Indoor Scene Recognition using Deep Learning

Ashwin Sahasrabudhe
Department of Robotics Engineering
Worcester Polytechnic Institute
Worcester, MA
Email: amsahasrabudhe@wpi.edu

Nalin Raut
Department of Robotics Engineering
Worcester Polytechnic Institute
Worcester, MA
Email: nraut@wpi.edu

Tushar Sawant
Department of Robotics Engineering
Worcester Polytechnic Institute
Worcester, MA
Email: tysawant@wpi.edu

***Abstract*—Indoor scene recognition is a challenging and open-ended problem. This problem plays an important role particularly in localizing indoor robots like companion robots, floor cleaning robots,etc. Thus, the project attempts to solve this problem using Convolutional Neural Networks. This project also utilizes and compares various approaches such as Transfer Learning and the bag-of-words model for the purpose of indoor scene recognition.**

Keywords*—*Scene Recognition, Computer Vision, Deep Learning, CNN, Transfer Learning.

I. Introduction

Currently, Scene recognition is a challenging task and an open ended problem amongst Computer Vision researchers. Several research groups at Massachussetts Institute of Technology have worked on scene recognition for outdoor as well as indoor scenery. However, it has been noted that recognizing an outdoor scene is easier and algorithms seem to give higher accuracy as compared to Indoor scene recognition[1]. The major reason behind this is that, most of the indoor scenes have considerably more clutter as compared to outdoor scenes. This results in the object detection or scene recognition algorithm getting confused and predicting scenes with less accuracy. To tackle this problem, we try to make use of Convolutional Neural Network to recognize indoor scenes from images.

A. Motivation

Our primary motivation for pursuing this project was to get familiar with the use of Deep Learning for scene recognition tasks. Learning how to compile the different layers of a convolutional neural network and train it from scratch was another one of our objectives for this project. Finally, we wanted to compare the accuracy of other scene recognition models, like the bag-of-words model, with the accuracy of deep learning models and draw out a consensus about their overall performance given a particular scene.

B. Problem Statement

Our problem includes scene recognition including different scene categories such as: Kitchen, Bedroom, and Corridor. Furthermore, implementing different classification techniques and comparing them is also part of our study.

The rest of this article is organized in the following way: first, we describe the methods used in this project implementation i.e. technical approach. Next, the figures of merit used to measure the performance of our implementation are defined, followed by results. Finally, conclusions and directions for future work are summarized.

II. Dataset

For this project we have used 'MIT Places 365 Dataset'[7] with 365 Scene Categories. We used only 3 categories - Bedroom, Corridor and Kitchen for purpose of this project. We also created our custom dataset by downloading images from Google Images for testing purposes.

III. Technical Approach

In this project, we have implemented scene recognition using three different methods. Firstly, we have implemented scene recognition using a Convolutional Neural Network (CNN) built from scratch, having trained the CNN to recognize a variety of indoor scenes. Secondly, we used transfer learning on a pre-trained CNN called Alexnet, available in the neural network toolbox for MATLAB, to recognize indoor scenes and classify them according to the categories. And finally, we have implemented the bag of words method for scene recognition on the dataset and have classified the different categories using the nearest neighbor distance (NND) metric and support vector machine(SVM) clasification. We used a subset of the Places365 Dataset consisting of 1000 images per category in three indoor scene categories; namely the bedroom, the kitchen and the corridors as the dataset in this project.

A. Scene Recognition using a CNN trained from scratch

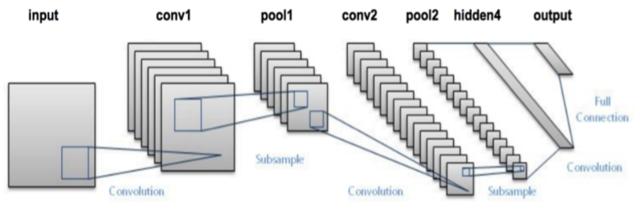

Fig. 1. Basic CNN architecture[6]. The figure shows the different layers in a basic Convolutional Neural Network.

CNNs can be used for a wide variety of classification tasks and differ slightly from neural networks. Neural networks, are made up of neurons with learnable weights and biases. Each neuron receives several inputs, takes a weighted sum over them, passes through an activation function and, responds with an output. The main difference between a CNN and a Neural Network is that the convolutional neural network (CNN) has layers of convolution and pooling. This means that the first layers that come after the input, do not use all input features that are connected. The different layers usually used in CNNs can be detailed as follows. CNNs consists of four main layers - the Convolutional Layer, the Pooling Layer, the Rectified Linear Unit (ReLU layer) and the Fully Connected Layer. A Convolutional layer applies a convolutional operation to the input, passing the result to the next layer. The convolution emulates the response of an individual neuron to visual stimuli.Another important layer in the CNN is the local or global pooling layer, which combines the outputs of the neuron clusters at one layer into a single neuron in the next layer. A ReLU layer performs a threshold operation to each element of the input, where any value less than zero is set to zero.Finally, the fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network.

Other layers usually implemented in a CNN include softmax layers, classification output layers and dropout layers. A softmax Layer applies a softmax function to the input. The classification output layer holds the name of the loss function that the software uses for training the network for multi-class classification, the size of the output, and the class labels. A dropout layer randomly sets input elements to zero with a given probability. For the CNN implemented from scratch in this project, we used 2 convolutional layers, 2 pooling layers, 1 fully connected layer, 2 cross channel normalization layers, 2 relu layers, 1 softmax layer and 1 classification layer.

B. *Scene Recognition using Transfer Learning*

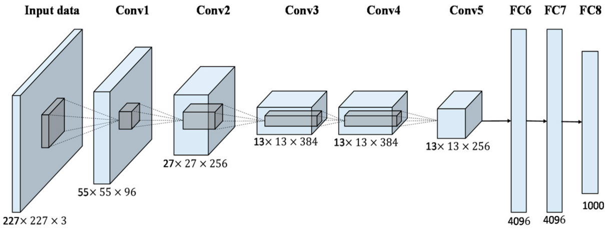

Fig. 2. AlexNet architecture[2]. The figure shows the different layers in AlexNet. The size of the input layer AlexNet is 227x227x3 and it is pre-trained for 1000 class classification. This can be used for transfer learning by extracting and replacing FC8 for the new classification task.

Next, we implemented transfer learning to use a pretrained network as a starting point for the scene recognition problem. We used the pretrained CNN called Alexnet for transfer learning wherein we deleted the last three layer of alexNEt. These Layers, the fully connected layer, the softmax layer and the classification layer, are configured for classifying 1000 classes. Hence, they must be fine tuned for the new classification task. This is done by extracting these three layers and replacing them with a new fully connected, softmax and classification layer. The options for the new fully connected layer must be set according to the new data. Thus, we set the fully connected layer to have the same size as the number of classes in the dataset used, i.e., 3 classes.Increasing the ”weight learn rate factor” and the ”bias learn rate factor” values of the fully connected layer assists the new layers to learn faster than the layers transferred from AlexNet[3]. Since the training images differed in size the image input layer of Alexnet, they were resized to match the size mentioned.

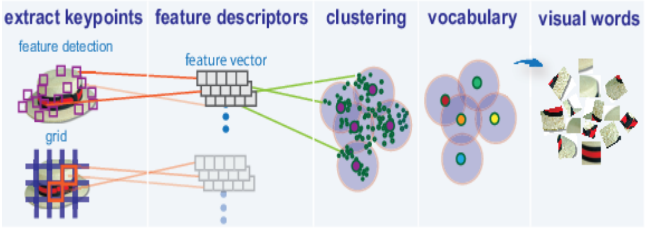

Fig. 3. Bag of words model [5]. The figure shows the bag of words model used for image classification. This method works by clustering feature vectors and labelling them as codewords which build the vocabulary. Each image is then represented by these codewords which correspond to different patches in the image.

C. *Scene Recognition using Bag of Words*

Finally we used our dataset to compute the bag of words and classified it using the nearest neighbor metric. And then using the same bag of features we used the support vector machine classifier as well. The bag of words model treats an image as a document. Hence, words need to be defined as well. This is accomplished in three steps: Feature Detection, Feature Description and codebook generation. After feature detection is completed, each image is abstracted by several local patches. These patches are represented as numerical vectors called feature descriptors. Feature descriptors such as SIFT or SURF can be used for this purpose. We have implemented SURF feature descriptors in this project. The final step is to convert the patches represented as vectors into codewords. K-means can be done to find clusters of the vectors whose centers are then defined as the codewords. Therefore, each patch in an image belongs to a certain cluster, i.e, a particular codeword. The entire image can be thus be represented as a histogram of codewords.

IV. Key Results

We compare all the four methods and find that methods including Convolutional Neural Network give a better accuracy as compared to the bag of features with both Nearest Neighbor and SVM classifiers. Table 1 shows accuracy values obtained with each method.

TABLE I. This table shows the accuracy values for each method discussed in our approach

Method Used	Percentage Accuracy
Bag of features using Nearest Neighbor Classifier	63.00
Bag of features using SVM classifier	72.62
CNN trained from scratch	77.67
Transfer learning using a pre-trained CNN	91.33

Additionally, if we have a look at the confusion matrices in Fig.4 and Fig.5 it is evident that the accuracy of scene classification for ’corridor’ is high given the fact that a corridor is more monotonous and has less clutter.

Observing the transition between confusion matrices in Fig.2, Fig.2, Fig.4, and Fig.5, we can say that the accuracy for each category increases. In other words, more number of test images in each category are correctly classified.

V. Conclusion

Through this project, we learned all about the intricacies of convolutional neural networks and their use in image recog-

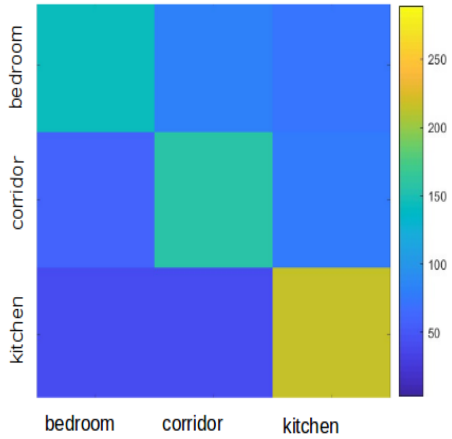

Fig. 4. Confusion Matrix (Bag of Features with Nearest Neighbor Classifier). The figure shows confusion matrix for scene classification using Bag of Features with Nearest Neighbor Classifier.+

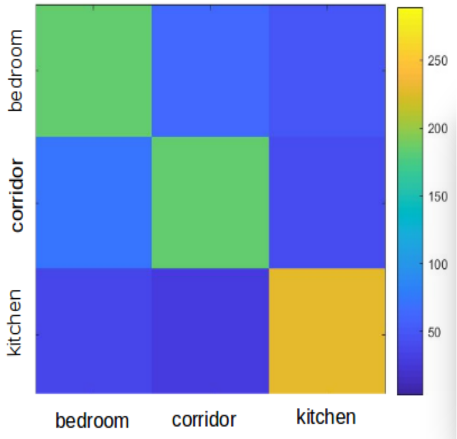

Fig. 5. Confusion Matrix (Bag of Features with SVM Classifier). The figure shows confusion matrix for scene classification using Bag of Features with SVM Classifier.

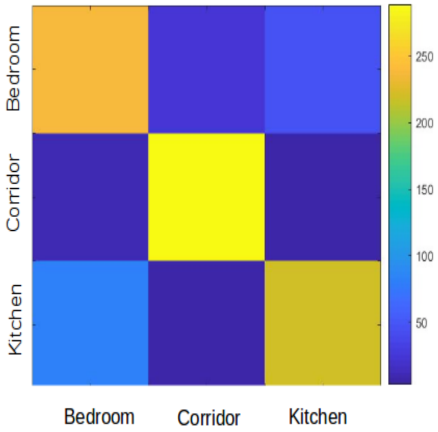

Fig. 6. Confusion Matrix (CNN built from scratch). The figure shows confusion matrix for scene classification using CNN designed from scratch.

nition tasks. Based on the comparative results, we concluded that Neural Networks perform better than other classification methods used in this project for the purpose of indoor scene recognition. We also saw that the bag-of-words model was more confident in classifying images from the kitchen, whereas the CNNs were most confident about classifying corridors. Another important observation was that to improve the accuracy of CNNs, a huge dataset of objects or scenes is required.

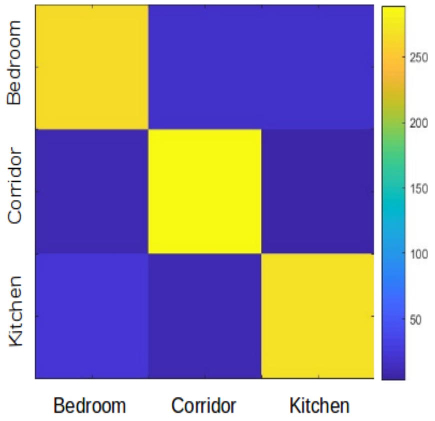

Fig. 7. Confusion Matrix (Transfer Learning with a Pre-Trained CNN). The figure shows confusion matrix for scene classification using Transfer Learning with a pre-trained network.

Acknowledgment

We would like to thank Professor John Nafziger from the Department of Robotics Engineering at Worcester Polytechnic Institute for his patience, guidance, advice and support over the course of the project.

Contribution

All the team members contributed equally towards this project. The tasks accomplished by each member are as follows:

Tushar Sawant: Preprocessing of Images, Transfer Learning and CNN from scratch.

Nalin Raut: Preprocessing of Images, CNN from scratch and other methods of classification including Bag of features with Nearest Neighbor and SVM Classifier.

Ashwin Sahasrabudhe: Preprocessing of Images, Transfer Learning and other methods of classification including Bag of features with Nearest Neighbor and SVM classifier.

We made sure knowledge gained by each one of us was exchanged amongst us.

References

[1] Lu Li, Siripat Sumanaphan, ”Indoor Scene Recognition”, Stanford University CS229 Autumn 2011 Final Project Writeup.

[2] Xiaobing Han, Yanfei Zhong , Liqin Cao, and Liangpei Zhang, ”Pre-Trained AlexNet Architecture with Pyramid Pooling and Supervision for High Spatial Resolution Remote Sensing Image Scene Classification”, Remote Sensing Big Data: Theory, Methods and Applications, Volume 9, Issue 8, 2017.

[3] https://www.mathworks.com/help/nnet/examples/transfer-learning-using-alexnet.html

[4] Bavin Ondieki, ”Convolutional Neural Networks for Scene Recognition”, Stanford University CS 231N Final Project.

[5] https://www.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html

[6] https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/

[7] MIT Places365 dataset. http://places2.csail.mit.edu/download.html

[8] Bolei Zu et. al, ”Learning Deep Features For Scene Recognition using Places Database”.

[9] Szegedy, Christian, et al. ”Intriguing properties of neural networks.” arXiv preprint, 2013.