Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2019) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2019)

Indoor Home Scene Recognition Using Capsule Neural Networks

Amlan Basu[a]*, Lykourgos Petropoulakis[b], Gaetano Di Caterina[c], John Soraghan[d]

[a]*Ph.D. Student, University of Strathclyde, Glasgow G1 1XQ, United Kingdom*
[b]*Senior Knowledge Exchange Fellow, University of Strathclyde, Glasgow G1 1XQ, United Kingdom*
[c]*Research Fellow, University of Strathclyde, Glasgow G1 1XQ, United Kingdom*
[d]*Professor, University of Strathclyde, Glasgow G1 1XQ, United Kingdom*

Abstract

This paper presents the use of a class of Deep Neural Networks for recognizing indoor home scenes so as to aid Intelligent Assistive Systems (IAS) in performing indoor services to assist elderly or infirm people. Identifying exact indoor location is important so that objects associated with particular tasks can be located speedily and efficiently irrespective of position or orientation. In this way, IAS developed for providing services may become more efficient in accomplishing designated tasks satisfactorily. There are many Convolutional Neural Networks (CNNs) which have been developed for outdoor scene classification and, also, for interior (not necessarily indoor home) scene classification. However, to date, there are no CNNs which are trained, validated and tested on indoor home scene datasets as there appears to be an absence of sufficiently large databases of home scenes.

Nonetheless, it is important to train systems which are meant to operate within home environments with the correct relevant data. To counteract this problem, it is proposed that a different type of network is used, which is not very deep (i.e., a network which does not have too many layers) but which can attain sufficiently high classification accuracy using smaller training datasets. A type of neural network likely to help achieve this is a Capsule Neural Network (CapsNet). In this paper, 20,000 indoor home scenes were used for training the CapsNet, and 5000 images were used for testing it. The validation accuracy achieved is 71% and testing accuracy achieved is 70%.

Keywords: Convolutional Neural Network; Capsules; Intelligent Assistive Systems; Scene Recognition

* Corresponding author. Tel.: +44-7459802138.
E-mail address: amlan.basu@strath.ac.uk

1. Introduction

Scene recognition using neural networks is one of the most pioneering tasks and challenging areas in the field of computer vision. This is because of the complexity and also the inconsistency in the results obtained. In general, the classification accuracy of indoor scenes is substantially lower when compared to outdoor scenes. This is primarily due to the similarity between different indoor scenes, which causes the classification accuracy to be substantially lower compared outdoor scenes [1]. For instance, the presence of a table both in a living room and also in a bedroom makes it more difficult for a neural network to uniquely categorise the scenes.

Moreover, the lack of sufficiently large datasets available for indoor scenes exacerbates the reasons behind the reduced accuracy. So, solving this problem can either involve increasing the data size for indoor scenes or deploying neural networks capable of providing higher levels of accuracy using smaller datasets. Improving IAS ability to classify indoor scenes is essential if IAS deployment to assist the elderly or infirm people during daily tasks or in case of emergencies, is to be efficient.

In emergencies, it is important for IAS to know the exact location of individuals requiring assistance, along with exact required procedures keeping in mind the possible situations that may occur during an emergency. Therefore, it becomes important to make the system intelligent so that it can correctly identify the different locations inside the home as well as the location of individuals within it and covey this information to the emergency services.

Identifying the indoor areas of a house in order to provide services for elderly people, has become more important. This is because according to a study carried out by the United Nations on future population, the percentage of people above the age of 60 and 85 will be 21.4% and 4.2% respectively by 2050, i.e., double and quadruple the present percentage with an increased proportion selecting to live alone in their own homes [2, 3].

Limin Wang et al. [6] developed Multi-Resolution CNNs (MR-CNN), which is a combination of coarse and fine resolution CNNs and are said to be complementary to each other. The network is trained using ImageNet, Places and Places2 datasets [7]. The trained network achieved an accuracy rate of 86.7% and 72% on MIT67 indoor dataset and SUN397 [8] dataset, respectively. It achieved the second position in Places challenge in ILSVRC 2015[†] and secured first place in LSUN challenge in CVPR 2016. These are image classification challenges that are organized every year to encourage the researchers of computer vision to come up with different neural network architectures that have a higher precision level for image classification and scene recognition. Challenges like this also help to keep track of work going on in this field and opens opportunities for improving image classification and scene recognition ability for intelligent systems. The testing accuracy attained by the CNNs trained using Places dataset while tested on MIT67 indoor dataset were, AlexNet – 70.72%, GoogleNet – 75.14% and VGG – 79.76%. However, the obtained results are for 67 different scenes. P. Espinace et al. [9, 10] performed scene recognition using object detection. The way to achieve this is by using the adaptive search method. The complete technique depends on the database available with the system. Bolei Zhou et al. [7] presented the Places dataset, which contains 10 million images. Different CNNs like AlexNet, VGG, GoogleNet, and ResNet were trained using the dataset. All the trained CNNs were tested on different datasets. VGG trained using Places-205 dataset achieved the highest accuracy of 93.33% when tested on Event8 dataset.

More recently, a neural network architecture, known as Capsule Neural Network (CapsNet), has emerged as a candidate to help resolve some of the aforementioned issues. CapsNets appear to have higher levels of efficiency in providing good accuracy using lesser amounts of data and a less deep architecture. In this paper, to test this premise and to compare various neural network performances, a set of different neural networks are trained and tested using the MIT Places365 dataset (indoor home scenes). The neural networks used are the Faster and Fast RCNN (Region based Convolutional Neural Network), which are deployed with VGG 16 [4] and VGG 19, respectively. The RCNN used was first introduced by Ross Girshick et al. [5]. The mask RCNN used is deployed with ResNet-152 as its core CNN and, finally, a CapsNet was used. This selection of networks was made so that the performance of the CapsNet can be compared to CNNs known to perform well in image classification and scene recognition.

[†] ImageNet Large Scale Vision Recognition Challenge (ILSVRC) is a computer vision challenge based on ImageNet dataset first introduced in 2009. (http://www.image-net.org/challenges/LSVRC/)

2. Convolutional Neural Networks (CNNs)

CNNs are the most widely used neural networks in the field of computer vision. This neural network is created using convolutional layers and fully connected layers. However, there are many CNNs which do not have fully connected layers and are known as Fully Convolutional Neural Networks (FCNs).

The first CNN, known as LeNet [11], came into existence in 1998 by Yann LeCun. Then, in 2012, AlexNet [12] was introduced, which achieved substantial accuracy and was able to capture the attention of researchers and institutes working on computer vision. Since then, many CNNs have appeared, and the one with the best accuracy, in general, is ResNet-152 [13-15].

It is known that in CNNs the main layer is the convolutional layer (which is responsible for extracting the features from data) and along with it is the fully connected (FC) layer (responsible for keeping the information of all the learnt features). However, just convolutional layers and FC layers are not enough to form the complete CNN architecture as there are many computational requirements associated with it. To fulfill the computation requirements, many other layers and functions are introduced to build a complete CNN architecture.

In a CNN architecture, there are also activation functions applied along with convolutional layer so that only useful information is acquired helping to increase computation speed. The use of pooling layers help to solve memory size problems so that the information acquired is shrunk to a lower memory size along with the complete information acquired.

However, a CNN architecture cannot reveal the basic relationships between the features even if the FC layers have complete information of all the extracted features. This is because none of the CNN layers is tasked with learning the relationship between the features present in a given image. For example, a CNN presented with the face of a person may learn the features of the face such as ears, nose, lips, eyes, etc., but it will not learn the proper orientation or specific arrangements which link these features, i.e., on which part of the face these features belong to. In many cases, therefore, this can inevitably lead to wrong evaluation. Since in indoor home scenes it is very important to know the relationship between objects, CNN becomes a less efficient architecture for this task.

CNNs generally require large amounts of data to produce high accuracy, and at present, there is a lack of such large datasets relating to home indoor scenes. It is also important thing to know that CNNs are incapable of performing inverse graphics[‡] process as a human brain does; rather, they perform rendering[§] as computers do.

In addition, CNNs lack the capability of equivariance which helps to identify even those images which are not closely related to the training dataset. This capability becomes useful when there is a requirement of a versatile system which can work in slightly varying environments without requiring retraining. For example, if a person changes residence and deploys the same assisting system which was used in a previous residence then, even though the two environments may not be identical, the system can work efficiently without modifications.

3. Capsules

To overcome the problems in CNNs, capsule networks (or Capsules) were introduced by Geoffrey E. Hinton et al. [16]. The Capsules are designed in such a way that it can efficiently extract the information of the link between the features and also resize the image without loosing the important information.

The Capsules are equipped with affine transformation which helps to capture the information of features supplied by the convolutional layer, learns the orientation of the features and, most importantly, it learns the correlation between the different extracted features.

This property makes Capsules capable of capturing information on 3-Dimensional images. So, like a human, if the network is presented with an image scene taken form a given viewpoint, it is able to recognize the same image scene when presented from a different viewpoint. This is one of the reasons why capsules require less data to obtain sufficiently high levels of accuracy.

To control the data size in CapsNets the pooling layer used in CNNs is replaced with a squashing technique. Squashing ensures that the data is resized and that no information during this process is lost, unlike pooling layers.

[‡] Inverse graphics is the process in which features of the viewed objects are matched to existing learnt patterns irrespective of the viewing angle.

[§] In rendering the recognition is mostly dependent on the stored geometrical data of an object in form of arrays and matrices. These are used for representing the orientation and relative position of objects by matching it with the already stored geometrical data in the memory to produces an image.

This is an important process for increasing the computational speed of the network and managing the memory of the network from getting overflowed. A simple capsule overview architecture diagram is shown in figure 1.

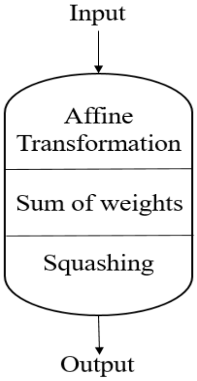


Fig.1. A simple capsule diagram

The mathematical formulae [17] for the processes involved in the capsule are as follows,
For affine transformation,

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

Where,

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

Sum of weights,

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

Where,

- c_{ij} – Coupling coefficient

Non-Linear Activation (Squashing),

$$v_j = \frac{||s_j||^2}{1+||s_j||^2}\ \frac{s_j}{||s_j||} \tag{3}$$

Where,

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

4. Methodology

The Places365 dataset, with image size 256×256, was reshaped to 128x128 and was used for the entire process. In this case, 5 different categories of the indoor home scenes were chosen from which 20,000 images were used for training the network, and 5,000 images were used for testing the trained network.

The neural networks deployed for the task are CapsNet, Fast RCNN [5, 18] with VGG[4] as CNN, Faster RCNN [19] with VGG as CNN and Mask RCNN [20] with ResNet-152[21] as CNN. The CapsNet used for the task is the one introduced by S. Sabour et al. [17].

The CapsNet has 6 layers: a convolutional layer, a primary capsule layer, a digit capsule layer and 3 fully connected (FC) layers. The convolutional layer has 256 filters, kernel size of 9 and 1 stride along with a ReLU (Rectifier Linear

Unit) activation function. There are 32 capsules in the primary capsule layer with 64 channels, kernel size of 10 and 4 strides. The Digit capsule layer has 5 digit capsules as there are 5 indoor home scenes (bedroom, bathroom, kitchen, living room and dining room) used for classification. The first and second FC layers have 512 and 1024 neurons respectively along with a ReLU activation function [22]. The last FC layer has 728 neurons with a sigmoid activation function.

The loss occurring in the CapsNet is calculated using the following equation:

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

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.
- $\lambda (1 - T_c) \max(0, ||v_c|| - m^-)^2$ Calculates the loss for incorrect digitcap, i.e. when , T_c is 0.

The complete process was carried out using the python programming and hardware support of 4 GeForce GTX 1080Ti GPU was taken to speed up the process. The training process may take longer because of the higher number of parameters and also the presence of routing in capsules. A routing process helps to connect the lower level capsule to a higher level capsule. The dynamic routing algorithm is explained in [17]. Training speed mostly depends on the hardware availability, i.e., the speed of the process increases with the number of GPUs available.

Since, in this case, there was no requirement for object detection, the bounding box property for the Fast, Faster, and Mask RCNNs was not used. These CNNs also have RPN (Regional Proposal Network), FCN (Fully Convolutional Network) and RoI (Region of Interest) align layers, which help to improve performance; these layers were used in this study so as to maximize the performance of these networks.

5. Results and Discussion

Figure 2 shows the architecture of a capsule neural network used for indoor home scene recognition. The features are first extracted from the data using the convolutional layer and then sent to the primary capsules which, by using the affine transformation, obtain the information on how the different extracted features are associated with each other. The squashing process resizes the data, which is then stored, in the form of digits, in the digit capsule layer. Here, 5 digit capsules are used as there are 5 different classifications. Then, using the FC layers which act as decoders the information is stored in the units of the FC layers for completing the classification task.

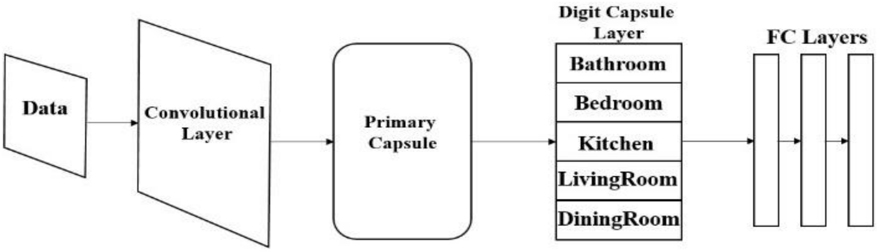


Fig. 2. Capsule Neural Network used for scene recognition

Figure 3 shows the architecture of Faster RCNN. The data first goes through a feature extraction process using a CNN, which in this case, is a VGG-16. This results in feature maps the information of which is then fed into Regional Proposal Network (RPN). The latter helps to focus only on those points, which are important to acquire for proper classification. These points are then accumulated by an RoI (Region of Interest) pooling layer, and they are sent to FC layers where the accumulated information is stored. These FC layers are connected with a classifier, which in this case is softmax to produce the required output. The softmax function helps to generate answers with different probability, and the answer with the highest probability is selected.

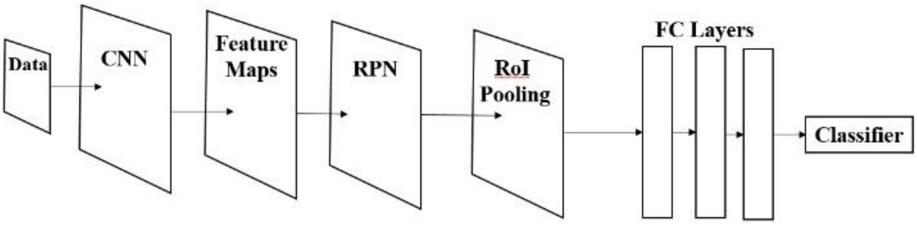


Fig. 3. Faster RCNN used for scene recognition

Figure 4 shows the architecture of Fast RCNN. The function of the layers is the same as discussed for Faster RCNN. However, the CNN used in this architecture is a VGG-19, and this architecture does not have an RPN.

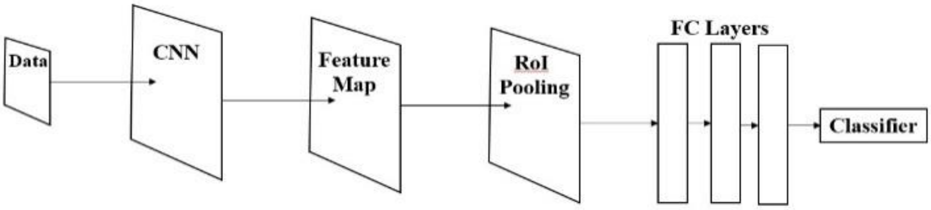


Fig. 4. Fast RCNN used for scene recognition

Figure 5 shows the architecture of Mask RCNN. The CNN used in this network is a ResNet-152. The extracted feature maps by CNN and the output of RPN are fed together to RoI align. RoI pooling is replaced with RoI align in this network as, not only it digitizes the cell boundaries and equalizes the size of target cells, but it also helps to calculate the values of feature maps using interpolation properly.

In the CNNs bounding boxes and mask were not used as there was no requirement for object detection, semantic segmentation and instantaneous segmentation.

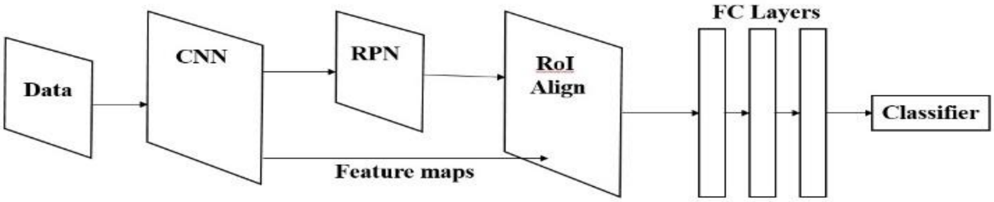


Fig. 5. Mask RCNN used for scene recognition

The trained data is validated to understand how good the model works on the given data to get a valid classification. From table 1, the following information can be obtained:

The trained model of CapsNet gives an accuracy of 71% with an error rate of 29%. For Faster RCNN, the accuracy is 76.9%, and the error rate is 23.1%. For Fast RCNN, the accuracy is 75.3%, and the error rate is 24.7%. For Mask RCNN, the accuracy is 68.6%, and the error rate is 31.4%. Thus, among the trained models, the Faster RCNN model displays the best performance on the data used. The accuracy and error percentages discussed are related to the training of the neural networks used. The testing accuracy for CapsNet, Mask RCNN, Fast RCNN, and Faster RCNN are 70%, 67.6%, 73.6%, and 74.2% respectively.

Table 1. The validation and testing accuracy for different deployed neural networks using 20,000 images for training and 5000 images for testing

Neural Network	Validation Accuracy	Testing Accuracy
Mask RCNN	68.6%	67.6%
CapsNet	**71%**	**70%**
Fast RCNN	75.3%	73.6%
Faster RCNN	**76.9%**	**74.2%**

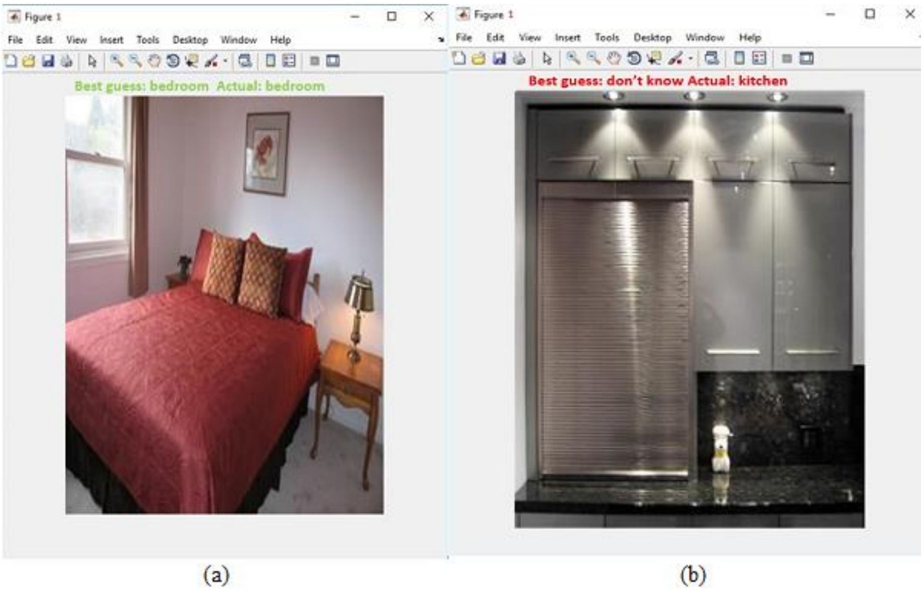
To test the premise that CapsNets could be trained equally well with smaller datasets, the same CapsNet architecture was trained using, 5000 images which were equally distributed in 5 different categories, and the resulting validation accuracy remained the same at 71%. By contrast, under the same conditions, the validation accuracies for Fast RCNN, Faster RCNN, and Mask RCNN reduced to 67%, 69.1%, and 66.2% respectively and testing accuracies reduced to 66%, 68.2% and 64.4% respectively. The results of these tests are shown in table 2. This indicates that, unlike other networks, CapsNets can produce good results even with relatively small datasets. These results are comparable to results produced by other neural networks requiring a lot more data and training time. The CapsNet accuracy was only reduced to a very low level, i.e., 52% when trained using only 1500 images.

Table 2. The validation and testing accuracy for different deployed neural networks using 5000 images for training and 1250 images for testing

Neural Network	Validation Accuracy	Testing Accuracy
Mask RCNN	66.2%	64.4%
CapsNet	**71%**	**70%**
Fast RCNN	67%	66%
Faster RCNN	**69.1%**	**68.2%**

If the accuracy rates shown in table 1 are compared with the accuracy rate of other available work on the same topic (already discussed in the introduction section), the presented Capsule Network displays a good performance. As a comparison, the MR-CNN trained on MIT Places365 dataset, which has 365 categories and 10 million images, was able to produce an accuracy of 72% when tested on the MIT67 indoor dataset [6].

In Figure 6, an illustration of a classification performed using the trained CapsNet can be seen. Figure 6 (a) is a bedroom scene (not used in the training set), which the CapsNet predicted correctly. The message appears in green along with the correct answer. Figure 6 (b) is an image of a kitchen scene (used in the training set) which the trained CapsNet was unable to classify correctly. In this case, a threshold value has been imposed, indicating the minimum degree of classification confidence the Capsule Net has. A classification below this value results in an output of "don't know" indicating that the credibility of the classification is too low to be considered as valid. In this case, the threshold value was set at 65%. It is important to note that this approach can only be applied to images which have been used for training (i.e., the network knows what the expected output should be).



Best guess: bedroom Actual: bedroom

Best guess: don't know Actual: kitchen

Figure 1
File Edit View Insert Tools Desktop Window Help
(a)
Figure 1
File Edit View Insert Tools Desktop Window Help
(b)

Fig. 6. Images tested for image classification. (a) The bedroom scene (image not used in the training set) was tested in which the trained neural network was able to predict it correctly (b) A kitchen scene (image used in the training set) which the neural network was unable to recognize.

6. Conclusion

In this paper, a set of four neural networks were trained using 20,000 images and tested on 5000 images and their classification performance was compared. The images were evenly distributed under 5 categories on indoor home scenes. Three of these networks were CNN-based while the fourth was a Capsule neural network which has fewer layers and a less complex architecture than the CNNs. The tests indicated that all networks produced comparable results (with an approximately 8% difference between the best and the worst performances).

The same networks were also trained with a reduced size dataset (only 5000 images) to test their ability to learn with reduced datasets. Unlike the three CNN-based networks whose performance was reduced substantially, the Capsule network retained the same performance level in this case. Only when the dataset was reduced to 1500 images, the performance of the Capsule network dropped to unacceptable levels; this is attributed to the reduction in the minimum required information

Given the more complex structure of the CNN-based networks which have many layers along with their special employed techniques, such as batch normalization, it can be argued that the CapsNet has produced a very comparable classification accuracy rate despite its less deep architecture. It has also been shown that a CapsNet could be more useful when large datasets are not readily available for training.

The reasons behind the slightly lower accuracy reported by the CapsNet as compared to Faster RCNN may be because of the higher number of parameters resulting in dying ReLUs and gradient explosion/vanishing. The accuracy of the CapsNet may increase if there are proper changes made in its architecture. These are research issues, and they are under investigation.

Acknowledgments

We are grateful to the University of Strathclyde, Glasgow, for all the required resources for the work, and we are also thankful to our deep learning team for their unconditional support.

References

[1] A. Quattoni and A. Torralba (2009), "Recognizing indoor scenes," *IEEE Conference on Computer Vision and Pattern Recognition*: 413-420.

[2] M. E. Pollack (2007), "Intelligent assistive technology: the present and the future," *International Conference on User Modeling Springer* **4511**: 5-6.

[3] M. E. Pollack (2005), "Intelligent technology for an aging population: The use of AI to assist elders with cognitive impairment," *AI magazine* **26 (2)**: p. 9.

[4] K. Simonyan and A. Zisserman (2014), "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik (2016), "Region-based convolutional networks for accurate object detection and segmentation," *IEEE transactions on pattern analysis and machine intelligence* **38 (1)**: 142-158.

[6] L. Wang, S. Guo, W. Huang, Y. Xiong, and Y. Qiao (2017), "Knowledge guided disambiguation for large-scale scene classification with multi-resolution CNNs," *IEEE Transactions on Image Processing* **26 (4)**: 2055-2068,.

[7] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba (2017), "Places: A 10 million image database for scene recognition," *IEEE transactions on pattern analysis and machine intelligence* **40(6)**: 1452-1464.

[8] G. Patterson and J. Hays (2012), "Sun attribute database: Discovering, annotating, and recognizing scene attributes," *IEEE Conference on Computer Vision and Pattern Recognition*: 2751-2758.

[9] P. Espinace, T. Kollar, A. Soto, and N. Roy (2010) "Indoor scene recognition through object detection," *IEEE International Conference on Robotics and Automation*: 1406-1413.

[10] P. Espinace, T. Kollar, N. Roy, and A. Soto (2013), "Indoor scene recognition by a mobile robot through adaptive object detection," *Robotics and Autonomous Systems* **61 (9)**: 932-947.

[11] Y. LeCun, K. Kavukcuoglu, and C. Farabet (2010), "Convolutional networks and applications in vision," *IEEE International Symposium on Circuits and Systems*: 253-256.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton (2012), "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing* systems: 1097-1105.

[13] K. He, X. Zhang, S. Ren, and J. Sun (2016), "Deep residual learning for image recognition," *IEEE conference on computer vision and pattern* recognition: 770-778.

[14] K. He, X. Zhang, S. Ren, and J. Sun (2015), "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *Proceedings of the IEEE international conference on computer vision*: 1026-1034.

[15] K. He, X. Zhang, S. Ren, and J. Sun (2016), "Identity mappings in deep residual networks," *European conference on computer vision Springer* **9908**: 630-645.

[16] G. E. Hinton, A. Krizhevsky, and S. D. Wang (2011), "Transforming auto-encoders," *European conference on computer vision Springer* **6791**: 44-51.

[17] S. Sabour, N. Frosst, and G. E. Hinton (2017), "Dynamic routing between capsules," *Advances in neural information processing systems*: 3856-3866.

[18] R. Girshick (2015), "Fast r-cnn," *Proceedings of the IEEE international conference on computer vision*: 1440-1448.

[19] S. Ren, K. He, R. Girshick, and J. Sun (2017), "Faster R-CNN: towards real-time object detection with region proposal networks," *Advances in neural information processing systems*: 91-99.

[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick (2017), "Mask r-cnn," *IEEE international conference on computer vision*, 2961-2969.

[21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi (2017), "Inception-v4, inception-resnet and the impact of residual connections on learning," *Thirty-First AAAI Conference on Artificial Intelligence* **4 (12)**: 4278-4284.

[22] V. Nair and G. E. Hinton (2010), "Rectified linear units improve restricted boltzmann machines," *Proceedings of the 27th international conference on machine learning*: 807-814.