Evaluating Backbone Modifications on Capsule Networks for Low-Resolution Image Classification

Hasindu Dewasurendra
Department of Information and Communication Engineering
Chungbuk National University
Chungju 28644, South Korea
hasindukd@gmail.com

Taejoon Kim
Department of Information and Communication Engineering
Chungbuk National University
Chungju 28644, South Korea
ktjcc@chungbuk.ac.kr

***Abstract*—Practically constrained applications like intelligent traffic perception and medical imaging often produce very low-resolution (VLR) images, making the classification of such images a crucial, yet challenging task. State-of-the-art (SOTA) classification networks tend to struggle with VLR images due to the limited region of interest (ROI) and lack of distinguishable features. Compared to convolutional neural networks (CNN), capsule networks (CapsNets) that use pose information for classification have proved more robust against affine transformations and adversarial attacks, showing promise in generalizing to VLR image classification. Existing research on CapsNets has favored the development of the routing algorithm while overlooking any modifications to the backbone. We explore a range of architectures including CNNs, transformers, mixers, and hybrids as potential replacements for the conventional backbone. We evaluate these configurations using the dynamic routing algorithm on VLR CIFAR-10 data. Our findings reveal that simple changes to the backbone yield significant improvements: enhancing the performance of the baseline CapsNet by up to 4.48% while using 32% fewer parameters.**

***Index Terms*—capsule networks, backbone, image classification, low-resolution, transformers**

Fig. 1. The original 32×32 High-resolution (HR) samples (top) and the corresponding very low-resolution (VLR) samples (bottom) from the CIFAR-10 [1] dataset.

I. Introduction

Modern deep learning frameworks approach image classification by training powerful networks that extract and learn the spatial features of an image. These include edges, textures, and shapes that help the model differentiate between classes based on their representative features. VLR images with an 8×8 region of interest (ROI) inherently lack these features as shown in Fig. 1, causing a significant decline in the performance of many state-of-the-art (SOTA) models. Despite their practical applications in long-distance surveillance, satellite imagery, remote sensing, and medical imaging, the challenge of VLR image classification remains largely unaddressed [2], [3].

Before the recent adoption of vision transformers, convolutional neural networks (CNN) were the established benchmark in image classification. The success of CNNs is in part due to the pooling layers, which filter spatial information while preserving the necessary features for classification. In addition to enhancing the receptive field, pooling helped CNNs achieve translation invariance by shifting the key features towards the center of feature maps as the network deepened. However, this method comes at the expense of losing exact spatial location information, preventing the network from capturing the spatial hierarchies of objects. For example, CNN might misclassify a truck as a car from a different viewpoint, since the loss of positional information such as the distance between the wheels hinders its ability to differentiate.

Inspired by inverse graphics, capsule networks (CapsNets) were introduced to retain pose information in its fundamental unit, the capsule [4]. A capsule is a group of neurons that learns to encode both: the features and pose of a detected object. This allowed CapsNets to preserve the spatial structures and hierarchies from a visual scene, demonstrating superior robustness against affine and adversarial transformations. CapsNets are particularly appealing for VLR classification tasks since image resolution does not affect the pose information of objects.

II. Background

Most existing research on CapsNets has focused on the routing algorithm; while this is well-founded, previous studies suggest that using a suitable and powerful backbone considerably enhances both the performance and parameter efficiency of CapsNets. Phaye et al. [5] replaced the standard convolutional backbone with dense convolutions leading to better results in complex datasets. The efficient CapsNet [6] leveraged depthwise convolutions as a parameter-efficient alternative in the backbone. Furthermore, Vu et al. [7] achieved SOTA results on various vision tasks by using pre-trained backbones. Recent works have also explored residual CNNs [8], [9], and mixers [10] as backbones to boost performance in vision tasks.

The primary role of a backbone in CapsNets is to extract diverse spatial features and optimize the information encoded in the primary capsule layer. Integrating a suitable backbone improves the CapsNet's capacity to process more complex data. Despite advancements in routing mechanisms [11], [12], many algorithms still use the underlying concept of routing by agreement, where capsules are clustered towards abstract class representations, as implemented in the traditional dynamic routing algorithm [4]. Therefore, we adopt the dynamic routing algorithm as a representative method to evaluate different backbone setups and analyze their effectiveness on VLR CIFAR-10 images.

III. Methodology

We start with the baseline dynamic routing CapsNet (DR-CapsNet) from [4], adjusted for three color channels. DR-CapsNet features a backbone consisting of two convolutional layers with 9×9 kernels and strides of 1 and 2, respectively. With the goal of refining VLR classification performance and reducing model size, we evaluate the effectiveness of a variety of model configurations.

- **Depth-wise convolutions:** The poor performance of CapsNets with complex image data is partly attributed to the limited representation of primary capsules, i.e., the capsules cannot encode all the detected entities. Depth-wise convolutions might form better discrete capsule vectors, that collectively encode more information. We retain the first convolution layer from [4] and modify the second layer to use depth-wise convolutions with similar parameters.
- **ResNet-32 backbone:** While the DR-CapsNet was designed for simpler datasets like MNIST [13], a more powerful backbone may be better suited for handling the complex spatial structures in the CIFAR-10 dataset. This test replaces the baseline backbone with a ResNet-32 [14], excluding its final layers.
- **DenseNet-BC-100 backbone:** This configuration evaluates a DenseNet-BC-100 backbone [15], used with a growth rate of 12 and a compression factor of 0.5. The output feature maps are processed by a 3×3 convolutional layer with a stride and padding of 1 to adjust the number of feature maps before reshaping them into the primary capsules. This ensures that the number of capsules formed remains consistent with those in other methods.
- **2D convolutional patches:** Inspired by the vision transformer (ViT) [16], this approach utilizes two convolutional layers with a kernel size and stride of 2 to extract non-overlapping patches. The first convolutional layer consists of 64 filters, followed by the second layer with 16 filters. Each patch corresponds to a 4×4 receptive field, which is subsequently encoded into a high-dimensional (depth-16) primary capsule. This method aims to enhance the CapsNet's ability to represent complex backgrounds, potentially improving the overall performance.
- **ViT style embeddings:** To explore the impact of patch embeddings used in a ViT, we extract 4×4 patches, flatten, and project them to a depth of 16 using a linear layer. This way, a patch is encoded as a single primary capsule with a depth of 16. We rely on the dynamic routing algorithm to organize and route these embeddings.
- **ViT backbone:** This method adopts the ViT architecture, excluding the MLP head, to replace the conventional backbone. Specifically, the ViT backbone comprises six multi-head attention (MHA) layers with each MHA layer featuring eight attention heads. The tokens generated by the ViT are used as the primary capsules of the CapsNet. The attention mechanism captures long-distance dependencies. Therefore, we expect these tokens to contain more semantic information, potentially improving performance on high-dimensional VLR CIFAR-10 data.
- **ConvMixer backbone:** The ConvMixer [17] has proven to be a conceptually simple yet powerful network, outperforming the ViTs on image classification benchmarks. ConvMixer uses depthwise and pointwise convolutions to mix information spatially and channel-wise, respectively. For our study, we employ a ConvMixer with a depth of 8, 256 filters, a kernel size of 5, and a patch size of 4. Given its effectiveness as a backbone in capsule autoencoders [10], we evaluate its performance on VLR image classification. Furthermore, we investigate another approach by replacing the depthwise and pointwise convolutions with standard 2D convolutions for mixing. Since capsules are high-dimensional vector representations of features, it is consistent to use entire vectors during mixing, rather than discretely as in the original ConvMixer.
- **Combined residual and mixer backbone:** This configuration integrates a simple residual layer with three residual blocks—each containing 64 channels and strides of 1, 2, and 2, respectively—for initial feature extraction, followed by a ConvMixer. The residual layer efficiently generates feature maps, while the ConvMixer mixes the extracted information to capture long-distance dependencies. The mixed features are then transformed into the primary capsule layer. This hybrid approach leverages the strengths of both architectures to enhance performance.

TABLE I
THE COMMON HYPERPARAMETERS USED IN TRAINING THE CAPSNETS ON VLR CIFAR-10 DATA.

Parameter	Value
Optimizer	Adam [18]
Initial learning rate	0.001
Weight decay	0.95
Batch size	100
Epochs	100
Routing iterations	3
Weight of margin loss	1
Weight of reconstruction loss	0.0005

IV. Experiments

The training hyperparameters for the CapsNets are summarized in Table I. Note that the squash activation function from the DR-CapsNet was unchanged. The CNNs in Table II are trained using an Adam optimizer with an initial learning rate of 0.0001 except for VGG-19 [19] which uses an SGD

TABLE II

CLASSIFICATION RESULTS FOR VARIOUS BACKBONE MODIFICATIONS ON VLR CIFAR-10. THE CAPSULE PARAMETERS, TOP-1 ACCURACY, AND PARAMETER COUNTS ARE RECORDED FOR EACH METHOD. METHODS WITH FEWER PRIMARY CAPSULES ARE ASSIGNED A HIGHER CAPSULE DEPTH FOR BETTER PERFORMANCE.

	Backbone model	Primary caps # Caps	Primary caps Depth	Class caps depth	Params (M)	Test Acc.
CNNs	VGG-19* [19]	-	-	-	21.07	72.65
	DenseNet-201* [15]	-	-	-	20.82	68.31
	EfficientNet-B7* [20]	-	-	-	67	71.05
Conv Backbones	DR-CapsNet [4]	2048	8	16	11.75	68.05
	Depth-wise conv	2048	8	16	6.46	65.30
	ResNet-32	128	8	16	4.39	69.35
	DenseNet-BC-100	2048	8	16	7.94	**72.53**
Transformer Backbones	2D conv patches	64	16	32	**4.17**	60.16
	ViT style embeddings	64	16	32	**4.17**	54.31
	ViT	64	16	32	4.22	62.97
Mixer Backbones	ConvMixer	2048	8	16	6.99	71.05
	ConvMixer with 2D conv	512	8	16	5.24	68.35
Hybrid Backbone	Combined residual and mixer	2048	8	16	7.12	70.82

* denotes transfer learned.

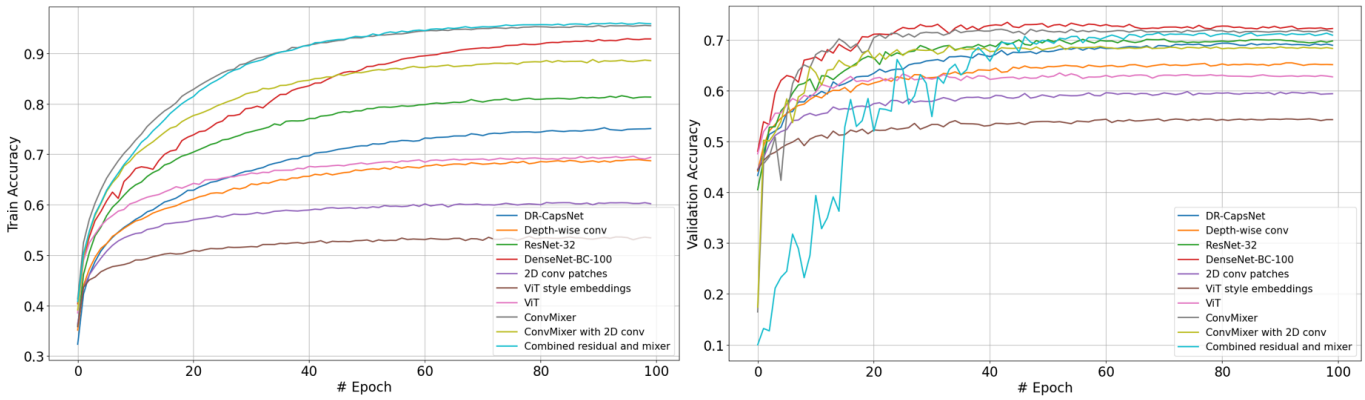

Fig. 2. The convergence plots for the tested backbone configurations illustrate both, the training and validation performance, over 100 epochs on the VLR CIFAR-10 dataset. Note that the validation accuracy is computed from a subset of the training data and differs from the test accuracy reported in Table II.

optimizer with a learning rate of 0.001 and a momentum of 0.9. Pre-trained weights from ImageNet-1k [21] were assigned at the start of training for the CNNs. An input image resolution of 48×48 was used with CNNs since they do not function well with smaller images. The top-1 classification accuracy is reported in Table II along with the parameter count for each method. All experiments are conducted using an NVIDIA RTX-4090 GPU and each CapsNet was trained from scratch for fair testing.

A. Dataset and Augmentation

CIFAR-10 dataset consist of 32×32 natural color images belonging to 10 classes. This dataset was chosen since it represents a more practical and complex real-world data distribution. During training and testing, we down-sample the images to 8×8 before up-sampling back to 32×32 (48×48 for CNNs) using bilinear interpolation to generate VLR images. We also apply data augmentations in the form of translations, random rotations in the range of 10 degrees, and random horizontal flips.

B. Classification Results

The tested SOTA CNNs show significant performance degradation on VLR images compared to their HR benchmarks [19], [15], [20]. In contrast, the tested CapsNets are more parameter efficient and demonstrate competitive or greater performance against these complex CNNs, with the baseline DR-CapsNet achieving a classification accuracy of 68.05%. Among the evaluated backbones, DenseNet-BC-100 displays the highest accuracy at 72.53% while being 32.4% more parameter efficient compared to the DR-CapsNet. In terms of parameter efficiency, using 2D convolutional patches stands out with just 4.17M parameters, 64.5% fewer than the baseline. However, this efficiency is achieved at the cost of a notable 7.89% drop in accuracy.

Between the convolutional backbones, ResNet-32 yields an accuracy of 69.35% with a significantly lower parameter count

of just 4.39M, while using depth-wise convolutions halve the parameter count with a 2.75% drop in performance over the DR-CapsNet. From a high-level outlook, the mixer-type backbones enhance performance with fewer parameters, whereas the transformer-style backbones underperform, despite having a greater capsule depth in both the primary and class capsule layers. The best result with a transformer-type backbone was obtained using a ViT itself, which only reached 62.97%. Among the mixer-type configurations, both approaches improve the accuracy with a notable reduction in parameters. Particularly, the ConvMixer achieves a performance of 71.05% with just 6.99M parameters. Lastly, the hybrid backbone modification records an accuracy of 70.82%, slightly below the ConvMixer, with a marginally higher parameter cost.

In summary, we have presented a comparative analysis of various backbone architectures, highlighting their potential to improve classification accuracy at a lower cost of parameters. All models were trained for 100 epochs rather than until full convergence; however, based on Fig. 2, it is reasonable to assume that most models had sufficiently converged. Nevertheless, several modifications have outperformed the DR-CapsNet, emphasizing the impact of a good backbone choice in CapsNets.

V. Conclusion and Future Works

In conclusion, our study highlights the role of a well-designed backbone in a CapsNet to achieve SOTA performance in VLR image classification. We explore the impacts of several popular architectures as backbones used in conjunction with the dynamic routing algorithm. The findings demonstrate that selecting a compatible backbone significantly enhances classification performance while reducing the parameter complexity, both crucial for surpassing existing benchmarks.

The dynamic routing algorithm performs a form of unsupervised clustering, but more advanced approaches like attention mechanisms, have already been integrated into CapsNets [8], [22]. While this work focuses solely on the backbone, combining novel routing mechanisms with suitable backbones may certainly improve the CapsNets' ability to handle more complex data, presenting a promising direction for future research.

Acknowledgment

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant RS-2023-00244014.

References

[1] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[2] L. Zhang, R. Dong, S. Yuan, W. Li, J. Zheng, and H. Fu, "Making low-resolution satellite images reborn: a deep learning approach for super-resolution building extraction," *Remote Sensing*, vol. 13, no. 15, p. 2872, 2021.

[3] V. Thambawita, I. Strümke, S. A. Hicks, P. Halvorsen, S. Parasa, and M. A. Riegler, "Impact of image resolution on deep learning performance in endoscopy image classification: An experimental study using a large dataset of endoscopic images," *Diagnostics*, vol. 11, no. 12, p. 2183, 2021.

[4] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in neural information processing systems*, vol. 30, 2017.

[5] S. S. R. Phaye, A. Sikka, A. Dhall, and D. Bathula, "Dense and diverse capsule networks: Making the capsules learn better," *arXiv preprint arXiv:1805.04001*, 2018.

[6] V. Mazzia, F. Salvetti, and M. Chiaberge, "Efficient-capsnet: Capsule network with self-attention routing," *Scientific reports*, vol. 11, no. 1, p. 14634, 2021.

[7] D. T. Vu, L. B. T. An, J. Y. Kim, and G. H. Yu, "Towards feasible capsule network for vision tasks," *Applied Sciences*, vol. 13, no. 18, p. 10339, 2023.

[8] Y.-H. H. Tsai, N. Srivastava, H. Goh, and R. Salakhutdinov, "Capsules with inverted dot-product attention routing," *arXiv preprint arXiv:2002.04764*, 2020.

[9] T. Hahn, M. Pyeon, and G. Kim, "Self-routing capsule networks," *Advances in neural information processing systems*, vol. 32, 2019.

[10] M. Everett, M. Zhong, and G. Leontidis, "Masked capsule autoencoders," *arXiv preprint arXiv:2403.04724*, 2024.

[11] H. Dewasurendra and T. Kim, "Deep hybrid architecture for very low-resolution image classification using capsule attention," *IEEE Access*, 2024.

[12] Z. Zhao and S. Cheng, "Capsule networks with non-iterative cluster routing," *Neural Networks*, vol. 143, pp. 690–697, 2021.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[16] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[17] A. Trockman and J. Z. Kolter, "Patches are all you need?," *arXiv preprint arXiv:2201.09792*, 2022.

[18] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] K. Simonyan, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[20] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[22] J. Choi, H. Seo, S. Im, and M. Kang, "Attention routing between capsules," in *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pp. 0–0, 2019.