A PRELIMENERY REPORT ON

PERSONALIZED E-DÉCOR USING DEEP LEARNING AND COMPUTER VISION

SUBMITTED TO SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING

(COMPUTER ENGINEERING)

Cummins College of Engineering for Women, Pune, Department of Computer Engineering 2018-2019

SUBMITTED BY

SHUBHANGI GARNAIK **Exam No: B150204256**

TANISHA SHROTRIYA **Exam No: B150204385**

POOJA VIRANI **Exam No: B150204402**

UNDER THE GUIDANCE OF

DR. NEETA MAITRE

DEPARTMENT OF COMPUTER ENGINEERING

CUMMINS COLLEGE OF ENGINEERING FOR WOMEN, PUNE

SAVITRIBAI PHULE PUNE UNIVERSITY

2018 – 2019

CERTIFICATE

This is to certify that the project report entitled

"PERSONALIZED E-DÉCOR USING DEEP LEARNING AND COMPUTER VISION"

Submitted by

SHUBHANGI GARNAIK	**Exam No: B150204256**
TANISHA SHROTRIYA	**Exam No: B150204385**
POOJA VIRANI	**Exam No: B150204402**

has been carried out under the supervision of Dr. Neeta Maitre and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering) 2018-19.

(Dr. Neeta Maitre)
Guide

Department of Computer Engineering

(Dr. Supriya Kelkar)
Head

Department of Computer Engineering

(Dr. M. B. Khambete)
Director,

Place: Pune
Date:

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the project report on **"PERSONALIZED E-DÉCOR USING DEEP LEARNING AND COMPUTER VISION"**

We would like to take this opportunity to thank our internal guide Dr. Neeta Maitre for providing her inputs and guidance at every stage of the development of this report. We are grateful to her for the support she has provided throughout.

We are also grateful to Dr. Mrs. Supriya Kelkar, Head of Computer Engineering Department, Cummins College of Engineering for Women, Pune for her support and guidance.

For the timely updates and help with technical issues regarding the guidelines to be followed for completion of the project, we thank the project committee members of Cummins College of Engineering for Women, Pune.

Shubhangi Garnaik

Tanisha Shrotriya

Pooja Virani

ABSTRACT

Today's fast paced world leaves very less time for shopping for basic amenities. Hence, a whole new shopping platform has established itself online to allow these users, to shop in the comfort of their homes, or while travelling between places. As of 2017, India's furniture market is the fifth largest in the world, making India one of the largest exporters of wooden furniture globally. Yet, out of the total furniture market, about 85 – 90 % is unorganized. This $17 billion unorganized market, has commenced to move online, making the rest of the branded furniture owners feel the need to set up their businesses on online marketplaces like Amazon.

However, there are still a lot of challenges that this industry faces, as compared to other types of items bought online. For example, there is a growing need for the local touch and feel. There continues to be a low trust rate amongst online buyers, when it comes to furniture shopping, as usually these tend to have a higher budget, and so good delivery is expected. Another issue with online web applications is a high rate of return requests, caused by a dislike towards the actual product compared to the picture which is uploaded. This dislike may also be due to factors like, low quality of the product; or the product not fitting in with the room lighting, existing furniture items or simply not fitting into the space. Thus, the developed system will try to address some of these issues faced by many online shoppers.

The developed system saves the time invested by the user while searching for items on a shopping website, by giving recommendations for the item based on the style of the user's room. It provides user more control over the search space at the same time gives assistance. In this way it will increase the number of online shoppers for furniture items.

CONTENTS

LIST OF ABBREVIATIONS

Abbreviation	Meaning
SRS	System Requirements Specification
DFD	Data Flow Diagram
UML	Unified Modeling Language
Bi-LSTM	Bidirectional Long Short Term Memory
GUI	Graphical User Interface
CNN	Covolutional Neural Networks
GPU	Graphics/General Processing Unit
SSL	Secure Socket Layer
SDLC	Software Development Life Cycle

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1. OVERVIEW

The developed system serves the purpose of giving a recommendation, based on various colors seen in the room of the user. The intended audience for this application is anyone who may be interested in buying furniture online. The application works for a furnished and an unfurnished room.

In the gaming industry, efforts have been made in the application of computer vision to generate 3D visualizations, which is slowly moving towards the retail sector. Research papers which focus on building tools for interior decorators have been studied and further elaborated in the literature survey of this project report.

1.2. MOTIVATION

Today, the online furniture shopping industry faces quite a few problems. To understand the demographics of furniture buyers amongst our network of friends, family and colleagues, a small local survey was conducted.

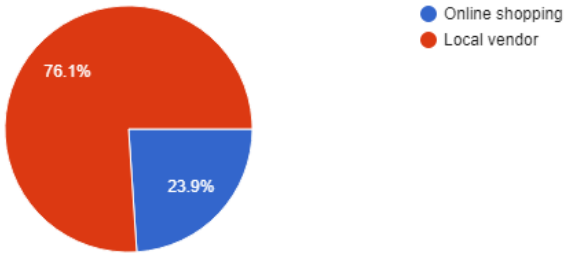

Figure 1.1 Local vendors vs online shopping preference

The questions asked their preference between local vendors and online sites; the time invested by them, in case of a trip to a local vendor; and the frequency with which they refurnished their home/room. Around 80% responders preferred shopping at a local vendor. In the opinion of 62% responders each trip required at least 2 to 3 hours for a satisfied outcome, while 11% of the responses suggested 5 or more hours invested at a local vendor. It was also a general opinion that more than one trip may be required in case a trip is aimed at refurnishing an entire room.

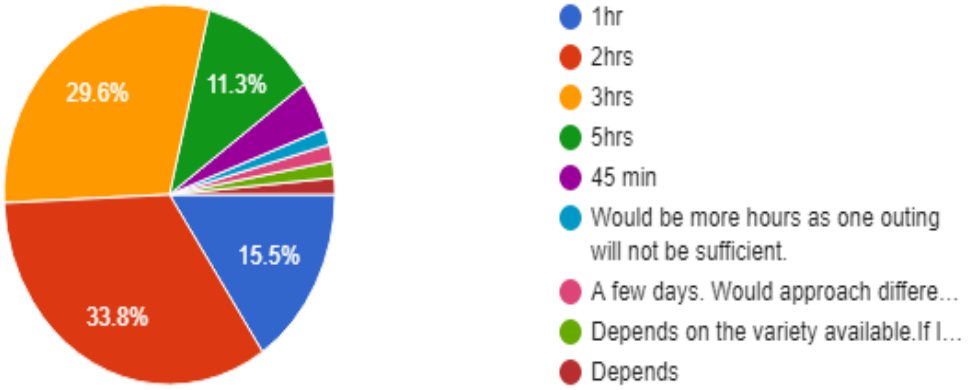

Figure 1.2 Shopping time invested locally vs online

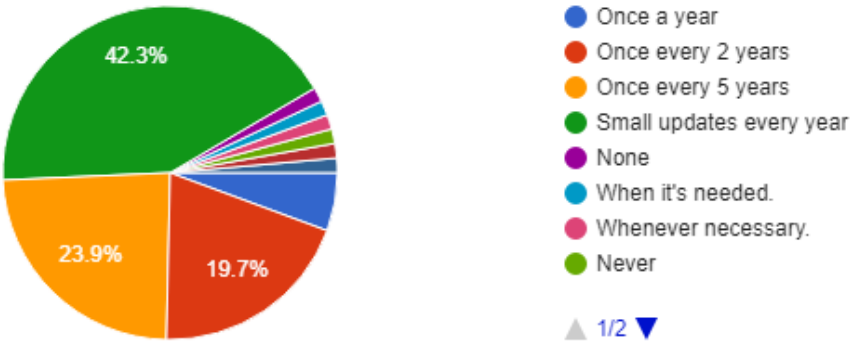

Figure 1.3 Frequency of Refurnishing

As for the frequency of refurnishing, 50% of responses ranged between small updates each year, to whenever necessary; the other 50% being updates made once every 2 to 5 years. This survey shows us that most of these people make small changes from time to time but must invest a lot of time even for trivial updates, when shopping at a local vendor. Other factors such as cost of travel, process of selecting between two vendors and finally ensuring that the furniture they buy does indeed look good with the current furniture in their house, may add more time to their shopping trip. The results of our survey show that the process of buying furniture is truly cumbersome, to say the least.

Age Group	People in same home?	Preferred Method	Time Invested at Local Vendors	How often do you refurnis	Comfort to share image of room online
50-60	5	Online shopping	3hrs	Once every 2 years	Yes
20-30	4	Local vendor	3hrs	Once every 2 years	Maybe
20-30	4	Local vendor	2hrs	Once every 2 years	Maybe
20-30	3	Local vendor	5hrs	None	Maybe
20-30	3	Online shopping	3hrs	Small updates every year	No
20-30	4	Local vendor	3hrs	Once every 2 years	Yes
30-40	1	Local vendor	2hrs	Once every 2 years	Maybe
20-30	6	Local vendor	3hrs	Once every 5 years	Maybe
20-30	4	Online shopping	2hrs	Once every 2 years	Yes
20-30	5	Online shopping	1hr	Small updates every year	Yes
40-50	4	Online shopping	Would be more hours as one outin	Once every 2 years	Yes
20-30	3	Online shopping	3hrs	Small updates every year	No
20-30	4	Local vendor	3hrs	Once every 5 years	Maybe
20-30	3	Local vendor	3hrs	When it's needed.	Maybe
20-30	3	Local vendor	2hrs	Once every 5 years	Yes
60+	5	Online shopping	1hr	Small updates every year	Maybe
Below 20	4	Local vendor	2hrs	Whenever necessary.	Maybe
20-30	3	Local vendor	3hrs	Small updates every year	Yes
20-30	3	Local vendor	2hrs	Small updates every year	Maybe
20-30	4	Local vendor	3hrs	Small updates every year	Maybe
20-30	4	Online shopping	1hr	Small updates every year	No
20-30	4	Local vendor	2hrs	Small updates every year	No
20-30	4	Local vendor	1hr	Once every 5 years	Yes
20-30	8	Local vendor	3hrs	Once every 5 years	Yes

Table 1.1 Sample from Local Survey

1.3. PROBLEM DEFINITION AND OBECTIVES

The personalized e-décor is an online application dedicated to furniture shopping. The user registers on the website and is sent a link to confirm email. After successful confirmation, the user may login to the website. After logging in, he/she is asked to upload a picture of the room for which the user wishes to shop furniture.

The system then extracts the colours in the room and gives a probability of which colour is found in higher proportion. Based on the colour seen in maximum proportion, the system provides the user with a range of furniture items which match the style. The user may also disable some of these recommendations.

1.4. PROJECT SCOPE

The purpose of this application is to reduce, the long hours invested for furniture shopping and, the number of return requests due to poor customer satisfaction. So, this application helps individuals to browse through a wide range of choices from the comfort of their home.

The intended audience for this application is anyone who may be interested in buying furniture online. It starts by the user logging in with his/her credentials and uploading an image of a room. The current scope will allow the picture upload from the device of the user. Other sources such as a cloud storage or through a real-time camera image will not be allowed. Currently, the database consists of bed, table and chair category.

1.5. LIMITATIONS

The developed system suffers from the following limitations. It is assumed that the house is pre-painted; in case of a white-washed house, furniture recommendations will also be according to the colour of white-washed walls. Lighting of the picture uploaded by the user may result in issues with the furniture selected at the time of purchase. For example, a darker picture, taken in the evening may result in the algorithm recognizing a different shade of the actual wall, and the furniture recommended based on this may not look exactly as shown by the software.

The current application does not show virtual representations in the image of the room. The virtual creator function will recommend arrangements for the new piece of furniture based on the current positioning of other furniture in the room. If these items can be shifted to make more space, then those arrangements will not be considered. In other words, the already present items in the image cannot be moved around along with the new furniture item.

1.6. METHODOLOGIES OF PROBLEM SOLVING

The problem of image recognition and categorization can be solved in the following ways:

1. Local feature descriptors: Algorithms like Scale Invariant Feature Transform are used to identify features in an image. These cannot be used for large scale datasets.

2. Visual Words/ Bag of Words/ Machine Learning: By performing K-means on SIFT descriptors a visual dictionary can be obtained. A histogram obtained from the visual words is a robust way to represent images. But these methods need hand crafted features and cannot be trained.

3. Convolutional Neural Network models: Features are not hand-crafted like in the previous two techniques. Instead a layer of neurons is used a featured to perform

convolution with the image and then a classifier like Support Vector Machine is used. CNNs are trainable which makes then highly adaptive.

The problem of object detection can be solved in the following ways:

1. Fast RCNN:

In Fast RCNN input images are fed to the CNN to generate a convolutional feature map. From the convolutional feature map, the region of proposals is identified and warped into squares and by using a RoI pooling layer they are reshaped into a fixed size so that it can be fed into a fully connected layer. From the RoI feature vector, a softmax layer is used to predict the class of the proposed region and the offset values for the bounding box.

2. SSD: Single Shot MultiBox Detector in TensorFlow

SSD is an object detection unified framework with a single network. The SSD contains a TensorFlow re-implementation of the original Caffe code. It implements VGG-based SSD networks (with 300 and 512 inputs). The architecture is modular and makes the implementation easy.

CHAPTER 2

LITERATURE SURVEY

The following papers were a source of inspiration and ideas for the methods adapted in the project design.

1) Ivona Tautkute 1, 3 et al. ***"What Looks Good with my Sofa: Multimodal Search Engine for Interior Design".*** **In proceedings of the 2017 Federated Conference on Computer Science and Information Systems.** This paper is analogous to the objectives of Personalized eDecor. It proposes a multi-modal search engine for interior design that combines visual and textual queries. The engine retrieves interior objects, e.g. furniture or wall clocks, that share visual and aesthetic similarities with the query. It allows the user to take a photo of a room and retrieve with a high recall a list of items identical or visually like those present in the photo.

2) Reza Fuad Rachmadi ∗ and I Ketut Eddy Purnama, ***"Vehicle Color Recognition using Convolutional Neural Network".*** **Latest version 15 August 2018.** This paper presents a vehicle colour recognition method using convolutional neural network (CNN). It proves that CNN can also learn classification based on colour distribution and is not restricted to shape-based classification task. This paper is an inspiration for building a classifier that detects colours in images of a room.

3) Yangbangyan Jiang, Qianqian Xa, Xiaochün Cao, Qingming Huang
"Who to Ask: An Intelligent Fashion Consultant?". **International conference on Multimedia Retrieval 2018.** In this paper, researchers have proposed an end-to-end outfit recommendation system called Stile. The system uses machine learning to provide an intelligent fashion consultant that answers the question- what to wear by showing the outfits that are most compatible with each other based on current trends in fashion. This paper is useful for future scope of Personalized eDecor.

4)Andrew G. Howard, Menglong Zhu, Bo Chen et al. ***"MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications".*** **Publication: April 2017.** MobileNets are a class of efficient models for mobile and embedded vision applications. MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks. This paper demonstrates the effectiveness of MobileNets across a wide range of applications and use cases including object detection, fine grain classification, face attributes and large-scale geo-localization.

5) Jason Yosinski, 1 Jeff Clune, 2 Yoshua Bengio, 3 and Hod Lipson 4. ***"How transferable are features in deep neural networks?".*** **In Advances in Neural Information Processing Systems (NIPS 2014).** Many deep neural networks trained on natural images exhibit a common phenomenon: on the first layer they learn features like Gabor filters and colour blobs. Such first-layer features appear not to be specific to a dataset or task, but they are applicable to many datasets and tasks. This paper experimentally quantifies the generality versus specificity of neurons in each layer of a deep convolutional neural network and report the results. This paper is used to use a more optimal approach for classification of styles in Personalized eDecor.

CHAPTER 3

SYSTEMS REQUIREMENTS SPECIFICATION

3.1. ASSUMPTIONS

3.1.1. Browser compatibility

The system developed is compatible with all web-browsers and to the most recent versions of the browser. Browser compatibility is important to keep the look and feel of the system similar in all the browsers.

3.1.2. User profile and picture upload.

The user should have created a profile, to gain access to a personalized recommendation. He/she should be comfortable uploading a picture of a room to use the virtual creator.

3.1.3. Assuming user does not have a budget

At this stage it is assumed that the user does not have a budget, so the system does not filter the items based on a budget range.

3.2. DEPENDENCIES

3.2.1. Good data set for user to select their pick

A large collection of data items to select from, allows to test the efficiency of our algorithm, and make it a more realistic application. It is necessary to give a user a large variety of options to choose from to ensure user satisfaction.

3.2.2. Browsers compatibility issues

Many a times due to browser compatibility issues there are cases of distorted images or no picture at all. Also, for the system to work efficiently, a good internet connectivity is required.

3.2.3. Compulsory user login for picture upload

For security reasons, it is expected that the user logs into the system before using the software. Also, it provides user with a hassle-free usage of the system, (uploading an image every time, the same room is to be furnished). An image uploaded by the user will be saved after he/she logs out, and reloaded once he/she logs in.

3.3. FUNCTIONAL REQUIREMENTS

3.3.1. Login Authentication

Authentication is the process of recognizing a user's identity. It is the mechanism of associating an incoming request with a set of identifying credentials. In the developed system the credentials provided will be compared to those on file in a database. The login details of a user should be authenticated to ensure security of the user.

3.3.2. Recommendation Engine

This engine will be designed to recommend a top range of items based on the styles obtained from the uploaded room image. The application will also show items which fit a few of the styles.

3.3.3. Image Data

A facility is necessary to store data used for training the model.

3.3.4. Model Re-train

It facilitates to retrain the recommendation model, overtime as image data changes. It also helps in providing an efficient and up to date configuration for making recommendations for the user.

3.3.5. Object Detection

Image processing is a method to convert an image into digital form and perform some operations on it. It is used to obtain an enhanced image or to extract some useful information from it. It can used be used to solve the problem of object detection in computer vision.

3.3.6. Legal or regulatory requirements

Legal or regulatory requirements will deal with the policies related to using the data available online. For example, the model will be trained on images scraped from various interior décor web applications, google etc. During deployment of the application, it must be ensured that everything being used is falling within Indian laws.

3.4. EXTERNAL INTERFACE REQUIREMENTS

3.4.1. User Interface Requirements

The navigation bar has a home, login and a create an account button. After the user registers successfully, he/she gets a verification email. The profile page of the user has an option to upload his room image with dimensions of the room, see all the images he has uploaded, view his shopping cart and an option to shop. On clicking the shop button, the user is shown a list of furniture items. On the right side, will be a toggle button to add and remove the filters. The top right corner has a button which shows the current status of items in the cart.

On clicking any of the picture, user will have an option to update the quantity of the items to be bought and soon add it to the cart. Further the user can continue shopping or opt to checkout. The upload option for image of room, will only allow the user to upload images from the device. Other sources such as camera, a picture from cloud storage and so on are currently not covered in the scope.

3.4.2. Hardware Interface Requirements

This project doesn't require any specific hardware interfaces like a microprocessor or controller, apart from the one already existing in the device through which the user is accessing the web application.

3.4.3. Software Interface Requirements

This project will consist of a CNN Classifier which is not part of the GUI. A trained model will be loaded in the GUI directly.

3.4.3.1. Convolutional Neural Network Classifier

The classifier is trained on images from a dataset created using images taken from the web, using web scrapping. The CNN forms weights for each class as per these images. Classes are currently divided based on colors but can be extended to the styles considered in interior decoration, like retro, vintage etc, or colours. The image uploaded by the user will also play a role in finding the items that match the current furniture type in the room. Using the weights for each class and the requirements stated by a new user, the model will scan through the available images and classify them into the top most choices which fit the extracted style, the medium range choices and the lower ranged choices.

3.5. NON-FUNCTIONAL REQUIREMENTS

3.5.1. Performance Requirements

Dynamically Changing Web Page

For this application a customer first uploads a room image. This means every time a new image is uploaded filter, the recommendation list is updated. Thus, this system requires dynamic webpages to show different recommended items.

Low Response Time

Low response time issue is handled using caching and optimization of images on client side. When a user uploads a picture of his room it is stored in the cache memory. Then this picture is used repeatedly. This reduces the response time for the user, and he/she is not affected by the other users who are using the website at the same time.

Large volume of un-optimized images is one of the reasons that lead to slow web applications. High resolution images also consume large amounts of bandwidth. Uploading

larger sized images and then scaling them down can unnecessarily increases the size of the web page causing the web application to load slowly.

So, first the size of the image is obtained using waterfall tests and then optimized on client side before uploading to the server to avoid low response time. The image format is another important factor to consider. For example, JPEG images are much smaller in size compared to other image formats like PNG, which gives better performance and leads to lower bandwidth cost.

3.5.2. Security

Multiple Sessions by Single User

If the user is trying to log in again from another browser, then it is a potential security risk. Three scenarios may arise in this case. In the first case, an attacker somehow steals a user's credentials while the user authenticates the application. The attacker immediately tries to log in with the stolen credentials. The application sees that the user is already logged in and returns an error to the attacker, thus temporarily protecting the account.

Second, an attacker shoulder surfs to obtain a valid username. He then immediately proceeds to run a password guessing or dictionary attack hoping to determine the password. If his attack happens to be successful during the time the legitimate user is logged in, it would prevent the attacker from gaining access.

Third, an attacker and a victim log in such that their sessions overlap. The application displays an error message that alerts the victim that someone else is using his account. The victim may contact the site owner, spurring an investigation, which might uncover a compromised account.

Securing User Information Safely

Any information submitted to a web application is sent as plain text and can be viewed if an attacker is able to intercept the information. SSL encrypts the information that is shared between the browser and the web server, so that even if a user's information is intercepted, it remains hidden. Thus, to obtain optimal security, search performance, and credibility with users, SSL and HTTP will be used.

Room Image Operations

Suppose, a user uploads an image of the room and obtains the corresponding recommendation. Now, user wants to try it for other rooms as well. For this he would delete the previous image and expect it to be deleted from every other storage. All the user's information must be secured and without user's consent, it must not be saved or forwarded anywhere else.

3.5.3. Safety

Safety as a non-functional requirement deals with steps taken to ensure safety of user's privacy rights and no hazardous side effects to the user. There shouldn't be any data leakage from the web application about the buyers of the furniture to the sellers. For example, a user may sign-in with their email id, which if leaked out to a vendor, may result in the user receiving unwanted spam from the vendor. During the creation of account, a two-step verification process can be included by using two methods of communication for receiving the user's agreement. This will help to ensure that the account of the user isn't already compromised in the initial stages of account creation. The application cannot be held liable once both forms of communication have been approved from the user side.

3.5.4. Software Quality Assurance

Cart Operations

In any e-commerce site, the first step in buying a product is adding it to the cart. A clear feedback of the item being successfully added to the cart allows the user to confidently continue shopping and gives him a sense of control. This feedback reassures the user that he hasn't added a wrong item to the cart. To ensure that the site processes the Add-to-Cart action and the correct product is added, a clear badge superimposed on the cart icon. It will indicate the total number of items in the cart. This assures the user that their product has been added to the cart successfully.

Proper Bootstrapping

Bootstrap provides one of the best responsive, mobile first grid systems. It helps to scale a single web application design from smallest mobile device to high definition displays. It also allows reuse of CSS and JavaScript components and is compatible with all the major browsers and platforms available.

Bootstrap offers several ways to customize the default design. All the default CSS and JS behaviour can be overridden. Thus, flexibility will be provided to add updates in the future versions of the system based on user feedback.

3.5.5. Performance Evaluation

To evaluate the actual functioning of the application on all browsers and devices, it is even more important to ensure that it does not crash under heavy loads. Load testing tools may be used to evaluate the performance of the application under varying loads on the server effectively. Load testing will be performed after every iteration in the SDLC cycle, alongside other forms of testing, to avoid errors being carried forward.

3.6. SYSTEM REQUIREMENTS

3.6.1. Database Requirements

MySQL to store images and their characteristics.

3.6.2. Software and Hardware Requirements

- 64-bit Operating System (Windows, Linux (any distribution) or MacOS)
- Python 3.6 or 3.7, Keras, Tensor flow, OpenCV
- Memory requirements- 4GB-8GB RAM
- GPU may be required for model training.
- Django version 2.1.5

CHAPTER 4

SYSTEM DESIGN

4.1. SYSTEM ARCHITECTURE

An architectural diagram gives an idea of the underlying architecture of the entire system.

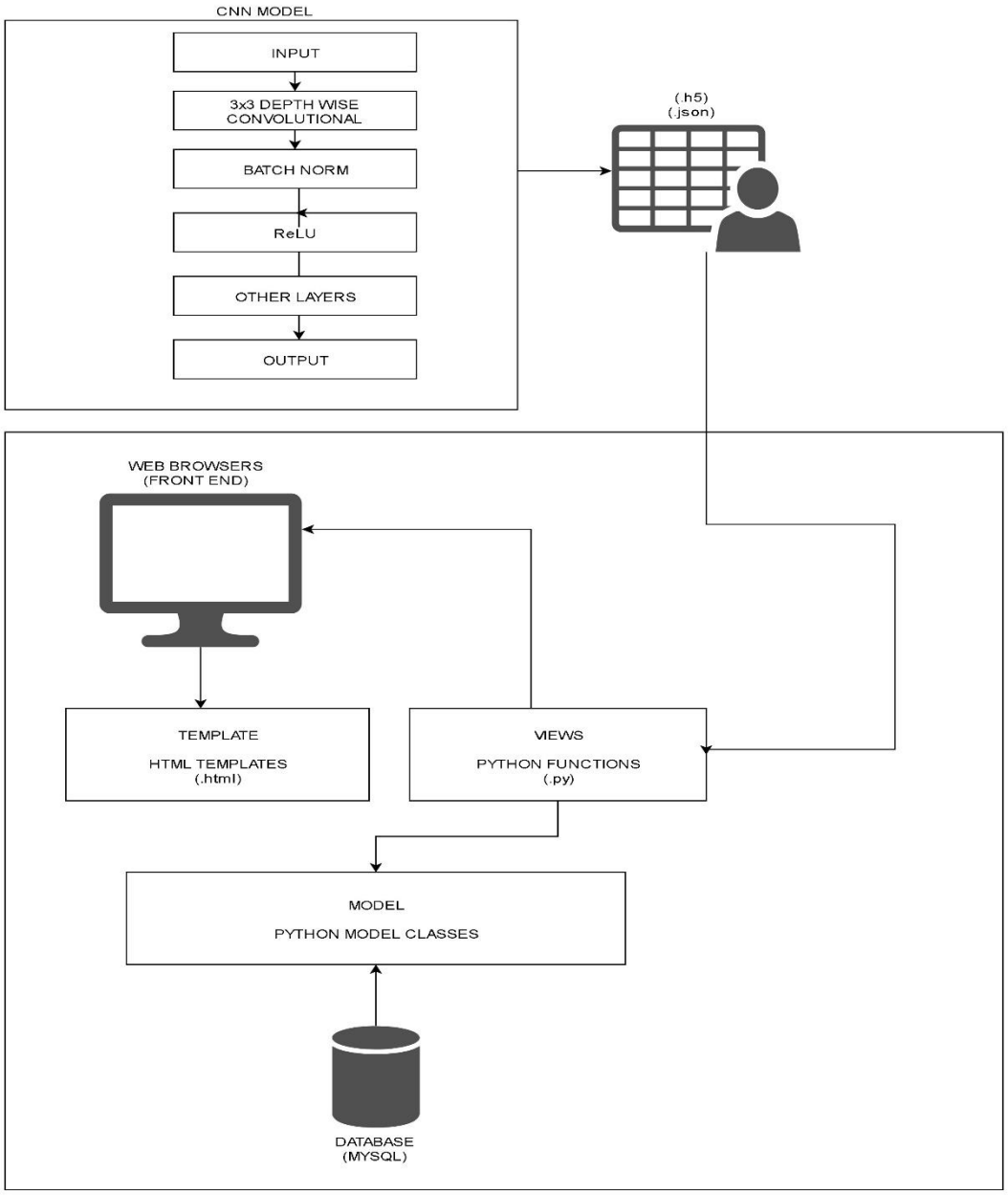

Figure 4.1 Architecture Diagram

4.2. MATHEMATICAL MODEL

Algorithm 1: Object Detection

System description:

- Let S be the system,

$S=\{s,e,X,Y,success,failure,DD,NDD,CPU_{corecnt}\}$

X=Set of Inputs

Y=Set of outputs

Fc=Set of functions

Fc ={Fme}

Fme=Kernel / Classifier

S1=initial/start state

E1=end/stop state

success=set of outputs when function is successful

failure=set of outputs when function fails

DD=deterministic data

NDD= non-deterministic data

$CPU_{corecnt}$ =CPU core count

Where,

- Input: X= {Image of customer's room}

- S1= {Empty Set}
- E1= {List of Detected Objects}

Where, Objects are furniture items in room uploaded by each user

- Output: Y= {Y1, Y2, Y3}
    - Y1 € Success
        - Y1=All objects successfully detected
    - Y2, Y3 € Failure
        - Y2=system fails to detect the objects
        - Y3=complete object is not detected
- Function Fme= {Fk1, Fk2}
    - Fk1= loadModel()
    - Fk2= detect()
- DD= {labels of objects}
- NDD= {Image of room}

Algorithm 2: Style Classification

System description:

- Let S be the system,

S={s,e,X,Y,success,failure,DD,NDD,$CPU_{corecnt}$}

X=Set of Inputs

Y=Set of outputs

Fc=Set of functions

Fc ={Fme,Fr}

Fme=Kernel / Classifier

Fr= Friend for image processing

S1=initial/start state

E1=end/stop state

success=set of outputs when function is successful

failure=set of outputs when function fails

DD=deterministic data

NDD= non-deterministic data

$CPU_{corecnt}$ = CPU core count

Where,

- Input: X= {Images of different styles}
- S1= {random weights assigned to network}
- E1= {trained model with weights assigned as per data}

Where, Objects are furniture items in room uploaded by each user

- Output: Y= {Y1, Y2, Y3, Y4}
    - Y1 € Success
        - Y1=Image classified accurately into style
    - Y2, Y3, Y4 € Failure
        - Y2=system fails to classify the image
        - Y3=image specification is incorrect
        - Y4=mis-classification
- Function Fme= {Fk1, Fk2, Fk3}
    - Fk1=train()
    - Fk2=test()
    - Fk3=classify()

- Function Fr= {Fk1,Fk2,Fk3}
    - Fk1=feature_extraction/convolution()
    - Fk2=dimensionality reduction/pooling()
    - Fk3=2D-1D/flattening()
- DD = {training set of images, style categories}
- NDD = {batch input of detected furniture items}

4.3. DATA FLOW DIAGRAM

A Data Flow Diagram shows us the flow of data between various modules.

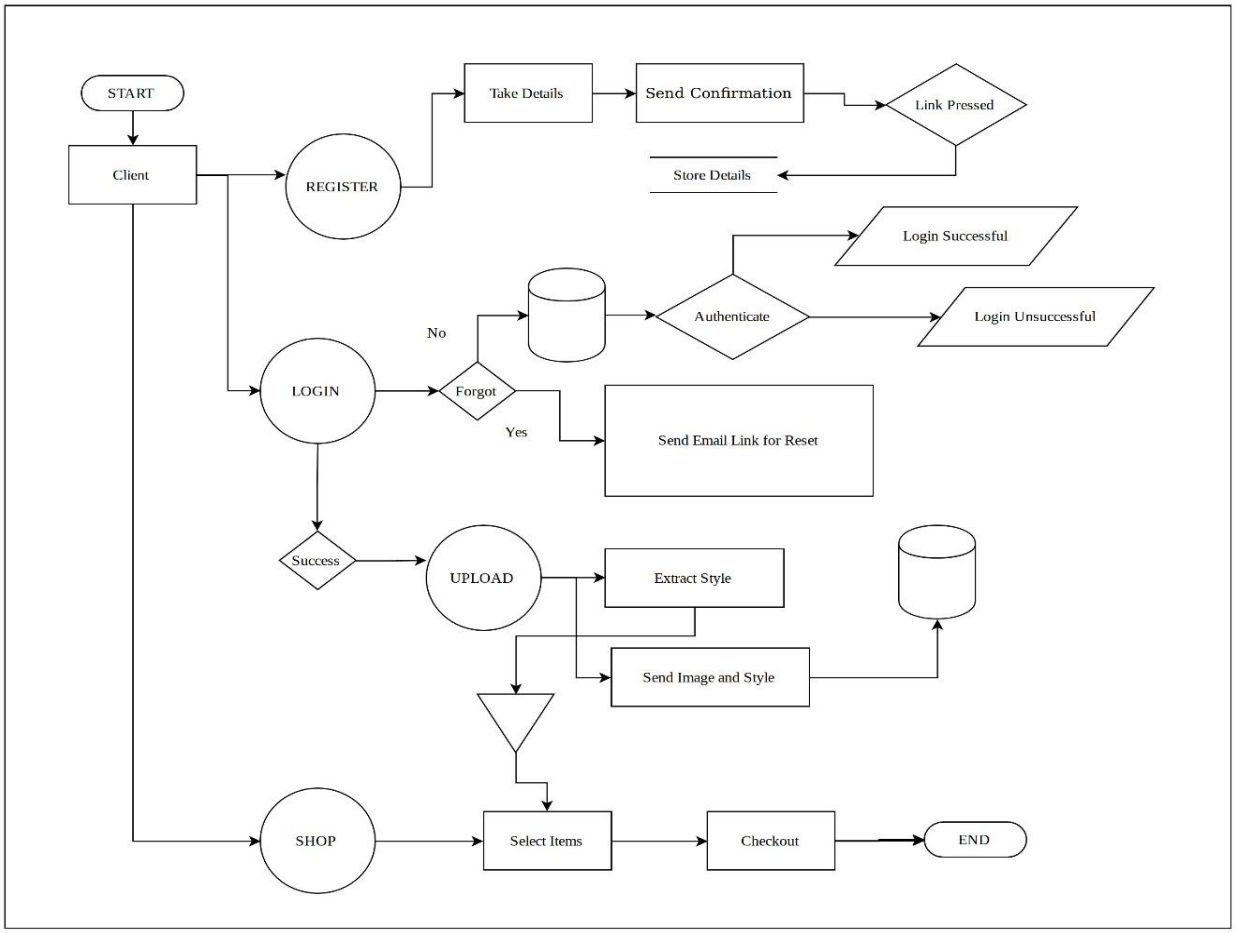

Figure 4.2 Data Flow Diagram

4.4. ENTITY RELATIONSHIP DIAGRAM

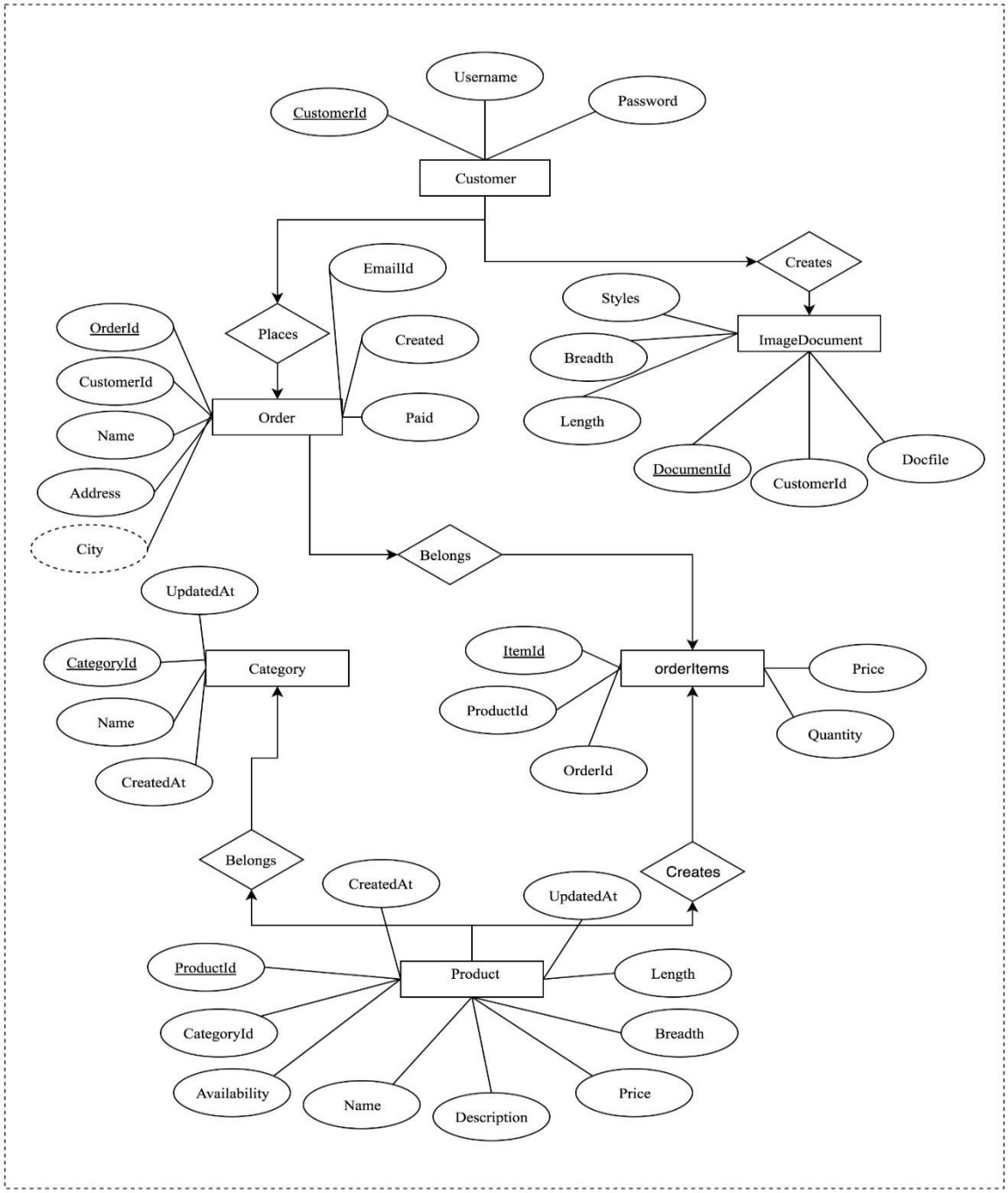

Figure 4.3 Entity Relationship Diagram

4.5. UML DIAGRAMS

4.5.1. USE CASES

Table 4.1 List of Use Cases

Sr. No.	Use Case	Description	Actors	Assumptions
1	Client Register	Create a new client account.	Human Client	Client doesn't already have an account.
2	Upload Image	Upload image of the room intended for the furniture	Human client	Format is jpeg.
3	Classify	Classify the items available for sale as per the room style.	System	Model is trained for classification.
4	Select Items	Select preferred items from the recommended list.	Human Client	A large database of sale items is available.
5	Cart	Add or delete items from cart	Human Client	Works independent of other modules.

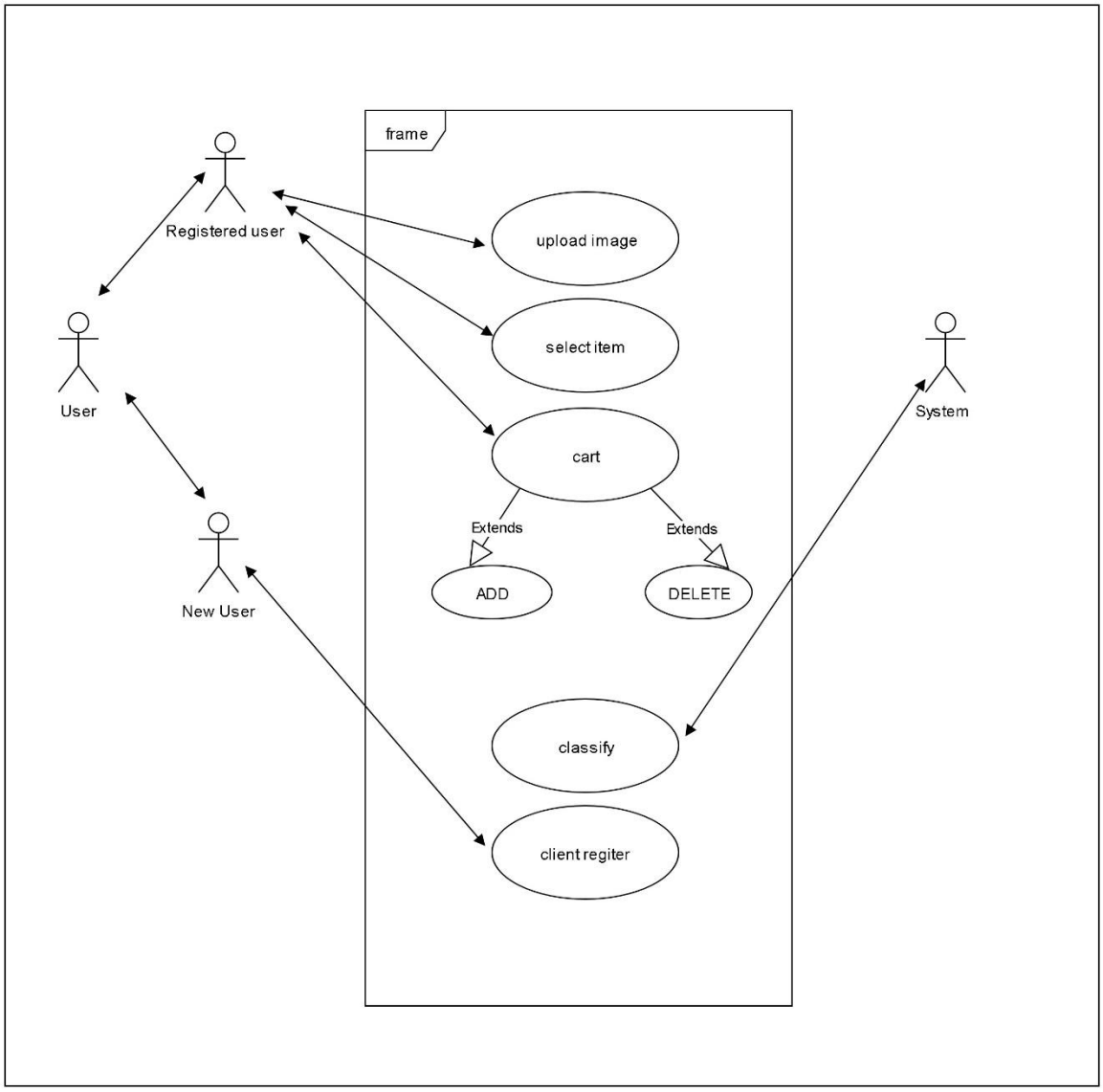

Figure 4.3 Use Case Diagram

4.5.2. SEQUENCE DIAGRAM

A sequence diagram helps to understand the complete sequential execution of the various modules of the project.

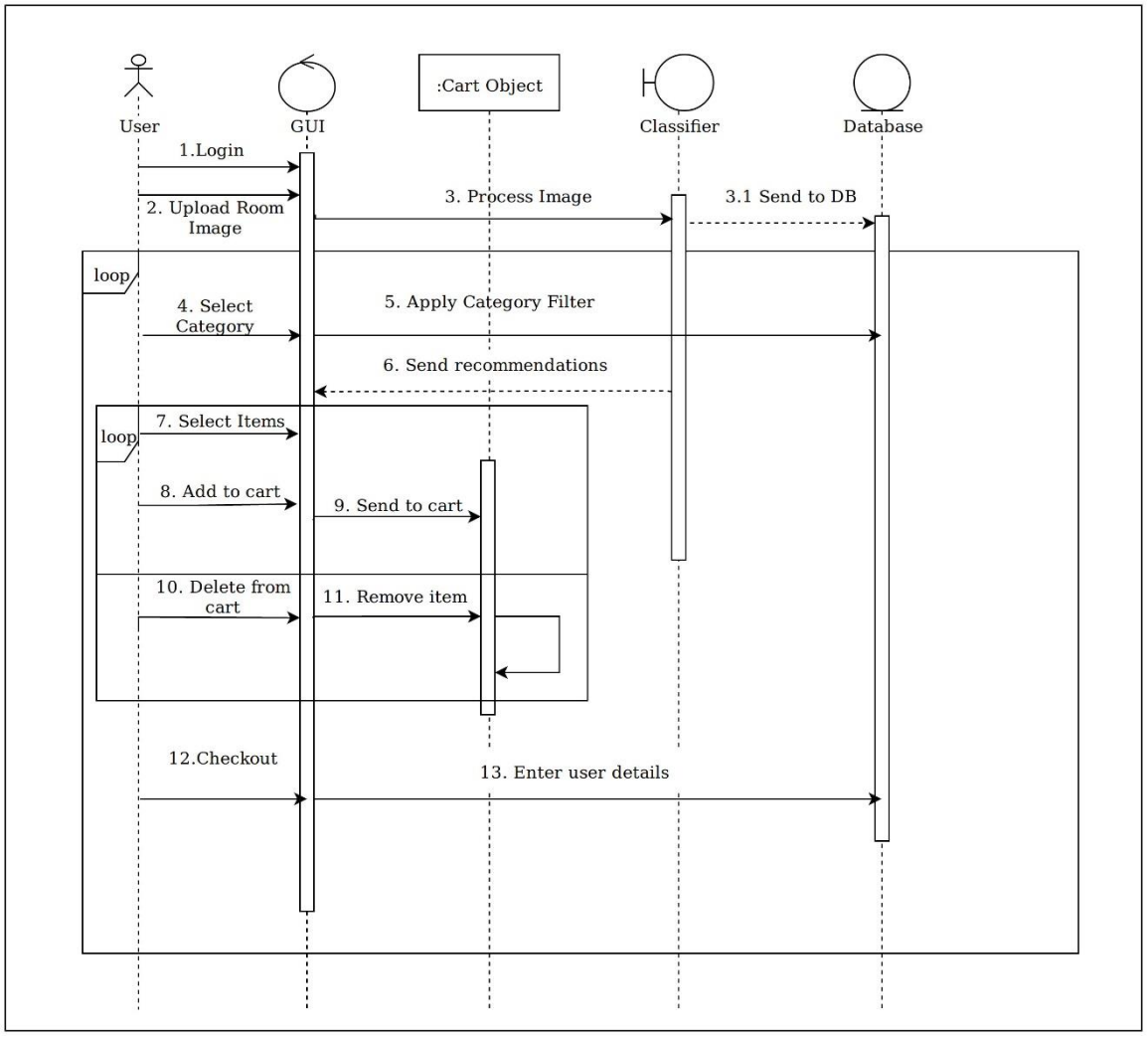

Figure 4.4 Sequence Diagram for User Operations

4.5.3. DEPLOYMENT DIAGRAM

This diagram shows the hardware and communication interactions which will happen amongst the user, software front end and backend.

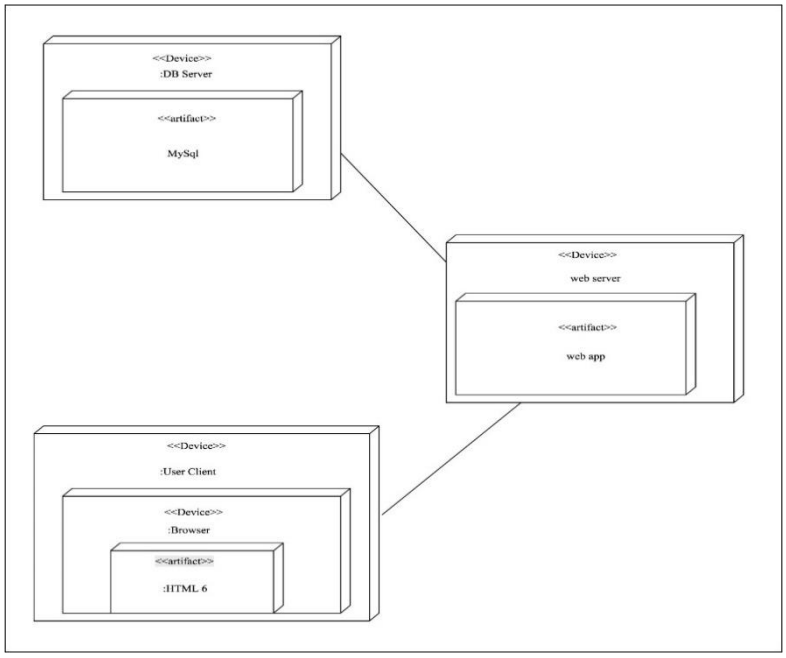

Figure 4.5 Deployment Diagram

CHAPTER 5

PROJECT PLAN

5.1. PROJECT ESTIMATE

The "Agile model" was used for system development. Agile methodology is a practice that promotes continuous iteration of development and testing throughout the software development lifecycle of the project. Both development and testing activities are concurrent.

5.1.1. RECONCILED ESTIMATES

The reconciled estimates are as follows

Cost estimates - We estimate a cost of Rs.500 for domain name, to complete our project.

Time estimates - To complete the entire project including planning, designing, implementation and testing, we would approximately require 10-11 months.

5.1.2. PROJECT RESOURCES

1. Hardware: Laptops
2. Software: Operating system, Virtual environment setup
3. Tools: Spyder notebook, sublime, deep learning python libraries, Django

5.2. RISK MANAGEMENT

Risk management deals with identification and handling of all the risks associated with the project.

5.2.1. RISK IDENTIFICATION

The problem of furniture recommendation based on room styles is NP hard because of overlap of different styles and availability of large set of colours. The domain of the problem has been reduced to include fixed set of styles and colours. A model has been designed for mobile devices to increase efficiency thus making it NP complete. Project risks and the approaches to manage them are discussed below.

- The risks which can harm successful deployment of the project have been identified through SRS document as well as test case validation plan.
- The risks can arise mainly through test cases generated in case of any function failure.
- Risk mitigation strategies need to be implemented as well as any hidden risks need to be identified during testing phase.

Risks are as follows: -

- Device Failure: Quick response is expected on client side of the web application. In case of system overload, system failure may occur.
- Incorrect Image Format: Uploaded room image format is not of required specification.
- Model Size and Computation: Model should be of optimal size to fit into buffer and perform efficient computation on client side.
- User Data: Room image of user may be susceptible to attacks.

5.2.2. RISK ANALYSIS

The following tables shows the analysis for each identified risk and the overall impact it will have on the system.

Id	Risk Description	Probability	Impact on Quality	Overall Risk

1.	Device Failure	Low	High	Medium
2.	Incorrect Image Format	Medium	Medium	Medium
3.	Model Size and Computation	Medium	High	Medium
4.	User Data	Low	Medium	Low

5.2.3. OVERVIEW OF RISK MITIGATION, MONITORING, MANAGEMENT

- Device Failure: Proper deployment and server-side management of web traffic.
- Incorrect Image Format: Handling multiple data formats like jpeg, .png and so on.
- Model Size and Computation: Using optimized version of model. For example, for object detection task, instead of using Faster RCNN model, which is computationally intensive, MobileNet SSD model was used, which is faster.
- User Data: Not storing user's images on the server side, instead storing it onto their google drive and accessing it using google drive API.
- Security: Providing good encryption mechanisms.

5.3. PROJECT SCHEDULE

The schedule for this project is described below.

5.3.1. PROJECT TASK SET

Major Tasks in the project stages are:

- Task 1: Front End
    - Task 1.1: Register new user [Ref: Table 4.1]
    - Task 1.2: Login existing user [Ref: 3.3.1]
    - Task 1.3: Upload Image [Ref: Table 4.1]
    - Task 1.4: Cart Operations [Ref: Table 4.1]
- Task 2: Recommendation Engine
    - Task 2.1: Training the system – provide the system with sufficient image data sets to efficiently train the neural networks.

- Task 2.2: Applying CNN (Convolutional neural network) – Using Python CNN libraries to classify the images appropriately.
- Task 2.3: Testing – Providing all sorts of test cases including edge test cases to test the system appropriately.

5.3.2. TASK NETWORK

The diagram shows the set of tasks implemented by the system.

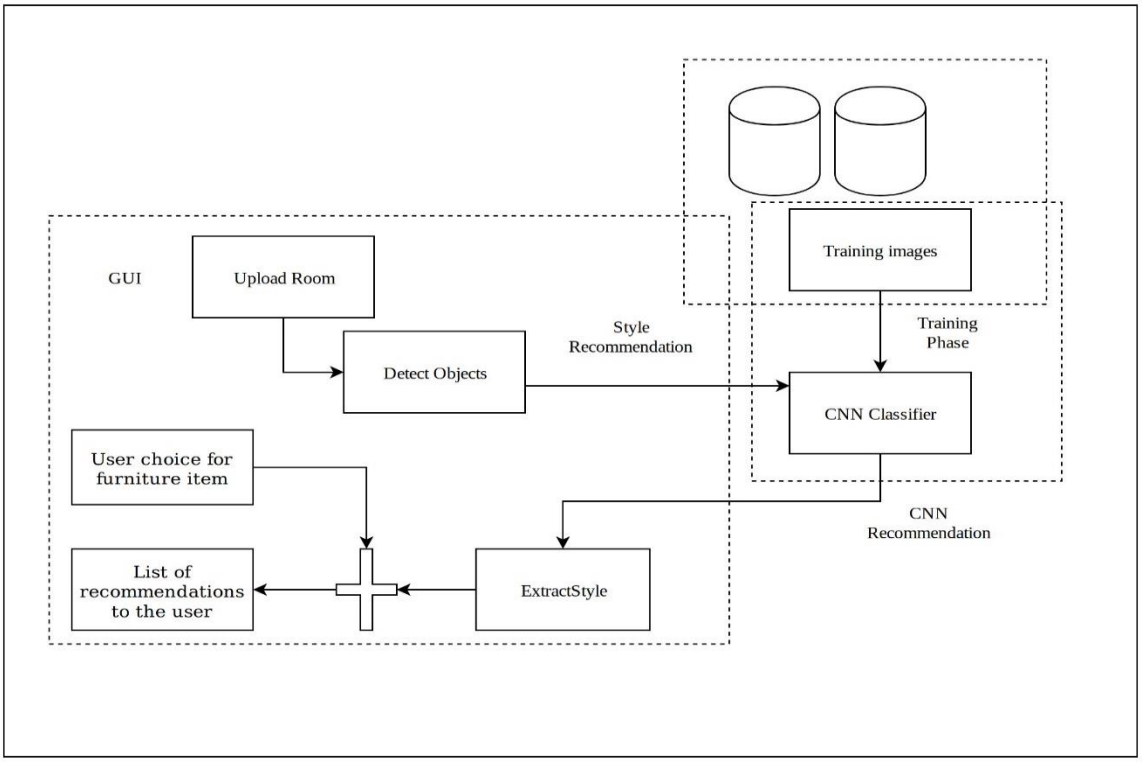

Figure 5.6 Task Network of System

5.3.3. SOFTWARE DEVELOPMENT LIFE CYCLE

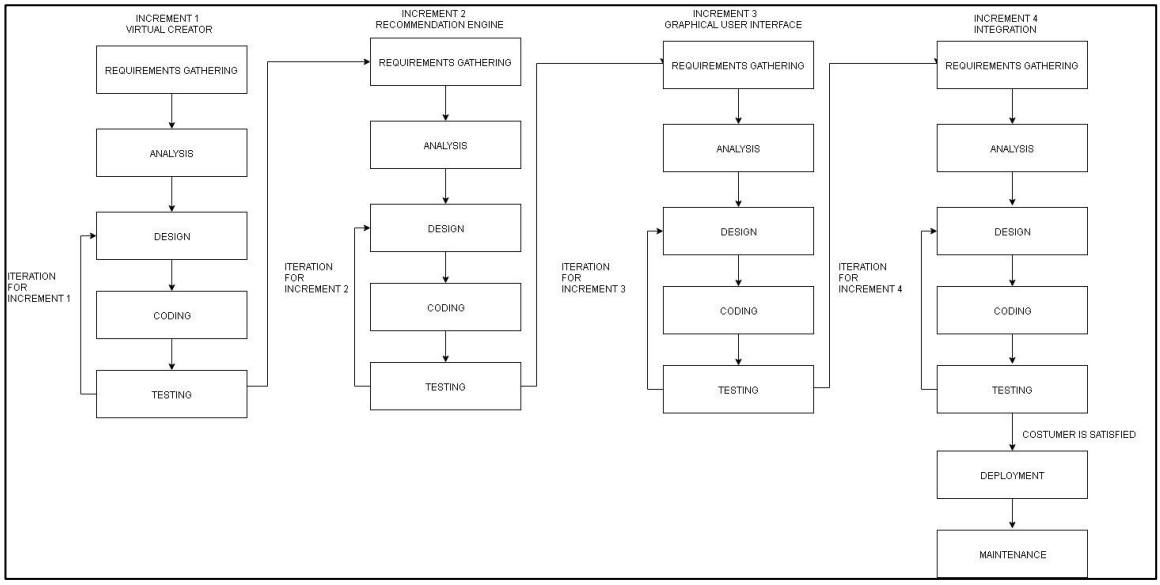

Figure 5.1 SDLC

5.3.4. SYSTEM IMPLEMENTATION PLANS

Sr. No.	Task Name	Dates
1.	Team Formation and Brainstorming for Ideas	20/06/18 - 01/08/18
2.	Idea Finalization	01/08/18 - 01/09/18
3.	Requirements Analysis	01/09/18 - 01/10/18
4.	Design Phase	01/10/18 - 15/10/18
5.	Tech Stack Finalization	15/10/18 - 20/10/18
6.	Data Collection	01/01/19 - 15/02/19
7.	Data Cleaning and Style Classifier Training	15/02/19 - 28/02/19
8.	Object Detection and Style Classifier Integration	01/03/19 -10/03/19
9.	Front End Development	01/03/19 -10/03/19
10.	Integration with Front End	10/20/19 - 10/25/19
11.	Project Review 1	25/03/19
12.	Integration and Testing Phase 1	25/03/19 - 10/04/19
13.	Documentation Updated	10/04/19 - 18/04/19
14.	Integration and Testing Phase 2	18/04/19 - 23/04/19
15.	Final Testing	23/04/19 - 12/05/19
16.	Final Guide Meeting	14/05/19

Table 5.1 System Implementation Plan - Time Chart

5.4. TEAM ORGANIZATION

5.4.1. TEAM STRUCTURE

PROJECT OPTION: In-House

INTERNAL GUIDE: Dr. Neeta Maitre

5.4.2. MANAGEMENT REPORTING AND COMMUNICATION

- In person meetings with guide and among the team mates on a 15-day to weekly basis.
- Communication through WhatsApp and phone calls.
- Usage of emails to share resources on a frequent basis.
- Usage of GitHub to share code.

CHAPTER 6

PROJECT IMPLEMENTATION

6.1. OVERVIEW OF PROJECT MODULES

The following diagrams give an overview of each project module.

Admin Module: Allows tasks like updating products and categories from the server or admin side.

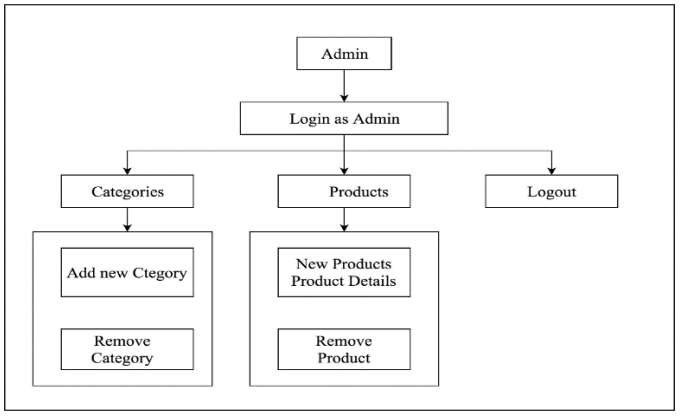

Figure 6.1 Admin Module

Client-Side Modules: These include three main modules Profile related operations, Register and Search.

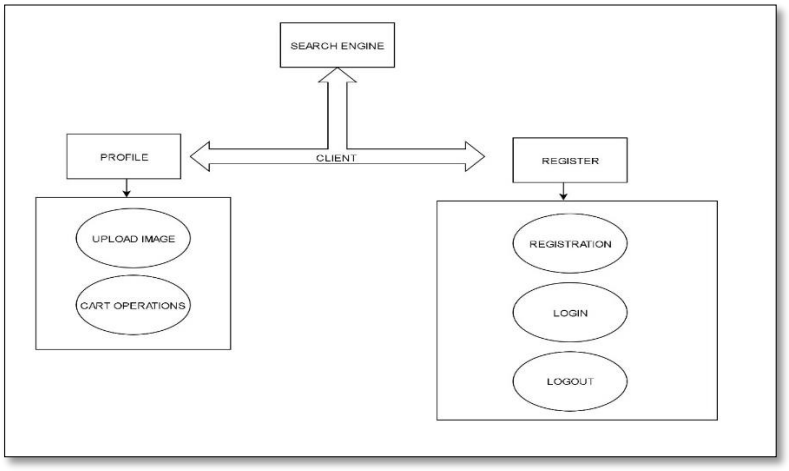

Figure 6.2 Client Modules

6.2. TOOLS AND TECHNOLOGIES USED

1. Programming Language: Python 3.7
2. Libraries: Keras, TensorFlow.
3. Frameworks: Anaconda2, Django 2.1.5.
4. IDE: Spyder notebook.

6.3. ALGORITHM DETAILS

The following section discusses the algorithmic approaches and pseudo code for the system.

6.3.1. Convolutional Neural Network (CNN)

A convolutional neural network (CNN or ConvNet) is a class of feed-forward and deep artificial neural networks which is applied to analyse visual imagery. CNNs take inspiration from a biological visual cortex. The visual cortex contains tiny regions of cells that are sensitive to specific regions of the visual field. This idea was expanded by an experiment where they showed that some individual neuronal cells in the brain responded only in the presence of edges of a certain orientation.

A CNN has an input and an output layer, also multiple hidden layers. The hidden layers of a CNN typically have convolutional layers, pooling layers, fully connected layers and normalization layers.

Convolutional

A convolutional operation to the input is applied by convolutional layers, passing the result to the next layers. The response of an individual neuron to visual stimuli is emulated by convolution. That is processed by each convolutional neuron for its receptive field.

Pooling

Convolutional networks include local or global pooling layers, which combine the outputs of multiple neurons i.e. neuron clusters at one layer into a single neuron in the next layer.

Fully connected

Fully connected layers connect every neuron in one layer to every neuron in another layer.

Weights

CNNs share weights in convolutional layers, which indicates that the same filter (weight's bank) is used for each receptive field in the layer; this results in reducing memory footprint and improving performance.

6.3.2. Psuedo code:

Create paths for variables:

FROZEN_GRAPH -> the graph for pre-trained object detection model

1. **define function run_inference_for_single_image with input parameters as IMAGE and FROZEN_GRAPH:**
   1.1. Store FROZEN_GRAPH in graph variable called detector
   1.2. Run detector on IMAGE
   1.3. return variable coordinates which contains coordinates of detected objects
2. **define function ObjectExtraction with input parameters as coordinates and IMAGE**:
   2.1. Store width of image in variable width

2.2. Store length of image in variable length

2.3. from each entry in coordinates:

2.3.1. get coordinates as->

2.3.2. ymin=each[0]

2.3.3. xmin=each[1]

2.3.4. ymax=each[2]

2.3.5. xmax=each[3]

2.3.6. rescale left,right,top,bottom coordinates as :

2.3.7. left=xmin*width

2.3.8. right=xmax*width

2.3.9. top=ymin*height

2.3.10. bottom=ymax*height

2.3.11. crop coordinates (left,top,right,bottom) from IMAGE

2.3.12. add newly created image to list l1

2.4. return list l1

3. define function LoadStylizer with input paramters as list of scaled detected objects l2:

3.1. load pre-trained model for classification task in model variable

3.2. store results from model in result variable

3.3. create dictionary called classes with key as class_label and value as number of objects

3.4. for each in result:

3.4.1. if(each==0):

3.4.1.1. Increment value in class_label=0

3.4.2. if(each==1):

3.4.2.1. Increment value in class_label=1

3.5. arrange all class_labels in descending order

3.6. style1 variable stores the class with maximum number of objects

3.7. style2 variable stores the class with next best number

3.8. send styles to database

3.9. use this ordering to show images to the user's screen

4. define function main:

4.1. take user input for image variable

4.2. call function run_inference_for_single_image and store in coordinates

4.3. call function ObjectExtraction and store in list1

4.4. call function LoadStylizer on list1

6.3.3. Front End: Django

Django is a high-level Python web framework. It enables rapid development of secure and maintainable websites. Django has been referred to as an MTV framework because the controller is handled by the framework itself and most of the great work happens in models, templates and views.

The Django web framework includes a default object-relational mapping layer (ORM) that can be used to interact with application data from various relational databases. Django 2.1 release which supports Python 3.7 has been used. Below is a description of all layers in Django.

The Model Layer

Django provides an abstraction layer (the "models") for structuring and manipulating the data of your Web application. In Django, models must be created in the models.py file.

A model is the single, definitive source of information about your data. It contains the essential fields and behaviours of the data you're storing. Generally, each model maps to a single database table.

The View Layer

Django has the concept of "views" to encapsulate the logic responsible for processing a user's request and for returning the response. A view function, or "view" for short, is

simply a Python function that takes a web request and returns a web response. This response can be the HTML contents of a Web page, or a redirect, or a 404 error.

In Django, views are created in the app views.py file.

The Template Layer

The template layer provides a designer-friendly syntax for rendering the information to be presented to the user. Being a web framework, Django needs a convenient way to generate HTML dynamically. The most common approach relies on templates. A template contains the static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

A Django project can be configured with one or several template engines (or even zero if you don't use templates). Django ships built-in backends for its own template system, creatively called the Django template language (DTL).

The Forms

To take input from our website visitors, we will need to understand and use forms. Django provides a range of tools and libraries to help you build forms to accept input from site visitors, and then process and respond to the input.

Django's role in forms:

Django's form functionality can simplify and automate vast portions of this work,and can also do it more securely than most programmers would be able to do in code they wrote themselves.

Django handles three distinct parts of the work involved in forms:

- preparing and restructuring data to make it ready for rendering
- creating HTML forms for the data
- receiving and processing submitted forms and data from the client

It is possible to write code that does all of this manually, but Django can take care of it all

The Django Admin site:

One of the most powerful parts of Django is the automatic admin interface. It reads metadata from your models to provide a quick, model-centric interface where trusted users can manage content on your site. The admin's recommended use is limited to an organization's internal management tool. It's not intended for building your entire front end around

Admin actions:

The basic workflow of Django's admin is, in a nutshell, "select an object, then change it." This works well for many use cases. However, if you need to make the same change to many objects at once, this workflow can be quite tedious.

In these cases, Django's admin lets you write and register "actions" – simple functions that get called with a list of objects selected on the change list page.

If one looks at any change list in the admin, they will see this feature in action; Django ships with a "delete selected objects" action available to all models.

CHAPTER 7

SOFTWARE TESTING

7.1. TYPE OF TESTING

1. Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done to test that the unit implemented is producing required output against corresponding input.

2. Integration Testing

Integration testing is the testing in which a group of components are combined to produce output. Also, the interaction between the software and hardware are tested in integration testing if software and hardware components have any relation. It may fall under both black box testing and white box testing.

3. Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

4. System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g. Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing. This project is python based and hence it is portable in nature. Python runs on every platform.

7.2. TEST CASES AND RESULTS

The system has been tested for the following test cases

1. Verify that user can navigate through all the products across different categories and all the links and banners are redirecting to correct product/category pages and none of the links are broken.
2. Verify that user can add to cart one or more products.
3. Verify that user cannot add more than available inventory of the product.
4. Verify that all the specified fields are present on the registration page and the required/mandatory fields are marked with * against the field.
5. Verify that for better user interface dropdowns, radio buttons and checkboxes etc. fields are displayed wherever possible instead of just textboxes.
6. Verify that clicking submit button after entering all the required fields, submits the data to the server.
7. Verify that not filling the mandatory fields and clicking submit button will lead to validation error.
8. Verify that not filling the optional fields and clicking submit button will still send data to server without any validation error.
9. Check validation on date and email fields (only valid dates and valid email Ids should be allowed.
10. Check validation on numeric fields by entering alphabets and special characters.
11. Verify that after making a request to the server and then sending the same request again with the same unique key will lead to server-side validation error.
12. Verify that authenticated sellers get access to product creation panel specific to the authorized categories.
13. Verify that product creation panel is working fine for single product creation and multiple product creation.
14. Verify that duplicate product creation is restricted through panel.
15. Verify that seller can update information and price of existing products.

CHAPTER 8

RESULTS

7.3. OUTCOMES

7.4. SCREEN SHOTS

CHAPTER 9

CONCLUSION

8.1. FUTURE WORK

- Extensions can be made to the GUI of the web-application. The system can be made more user-friendly by adding support for local languages like Marathi, Hindi and so on. Extensions can also be added for physically challenged persons. For example, for a blind user, a section can be created, where the item is described verbally to the user and the various possible arrangements are also described verbally.
- A functionality can be added which enables users to compare two or more choices for a furniture item at the same time. This can be achieved by showing arrangements for each choice in a list or slider format.
- A 360 degrees view can be shown of the furniture item when the user hovers over an image while selecting items.
- Different arrangements of the items selected by the user can be shown virtually in the uploaded room image.

8.2. APPLICATIONS

The Personalized e-Décor application can be used by anyone who wants to decorate their home. It can be extended to different buildings like offices, shops etc. Finally, this web application can be used by local vendors of furniture, to show how their items will look in the customer's home.

Also, interior designers may use it to show their work virtually first and reduce their work of drawing. This can be achieved by extending the scope of this project to include more technical specifications required by interior designers. The students of interior design can improve their imagination with the help of Personalized e-Décor, by learning how to

design different places, test their skills and create a portfolio, or by comparing the differences in various arrangements of the same item in the same room.

8.3. CONCLUSION

The developed system is designed to recommend furniture to a user based on the style of the room. Convolutional neural networks, a deep learning method for classifying images, has been used for recommending furniture items for an everyday online shopper.

APPENDIX A

I	D	E	A
INCREASE- The categorization capability of machine	**DRIVE-** Decrease generated return requests and save time spent shopping offline for furniture.	**EDUCATE-** Machines, end users	**ACCELERATE-** Feature extraction process
IMPROVE- Accuracy of the classifier	**DELIVER-** A well-furnished room with newly added objects	**EVALUATE-** The classified images for accuracy and precision	**ASSOCIATE-** End users and furniture vendors
IGNORE- Low quality images	**DECREASE-** Inaccuracy of classification	**ELIMINATE-** Intermediaries in Interior Decoration	**AVOID-** Human dependency for classification and virtual creation of the room

Table 9.1 IDEA Matrix

Definitions:

Non-Polynomial(NP) – The set of problems for which no polynomial time algorithm is known. This includes problems for which the only known algorithm require a number of steps which increases exponentially with the size of the problem, and those for which no algorithm is known.

Many of these problems are computationally related. The two classes of NP problem

NP complete: It is a decision problem that can be solved with Non-Deterministic Turing machine.

NP hard: If these problems can be solved in polynomial time then the NP complete problems can be solved in polynomial time.

Satisfiability problem:It involves finding out whether given expression is true for some assignment of truth values to the variables in that expression.

Problem statement:

The personalized e-décor is an online application dedicated to furniture shopping. It serves two purposes. Firstly, it asks the user to decide on the item he/she wants to buy, for example a bed.

Then he/she is asked to select filters for this item, a décor style such as retro, vintage, modern or a colour shade. The user is also asked to provide the minimum dimensions of the intended room for this item, in terms of length, breadth and width.

Now, e-décor provides its user with a range of choices for the furniture item, which fits in with the filters applied by him/her. After a user is satisfied with the choices in the cart,

they may see how it will look in their room. This is the second purpose served by personalized e-décor. A picture of the room is to be uploaded when the user logs into the application. A user may continue to add and delete choices from the cart and iteratively use the virtual creator to visualize them in this room. The user may continue to do this until an item matches their expectations.

The original problem falls in the category of NP hard problems as:

1. There are number of scenes in which images can be classified.
2. It is difficult to partition different scenes in single image.
3. The variation in quality and resolution of image is difficult to handle.
4. There is a huge dataset of different type of images which are used for training the model.
5. Some images with low light and obstructing objects are difficult to classify.

So, the problem is reduced to NP complete.

1. Considering few categories to classify the scenes like table, lamps, curtains, books, bed etc.
2. Considering high quality images.
3. Dealing with standard and specific set of images.

APPENDIX B

10.1. ACM TECHNICAL KEYWORDS

[1] Artificial Intelligence

a. Machine Learning

i. Deep Learning

1. Convolutional Neural Networks

[2] Computer Vision

a. Object Detection

b. Object Recognition

c. Image Representation

APPENDIX C

REFERENCES

[1] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan, "Show and Tell: A Neural Image Caption Generator", arXiv:1411.4555v2 [cs.CV] 20 Apr 2015.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", arXiv:1411.4555v2 [cs.CV] 6 Feb 2015.

[3] Yangbangyan Jiang, Qianqian Xu, Xiaochun Cao, Qingming Huang∗, "Who to Ask: An Intelligent Fashion Consultant", ICMR'18, June 11-14, 2018.

[4] Peter K´an, Hannes Kaufmann, "Automated Interior Design Using a Genetic Algorithm", VRST2017, November 2017, Gothenburg, Sweden 2017. ACM ISBN 123-4567-24-567/08/06.

[5] Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, Vladlen Koltun, "Interactive Furniture Layout Using Interior Design Guidelines", ACM Trans. Graph. 30, 4, Article 87 (July 2011),

[6] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", arXiv:1411.4555v2 [cs.CV] 17 Apr 2017.

[7] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, "How transferable are features in deep neural networks?", Part of Advances in Neural Information Processing Systems 27 (NIPS 2014)