RCN2: Residual Capsule Network V2

Arjun Narukkanchira Anilkumar
IOT Collaboratory IUPUI
Department of Electrical and Computer Engineering
Purdue School of Engineering and Technology
Indianapolis, USA
arjuna@iupui.edu

Mohamed El Sharkawy
IOT Collaboratory IUPUI
Department of Electrical and Computer Engineering
Purdue School of Engineering and Technology
Indianapolis, USA
melshark@iupui.edu

Abstract **Unlike Convolutional Neural Network (CNN), which works on the shift-invariance in image processing, Capsule Networks can understand hierarchical model relations in depth[1]. This aspect of Capsule Networks let them stand out even when models are enormous in size and have accuracy comparable to the CNNs, which are one-tenth of its size. The capsules in various capsule-based networks were cumbersome due to their intricate algorithm. Recent developments in the field of Capsule Networks have contributed to mitigating this problem. This paper focuses on bringing one of the Capsule Network, Residual Capsule Network (RCN) to a comparable size to modern CNNs and thus restating the importance of Capsule Networks. In this paper, Residual Capsule Network V2 (RCN2) is proposed as an efficient and finer version of RCN with a size of 1.95 M parameters and an accuracy of 85.12% for the CIFAR-10 dataset.**

Keywords ***Convolution Neural Networks, Capsule Network, Residual Capsule Network, ResNet, CIFAR 10, Residual Capsule Network V2***

I. Introduction

Image processing utilizing Convolutional Neural Networks (CNNs) has been prevalent for more than two decades. CNN leverage filtering techniques used in signal processing, along with spatial relationships[2]. Humans attempted mimicking what nature has brought forth through millions of years of evolution within the past couple of decades via CNNs. One such attempt that motivated us is AlexNet, in which the authors tried to emulate the human visual system[1]. From the image of the human visual pathway given in Fig. 1, it can be observed optic nerves from each temporal retina, which has two optic tracts, interchange one of it when they meet at the optic chiasma, and they finally merge at the primary visual cortex[3]. This model of the network of nerves can be compared to the structure of AlexNet[4].

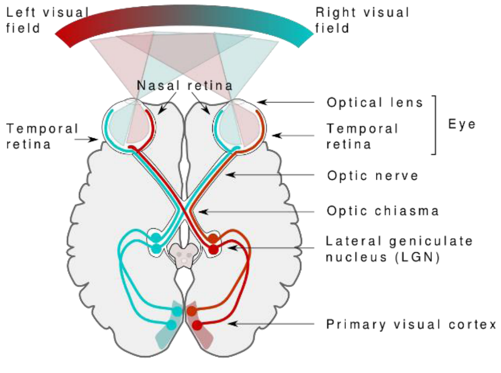


Fig. 1. Human Visual Pathway [5]

Capsule Networks[1] is an assuring machine learning concept awaiting to unfold its full potential. This model is proposed by one of the authors of AlexNet, i.e., Geoffrey Hinton[1], [4]. Capsule Network is another step in imitating the human perception, a capsule for each characteristic of the object, and connections between layers based on agreement[1]. Many developments are brought into the Capsule network to advance its application from a simple handwritten digit recognition model to a full-grown image segmentation model. This research focuses on improving the accuracy of one such model.

In this paper, new architecture Residual Capsule Network V2 (RCN2) is proposed, which is an improvement to a contemporary architecture Residual Capsule Network (RCN)[6], to work on an image classification task by improving the model's complexity to work efficiently and achieve a fair trade-off between model accuracy and size. The paper focuses on improving the model size by including modifications that have been proven in different Capsule Network models while keeping the accuracy close to the original model. The modifications include changes in initial convolutional layers, routing by agreement algorithm, reconstruction network, and activation functions.

II. Background

A. Convolutional Neural Networks

CNN grew fast in the past two decades, especially in the domain of image segmentation and classification. They are also called space invariant artificial neural networks since they are shift invariant[7]. CNN's are considered as learned feature detectors, as they use knowledge derived from a region of the image to other areas of the image[7]. They feed these detected features to scalar-feature detectors and utilize pooling to focus only on an average value or maximum value at a position in the image to reduce complexity.

B. Capsule Network

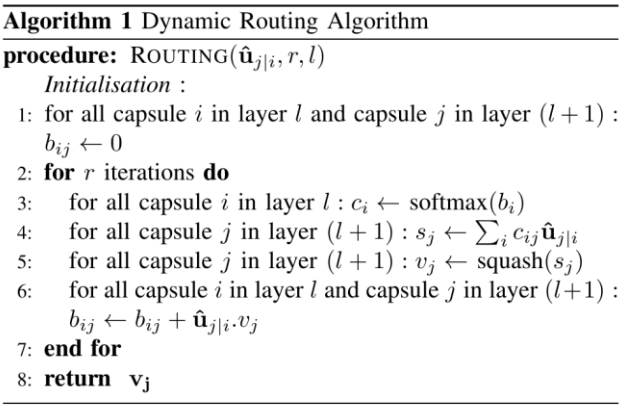
Addressing the above-mentioned scalar-feature detectors as well as the pooling layers is what was achieved by Sabour et al. by giving a new start to a completely different set of Neural Networks, Capsule Networks [7]. Taking the shift-invariance and inundating it with equivariance has led to the Capsule Neural Networks branch's unfolding. To overcome the pooling layer constraint, they introduced a new idea for routing by agreement in the paper Dynamic Routing between Capsules [7].

A Capsule is a collection of neurons that independently activate depending on various features of an object, such as size, hue, and position[6]. Each capsule represents a significant feature of the task at hand. The probability of each aspect of an object present in the current input is represented

This is the author's manuscript of the article published in final edited form as:

Anilkumar, A. N., & El-Sharkawy, M. (2021). RCN2: Residual Capsule Network V2. 2021 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS), 1–5. https://doi.org/10.1109/DTS52014.2021.9498216

by each capsule's output vector [1]. A capsule filled with routing by agreement creates an interesting pooling effect in reducing the neural connections but, at the same time, not losing any information [7]. The idea of routing nodes that agree to a capsule is an imitation of human perception. One simple way of doing the routing by agreement is with dynamic routing. The dynamic routing algorithm is as per Fig. 2 [1].



```
Algorithm 1 Dynamic Routing Algorithm
procedure: ROUTING($\hat{\mathbf{u}}_{j|i}, r, l$)
  Initialisation :
  for all capsule $i$ in layer $l$ and capsule $j$ in layer $(l+1)$ : $b_{ij} \leftarrow 0$
  for $r$ iterations do
      for all capsule $i$ in layer $l$ : $c_i \leftarrow \text{softmax}(b_i)$
      for all capsule $j$ in layer $(l+1)$ : $s_j \leftarrow \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$
      for all capsule $j$ in layer $(l+1)$ : $v_j \leftarrow \text{squash}(s_j)$
      for all capsule $i$ in layer $l$ and capsule $j$ in layer $(l+1)$ : $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i}.v_j$
  end for
  return $\mathbf{v_j}$
```

Fig. 2. Algorithm 1: Dynamic Routing Algorithm[1]

The squashing function used in Algorithm 1 is [1]:

$$v_j \quad \frac{\|s_j\|^2}{1+\|s_j\|^2}\frac{s_j}{\|s_j\|} \tag{1}$$

C. Residual Capsule Network

RCN was a merger of ResNet architecture to Capsule Network[6]. The main idea was to create a complex initial layer compared to the capsule network. The capsule network depends on a simple convolution layer in the initial layer, and this is improved by the authors Bhamidi and El-Sharkawy of RCN utilizing residual convolutional neural networks from ResNet [6], [8]. This helps the capsule network to be a deeper neural network as the ResNet gives the flexibility to avoid vanishing gradient [6], [8]. This interesting modification was achieved without affecting the viewpoint of Hinton et al. by not using any max pooling or average pooling that limit the network to be only shift-invariant. The residual network is combined and is fed in primary capsules, which give an output of 8-dimensional capsules, and this is further added into the digit capsules, same as the Capsule Network with a space conversion to 16-dimensional capsule output[6]. This network also applies the decoder network of 3 fully connected layers. This model produced an accuracy of 84.16% with 11.86M parameters for the CIFAR-10 dataset compared to the accuracy of 89.40% with the seven ensemble Capsule Network model with 101.5M parameters[6].

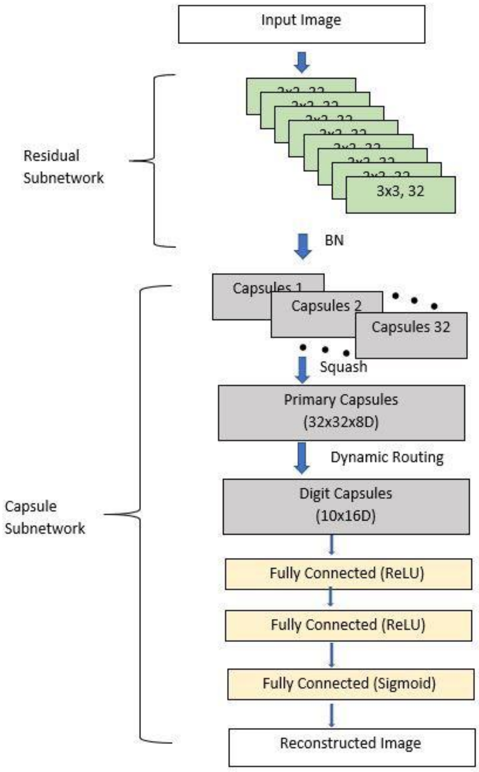
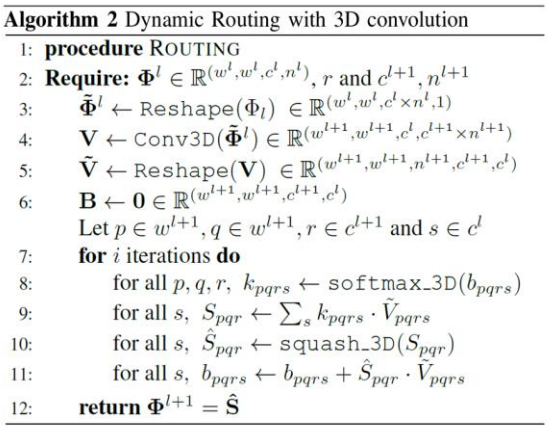


Fig. 3. Residual Capsule Network[6]

D. DeepCaps

While RCN was taking form another model with a similar aim, DeepCaps was also forming. DeepCaps focused on improving the routing by agreement algorithm rather than the structural changes from Capsule Network[9]. This new routing by agreement algorithm introduced by DeepCaps, 3D convolution-based dynamic routing, is given below as Algorithm 2[9]. The initial dynamic routing algorithm routed each capsule in layer '*L*' to each capsule in layer *'L+1'*[9]. Considering closer capsules share similar information, the authors of DeepCaps eliminates the above redundancy by including a 3D convolution that groups lower-level capsules and then routes this information to the higher-level capsules[9]. This helped the model become the state of the art capsule network to classify CIFAR-10 [10].



```
Algorithm 2 Dynamic Routing with 3D convolution
procedure ROUTING
Require: $\Phi^l \in \mathbb{R}^{(w^l,w^l,c^l,n^l)}$, $r$ and $c^{l+1}, n^{l+1}$
    $\tilde{\Phi}^l \leftarrow \texttt{Reshape}(\Phi_l) \in \mathbb{R}^{(w^l,w^l,c^l \times n^l,1)}$
    $\mathbf{V} \leftarrow \texttt{Conv3D}(\tilde{\Phi}^l) \in \mathbb{R}^{(w^{l+1},w^{l+1},c^l,c^{l+1} \times n^{l+1})}$
    $\tilde{\mathbf{V}} \leftarrow \texttt{Reshape}(\mathbf{V}) \in \mathbb{R}^{(w^{l+1},w^{l+1},n^{l+1},c^{l+1},c^l)}$
    $\mathbf{B} \leftarrow \mathbf{0} \in \mathbb{R}^{(w^{l+1},w^{l+1},c^{l+1},c^l)}$
    Let $p \in w^{l+1}, q \in w^{l+1}, r \in c^{l+1}$ and $s \in c^l$
    for $i$ iterations do
        for all $p, q, r,$ $k_{pqrs} \leftarrow \texttt{softmax\_3D}(b_{pqrs})$
        for all $s,$ $S_{pqr} \leftarrow \sum_s k_{pqrs} \cdot \tilde{V}_{pqrs}$
        for all $s,$ $\hat{S}_{pqr} \leftarrow \texttt{squash\_3D}(S_{pqr})$
        for all $s,$ $b_{pqrs} \leftarrow b_{pqrs} + \hat{S}_{pqr} \cdot \tilde{V}_{pqrs}$
    return $\Phi^{l+1} = \hat{\mathbf{S}}$
```

Fig. 4. Algorithm 2: Dynamic Routing with 3D convolution[9]

The algorithm proposed in DeepCaps reduces the number of parameters per routing by a factor of $c * (w^L w^{L+1})^2$

contrast to the previous routing algorithm[9]. The squashing function used in Algorithm 2 is [1]:

$$\hat{S}_{pqr} \quad \frac{\|s_{pqr}\|^2}{1+\|s_{pqr}\|^2}\frac{s_{pqr}}{\|s_{pqr}\|} \tag{2}$$

E. 3-Level Residual Capsule Network

3-Level Residual Capsule Network used the idea from the very efficient YOLO v3 model and included it in the Residual Capsule Network[11]. In Residual Capsule Network, every layer before primary capsules is connected in a feed-forward manner giving the final convolution layer, which is then given as input of primary capsules[6]. This architecture, in Fig. 3, has deviated from this straight feed-forward layer to 3 level stages like YOLO v3 enabling the network to see different scopes of the image[11]. This model's accuracy increased from 84.16% of Residual Capsule network to 86.42%, along with a parameter reduction of 8.9%[11].

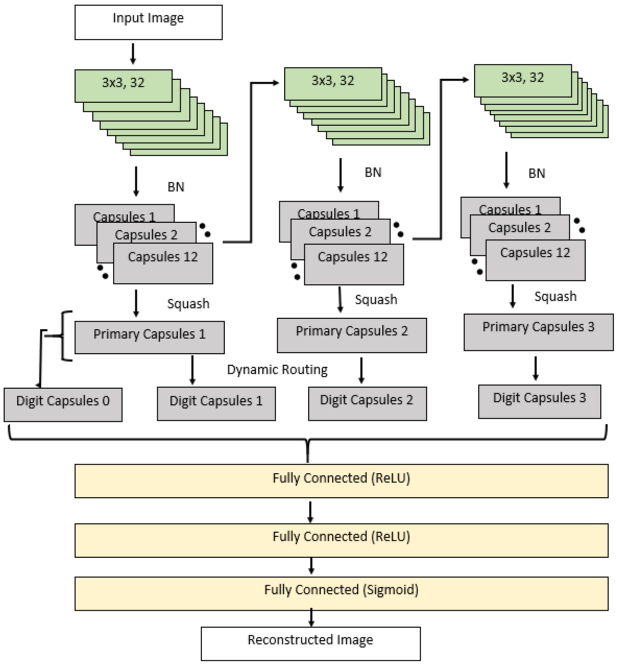


Fig. 5. 3 Level Residual Capsule Network [11]

III. PROPOSED RCN2: RESIDUAL CAPSULE NETWORK V2

The below sections demonstrate the structural improvements incorporated to RCN and thereby create the proposed RCN2: Residual Capsule Network V2. Fig. 7 shows the new model RCN2.

A. Residual Convolution Layers with bottleneck

To achieve higher accuracy, convolution layers should extract distinguishable features between classes expertly[6]. Although Capsule Network works on the principle of routed capsules, the initial layers are convolutions. Multiple layers of convolution layers give the model a better start in learning complex features, but this introduces the vanishing gradient issue. To address this, Residual Network architecture is embedded in the primary layers[6].

The Residual Capsule Network holds eight layers of the residual network. The residual network, which was inspired from [8] as mentioned by the contributor of RCN, did not include the deeper bottleneck architecture which was brought forth in the ResNet architecture [6], [8]. It was understood that including a bottleneck architecture to the given layer will decrease the training time [8]. Each residual network where 2-layers are used is changed to a 4-layer structure containing 1x1, 3x3,1x1, and 3x3 convolutions. Embedded with bottleneck technique, which, along with Identity shortcuts, gives less time complexity and less model size is achieved [6]. This network was then configured like a 3-Level Residual Capsule Network[11] to incorporate the view of different scaled images for the network to improve accuracy. The number of layers of residual networks was reduced to two per primary capsule.

B. Residual Capsule Network with 3D convolution-based dynamic routing

The opening convolutional layer of a Capsule Network only transforms the pixel depths to vector projects from local feature detectors[1]. These are then connected to primary capsules, which are then connected to Digit capsules, and finally, each digit capsule is merged to produce the desired classification result, from which there is a reconstruction network that helps reassure the network to encode the inputs instantiation parameter[1].

The residual convolution network is followed by Primary capsules that produce 8-dimensional capsules as output, which is then fed into Digit capsule layers that convert these 8-dimensional input vectors to 16-dimensional capsule vectors[6]. In the Digit capsule layer, feature maps are squashed and routed via a dynamic routing algorithm. We replaced the default routing algorithm in the capsule layer, mentioned in Fig. 3 Algorithm 1, with a 3D convolution-based dynamic routing algorithm proposed in DeepCaps[9], which tremendously dropped the repetition by routing blocks of capsule *'s'*, from layer *'L'* to layer *'L+1'*, instead of routing all capsules in layer *L* separately[9]. This algorithm is mentioned in Fig. 4 Algorithm 2. This modification reduced the parameter by a factor of $c * (w^L w^{L+1})^2$when compared to the existing routing algorithm[9].

C. 3D reconstruction by decoder network

After the Digit capsules giving the classification output together, the reconstruction network of the RCN does not include 3-dimensional reconstruction but restricted to a single layer reconstruction. Although this is okay to do, this is not the best that can be done, and therefore we included the 3D reconstruction as well as modified the reconstruction network to be more complex by incorporating class independent decoder network [9].

D. Mish activation function

Activation functions give the Neural Networks the non-linearity required to learn mapping functions, which contributes to an integral part of performance and training[12]. Between the layers of ResNet, the activation function used in each layer is ReLU activation. The Mish activation function replaced this as shown via experiments that Mish worked better than ReLU[12]. It has several other advantages. It ameliorates overfitting and is self-regularizing. It prevents saturation caused by near-zero gradients and provides better generalization[12].

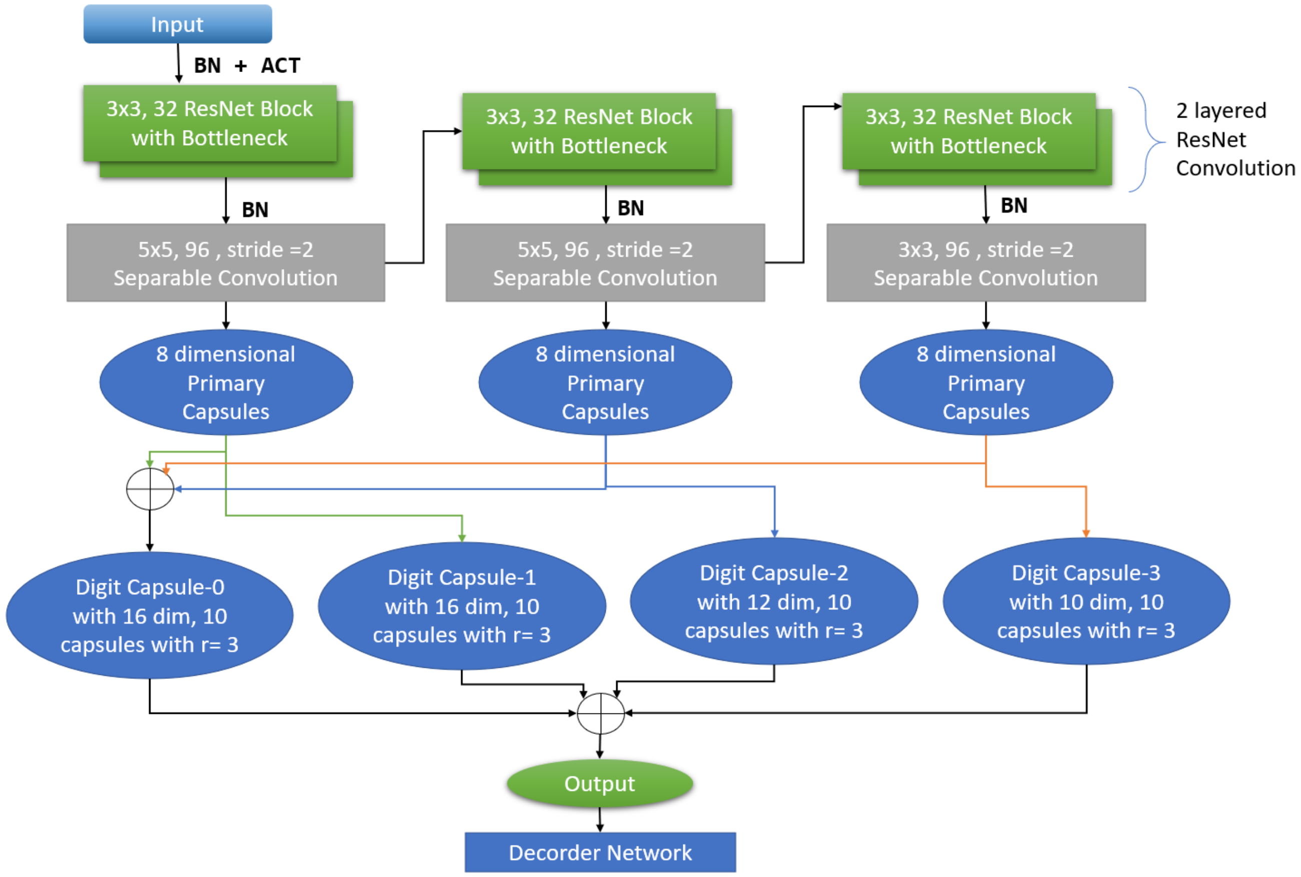

Input
BN + ACT
3x3, 32 ResNet Block with Bottleneck
BN
5x5, 96 , stride =2 Separable Convolution
8 dimensional Primary Capsules
3x3, 32 ResNet Block with Bottleneck
BN
5x5, 96 , stride =2 Separable Convolution
8 dimensional Primary Capsules
3x3, 32 ResNet Block with Bottleneck
BN
3x3, 96 , stride =2 Separable Convolution
8 dimensional Primary Capsules
2 layered ResNet Convolution
Digit Capsule-0 with 16 dim, 10 capsules with r= 3
Digit Capsule-1 with 16 dim, 10 capsules with r= 3
Digit Capsule-2 with 12 dim, 10 capsules with r= 3
Digit Capsule-3 with 10 dim, 10 capsules with r= 3
Output
Decorder Network

Fig. 7. Residual Capsule Network V2

E. Summary

In summary, as input travels through various ResNet blocks, the primary capsules capture them at different stages allowing the network to analyze the image at various stages of feature extraction. These primary capsules produce eight-dimensional vectors, each representing a feature of the object, which moves through a digit capsule layer with three iterations of routing to produce various dimensional vectors, after which it is merged to produce an output. Furthermore, there is a decoder network to decode each output to its corresponding input.

IV. Training Setup

We have used Lenovo ThinkSystem SD530 compute node with Intel Xenon Gold 6126 processors, 64GB RAM, and GPU acceleration with NVIDIA Tesla P100 for training the model [13]. CIFAR-10 Dataset was utilized for training and testing the model's efficiency[10]. The model was trained for 120 epochs, with a batch size of 128, a learning rate of 0.001, optimizer as adam, and learning rate decay of 0.9.

V. Results

The proposed model was trained and tested against the CIFAR-10 benchmark dataset[10]. RCN2 produced an accuracy of 85.12% in the test, and the evaluation model size is 1.95 M parameters. We have compared the result of RCN2 with results of other capsule network models like Baseline Capsule Network by Sabour et al., DCNET, DCNET++, Residual Capsule Network, and 3-Level Residual Capsule Network by Bhamidi et al. We utilized the software of Residual Capsule Network by Bhamidi et al. to generate the modified software of RCN2.

CIFAR-10 dataset contains 60,000 images of 10 classes with 6000 images per class. Out of these, 10,000 images are used for testing, and the rest is for training the model.

When the number of parameters after discounting reconstruction network, which is not included in the evaluation model, we reached a size of 1.95 M parameter, comparable to CNNs such as MobileNetV3, which has 1.8 M parameters. Comparison to other models is as per Table 1 [1], [6], [11].

Table I
Performance of various network models on CIFAR10

Model	**Number of Parameters**	**Test Accuracy**
Proposed RCN2	1.95 M	85.12 %
Residual Capsule Network	11.86 M	84.16 %
3-Level Residual Capsule Network	10.8 M	86.42%
Baseline Capsule Network	101.5 M	89.40%
DCNet	11.8 M	82.63%
DCNet++	13.4M	89.71%
MobileNetV3	1.8 M	88.93%

VI. Conclusion

In this paper, a new architecture RCN2 is introduced. Using ResNet with bottleneck, Capsules with 3D convolution-based dynamic routing for routing by agreement, with an architecture inspired from 3-Level Residual Capsule Network, 3D reconstruction, and Mish activation function, we achieved a model that is efficient with an accuracy of 85.12 % and finer with the model size of 1.95 M parameters when trained and tested on CIFAR-10. Thus the proposed RCN2 is comparable to embedded models like MobileNetV3.

Acknowledgment

The authors would like to thank Lilly Endowment Inc for their support to Indiana University Pervasive Technology Institute.

References

[1] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic Routing Between Capsules," in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Long Beach, CA, USA., p. 11, Accessed: Jan. 10, 2021. [Online]. Available: https://papers.nips.cc/paper/2017/file/2cad8fa47bbef282badbb8de5374b894-Paper.pdf.

[2] G. W. Lindsay, "Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future," *Journal of Cognitive Neuroscience*, pp. 1–15, Feb. 2020, doi: 10.1162/jocn a 01544.

[3] P. F. Sharp and R. Philips, "Physiological Optics," in *The Perception of Visual Information*, W. R. Hendee and P. N. T. Wells, Eds. New York, NY: Springer, 1997, pp. 1–32.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[5] M. Perello Nieto, "File:Human visual pathway.svg - Wikimedia Commons." https://commons.wikimedia.org/wiki/File:Human vis ual pathway.svg (accessed Jan. 10, 2021).

[6] S. B. S. Bhamidi and M. El-Sharkawy, "Residual Capsule Network," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, Oct. 2019, pp. 0557–0560, doi: 10.1109/UEMCON47517.2019.8993019.

[7] "Convolutional neural network - Wikipedia." https://en.wikipedia.org/wiki/Convolutional neural n etwork (accessed Jan. 10, 2021).

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[9] J. Rajasegaran, V. Jayasundara, S. Jayasekara, H. Jayasekara, S. Seneviratne, and R. Rodrigo, "DeepCaps: Going Deeper With Capsule Networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 10717–10725, doi: 10.1109/CVPR.2019.01098.

[10] R. C. Calik and M. F. Demirci, "Cifar-10 Image Classification with Convolutional Neural Networks for Embedded Systems," in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, Aqaba, Oct. 2018, pp. 1–2, doi: 10.1109/AICCSA.2018.8612873.

[11] S. B. S. Bhamidi and M. El-Sharkawy, "3-Level Residual Capsule Network for Complex Datasets," p. 4.

[12] D. Misra, "Mish: A Self Regularized Non-Monotonic Activation Function," *arXiv:1908.08681 [cs, stat]*, Aug. 2020, Accessed: Nov. 29, 2020. [Online]. Available: http://arxiv.org/abs/1908.08681.

[13] C. A. Stewart, V. Welch, B. Plale, G. Fox, M. Pierce, and T. Sterling, "Indiana University Pervasive Technology Institute," Sep. 2017, doi: 10.5967/K8G44NGB.