Deep Learning Framework for Scene Based Indoor Location Recognition

[1]Akkamahadevi Hanni[*], [2]Satyadhyan Chickerur[^], [3]Indira Bidari[^]
*Department of Computer Science and Engineering, B.V.B.College of Engineering and Technology,
^Centre for High Performance Computing, KLE Technological University
Hubballi-580 031, Karnataka, INDIA
[1]arhanni18@gmail.com, [2]chickerursr@kletech.ac.in, [3]indira_bidari@bvb.edu

***Abstract*—Scene recognition and Object detection have made momentous progress lately. While mobile robotics and drone analysis has already reached its growth apex, for robots, which can interact with humans and the indoor environment, having a sense of discerning indoor scenes or interiors of a building is an added benefit. Although, many approaches have been proposed to detect objects and locations such as Indoor Positioning System (IPS), feature based, content based recognition systems etc., indoor scene recognition is yet to gain ground. This is quite justified since, unlike outdoor scenes, indoor scenes lack distinctive local or global visual substance patterns. This paper proposes a new technique of achieving this goal by considering the data from the scene's RGB and Depth images. With advancement in machine learning methodologies such as neural networks, deep neural networks and Convolutional Neural Networks, the workable accuracy in scene recognition is no longer hypothetical. A Deep CNN framework is used with a transfer learning approach for indoor scene recognition implemented on Tensor Flow (python), using the RGB as well as point cloud data i.e., RGB-D images. With this proposed system of deep CNN model, accuracy is able to reach up to 94.4% on the indoor dataset. Further a comparison of the proposed model performance with that of the digits' GoogLeNet and AlexNet framework is conducted. Also a training of the algorithm on the benchmark NYUv2 dataset and have achieved an accuracy of 75.9% which beats the highest accuracy obtained on that model (64.5%).**

***Index Terms* — Tensor Flow; deep CNN; tensors; logits; pcd data; RGB-D images;**

I. Introduction

In domestic and indoor spaces, robots must interact with the surrounding to carry out various tasks which may seem trivial to humans. But training a system to have the functional understanding and accurate perception of the scene is tricky. Also, it is important in calibrating the sense of location in a domestic environment, for example: deciding whether the robot is facing the window or balcony, whether it has entered a study room or a living room, to know the exit point through a door when present in a room, etc.

Deep learning has formed into an intelligent machine that is a flexible rational agent that perceives its environment and takes actions that maximize its chance of success at some task. Applications of automatic location identification has gone far enough for identification of location by looking over thousands of images, relaying location of caller during a call reception, displaying user or device position on maps, etc. Unlike the indoor scene recognition [3] which considers global spatial properties of objects contained in an image, this paper adopts a salient approach of recognition. This paper intends to build a model for classification of indoor locations on the bases of RGB-D images. These RGB-D images are combination of normal RGB images and depth images or point cloud data. Such dense representations are sets of 3D points encoding the scene geometry, and are often bases for high-level applications such as mobile robot navigation or manipulation. Further this data is used to train, validate and test the model for indoor scene based location identification. Unlike designing a classification model from scratch, this paper fine-tunes an existing inception model which is pre-trained over a large dataset. This fine-tuning process is known as transfer learning, in which only the last layer is built from scratch as per the required classification and is retrained from the existing weights for new classes. It is primarily known that 3D based classification in dynamic environment is challenging [1], the proposed system attempts to make use of the point cloud data in its minimalistic form and combines it with the visual images of the scene. When compared to classification and 3D object detection mechanisms built solely on 2D image datasets, that have reached high accuracy levels this system does not calibrate to those levels. However, the proposed model confirms to an accuracy level of 94.4% on the dataset described in the following sections. This classification is intended to be used for many real-time applications such as drone mapping i.e., when a drone is moving in an autopilot mode, it maps its surroundings and decides its trajectory. Labelling the locations that are possibly impending in its path will help improve the auto-navigation system.

The rest of the paper is organized as follows: Section II discusses relevant previous work on indoor computer vision, Section III presents the experimental tools and dataset used for the study, Section IV provides details about the implementation of the deep CNN model used in this work, Section V presents an evaluation of the proposed method and a comparison with state-of-the-art approaches. Finally, Section VI presents the main conclusions of this work and future avenues of research.

978-1-5386-4021-0/17/$31.00 ©2017IEEE

II. RELATED WORK

In [3,4], the paper aims at designing a scene recognition model specifically tailored to the task of indoor scene recognition after observing that the traditional recognition algorithms dramatically fail or perform poorly when applied to indoor images. This is exploited by including the information of both global spatial properties of images and the objects they contain. Basically, it uses image prototypes to define a mapping between images and scene labels that can capture the fact that images containing similar objects must have similar scene labels and that some objects are more important than others in defining a scene's identity. The work is based on learning distance functions, i.e., the method learns a weighted combination of elementary distance functions for a set of prototypes by directly minimizing a classification objective. The implementation goes about by defining ROI (regions of interest) and scene prototypes, where each set of prototypes best describes its class. The results are verified by analyzing the Gist SVM, ROI Segmentation, ROI Annotation, ROI+Gist Segmentation and ROI+Gist Annotation. Additionally, this paper also contributes towards providing a large dataset [3] of 67 classes describing the indoor scenes containing 15620 images.

The Scene semantic segmentation from indoor RGB-D images using encode-decoder fully convolutional networks [2] practices Multiple Kernel Maximum Mean Discrepancy (MK-MMD) as a distance measure to find common and special features of RGB and D images in the network to enhance performance of classification automatically.The Indoor Scene Recognition in [6] claims to have a better average accuracy rate (68.295%) than any other state-of-the-art result on NYUv2 dataset [3] using only the deep features trained from ImageNet. It implements a selective search for finding ROI, and then a pre-trained CNN is applied to each ROI to get a deep feature vector. Further, three-level spatial pyramid representations of the image with deep features are used to create the final feature representation. Finally, multiple one-vs-all linear SVMs are used to do the scene classification. This paper uses a generic pre-trained model which has been applied/used for object detection, feature extraction etc., which in observation performs less efficiently when compared to operation specific models.

The SIFT (Scale Invariant Feature Transform) approach used by [5] for identifying a specific location's images when given a large set of images and extended to also include binary classification of other classes as well as multi-class classification. It derives its work from linear spatial pyramid matching using sparse coding for image classification [9]. It then uses K-means and Sparse Coding for Vector Quantization. In addition to classification based only on the histogram (or max-pooling) of quantized SIFT features [13] extracted from the entire image, it also extends its feature vectors to capture more spatial information of an image by concatenating with location-specific histograms.

Segmentation process has many limitations, the major hurdle present in the segmentation of the cluttered images and this difficulty increases if the images are blurred [11]. The semantic segmentation of depth images [9] is more closely related to the work undertaken in this paper, by that we mean it uses a multiscale convolutional network to learn features directly from the images and the depth information i.e., it makes use of RGB and D (depth) images [10, 11] for segmentation. The input images are parsed frame by frame using a multiscale network and super pixels, the RGBD images are transformed to a Laplacian pyramid, the feature maps of coarser-scale and finest-scale are concatenated, along with this a single segmentation of the image into super pixel is computed and in the end labeling is obtained by the aggregation of the classifier predictions into the super pixels. The multi-scale CNN with dense feature extractor adopted by this paper, produces a series of feature vectors for regions of multiple sizes centered on every pixel in the image. It contains multiple copies of a single network that are applied to different scales of a Laplacian pyramid version of the RGBD input image. This method reports a separate class-wise and pixel-wise accuracy of 64.5% on the NYU-v2 [22] dataset.

Likewise, in [10], indoor segmentation and inference support is implemented using 3D cues (RGBD data) along with contributing a dataset of 1449 RGBD images, capturing 464 diverse indoor scenes [22]. With the input of raw and in-painted depth images the overall 3D structure of the scene is parsed jointly, to detect major surfaces, compute surface normal and align point clouds to three dominant orthogonal directions. These are further segmented into regions using hierarchical [12] segmentation which iteratively merges regions based on the likelihood of two regions belonging to the same object instance. Finally, the features are extracted for support classification which outputs structure labels and support relations.

In line with [9, 10], [12] implements indoor scene recognition through object detection. Overcoming the drawbacks of traditional segmentation algorithms and manual strategies to identify relevant intermediate properties, it proposes recognition based on a probabilistic hierarchical representation that uses common objects as an intermediate semantic representation.

III. EXPERIMENTAL SETUP

The tools involved in the implementation of this project include Microsoft Kinect camera for capturing the surroundings (raw and depth images), ROS for Kinect support, rviz interface for capturing RGB-D images, Tensor Flow with GPU support for training and validating the model, Digits framework for comparing the results with Googlenet and Alexnet models.

The proposed system makes use of the ROS (Robot Operating System) Indigo version, which is an open source tool primarily targeted at the Ubuntu 14.04 LTS (Trusty) release [18]. This contains the necessary launch files for freenect_camera [19], Openni_launch [20], Rviz for 3D visualization [21]. Implement the model on Tensor Flow which is an open source software library for machine learning in various kinds of perceptual and language understanding tasks.

The implementation makes use of the python API with GPU support. It is complex to build a befitting dataset for indoor environments as the interiors of a room, even parts of it, change often. The number of objects, lighting conditions and the number of obstacles may vary substantially. Also, unlike conventionally each image cannot accommodate the entire space of a room, multiple images together will be needed to comprise the complete information of a room/indoor location. Therefore, multiple such set of images are gathered as training data. The dataset is consolidated and comprises of RGB and Depth images under various conditions mentioned above. In this analysis, three classes for classification are considered, where each class has a training set of 150-200 images captured from various angles and with different objects inside the central image frame, by resizing the image dataset to 256 X 256. However, even test the model on the standard NYUv2 dataset which was constructed for Indoor Segmentation and Support Inference from RGBD Images and the observations are described in the results section.

A sample of RGB images and corresponding depth images from the dataset is shown in fig 1 and the sample images from the NYUv2 is shown in fig 2. For simplicity of use only six classes from the labelled NYUv2 dataset are considered.

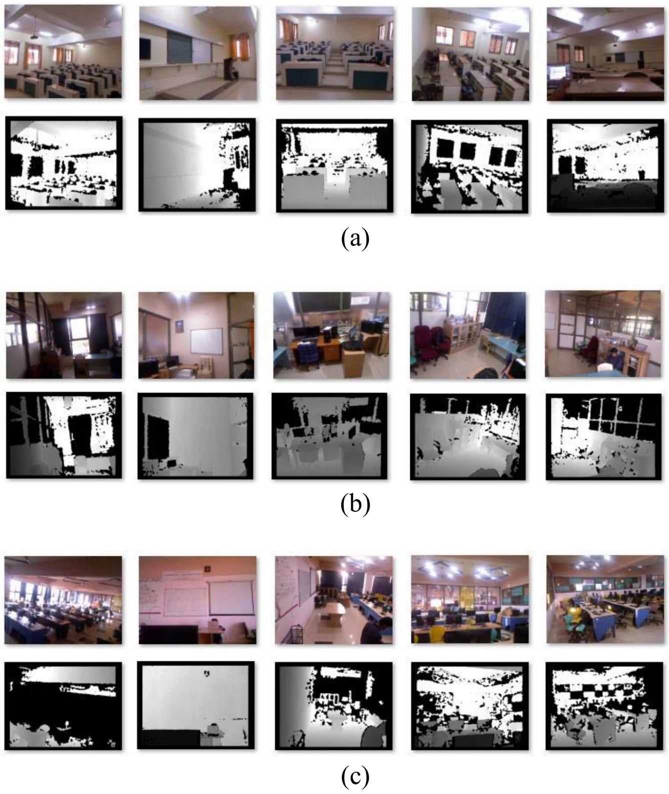
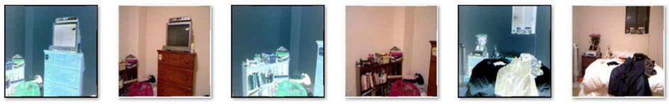


Fig 1. Indoor dataset of KLETECH (three classes). (a) Classroom, (b) Cabin and (c) Lab.



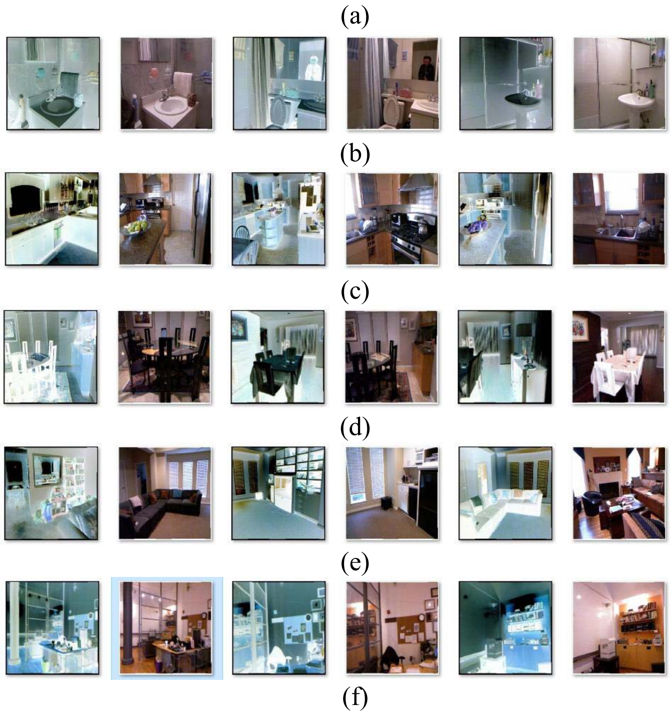


Fig 2. NYUv2 dataset (5 classes). (a) bedroom, (b) bathroom, (c) kitchen, (d) dining (e) living room and (f) office.

IV. MODEL IMPLEMENTATION

The perception makes vision incredibly simple. It does not take any effort for humans to tell apart a puma and a panther, an office and a living room, or recognize different human faces. But, for a computer it is extremely difficult to have the same level of perception and understanding: it only seems easy for humans because the brains are incredibly good at understanding images as they have been trained and taught about different things. Likewise, if a machine is trained with enough ground or capacitive data, which can give it a perception close to humans, it is a milestone achievement. There has been tremendous progress in this regard, specifically, deep convolutional neural network (CNN) can achieve reasonable performance on hard visual recognition tasks, matching or exceeding human performance in some domains. Many related studies have demonstrated steady improvements in the field of image processing by validating their results against ImageNet- an academic benchmark for computer vision. Successive models continue to show improvements and cutting-edge methodologies, each time achieving a new state-of-the-art result, viz: QuocNet, AlexNet, Inception (GoogLeNet), BN-Inception-v2. Tensor Flow has taken the next step by releasing code for running image recognition on the latest model, Inception-v3.

In this paper, a concept called transfer learning [8] is implemented by taking a fully-trained model for a set of categories like ImageNet, and retrain from the existing weights for new classes i.e. retrain the final layer from scratch, while leaving all the others untouched. The CNN applies a series of filters to the raw pixel data of the images to extract and learn the higher-level features. The proposed model consists of 3 inception layers, 10 mixed layers, 3 pool layers (max_pool) and the final layers as shown in figure 3. All the other layers mentioned lay beneath the mixed_10 layer and are not shown in the image to draw focus only on the final few layers. The convolution layers apply a specific set of filters to the image and apply a ReLU [14] activation function to introduce non-linearity into the model. The pool layer divides the pixels into subsections and retains their maximum value while discarding the others. The mixed layers perform classification on the features extracted and every node in this layer is connected to every other node in the preceding layer. The input layer is nothing but the reshape function which passes the image dimensions and channels. After this, the final_training_ops function is added which adds a new softmax and fully connected layer for training.

The algorithm is discussed in detail below: it explains how the inception v3 model which is pre-trained on the ImageNet dataset by only training the new top layer to recognize the classes defined in this study. This top layer receives an input of 2_{11}-dimensional vector for each image and train a softmax layer on top of this representation. This means that, if the softmax layer contains N labels, this corresponds to learning N + 2048*N model parameters corresponding to the learned biases and weights.

The first step is to split the image dataset into training, validation and testing sets based on a hash function to group related images together. A list of all the images is created based on the associated labels and checks for any kind of distortions. If the images are distorted, operations like cropping, scaling, increasing central image brightness, improving sharpness of the image etc., are applied on the image. The most important step here is calculating the bottleneck layer values which is nothing but the layer executed before the final output layer. These bottlenecks are cached so that repetitive use of the images for training and validation does not make the process slow. This step makes use of the list of images with the labels, input tensor for the jpg data and bottleneck tensor which is the penultimate output layer of the graph. After this, a new softmax and fully connected layer is added to the underlying layers which sets up all the gradients for the backward pass. This is the step where identification of the new classes is performed and train it based on these specific desired labels. This includes computing pre-activations, logits (which returns raw values for predictions), activations and final tensor of the summary histogram. The training is carried out for as many steps as specified in the input (in this case the set it to 2000 steps) and the summaries are written to the log file which can later be used in tensorboard for visualization of graphs. As the process continues reported accuracy improves, by analyzing the cross entropy and accuracy of train and validation. Now that the training is processed, later evaluation of the accuracy on the results with the help of ground truth data is performed. A final test evaluation is done on the set images that weren't used before, which gives the accuracy of the model. This test evaluation is the best estimate of how the trained model will perform on the given classification task and a measure of the model's accuracy. Observations show an accuracy value of around 90% to 100%. This number is based on the percent of the images in the test set that are given the correct label after the model is fully trained.

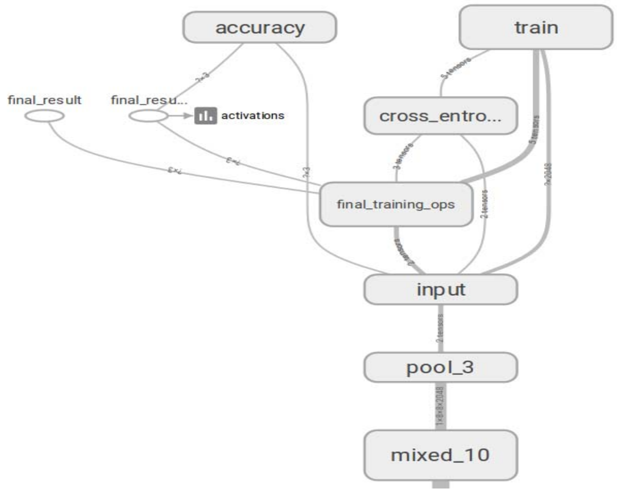
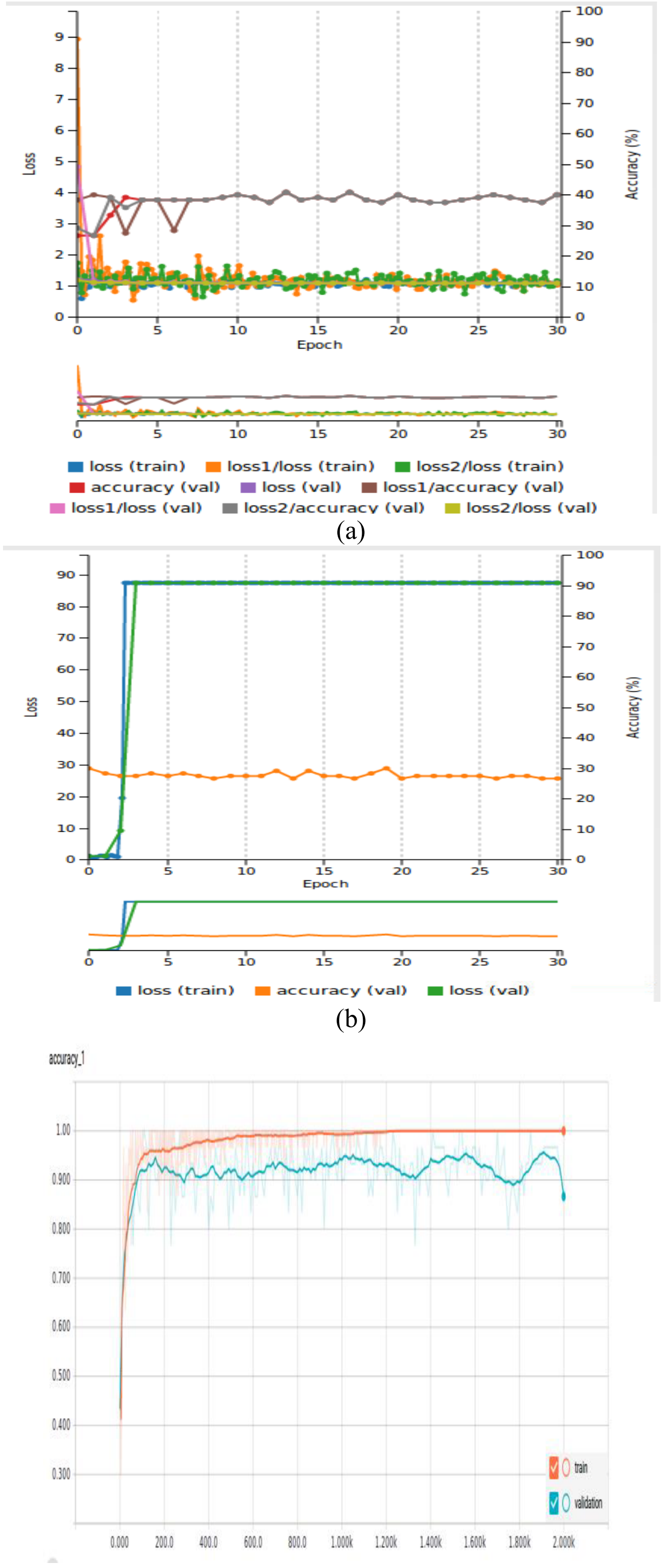


Fig 3. Proposed model and composite layers.

The above image describes the flow as explained previously. After the output graph is generated, any image can be tested on the trained model by using the classification file. This file inputs the test image and outputs the classes it classifies into along with the percentage of accuracy against each class. A common way of improving the results of image training is by preprocessing the image like deforming, cropping, increasing sharpness or brightening the training input images. This has the advantage of expanding the effective size of the training data due to all the possible variations of the same images, and tends to help the network learn to cope with all the distortions that will occur in real-life uses of the classifier.

V. RESULTS

After every training level, a part validation is done by the model and in the end a large test set is validated which is considered as the accuracy score of the model, which is 94.4%. Comparison is performed on the accuracy results attained from the proposed model against that of AlexNet and GoogleNet. Observation shows that for the RGB-D dataset the results obtained in AlexNet and GoogleNet are less efficient. In fig 4.a. observation shows that the accuracy attained in GoogleNet is 40% at epoch 30 and fig 4.c shows that the validation accuracy attained in AlexNet as 26.666% at epoch 30, which are much less than the expected workable accuracy. Likewise, fig 5.a, 5.b and 5.c shows the accuracy graph obtained in GoogleNet, AlexNet and proposed system for the NYUv2 dataset which is 30.236%, 30.34% and 75.9% respectively. Note that the following graphs in fig 4.a, 4.b, 5.a and 5.c are scaled differently from 4.c and 5.c, as they are generated in different frameworks. However, each model was trained till a maximum desired accuracy was attained with respect to each model.

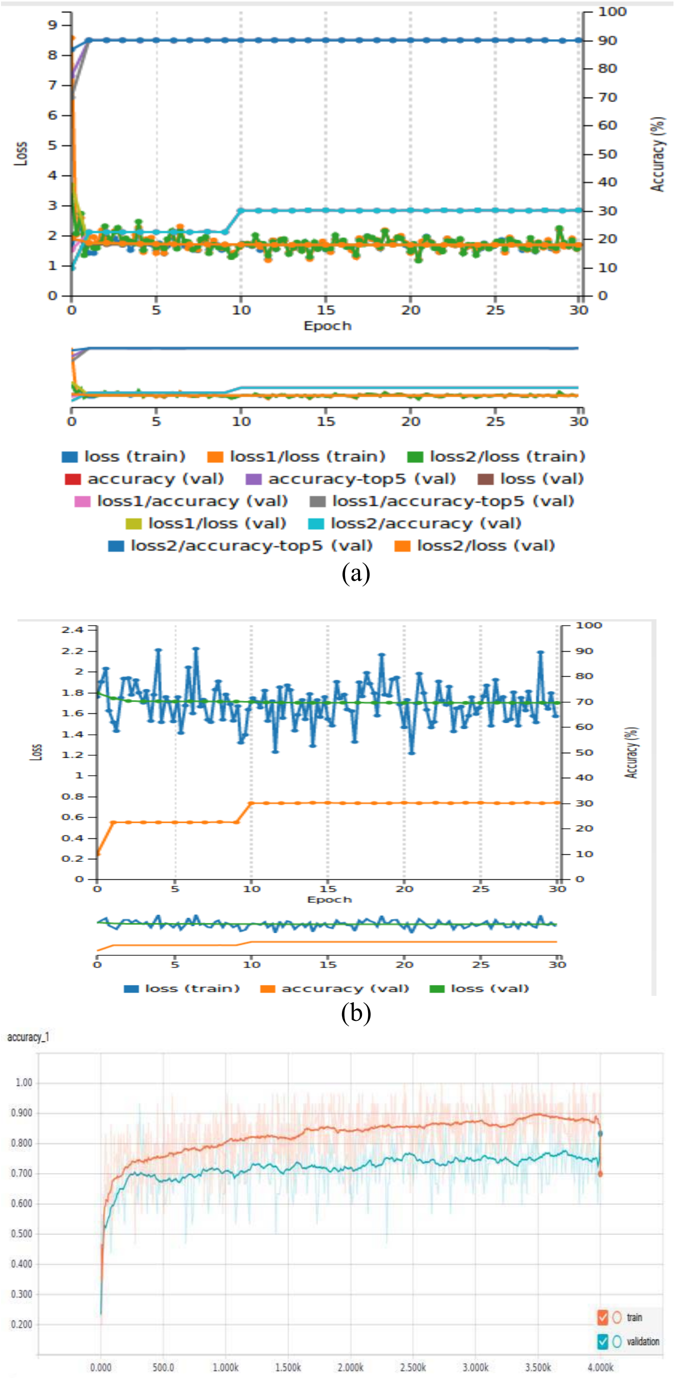


(a)

(b)

(c)

Fig 4. (a) GoogleNet accuracy graph for RGBD dataset, (b) Alexnet accuracy graph for RGBD dataset and (c) Proposed model accuracy graph for RGBD dataset.



(a)

(b)

(c)

Fig 5. (a) GoogleNet accuracy graph for NYUv2 dataset, (b) Alexnet accuracy graph for NYUv2 dataset and (c) Proposed model accuracy graph for NYUv2 dataset.

A. Test images results:

Referring to images from dataset in figure 6, the following table gives the test results obtained for a set of 15 images (including both depth and RGB images) tested against the proposed model. The values noted are the percentages of probability of the image belonging to a specific class.

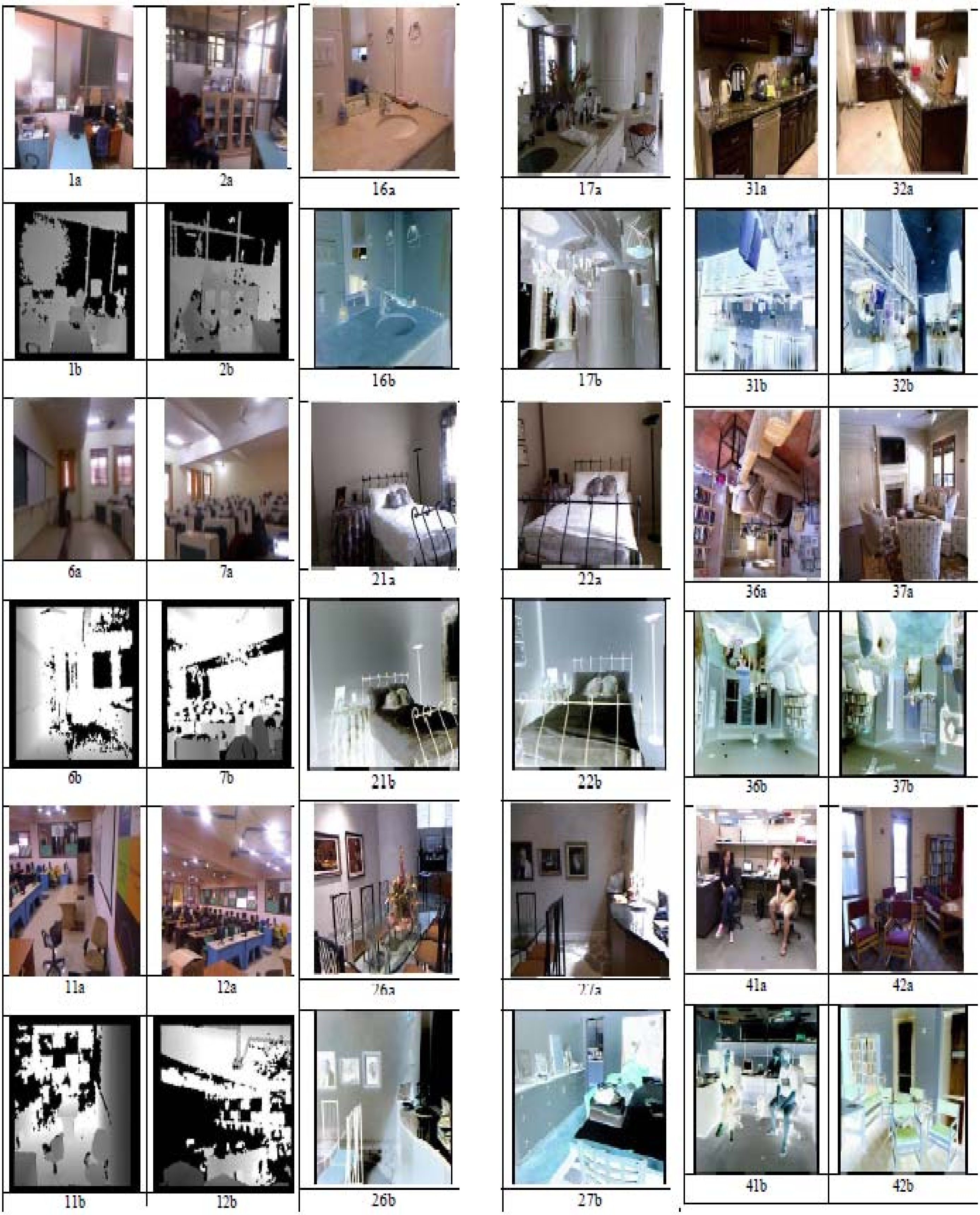


Fig 6. 1-15 RDBD test data and 16-40 NYUv2 test data

TABLE I. TEST RESULTS FOR PROPOSED MODEL

Image	Cabin	Classroom	Lab	Actual class
1a	98.43	0.36	1.20	Cabin
1b	37.84	0.71	61.44	Cabin
2a	96.97	0.04	2.98	Cabin
2b	99.79	0.05	0.14	Cabin
6a	0.11	99.84	0.04	Classroom
6b	0.26	98.73	0.26	Classroom
7a	0.24	97.48	2.27	Classroom
7b	0.06	98.08	1.84	Classroom
11a	1.02	5.00	93.97	Lab
11b	0.93	0.25	98.81	Lab
12a	0.00	0.05	99.94	Lab
12b	0.26	1.03	98.70	Lab

In table 2: the class labels are mapped as follows. “Bedroom” as “A”, “Bathroom” as “B”, “Kitchen” as “C”, “Dining” as “D”, “Living Room” as “E” and “Office” as “F”.

TABLE II. MAPPING OF CLASS LABELS

Image	Bathroom (A)	Bedroom (B)	Kitchen (C)	Dining (D)	Living Room (E)	Office (F)	Actual Class
16a	99.53	0.01	0.11	0.00	0.01	1.47	A
16b	92.73	0.13	4.28	0.31	0.50	2.02	A
17a	71.02	0.11	15.93	9.16	2.3	1.47	A
17b	68.15	0.12	5.70	2.52	20.76	2.71	A
21a	0.31	54.78	0.08	0.53	44.23	0.04	B
21b	0.07	87.73	0.02	0.75	11.39	0.01	B
22a	0.03	96.36	0.08	0.14	3.29	0.08	B
22b	0.03	97.69	0.05	0.44	1.73	0.04	B
26a	0.00	0.00	99.98	0.00	0.00	0.02	C
26b	1.29	1.35	46.08	20.45	4.77	26.07	C
27a	0.00	0.00	100	0.00	0.00	0.00	C
27b	1.82	1.48	92.57	0.46	3.12	0.56	C
31a	0.01	0.10	0.07	99.08	0.71	0.04	D
31b	1.99	1.27	0.95	72.43	21.94	1.42	D
32a	1.63	0.12	0.58	95.94	1.55	0.18	D
32b	0.98	12.28	6.17	28.43	22.96	29.18	D
36a	0.00	1.89	0.07	0.14	97.34	0.55	E
36b	0.06	9.06	1.67	31.06	48.55	9.61	E
37a	3.90	0.45	11.72	30.56	52.79	0.58	E
37b	0.03	3.72	0.43	3.45	92.02	0.34	E
41a	0.10	3.88	0.58	0.77	24.20	70.48	F
41b	0.05	10.10	9.04	0.61	16.82	63.37	F
42a	3.09	14.54	0.54	0.82	3.42	77.59	F
42b	9.14	3.46	12.93	0.95	2.07	71.44	F

VI. CONCLUSION

This paper proposed a classification methodology for highly variable indoor scenes, based on transfer learning approach implemented using tensor flow. The deep CNN approach has been used as it has inbuilt feature extraction and has contributed to a large decrease in error rate. To better represent a scene, model is trained with RGB-D images, which not only describes the relative view of a scene but also captures their spatial interrelationships. Experimental results demonstrate that the proposed indoor scene recognition method achieves significantly higher accuracy than the state-of-the-art architectures AlexNet and GoogleNet (without optimization). This paper also contributes a minimalistic dataset of RGBD of the KLETECH campus images comprised of three classes. The future work would comprise of study of classification of indoor scenes with 3D datasets.

VII. ACKNOWLEDGEMENT

We would like to thank NVIDIA for providing GPU’s to support this research work. We also thank our university and higher management for encouraging this research.

REFERENCES

[1] Alex Teichman, Jesse Levinson, Sebastian Thrun, Stanford Artificial Intelligence Laboratory “Towards 3D Object Recognition via Classification of Arbitrary Object Tracks”, [Online] Available: http://cs.stanford.edu/people/teichman/papers/icra2011.pdf [Accessed October 12, 2017].

[2] Wang, Z., Li, T., Pan, L., and Kang, Z, “Scene semantic segmentation from indoor RGB-D images using encode-decoder fully convolutional networks” The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-2/W7, 2017 ISPRS Geospatial Week 2017, 18–22 September 2017, Wuhan, China.

[3] Nathan Silberman, Pushmeet Kohli, Derek Hoiem, Rob Fergus. Indoor scene recognition database. [Online] Available: http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html. [Accessed September 7, 2017].

[4] Ariadna Quattoni, Antonio Torralba, CSAIL, MIT, UC Berkeley EECS & ICSI “Recognizing Indoor Scenes” [Online] Available: http://www.csd.uwo.ca/~olga/Courses/Fall2014/CS9840/PossibleStudentPapers/indoor.pdf. [Accessed October 1, 2017].

[5] Lu Li and Siripat Sumanaphan, “Indoor Scene Recognition”, Stanford University. [Online] Available: http://cs229.stanford.edu/proj2011/LiSumanaphan-IndoorSceneRecognition.pdf. [Accessed October 15, 2017].

[6] Yangzihao Wang and Yuduo Wu “Scene Classification with Deep Convolutional Neural Networks”, University of California, Davis. [Online] Available: http://www.idav.ucdavis.edu/~yzhwang/ecs289h-vision.pdf. [Accessed October 6, 2017].

[7] Jianchao Yang, Kai Yu, Yihong Gong, and T. Huang. “Linear spatial pyramid matching using sparse coding for image classification” Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 1794 –1801, June 2009.

[8] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, Trevor Darrell, “DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition”, UC Berkeley & ICSI, Berkeley, CA, USA. [Online] Available: https://arxiv.org/pdf/1310.1531v1.pdf [Accessed November 12, 2017].

[9] Camille Couprie, Cl´ement Farabet, Laurent Najman and Yann LeCun, “Indoor Semantic Segmentation using depth information”, IFP Energies Nouvelles Technology, Computer Science and Applied Mathematics Division, Rueil Malmaison, France. [Online] Available: https://arxiv.org/abs/1301.3572 [Accessed October 10, 2017].

[10] Nathan Silberman, Derek Hoiem, Pushmeet Kohli and Rob Fergus, Courant Institute New York University, Department of Computer Science University of Illinois at Urbana-Champaign, Microsoft Research Cambridge. “Indoor Segmentation and Support Inference from RGBD Images”, [Online] Available: http://cs.nyu.edu/~silberman/papers/indoor_seg_support.pdf [Accessed October 20, 2017].

[11] Chickerur, Satyadhyan, and Aswatha Kumar. "A Robust Cluster Based Approach for Image Restoration." In Image and Signal Processing, 2008. CISP'08. Congress on, vol. 1, pp. 370-374. IEEE, 2008.

[12] P. Espinace, T. Kollar, A. Soto, and N. Roy, “Indoor Scene Recognition through Object Detection”. 2010 IEEE International Conference on Robotics and Automation Anchorage Convention District May 3-8, 2010, Anchorage, Alaska, USA.

[13] Tarek Elguebaly and Nizar Bouguila, “Indoor Scene Recognition with a Visual Attention-Driven Spatial Pooling Strategy”, Faculty of Engineering and Computer Science Concordia University Montreal, Canada. 2014 Canadian Conference on Computer and Robot Vision.

[14] Rectifier (neural networks) [Online] Available: https://en.wikipedia.org/wiki/Rectifier_(neural_networks) [Accessed October 15, 2017].

[15] Cross entropy [Online] Available: https://en.wikipedia.org/wiki/Cross_entropy [Accessed July 1, 2017].

[16] Ariadna Quattoni and Antoni Torralba. Indoor scene recognition database. [Online] Available: http://web.mit.edu/ torralba/www/indoor.html. [Accessed September 5, 2017].

[17] Ariadna Quattoni and Antoni Torralba, “Recognizing Indoor Scenes”, CSAIL, MIT UC Berkeley EECS & ICSI. [Online] Available: http://www.csd.uwo.ca/~olga/Courses/Fall2014/CS9840/PossibleStudentPapers/indoor.pdf. [Accessed November 10, 2017].

[18] ROS Indigo Igloo [Online] Available: http://wiki.ros.org/indigo [Accessed October 26, 2017].

[19] Freenect camera [Online] Available: http://wiki.ros.org/freenect_camera.[Accessed October 6, 2017].

[20] Openni launch [Online] Available: http://wiki.ros.org/openni_launch. [Accessed July 1, 2017].

[21] RVIZ [Online] Available: http://wiki.ros.org/rviz [Accessed September 19, 2017].

[22] Nathan Silberman, Pushmeet Kohli, Derek Hoiem, Rob Fergus, http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html