Modified Capsule Neural Network (Mod-CapsNet) for Indoor Home Scene Recognition

Amlan Basu
Ph.D. Student
Department of EEE
University of Strathclyde, Glasgow
amlan.basu@strath.ac.uk

Keerati Kaewrak
Ph.D. Student
Department of EEE
University of Strathclyde, Glasgow
keerati.kaewrak@strath.ac.uk

Lykourgos Petropoulakis
Senior Knowledge Exchange Fellow
Department of EEE
University of Strathclyde, Glasgow
l.petropoulakis@strath.ac.uk

Gaetano Di Caterina
Lecturer
Department of EEE
University of Strathclyde, Glasgow
gaetano.di-caterina@strath.ac.uk

John J. Soraghan
Professor
Department of EEE
University of Strathclyde, Glasgow
j.soraghan@strath.ac.uk

***Abstract* - In this paper, a Modified Capsule Neural Network (Mod-CapsNet) with a pooling layer but without the squash function is used for recognition of indoor home scenes which are represented in grayscale. This Mod-CapsNet produced an accuracy of 70% compared to the 17.2% accuracy produced by a standard CapsNet. Since there is a lack of larger datasets related to indoor home scenes, to obtain better accuracy with smaller datasets is also one of the important aims in the paper. The number of images used for training and testing is 20,000 and 5000 respectively, all of dimension 128X128. The analysis proves that in the indoor home scene recognition task the combination of the capsule without a squash function and with max-pooling layers works better than by using capsules with convolutional layers. Indoor home scenes are specifically focused towards analysing capsules performance on datasets whose images have similarities but are, nonetheless, quite different. For example, tables may be present in living rooms and dining rooms even though these are quite different rooms.**

Keywords – Capsule Neural Network, Modified Capsule Neural Network, Capsules, Pooling layer, Scene Recognition.

I. INTRODUCTION

Intelligent Assistive Systems (IAS) to aid infirm and/or disabled people are going to play a vital role in the coming future. A United Nations' report on the world population clearly states that by 2050 the population of people over 60 and 85 will double and quadruple respectively [1, 2]. A large number of this population will choose to live alone at home and therefore are likely to require assistance for their needs and monitoring for their well-being – hence a need for Intelligent Assistive Systems to provide such services. For IAS to assist people in many and diverse ways, they must be able to classify their indoor home environment. At present, the accuracy of recognizing indoor scenes is very poor when compared to outdoor scene recognition [3]. The major reason behind this is the similarities that exist between different indoor scenes. Therefore, if IAS are going to have an impact, it becomes important to rectify such issues and improve the accuracy in indoor home scene recognition.

It should be noted that an additional factor that currently inhibits better performance in indoor scene recognition is the lack of sufficient data for indoor home scenes. This is significantly less when compared to outdoor scenes, thus restricting neural network training. Therefore, a neural network that can counteract the aforementioned issues and produce better accuracy with less data is currently required.

There are some neural networks that have been used in the past for scene recognition like Multi-Resolution Convolutional Neural Network (MR-CNN) [4] which reported an accuracy of 86.7% and 72% on the MIT67 indoor [3] and the SUN397 [5] datasets, respectively. Places-CNN trained on a dataset containing millions of images [6], and Unified CNN performs both scenes and objects recognition simultaneously [7], etc. An expert system has also been developed by P. Espinace et al. [8, 9] which, on the basis of object detection predicts the scenes. However, there are no neural networks which have been specifically designed for indoor home scene recognition and, also, capable of object recognition at the same time. However, all the results produced in the mentioned works used a dataset that has millions of data for training the neural networks. Additionally, the dataset used for scene recognition had both interior and exterior scenes i.e. no dataset particularly dedicated to indoor home scene recognition was used.

CapsNets have never been used for scene recognition tasks, but have been used for object recognition mainly due to their ability to recognise objects irrespective of orientation. The challenge therefore in this work was to evaluate whether CapsNets (or modified versions) could be used for indoor scene recognition to high enough accuracy levels and whether CapsNet is useful in acquiring better accuracy even with a smaller dataset. Armed with the knowledge provided by Sabour et al. [11] where a standard CapsNet did not meet

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

expectations a modified CapsNet network was developed for this study.

II. CAPSULES

Capsules are the most important part of CapsNet [10]. The idea behind the invention of capsules is to eradicate the problems associated with CNNs, like failing in capturing the proper orientation of anything and loosing important information during the transfer of information from convolutional layers to pooling layers.

Capsules have three basic functions to perform shown in figure 1,

- Affine Transformation: This helps in capturing the actual orientation using the features extracted by the convolutional layer. The mathematical formula is,

$$\hat{u}_{j|i} = W_{ij}\, u_i \quad (1)$$

Where, $\hat{u}_{j|i}$ – Prediction vectors, W_{ij} – Weight matrix and u_i – Output of a capsule

- Weighted Sum: This is similar to the sum of weights that happens in deep neural networks (DNN). The mathematical formula is,

$$s_j = \sum_i c_{ij}\, \hat{u}_{j|i} \quad (2)$$

Where, c_{ij} – Coupling coefficient

- Squash: This is a non-linear activation function which takes the input in a vector format and resizes it to a unit length vector without changing its direction. The mathematical formula is,

$$v_j = \frac{||s_j||^2}{1+||s_j||^2} \quad \frac{s_j}{||s_j||} \quad (3)$$

Where, v_j - Output vector of capsule j, s_j – Total input of capsule j and $\frac{||s_j||^2}{1+||s_j||^2}$ – squashing and $\frac{s_j}{||s_j||}$ - unit scaling.

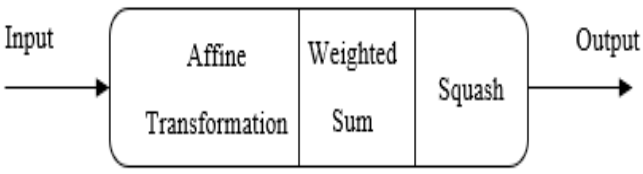

Fig. 1. A Capsule with all its functions

It is clear, therefore, that the resizing task in CapsNet is done by the squash function which ensures that the memory of the network is not exceeded. The same task in a CNN is performed by the pooling layer, in which the spatial size of the information extracted by the convolutional layer is reduced which reduces the total number of parameters. This in return takes care of the learning speed and memory size of CNN.

In the implementation of the work presented in this paper, the squash function is removed and a max-pooling layer with a convolutional layer is introduced. In other words, instead of using squash function for suitable resizing of the data, a max-pooling is used. Only the affine transformation for acquiring the proper orientation information is used along with the weighted sum as shown in figure 2.

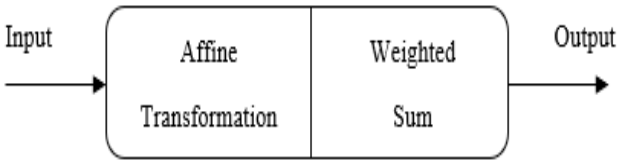

Fig. 2. A modified capsule without squash function

III. CAPSULE NEURAL NETWORK (CAPSNET)

Sara Sabour et al. [11] introduced the CapsNet architecture whose schematic diagram is shown in figure 3. The architecture consists of a convolutional layer that has a kernel size of 256 and a stride of 1. The output of the convolutional layer, which is then connected to the primary capsule, supplies the input for this primary capsule.

The number of capsules used is 32 with 8 channels. The output of the primary capsule is connected to the input of digit capsules. The number of digit capsules used in this case is 10 as the dataset used for training the CapsNet is MNIST dataset that has 10 labels and has 8 channels. Then the output of the digit capsule (Digitcap) is connected to the fully connected layer. There are 3 layers of fully connected (FC) layer in this case with a different number of neurons. The first FC layer has 512 neurons, the second FC layer has 1024 neurons and the third FC layer has 784 neurons.

All the FC layers have ReLU (Rectified Linear Unit) [12] activation functions. The accuracy achieved on the MNIST dataset using the CapsNet is around 73%. The number of routings used in the architecture is 3.

The calculation of loss is done using the following mathematical formula,

$$L_c = T_c \max(0, m^+ - ||v_c||)^2 + \lambda\,(1 - T_c) \max(0, ||v_c|| - m^-)^2 \quad (4)$$

Where, L_c – Loss term for one DigitCap, T_c - Loss function of DigitCap, λ – Coefficient used for numerical stability and its value is fixed at 0.5. $T_c \max(0, m^+ - ||v_c||)^2$ Calculates the loss for correct digitcap, i.e. when, T_c is 1 and $\lambda\,(1 - T_c) \max(0, ||v_c|| - m^-)^2$ calculates the loss for incorrect digitcap, i.e. when , T_c is 0.

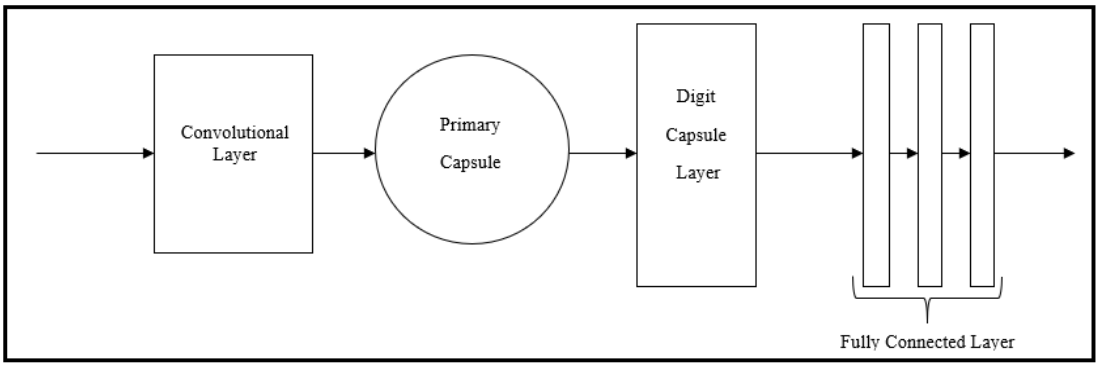

Fig. 3. Capsule Neural Network

IV. Implementation of Work

To implement the scene recognition using CapsNet the indoor home scene data is taken from the Palces365 dataset. Five indoor home scenes that are taken from the Places365 dataset are bedroom, bathroom, kitchen, dining room and living room. Each scene has 5000 images. 4000 images in each category were used for training and the rest 1000 from each category were used for testing. In total, 20,000 and 5000 images were used for training and testing respectively. Each Image had a size of 256X256 and was present in RGB. To reduce the number of parameters all the images were resized to 128X128 and then were converted to the grayscale image from RGB which is shown in figure 4. The Grayscale conversion of the images is done so that the number of channels to be used becomes 1 which again drastically reduces the total number of trainable parameters for CapsNet. The whole dataset was then fed into the CapsNet shown in figure 3. It should be noted that the number of DigitCaps, in this case, changes to 5 as the dataset has only 5 categories.

Fig. 4. The conversion of 256X256 (RGB) image to 128X128 (Grayscale) image

The same converted data is then fed into the Mod-CapsNet whose architecture is shown in Figure 5. In Mod-CapsNet, there are convolutional layers, max-pooling layers, primary capsule, digit capsules, 2 FC Layers and, finally, a sigmoid function. There are three convolutions in each convolution layer. These convolution layers are not connected to each other but are concatenated. The output after concatenation is fed to the Max-Pool layer. The number of filters of the first, second, third, fourth and fifth three convolutional layers' combination is 64, 128, 256, 512 and 512 respectively. The channel size for all the convolutional layers is 3X3, and a stride of 1 is used for these layers.. Each max pool layer has a pool size of 2X2. The stride for all max pool layers is 2 in this architecture. The output from the fifth three convolutional layers' is fed to the primary capsule that has no squash function. The number of primary capsules used is 32 along with 16 channels, strides 2 and valid padding. Then the output of the primary capsule is taken as input by the DigitCaps which is 5 in number and has 16 channels. The FC Layers take all the information into them after DigitCaps. The first and second FC Layers have 4096 neurons respectively and the third FC Layer has 5 neurons. Both FC Layers have ReLU activation function. Finally, the sigmoid function is used for the

proper recognition of the scenes' category. The Mod-CapsNet specifications discussed are shown in table 1.

The primary capsule with squash was also trained with the same dataset and the architecture is shown in figure 4. All the parameters discussed were the same in this case as it was in the primary capsule without squash.

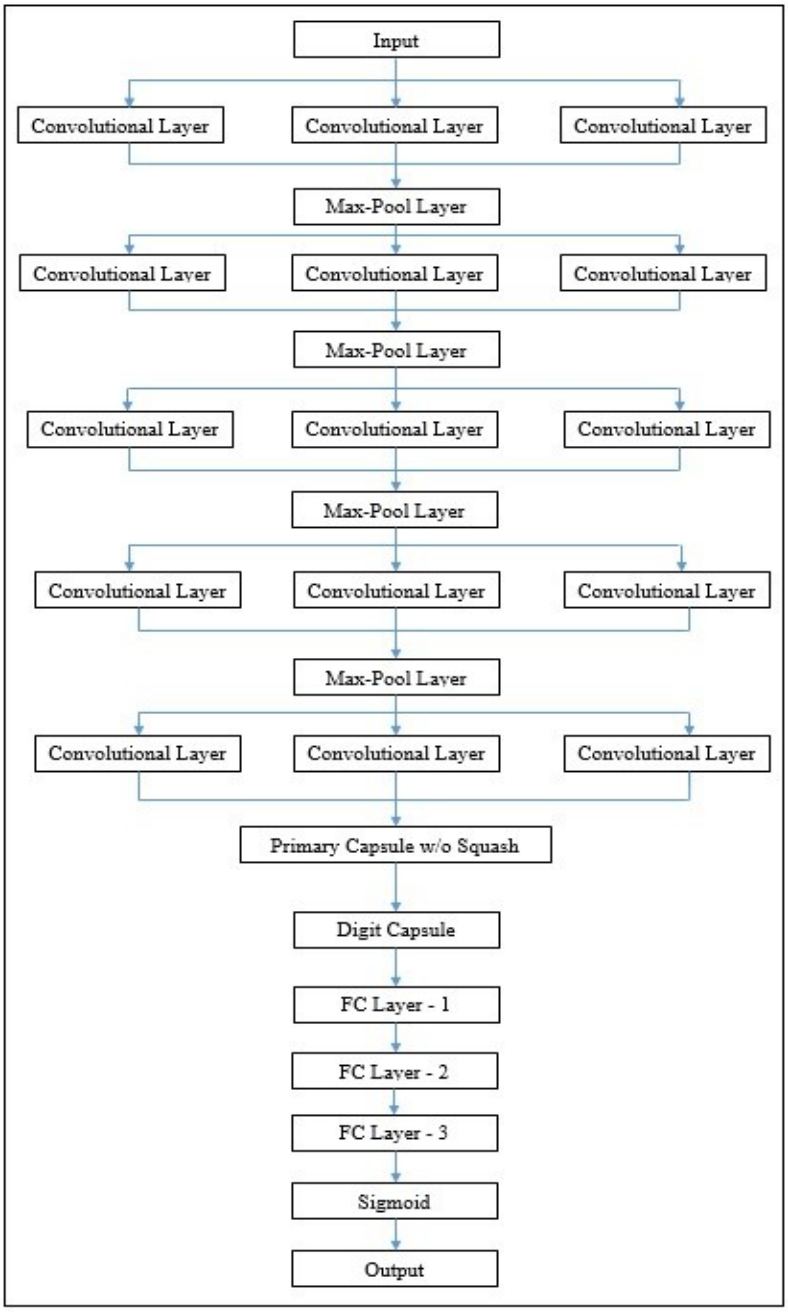

Fig. 5. Modified Capsule Neural Network (Mod-CapsNet)

The filter number is kept in ascending order so that the convolutional layer with a smaller number of filters can extract the local information from the inputs whereas the convolutional layer with a higher number of filters can extract the higher level of information from the inputs. Padding in every case is kept same so that the output size is equivalent to the input size. Strides in every case of convolutional layer is 1 so as to minimize the pixel skip. Max-pool layer is used to downsample the representation of the input. The

channel size of every max-pool layer is kept the same so that there is uniformity in the input and output information. Additionally, channel size in both convolutional layers and max-pool layers is kept small. Keeping the channel size small helps capturing the smaller and complex features of the data. This is important for indoor home scenes which contain a lot and complex features. The number of capsules is also selected based on the complexity of the information, i.e. more capsules required as complexity increases. However, it must be noted that increasing the number of capsules implies large increases in parameters which may also lead to slow training and reduced efficiency. Hence a compromise must be found. . The number of DigitCaps is 5 as there are only 5 classes present in the dataset. For the FC layers the number of neurons chosen on the basis of feature space of a problem. However, the exact number that may work for the developed neural network depends ultimately on thorough experimental analysis.

TABLE I. MOD-CAPSNET SPECIFICATION TABLE

Layer	Name	Specifications	Strides
Input		Image size: 128X128 (Grayscale)	-
Layer-1	Conv 1_1 Conv 1_2 Conv 1_3	Output Concatenated Filters: 64 Channel Size: 3X3 Padding: Same	1
Layer-2	Max-Pool	Channel Size: 2X2 Padding: Same	2
Layer-3	Conv 2_1 Conv 2_2 Conv 2_3	Output Concatenated Filters: 128 Channel Size: 3X3 Padding: Same	1
Layer-4	Max-Pool	Channel Size: 2X2 Padding: Same	2
Layer-5	Conv 3_1 Conv 3_2 Conv 3_3	Output Concatenated Filters:256 Channel Size: 3X3 Padding: Same	1
Layer-6	Max-Pool	Channel Size: 2X2 Padding: Same	2
Layer-7	Conv 4_1 Conv 4_2 Conv 4_3	Output Concatenated Filters: 512 Channel Size: 3X3 Padding: Same	1
Layer-8	Max-Pool	Channel Size: 2X2 Padding: Same	2
Layer-9	Conv 5_1 Conv 5_2 Conv 5_3	Output Concatenated Filters: 512 Channel Size: 3X3 Padding: Same	1
Layer-10	Primary capsule	Number of capsules: 32 Number of channels: 16 Strides: 2 Padding: Valid	-
Layer-11	Digit Capsule	Number of capsules: 5 Number of channels: 16 Routings: 3	-
Layer-12	FC Layer -1	Number of Neurons: 4096	-
Layer-13	FC Layer-2	Number of Neurons: 4096	-
Layer-14	FC Layer-3	Number of Neurons: 5	
Layer-15	Sigmoid Function		-

V. RESULTS

As already mentioned, for training and testing the different CapsNets 20,000 images and 5000 images were used respectively. These were equally distributed under 5 categories. According to table 2, Sabour et al. CapsNet design was first used for the task. The validation and testing accuracies produced in this case were 20% and 17.2% respectively. This is very low and it is not considered acceptable for the task of indoor home scene recognition.

Secondly, Mod-CapsNet with squash function and the network having pooling layers in it produced validation and testing accuracy results of 64.7% and 64% respectively.

Finally, a Mod-CapsNet architecture without squash function is used which produces validation and testing accuracies of 70.8% and 70% respectively.

From these results, it is clear that a Mod-CapsNet without squash function and with pooling layers produces the best accuracy rate among the three CapsNets.

TABLE II. VALIDATION AND TESTING ACCURACY OF CAPSNETS

Neural Network	Validation Accuracy (%)	Testing Accuracy (%)
Sabour et al. CapsNet	20	17.2
Mod-CapsNet with squash function	64.7	64
Mod-CapsNet without squash function	**70.8**	**70**

VI. CONCLUSION

As per the analysis and results produced by different CapsNets, it is clear that the Mod-CapsNet without squash and having pooling layers in the architecture with convolutional layers works better in learning and recognising the indoor home scene.

Using the standard Capsules along with the convolutional layer failed completely to learn anything about the dataset. Sabour et al. CapsNet accuracies show that the network failed to capture the information and learnt almost nothing. When the CapsNet is modified and Mod-CapsNet is constructed with the same primary capsules the accuracy drastically increases. Therefore, this shows the pooling layers help the capsules to learn better. Also, a smaller dataset (only 20,000 images for training) produced considerably better accuracy,

However, as the function of the pooling layer and squash is almost the same, when the squash is removed and the Mod-CapsNet without squash function in the primary capsule is used, then the accuracy improves even more. Therefore, it can be concluded that the affine transformation and weighted sum in primary capsules work better with the information processed through a combination of convolutional layers and max-pooling layers.

However, a further increase in the accuracy of CapsNet in indoor home scene recognition could be achieved by making some structural changes in the architecture. This is still a research issue and further investigation is pending.

REFERENCES

[1] M. E. Pollack, "Intelligent assistive technology: the present and the future," 2007: Springer, pp. 5-6.

[2] M. E. Pollack, "Intelligent technology for an aging population: The use of AI to assist elders with cognitive impairment," *AI magazine,* vol. 26, no. 2, p. 9, 2005.

[3] A. Quattoni and A. Torralba, "Recognizing indoor scenes," 2009: IEEE, pp. 413-420.

[4] L. Wang, S. Guo, W. Huang, Y. Xiong, and Y. Qiao, "Knowledge guided disambiguation for large-scale scene classification with multi-resolution CNNs," *IEEE Transactions on Image Processing,* vol. 26, no. 4, pp. 2055-2068, 2017.

[5] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," 2012: IEEE, pp. 2751-2758.

[6] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE transactions on pattern analysis and machine intelligence,* 2017.

[7] L. Wang *et al.*, "A Unified Optimization Approach for CNN Model Inference on Integrated GPUs," 2019: ACM, p. 99.

[8] P. Espinace, T. Kollar, A. Soto, and N. Roy, "Indoor scene recognition through object detection," 2010: IEEE, pp. 1406-1413.

[9] P. Espinace, T. Kollar, N. Roy, and A. Soto, "Indoor scene recognition by a mobile robot through adaptive object detection," *Robotics and Autonomous Systems,* vol. 61, no. 9, pp. 932-947, 2013.

[10] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," 2011: Springer, pp. 44-51.

[11] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," 2017, pp. 3856-3866.

[12] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," 2010, pp. 807-814.