International Journal of Research in Engineering, Science and Management
Volume 5, Issue 7, July 2022
https://www.ijresm.com | ISSN (Online): 2581-5792

Customary Superintelligence for An Enterprise

P. Geethika[1*], Jyothi Shetty[2]

[1]*Student, Department of Computer Science and Engineering, NMAM Institute of Technology, Udupi, India*
[2]*Professor, Department of Computer Science and Engineering, NMAM Institute of Technology, Udupi, India*

***Abstract*: A chatbot or chatterbot is an AI technology to capable of holding a valuable conversation with a user to meet a specific requirement. These chatbot can hold conversation between services and user to meet customer and can remove any limitation to customer support. This work is to build a conversational AI for a paint manufacturer with RASA architecture to help customers to select wall colours for their home or office with image processing and manipulation using OpenCV.**

***Keywords*: Chatbot, dialog flow scripting.**

1. Introduction

The artificial intelligence (A.I.) has grown in popularity for influencing human interactions with gadgets and devices in our daily life especially on mobile platforms. Since the proposal of Turing test, the Artificial Intelligence have been a wide subject of interest in our technological community. From then, different chatbots are developed. The functionality of these chatbots ranges in the field of education, business and e-commerce. Health and entertainment. Productivity, scalability and reduced human resources are the most common values a well employed chatbot can bring to an enterprise.

The integration of personal assistance to phones and mobile devices leaped artificial intelligence to the next step of speech engines and image processing. The arrival of apple Siri, google now, and amazon Alexa are most common example of this advancement. Microsoft Xiaoice is an AI chatbot designed to satisfy the need of human socialization. It has both intelligent and emotional quotient which enables a long emotional relationship to user taking to account for cultural and ethical peculiarities [1]. Today chatbots can use machine learning algorithms to understand and answer the user without comparing developer defined responses. But this intelligence requires a huge amount of training to make the chatbot to answer contextually.

2. Literature Survey

Chatbots are generally developed by two approaches 1. Pattern based approaches and 2. Machine learning based approaches.[1]. In pattern-based approaches user input from a set of predefined queries and the chatbot replies with the use of pattern matching algorithms. Eliza, Parry, Alice, Jabberwacky and so many employed this technique. These systems normally do not gives new conversation input as the question and answer is written by the developer in the creation itself. This is the major downside of such system as the responses are mechanical and extensive set of well defines rules pattern will take to make the chatbot more humane. This can be achieved by multi turn pattern selection technique which feedback every response to choose a response suitable for the whole interaction. [1] There are three most common languages user in the development of chatbot using pattern matching technique, 1. Artificial intelligence markup language (AIML), 2. River script and 3. Chatscript. The AIML is developed in the latter half of 1990s and is based on XML. ALICE was the first bot to employ knowledge based AIML. AIML Data bases include catalogue. The AIML contains data objects which includes categories relevant to the aspect of the specific chatbot. The Riverscript is line-based scripting language implementing the knowledge in rule based chatbot where the syntax “+” means a user input and “-” means chatbot output. The user inputs are matched by the interpreter matches with stored responses and determines the most suitable reaction [2]. ChatScript is a law-based engine, where rules are made by human authors in program scripts through a process called dialog flow scripting. These use text metalanguage (or "script") as their source code. [12]

Chatbots that use Machine Learning Methods instead of Pattern Matching extract the content of user input using Natural Language Processing (NLP), and lose the ability to learn in conversations. They consider the whole topic of discussion, not just the current opportunity, and do not need a predictable response to each user's input. In general, they need a comprehensive training set, the acquisition of which may create significant difficulties as the available data sets may not be sufficient. [1].

For a variety of applications, RASA is using an open source NLU model and DIET. It can interact with database, api, chat flow, interactive reading and strengthen Neural network. In their study Mr. Rakesh Kumar Sharma and Mr. Manoj Joshi explores key aspects of rasa such as spaces, forms, interactive supervised learning, api integration, and website making a complete framework that can be used to perform more complex tasks. Rasa-based chatbot has more capabilities than any other open-source method. Additionally, in this paper, the interior of the rasa has been modified to create custom log data for client ip and hole. All insiders and custom action have been reviewed which further means that rasa is a complete open-source framework for the development of chatbots and for engineers who do not want to get inside the natural language processing.

*Corresponding author: geethikapnair@gmail.com

Originally developed by Intel, OpenCV (Open-Source Computer Vision) is a free cross-platform image processing library that has become a common tool for all computer Vision-related objects. The first version was released in 2000 under the BSD license and since then, its performance has been greatly improved by the scientific community. In 2012, the nonprofit organization OpenCV.org took over the responsibility of maintaining the support site for developers and users [13]. Image processing is a form of signal processing where the input is an image-like image or video frame, output an image or a set of image-related features. OpenCV is a library of editing activities used primarily for image processing. [7] Examines the various cv connections open in Python, Java and MATLAB. Because of this compatibility OpenCV can work on various platforms such as windows, android, blackberry, openBSD, iOS and Linux.

3. Implementation

This work is to develop a chatbot for a paint manufacturing company to select suitable wall color for their walls. The design of chat bot is too great and welcome the user, ask for specific requirements like placement of wall, intended texture, and specific product. After receiving these inputs ask for the color from shown list and then to input the desired image and give desired results

The image processing by open cv is done by two methods separately for quality assessment which are *a*. operational method and *b*. database method.

A. Operational Method

The image processing in this project is done by image manipulation using set of operations which converts the given image to a processed. The process can be categorized into three parts, 1. Preparing image, 2. Background threshold, 3. Coloring.

1) Preparing Image: The code makes a copy of image and slightly blurs the image so that algorithm doesn't get triggered by minor contours on the image. After that script converts image to grayscale - Gray scaling the image is important in cv, It reduces model complexity that may arise if we use RGB images. It drastically reduces computational requirements. After Gray scaling the script applies Sobel Operator. It computes image edges and modifies the image to contours only based on parameters. At this stage the image will look like a initial stage of pencil sketch.
2) Background Thresholding: After Stage 1, we input thresholding – it identifies largest area containing a particular pixel. Here we threshold with white color. So, after applying this operator the script returns largest same colored area in the image. After getting largest white area in the image, we find edges of that area using find Contours method
3) Coloring: At this stage we have all the edges of largest same colored area in the pixel. after that we color, that area using draw Contours method.

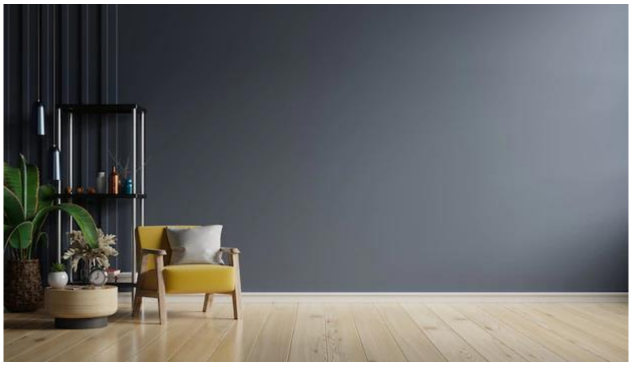

Fig. 1. Sample image for processing

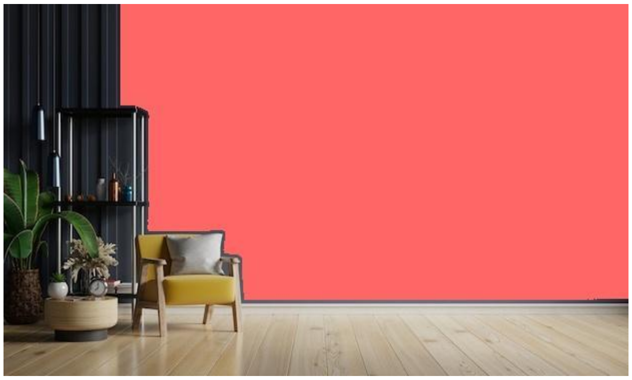

Fig. 2. Processed image

B. Database method

In order to use database for image processing and have a better quality we can add existing data base to object detection and train the bot to exclude those objects from coloring. But we decide to create our on-data base to reduce computational power.

We are focusing on 3 objects to train the bot to exclude from the coloring additional to operational method for better accuracy for object detection, Coffee table, flower pot and sofa. There are several steps for training a chatbot to detect object in python.

a) Creating separate directory for each object with quality images. This can be identified as the positive image directories.
b) Create an annotation file with path to those directories
c) Create a .vec file with images of object in binary format
d) Create a negative image file of your object. Negative image of an object is defined as the object is absent in the image. Creating negative image that resembles the same environment is advised as their computational power is increases as the environment of both positive and negative images differ.
e) Run the bot with directories for training.

The fig. 3 is the one processed with database and the chatbot is successfully detected and avoided painting inside and in the edges whereas in fig. 4 couldn’t avoid the edges and carried out drawing thinking it was wall boundaries and over drew on the tile edges because of the edge detection we set low. The difference between these two images is shown in fig. 5.

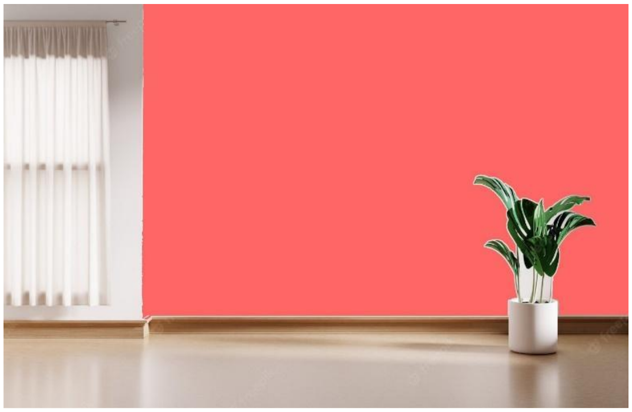

Fig. 3. Processed image using database

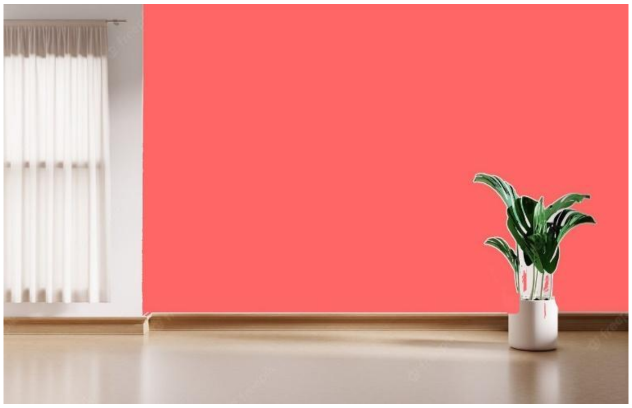

Fig. 4. Processed image without using database

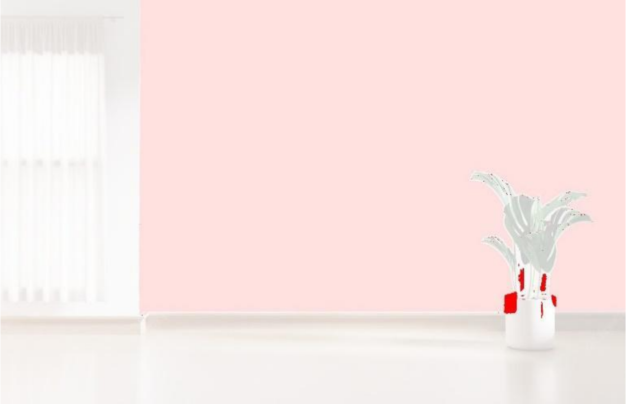

Fig. 5. Difference of fig. 3 and fig. 4

Here in operational method, the portion highlighted in red is not identified as pot or any other object and went onto color. In fig. 3 the pot is identified with the data base, and excluded for painting.

By comparing these two methods we can come to the conclusion that by employing database in object detection can significantly advance the result. Building a reliable database is the key and tedious part of this method.

4. Conclusion

This paper presented an overview on customary superintelligence for an enterprise.

5. Future Scope

As artificial intelligence advances, scope of chatbot is only limited to the developer's imagination. Chatbot can replace any online interaction with integrating speech and face recognition engines which are capable of any complex task like sales and service of any retail industry, Automated farming units, receptionists, presenters for any public or private experience venues.

References

[1] Eleni Adamopoulou, Lefteris Moussiades "Chatbots: History, technology, and applications."

[2] Munira Ansari, Saalim Shaikh, Mohammed Saad Parbulkar, Talha Khan, Anupam Singh, 'Intelligent Chatbot' IJERT 2021.

[3] Rakesh Kumar Sharma, Manoj Joshi, "An Analytical Study and Review of open Source Chatbot framework, RASA" IJERT 2020.

[4] Tom Bocklisch, Joey Faulkner, Nick Pawlowski. Alan Nichol. "Rasa: Open-Source Language Understanding and Dialogue Management" NIPS 2017.

[5] Anran Jiao, "An Intelligent Chatbot System Based on Entity Extraction Using RASA NLU and Neural Network," J. Phys.: Conf. Ser. 1487 012014, 2020.

[6] Daniel Braun, Adrian Hernandez Mendez, Florian Matthes, Manfred Langen, "Evaluating Natural Language Understanding Services for Conversational Question Answering Systems."

[7] M. Naveenkumar, A. Vadivel, "OpenCV for Computer Vision Applications," NCBDC 2015.

[8] Haumik Kohli, Tanupriya Choudhury, Shilpi Sharma, Praveen Kumar, "A Platform for Human-Chatbot Interaction Using Python."

[9] Rajeshwar Kumar Dewangan, Yamini Chouhan, "A Review on Object Detection using Open CV Method," IJERT 2020.

[10] Sidra Mehtab, "Object Detection and Tracking Using OpenCV in Python."

[11] Ayushe Gangal, Peeyush Kumar and Sunita Kumari, "Complete Scanning Application Using OpenCv."

[12] https://github.com/ChatScript/ChatScript

[13] Gloria Bueno García, Oscar Deniz Suarez, José Luis Espinosa Aranda, Jesus. Salido Tercero, Ismael Serrano Gracia, Noelia Vállez Enano, "Learning Image Processing with OpenCV," Pakt Publishing, 2015.

[14] Deborah T. Joy, Gitika Kaur, Aarti Chugh and Shalini B. Bajaj, "Computer Vision for Color Detection," IJIRCST 2021.

[15] https://www.geeksforgeeks.org/

[16] Guobo Xie, and Wen Lu, "Image Edge Detection Based on Opencv," in International Journal of Electronics and Electrical Engineering, vol. 1, no. 2, June 2013.