BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 71(6), 2023, Article number: e147338
DOI: 10.24425/bpasts.2023.147338

Improved efficient capsule network for Kuzushiji-MNIST benchmark dataset classification

Michał BUKOWSKI, Izabella ANTONIUK, and Jarosław KUREK*

Department of Artificial Intelligence, Institute of Information Technology, Warsaw University of Life Sciences, Nowoursynowska 159, Warsaw, 02-776, Poland

Abstract. In this paper, we present an improved efficient capsule network (CN) model for the classification of the Kuzushiji-MNIST and Kuzushiji-49 benchmark datasets. CNs are a promising approach in the field of deep learning, offering advantages such as robustness, better generalization, and a simpler network structure compared to traditional convolutional neural networks (CNNs). Proposed model, based on the Efficient CapsNet architecture, incorporates the self-attention routing mechanism, resulting in improved efficiency and reduced parameter count. The experiments conducted on the Kuzushiji-MNIST and Kuzushiji-49 datasets demonstrate that the model achieves competitive performance, ranking within the top ten solutions for both benchmarks. Despite using significantly fewer parameters compared to higher-rated competitors, presented model achieves comparable accuracy, with overall differences of only 0.91% and 1.97% for the Kuzushiji-MNIST and Kuzushiji-49 datasets, respectively. Furthermore, the training time required to achieve these results is substantially reduced, enabling training on non-specialized workstations. The proposed novelties of capsule architecture, including the integration of the self-attention mechanism and the efficient network structure, contribute to the improved efficiency and performance of presented model. These findings highlight the potential of CNs as a more efficient and effective approach for character classification tasks, with broader applications in various domains.

Key words: efficient capsule networks; Kuzushiji-MNIST; Kuzushiji-49; deep learning.

1. INTRODUCTION

In recent years, deep learning algorithms have emerged as a prevalent method for addressing a diverse range of problems. The context of convolutional neural networks (CNNs) and their numerous applications in image-processing tasks is an especially interesting area of research [1–3]. Such tasks can include object recognition within images as well as assessment of various image parameters, for which CNN-based approaches are typically applied. However, to achieve high levels of accuracy, substantial volumes of training data are required.

Recent research has predominantly concentrated on incremental improvements in performance, e.g. enhancing accuracy by 0.01% percentage points. Although the quality of the obtained results has exhibited a gradual increase, this progress is accompanied by a simultaneous escalation in network complexity and the quantity of data needed to attain the desired accuracy. Unfortunately alternative approaches to the problem are often overlooked.

One approach to addressing this issue involves applying a distinct model, as exemplified by the introduction of convolutional neural networks (CNNs). In order to provide significant improvement to general quality of used solutions, some novel approaches are necessary. In that aspect, capsule networks (CapsNets or CNs) represent a promising methodology, offering an innovative perspective on object classification [4,5].

The primary objective of the authors in developing capsule networks was to enhance the capabilities of CNNs by designing a more efficient solution. The architecture of CNs is characterized by a shallower structure and fewer parameters, which results in improved generalization, especially when encountering new viewpoints. Capsule layers within the network can capture complex relationships between object parts and effectively represent the it as a whole. The learning process for these relationships, known as routing, has been analyzed by several researchers who have attempted to either improve [6–8] or eliminate it [9]. Additionally, other studies have focused on examining the theoretical properties of routing [10, 11].

In their early implementation, capsule networks (CNs) demonstrated state-of-the-art performance on the MNIST dataset. Additionally, obtained results were superior for overlapping digits when compared to CNN-based solutions [11]. CNs at this point were not able to achieve comparable performance on other datasets, such as CIFAR-10 with a 10.6% error rate. It is important to note though, that these scores remained within the range of initial CNN implementations prior to subsequent architectural improvements. CNs offer a series of advantages, such as preservation of position and pose information, reduced training data requirements, and robustness to translations, rotations, and other affine transformations. Considering those factors, it is worthwhile to assess and explore the full potential of these networks.

The CapsNet, is an advanced deep learning architecture introduced in 2017 by Geoffrey Hinton and his research team. This innovative neural network is designed to overcome the limitations inherent in CNNs by applying the concept of cap-

*e-mail: jaroslaw_kurek@sggw.edu.pl

Manuscript submitted 2023-04-22, revised 2023-07-16, initially accepted for publication 2023-09-04, published in December 2023.

sules. Capsules are diminutive components engineered to encapsulate the properties of an object in a more robust and comprehensible manner compared to conventional convolutional methods. In this manuscript, we will explore the fundamental principles of capsule networks and elucidate the associated advantages. Main improvement is provided by architecture modification and used parameter sets. Capsule networks exhibit exceptional performance in diverse tasks with a reduced number of parameters compared to state-of-the-art solutions.

In the conducted experiments, the Efficient Capsule Networks methodology is applied, as delineated by Mazzia *et al.* [12]. Presented model integrates the self-attention technique as a routing mechanism [13]. The rationale for selecting this model is threefold. Firstly, the self-attention mechanism has demonstrated remarkable success in large-scale language models [14]. Secondly, the implementation necessitates fewer parameters than the original capsule network. Lastly, it exhibits superior benchmark performance on the MNIST and smallNORB datasets, offering a robust codebase for the present investigation (available at [15], as a source code to [12]). Although the Efficient approach has received comparatively less attention in the existing literature, it holds substantial potential for a wide array of future applications.

In order to assess the efficacy of the proposed network, the KMNIST dataset was applied as a benchmark [16]. The primary motivation behind this selection was to demonstrate the suitability of capsule networks for 2D datasets, which exhibit fewer viewpoint parameters compared to 3D spaces due to the absence of perspective information. Furthermore, the Kuzushiji-49, a constituent of the KMNIST dataset, encompasses 49 distinct classes, presenting a five-fold increase in complexity in comparison with the MNIST dataset [17]. This serves to illustrate the adaptability of the proposed method to tackle more intricate challenges.

In this paper, we present an improved efficient Capsule Network model for the classification of the Kuzushiji-MNIST benchmark dataset. The model is based on the Efficient CapsNet architecture, which integrates the self-attention mechanism as a routing mechanism. The self-attention mechanism has demonstrated remarkable success in large-scale language models and offers a more efficient alternative to the dynamic routing mechanism used in the original CapsNet.

The primary objective of this research is to evaluate the efficacy of the proposed network on the Kuzushiji-MNIST dataset. We compare the performance of our model to the top solutions in the benchmark and assess its accuracy, training time, and number of parameters. By demonstrating the suitability of capsule networks for this dataset, we aim to highlight the unique contributions of our study and the potential of CNs as a more efficient approach to object classification.

2. CAPSULE NETWORK ADVANTAGES, DISADVANTAGES AND APPLICATIONS

In 2017, Geoffrey Hinton and his colleagues introduced a novel class of neural networks known as capsule networks [6]. The primary components of these networks, termed capsules, aim to encapsulate the attributes of an object, including its orientation, dimensions, and spatial location. Compared to conventional convolutional methods CapsNets offer a more robust and comprehensible approach.

CapsNets are designed to capture the hierarchical relationships between different features in an image. They are also designed to be more robust to variations in the position, scale, and orientation of objects in the input data, improving performance in tasks where these factors are important. Another improvement is dynamic routing (routing by agreement) used to guide information between capsules in a more efficient and meaningful way. This can lead to better performance and improved generalization when object classification is considered. When combined with reduced number of pooling layers than the traditional CNNs, more spatial information can be preserved. Finally, CNs preserve spatial relationships between features, resulting in better handling of overlapping objects.

At the same time CNs are not without drawbacks. They require more complex routing algorithms to establish relationships between different layers. CNs are also relatively new area of research. Fewer pre-trained models, optimization techniques, and best practices are readily available, than in the case of better explored solutions. Due to their increased computational complexity, CapsNets can be challenging to scale for larger input sizes or deeper architectures. Furthermore, while CNs are designed to be more robust to changes in viewpoint, pose, and other affine transformations, it might not be the case for adversarial examples or other kinds of noise. The dynamic routing algorithm used in CapsNets can make it challenging to interpret the learned features and understand the network decision-making process. Also due to routing algorithm and the need for careful hyperparameter tuning, they can be difficult to train. CapsNets have shown promise in certain computer vision tasks, but their applicability and performance across various domains have not yet been thoroughly explored.

Application of CNs for image segmentation might require some adjustment, either by appropriate data preprocessing, or incorporating additional methods in the overall solution. At the same time it can be clearly seen that this approach shows great promise. CNs are able to achieve similar results to state-of-the-art solutions, with fewer parameters and better generalization. When it comes to CNs applications, there are few areas of research, where such solutions are used.

First set of algorithms focuses on different aspects of image segmentation. Approaches belonging to this section are often parts of other, more complex systems, but they can also be a solution in itself. In [18] authors use locally-constrained routing and transformation matrix sharing, in the image segmentation of computer tomography scans of pathological lungs, muscle and adipose (fat) tissue from magnetic resonance imaging scans (MRI) of human subjects' thighs. CN-based processing was able to outperform other methods on all datasets, with less than 5% of the parameters used by U-Net: state-of-the-art solution in biomedical image segmentation. Similarly, in [19] CNs were used for object segmentation in medical data for the pathological lungs from low dose of CT scans, reducing number of parameters by 95.4%, while still maintaining better segmen-

tation accuracy. Different approaches focus on magnetic resonance images of the left ventricle [20], brain tumour automatic segmentation [21] or categorizing cervical lesion imagery [22]. In all cases, CN-based solutions achieve high accuracy, with significantly lower number of parameters.

Second, fairly common set of applications for the capsule networks are ones connected to text analysis elements. CNs are able to handle different contexts and can be well adapted to various problems in this area. Such tasks can include cyberbullying detection [23], text sentiment classification [24] or general text classification [25]. Different set of solutions focuses on improving certain aspects of this process. In [26] in order to reduce number of parameters used for creating word embedding, compositional weighted coding method is proposed. Authors of [27] consider question-answering systems, providing Deep Refinement pipeline. CN and attention mechanism are used, while the pipeline is applied to primarily classify the text into two categories: sincere and insincere. Proposed question classification method outperforms the previously used ones, with the F1 score equal to 0.978. Authors of [28] consider the increased performance of capsule-based solutions, classifying hierarchical multi-label text with a simple CN. In [29] authors explore the possibility of sharing knowledge between related tasks in order to increase the amount of training data. They use capsule-based learning architecture for multi-task purpose. Final claim denotes it as unified, simple and effective, with routing algorithm able to cluster the features for each task in the network.

Solutions focusing on image recognition problems are the most interesting from the point of view of research presented in this paper. There are quite a few approaches in this field, showing that CNs handle such problems relatively well. In [30] the problem of sign language recognition is considered. Traffic sign detection using capsule network is the main topic presented in [31]. Authors use dynamic routing and route by agreement algorithms to instantiate object parameters, such as pose and orientation. As shown in [32], CNs can even be applied to military grade object detection. Authors introduce architecture based on CapsNet, with the presented variant denoted as multi-level CapsNet framework and report that the obtained precision for the object recognition task was superior to many other algorithms. One especially important factor for CNs in such complex problems is routing algorithm used. In [8] authors propose a general-purpose "routing by agreement" method, and the proposed method was able to improve the overall performance of the CNs. Another interesting application, presented in [33], uses capsule networks in the Q-Learning based game algorithms, while in [34] CNs are used in a complex, realistic scenarios of the real world navigation.

Authors of [35] consider similar problem to the one presented in this paper – handwritten character recognition. They use data augmentation to generate realistically modified examples, reflecting actual variations that tend to happen in human writing. Initial number of used samples was equal to 200 per class. Final solution was able to surpass results for the EMNIST-letter dataset, and achieve the results present in EMNIST-balanced, EMNIST-digits, and MNIST datasets. In [12] – solution used in experiments presented in this paper – authors investigate the overall CN efficiency. Proposed algorithm was able to achieve state-of-the-art results on three different datasets, with only 160K parameters – 2% of parameters used by CapsNet.

Despite the prevalent applications of capsule methodologies to image data, a number of studies have concentrated on the video domain. In a manner analogous to the generalization of 2D image-based convolutions to 3D convolutions for processing video frame sequences [36], the traditional 2D convolutional capsule routing was extended to 3D convolutional routing as described in [37]. This 3D convolutional routing approach enables the routing of capsules that are not only spatially proximate but also temporally related, thereby facilitating the generation of higher-layer capsule outputs. In [37] authors introduced a novel video capsule network, denoted as VideoCapsuleNet, which facilitates end-to-end action detection. Study presented in [38] introduces a novel approach called CapsuleVOS, designed for video object segmentation tasks. This method requires an input video sequence with frames containing initial object segmentation. The primary objective of CapsuleVOS is to accurately propagate the object segmentation across the entire video sequence.

Overall, capsule networks tend to work well for image-based problems, offering high accuracy with relatively low number of parameters and better efficiency. CNs in general are an interesting and very promising solution, with vast possibilities. Those advantages were the main reason, why CN-based solution was chosen as a focus of research presented in this paper. For an extensive review of various CapsNet applications, the reader is referred to the comparative study by Vijayakumar *et al.* [39].

3. DATASET

Dataset selection was one of key problems considered for the chosen research area. The CNs have diverse possibilities, but in order to show them, the images used need to be appropriate to the network capabilities.

In that aspect, the Kmnist dataset was considered [16]. The full dataset contains total of three subsets, each with increasing level of complexity. The images are represented as 2D grayscale ones, with examples in each set retaining common size. The first subset, Kuzushiji-MNIST, is a straight-up replacement of original MNIST dataset [40]. This dataset replicates the number of examples for train and test datasets (respectively 60 000 and 10 000), the number of classes (10 total) and image dimensions (grey-scale, 28×28 pixels each). Second subset, Kuzushiji-49 keeps the format but contains 270 912 samples belonging to total of 49 classes. It was designed to engage the machine learning in the field of Japanese literature, and contains instances of Hiragana characters. Final subset, Kuzushiji-Kanji, contains total of 3832 Kanji characters, represented by 140,426 images, with size equal to 64×64 pixels.

While it is a good initial benchmark, first dataset was deemed not complex enough to show full CN capabilities. On the other hand, the third dataset is highly unbalanced – some classes are represented by only single image – and due to high risk of

this imbalance influencing the model performance, it is also not the best fit. For the testing purposes, the Kuzushiji-MNIST and Kuzushiji-49 sets were chosen, as they introduces both higher level of complexity and balance required to properly evaluate the CN capabilities. Example images from the Kuzushiji-49 dataset are presented in Fig. 1.

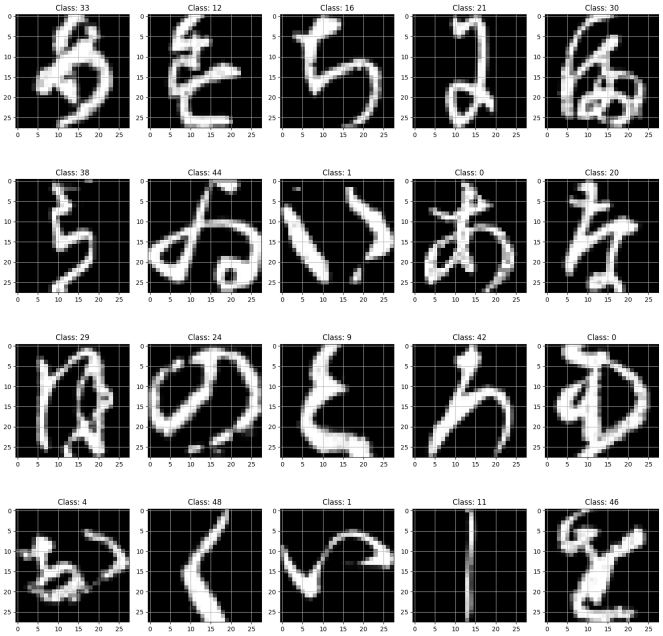

Fig. 1. Sample images representing different Hiragana characters from the Kuzushiji-49 dataset

4. MODELLING AND SETUP

The capsule networks are a new solution, with great promise due to their different approach to the object classification. Last iteration of capsule networks, called CapsNet uses the dynamic routing approach [6], with reconstruction network as regularizer and explainability mechanism.

The model network is a simple structure of single convolutional layer of 256 filters, layer of primary capsules which are connected to digits capsules. It encapsulates multiple scalars to form a vector, where length is the activation of the capsule (scaled to stay in 0–1 range). The vector dimensions are instantiating parameters of an object the capsule is capturing. To interpret those parameters, the generator regularization networks can be used, so each dimension can be perturbed and see what effect on reconstruction it will have. The general structure of CapsNet model is shown in Fig. 2

While the CapsNet model is interesting in the general approach, it is also a base model, which was already improved. In order to accurately assess the possibilities that CN provide, the Efficient CapsNet model was chosen as a base for research presented in this paper.

4.1. Efficient CapsNet

The Efficient CapsNet version of the CN model was chosen for benchmarking in this paper due to a few important reasons.

First of all, this solution was already tested on the MNIST dataset, achieving better results than the initial model, and therefore proving its superior capabilities. Additionally, Efficient CapsNet was also tested on different datasets, including one used for character recognition in the case of letters: EMNIST [35, 41]. In both cases, the model achieved good results.

The Efficient CapsNet introduces some important changes in comparison to the original solution [12]. It exchanges the dynamic routing in the first approach, with the self-attention mechanism. Since it is a non-iterative method, this provides significant improvement to the model efficiency in terms of required operations. The study examines the efficiency of capsule networks. It is demonstrated, that an extreme architecture with only 161 000 parameters can still achieve state-of-the-art results on three distinct datasets, using just 6 800 000 (2%) of the original CapsNet parameters. The authors of model chosen as a base for research presented in this paper, also provide a very good python implementation of their solution, using tensorflow [15]. This fact is extremely important, since obtained results can be easily reproduced. The overall architecture of the Efficient CapsNet model is presented in Fig. 3.

To further improve the performance, we introduce a novel non-iterative, highly parallelizable routing algorithm. It can effectively handle a smaller number of capsules, replacing the dynamic routing mechanism. Comprehensive experiments with alternative capsule implementations have confirmed the efficacy of our approach and the capacity of capsule networks to efficiently incorporate visual representations more conducive to generalization. Our solution uses deeper architecture, with 4 convolutional layers before the 2 layers of capsules.

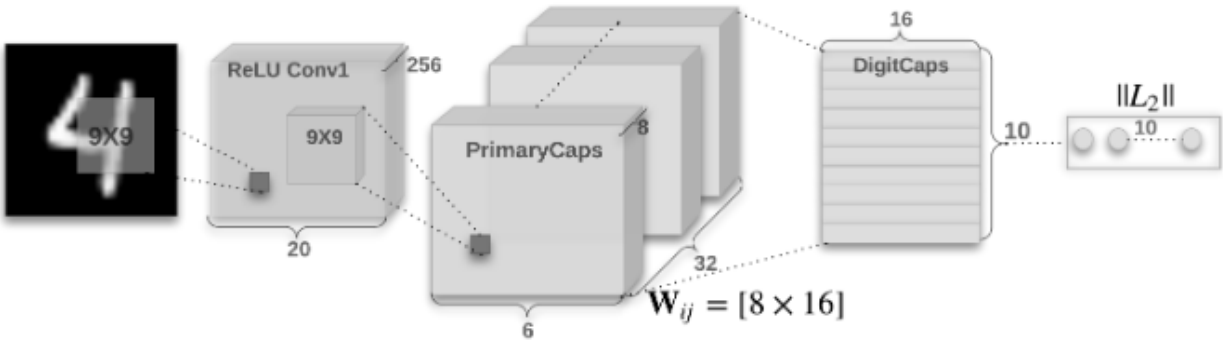

Fig. 2. Architecture of Original Capsule Network [6]

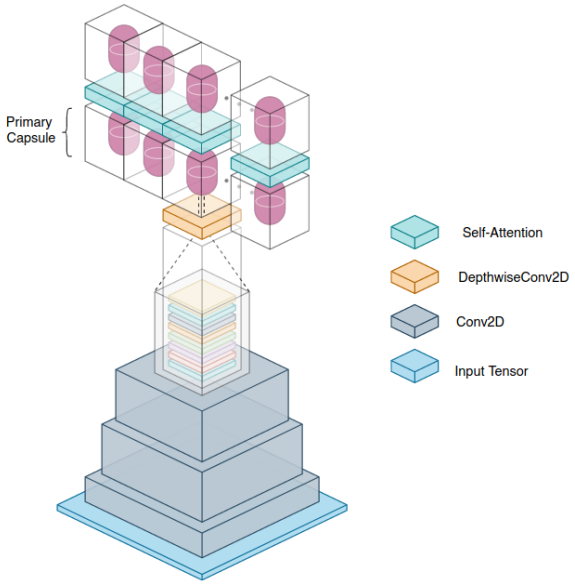

Fig. 3. Architecture of Efficient CapsNet Network [12]

4.2. Improvements and setup

All experiments were performed on a workstation with an Nvidia RTX3080 GPU with 10GB of memory and 32GB of DDR4 SDRAM. We use the TensorFlow 2.10.0 framework with CUDA 11 using Python 3.10.8.

For the dataset division, we used the well-established practices, present in the general CNN approaches. The sets were divided into three subsets: train, eval and test. The last one was provided by the authors of KMNIST. Finally, 20% of training set was used as validation for establishing the number of epochs and hyperparameters in order to select the best model.

The model was trained for 150 epochs using hyperparameters provided in [12]. The learning rate was modified to start at 0.0002 and minimal learning rate was set at 0.00003.

The overall architecture was also modified, increasing number of convolutional layers and primary capsules. In the presented approach total of 6 convolutional layers were used, each followed by batch normalization. After convolution layers, the primary capsule layer was placed. For the Kuzushiji-MNIST dataset, this layer consisted of 32 8-dimensional capsules. In case of Kuzushiji-49, the number of dimensions was increased, resulting in 32 10-dimentional capsules. Final layer contained number of capsules corresponding directly to number of recognized classes: 10 for the first set, and 48 for the Hiragana character set.

After the model setup, the following number of parameters were obtained for each dataset: 581 792 for Kuzushiji-MNIST, and 1 741 120 for Kuzushiji-49. Out of both parameters sets, the number of non-trainable ones equalled 1152 for the first and 1792 for the second one. Full layer structure for both models – including type of layer, output shape and number of used parameters - are presented in Table 1 and 2, respectively.

5. COMPARISON AND PERFORMANCE ANALYSIS

The novelty of our approach lies in the modifications made to the Efficient CapsNet model, which resulted in improved performance on the Kuzushiji-MNIST and Kuzushiji-49 datasets. By comparing the two models, we can gain valuable insights into the specific improvements made and their impact on performance.

The original CapsNet model, introduced by Sabour *et al.* [6], was designed to overcome the limitations of convolutional neu-

Table 1
Network architecture for the Efficient CapsNet trained on Kuzushiji-MNIST dataset

Layer (type)	Output Shape	Param
input_8 (InputLayer)	[(None, 28, 28, 1)]	0
conv2d_5 (Conv2D)	(None, 24, 24, 32)	832
batch_normalization_5 (BatchNormalization)	(None, 24, 24, 32)	128
conv2d_6 (Conv2D)	(None, 22, 22, 64)	18496
batch_normalization_6 (BatchNormalization)	(None, 22, 22, 64)	256
conv2d_7 (Conv2D)	(None, 20, 20, 96)	55392
batch_normalization_7 (BatchNormalization)	(None, 20, 20, 96)	384
conv2d_8 (Conv2D)	(None, 18, 18, 128)	110720
batch_normalization_8 (BatchNormalization)	(None, 18, 18, 128)	512
conv2d_9 (Conv2D)	(None, 8, 8, 256)	295168
batch_normalization_9 (BatchNormalization)	(None, 8, 8, 256)	1024
primary_caps_1 (PrimaryCaps)	(None, 32, 8)	16640
fc_caps_1 (FCCaps)	(None, 10, 32)	82240
length_capsnet_output (Length)	(None, 10)	0
Total params:	581 792	
Trainable params:	580 640	
Non-trainable params:	1 152	

Table 2
Network architecture for the Efficient CapsNet trained on Kuzushiji-49 dataset

Layer (type)	Output Shape	Param
input_9 (InputLayer)	[(None, 28, 28, 1)]	0
conv2d_6 (Conv2D)	(None, 24, 24, 32)	832
batch_normalization_6 (BatchNormalization)	(None, 24, 24, 32)	128
conv2d_7 (Conv2D)	(None, 22, 22, 64)	18496
batch_normalization_7 (BatchNormalization)	(None, 22, 22, 64)	256
conv2d_8 (Conv2D)	(None, 20, 20, 96)	55392
batch_normalization_8 (BatchNormalization)	(None, 20, 20, 96)	384
conv2d_9 (Conv2D)	(None, 18, 18, 128)	110720
batch_normalization_9 (BatchNormalization)	(None, 18, 18, 128)	512
conv2d_10 (Conv2D)	(None, 16, 16, 256)	295168
batch_normalization_10 (BatchNormalization)	(None, 16, 16, 256)	1024
conv2d_11 (Conv2D)	(None, 7, 7, 320)	737600
batch_normalization_11 (BatchNormalization)	(None, 7, 7, 320)	1280
primary_caps_1 (PrimaryCaps)	(None, 32, 10)	16000
fc_caps_1 (FCCaps)	(None, 49, 32)	503328
length_capsnet_output (Length)	(None, 49)	0
Total params:	1 741 120	
Trainable params:	1 739 328	
Non-trainable params:	1 792	

ral networks (CNNs) by using capsules instead of neurons. Capsules are groups of neurons that represent different properties of an object, such as its orientation, dimensions, and spatial location. The primary advantage of capsules is their ability to capture hierarchical relationships between different features in an image, making them more robust to variations in position, scale, and orientation.

The Improved Efficient CapsNet model builds upon the original CapsNet model by incorporating the self-attention mechanism as a routing mechanism, as proposed by Mazzia *et al.* [12]. This modification replaces the dynamic routing mechanism used in the original model, resulting in a more efficient and parallelizable routing algorithm. The self-attention mechanism has been successful in large-scale language models and has demonstrated the ability to capture complex relationships between different parts of an object.

In terms of performance, the Improved Efficient CapsNet model achieved comparable accuracy to the original CapsNet model on the Kuzushiji-MNIST and Kuzushiji-49 datasets. However, the Improved Efficient CapsNet model achieved this performance with significantly fewer parameters, making it a more efficient solution. The total number of parameters used in the Improved Efficient CapsNet model was 0.58M for the Kuzushiji-MNIST dataset and 1.7M for the Kuzushiji-49 dataset, compared to the original CapsNet model, which used 26.2M parameters. This reduction in parameters is particularly important for practical applications, as it allows for faster training times and reduces the computational resources required.

Furthermore, the Improved Efficient CapsNet model demonstrated faster training times compared to the original CapsNet model. The Improved Efficient CapsNet model was trained for 150 epochs, with a total training time of 50 minutes for the Kuzushiji-MNIST dataset and 4 hours and 30 minutes for the Kuzushiji-49 dataset. In contrast, the original CapsNet model required 1800 epochs and a training time of 290 hours for the Kuzushiji-49 dataset. This significant reduction in training time makes the Improved Efficient CapsNet model more practical and accessible for researchers and practitioners.

In conclusion, the Improved Efficient CapsNet model presented in this paper offers several improvements over the original CapsNet model. The incorporation of the self-attention mechanism as a routing mechanism results in a more efficient and parallelizable routing algorithm. This modification allows for faster training times and reduces the number of parameters required, while still achieving comparable accuracy on the Kuzushiji-MNIST and Kuzushiji-49 datasets. The Improved Efficient CapsNet model provides valuable insights into the specific improvements made and their impact on performance, making it a promising solution for character classification tasks.

6. RESULTS AND DISCUSSION

The main goal of research presented in this paper was to evaluate the Efficient CapsNet model in terms of applicability and overall performance. As shown in various research paper, the CNs can be used to solve different problems, ranging from text classification, to image segmentation and object recognition [19, 23, 26, 30–32]. What is even more important is that prepared models were able to achieve similar accuracy to state-of-the-art solutions in selected areas, using significantly fewer parameters.

Two subsets of Kmnist dataset were used to evaluate prepared model. Prepared solution is a modification of Efficient CapsNet approach [12]. According to specifications provided by the authors of the dataset, the metric accuracy on the Kuzushiji-49 was balanced. In order to evaluate the model performance, mean accuracy for all classes was calculated. Additionally, model trained on Kuzushiji-MNIST subset was used as a base evaluation for the modified solution used in experiments. Final results were compared to the best-performing models in given dataset, and are presented in Tables 3 and 4, as of December 2022.

Overall, both models managed to score in the top ten in their respective benchmarks. The network trained with Kuzushiji-MNIST dataset scored in 4th place, while the Kuzushiji-49 one was placed on 6th position. Both models achieved relatively high accuracy, reaching overall score equal to respectively 98.43% and 96.32%. While neither model was able to reach first place, both approaches performed reasonably well. The results are more than satisfactory, showing that capsule network architecture generalizes well for high number of classes. Presented solution can be trained in reasonable amount of time on consumer grade hardware. At the same time the achieved accuracy is on par with models using far larger number of parameters. For the Kuzushiji-MNIST dataset the overall accuracy difference reaches only 0.91%, while in the case of Kuzushiji-49 it equals 1.97%. While the difference is not negligible, it is small enough that the presented approach has a resonable chance to beat those scores after some improvements. In that aspect, one possible area of future work might include using ensemble solutions, or different overall network structure.

In Table 3, we present the eight leading models regarding their accuracy on the Kuzushiji-MNIST dataset as of December 2022. The shake-shake-26 2x96d (S-S-I) model, with Cutout 14, holds the first place, boasting an impressive accuracy of 99.34% but with a relatively high number of parameters (26.2M). Interestingly, despite not using a pre-trained network, it only takes roughly 6 hours and 46 minutes to train.

The "Improved Efficient Capsnet," the model at the core of study presented in this paper, achieved an accuracy of 98.43%, putting it at 4th place. Notably, it achieved this performance with significantly fewer parameters (0.58M), demonstrating its efficacy and computational efficiency. It completed training in just 50 minutes over 150 epochs, underscoring its relatively fast training time.

In comparison, the ResNet18 + VGG Ensemble model, with a slightly higher accuracy of 98.90%, required 26M parameters and had the benefit of using a pre-trained network, showing that our model can perform competitively without such advantages, while being significantly lighter.

Moving on to Table 4, which presents the top eight models regarding their balanced accuracy scores on the more complex Kuzushiji-49 dataset. Here, the "Improved Efficient Cap-

Table 3
Top 8 accuracy scores for Kuzushiji-MNIST as for December 2022

Place	Model	Accuracy	Number of parameters	Pretrained network	Number of epochs	Training time
1	shake-shake-26 2x96d (S-S-I), Cutout 14	99.34%	26.2M	No	200	6h46m
2	ResNet18 + VGG Ensemble	98.90%	26M	Yes	N/A	3m
3	PreActResNet-18 Manifold Mixup	98.83%	11M	No	200	N/A
4	**Improved Efficient Capsnet**	**98.43%**	**0.58M**	**No**	**150**	**50m**
5	PreActResNet-18 + Input Mixup	98.41%	11M	No	200	N/A
6	PreActResNet-18	97.82%	11M	No	200	N/A
7	Original Capsule Networks	97.66%	6.8M	No	150	1h35m
8	Keras Simple CNN Benchmark	94.63%	1.2M	No	12	N/A

Table 4
Top 8 balanced accuracy scores for Kuzushiji-49 as for December 2022

Place	Model	Accuracy	Number of parameters	Pretrained network	Number of epochs	Training time
1	Shake-Shake-26 2x96d (cutout 14))	98.29%	26.2M	No	1800	290h
2	PreActResNet-18 + Manifold Mixup	97.33%	11M	No	200	N/A
3	DenseNet-100 (k=12)%	97.32%	7M	No	1500	47h39m
4	PreActResNet-18 + Input Mixup	97.04%	11M	No	200	N/A
5	PreActResNet-18	96.64%	11M	No	200	N/A
6	**Improved Efficient Capsnet**	**96.32%**	**1.7M**	No	150	4h30m
7	Original Capsule Networks	91.37%	12.5M	No	150	14h
8	Keras Simple CNN Benchmark	89.36%	1.2M	No	12	N/A

snet" placed 6th with an accuracy of 96.32%. Although this is a slightly lower ranking than the previous table, it is worth noting that the model only used 1.7M parameters, confirming its ability to perform well even with lower computational resources. The training time was also relatively short, requiring only 4 hours and 30 minutes for 150 epochs.

In comparison, the leading Shake-Shake-26 2x96d (cutout 14) model achieved an accuracy of 98.29% but at the cost of a substantially larger number of parameters (26.2M) and a remarkably long training time of 290 hours.

In summary, the "Improved Efficient Capsnet" model demonstrates competitive performance on both the Kuzushiji-MNIST and Kuzushiji-49 datasets, achieving high accuracy rates with relatively few parameters and shorter training times, indicating its potential as a more efficient approach for Kuzushiji character recognition tasks.

The accuracy metric is commonly used due to its simplicity and direct interpretation. However, it does not provide a comprehensive picture of the model performance, particularly when the classes are imbalanced. For this reason, we consider precision, recall, and F1-score metrics in addition to accuracy for our performance assessment.

Precision is the ratio of true positive predictions to the total positive predictions, which indicates the exactness or quality of the model. Recall (also known as sensitivity) is the ratio of true positive predictions to the total actual positives, which illustrates the completeness or quantity the model can provide. The F1-score is the harmonic mean of precision and recall, providing a balanced measure between precision and recall.

In an interesting turn of events, we observed that the average and weighted average of precision, recall, and F1-score metrics across all classes are identical to the accuracy metric for our capsule network model applied on the Kuzushiji-MNIST dataset. This unusual equivalence is consistent for all models, even when tested on the Kuzushiji-49 dataset, with differences falling within a narrow margin of ± 0.01 percentage points.

To illustrate, we present a detailed example of precision, recall, and F1-score metrics for each class from the Original Capsule Networks model on the Kuzushiji-MNIST dataset in Table 5.

The occurrence of such equalities is atypical, given the fundamental differences in these performance metrics. They each serve distinct purposes and are not expected to agree so closely unless the dataset is perfectly balanced and the model performs equally well on all classes. This unique finding indicates a remarkable robustness and balance in the classification capability of our improved capsule network model.

The analysis prompts a more in-depth investigation into the properties and configuration of the capsule network that yield such an unusual performance consistency across metrics. This will form the basis for subsequent research aimed at unravelling the inherent characteristics and peculiarities of this model, particularly in the context of Kuzushiji character recognition.

It is also important to point out that while analysing the code of better-scoring solutions, it was noted that some were using test set as a validation one for the training purposes. Such actions can lead to model overfitting, and while it achieves better results on the original dataset, the overall solution versatility will suffer. Both models presented in this paper were trained using the established best practices for similar problems, splitting the dataset into train, eval and test datasets.

Table 5
Example comparison of accuracy metric with other metrics for the Original Capsule Networks model on the Kuzushiji-MNIST dataset

No. of class	Precision [%]	Recall [%]	F1-score [%]
0	96.46	98.20	97.32
1	97.98	97.10	97.54
2	98.25	95.40	96.80
3	97.16	99.10	98.12
4	96.65	95.20	95.92
5	98.18	97.20	97.69
6	97.82	98.70	98.26
7	99.10	98.70	98.90
8	98.11	98.40	98.25
9	96.95	98.60	97.77
Average [%]	97.66	97.66	97.66
Weighted average [%]	97.66	97.66	97.66
Accuracy [%]	97.66		

Presented approach is time- and resource-efficient, allowing incorporation of the k-fold cross-validation. In the case of the deep learning approach, due to long computational time, we utilized 5-fold cross-validation ($k = 5$). The validation was performed k-times with increasing training time. The computation was efficient enough to still remain within capabilities of personal workstation with reasonable overall training period. It is important to note that the top performing solution for the Kuzushiji-49 set (shake-shake-26 2x96d) required 1800 epochs and total of 290 hours of training. Experiments for that network were performed with eight Tesla V100 GPUs, with 26.2M of final parameters. The Efficient CapsNet approach presented in this paper used total of 150 epochs, while overall training time equalled 4.5 hours. The experiments were done on significantly less efficient machine using single GPU (full specification of the used workstation is presented in 4.2), while final parameters count equalled only 1.7M for the resulting model.

7. CONCLUSIONS

In this paper an approach to character classification for Kuzushiji-MNIST and Kuzushiji-49 datasets is presented. The solution uses model based on Efficient CapsNet architecture, taking advantage of strong points of capsule-network-based solutions, such as robustness, simpler structure and better generalization.

As shown by the obtained results, capsule networks are a promising solution, when chosen benchmark field is considered. Prepared model was able to score in the top 10 solutions in the two chosen trials. It was able to achieve very similar results to the top-performing ones, with significantly fewer net-

work parameters used. The total accuracy difference between solution presented in this paper, and ones that scored in the first place for both benchmark was 0.91% for the Kuzushiji-MNIST dataset, and 1.97% for Kuzushiji-49. These results were achieved in much shorter training times, allowing the overall process to be performed on not-dedicated GPU workstations.

Overall, CN-based solutions show great promise, and while current approach to Efficient CapsNet modification was not able to achieve top results, it still offers significant advantages. CNs have better generalization and use fewer parameters. The total accuracy loss is more than compensated with shorter training time. Simpler network structure with fewer parameters used results in solution with much wider range of applications, while still leaving some room for future improvements.

REFERENCES

[1] J. Kurek, B. Swiderski, A. Jegorowa, M. Kruk, and S. Osowski, "Deep learning in assessment of drill condition on the basis of images of drilled holes," in *Eighth International Conference on Graphic and Image Processing (ICGIP 2016)*, Y. Wang, T. D. Pham, V. Vozenilek, D. Zhang, and Y. Xie, Eds., vol. 10225, International Society for Optics and Photonics. SPIE, 2017, p. 102251V, doi: 10.1117/12.2266254.

[2] A. Jegorowa, J. Kurek, I. Antoniuk, W. Dołowa, M. Bukowski, and P. Czarniak, "Deep learning methods for drill wear classification based on images of holes drilled in melamine faced chipboard," *Wood Sci. Technol.*, vol. 55, no. 1, pp. 271–293, Jan 2021, doi: 10.1007/s00226-020-01245-7.

[3] J. Kurek *et al.*, "Classifiers ensemble of transfer learning for improved drill wear classification using convolutional neural network," *Mach. Graph. Vis.*, vol. 28, no. 1/4, p. 13–23, Dec. 2019, doi: 10.22630/MGV.2019.28.1.2.

[4] G. Hinton, A. Krizhevsky, and S. Wang, "Transforming auto-encoders," in *Artificial Neural Networks and Machine Learning – ICANN 2011*, vol. 6791, 06 2011, pp. 44–51, doi: 10.1007/978-3-642-21735-7_6.

[5] A. Jegorowa, J. Górski, J. Kurek, and M. Kruk, "Use of nearest neighbors (k-nn) algorithm in tool condition identification in the case of drilling in melamine faced particleboard," *Maderas-Cienc. Tecnol.*, vol. 22, no. 2, p. 189–196, 2020, doi: 10.4067/S0718-221X2020005000205.

[6] S. Sabour, N. Frosst, and G.E. Hinton, "Dynamic routing between capsules," 2017, doi: 10.48550/ARXIV.1710.09829. [Online]. Available: https://arxiv.org/abs/1710.09829

[7] F.D.S. Ribeiro, G. Leontidis, and S.D. Kollias, "Capsule routing via variational bayes," *CoRR*, vol. abs/1905.11455, 2019. [Online]. Available: http://arxiv.org/abs/1905.11455

[8] F.A. Heinsen, "An algorithm for routing capsules in all domains," *arXiv preprint arXiv:1911.00792*, 2019.

[9] A. Byerly, T. Kalganova, and I. Dear, "A branching and merging convolutional network with homogeneous filter capsules," *CoRR*, vol. abs/2001.09136, 2020. [Online]. Available: https://arxiv.org/abs/2001.09136

[10] S.R. Venkatraman, A. Anand, S. Balasubramanian, and R.R. Sarma, "Learning compositional structures for deep learning: Why routing-by-agreement is necessary," *CoRR*, vol. abs/2010.01488, 2020. [Online]. Available: https://arxiv.org/abs/2010.01488

[11] D. Wang and Q. Liu, "An optimization view on dynamic routing between capsules," 2018. [Online]. Available: https://openreview.net/forum?id=HJjtFYJDf

[12] V. Mazzia, F. Salvetti, and M. Chiaberge, "Efficient-CapsNet: capsule network with self-attention routing," *Sci. Rep.*, vol. 11, no. 1, jul 2021, doi: 10.1038/s41598-021-93977-0.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, doi: 10.48550/ARXIV.1706.03762. [Online]. Available: https://arxiv.org/abs/1706.03762

[14] T.B. Brown, B. Mann, N. Ryder, and E.A. Subbiah, "Language models are few-shot learners," 2020, doi: 10.48550/ARXIV.2005.14165. [Online]. Available: https://arxiv.org/abs/2005.14165

[15] V. Mazzia, F. Salvetti, and M. Chiaberge, "Github repository for efficient capsnet." 2021. [Online]. Available: https://github.com/EscVM/Efficient-CapsNet

[16] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. (2018) Deep learning for classical japanese literature.

[17] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

[18] R. LaLonde, Z. Xu, I. Irmakci, S. Jain, and U. Bagci, "Capsules for biomedical image segmentation," *Med. Image Anal.*, vol. 68, p. 101889, 2021.

[19] R. LaLonde and U. Bagci, "Capsules for object segmentation," *arXiv preprint arXiv:1804.04241*, 2018.

[20] Y. He, W. Qin, Y. Wu, M. Zhang, Y. Yang, X. Liu, H. Zheng, D. Liang, and Z. Hu, "Automatic left ventricle segmentation from cardiac magnetic resonance images using a capsule network," *J. X-Ray Sci. Technol.*, vol. 28, no. 3, pp. 541–553, 2020.

[21] M. Elmezain, A. Mahmoud, D.T. Mosa, and W. Said, "Brain tumor segmentation using deep capsule network and latent-dynamic conditional random fields," *J. Imaging*, vol. 8, no. 7, p. 190, 2022.

[22] X. Zhang and S.-G. Zhao, "Cervical image classification based on image segmentation preprocessing and a capsnet network model," *Int. J. Imaging Syst. Technol.*, vol. 29, no. 1, pp. 19–28, 2019, doi: 10.1002/ima.22291.

[23] A. Kumar and N. Sachdeva, "Multimodal cyberbullying detection using capsule network with dynamic routing and deep convolutional neural network," *Multimedia Syst.*, vol. 28, p. 2043–2052, 2022.

[24] B. Chen, Z. Xu, X. Wang, L. Xu, and W. Zhang, "Capsule network-based text sentiment classification," *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 698–703, 2020.

[25] J. Kim, S. Jang, E. Park, and S. Choi, "Text classification using capsules," *Neurocomputing*, vol. 376, pp. 214–221, 2020.

[26] H. Ren and H. Lu, "Compositional coding capsule network with k-means routing for text classification," *Pattern Recognit. Lett.*, vol. 160, pp. 1–8, 2022.

[27] D.K. Jain, R. Jain, Y. Upadhyay, A. Kathuria, and X. Lan, "Deep refinement: Capsule network with attention mechanism-based system for text classification," *Neural Comput. Appl.*, vol. 32, pp. 1839–1856, 2020.

[28] J.S. Manoharan, "Capsule network algorithm for performance optimization of text classification," *J. Soft Comput. Paradigm (JSCP)*, vol. 3, no. 01, pp. 1–9, 2021.

[29] L. Xiao, H. Zhang, W. Chen, Y. Wang, and Y. Jin, "Mcapsnet: Capsule network for text with multi-task learning," in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 4565–4574.

[30] F. Beşer, M.A. Kizrak, B. Bolat, and T. Yildirim, "Recognition of sign language using capsule networks," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2018, pp. 1–4.

[31] A.D. Kumar, "Novel deep learning model for traffic sign detection using capsule networks," *arXiv preprint arXiv:1805.04424*, 2018.

[32] B. Janakiramaiah, G. Kalyani, A. Karuna, L.N. Prasad, and M. Krishna, "Military object detection in defense using multi-level capsule networks," *Soft Comput.*, vol. 27, p. 1045–1059, 2023, doi: 10.1007/s00500-021-05912-0.

[33] P.-A. Andersen, "Deep reinforcement learning using capsules in advanced game environments," *arXiv preprint arXiv: 1801.09597*, 2018.

[34] T. Molnar and E. Culurciello, "Capsule network performance with autonomous navigation," *arXiv preprint arXiv:2002.03181*, 2020.

[35] V. Jayasundara, S. Jayasekara, H. Jayasekara, J. Rajasegaran, S. Seneviratne, and R. Rodrigo, "TextCaps: Handwritten character recognition with very small datasets," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, jan 2019, doi: 10.1109/wacv.2019.00033.

[36] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, dec 2015, pp. 4489–4497, doi: 10.1109/ICCV.2015.510. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCV.2015.510

[37] K. Duarte, Y. S. Rawat, and M. Shah, "Videocapsulenet: A simplified network for action detection," 2018.

[38] D. Ma and X. Wu, "Capsulerrt: Relationships-aware regression tracking via capsules," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 10 948–10 957.

[39] T. Vijayakumar, "Comparative study of capsule neural network in various applications," *J. Artif. Intell.*, vol. 1, no. 01, pp. 19–27, 2019.

[40] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[41] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: an extension of MNIST to handwritten letters," *CoRR*, vol. abs/1702.05373, 2017. [Online]. Available: http://arxiv.org/abs/1702.05373