applied sciences

Article

An Indoor Room Classification System for Social Robots via Integration of CNN and ECOC

Kamal M. Othman and Ahmad B. Rad *

Autonomous and Intelligent Systems Laboratory, School of Mechatronic Systems Engineering Simon Fraser University, Surrey, BC V3T 0A3, Canada; kamal_othman_2@sfu.ca

* Correspondence: arad@sfu.ca; Tel.: +1-778-782-8512

Received: 14 December 2018; Accepted: 26 January 2019; Published: 30 January 2019

Abstract: The ability to classify rooms in a home is one of many attributes that are desired for social robots. In this paper, we address the problem of indoor room classification via several convolutional neural network (CNN) architectures, i.e., VGG16, VGG19, & Inception V3. The main objective is to recognize five indoor classes (bathroom, bedroom, dining room, kitchen, and living room) from a Places dataset. We considered 11600 images per class and subsequently fine-tuned the networks. The simulation studies suggest that cleaning the disparate data produced much better results in all the examined CNN architectures. We report that VGG16 & VGG19 fine-tuned models with training on all layers produced the best validation accuracy, with 93.29% and 93.61% on clean data, respectively. We also propose and examine a combination model of CNN and a multi-binary classifier referred to as error correcting output code (ECOC) with the clean data. The highest validation accuracy of 15 binary classifiers reached up to 98.5%, where the average of all classifiers was 95.37%. CNN and CNN-ECOC, and an alternative form called CNN-ECOC Regression, were evaluated in real-time implementation on a NAO humanoid robot. The results show the superiority of the combination model of CNN and ECOC over the conventional CNN. The implications and the challenges of real-time experiments are also discussed in the paper.

Keywords: social robots; NAO robot; room classification; convolutional neural network; multi-binary classifiers; ECOC

1. Introduction

The prospect of a social robot in every home may be realized within the next two decades. There are already many researchers in academia and tech industries that are actively studying and designing prototypes of such robots. The open research objectives are diverse and include, but are not limited to, emotion recognition, perception, pattern recognition (face, object, scene, and voice), and navigation. These robots are expected to be employed as companions to seniors and children, housekeeping, surveillance, etc. [1,2]. In order to accomplish such tasks, it is essential that the robot seamlessly recognizes its own location inside the home—similar to humans who are effortlessly aware of their whereabouts at any instant, e.g., kitchen or living room. This knowledge is a pretext for many indoor navigation scenarios, and facilitates the robot's movement in the house. This paper is not about designing social robots per se; it addresses one of the many problems that collectively contribute towards efficient operation of such robots; namely knowing its location in the house at any given instant. Classification is a core computer vision problem whereby data streams are categorized into specific classes in accordance to learning their specific features. The problem has been addressed by different supervised machine learning algorithms [3]. The convolutional neural network (CNN) [4,5] is generally regarded as the state-of-the-art algorithm in deep learning for visual purposes, e.g., face recognition and object detection, especially after the pioneering work reported in Reference [6]. This algorithm surpasses conventional machine learning algorithms by integrating the feature extraction and classification problems without the requirement of careful human design [7].

The main objective of this paper is to identify different household rooms for social robotic applications in houses. The study is part of a larger project of designing social robots to be employed in such environments. The problem of indoor navigation can be addressed by different approaches; here, we propose a CNN solution for real time implementation for such social robots. We examined several CNN architectures within a home setting. The latest scene dataset, called Places [8], was adopted in the study. We downloaded the most common five indoor classes for houses (bedrooms, dining rooms, kitchens, living room, and bathrooms) from the dataset. However, we noted that each class included a sizeable number of irrelevant scenes; we therefore reduced the number of samples by removing unrelated images from each class. We then propose a combination solution of CNN with multi-binary classifiers, referred to as ECOC [9]. These models were evaluated experimentally on a NAO humanoid robot [10].

The rest of the paper is organized as follows: Section 2 discusses the related reported literature that addresses the room classification problem for robotics employing different methods. In Section 3, we briefly review CNN and ECOC algorithms. We begin with fundamental components of CNN that are used in most CNN architectures. This section also explains the main idea of ECOC approach. Section 4 focuses on all simulation experiments and examines the results. We start with a brief overview of the scene dataset. We then present simulation studies of multi-class experiments for several CNN architectures as well as results of multi-binary classifiers on the best CNN architecture. Section 5 shows the results of real-time experiments on all three models tested on a NAO humanoid robot. The paper is concluded with a discussion of the results in Section 6.

2. Related Work

Recognizing different rooms in a home environment based on their specific function is an important problem for social robots, and its solution not only facilitates seamless movement from one place to another, it is the basis of all other tasks, including assistance to humans inside the house or performing various functions in the context of the robot's overall tasks. An interaction with a human might be in the form of "*Please go to the kitchen and bring a cup of water*". This problem has attracted the attention of robotics researchers in the last decade, and several conventional machine learning methods have been employed to address room classification in indoor settings. One of the early studies reported by Burgard's group in [11] was to address semantic place classification of indoor environments by extracting features from a laser range data using AdaBoost algorithm. The experiments were conducted in a real office environment using sequential binary classifiers for differentiating between room, corridor, doorway, and hallway. It was suggested that the sequential binary AdaBoost classifiers were much more accurate than multi-class AdaBoost. The study was further extended in References [12,13] by extracting features from laser and camera for classifying six different places: doorways, a laboratory, a kitchen, a seminar room, and a corridor, as well as examining the effect of the Hidden Markov Model on the final classification. The same algorithm, i.e., AdaBoost, was trained in Reference [14] using SIFT features of online images for seven different rooms. It examined the performance of different number of classes and different possible pairs of classes, where the success of the average of binary classifiers was 77%.

Robotics researchers also employed the well-known support vector machine (SVM) algorithm for the room classification problem using different sensors. In Reference [15], laser data was used to build a hierarchical model, in which the hierarchy is employed for training and testing SVMs to classify 25 living rooms, 6 corridors, 35 bathrooms, and 28 bedrooms. Although this study reported an accuracy of 84.38%, laser data generally do not provide rich information, and require substantial processing to extract useful features. In contrast, vision features are used in other studies in order to train SVMs. In Reference [16], a voting technique was used to combine 3D features to GIST 2D features, and these were used for training SVMs to classify six indoor places: bathrooms, bedrooms, eating places, kitchens, living rooms, and offices. Furthermore, SVM and Random Forests (RF) classifiers were used and compared in Reference [17] to classify five places: corridors, laboratories, offices, kitchens, and study rooms using RGB-D images from a Kinect sensor. Room detection has also been addressed as an unsupervised learning problem using unlabeled images. In Reference [18], SIFT features and 3D representation were used to extract convex spaces for clustering images based on similarities. In addition, stereo imagery was used in Reference [19] for room detection and modeling by fusing 2D features with geometry data acquired from pixel-wise stereo for representing 3D scenes. The study was completed by modeling walls, rooms, and doorways using many techniques of extracting features, depth diffusion, depth segmentation, and clustering in order to detect room functionalities. The problem has also been addressed from different perspectives, such as the study in Reference [20], in which the authors addressed the context-awareness problem for service robots by developing a system that identified 3D objects using online information. As we can note from previous research, the main drawback is the huge effort required to extract features. This weakness can be overcome by adopting a convolutional neural network (CNN) algorithm.

CNN is a category of deep neural network that has demonstrated successful results in the field of computer vision, such as face recognition and object detection. It was proposed by LeCun [5], who introduced the first CNN architecture called LeNet in 1998, after several successful attempts since the 1980s [4,21]. There are two main advantages of this algorithm over other machine learning algorithms and the conventional fully connected feedforward neural networks. First, CNN extracts and learns features from raw images without requiring a careful engineering design for extracting features in advance [7]. Second, CNN considers the spatial structure of the image by translating inputs to outputs through shared filters [22]. After the huge improvements in data collection and computer hardware between 1990s and 2012, AlexNet was introduced [6] for addressing the object detection problem using the ubiquitous *ImageNet* to classify 1.2 million images into 1000 different classes. Since 2012, many articulate architectures have been proposed that are essentially built on the early architecture of LeCun, in order to improve the performance on ImageNet database [23]. However, the effective progress that was demonstrated on ImageNet for object classification by these pre-trained models has not shown the same success for the scene classification problem. Consequently, the first significant dataset for scene-centric images, referred to as Places, was proposed in Reference [8]. In general, indoor scene classification is challenging due to features' similarity in different categories. This problem has been studied with different learning methods as well as CNN, which so far has been employed in few studies. In Reference [24], a solution was proposed by designing a model that combined local and global information. The same problem was addressed by applying a probabilistic hierarchical model, which associates low-level features to objects via an object classifier, and objects to scenes via contextual relations [25]. There are also some research studies that have employed CNN for learning robots in indoor environments. Ursic et al. [26] addressed the room classification problem for household service robots. The performance of a pre-trained hybrid-CNN model was examined in Reference [8] on segmented images, i.e., learning through parts, of eight classes from the *Indoor67* dataset. The result generated 85.16% accuracy using a part-based model, which is close to the accuracy of the original hybrid-CNN, 86.45%. However, learning through parts gave much better accuracies on deformed images than the original model. The authors in Reference [27] took advantage of CNN for scene recognition in laboratory environments, with 89.9% accuracy, to enhance the indoor localization performance of a multi-sensor fusion system using smartphones. Furthermore, the objective of Reference [28] was to find the best retraining approach for a dynamically learning robot in indoor office environments. The paper examined and compared different approaches when adding new features from new images into a learned CNN model, considering the accuracy and training time. The new added images to the features database were the failed ones that were selected and corrected by the user. The authors simulated one of the categories to be the new environment. This paper reported that a pre-trained CNN with a KNN classifier was the most appropriate approach for real robots, as it gave a reasonable accuracy with the shortest training time. All their experiments were executed on the VidRILO dataset [29] using only its RGB frames, i.e., excluding the D frame. The methodology of this presentation, however, is different from previous studies, as we examined several CNN architectures with five categories of indoor scene rooms, i.e., bathrooms, bedrooms, dining rooms, kitchens, and living rooms downloaded from the Places dataset. In addition, these models were examined after cleaning and reducing the number of samples. Furthermore, a combination of CNN and multi-binary classifiers method called ECOC was proposed and evaluated in order to improve the real-time performance on a NAO humanoid robot.

Error correcting output code (ECOC) is a decomposition technique that was proposed by Dietterich and Bakiri for addressing multiclass learning problems [9]. There are a few studies in reported literature that have employed ECOC within CNN architecture, but in a different perspective than the employed in this paper. Deng et al. [30] used ECOC in order to address the target code issue by replacing the one-hot encoding with Hamming code in the last layer of CNN, which helped reduce the number of neurons in that layer. Then, the CNN model and CNN-ECOC, i.e., CNN with the new target codes, were trained and evaluated separately and the results were compared. Additionally, the same problem was solved in Reference [31] using a different code algorithm referred to as Hadamard code. ECOC within CNN has also been employed in medical applications [32,33], in which a pre-trained CNN was employed only for extracting features, then multi-binary SVM classifiers trained and combined with ECOC, referred to as ECOC-SVM. Up to our knowledge, this is the only reported work combining and fine-tuning CNN with ECOC for robotics applications that design multi-binary classifiers of CNN and compare the performance with regular CNN for multiple classes.

3. Adopted Approaches for Room Classification Problem

3.1. Convolutional Neural Network (CNN) Architectures

CNN is a category of neural networks that has been successfully employed for image detection and classification by extracting features such as edges, lines, shapes, or colors, without careful human design. The fundamental architectures are LeNet [22] and AlexNet [6]. In general, such networks have three main layers in each stage: convolutional, rectified linear unit (ReLu), and pooling layers. The architecture was completed by fully connected (FL) layers for classifying the image from high level features that have been extracted in multi-stages. In addition, the dropout technique was used to overcome the overfitting problem by ignoring some neurons in the training stage.

3.1.1. Convolutional Layers

The main purpose of these layers is to extract features from inputs $x = \{L_1, L_2, \ldots, L_i\}$, where x is an input image in the first stage while its input's features are in the middle stages. Every input can be divided in to a set of local receptive fields l_i. There are kernel filters, e.g., 3-by-3 trainable-filters $K = \{k_1, k_2, \ldots, k_j\}$ that are used to produce feature maps $FM = \{FM_1, FM_2, \ldots, FM_j\}$. The size of each kernel filter and each local receptive field is similar, in which they are used with the bias for calculating each output cell of FM_j as follows:

$$FM_j^i = b_j + \sum k_j \times L_i \tag{1}$$

3.1.2. ReLu Layers

As most real time applications are non-linear, the previous layer of linear filters is followed by a non-linear operation. The non-linear function used in most CNN models is *Rectified Linear Unit (ReLu).*

$$f(FM) = \max(0, FM) \tag{2}$$

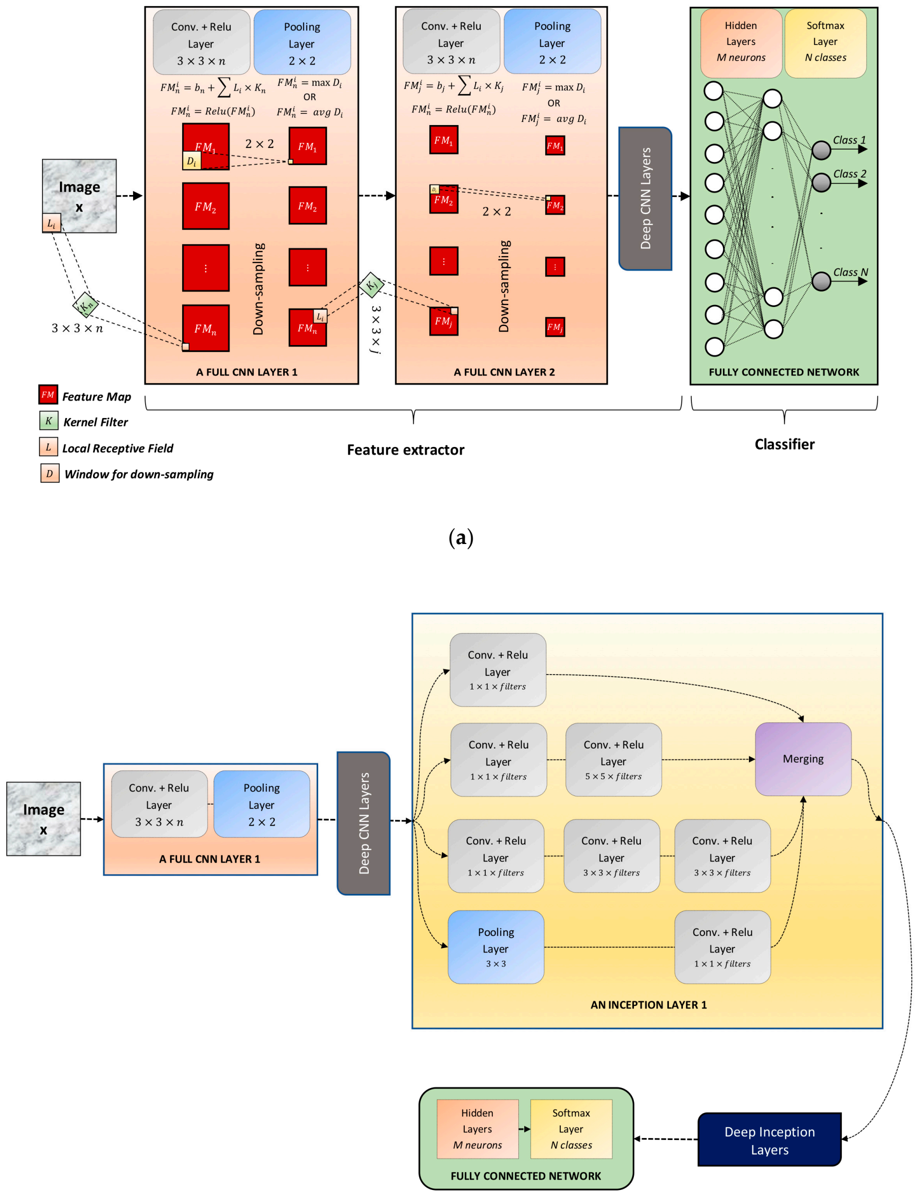
3.1.3. Pooling Layers

The role of this layer is to reduce the dimension of all features maps while keeping the most important information. It is carried out by taking the average or maximum value of every window of features maps.

3.1.4. FL Layers

This is a regular neural network that uses all high-level features for classifying an input image to specific class based on training a set of images.

Based on these fundamental architectures, many different CNN architectures have been crafted to improve overall performance. We considered several such architectures in this project based on their popularity and performance, i.e., VGGNet [34] and Inception [35]. The main objective of the VGG network is to improve the performance by increasing the depth of layers to 16 or 19. Meanwhile, the Inception network focuses on reducing the high computational cost in the VGG network by merging 3 × 3 and 5 × 5 filters that are preceded by 1 × 1 filters. Figure 1 shows the basic architecture of CNN and the improvement of the architecture in the Inception model.



(**a**)

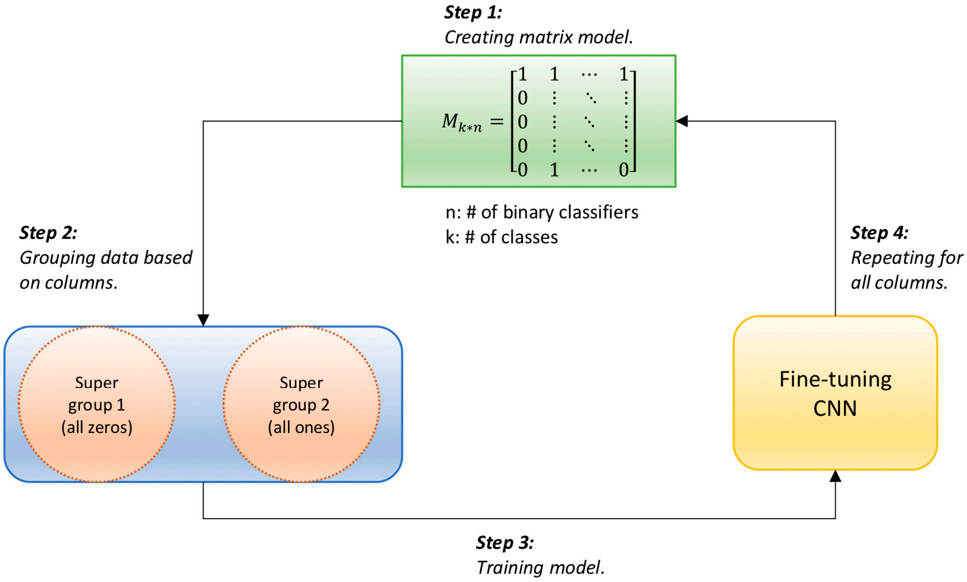
(**b**)

Figure 1. Convolutional neural network (CNN) architectures. (**a**) Fundamental components of CNN and (**b**) Inception [35].

3.2. Error Correcting Output Code (ECOC)

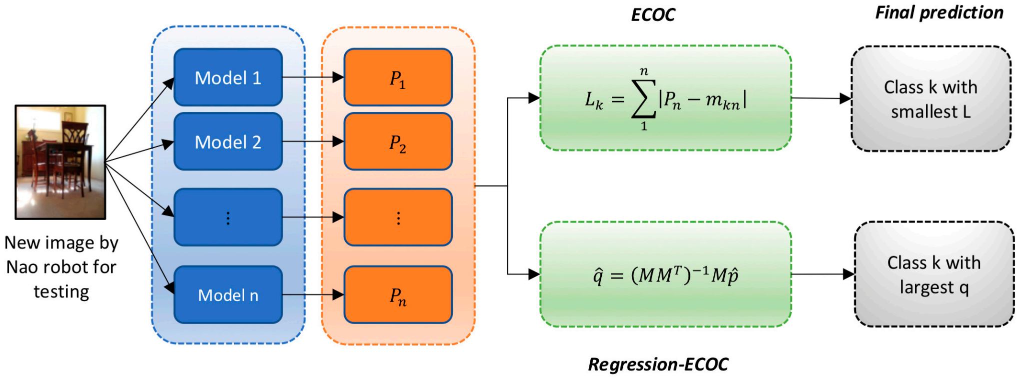
In practice, a single classifier for multiple room classes may not be sufficient to assist the robot in making the right decision, due to high similarities among the rooms. Thus, it is very important to find a way to improve the robot's best decision, even when applying a CNN classifier with high accuracy. We suggest adopting a decoding approach to address this problem [36]. The purpose of this technique is to improve the practical performance by designing a new classifier that combines multi-binary classifiers through an algorithm. This approach is also known in literature as decomposition [37] or plug-in classification technique (PICT) [38]. There were two main reasons for adopting this technique for this project. The first reason was to take advantage of higher accuracy with binary classes. The second reason was that designing multi-binary classifiers was feasible in the room classification problem, as the number of room classes in houses is limited.

One of the most popular decomposition techniques is error correcting output code (ECOC). It was proposed by Dietterich and Bakiri to address multiclass learning problems [9]. The main concept of this algorithm is to create a binary matrix code that represents multi-binary classifiers of two super groups. Each group consists of many classes, in order to alleviate the overall error in order to obtain the right classification. As shown in Figure 2, the algorithm consists of two stages. The first stage is to create and train multi-binary classifiers. It starts with creating a binary matrix code, in which the number of rows represents the number of classes and the number of columns represents the number of binary models. The data is then classified into two super groups based on zeros and ones in respective columns, where all classes with zeros are assembled in the first group, and the rest of classes are collected in the second group. The last step in this stage is to train all binary classifiers, i.e., the best CNN architecture for this problem, based on columns of the matrix using their super groups. The second stage is to predict a new image using all trained models. Each model gives a probability of predicting one of the super groups. After getting all probabilities of all models, we calculate the distance between all predictions p and each row in the matrix code. The smallest distance will be considered the correct class of that input image.


Step 1:
Creating matrix model.
$M_{k*n} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & \vdots & \ddots & \vdots \\ 0 & \vdots & \ddots & \vdots \\ 0 & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 0 \end{bmatrix}$
n: # of binary classifiers
k: # of classes
Step 2:
Grouping data based on columns.
Super group 1 (all zeros)
Super group 2 (all ones)
Step 4:
Repeating for all columns.
Fine-tuning CNN
Step 3:
Training model.

(a)

Figure 2. *Cont.*



(**b**)

Figure 2. Error correcting output code (ECOC) process for addressing multi class learning problems. (**a**) Stage 1: training all binary classifier processes, and (**b**) Stage 2: predicting a class of new image process.

The alternative way to get the overall classification is *Regression-ECOC*, which is using the least squares instead of Euclidian distance. Thus, the correct class will be the maximum value of the following equation:

$$\hat{q} = \left(MM^T\right)^{-1} M\,\hat{p} \tag{3}$$

4. Dataset & Simulation Experiments

The process of this work can be divided into three phases, as shown in Figure 3. Phase 1 was aimed at fine-tuning three different CNN models, i.e., VGG16, VGG19, and Inception V3, on five categories of rooms from the Places205 dataset, in order to select the best model for real experiments on a NAO robot. In addition, all models were examined in this phase after cleaning the dataset by removing all unrelated images to the scenes, and the results were compared before and after cleaning the dataset. In phase 2, the goal was to design multi-binary classifiers of the selected CNN from phase 1 and combine their results through the ECOC algorithm and ECOC-REG. Finally, testing the selected CNN, CNN-ECOC, and CNN-ECOC REG through real experiments on a NAO robot, and comparing the results was the goal of phase 3. The importance of this phase is to show how these models performed on images from robots, e.g., NAO, in which those images are quite different in the level of view from the existed dataset. Phase 1 and phase 2 are explained in detail in this section, whereas phase 3 is explained in the next section.

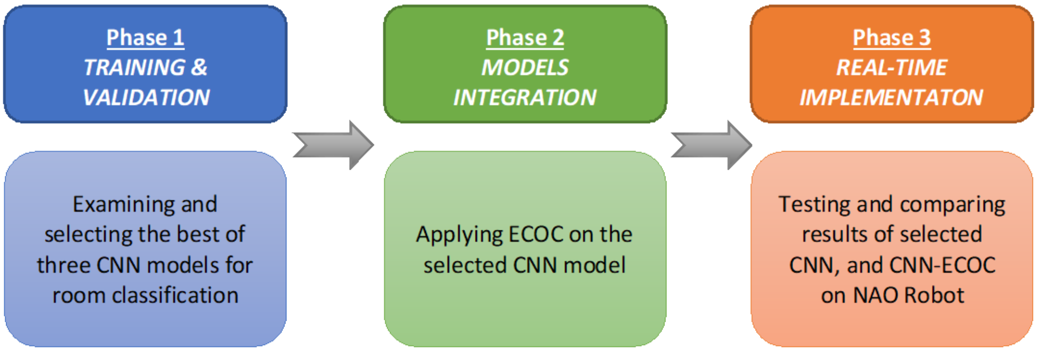

Phase 1
TRAINING & VALIDATION
Phase 2
MODELS INTEGRATION
Phase 3
REAL-TIME IMPLEMENTATON
Examining and selecting the best of three CNN models for room classification
Applying ECOC on the selected CNN model
Testing and comparing results of selected CNN, and CNN-ECOC on NAO Robot

Figure 3. Process of simulation and real-time implementation.

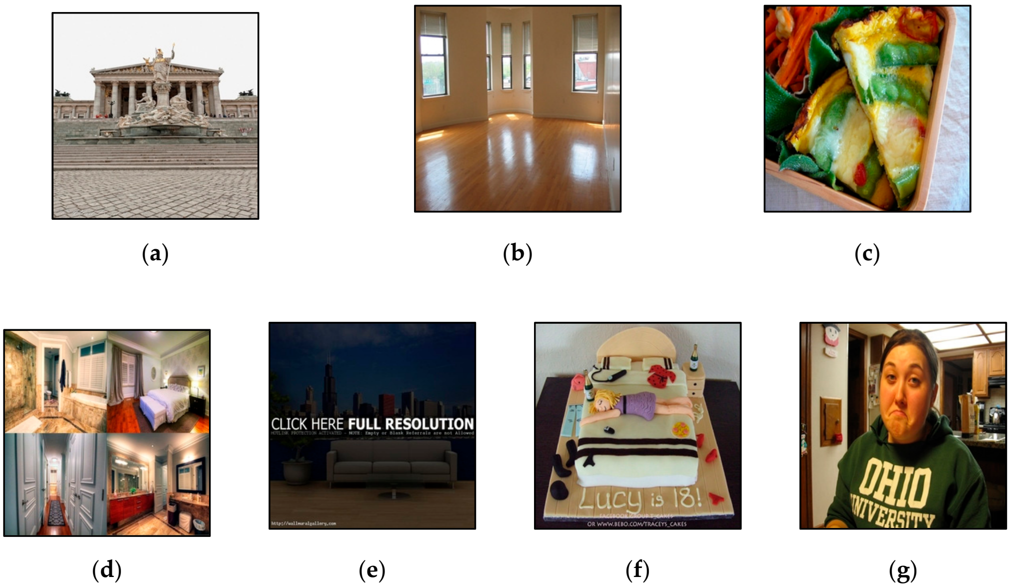
4.1. Scene Dataset

There are several scene datasets proposed in the literature for addressing object/scene detection or classification problems. Some of them are small-scale datasets such as the 15-scene dataset, UIUC Sports, and CMU 300, and some are large-scale datasets such as 80 Million Tiny Image Dataset, PASCAL, ImageNet, LabelMe, SUN, and Places [39]. The dataset can be 3D scene, such as SUNCG [40], or it can be images for a particular environment with geo-referenced pose information for each image, such as TUM and NavVis [41]. These datasets can be classified into two view types: object-centric datasets, e.g., ImageNet, and scene-centric datasets, e.g., Places [8]. Places is the latest and largest scene-centric dataset, which is provided by MIT Computer Science and Artificial Intelligence Laboratory for the purpose of CNN training. It has a repository of around 2.5 million images classified into 205 categories, and for this reason it is called the Places205 dataset. This dataset is updated and extended with more images classified into 365 categories in Reference [42], which is called Places365.

Since this project is within the scope of household robotics applications, five categories of images were selected to be downloaded from Places205 for addressing room–scene classification problems for social robots using CNN models. The five categories are: bedroom, dining-room, kitchen, living-room, and bathroom, which most, if not all, houses have. It should be noted that the corridor category is not available in Places205 and Places365 at the time of this work. This category is important in this research, and will be incorporated in the design once it is available. 11,600 images/category were used to train the CNN model, where 20% of images were used for validation, i.e., 2320 images/category.

Cleaning Data

The Places dataset is regarded to be very important in the field of computer vision and deep learning. However, there are some issues with the downloaded images for real time robotic applications that affect the learning process. Therefore, we manually excluded some images from all five categories, based on criteria that are shown in the few examples in Figure 4.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**) (**g**)

Figure 4. Examples of removed images from room dataset of Places. (**a**) Not belonged to; (**b**) no furniture; (**c**) not wide scene; (**d**) multi-scenes in an image; (**e**) including texts; (**f**) fake scene; (**g**) focusing on people.

Table 1 shows the percentage of the data that were deemed irrelevant form each category. After cleaning the data, we noted that the remaining images for bedrooms were the highest and for kitchens were the lowest (Table 1). The reader might note the high percentage of irrelevant images in each category, which justifies the need for cleaning data.

Table 1. Number of images for each class after cleaning.

Class	# Images out of 11,600	% of Removed Images
Bedrooms	9323	19.6%
Dining Rooms	7919	31.7%
Kitchens	6611	43.0%
Living Rooms	8571	26.1%
Bathrooms	8959	22.8%

4.2. Multi-Class Room Classification Experiments Using Fine-Tuning Pre-Trained Models

Since our main concern was to recognize only five room classes, we did not require huge amounts of data. Additionally, the learned features from pre-trained models in literature are relevant to the room classification problem, therefore fine-tuning pre-trained models was the best strategy for this work, instead of training a CNN model from scratch. Fine-tuning can be achieved through two main steps. The first step is to proceed room images to non-trainable ConvNet in order to extract features, then use these features to train our new classifier, i.e., softmax layer. The second step is to retrain the whole network, i.e., ConvNet and classifier, with a smaller learning rate, while freezing a few layers of the ConvNet.

All experiments were completed through the Graham cluster provided by Compute Canada Database [43] using Keras API, which is written in a python deep learning library [44]. Several CNN models were fine-tuned for this project, i.e., VGG16, VGG19, and Inception V3, with different freezing layers to be trained. All these CNN models were followed by a similar fully connected (FC) layer. FC begins with an average pooling layer, then a layer of 1024 neurons with the *relu* activation function, and ends with a logistic layer to predict one of the five classes. Keras provides a compile method with different optimizers for learning process. In the first stage of the fine-tuning, the *adam* optimizer was used with a 0.001 learning rate, whereas we applied a SGD (stochastic gradient descent) optimizer in the second stage with learning rate of 0.0001 and momentum of 0.9. All models were trained for 10 epochs in each stage with both the original data as well as the cleaned data, and with different non-trainable layers. It was noticed that training with more epochs did not provide that much improvement in the final accuracy, but it took a very long time in the training process. Table 2 shows the superior results of all models trained with clean data compared to all data. The best result shown from these experiments is VGG19 and VGG16 with 0 freezing layers using clean data, which gives an accuracy of 93.61% and 93.29%, respectively.

Table 2. Comparison of accuracies of fine-tuning different CNN models using all data and clean data (the yellow shaded ones are the best for the real-time experiment).

CNN Models	Non-Trainable Layers	All Data		Clean Data	
		Time	*Accuracy %*	*Time*	*Accuracy %*
VGG16	*15*	11:50:40	86.03	8:16:16	89.69
	11	11:50:41	88.09	8:15:38	91.49
	7	11:53:42	88.9	8:18:49	93.22
	0	12:13:55	87.78	8:34:45	93.29
VGG19	*20*	13:11:07	78.69	9:16:00	82.50
	17	13:14:56	86.22	9:18:38	89.65
	0	13:43:40	90.30	9:40:52	93.61
Inception V3	*299*	10:17:29	75.12	7:8:33	78.83
	249	10:17:46	79.11	7:07:50	84.05

4.3. Binary-Class Room Classification Experiments Using a Matrix Code

The best two models in phase 1 were VGG19 and VGG16 with all layers fine-tuned. Although this work was carried out through one of Compute-Canada Servers, i.e., Graham, there are many works in robotic applications that can be processed using local machines. Therefore, considering the time of training is an important factor for this phase, which has multiple binary classifiers for training. For this reason, the selected model to be trained in this phase was the VGG16 with 0 freezing layers, which has an accuracy of 93.29%, which is quite similar to the best one. The binary classifiers can be designed through grouping classes based on an exhausted matrix code, as explained in Reference [9]. The following 5 × 15 matrix is the best for this experiment, as it does not have any repeated and complimented columns.

$$M_{5X15} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} \textit{bathroom} \\ \textit{bedroom} \\ \textit{dining room} \\ \textit{kitchen} \\ \textit{living room} \end{bmatrix} \quad (4)$$

Let us take an example of the classifier in column 3 of matrix M, which has [1 0 0 1 0] values. The first group of this classifier would be the classes with zero value, i.e., bedrooms, dining rooms, and living rooms. The second group is the classes with ones, i.e., bathrooms & kitchens. Table 3 shows the validation accuracies of all 15 binary fine-tuned VGG16 classifiers. The main advantage of the binary classifier is high accuracy depending on classification, as it reached 98.5% for this project, and the average of all 15 classifiers was 95.37%, which is still higher than the multi-classification approach.

Table 3. 15 binary classifier accuracies.

Binary Classifiers	**1 Bath vs. All**	**2**	**3**	**4**	**5**	**6**	**7**	**8 Bed vs. All**	**9**	**10**	**11**	**12 Dining vs. All**	**13**	**14 Kitchen vs. All**	**15 Living vs. All**	**Average**
Accuracy %	***98.50***	*93.62*	*97.38*	*92.01*	*94.84*	*93.95*	*96.06*	*95.73*	*95.59*	*96.10*	*94.94*	*95.89*	*92.76*	*97.95*	*95.31*	***95.37***

4.4. Discussion

One of the most important features that has been studied by researchers is deepness, which is significant in most of the known CNN architectures. It is reported that the deeper and wider the architecture is designed, the better features will be learned [35]. However, this requires a huge dataset, i.e., millions of samples, in order to avoid the overfitting issue. Unfortunately, this is not always the case in robotics applications, wherein in most applications, the number of classes for a specific robotic problem such as room classification is very limited, which means the number of samples might be only thousands, i.e., a small dataset. Therefore, the very deep CNN architecture will most likely lead to overfitting problem, as happened with ResNet [45] in this work, which is why their results are excluded from these experiments. Although the Inception V3 results were not over-fitted, they were less accurate than the architectures with less deepness, i.e., VGG16 and VGG19. The reason might be related to the scene-centric type of dataset, in which learning hierarchical representations can be difficult with more deepness. For this work, we had two ways to address this problem for robotic application: either designing a new CNN architecture for a small dataset, or adopting an existing CNN with less deepness and improving robot's decision by integrating another method. The latter solution was preferable, so we adopted VGG16, i.e., the least deepness of all three architectures, and integrated it with ECOC in a way it was practicable for real time robotic implementation. One could ask why we adopted CNN from the beginning—as discussed before, CNN extracts and learns features from raw images without requiring a careful engineering design that shown superiority over conventional approaches in computer vision.

Binary classifier results for ECOC explain the challenge of feature similarities in a house's rooms. Let us discuss the obvious results of 'class vs. all' in classifiers number 1, 8, 12, 14, and 15. The most distinguishable rooms are the bathroom and kitchen, as shown in classifiers 1 and 14 respectively. Meanwhile, the other three classes are very similar to each other. There are many reasons related to the dataset or the architecture of VGG16 that the model is less accurate with these similar rooms than the distinguishable ones. The first reason is having some sharable objects in different rooms, such as tables or TVs, or the wide variety styles of those rooms such as open/closed spaces, or even culture-based styles, e.g., no beds for sleeping. The second reason is that the architecture with less deepness will not be able to differentiate between objects similar in shape, e.g., rectangular shapes in dining tables, coffee tables, and beds from different rooms. Therefore, it is a tradeoff between learning deep features and having a small dataset. For our future work, we will consider the associated models between scenes and objects for better robot decisions.

5. Real-Time Experiment on NAO Humanoid Robot

The proposed solution for integration of CNN and the multi-binary classifiers for room classification were tested on a NAO humanoid robot. The results are compared and discussed in the next sub-sections, after giving an overview about the platform adopted in this work.

5.1. NAO Humanoid Robot

The physical platform, shown in Figure 5, used to experimentally validate this work is the humanoid robot NAO V4 (H25), i.e., twenty five degrees of freedom (DOF). Twenty five DOF means the robot consists of 25 different motors for controlling different actuators. NAO is a human-like, medium-sized humanoid robot, and is fully programmable to perform many tasks autonomously. It was designed by Aldebaran Company, owned afterward by SoftBank Group. It is equipped with many proprioceptive and exteroceptive sensors. Proprioceptive sensors acquire internal information of the robot, while exteroceptive sensors acquire external information of the environment. For this project, the top camera, which located in robot's forehead, was used to capture room images. The important specification for this work is the height of the robot, which is 573 mm, while the top camera was at the level of 514.29 mm. Through the main software of Nao, i.e., *Naoqi*, several methods can be used from *AlPhotoCapture* module for real experiments.

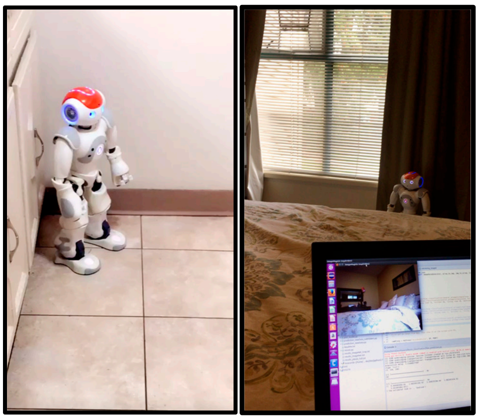


Figure 5. NAO humanoid robot during experiments.

5.2. Practical Implementation Results

The proposed methods were tested and compared practically in five different houses using the NAO humanoid robot. The goal was that the robot should be able to predict the room class using the three different models, i.e., CNN, CNN-ECOC & CNN-ECOC-REG, with all models returning the probability for each class. Since rooms in the houses had different sizes, layouts, furniture, etc., NAO was positioned in different spots in the rooms, i.e., center, corners, beside the wall or the door, during different time of the day under different light conditions. Accordingly, 56 images were taken by its top camera as follows: 12 bathrooms, 13 bedrooms, 6 dining rooms, 13 kitchens, and 12 living rooms, while the robot's head faced the x direction and the top camera covered 60.97° and 47.64° on the y and z directions, respectively. Figure 6 shows some examples of scenes taken by the NAO humanoid robot. An important point to be noted is that NAO is very short compared to an average human's height, and its camera is mounted about 514 mm from the floor level. This is quite different from the field of view of the images from the adopted dataset. Consequently, the highest (top-1) probabilities were negatively affected, especially in the kitchen class. However, the second highest (top-2) probabilities show the superiority of CNN-ECOC and CNN-ECOC-REG over the regular CNN for prediction of most of the five classes.

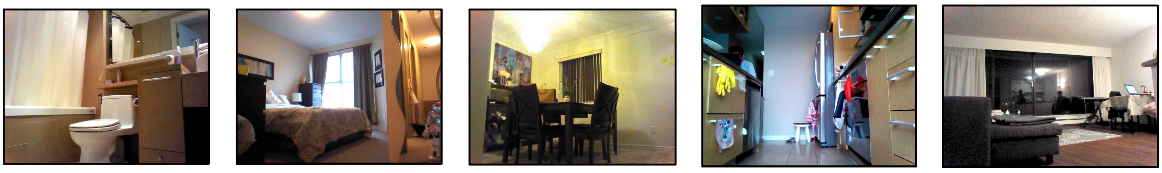


Figure 6. Scenes examples taken by NAO humanoid robot.

Table 4 shows the testing accuracy of the five classes, whereas Table 5 gives confusion matrix for both top-1 and top-2 predictions. Notice that the confusion matrix of top-1 tables has all 56 images, however the confusion matrix of top-2 tables includes only the number of false predicted images in the top-1 table. The best room prediction was the bathroom, where accuracy was 100% with all three models. Therefore, the bathroom was considered to be the most distinctive room. All three models were similar at predicting bedrooms, in which their top-1 accuracy was 69.2%, increasing to 100% in top-2. The distinction in performance between these models is shown in the prediction of dining rooms, kitchens, and living rooms. The top-1 predictions of dining rooms for three models were similar, however, the top-2 prediction of CNN-ECOC and CNN-ECOC-REG increased to 100%, which was better than CNN. In the living room cases, the CNN and CNN-ECOC-REG gave the best rate of 50% in the top-1, however the CNN-ECOC and CNN-ECOC-REG were able to increase up to 91.7% and 100% in their top-2 predictions, respectively. Although the kitchen top-1 prediction was the worst in all models, it surprisingly showed a huge improvement in top-2 prediction for CNN-ECOC and CNN-ECOC-REG compared to CNN. Overall, the three models performed similarly in the top-1 prediction, however the multi-binary classifier solutions, i.e., CNN-ECOC and CNN-ECOC-REG, gave much better performance in the top-2 results in contrast to the CNN for multi-classification.

Table 4. Testing accuracies of all models with top-1 (T1) and top-2 (T2) for all five classes.

Model	**Bath**		**Bed**		**Dining**		**Kitchen**		**Living**	
	T1	**T2**	**T1**	**T2**	**T1**	**T2**	**T1**	**T2**	**T1**	**T2**
CNN	*100*	-	*69.2*	*100*	*66.7*	*83.3*	*15.4*	*61.5*	*50*	*75*
CNN-ECOC	*100*	-	*69.2*	*100*	*66.7*	*100*	*15.4*	*100*	*41.7*	*91.7*
CNN-ECOC-REG	*100*	-	*69.2*	*100*	*66.7*	*100*	*30.8*	*100*	*50*	*100*

Table 5. Confusion matrix of testing images with top-1 & top-2 for five classes.

		# Images	***# False Images in Top 1***	***bath***	***bed***	***din***	***kit***	***liv***
Top-1 with CNN	*bath*	*12*		12				
	bed	*13*		4	9			
	din	*6*		2		4		
	kit	*13*		10			2	1
	liv	*12*		4	2			6
Top-2 with CNN	*bath*		*0*					
	bed		*4*		4			
	din		*2*		1	1		
	kit		*11*	1	4		6	
	liv		*6*		2	1		3
Top-1 with CNN-ECOC	*bath*	*12*		12				
	bed	*13*		4	9			
	din	*6*		2		4		
	kit	*13*		11			2	
	liv	*12*		6	1			5
Top-2 with CNN-ECOC	*bath*		*0*					
	bed		*4*		4			
	din		*2*		1	1		
	kit		*11*				11	
	liv		*7*			1		6
Top-1 with CNN-ECOC-REG	*bath*	*12*		12				
	bed	*13*		4	9			
	din	*6*		2		4		
	kit	*13*		9			4	
	liv	*12*		4	1	1		6
Top-2 with CNN-ECOC-REG	*bath*		*0*					
	bed		*4*		4			
	din		*2*			2		
	kit		*9*				9	
	liv		*6*					6

5.3. Discussion

After obtaining results of the multi-binary classifiers in Table 3 we expected to get the best results with the most distinguishable rooms, i.e., bathrooms & kitchens, on phase 3. The results were as expected with bathrooms, but not with kitchens. The reason is the short height of NAO, which captured scenes that are totally different from what was learnt from the dataset. In the kitchen case, the robot captured the cabinets and drawers rather than capturing the top view of stoves or other appliances. Thus, the robot's first prediction was mostly bathrooms instead of kitchens. The ideal solution is to have a room dataset for robotics in a height range of 0.5–1.0 m, but this would be an expensive process. The other solution, the approach adopted in this work, is to use an existing dataset for training and validating a learning model, and then to design a method for real time experiments with the robot. For this work, the improvement was shown in the second decision of the robot, so how the robot can make its decision? The simple way is that the robot can select the second prediction only when the first prediction probability is below a threshold. The other solution is to make a decision based on a combination of multiple images from the same room.

6. Conclusions and Future Work

This paper focused on addressing the room classification problem for social robots. The CNN deep learning approach was adopted for this purpose because of its superiority in the areas of object detection and classification. Several CNN architectures were examined by fine-tuning them on five room classifications of the Places dataset, in order to find out the best model for real life experiments. It was found that VGG16 is the best adopted model, with 93.29% of validation accuracy after cleaning the dataset by excluding all mislabeled images. In addition, we proposed and examined a combination of CNN with ECOC, a multi-binary classifier approach, in order to address the error in practical prediction. The validation accuracy reached 98.5% in one of the binary classifiers and 95.37% in the average of all binary classifiers. The CNN and the combination model of CNN and ECOC in both forms, i.e., CNN-ECOC and CNN-ECOC-REG, were evaluated practically on a NAO humanoid robot. The results show the superiority of the combination model over the regular CNN.

There are many challenges that should be considered for future work. First, the real time experiments for domestic robots need their own dataset which is compatible with most social robots' heights. For example, the images captured by NAO robots are almost at the level of 0.5 m from the floor. This implies that the real time prediction will be negatively affected if the model is trained using the existing datasets. Second, a final prediction based on one image is practically not sufficient for social robots, as the captured images depend on the angle of the view or other image factors, e.g., resolution or light. Therefore, combining many images of the same room or many frames from a video are important for getting the right decision. Third, corridors in houses are an important class that should be added for future work. Lastly, detecting and locating the door through the same CNN architecture is significant for the purpose of indoor navigation.

Author Contributions: K.M.O. is a PhD candidate undertaking his research under the supervision of A.B.R.

Funding: This research received no external funding.

Acknowledgments: The first author acknowledges the financial support of Umm Al-Qura University in Saudi Arabia that represented by the Saudi Arabian Cultural Bureau in Canada. This research was enabled in part by support provided by WestGrid (https://www.westgrid.ca/) and Compute Canada (https://www.computecanada.ca).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Campa, R. The Rise of Social Robots: A Review of the Recent Literature. *J. Evol. Technol.* **2016**, *26*, 106–113.
2. Mejia, C. Bibliometric Analysis of Social Robotics Research: Identifying Research Trends and Knowledgebase. *Appl. Sci.* **2017**, *7*, 1316. [CrossRef]
3. Louridas, P.; Ebert, C. Machine Learning. *IEEE Softw.* **2016**, *33*, 110–115. [CrossRef]
4. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
5. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2323. [CrossRef]
6. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *1*, 1097–1105. [CrossRef]
7. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
8. Zhou, B.; Lapedriza, A.; Xiao, J.; Torralba, A.; Oliva, A. Learning Deep Features for Scene Recognition using Places Database. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 487–495.
9. Dietterich, T.G.; Bakiri, G. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *J. Artif. Intell. Res.* **1995**, *2*, 263–286. [CrossRef]
10. SoftBank Robotics. Available online: Https://www.ald.softbankrobotics.com/en/press/press-releases/softbank-increases-its-interest (accessed on 29 January 2019).
11. Mozos, O.M.; Stachniss, C.; Burgard, W. Supervised learning of places from range data using AdaBoost. In Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; Volume 2005, pp. 1730–1735.

12. Rottmann, A.; Mozos, Ó.M.; Stachniss, C.; Burgard, W. Semantic Place Classification of Indoor Environments with Mobile Robots using Boosting. In Proceedings of the 20th National Conference on Artificial Intelligence, Pittsburgh, PA, USA, 9–13 July 2005; Volume 3, pp. 1306–1311.
13. Mozos, Ó.M.; Triebel, R.; Jensfelt, P.; Rottmann, A.; Burgard, W. Supervised semantic labeling of places using information extracted from sensor data. *Rob. Auton. Syst.* **2007**, *55*, 391–402. [CrossRef]
14. Ayers, B.; Boutell, M. Home interior classification using SIFT keypoint histograms. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007.
15. Ursic, P.; Kristan, M.; Skocaj, D.; Leonardis, A. Room classification using a hierarchical representation of space. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Algarve, Portugal, 7–12 October 2012; pp. 1371–1378.
16. Swadzba, A.; Wachsmuth, S. Indoor scene classification using combined 3d and gist features. In Proceedings of the 10th Asian Conference on Computer Vision, Queenstown, New Zealand, 8–12 November 2010; Lecture Notes in Computer Science. Springer: Berlin/Heidelberg, Germany, 2011; Volume 6493, pp. 201–215.
17. Mozos, O.M.; Mizutani, H.; Kurazume, R.; Hasegawa, T. Categorization of indoor places using the Kinect sensor. *Sensors (Switzerland)* **2012**, *12*, 6695–6711. [CrossRef] [PubMed]
18. Zivkovic, Z.; Booij, O.; Kröse, B. From images to rooms. *Rob. Auton. Syst.* **2007**, *55*, 411–418. [CrossRef]
19. Varadarajan, K.M.; Vincze, M. Functional Room Detection and Modeling using Stereo Imagery in Domestic Environments. In Proceedings of the Workshop on Semantic Perception, Mapping and Exploration at IEEE International Conference on Robotics and Automation (ICRA 2011), Shanghai, China, 9–13 May 2011.
20. Varvadoukas, T.; Giannakidou, E.; Gómez, J.V.; Mavridis, N. Indoor furniture and room recognition for a robot using internet-derived models and object context. In Proceedings of the 10th International Conference on Frontiers of Information Technology (FIT 2012), Islamabad, Pakistan, 17–19 December 2012; pp. 122–128.
21. Jackel, L.D.L.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; LeCun, B.; Denker, J.; Henderson, D. Handwritten Digit Recognition with a Back-Propagation Network. *Adv. Neural Inf. Process. Syst.* **1990**, *2*, 396–404.
22. LeCun, Y.; Kavukcuoglu, K.; Farabet, C. Convolutional networks and applications in vision. In Proceedings of the IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems, Paris, France, 30 May–2 June 2010; pp. 253–256.
23. Canziani, A.; Paszke, A.; Culurciello, E. An Analysis of Deep Neural Network Models for Practical Applications. *arxiv* **2016**, arXiv:1605.0767.
24. Quattoni, A.; Torralba, A. Recognizing indoor scenes. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 413–420.
25. Espinace, P.; Kollar, T.; Soto, A.; Roy, N. Indoor scene recognition through object detection. In Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 4–8 May 2010; pp. 1406–1413.
26. Ursic, P.; Mandeljc, R.; Leonardis, A.; Kristan, M. Part-based room categorization for household service robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 2287–2294.
27. Liu, M.; Chen, R.; Li, D.; Chen, Y.; Guo, G.; Cao, Z.; Pan, Y. Scene Recognition for Indoor Localization Using a Multi-Sensor Fusion Approach. *Sensors* **2017**, *17*, 2847. [CrossRef] [PubMed]
28. Cruz, E.; Rangel, J.C.; Gomez-Donoso, F.; Bauer, Z.; Cazorla, M.; Garcia-Rodriguez, J. Finding the Place: How to Train and Use Convolutional Neural Networks for a Dynamically Learning Robot. In Proceedings of the International Joint Conference on Neural Networks, Rio de Janeiro, Brazil, 8–13 July 2018.
29. Martínez-Gómez, J.; García-Varea, I.; Cazorla, M.; Morell, V. ViDRILO: The visual and depth robot indoor localization with objects information dataset. *Int. J. Rob. Res.* **2015**, *34*, 1681–1687. [CrossRef]
30. Deng, H.; Stathopoulos, G.; Suen, C.Y. Error-correcting output coding for the convolutional neural network for optical character recognition. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), Barcelona, Spain, 26–29 July 2009.
31. Yang, S.; Luo, P.; Loy, C.C.; Shum, K.; Tang, X. Deep Representation Learning with Target Coding. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.

32. Abd-Ellah, M.K.; Awad, A.I.; Khalaf, A.A.M.; Hamed, H.F.A. Two-phase multi-model automatic brain tumour diagnosis system from magnetic resonance images using convolutional neural networks. *EURASIP J. Image Video Process.* **2018**, *2018*, 97. [CrossRef]
33. Dorj, U.O.; Lee, K.K.; Choi, J.Y.; Lee, M. The skin cancer classification using deep convolutional neural network. *Multimed. Tools Appl.* **2018**, *77*, 9909–9924. [CrossRef]
34. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
35. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
36. Rocha, A.; Goldenstein, S.K. Multiclass from binary: Expanding One-versus-all, one-versus-one and ECOC-based approaches. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 289–302. [CrossRef]
37. Aly, M. Survey on multiclass classification methods. *Neural Netw.* **2005**, *19*, 1–9.
38. James, G.; Hastie, T. The error coding method and PICTs? *J. Comput. Graph. Stat.* **1998**, *7*, 377–387.
39. Chen, C.; Ren, Y.; Jay, K.C. *Big Visual Data Analysis Scene Classification and Geometric Labeling*; Briefs in Electrical and Computer Engineering; Springer: Singapore, 2016.
40. Song, S.; Yu, F.; Zeng, A.; Chang, A.X.; Savva, M.; Funkhouser, T. Semantic scene completion from a single depth image. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), San Juan, Puerto Rico, USA, 24–30 June 2017; Volume 2017, pp. 190–198.
41. Walch, F.; Hazirbas, C.; Leal-Taixe, L.; Sattler, T.; Hilsenbeck, S.; Cremers, D. Image-Based Localization Using LSTMs for Structured Feature Correlation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; Volume 2017, pp. 627–637.
42. Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; Torralba, A. Places: A 10 million Image Database for Scene Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**. [CrossRef] [PubMed]
43. Compute Canada. Available online: https://www.computecanada.ca (accessed on 29 January 2019).
44. Keras Documentation. Available online: https://keras.io (accessed on 29 January 2019).
45. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.