ME-CapsNet: A Multi-Enhanced Capsule Networks with Routing Mechanism
Jerrin Bright†, Suryaprakash Rajkumar† and Arockia Selvakumar Arockia Doss*†
School of Mechanical Engineering, Vellore Institute of Technology, Chennai, India
Email: jerriebright@gmail.com, suryaprakash.rajkumar@gmail.com, arockiaselvakumar@vit.ac.in
†These Authors Contributed Equally To This Work

Abstract—Convolutional Neural Networks need the construction of informative features, which are determined by channel-wise and spatial-wise information at the network’s layers. In this research, we focus on introducing a novel solution that uses sophisticated optimization to improve both the spatial and channel components within the receptive field of each layer. Capsule networks were used to understand the spatial association between features on the feature map. Standalone capsule networks have shown good results in comparatively simple datasets than in complex datasets as a result of the inordinate amount of feature information. Thus, to tackle this issue, we have proposed ME-CapsNet by introducing deeper convolutional layers to extract important features before passing through modules of capsule layers strategically to improve the performance of the network significantly. The deeper convolutional layer includes blocks of Squeeze-Excitation networks which use a stochastic sampling approach for progressively reducing the spatial size, thereby dynamically recalibrating the channels by reconstructing their interdependencies without much loss of important feature information. Extensive experimentation was done using commonly used datasets, demonstrating the efficiency of the proposed ME-CapsNet, which clearly outperforms various research works by achieving higher accuracy with minimal model complexity in complex datasets.

Index Terms—Deep Learning, Capsule Network, Squeeze-Excitation, Stochastic Pooling

I. INTRODUCTION

Convolutional Neural Network (CNN) is a cutting-edge image classification [1]–[3] approach that excels in detecting characteristics but struggles to grasp the spatial connection between them, such as their direction, perspective, and size. For scene understanding or any vision-based strategy, improving categorization accuracy is critical. In addition, transmitting information from one layer to the next (routing) is less efficient in CNNs. This may be improved by identifying important characteristics using a variety of representations and building well-organized routing strategies for transferring data between layers. Investigation of many elements of network architecture (spatial, channel) is now a very prominent research subject in CNN, and it may greatly improve the network’s understanding ability [4].

Capsule networks (CapsNet) were introduced by Sabour et al. [5] to solve common problems with CNNs (rotational invariance and failure to capture hierarchical spatial information) by retrieving the most important information from the features and the spatial relationship efficiently. Instead of pooling operations, dynamic routing was adapted to prevent losing important features. Also, capsules where used instead of neurons thereby passing the spatial relationship from one layer to next efficiently. However, CapsNet tends to fail as a result of more feature information in complex datasets. Thus, adding of deeper convolutional layers to the CapsNet in a strategic way is one approach to solve this issue.

Thus, we propose a novel network called Multi-Enhanced Capsule Networks (ME-CapsNet) which finds robust features, obtaining good channel and spatial relationships between layers. The squeeze-excitation network or SENets [6] with very little computation difference models the interconnection of channels by adaptively recalibrating the channel features. Once the important features are learned, the likelihood and the properties of each and every feature are estimated by passing it to the CapsNet module, which inturn uses EM Routing to prevent feature losses. EM Routing [7] is used to group capsules with the parent capsule using the proximity of the corresponding estimated votes on the pose matrix. Furthermore, reconstruction is included in the proposed work, unlike the originally proposed CapsNet with EM Routing.

Some contributions of this research work are mentioned here as follows:

- A novel network which multi enhances the features, that is, both spatial and channel-wise information at each layer
with deep convolutional layers (SE Nets) and CapsNet:
- Improvements in the Squeeze phase of the originally proposed SENets by using Stochastic Spatial Sampling Pooling (S3P) to reduce feature information losses, computation overload and training time when compared to the originally used average pooling operation.
- Exploring the capabilities of the proposed system by fusing it with several state-of-the-art base models including ResNet, VGG, Inception, and MobileNet and analyzing the performance of the proposed network by calibrating with different parameters.

The rest of the research paper is systematized as follows. Section II consists of related research work related to various networks related to SENets and CapsNets followed by Section III detailing the proposed architecture and its modules. Then in Section IV, various experiments are performed testing by tweaking parameters and in Section V the proposed architecture is tested with various state-of-the-art datasets and networks. Also, fewer predominantly efficient networks were handpicked and tested alongside our network. Finally, in Section VI the proposed architecture is summarized with its advantages and is followed by future work aimed at increasing the robustness of the currently proposed system.

II. RELATED WORKS

Since the introduction of deep convolutional networks [8] significant improvement in the deep learning regime has been witnessed. Layer stacking as a result of deep networks significantly improved performance but also introduced vanishing gradient problems [9]. Also, using operating layers like max pooling introduced loss of spatial information. CapsNet attempts to solve these problems by nesting layers rather than stacking them. These recent techniques in deep learning have added fuel to research in this field drastically.

CapsNet are considered a great leap into neural networks considering its significance and scope of improvement over the traditional techniques. Some efficient research works on CapsNet are explained here. Mazzia et al. [10] proposed a non-iterative routing technique that exploits self-attention mechanisms. Deliege et al. [11] introduced a counterpart for DigitCaps layer using a custom Hit-or-Miss layer. Huang et al. [12] proposed addition of two attention layers to the original CapsNet, to improve the overall performance of the network.

SE Nets are predominantly used recently considering the significant improvement in performance with minimal amount of computation rise. A lot of customization to SENets have also been done, including Yue Cao et al. [13] performing a unification of NLNet and SENet resulting in a Global Context (GC) network. A lot of SENets applications has been established including some works by Qiu et al. [14], Shunjun Wei et al. [15], and Han et al. [16] where fusion of SENets were done with various architectures of CNN for fish image classifications, PRI modulation recognition, sea ice image sensing respectively.

Yang et al. [17] proposes an approach to improve CapsNet feature extraction capability using a modified version of ResNet called Res2Net and SENet in the base layers to enhance CapsNet performance. This is the closest work relevant to our work. We have strategically employed the SE blocks in between various layers and used a more powerful variant of pooling to improve the performance of the squeeze operation.

III. METHODOLOGY

In this section, the different components equipped by our work to complete the robust architecture are discussed extensively. Our architecture predominantly has three blocks to multi-enhance the feature relationship of CNN: the SE Primary Capsule (SE-PC) block, SE Classification Capsule (SE-CC) block, and the Mask by Layer (MbL) block. That is, an input image in the form of a tensor ($U$) is fed to our network, which is first converted to a tensor upon transformation ($U_{tr}$). Then, $U_{tr}$ is sent through the SE-PC block where the squeeze and excitation operation is performed, resulting in $U_{o tr}$, which has recalibrated weights based on channel dependence. $U_{o tr}$ is then passed through the primary capsule layer. The Squeeze and Excitation operation is done three times continuously to obtain the improved calibrated weights for the corresponding channels. The squeeze operation is done using S3P [18] which will be explained in Section III-A. Then the output of the primary layer of the SE-PC is fed as input to the SE-CC wherein the features are highlighted using SENets first and then sent to the classification module. Then, we designed the MbL block inspired by the works for Huang et al. [19] for reconstructing the output capsules from the probability vectors obtained from the SE-CC block. Advantages of including the reconstruction block include regularization and ability to regenerate data.
A. Squeeze-Excitation Network

SE Nets directly influence the performance of the network and its layers through adaptive adjusting of the weights obtained from each and every feature map in the channels of the convolutional block by adding supplementary parameters. This interfacing with weights radically increases the sensitivity to beneficial features in the map which will be exploited in succeeding transformations.

![Image](image.png)

**Fig. 3. Schema of the input (SE-ResNet) architecture for Capsule Network**

Generally, in the squeeze phase, Global Average Pooling (GAP) or Maximum Pooling (GMP) techniques are equipped, each has its limitations and advantages. GAP (represented as $z_{avg}$ in equation 1) results in smooth average pixels and does not preserve pixel information. GAP takes into consideration all the features, thus it can result in a biased feature map.

$$z_{avg} = \frac{1}{H \cdot W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j)$$

(1)

GMP (represented as $z_{max}$ in equation 2) extracts low-level features (edges, textures) better than GAP but captures a lot of noise and is independent of neighboring pixels.

$$z_{max} = max_{i,j} \cdot u_c(i, j)$$

(2)

In this work, we will use a pooling approach proposed by Zhai et al. [18]. A two-step process includes GMP ($z_{max}$) and stochastic down sampling ($D_{s3p}^g(.)$) is done on the feature map to learn better abstract representations in a larger receptive field. Considering it is stochastic, different feature map is obtained for the same training sample. This means that it performs a virtual data augmentation [20]. Equation 3 defines the S3P pooling, where $s$ represents strides and $g$ represents the grid size.

$$z_{s3p} = D_{s3p}^g \cdot (z_{max})$$

(3)

Now, the output of the squeeze phase ($z_{s3p}$) is passed to the excitation phase for adaptive scaling of weights in the respective channel. Excitation is done using the Multi-Layer Perceptron (MLP) approach. That is, two fully connected layers are used with weights $W_1$ and $W_2$. The first fully connected layer is connected to ReLU activation [21] followed by another fully connected layer, which in turn is connected to the Sigmoid activation function. This can be represented as shown in equation 4. While training, $W_1$ and $W_2$ are learned.

$$s = \sigma \cdot (W_2 \cdot \delta(W_1 \cdot z_{s3p}))$$

(4)

In other words, channelwise dependence is found inside the excitation phase by learning the weights $W_1$ and $W_2$ using a compression ratio to enable efficient learning.

Figure 3 shows the SENet schema with the Residual Network (ResNet) backbone used in this work. In other words, the output of figure 3 is the input of the primary capsule layer. A very similar architecture without ResNet, leveraging the output of capsule layers are used for the SENets equipped before the classification capsule layer.

B. Capsule Network

Appropriate activation can be given if the likeliness of the feature and the feature information is passed via each and every neuron. Thus, these neurons are termed capsules, and they output a vector (called an activity vector) instead of information about individual features. Knowing the likeliness of a feature, we can easily extrapolate different variants of features in a capsule, thereby reducing the training data. The product of the pose matrix ($M_i$) represented by a Gaussian distribution with transformation ($W_{ij}$) is calculated to calculate the vote ($v_{ij}$).

$$v_{ij} = M_i \cdot W_{ij}$$

(5)

In equation 5, $i$ corresponds to the parent capsule and $j$ to the current capsule. The transformation is learned through back-propagation and a cost function.

C. Routing Mechanism

Transferring information from one layer to another is called Routing. ReLU is used as the routing mechanism in fully convolutional neural networks.

$$y_i = ReLU(\sum_j W_{ji} \cdot x_j + b_i)$$

(6)

Instead of ReLU, a squashing function is used to convert small vector magnitudes to zero and big vector magnitudes to unit vectors. Grouping of capsules using the EM clustering technique is called EM Routing. Parent capsules are predicted in the pose matrix in terms of ‘votes’. It is based on proximity, so even when the viewpoints change, the transformation matrix remains constant. Assignment probabilities ($r_{ij}$) estimate the connection at run time with respect to the parent and other capsules. The cost of all low-level capsules is shown in equation 7.

$$J^p_j = (ln(\sigma^p_j) + k) \sum_i r_{ij}$$

(7)
Activation \((A_j)\) in a particular capsule can be estimated using the inverse temperature parameter \((\lambda)\) and parameters iteratively estimated using EM Routing by equation 8.

\[
A_j = \text{sigmoid}(\lambda(b_j - \sum_h J_j^h)) \tag{8}
\]

The data points fit into a Gaussian model and the steps E and M are triggered alternatively iterating for ‘t’ steps. The Gaussian model is updated using the M step based on the initialized \(r_{ij}\) (distributed uniformly), which will then be reshaped to form the pose matrix of the parent capsule. The \(r_{ij}\) is recomputed using the E-step for every data point.

Upon the completion of EM Routing, the capsules with best activation is selected to make the appropriate predictions using the MLP approach used in the excitation phase of the SE Blocks of ME-CapsNet.

D. Reconstruction

The output of the classification capsule is used to reconstruct the input image by passing through a decoder. The output is a vector that consists of the probability vector squashed to get values between 0 and 1. The idea behind the used non-linear squashing function is to compress small vectors to zero and long vectors to one. The decoder network consists of two fully connected layers with ReLU activation and a sigmoid function. Mean Square Error (MSE) is used to estimate the reconstruction loss between the true \((y)\) and predicted \((y^0)\) labels as shown in equation 9.

\[
L(y, y^0) = \frac{1}{N} \sum_{i=0}^{N} (y - y^0) \tag{9}
\]

IV. Experimentation

In this section, experiments are steered from the architecture proposed in the previous section using MNIST, FashionMNIST and CIFAR-10 datasets, and the performance is evaluated with respect to other networks, previous effective research works, and parameter tuning.

Implementation of ME-CapsNet is done using several tuned parameters for efficient training, and some important parameters are mentioned below. We have built ME-CapsNet using the Keras framework and tensorflow. Tensorboard visualization toolkit was used for analyzing the performance of the proposed network. The network was compiled using the NVIDIA GTX 1650 GPU. The dataset was publicly available for model training. For the training procedure, 20 epochs were done uniformly for all datasets, a batch size of 128, Adam optimizer [22], squashing constant of 0.5 and decay factor of 0.9.

The section has been split into three on the basis of the target calibration. In section IV-B, the parameters of SENet is tweaked and the resulting top-1% and top-10% error rates are obtained. This is followed by section IV-A, where the capsule parameters are calibrated using various dimensions and quantity.

A. Capsule parameters

The number of capsules affects the overall accuracy and performance of the proposed ME-CapsNet. In table I, different capsule dimensions are selected and the influence of capsule dimension on the overall accuracy of the network was tested. We have got the best accuracy when testing ME-CapsNet with dimension of capsule set to be 8. This testing has been done using CIFAR10 dataset.

| Dimension | Accuracy  |
|-----------|-----------|
| 8         | 89.84%    |
| 12        | 89.12%    |
| 16        | 88.57%    |

In Table II, we tested our architecture with three variations in the total number of capsules and the respective accuracy corresponding to it has been recorded. The best results were obtained when ME-CapsNet was tested with 5 capsules. This testing was carried out using the CIFAR10 dataset and the dimension of the capsule set to be 8.

| Number | Accuracy  |
|--------|-----------|
| 5      | 88.79%    |
| 8      | 89.84%    |
| 10     | 86.44%    |

B. SE Network parameters

As discussed in previous sections, SENets are split into Squeeze and Excitation operation. The squeeze operation consists of a pooling technique which includes max pooling, average pooling, and the proposed S3P. All the pooling techniques are extensively studied in section III-A. Similarly, excitation operation is predominantly carried by the activation function like Sigmoid, ReLU, LeakyReLU and TanH.

|                  | top-1% | top-5% |
|------------------|--------|--------|
| Squeeze - Max    | 22.57  | 6.09   |
| Squeeze - Avg    | 22.28  | 6.03   |
| Squeeze - S3P (Ours) | **21.78** | **5.83** |
| Excitation - Sigmoid | 22.28  | 6.03   |
| Excitation - ReLU | 23.47  | 6.98   |
| Excitation - LeakyReLU | 23.22  | 6.91   |
| Excitation - TanH | 23.00  | 6.38   |

Sigmoid is an activation function that adds nonlinearity and ranges between 0 to 1. Similarly, TanH is an activation function...
function very similar to the properties of Sigmoid, but it ranges from -1 to 1. ReLU is a linear function that outputs the input if positive and 0 if negative. Meanwhile, LeakyReLU works similar to ReLU for positive inputs, but outputs a very small positive value when the input tends to be negative.

V. RESULTS

In this section, ME-CapsNet equipped with the best parameters obtained from the experimentation done in Section IV is compared alongside different state-of-the-art backbone models (Section V-B) and novel research works related to CapsNet (Section V-C). The details of the datasets used to test the proposed architecture are also explained in V-A.

A. Datasets

Some commonly used datasets used to experiment the performance of ME-CapsNet include MNIST [23], FashionMNIST [24], CIFAR10 [25] and KMNIST [26]. The performance of ME-CapsNet when tested alongside these datasets can be visualized in 4

MNIST and FashionMNIST both have 10 classes with 70,000 images of size $28 \times 28 \times 1$. CIFAR10 has 10 classes with 60,000 images of size $32 \times 32 \times 3$. KMNIST or Kuzushiji-MNIST dataset has 10 classes, is a drop-in replacement of MNIST and FashionMNIST, consisting of 70,000 images of size $28 \times 28 \times 1$ just like those datasets. For MNIST, KMNIST and FashionMNIST datasets, 50,000 images were used for training and 10,000 images for testing. For the CIFAR10 dataset, 60,000 images were used for training and 10,000 for testing.

B. Comparing various backbones

In this subsection, various backbone architectures including residual network with various levels of layers, inception network and VGG network is fused as input layer for ME-CapsNet and the respective test accuracy is logged and compared systematically for four different datasets.

| TABLE IV | COMPARISON OF VARIOUS BACKBONE ARCHITECTURES |
|----------|---------------------------------------------|
|          | MNIST | FashionMNIST | CIFAR10 | KMNIST |
| ResNet-50 | 99.57  | 94.85        | 88.87   | 96.75   |
| ResNet-101| 99.83  | 95.63        | 89.37   | 97.84   |
| ResNet-152| 99.78  | 95.91        | 89.84   | 97.11   |
| VGG-16    | 98.54  | 95.11        | 87.72   | 96.49   |
| Inception | 98.63  | 95.38        | 88.98   | 97.34   |

In this subsection, some state-of-the-art research work aligning to CapsNet is studied in terms of accuracy, number of parameters, and the ability to reconstruct. Table V clearly shows our network, ME-CapsNet outperforms the other research works significantly with a comparatively decent number of parameters using the CIFAR10 dataset.

| TABLE V | COMPARISON WITH VARIOUS PROPOSED NETWORK. |
|---------|-----------------------------------------|
| Accuracy | # of Params | Recon. |
| HitNet [11] | 73.30% | 8.89M | Yes |
| MS-CapsNet [29] | 75.70% | 11.20M | Yes |
| CapsNet Baseline | 79.24% | 11.98M | No |
| DeeperCaps [30] | 81.29% | 5.81M | Yes |
| DCNet [31] | 82.63% | 11.88M | Yes |
| DA-CapsNet [12] | 85.47% | 7M | Yes |
| Cv-CapsNet++ [32] | 86.70% | 2.69M | No |
| AR-CapsNet [33] | 88.94% | 9.60M | Yes |
| Sabour et. al. [5] | 89.40% | 14.36M | Yes |
| DCNet++ [31] | 89.71% | 13.4M | Yes |
| ME-CapsNet (Ours) | 89.84% | 7.24M | Yes |

ME-CapsNet easily outperforms CapsNet research work in CIFAR10 datasets, demonstrating its ability to capture important features and route without feature loss. Also, 1.23% better performance in terms of accuracy was estimated using our model with FashionMNIST dataset. Although the difference in precision between the proposed approach and the previous best
approach is only 0.13%, computationally our approach tends to be much cheaper. This is because of the less computational increase when using deep convolutional layers via SENets in between the CapsNet layers strategically with well-defined parameters.

VI. CONCLUSION AND FUTURE WORK

In this research, we have proposed a novel deep learning architecture that simultaneously enhances the spatial and channel-wise relationship of features with minimum computation. This multi-enhancement-based network has been coined ME-CapsNet which uses a stochastic pooling approach to strategically improve the performance of the SENets used in CapsNet. The originally proposed CapsNet was used as the base network and SENets have been fused strategically before and in between capsule layers. The squeeze operation proposed in the original SENets where altered in this work using S3P, a stochastic sampling-based pooling approach which in turn significantly improved the performance of the SENets used in the proposed ME-CapsNet. The architecture was evaluated with state-of-the-art datasets and comparatively better results in terms of accuracy and computational cost were obtained upon testing with the proposed model.

Future studies will focus on evaluating ME-CapsNet on broad and complex datasets. Though the aim of the originally proposed CapsNet was to prevent the use of pooling operations to ward off loss of information, ME-CapsNet, like many other networks, uses the advantages of pooling operators to enhance the performance of CapsNets especially for complex datasets. One right research direction to proceed from here would be to look for alternatives of pooling operators to increase performance, especially for complex datasets. In addition, attention to the routing mechanism can be added to our network to improve its performance. Computational time can be further reduced by replacing the fully connected layers with capsule layers.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Commun. ACM, vol. 60, p. 84–90, may 2017.

[2] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in Computer Vision – ECCV 2014 (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 818–833, Springer International Publishing, 2014.

[3] R. Mukhotietianov and J. Carrillo, “Capsnet comparative performance evaluation for image classification,” CoRR, vol. abs/1805.11195, 2018.

[4] X. Jiang, Y. Wang, W. Liu, S. Li, and J. Liu, “Capsnet, cnn, fc6: Comparative performance evaluation for image classification,” International Journal of Machine Learning and Computing, vol. 9, pp. 840–848, 12 2019.

[5] S. Nabour, N. Frosts, and G. E. Hinton, “Dynamic routing between capsules,” CoRR, vol. abs/1710.09829, 2017.

[6] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132–7141, 2018.

[7] G. E. Hinton, S. Nabour, and N. Frosts, “Matrix capsules with EM routing,” in International Conference on Learning Representations, 2018.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Commun. ACM, vol. 60, p. 84–90, may 2017.

[9] G. Ananthi and A. Arockia Selvakumar, Review on Deep Learning Algorithms in Medical Devices, pp. 1–27. Cham: Springer International Publishing, 2020.

[10] V. Mazzia, F. Salvetti, and M. Chiaberge, “Efficient-capsnet: Capsule network with self-attention routing,” CoRR, vol. abs/2101.12491, 2021.

[11] A. Deliège, A. Cioppa, and M. V. Droogenbroeck, “Hinet: a neural network with capsules embedded in a hit-or-miss layer, extended with hybrid data augmentation and ghost capsules,” CoRR, vol. abs/1806.06519, 2018.

[12] W. Huang and F. Zhou, “Da-capsnet: dual attention mechanism capsule network,” Sci Rep, 2020.

[13] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, “Gecnet: Non-local networks meet squeeze-excitation networks and beyond,” in 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 1971–1980, 2019.

[14] C. Qin, S. Zhang, C. Wang, Z. Yu, H. Zheng, and B. Zheng, “Improving transfer learning and squeeze-excitation networks for small-scale fine-grained fish image classification,” IEEE Access, vol. 6, pp. 78503–78512, 2018.

[15] S. Wei, Q. Qu, Y. Wu, M. Wang, and J. Shi, “Pri modulation recognition based on squeeze-and-excitation networks,” IEEE Communications Letters, vol. 24, no. 5, pp. 1047–1051, 2020.

[16] Y. Han, C. Wei, R. Zhou, Z. Hong, Y. Zhang, and S. Yang, “Combining 3d-cnn and squeeze-and-excitation networks for remote sensing sea ice image classification,” Mathematical Problems in Engineering, vol. 2020, pp. 364–375, 4 2020.

[17] S. Yang, F. Lee, R. Miao, J. Cai, L. Chen, W. Yao, K. Kotani, and Q. Chen, “Rs-capsnet: An advanced capsule network,” IEEE Access, vol. 8, pp. 85007–85018, 2020.

[18] S. Zhai, H. Wu, A. Kumar, Y. Cheng, Y. Lu, Z. Zhang, and R. Feris, “S3pool: Pooling with stochastic spatial sampling,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[19] G. Huang, Z. Liu, and K. Q. Weinberger, “Densely connected convolutional networks,” CoRR, vol. abs/1608.06993, 2016.

[20] P. Simard, D. Steinkraus, and J. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” in Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings., pp. 958–963, 2003.

[21] V. Nair and G. E. Hinton, “Rectifier linear units improve restricted boltzmann machines,” in Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, (Madison, WI, USA), p. 807–814, Omnipress, 2010.

[22] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” International Conference for Learning Representations, 2014.

[23] L. Deng, “The mnist database of handwritten digit images for machine learning research,” IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 141–142, 2012.

[24] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” CoRR, vol. abs/1708.07747, 2017.

[25] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Citeseer, Tech. Rep, pp. 32–33, 2009.

[26] T. Chnunwat, M. Bober-Irizar, A. Kitantom, A. Lamb, K. Yamamoto, and D. Ha, “Deep learning for classical japanese literature,” CoRR, vol. abs/1812.01718, 2018.

[27] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv 1409.1556, 09 2014.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” CoRR, vol. abs/1409.4842, 2014.

[29] C. Xiang, L. Zhang, Y. Tang, W. Zou, and C. Xu, “Ms-capsnet: A novel multi-scale capsule network,” IEEE Signal Processing Letters, vol. 25, no. 12, pp. 1850–1854, 2018.

[30] J. Rajasegaran, V. Jayasundara, S. Jayasekara, H. Jayasekara, S. Seneviratne, and R. Rodrigo, “Deepcaps: Going deeper with capsule networks,” CoRR, 2019.

[31] S. R. Phaye, A. Sikka, A. Dhall, and D. R. Bathula, “Dense and diverse capsule networks: Making the capsules learn better,” CoRR, vol. abs/1805.04001, 2018.

[32] Y. Cheng, J. He, J. He, and H. Xu, “Cv-capsnet: Complex-valued capsule network,” IEEE Access, vol. 7, pp. 85492–85499, 2019.

[33] J. Choi, H. Seo, S. Im, and M. Kang, “Attention routing between capsules,” CoRR, vol./abs/1907.01750, 2019.