**Capstone Project**

**Machine Learning Nanodegree**

**Indoor Classification**

1. **Definition:**
   1. **Project overview:**

Machine learning models has been the flavor of artificial intelligence field especially convolution neural network which exhibit a very prominent result in detecting patterns in images.

As engineers we have a great responsibility towards people to help them and making their life easier, for example helping blind people to easily interpret their surrounding and to easily interact with the surrounding environment.

This project aim to help blind people identify the different type of rooms in home by taking frames from a real-time video and process this frame and identify the type of the room and pronounce it loud to be heard.

To solve the problem in this project I used a transfer learning technique like Resnet50 and Mobilenet v2 , so to use the transfer learning you need preprocess the data like converting all the dataset into 4D tensor array by loading the data and converting it to numpy array and then expand the dimension. After creating the model and train it , I have implemented an algorithm to take real-time frames from the camera and pronounce the result of the classification loud using speakers.

* 1. **Domain background:**

Machine learning models has been the flavor of artificial intelligence field especially convolution neural network which exhibit a very prominent result in detecting patterns in images.

As engineers we have a great responsibility towards people to help them and making their life easier, for example helping blind people to easily interpret their surrounding and to easily interact with the surrounding environment.

* 1. **Datasets and inputs:**

The dataset contain 9 classes [‘bathroom’ , ’bedroom’ , ’closet’ , ’corridor’ , ’dining\_room’ , ’kitchen’ , ’living\_room’ , ’pantry’ , ’staircase’]

Total number of images =2122

* Training set =1520
* Validation set=243
* Testing set=359
  1. **Citation:**

Scene Parsing through ADE20K Dataset. Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso and Antonio Torralba. Computer Vision and Pattern Recognition (CVPR), 2017.

Semantic Understanding of Scenes through ADE20K Dataset. Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso and Antonio Torralba.arXiv:1608.05442.

Quattoni, and A.Torralba. Recognizing Indoor Scenes. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

* 1. **Problem Statement:**

Blind people have many difficulties to identify objects and they need to be assisted through daily life and navigating through homes.

Indoor recognition is a challenging problem as most recognition scene models were specially made for outdoor recognition and have poor performance in indoor classification.

The aim of this project is to help blind people easily navigate through homes by classifying the type of each room in a house whether it’s a bedroom ,kitchen ,bathroom ,dining room and other rooms that may exist in a home.

* 1. **Evaluation Metrics:**

The evaluation metric is based on accuracy ,which is being calculated by the given formula.

Accuracy=(True positive + True negative)/(True negative + True positive +False positive +False negative) which mean that we are looping through all the test dataset and compare the result of the predicted frame to its true value ,if the model predicted right we give a point and if not no point given, in the end we accumulate all the points and divided it by all the tested frames.

The accuracy model is suitable in this project because the all the dataset of each class is almost in equal quantity and there is no biasing for recall model or precision model as it’s not something dangerous like predicting a real sick person as healthy.

Another evaluation that also need to be considered is how fast the program can takes frames and process it ,because real time application need to be fast and interactive.

1. **Analysis:**
2. **Data Exploration:**

The dataset organized into folders ,each folder represent a one category and each folders contain images in .jpg format and each image have object and part segmentations are stored in two different .png format ,and also contain .txt format file which describe the object and parts in the image.

The dataset contain many instances and many category but in this project we will only use ten category that is suitable for home classification room .

The dimension of images used are different from each other, but they are all bigger than 256X256 which make it suitable because we are reducing the dimensionality of the images to 224X224.

Those images are a sample of the training data.



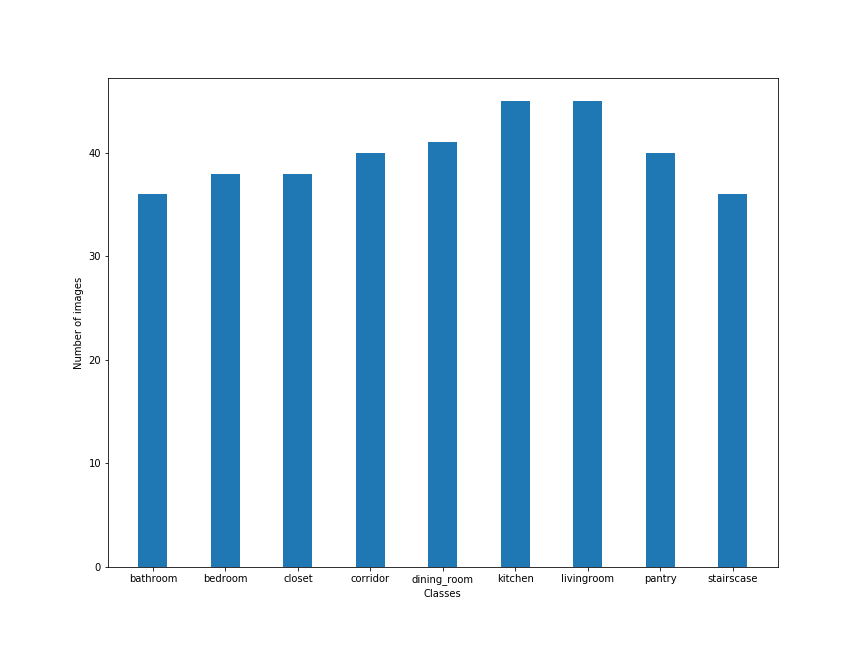
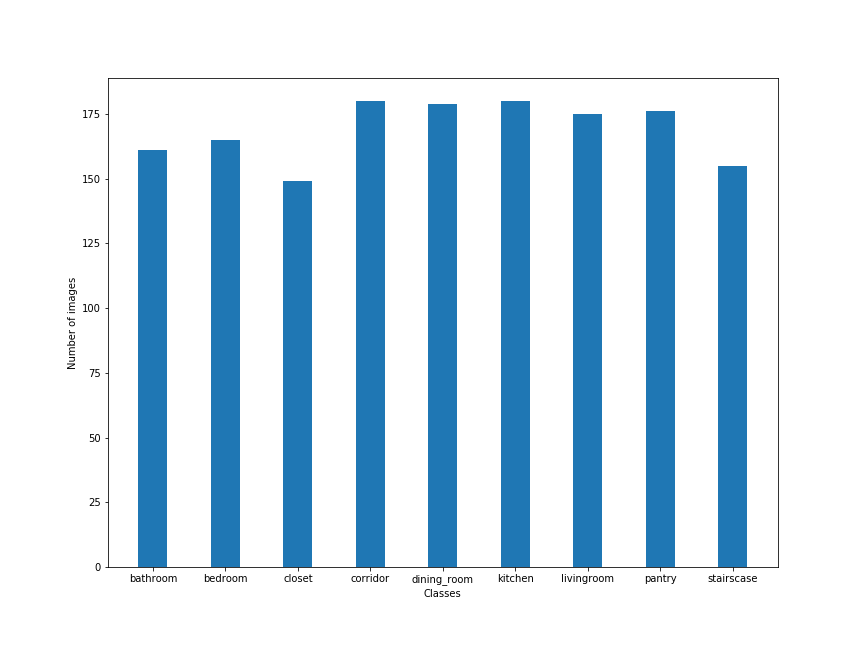


In figure(1) below is showing the number of images for each class in the training dataset.

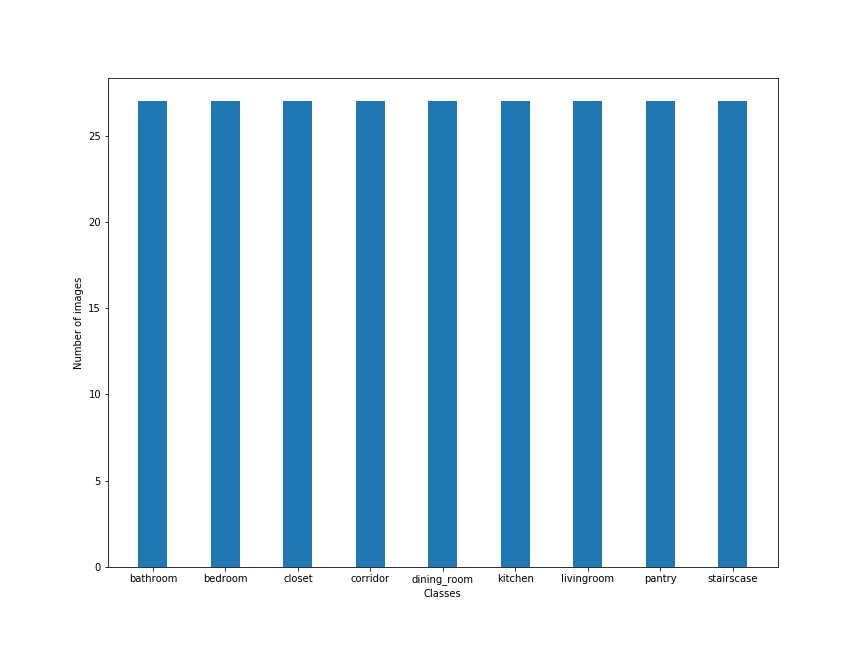
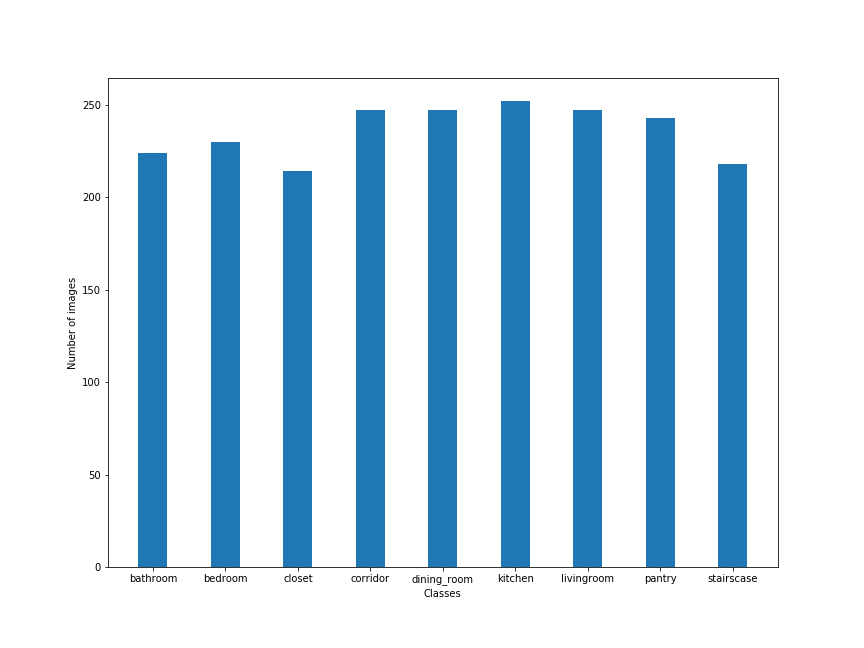
In figure(2) below is showing the number of images for each class in the testing dataset.

In figure(3) below is showing the number of images for each class in the validation dataset.

In figure(4) below is showing the number of images for each class in the all dataset.

** Fig(1)**

**Fig(2)**

** Fig(3)**

**Fig(4)**

The dataset is organized in folders where each folder represent a class.

The dataset can be downloaded from the following links [dataset1](http://groups.csail.mit.edu/vision/datasets/ADE20K/) and [dataset2](http://web.mit.edu/torralba/www/indoor.html)

1. **Algorithms and techniques**

In this section I am going to illustrate the basics of how convolution neural network work, well a CNN work as a layers of filters that can detect and extract complex pattern from images and sometimes its called “feature extractor”.

There are three types of main layer that is used to build a convoulution neural network :

1. Convolutional layer which is defined by the following parameter like “Number of filters used ,kernel size , strides , padding ,activation function , input size ”.
2. Pooling layer which is a layer responsible for reducing the dimensionality of input which is defined by following parameter ”the window size and the type of pooling performed whether it’s a max pooling or average pooling”.
3. The final dense layer where each node in the these layer represent an output class ,its possible to have many dense layer with different number of nodes but in the last layer ”output layer” the number of nodes “neurons” must be equal to the number of outputs.

There is two classifier used in this project both classifier are based on the Convolution neural network.

The first classifier is based on the residual neural network “Resnet50” and the second one is Mobile Network “Mobilenet V2” both classifier are already built in Keras­­.

Finally I implemented two function ,the first function is responsible for taking a stream of photo using a digital camera and convert the frame into 224X224 image to be fed to the classifier and the other function take the output of classifier and pronounce it loud using speakers.

1. **Benchmark:**

I have used a random model classifier as my benchmark and the accuracy of that classifier is around 10%.

I was able to achieve a much higher accuracy using Resnet50 with accuracy=83% and 68.5% using mobilenet v2 .

Unfortunately Resnet50 classifier cannot be used in real-time application as it takes a long time to classify one frame.

1. **Methodology:**
2. **Data preprocessing:**

First we load all the dataset and give each subfolder a number to present each class “category” then we split the data into three set (training set ,validation set, testing set).

The loaded data is represented by a 3D array (“row”, ”column” , ”channels”) and to be able to use the transfer learning to extract the bottleneck features the fed image must be consist of 4D array(“image number” , ”row” , ”column” , ”channels”) also referred as 4D tensor array.

After expanding the data dimension we have to reduce the image dimension to (1,224,224,3) also to be able to fed the array to the transfer function.

The last step is to preprocess the data using “preprocess\_input” function which it’s a built in function in “Keras “.

1. **Implementation:**

Implementation will be represent in a chronological order steps:

1. Load all the required libraries for the project
2. After gathering the data from two separate dataset and organizing the data into train ,valid and testing folders ,I loaded the image paths of the data and categorize each class type
3. Implemented a random model classifier as benchmark model
4. Created a function that takes an image path as an input and produce tensor image by resizing the image to (224,224,3) and then convert it to “numpy array” ,then expand the array dimension to(1,224,224,3)
5. Extract the bottleneck features using Resnet50 model trained on “imagenet dataset”
6. Save the bottleneck feature as “numpy array” for later use ,because this process takes a lot of time.
7. Load the bottleneck feature of Resnet50 and create a classifier model that takes an input size of (1,1,2024) with a dense layer of 9 nodes with activation function “softmax” .
8. The optimizer used is “adagrad” and loss function is “categorical\_crossentropy” and metrics is ”accuracy”
9. I have trained the model many times with different epoch and batch size
10. Calculate the accuracy of the model which is = 84%
11. I implemented two function the first one takes an input “argmax of the predicted output frame” and pronounce the object name loud , and the second function is takes frames using camera and rescale the frame and converted into a 4D tensor to extract bottleneck feature and then predict the frame and pass the output to the first function.
12. The same process is made for Mobilenet v2
13. In attempt to improve the accuracy of Mobile v2 is to augment the data but unfortunately it made the accuracy of the model worsen.
14. **Refinement :**

The initial solution was using Resnet50 and it achieved a very good accuracy with 83% but unfortunately it can not be used because resnet50 takes a lot of time to process only one frame about 15 second which is a very long time ,so this model will not be able to perform in a real time applications.

So the second the solution is to use light weight model that contain much less parameters ,so I used mobilenetV2 which does exhibit a very good time to process a frame about 0.2 second but unfortunately it’s accuracy is not as good as Resnet50 model which = 69.9%.

The following table below contain the result of tuning the model to achieve a better result by changing the model optimizer and the number of batch size , and it show that the best result have been achieved by optimizer=”RMStprop” and the batch size=80 ,this result have increased my first model by 1.5% better accuracy.

|  |  |  |
| --- | --- | --- |
| **Optimizer** | **Batch size** | **Accuracy** |
| **SGD** | **10** | **64.9025069637883** |
| **SGD** | **20** | **66.57381615598885** |
| **SGD** | **40** | **66.85236768802228** |
| **SGD** | **80** | **65.18105849582173** |
| **SGD** | **120** | **67.40947075208913** |
| **SGD** | **160** | **66.29526462395543** |
| **SGD** | **200** | **66.016713091922** |
| **RMSprop** | **10** | **66.57381615598885** |
| **RMSprop** | **20** | **67.13091922005572** |
| **RMSprop** | **40** | **68.80222841225627** |
| **RMSprop** | **80** | **69.91643454038997** |
| **RMSprop** | **120** | **68.80222841225627** |
| **RMSprop** | **160** | **68.52367688022284** |
| **RMSprop** | **200** | **68.52367688022284** |
| **Adagrad** | **10** | **65.45961002785515** |
| **Adagrad** | **20** | **68.52367688022284** |
| **Adagrad** | **40** | **68.24512534818942** |
| **Adagrad** | **80** | **67.96657381615599** |
| **Adagrad** | **120** | **67.13091922005572** |
| **Adagrad** | **160** | **67.68802228412257** |
| **Adagrad** | **200** | **68.24512534818942** |
| **Adadelta** | **10** | **67.68802228412257** |
| **Adadelta** | **20** | **68.52367688022284** |
| **Adadelta** | **40** | **67.96657381615599** |
| **Adadelta** | **80** | **67.68802228412257** |
| **Adadelta** | **120** | **68.52367688022284** |
| **Adadelta** | **160** | **68.24512534818942** |
| **Adadelta** | **200** | **66.29526462395543** |
| **Adam** | **10** | **68.24512534818942** |
| **Adam** | **20** | **68.52367688022284** |
| **Adam** | **40** | **67.13091922005572** |
| **Adam** | **80** | **67.68802228412257** |
| **Adam** | **120** | **67.96657381615599** |
| **Adam** | **160** | **65.73816155988858** |
| **Adam** | **200** | **68.24512534818942** |
| **Adamax** | **10** | **68.52367688022284** |
| **Adamax** | **20** | **69.08077994428969** |
| **Adamax** | **40** | **68.52367688022284** |
| **Adamax** | **80** | **67.13091922005572** |
| **Adamax** | **120** | **68.24512534818942** |
| **Adamax** | **160** | **67.40947075208913** |
| **Adamax** | **200** | **67.68802228412257** |
| **Nadam** | **10** | **67.96657381615599** |
| **Nadam** | **20** | **66.85236768802228** |
| **Nadam** | **40** | **66.29526462395543** |
| **Nadam** | **80** | **69.63788300835654** |
| **Nadam** | **120** | **68.52367688022284** |
| **Nadam** | **160** | **68.52367688022284** |
| **Nadam** | **200** | **68.24512534818942** |

1. **Result:**
2. **Model Evaluation:**

My final model is based on using mobilenet v2 as bottleneck feature extractor and then created a convolution neural network model that contain 9 dense nodes each node represent only one type of the classified objects and the activation function is “Softmax” to determine how sure the model classify the object , and the input shape of the model is the same as the output shape of the mobilenetV2 extractor.

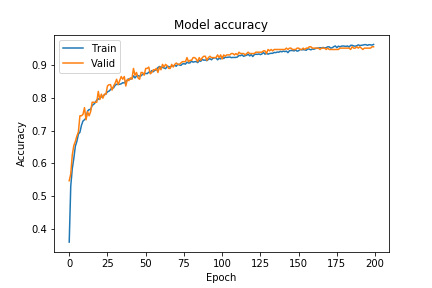
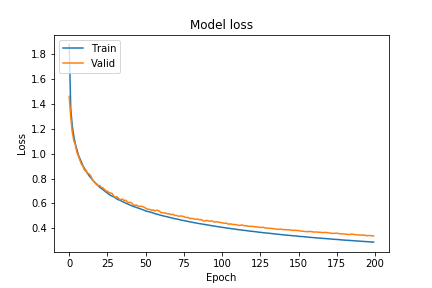
The model use “adagrad” as an optimizer , the loss function is “categorical\_crossentropy” and metric used is the “accuracy”

There is a validation dataset and contain 243 image , number of epoch =1000 and the batch size=10.

1. **Justification:**

The initial solution using Resnet50 as transfer function to extract the bottleneck feature achieved a very good result in term of accuracy but fail in the time processing ,instead Mobilenet achieved a moderate result in term of accuracy and a very good result in term of time processing ,both models actually were a great a success to the benchmark model which it achieved a very low accuracy but it sure will have the fastest processing time as classify objects randomly without any processing.

1. **Conclusion:**
2. **Free Form Visualization:**



**Fig(1) fig(2)**

Figure(1) above show that how the accuracy and the validation accuracy increase as the number of epochs increase.

Figure(2) show how is the training and validation loss decrease as the number of epochs increase.

1. **Reflection:**

Firstly we load the data and preprocess the data to be suitable for the model that we are going to use ,then train the model and then calculate the accuracy and the time taken to classify one frame.

There are many difficulties that faced me during these project ,the first one is collecting the data which I spent much time to find a suitable dataset and after I found the dataset it was too small ,so I had to search for another dataset and combine the two dataset.

I also tried to improve the accuracy of Mobilenet model by varying the batch parameter ,number of epoch ,different randomization of the data and augment the image to increase the dataset

1. **Improvement:**

The training data is concededly low in quality and quantity ,so to improve the model we need the dataset to be more accurately and much more images with high quality .