introduced, then the shallow FECapsNet is introduced later. Deep FECapsNet target is to reduce the complexity and increase the speed of DeepCaps while preserving the depth of the architecture, as well as achieving state-of-the-art accuracy. As shown in Fig. 4 there are four residual blocks composed of different convolution capsule layers. Each block or capsule cell has three 3x3 convolutional kernel layers cascaded by a 1x1 convolutional kernel layer. Feature maps are divided into groups with constant depth. All these convolutional capsule layers have their number of routing iterations set to one. There are three new blocks ’highlights’ detailed as follows:

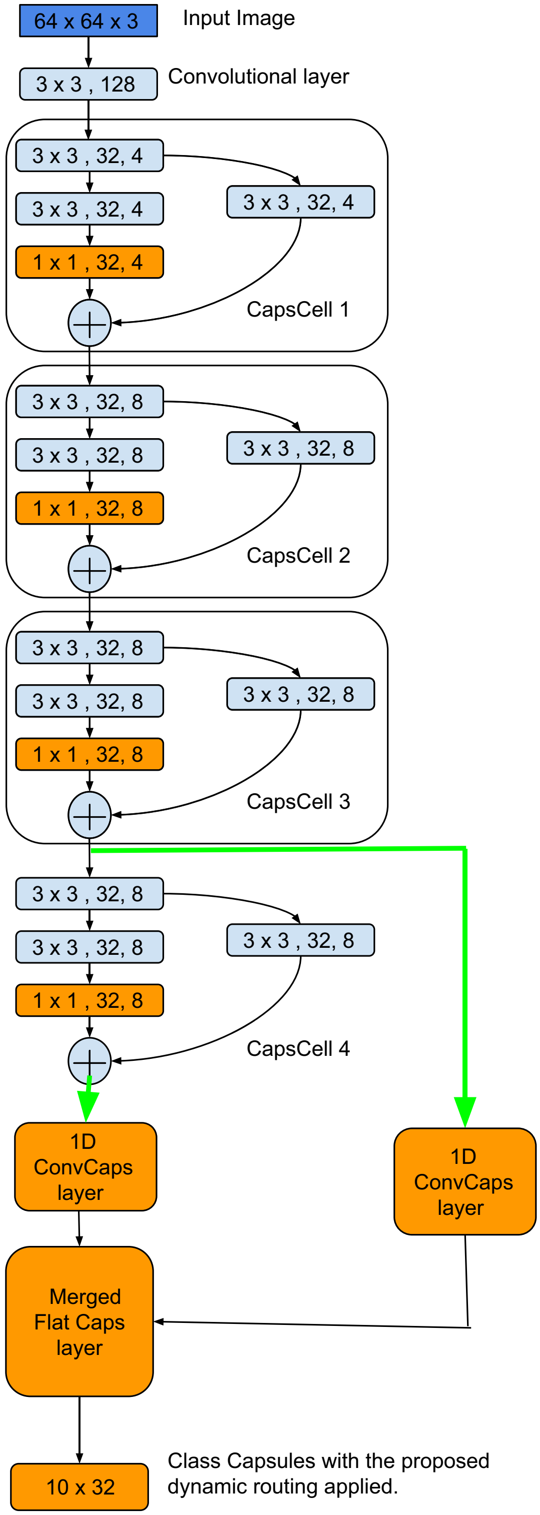

Fig. 4 The proposed deep-FEcapsNet architecture. the new novel blocks are colored in orange

1) The first block is the 1D-ConvCaps layer that has been introduced where capsule vectors give a compressed and embedded representation of the relationship between

these features using 1d-convolution kernels rather than having the features stacked on each other. The 1D-convolution kernels act similar to a fast encoder neural network.

2) The second block is an implicit one, which is the new dynamic routing using a new transformation process to map the children capsules to their parent capsules.
3) The third block is the Merged FlatCaps Layer that merges the newly formed capsules in a 2D-matrix, starting with the number of capsules as its first dimension, and the depth or dimension of the capsules as the second dimension as shown in Fig. 3. This layer is only seen in deep architectures that use residual learning, as this layer mainly combine the outputs of the skip connections forming the needed capsules, hence it won't be added with shallow FECapsNet as there are no skip connections or multiple outputs that need to be combined.

Full details of 1D-ConvCaps and the new dynamic routing process are provided in the following subsections.

3.1 1D-ConvCaps layer

The proposed 1D-ConvCaps layer is placed before any capsule layer that has more than one routing iteration as shown in the architecture in Fig. 4. This layer takes its input in the form of 4D matrices. As shown in Fig. 5, the input feature maps are divided into 32 groups and each group has a depth of 8. The first step is that all the 32 groups are merged into one with depth 32 × 8. So, the feature map changes into 3D instead of 4D, and it has all the high level features of objects in the image stacked over each other. Each pixel in the new feature map is a 256D vector. 1D-convolution kernels are also 256D vectors and each pixel vector in the feature map will undergo a 1D convolution process, which is in its essence a dot product process with these kernels, producing a scalar number that is an encoding for the relationship between the stacked features. 32 kernels are used to produce 32 scalars for each 256D-pixel vector in the feature map.

These 32 scalars are then stacked into a vector representing the new capsule. Before the 1D-ConvCaps layer the 256D-pixel vector in the feature map was divided into 32 capsules each is of 8 dimensions, but after the addition of 1D-ConvCaps, the 256D-pixel vector is transformed into one 32D capsule. It should be noted that '32' is the same number of dimensions of the class capsules and this will be the constant number of dimensions for all the capsules. The intuition behind making the dimension of the capsules constant came from the fact that both the children capsules "the part" and the parent capsules "the whole" lie within the same 3D space we live in, so they should have the same number of dimensions describing their pose.

3.2 The proposed dynamic routing using 1D convolution and element-wise multiplication

Let 'r' be the number of routing iterations, 'l' be the order of the layer, '$K_{\mathrm{j|i}}$' be the output predictions vectors from the element-wise multiplication between the 1D-ConvCaps output capsule vectors 'u_{i}' and a randomly initialized weight matrix W_{ij}. W_{ij} has three dimensions which are: the number of input capsules, the number of output capsules, and the constant number of the dimensions of the capsules, while 'u_{i}' has only two dimensions which are: the number of input capsules, the constant number of dimensions of the capsules. 'u_{i}' will have an expanded dimension as we will repeat this 2D vector until it becomes a 3D vector with the same dimensions as W_{ij}. Let the new dimension-expanded vector be u_{k}. The new transformation process now is just a simple element-wise multiplication described as:

$$K_{\mathrm{j|i}} = u_{\mathrm{k}} \times W_{\mathrm{ij}} \tag{1}$$

Proposed Dynamic Routing Algorithm:

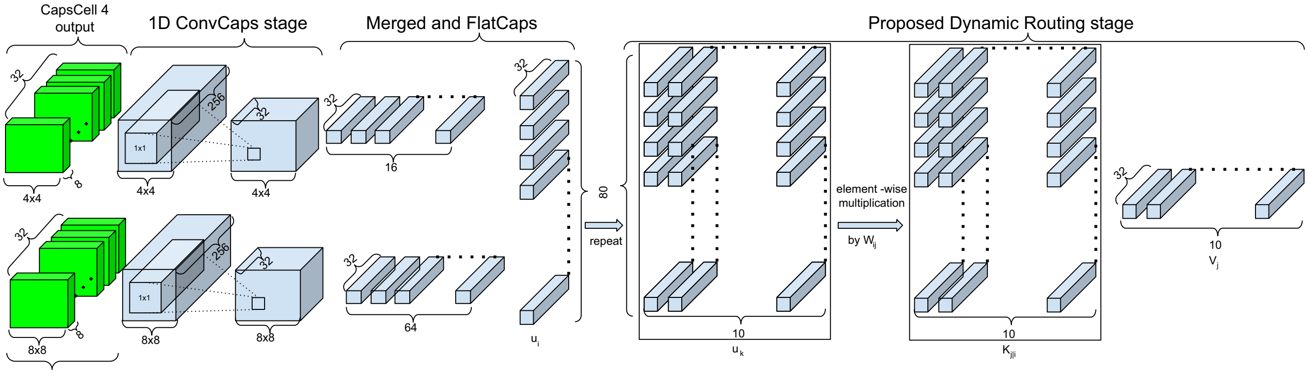

Fig. 5 Capsule formation inside the 1D-ConvCaps and the new Dynamic routing. The input here comes from ConvCaps cells 3 and 4 and it's shown by the green arrow in Fig. 4

```
$u_k \leftarrow$ repeat($u_i$, output capsules number)
$K_{j|i} \leftarrow u_k \times W_{ij}$ refer to equation(1)
For all capsule i in layer $l$ and capsule j in layer $l$+1:
    $b_{ij} \leftarrow 0$
for $r$ iterations do:
    for all capsule $i$ in layer $l$: $c_{ij} \leftarrow$ softmax($b_{ij}$)
    for all capsule $j$ in layer $(l + 1)$: $s_j \leftarrow \Sigma_i\, c_{ij}\, K_{j|i}$
    for all capsule $j$ in layer $(l + 1)$: $V_j \leftarrow$ squash( $s_j$)
    for all capsule i in layer l and capsule j in layer (l + 1):
  $b_{ij} \leftarrow b_{ij} + K_{j|i}.\ V_j$
return $V_j$
```

It should be noted that the dimensions of b_{ij}, and c_{ij} have been reduced along with W_{ij}. This resulted in having 25,920 parameters in the class capsule layer in DeepFECapsNet instead of 6,553,920 parameters in the Deepcaps baseline architecture [7].

To give a high level explanation of how capsules are formed by the orange blocks shown in Fig. 4, The new dynamic routing process and Fig. 5 should be viewed concurrently. As shown in Fig. 5 both capsule tensors from CapsCell 3 and CapsCell 4 will be transformed into a capsule vector by the 1D-ConvCaps layer and will be merged and flattened to get u_i. Then u_i is repeated 10 times as the next layer has 10 capsules "10 classes". This is because we expect each capsule in u_i to give 10 predictions for the 10 parent capsules. By using element-wise multiplication with the transformation matrix W_{ij}, the predictions vectors matrix '$K_{j|i}$' is produced. $K_{j|i}$ represents the predictions of each capsule of u_i capsules to the parent capsules. In the first routing iteration, all prediction vectors are equally weighted and summed together to get the final predictions V_j. Then, in the following iterations, coupling coefficients are updated according to the agreement with V_j and $K_{j|i}$.

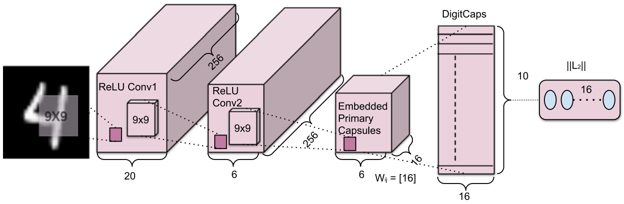

Fig. 6 Feature maps shapes of the proposed architecture FECapsNets for MNIST

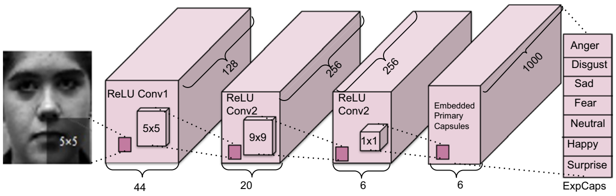

Fig. 7 Feature maps shapes of the proposed architecture 'FECapsNet' for FER

3.3 The proposed fast embedded CapsNet(FECapsNet) applied on MNIST

A new architecture is proposed to reduce complexity and increase the speed of CapsNets. This architecture is based on adding the 1D-ConvCaps layer representing the primary capsule layer and using the new dynamic routing instead of the one proposed by Hinton et al. The details of the feature maps of the new architecture are shown in Fig. 6. Before the primary capsule layer, feature maps are not divided the to 32 groups as proposed by Hinton et al; instead the whole 6 × 6 × 256 feature map is passed to the 1D-ConvCaps layer producing a new feature map that is 6 × 6 × 16. 16 dimensions/kernels were initially chosen as Hinton et al. mentioned that these dimensions are enough to capture the pose of any object. The transformation matrix responsible for the part-whole relationship, will be of dimensions 36 × 10 × 16 as here the part "children capsules layer" and the whole "parent capsules layer" both contain 16D vectors. Since both the part and the whole vectors are in our 3D space, then intuitively both should have the same number of dimensions describing their pose. During the dynamic routing process, the primary capsule vectors will undergo the new dynamic routing proposed in this paper.

In the original CapsNet architecture used on MNIST, the feature map depth was 256, so 1152 primary capsules were taken from it after they were divided into 32 8D groups. After using the 1D-ConvCaps layer, the feature map depth will be reduced from 256 to 16, so no division into groups is needed and the number of primary capsules will become 36 instead of 1152. This means that the transformation matrix dimensions is changed from 1152 × 8 × 10 × 16 in the original CapsNets to 36 × 10 × 16 in FECapsNets.

3.4 The proposed fast embedded CapsNet(FECapsNet) applied on CK+

By modifying the FERCaps architecture through adding the 1D-ConvCaps layer and applying the new dynamic routing algorithm, the following feature maps shown in Fig. 7 emerge. FECapsNet for FER has 3 convolutional layers. Conv1 has 128 5 × 5 convolution kernels with a stride of 1. Conv2 has 256 5 × 5 convolution kernels with a stride of 2. Then there are 256 9 × 9 kernels with a stride of 2, then 1000 1 × 1 embedding kernels, and the classification layer, that has the 7 basic emotion classes. Detailed explanation for choosing 1000 1D kernels will be provided in the later sections.

4 Experimental results and evaluation

Six experiments were carried out on CIFAR10, F-MNIST, MNIST and CK+, which are different datasets from complexity's point of view. The results were compared with various state-of-the-art methods in the capsule domain. The following subsections describe the system, the datasets, the experiments and the results in detail.

4.1 The system

The experiments are run on the "Google Colab" system with the following specifications:

Python Tensor-flow 1.15.2 is used with keras. The RAM is approximately 12.6 GB. The GPU has 2496 CUDA cores, 12GB GDDR5 VRAM. The CPU has one single core hyper threaded Xeon Processors @2.3Ghz i.e. (1 core, 2 threads).

4.2 The datasets

The Canadian Institute For Advanced Research (CIFAR10) [12]: The dataset consists of 60,000 32x32 colour images. There are 50,000 training images and 10,000 test images. There are 10 classes, with 6000 images per class. The classes are: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

Fashion-Modified National Institue of Standards and Technology database (F-MNIST) [13]: The dataset consists of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. The classes are: T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot.

The Modified National Institute of Standards and Technology database (MNIST) [14]: this dataset contains 80,000 images of the ten digits. 60,000 images are used for training and the rest is divided between validation and test samples equally.

The Extended Cohn-Kanade Dataset (CK+) [15]: It is for facial expressions recognition is also used for the purpose of generalization of the concept of using embedded capsules across different datasets. Around 800 training images for 7 classes are used while another 200 images are split equally between validation and testing.

4.3 The performance metrics

Three performance metrics are used to compare different architectures with the proposed ones, these metrics are the speed, accuracy, and complexity.

The speed is represented by the average epoch time in seconds during training for different architectures. By taking into account the effect of random initialization, 25 different runs of each experiment are performed, and the average epoch time was obtained from all epochs of all 25 runs.

The accuracy defines how well the model extracted the suitable features.

The complexity is represented by the number of trainable parameters and these parameters can be calculated in two different stages. The first stage is when normal CNN layers are used and the formula is:

$$N_{\text{conv}} = N_{\text{ker}} \times S_{\text{ker}} + N_{\text{ker}} \tag{2}$$

Where N_{conv} is the number of parameters introduced by the convolutional layer, N_{ker} is the number of kernels, and S_{ker} is the size of the kernel. The second stage is for the dynamic routing number of parameters and they can be calculated using this formula:

$$N_{\text{rout}} = N_{\text{caps}} \times N_{\text{c}} \times n + D_{\text{trns}} \tag{3}$$

Where N_{rout} is the number of parameters in the dynamic routing stage, N_{caps} is the number of children capsule being routed, D_{trns} represent the dimensions of transformation matrix, N_{c} is the number of target classes, and n represents the number of iterations of the routing algorithm (number of coupling coefficients).

4.4 The experiments and results

Experiment 1. Applying Deep-FEcapsNets architecture on CIFAR10 dataset:

In this Experiment the Deep-FEcapsNet is applied on CIFAR10 which is considered as a complex dataset. As shown in Table 1, Deep FE-CapsNet surpassed all the state-of-the-art methods in the capsule domain in terms of reduced complexity by a 50% reduction in the number of trainable parameters compared to the next best method which is DE-CapsNet [9]. The proposed method also has the least average epoch time during training indicating that it the fastest. As per Table 1 the accuracy of Deep-FEcapsNet comes in the second place after DE-CapsNet by 0.21% which is a small margin when compared to the doubled speed gain and the reduction in complexity.

Experiment 2. Applying Deep-FEcapsNets architecture on F-MNIST dataset:

In this Experiment the Deep-FEcapsNet is applied on F-MNIST which is considered as a simple dataset with plenty of images per class. As shown in Table 2, Deep FE-CapsNet surpassed all the state-of-the-art methods in the capsule domain in terms of reduced complexity, as it has less than half the number of trainable parameters of the next

Table 1 Comparing state-of-the-art methods for CIFAR10 dataset in the Capsule domain with the proposed methods with respect to accuracy, speed and complexity

Method/metric	Accuracy (%)	Average epoch time (secs)	Number of trainable parameters
CapsNets [5]	89.4	2700	22.4
DeepCaps [7]	91.01	492	13.43
DeepCaps (7 ensembles) [7]	92.74	3144	93.99
DE-CapsNet [9]	92.96	382	11.2
Deep FE-CapsNet (proposed)	92.75	178	5.76

Table 2 Comparing state-of-the-art methods for F-MNIST dataset in the Capsule domain with the proposed methods with respect to accuracy, speed and complexity

Method/metric	Accuracy (%)	Average epoch time (secs)	Number of trainable parameters
CapsNets [5]	93.9	2700	22.4
DeepCaps [7]	94.46	492	13.43
DeepCaps (7 ensembles) [7]	94.73	3144	93.99
DE-CapsNet [9]	93.64	382	11.2
Deep FE-CapsNet (proposed)	94.69	178	5.76

Table 3 Comparing the performance of CapsNet and FECapsNet on the digits in MNIST dataset

Evaluation metrics	CapsNet	FECapsNet (proposed)
Primary/digit capsules Dimensions	8/16	16/16
Average epoch time (secs)	2700	35
Number of dynamic routing parameters	1,509,120	6840
Average number of epochs till convergence	6	7
Accuracy	99.75%	99.36%

Table 4 Experimentation results of changing the dimensionality of capsules in FECapsNet to capture the target patterns of the digits in MNIST dataset

Evaluation metrics/capsule dimensions	8D	12D	16D	20D
Average epoch time (secs)	21	23	35	40
Dynamic routing parameters number	3960	5400	6840	8640
Average number of epochs till convergence	23	10	7	20

best method which is DE-CapsNet [9]. The proposed method also has the least average epoch time during training indicating that it has the highest speed among all the the capsule domain state-of-the-art methods. Regarding the accuracy, as shown in Table 2, Deep-FEcapsNet comes in the second place after DeepCaps (7 ensembles) by 0.04%.

In both experments, the accuracy metrics were referenced from [9], while the number of trainable parameters are calculated from equation (2) and equation (3) and are also given by the code.

Experiment 3. Applying FECapsNets architecture on MNIST dataset:

The experiment compares the performance metrics of the conventional CapsNet on the MNIST dataset with that of the proposed FECapsNet. Table 3 shows that FECapsNet is superior in terms of speed as there is around 98% speed gain and 99% reduction in the number of ***dynamic*** routing parameters in comparison with CapsNet. The proposed shallow FECapsNet reached 99.36% in accuracy while CapsNet reached 99.75%, so the proposed architecture almost provide preserved accuracy.

Experiment 4. Changing the depth dimension of both the primary and class capsules and studying the effect and performance metrics on MNIST:

In this experiment, the objective is to study the effect of changing the embedded capsules' dimension on the model's performance on a reasonably simple task like digit recognition with the MNIST dataset. The primary caps and Digit caps dimensions are changed starting from 8D till 20D vectors in four steps. At each dimension, 25 runs are performed to get statistically significant results. Table 4 shows the effect of this change on performance metrics.

It is observed that when increasing the number of dimensions of the object, the epoch's average time increases, but the total time for training decreases. This is expected as the dimensions of the vectors that describe the objects are increased, so complexity increases, and the time taken in one epoch increases. On the other hand, the model is now big enough to be trained on and to grasp the image's complex patterns in less training time and less number of training epochs. The conclusion here is that the number of dimensions of the capsules that describes objects, is a hyper-parameter that needs tuning, and that depends on the complexity of the patterns to be detected.

Experiment 5. Applying FECapsNet on CK+ FERdatabase:

This experiment investigates the performance of FECapsNets on the CK+ dataset for facial expressions recognition, to observe the effect of applying the FECapsNets on more complex patterns like facial expressions rather than simple digits. Faces and facial expressions are known for their complexity, so choosing a dataset like CK+ that has less training images per emotion/class will be an excellent example of a challenging dataset. The expectation here is that if the number of primary capsules reduces, then their dimensions must increase so that the complex face patterns can be described in a smaller number of capsules. Thirty six 1000D-capsules were chosen to be able to capture the complex facial expressions patterns. The average epoch time was reduced from 20.3 in FERCaps to 3.5 secs in FECapsNet as seen in Table 5, which is good but not as huge as the speed gain observed in the MNIST dataset. This is because CK+ has a much smaller training dataset than MNIST, but still FECapsNet is faster and has less dynamic routing parameters than FERCaps.

Experiment 6. Changing the depth dimension of both the primary and class capsules and studying the effect and performance metrics on CK+:

In this experiment, the range of capsules dimensions is large “in thousands” as the training samples are few per each class and the facial features are relatively complex.

When using 16D primary capsules in FECapsNets for FER on CK+, 50% validation accuracy is never reached. The training accuracy never passes 40%, which indicates that the model is too simple to catch the patterns needed in the dataset. That did not happen when FECapsNets was used on MNIST dataset as 16 dimensions were enough to describe digits, but 16 dimensions are not enough to detect various facial expressions in CK+ dataset. By inspecting Table 6, it is observed that when increasing the dimensions of the primary capsule vectors to 1000, the model starts to converge at 98% validation accuracy after 61 epochs and takes less number of epochs “21” when using 2000-dimensional capsules. When increasing the dimensions more than 2000, the number of epochs needed to reach 98% accuracy on validation increases, suggesting that the model is overshooting the global optima and needs more epochs to reach it as the learning rate decreases as ‘ADAM’ optimization algorithm is used in the model. This is a sign of over-fitting or that the model is becoming more complex than needed.

The first two experiments were run on keras as it is easier for handling deep and complex architectures, but slower than using tensor-flow directly. The next four experiments are run on tensor-flow directly as shallow architectures are used, so the time taken in the epochs for the shallow architectures is less than that of the deep architectures.

5 Conclusion

In this paper two new architectures ’FECapsNet’ and ’Deep-FEcapsNets’ are proposed. Inspiration was drawn from residual learning, Inception models and DeepCaps. 1D-convCaps layer where one 1D-convolution is used to form capsules from an embedded representation of relationships between the features rather than using the features directly, and a new simpler transformation process in the dynamic routing were introduced, producing more than 50% reduction in the number of parameters of the deep models in comparison with capsule domain state-of-the-art

Table 5 Comparing the performance of FERCaps and FECapsNet on the digits in CK+ dataset

Evaluation metrics	FERCaps	FECapsNet (proposed)
Primary/emotion capsules Dimensions	8/16	1000/1000
Average epoch time (secs)	20.3	3.5
Number of dynamic routing parameters	1,056,384	252,756
Average number of epochs till convergence	18	61
Accuracy	98.18%	98%

Table 6 Experimentation results of changing the dimensionality of capsules in FECapsNet to capture the target patterns of the digits in CK+ dataset

Evaluation metrics/capsule dimensions	1000D	2000D	3000D	4000D
Average epoch time (secs)	3.5	4.8	5.1	6.12
Dynamic routing parameters number (millions)	0.253	0.501	0.753	1.008
Average number of epochs till convergence	61	21	29	38

deep models, and more than 98% reduction for the shallow models that heavily depends on the dynamic routing layer.

There are two methods that can enhance the performance of FECapsNet when working on complex datasets like CIFAR10 and CK+. The two methods are to simply increase the number of dimensions of the capsules or to use deeper architectures to detect the needed complex features in deep layers. Experimental results tested on the CIFAR10, MNIST, F-MNIST and CK+ datasets confirm higher speeds and reduced complexity compared to CapsNet, FERCaps, DeepCaps and DE-CapsNet, while preserving state-of-the-art accuracy.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Kortli, Y., Jridi, M., Al Falou, A., & Atri, M. (Apr. 2018). "A comparative study of CFs, LBP, HOG, SIFT, SURF, and BRIEF techniques for face recognition," In *Proceeding of SPIE* (vol. 10649, Art. no. 106490M).
2. Mollahosseini, A., Chan, D., & Mahoor, M. H. (Mar. 2016). "Going deeper in facial expression recognition using deep neural networks," In *Proceeding of IEEE winter conference on applications of computer vision(WACV2016),* Lake Placid, NY, USA, 7-10, (pp. 1–10).
3. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM, 60*(6), 84–90.
4. LeCun, Y., Cortes, C., & Burges, C. "The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/.[Last Accessed, April 2021].
5. Sabour, S., Frosst, N., & Hinton, G. E. (2017). "Dynamic routing between capsules," In *Proceeding of neural information processing systems (NIPS)* Long Beach, CA, USA, pp. 1-11.
6. HU, QI-DI (Dec. 2019) "FERCaps: A capsule-based method for face expression recognition from frontal face images," In *Proceeding of Int. Conf. Power Energy Environmental Material Sci.(PEEMS2019)*, Sanya, China, 22-23, pp. 1-6.
7. Rajasegaran, J., Jayasundara, V., Jayasekara, S., Jayasekara, H., Seneviratne, S., & Rodrigo, R. (2019). "DeepCaps: Going Deeper With Capsule Networks", In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition 2019*, Long Beach, CA, USA, pp. 10717-10725.[Online]. https://doi.org/10.1109/2019.01098
8. He, K., Zhang, X., Ren, S., & Sun, J. (2016). "Deep Residual Learning for Image Recognition", In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) 2016*, Las Vegas, NV, USA, pp. 770-778.
9. Jia, B., & Huang, Q. (2020). DE-CapsNet: A diverse enhanced capsule network with disperse dynamic routing. *Applied Sciences, 10*(3), 884.
10. Berman, D. S. (2019). DGA CapsNet: 1D application of capsule networks to DGA detection. *Information, 10*(5), 157.
11. Butun, E., Yildirim, O., Talo, M., Tan, R. S., & Acharya, U. R. (2020). 1D-CADCapsNet: One dimensional deep capsule networks for coronary artery disease detection using ECG signals. *Physica Medica, 70*, 39–48.
12. Li, H., Liu, H., Ji, X., Li, G., & Shi, L. (2017). Cifar10-dvs: An event-stream dataset for object classification. *Frontiers in Neuroscience, 30*(11), 309.
13. Xiao, H., Rasul, K., & Vollgraf, R. (Aug 2017). "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms", arXiv preprint, arXiv:1708.07747.
14. LeCun, Y., Cortes, C., & Burges, C. (1998). "The MNIST database of handwritten digits,". http://yann.lecun.com/exdb/mnist/. [Last Accessed, April 2021].
15. Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (Jun.2010). "The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," In *2010 ieee computer society conference on computer vision and pattern recognition-workshops*, San Francisco, CA, USA, 13-18, pp.94-101.

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

Mohammed Abo-Zahhad (SM'00) received the B.S. and M.S. degrees in electrical engineering, both from the University of Assiut, Egypt, in 1979 and 1983, respectively, and the Ph.D. degree from the University of Kent, Canterbury, U.K., and Assiut University (channel system), in 1988. Since 1999, he has been a Professor of electronics and communication engineering. He is a member of the European Society of Circuit Theory and Applications in 1998, a member of the National Communication and Electronics Promotion Committee, and a Reviewer of the National Quality Assurance and Accreditation Authority, NAQQA, Egypt, since 2011. Currently he is the Dean of the School of Electronics, Communication and Computer Engineering, Egypt- Japan University of Science and Technology (E-JUST) since 2017. He was the former General Director of the Assiut University and EJUST Information and Communication Technology Centre. Recently, he has been selected by the ministry of higher education as an expert for setting the study plans and regulations for newly constructed Egyptian universities, 1,500,000rked as a consultant for curriculum reform for Egyptian, Jordanian and Saudi Arabia Universities. He is the founder of Biomedical and Bioinformatics Engineering program, E-JUST. Prof. Abo-Zahhad research interests include wireless communications and multimedia processing, Biomedical and genomic signal processing, image and video processing, switched-capacitor circuits and systems, data compression, wireless sensor networks, massive MIMO and millimeter wave communications, and electronic circuits. He has published more than 190 papers in recognized impacted international journals and conferences. He was a recipient of the Encouragement State Award in Engineering, from the Egyptian Research and Technology Academy, Ministry of Higher Education, Egypt, in 2005.

Islam Eldifrawi Received the BSc from communications department in Faculty of Engineering Alexandria University with Distinction with Honors. He joined ITI after his graduation then he joined the multinational company ’DELL’ and worked there as a python programmer and instructor, and also as a storage virtualization engineer for three years. He received many awards for developing intelligent programs that helps in customer service and troubleshooting storage virtualization problems. He also worked as a part time Machine Learning instructor at ’New Horizons Center’ in Cairo for the NTL initiative teaching Udacity’s artificial intelligence Nanodegree, then he worked as a Machine Learning Engineer in ’Algorithms Innovative Solutions’ company developing intelligent recommendation systems. Currently he is working as a teacher assistant and he received his MSc in Image Processing at the Faculty of Engineering in Egypt Japan University of Science at Borg Elarab.

Moataz Abdelwahab is a faculty member at the School of Electronics, Communication and Computer engineering, and Deputy Vice president for Regional and International Affairs at EJUST University, where he established the Image and Video processing group. He received his B.Sc and M.Sc in Electrical Engineering from Alexandria University, Egypt, and the PhD degree in Electrical Engineering from University of Central Florida, USA. Dr. Abdelwahab has been conducting research in the area of facial, human action, and Gesture recognition, traffic monitoring, Lung Cancer Detection and Image Analysis. He has attracted more than 1500000 L.E in Research Fund and published more than 45 publications, one book chapter and has one USA Patent and three best paper awards.

Ahmed H. Abd El-Malek received the B.Sc. and M.Sc. degrees in electrical engineering from Alexandria University, Egypt in 2007 and 2010, respectively. He received his Ph.D. degree from the Electrical Engineering Department, King Fahd University of Petroleum & Mineral (KFUPM), Saudi Arabia in 2016. Currently, he is an assistant professor in the Electronics and Communications Engineering Department, Egypt-Japan University of Science and Technology. His research interests are cognitive radio, design, and analysis of wireless networks, network coding, physical layer security, and interference cancellation.