School of Informatics

Informatics Research Review
Interactive feature extraction with Capsule Network for image Classification

[redacted]

Abstract

This literature review provides a thorough review of the architectures, methodologies, concepts in the capsule network's existing implementations. This paper also provides a review of performance evaluation on different image datasets for the image classification task showing, CapsNet gives state of art performance for simple datasets but fails to perform for a complex dataset. We further reviewed some of the architectural modifications suggested by researchers in terms of feature extraction layers. The modified architectures improve the accuracy for complex datasets by almost 10% to 15%. We highlight the successes, shortcomings, and opportunities for future research to present as a motivation to researchers and students to make the most of the full potential of this new concept.

Date: Friday 22nd January, 2021

Supervisor: [redacted]

Contents

1 Introduction

1.1 Background and Motivation

Convolutional Neural Network (CNN) has shown exemplary performance in fields like Image Processing[1] and Computer Vision[2]. Some of the exciting application areas of CNN include Image Classification, Object Detection, Image Segmentation, Video Processing, and Natural Language Processing[3]. It has also given the state-of-the-art performance in the research fields like medical, biological, and ecological, for example, CNN's have helped in the diagnosis of coronavirus from CT images[4]. Despite the huge success, CNNs has limitations. Two well-known limitations are they do not perform well when the input image is translated, rescaled, or transformed[5]. To solve this problem, the image augmentation exists, which scales, rotates, crop, zoom, and perform a variety of other transformations on image dataset to create all possible transformation combination, which is then passed through the network to learn better. But this is computationally very expensive and requires large data[6]. Another problem is caused by max pooling layer in CNNs architecture which can maintain the present information but ignore the positional one and lead to loss of information[7]. Recently Capsule networks (CapsNets)[8], a new model that employs the concept of a capsule is proposed to solve these two major shortcomings of the CNNs by implementing a complex architecture to recover lost

information around CNNs. Since their introduction, there has been an associated degree upsurge in implementing deep learning architecture with CapsNets as their main building block[9, 10, 11, 12, 13]. Owning to the fact CapsNets is a new network with a huge amount of research happening and showing promising result over CNN's[8, 6, 14], we choose to explore CapsNets for one of the Computer Vision tasks of image classification.

1.2 Aim of the Review

This paper is simple and comprehensible introduction to CapsNets and its performance for image classification task. The targeted audience are graduated students in computer science and related fields with strong knowledge of machine learning, deep neural networks and basic knowledge of vector, differential equations. The ultimate aim is to encourage, to motivate and to raise awareness of research on this subject. We will explore the following questions to do this:

- What is Capsule Neural Network?
- How does Capsule Neural Network overcome the limitations of Convolution Neural Network?
- Do they perform better than Convolution Neural Network for image classification?
- What are the suggested further improvements and what is the future vision of this network?

1.3 Paper Structure

Following the aim of the review, the papers for reviews are selected on certain criteria. The selected papers should give a brief introduction to existing CapsNet architectures and their implementation. Secondly, the papers should give experimentation justification for the performance of CapsNet Over CNNs for different image datasets classification.

The review paper is arranged as follows. Section 2 contains a brief introduction to CNN. This section starts by giving an overview of CNN architecture and review related to the major limitation of CNNs. Section 3 introduces Geoffrey Hinton proposed Capsule Neural Network, describing the working and architecture overview of CapsNets, following a review related to the changes in CapsNets concerning CNNs. Section 4 reviews the Performance of CapsNets on the image dataset in comparison with baseline CNNs. This section also gives a review of the analysis related to the learned features along with the performance in terms of test error for image classification tasks. Section 5 reviews the modified CapsNets architecture proposed in terms of modification of the initial feature extraction layers. Section 6 highlights the research areas and future vision of CapsNets. Lastly, section 7 gives a summary and a conclusion about the literature review.

2 Convolution Neural Network and Limitation

2.1 Architecture

As shown in fig 1 a basic convolutional network architecture is composed of alternating layers of convolution(c-layers) and sub-sampling layer also known as pooling layer(s-layers). C-layer

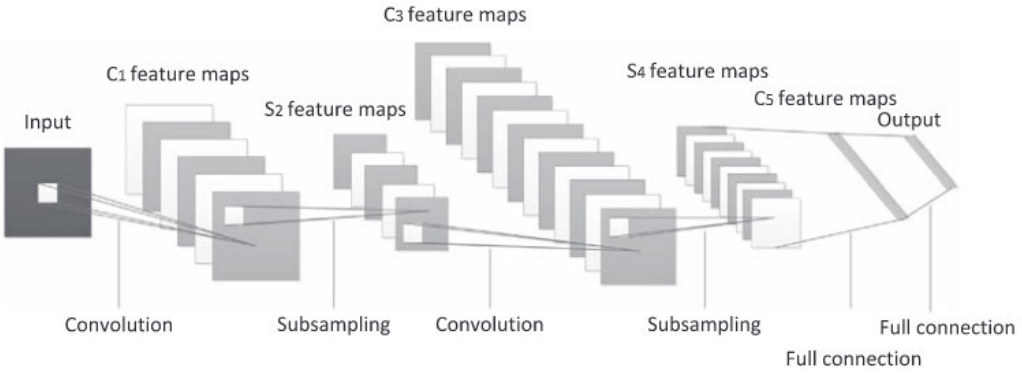

Figure 1: Schematic structure of CNNs[15]

track patterns and information found within local regions of the input images that are common all over the dataset. The patterns are extracted by moving the filter over the input image pixels and computing the dot product of the filter at each location within the image. The output of this layer is feature map m. A non-linear activation function is then applied element-wise to each feature map $m : a = f(m)$[7]. The resulting activations are forwarded to the s layer. Its purpose is to gradually reduce the size of the feature map representation to reduce the number of parameters and hence the network computation[15]. This procedure continues and leads to the feature maps in the following c-layers and s-layers. Finally, the fully connected layer takes the feature map values and converts them into a single vector which acts as the input to the next stage to get an output prediction, for example, image classification output[7].

2.2 Major Limitation

In general, the result of pooling operation is a new transformed more usable joint feature representation, preserving important information while discarding irrelevant details. Pooling layers transfer the activation information from one layer to the next layer in CNNs. It tells the layers about the presence of a part, but not the spatial relation between the parts. For example, In Average pooling the average feature activation matters, but not the exact spatial localization[16]. The pooling operation also treats each feature map independently hence it only gives information related to the present feature map not about the relation between other feature maps. Therefore, CNNs fails in a situation where features are strongly correlated and should be predicted jointly[17]. For example, if we try to classify an image as a face, we have to combine some features (2 eyes, nose, and a mouth) to classify it as a face. But in the case of CNNs, it would classify any image as a face if those features are present with high probability anywhere within the image.

Small translation or transformation of an image can drastically change the classification results[18]. Results obtained by researcher Weiss Y indicates that CNNs are not transformations invariant and can give very different classification results for images with no transformations and for images with transformations of a single pixel[5]. By experimenting with different CNN networks researcher observed that the chance of change in CNN output on a randomly chosen image after transformation by a single pixel can be as high as 30% which can highly deteriorate the CNN performance. It was concluded by the researcher that this generalization issue is due to the CNNs architecture with layers of convolution and pooling that ignores the sampling theorem[5].

3 Capsule Neural Network

3.1 Capsule and Dynamic Routing Algorithm

Drawing inspiration from neuroscience, Geoffrey Hinton described capsules as modules organized in the human brain to visualize information. With this in mind, the author proposed capsule networks with dynamic routing algorithms to estimate features of entities like pose (orientation, position, deformation, velocity, size, etc). Therefore, Capsule Neural Network is a new deep neural network type that processes visual information almost in a similar way as the human brain and can maintain hierarchical relationships.

CapsNets replaces the scalar-output feature maps of CNNs with a vector-output capsule which is a group of neurons creating an activity vector. Capsule vector output represents the probability of the object's existence and its instantiated attributes. The activation vector is obtained by a squashing function given by[8]

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

where v_j is the capsule j output vector and s_j is the input vector to capsule j

$$s_j = \sum_i c_{ij} u_{j|i} \tag{2}$$

For all layer of capsules except first, sj is a weighted sum of every prediction vectors $u_{j|i}$ from the below layer capsules. where $u_{j|i}$ is obtained by multiplying weight matrix W_{ij} and the capsule output u_i in the layer below given by $uj|i = W_{ij}u_i$. c_{ij} a coupling coefficient generated by applying softmax function to logits b_{ij} given by[8]

$$c_{ij} = \frac{exp(b_{ij})}{\sum_k 1 + exp(b_{ik})} \tag{3}$$

Dynamic Routing Algorithm also known as routing by agreement algorithm, is used to train a capsule network is shown in a fig 2. The first line of the algorithm takes all present capsules at layer L along with their outputs u, and the number of routing iterations given by the user r. The last line shows that the algorithm will return v_j as the output of a capsule present at layer L+1. The second line defines a temporary coefficient b_{ij} which is log prior probabilities of coupling between capsule i and capsule j. It is iteratively updated and stored in c_i. Algorithm initialize $b_{ij} = 0$ at the start. The third line defines that the lines 4–7 will be repeated r times. The fourth line computes the value of the vector c_i, which is the weight for the capsule i at level L. To ensure that each weight c_i is non-negative, and sum up to 1, softmax is applied. In the fifth line after calculating all the weights for L layer capsules, the algorithm calculates input vector s_j of capsule j in Layer L +1. Each input vector is weighted by the routing coefficient c_{ij} obtained in line 5. In the sixth line of algorithm squash, non-linearity is applied to vectors, which ensures the preservation of the direction component of the vector.[8]

In a nutshell, the vector parameters help the model to learn and predict. Vector parameters represent the input image attributes like orientation, size, position, etc this helps to solve the CNN problem related to transformations[8]. CNN's pooling is replaced with routing-by-agreement. In the routing by agreement concept, there is top-down feedback which helps the capsule to pass its output to the parents that agrees with its output. This process helps to perform prediction by considering features jointly and solve the problem caused by CNN pooling layer[9].

Procedure 1 Routing algorithm.

1: **procedure** ROUTING($\hat{\boldsymbol{u}}_{j|i}, r, l$)
2: for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow 0$.
3: **for** r iterations **do**
4: for all capsule i in layer l: $\mathbf{c}_i \leftarrow \texttt{softmax}(\mathbf{b}_i)$ ▷ `softmax` computes Eq. 3
5: for all capsule j in layer $(l+1)$: $\mathbf{s}_j \leftarrow \sum_i c_{ij} \hat{\mathbf{u}}_{j|i}$
6: for all capsule j in layer $(l+1)$: $\mathbf{v}_j \leftarrow \texttt{squash}(\mathbf{s}_j)$ ▷ `squash` computes Eq. 1
7: for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i}.\mathbf{v}_j$
return $\mathbf{v}_j$

Figure 2: Dynamic Routing Algorithum[8]

3.2 CapsNet architecture

Basic CapsNet architecture is shown in Fig 3. The architecture is explained by taking an example of digit image classification. The first layer is the convolutional layer with 256 kernels with each of size 9x9x1 and of stride 1, followed by ReLU activation which performs the detection of basic features. The second layer is a PrimaryCaps layer with 32 primary capsules, takes the convolutional layer identified basic features as input, and produces combinations of the features. Each primary capsule produces 6x6x8 output by applying 8, 9x9x256 convolutional kernels with stride 2 on the input of size 20x20x256[9]. The final Layer DigitCaps has 10, 16-dimensional capsules for each digit classification, each of these capsules get input from all the capsules in layer L i.e in the layer below. Routing only used in between the two following capsule layers PrimaryCapsules and DigitCaps. As the Convolution layer output is 1D, there is no orientation present for the agreement. Hence, no routing is present between the Convolution layer and PrimaryCapsules. Initially, All the routing log prior probabilities (b_ij) are zero. Therefore, initially, every parent capsule is given by a capsule output (u_i) with the same probability[8].

As shown in fig 4 decoder or reconstruction unit of CapsNet takes a 16-dimensional input vector from the DigitCap layer and learns to reconstruct it into a true digit image. For training, it uses only the correct DigitCap vector while incorrect ones are ignored. The decoder acts as a regularizer, it takes the correct DigitCap vectors and learns to decode an image of size 26 x 28. Euclidean distance is used as a loss function between the input image and the reconstructed image. In this way, the decoder forces capsules to train and learn the most useful features for recreating the true image[9]. The closer the reconstructed image and input image, the better the performance[8].

4 Examining Capsule Neural Networks Over Convolution Neural Network

4.1 Analysing Learned Features

To evaluate whether CapsNets truly operate differently than traditional CNNs there is a need to observe the differences in the capsule features.

Arjun Punjabi performed several analyses to analyze capsule features to discover whether CapsNets perform better than CNNs or not[6]. One of the major analysis used is activation maximization. In general, activation maximization is an way to produce and visualise images that

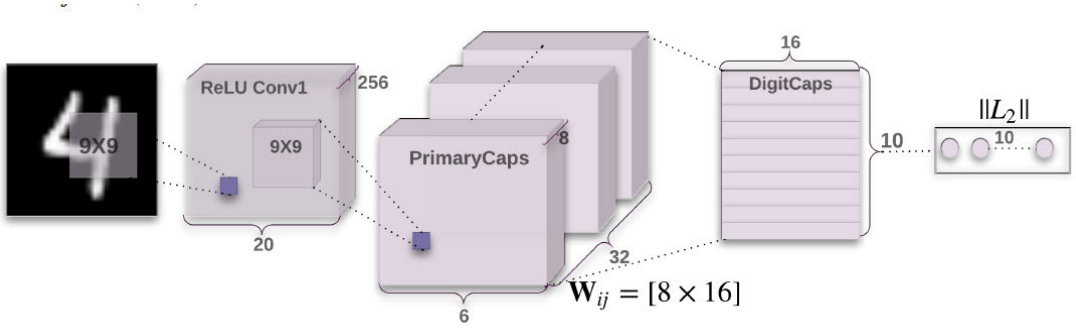

Figure 3: CapsNet encoder architecture[8]

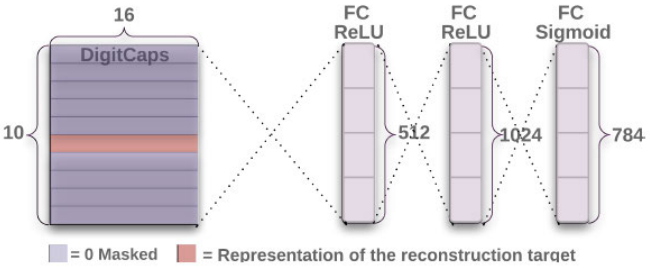

Figure 4: CapsNet decoder architecture[8]

can represent intermediate network learned features, its given by[6] $x^* = \text{argmax}_x(a_i(x) - R_\theta(x))$ Where x^* is the visualization output. x is the input image. $a_i(x)$ is the particular unit activation and $R_\theta(x)$ is some regularization function. Different value of i leads to a visualization of a different kind of information. The author evaluated the features of a network by modifying the above equation of activation maximization by restructuring it from maximization to a minimization, given by[6]

$$x^* = \underset{x}{\text{argmin}}(l(\phi(x), \phi_0(x)) - R_\theta(x)) \tag{4}$$

where $\phi(x)$ is input image, $\phi_0(x)$ is the target feature representation and $l(\phi(x), \phi_0(x))$ denotes a loss function between them.This equation creates an image from features that ensembles closely the input image, so author defined it as an activation matching rather than the activation maximization.

The author trained baseline CNN and basic CapsNet with or without decoder unit as shown in fig 3. The traning is performed on transformed digit images of MNIST[19] dataset. The baseline CNN has three convolutional layers of channels 256x256x128, each channel with 5x5 kernels and a stride of 1. Two fully connected layers of size 328 x 192 are connected to the last convolutional layers. Softmax layer of 10 classes each per digit is connected to the last fully connected layer. The researcher further used the proposed activation maximization analysis to evaluated learned features of both the networks. Fig.5 shows activation maximization algorithm created 100 images when applied to a CapsNet (with reconstruction unit and with no reconstruction unit) and for the baseline CNN. The created images are presented in a 10x10 grid with a decreasing activation value. Therefore, the top left gird shows the image with the highest activation value whereas the bottom right gird shows the image with the lowest activation value. By comparing activation maximization analysis images researcher concluded that Capsule network images were

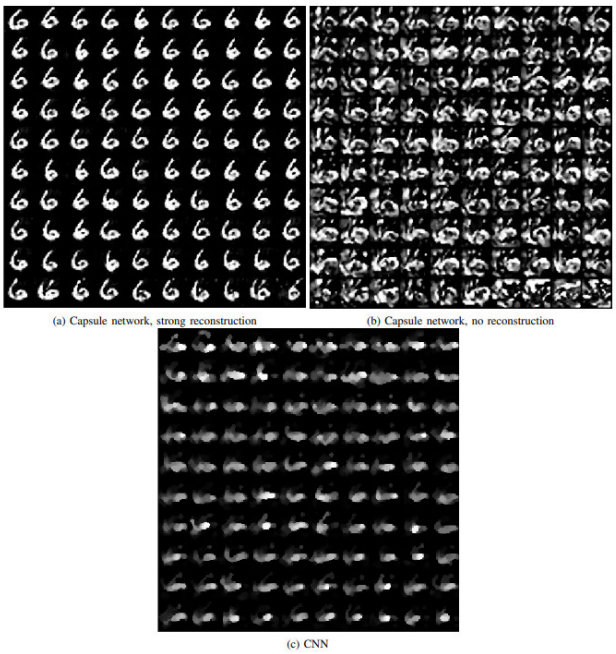

Figure 5: Activation maximizations for digit 6. a)Capsule Network with reconstruction b) Capsule network without reconstruction, c)CNN [6]

clearer than convolution Network images and were able to reconstruct the input image more precisely as compared to CNN's. hence Capsule network was better able to describe all image features than CCN and can perform better[6].

4.2 Performance on Image Datasets

Geoffrey Hinton evaluated the performance of CapsNets for image classification on image dataset like MNIST[19], MultiMNIST and complex image datasets like CIFAR10[20] and further compared the CapsNets performance with baseline CNN[8].

The first training is performed on MNIST[19] dataset images by shifting the input images by 2 pixels in each direction with zero padding. The dataset has 60K training images and 10K testing images. The table 1 shows the test error results obtained for CapsNet setups. CapsNet gives a state of the art performance with the lowest test error of 0.25% on a 3 layer network[8]. For this experiment the total number of parameters in CapsNet was 8.2M with reconstruction unit and 6.8M with no reconstruction unit. The number of parameters for baseline CNN was 35.4M. As the CapsNet with reconstruction unit gives higher performance compared to CapsNet without reconstruction unit, the author concludes that the reconstruction regularizer unit plays an important role in increasing the performance of CapsNets[6, 21].

We can describe dynamic routing as an attention mechanism that allows a particular capsule at layer L to attend some active capsules at the layer L-1 while ignoring others. According to Geoffrey Hinton, this mechanism should enable the model to recognize overlapping objects in an image[8]. To proof this, the author trained a 3-layer CapsNet Network and Baseline CNN with a self-created overlapping digit dataset known as the MultiMNIST dataset. For this dataset training set and test set size is 60M and 10M respectively. Classification test error obtained with MultiMNIST dataset is shown in the table 1. As shown in the table 1, three layer

CapsNet model achieves lower test error hence higher classification accuracy than baseline CNN showing CapsNets gives high performance in case of overlapping objects prediction compared to CNNs[8].

Fig 6 shows a sample reconstructed image by a CapsNet with 3 routing iterations for the MultiMNIST dataset. The lower part of the image shows two reconstructed overlayed digits in green and red. Input images are shown in the upper part of the image. L:(l1, l2) shows two overlap input digit labels, and R:(r1, r2) shows the two digits from DigitCaps used by CapsNet for reconstruction. The two columns in right shows wrong classification examples reconstructed from the prediction and the label. The (*) marked columns show reconstructions from a digit that is not present in the label or in the predicted value. These columns conclude that the model is not only searching for the best fit for digits present in the image but also for the digit that does not present[8]. Hence for label (5, 7) model cannot reconstruct digit 7 because it learned that there is a digit 5 and 0 that fits better for the given input. Therefore reconstructions in Fig 6 illustrate that CapsNet is capable to segment the input image into the actual digits[8]. The author noticed that as this segmentation does not occur at pixel level as CNN the model is giving a good performance with the overlap's digits. DigitCaps consists of encoding of each digit[8]. The decoder has learned to recreate a digit from input encoding data. The author further concluded that despite the overlap the ability of the decoder to reconstruct digits represents that every digit capsule is able to take up the location and style from the information obtained from the PrimaryCapsules layer[6].

The author further trained capsule model with 3 layers on more complex data like the CIFAR10 dataset and achieved 10.6% error. The CapsNet model used was the same as used for the MNIST dataset except a "none-of-the-above" category added for the routing softmax function. As the author does not expect the ten capsules final layer to explain all the information in the image, the author observed that the CapsNet is giving a similar performance as baseline CNN which is poor as compare to CapsNet performance for MNIST dataset. The author suggested that the backgrounds of CIFAR10 images are too different to model in a reasonable-sized CapsNet which leads to poor performance[8]. In a study with CapsNet, Selina Bing suggested that the difference between MNIST and CIFAR10 performance may possibly due to the reconstruction method[14]. In capsule networks regularization technique try to minimize the difference between the reconstructed image and the true image. The author further commented that the capsule network is robust for affine transformation therefore regularization works exceptionally well on 2D digit images of MNIST where all of the transforms are mostly affine. However, classification of a 3-dimensional object in the real world requires the capability to recognize objects regardless of viewing angles i.e. viewpoint variance. CIFAR10 images unlike digits in MNIST have more than one viewpoint for each class therefore 2d reconstruction regularization approach on 3d data of CIFAR10 can lead to inaccurate regularization values for reconstruction and hence deteriorating performance of capsule network on complex data[14].

5 Modified Capsule Neural Networks

CapsNets gives a state-of-the-art performance on datasets that are simple like MNIST but fail to perform with complex data like CIFAR10. To improve performance with complex datasets Geoffrey Hinton suggested a need to use an ensemble procedure, which requires more training time and include more trainable parameters[8]. CapsNets are new architecture and hence highly under-investigation. For this literature review section, we will explore some modified CapsNet capabilities with the modification of the initial feature extraction layers.

Method	reconstruction	routing	MNIST (%)	Multi MNIST(%)
Baseline	-	-	0.39	8.1
CapsNet	no	1	0.34±0.032	-
CapsNet	no	3	0.35±0.036	-
CapsNet	yes	1	0.29±0.011	7.5
CapsNet	yes	3	0.25±0.005	5.2

Table 1: CapsNet and Baseline CNN test error for image classification task. The given average and standard deviation values are computed from 3 trials[8]

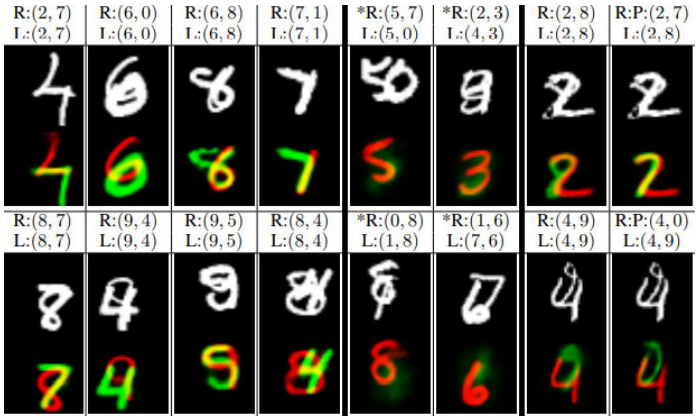

Figure 6: Sample reconstructed image for MultiMNIST image dataset given by a CapsNet with 3 routing iterations [8]

Rita Pucci has proposed deeper CapsNets by improving the initial convolution layers[9]. ResNetCaps proposed by adding ResNet18[22] architecture till the 2nd residual block before the capsule layers. In ResNet blocks residual connections is provided which directly adds the value at the input, x, to the end of the network block ($F(x)+x$). This residual connection doesn't go through activation functions and hence improves the feature extraction process[22]. ResNetCaps_IBN is proposed by improving the components of the ResNetCaps architecture inspired by IBN-Net[23]. In ResNetCaps_IBN batch Normalisation layer of ResNetCaps is merged with Instance in the second block layer. The modified layer known as IBN takes into account visual and appearance invariance and hence improves overall learning of the Network[9]. DenseNetCaps is proposed by adding DenseNet architectures[24] till first dense block and its corresponding first transition layer before the capsule layers. DenseNet architecture performs better and has fewer parameters as compared to ResNet. In DenseNet architecture, each preceding layer is connected to the succeeding layer in a feed-forward fashion[24]. Rita Pucci further trained basic CapsNet and all three modified network (ResNetCaps, ResNetCaps_IBN,DenseNetCaps) on CIFAR10. ResNetCaps, ResNetCaps_IBN, DenseNetCaps outperform over basic CapsNet with improvement of 10%, 16% and 14% accuracy respectively.[9].

Other modified architecture proposed such as In [25] author presented architecture consists of multiband feature matrix with a capsule network (CapsNet) to deal with ECG signals for emotion recognition. In [10] the convolutional layer replaced by an encoder block that learns features from Multiview datasets. In [26], authors introduce a hierarchical way of implementing primary capsules to deal with numerous image scales. In [13], authors proposed a deep capsule network with 3D capsules which enables the model to deal with complex datasets as CIFAR10. Results obtained by comparing modified networks with basic CapsNets shows that modification

improves the CapsNet performance which highlights the significance of investigating articulated architectures[9, 13, 25, 10].

6 Future Work and Vision

Geoffrey Hinton in recent paper[27] introduced a new modified CapsNet in which capsule consists of a logistic unit to represent the presence of a feature and a 4x4 pose matrix to represent the position of that feature. They also proposed a new routing procedure based on the EM algorithm known as EM routing. In EM routing capsule output at L layer will be given to the parent at L+1 layer depending upon the similar pose matrix votes. The inclusion of such a new concept of pose matrix and EM routing in CapsNet is a promising compelling and interesting research area in the future, as it improves capsule network's shortcomings on complex data[27]. Even though capsules performing better than CNNs there are some areas where their performance is not up to the mark. As the pooling layer is not present in the CapsNets therefore capsule will try to train and learn each and every feature pixel of the input image along with background noise deteorating performance for noisy images[28]. The performance of the capsule is different for different datasets, it given state of art performance for MNIST[19] dataset but fails to perform for CRIF10[20] dataset[8]. These areas require further research to improve the capsule Network.

CapsNet is a new architecture that is still under examination. Researchers are believing that it is an advanced method and requires some more study before it can out-perform a highly developed CNN technology. The fact that experiment conducted with a basic capsules system already giving a state of the art performance in the field of astronomy[29], automobiles[30], Natural Language Processing[31], Medical[32] etc is an early indication that the direction in direction of capsules is worth exploring.

7 Conclusion

One of the most relevant tasks in the Artificial Intelligence area is to process image data for machine vision. In this area, deep learning models such as CNNs have performed extremely well[1, 2]. However, they are not capable of recognizing the pose and deformation of entities along with loss of information. Therefore, capsules were introduced to solve the problems faced by CNNs and have performed well[8, 6]. However, as the area is new, it needs more understanding and research so that its full potential can be realized. This paper reviews the state of the art in Capsule networks and introduce present architectures and implementations. We analyzed and critically reviewed the performance of CapsNet over CNNs. Activation maximization analysis performed by Arjun Punjabi shows that CapsNet can learn more accurately than CNNs and hence can able to recreate the image more clearly using activation maximization[6]. The experiment performed by Geoffrey Hinton on different image datasets shows that CapsNet performed really well for simple datasets but performed similarly to Baseline CNN for the complex dataset, showing the need for architectural improvements to perform well on the complex dataset[8]. We further presented and reviewed the encoding phase modified CapsNets. By comparing the performance of modified networks with baseline CapsNet for complex dataset Rita Pucci concluded that all modified CapsNet are performing better than simple CapsNet showing the need for future research in articulated architectures[9]. Though the idea is strong, there is more to learn and develop. CapsNet with a matrix capsule algorithm, an enhancement over dynamic routing algorithm capsules, is a new idea proposed by Geoffrey Hinton that is worth researching[27].

References

[1] Leon O Chua and Tamas Roska. The cnn paradigm. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 40(3):147–156, 1993.

[2] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[3] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning Chapter 12 Application*, volume 1. MIT press Cambridge, 2016.

[4] Ali Abbasian Ardakani, Alireza Rajabzadeh Kanafi, U Rajendra Acharya, Nazanin Khadem, and Afshin Mohammadi. Application of deep learning technique to manage covid-19 in routine clinical practice using ct images: Results of 10 convolutional neural networks. *Computers in Biology and Medicine*, page 103795, 2020.

[5] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018.

[6] Arjun Punjabi, Jonas Schmid, and Aggelos K Katsaggelos. Examining the benefits of capsule neural networks. *arXiv preprint arXiv:2001.10964*, 2020.

[7] Matthew D Zeiler and Rob Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013.

[8] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.

[9] Rita Pucci, Christian Micheloni, Gian Luca Foresti, and Niki Martinel. Deep interactive encoding with capsule networks for image classification. *Multimedia Tools and Applications*, 79(43):32243–32258, 2020.

[10] Jian-wei Liu, Xi-hao Ding, Run-kun Lu, Yuan-feng Lian, Dian-zhong Wang, and Xiong-lin Luo. Multi-view capsule network. In *International Conference on Artificial Neural Networks*, pages 152–165. Springer, 2019.

[11]

[12] Lu Luo, Shukai Duan, and Lidan Wang. R-capsnet: An improvement of capsule network for more complex data. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2124–2129. IEEE, 2019.

[13] Jathushan Rajasegaran, Vinoj Jayasundara, Sandaru Jayasekara, Hirunima Jayasekara, Suranga Seneviratne, and Ranga Rodrigo. Deepcaps: Going deeper with capsule networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10725–10733, 2019.

[14] Edgar Xi, Selina Bing, and Yang Jin. Capsule network performance on complex data. *arXiv preprint arXiv:1712.03480*, 2017.

[15] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[16] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 111–118, 2010.

[17] Naila Murray and Florent Perronnin. Generalized max pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2473–2480, 2014.

[18] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015.

[19] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[21] Xuefeng Jiang, Yikun Wang, Wenbo Liu, Shuying Li, and Junrui Liu. Capsnet, cnn, fcn: Comparative performance evaluation for image classification. *International Journal of Machine Learning and Computing*, 9(6):840–848, 2019.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[23] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 464–479, 2018.

[24] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[25] Hao Chao, Liang Dong, Yongli Liu, and Baoyun Lu. Emotion recognition from multiband eeg signals using capsnet. *Sensors*, 19(9):2212, 2019.

[26] Sai Samarth R Phaye, Apoorva Sikka, Abhinav Dhall, and Deepti Bathula. Dense and diverse capsule networks: Making the capsules learn better. *arXiv preprint arXiv:1805.04001*, 2018.

[27] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *International conference on learning representations*, 2018.

[28] Sameera Ramasinghe, CD Athuraliya, and Salman H Khan. A context-aware capsule network for multi-label classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[29] Reza Katebi, Yadi Zhou, Ryan Chornock, and Razvan Bunescu. Galaxy morphology prediction using capsule networks. *Monthly Notices of the Royal Astronomical Society*, 486(2):1539–1547, 2019.

[30] Youngjoo Kim, Peng Wang, Yifei Zhu, and Lyudmila Mihaylova. A capsule network for traffic speed prediction in complex road networks. In *2018 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6. IEEE, 2018.

[31] Bodhisatwa Mandal, Suvam Dubey, Swarnendu Ghosh, Ritesh Sarkhel, and Nibaran Das. Handwritten indic character recognition using capsule networks. In *2018 IEEE Applied Signal Processing Conference (ASPCON)*, pages 304–308. IEEE, 2018.

[32] Aryan Mobiny and Hien Van Nguyen. Fast capsnet for lung cancer screening. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 741–749. Springer, 2018.