Proceedings of the International Conference on Artificial Intelligence and Smart Systems (ICAIS-2021)
IEEE Xplore Part Number: CFP21OAB-ART; ISBN: 978-1-7281-9537-7

Guidance System for Visually Impaired People

Kanchan Patil[1], Avinash Kharat[2], Pratik Chaudhary[3], Shrikant Bidgar[4], Rushikesh Gavhane[5]
Department of Information Technology, SRES's Sanjivani College of Engineering
Kopargaon-423601 (MH), India.
[1]patilkanchanit@sanjivani.org.in,
[2]avikharat9921@gmail.com,
[3]pratikchaudhary7057@gmail.com,
[4]shrikantbidgar50@gmail.com,
[5]rushikeshgavhane2000@gmail.com

2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS) | 978-1-7281-9537-7/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICAIS50930.2021.9395973

Abstract— **There are some visually impaired people throughout the world. Some of them may be around us. The visually impaired person finds difficulty while performing day-to-day life tasks. So this research work aims to develop a device which helps them as personal assistant. This paper represents the proposed device's integrated modules and functionalities that can help a blind person. The proposed idea is to provide a wearable device with a Virtual assistant system for the visually impaired person, for some of the basic tasks without requiring the help of others. The system is aimed to provide voice-over assistants for blind people to do tasks like understanding surroundings, looking for an object, recognizing the face of a person with emotion, and reading etc. There is a total of five components merged into one system in this project. The navigation through these components is possible through hardware buttons and voice-over commands given by the user. There are many deep learning methodologies and core libraries of python language used for programming. The complete project is dedicated to being simple to use by visually impaired people and making day to day tasks easy for them.**

Keywords—AIML; Python 3.4.7; JARVIS; gTTS; Pyttsx; Image Captioning; Face Recognition; Facial Expression Recognition; Object Detection; Yolo v3, Optical Character Recognition.

I. INTRODUCTION

In our societies, many people are suffering from different diseases or handicaps. In the world, so many people of different age range who are visually impaired are estimated to be 285 million, and out of them 39 million are blind according to WHO. These people may be from various backgrounds such as farmers, teachers, sportsmen, housekeeper, housewives and many more. They may be from different age groups. Some of them may be children, some are youngsters and some of them are old people. So for each and every individual who is visually impaired need to face different challenges every day. Even though they have different abilities and they can achieve many good things in life, sometimes they need help of someone for some tasks. To help such people and make their day-to-day activities easier is the main motivation behind this project. Blind people need to be provided with special facilities so that they can live comfortably. We wanted to give them a helping hand that they use as their assistant. The assistant idea came into the picture inspired by Jarvis which is a fictional character from the marvel world and also from Sanjaya - a character from Mahābhārata - The ancient Indian Hindu war epic. Sanjaya was the descriptor for Dhritarashtra, He described the whole war of Mahabharata to Dhritarashtra. We wanted to create a virtual assistant that acts like Sanjaya to a blind person. The system is a device with earphones, a camera and a processing device that stays in the pocket. The camera and earphones are attached with goggles and the processing device stays in the pocket. In this paper, all the modules used in the device and their functionality is described.

II. LITERATURE SURVEY

A. AIML Chatbot

R. Sangpal, T. Gawand, S. Vaykar and N. Madhavi, [5] have represented an intelligent chatbot assistant state of art based on python and AIML. The assistant is aimed to carry out the tasks just like a human assistant. The system provides spoken word solutions to any problem or any question asked. The python packages like GTTS (google text to speech) used for audio reply and speech recognition to convert audio command into text format. The pattern matching of the commands or text with existing dialogues and conversation is done by using AIML. The smooth conversation is carried out using large, stored dialogue in AIML syntax. The python interpreter is the backbone of this system.

N. N. Khin, K. M. Some [11] have implemented a university chatbot using AIML. The chatbot is designed for university students asking queries in Myanmar language. Pandorabots is used as interpreter in that chatbot. Researchers used 3 different AIML categories such as atomic categories, default categories, recursive categories. In the recursive category, following techniques are used for smooth conversation between user and bot.

a. *Symbolic reduction*: It helps to minimize the grammatically complex forms into simpler ones.

b. *Divide and conquer:* It breaks an input into two or many subparts. It adds the responses to one part.

c. *Synonyms resolution*: It is possible to appear different words with the same meanings.

d. *Keyword detection*: It is used to find same response when a definite keyword is discovered in input.

V. K. Shukla and A. Verma [10] have proposed a system which is a chat-bot for LMS (learning management system). They used AIML for dialogue storage and pattern matching and R as an interpreting language. They have also proposed architecture for the systemwhich includes Rule-based chat-bot and Retrieval Based Database chat-bot. Rule-based chat-bot compares the fixed words, expression with the given database and reply based on that only where Retrieval Based Database chat-bot system has capability to produce responses based on specific words. If a query does not have these words, then response will not be produced. Chat-bot itself do not have capability to produce new responses. To make effective rule based chat-bot, different keywords from query will be extracted with the help of R language pattern matching.

B. Image Captioning

Image captioning can be defined as a process of retrieving image insight in natural language. It describes the image in natural language in human understandable form. There are three types of existing image captioning methods. First one is Template based image captioning, second is Retrieval based image captioning and the third is Novel caption generation method.

To perform image captioning, Aneja [12] proposed a convolutional architecture. The architecture is four layered feed forward network and different layers are input embedding layer, image embedding layer, convolutional module, and output embedding layer. The recurrent function is not used in this architecture. An attention mechanism is used to gain more image features. The complete architecture is tested on MSCOCO dataset.

Wang [13] proposed another Convolutional Neural Network for image captioning. It is homogeneous to the method explained by Aneja [12], but it uses attention module which connects to vision-Convolutional Neural Network and language - Convolutional Neural Network

Gu [14] proposed a Convolutional Neural Network language model-based image captioning. For statistical language modeling, it uses language-CNN. But, this method is not capable of modeling the dynamic temporal behavior of the language model. The current network is combined with the language CNN for modeling the temporal dependencies properly.

A Novel Object Captioner (NOC) was introduced by Venugopal [15] to generate captions for objects which were not visible in the image. To recognize such objects, external sources and semantic knowledge was used.

In one of the paper, Huang [16] discussed that the selection of appropriate attribute will lead to the increase in performance of image captioning model. The authors used multimodal attribute detector (MAD) and subsequent attribute predictor (SAP). Using MAD and SAP together efficiently extract high semantics from image results in good performance of image captioning model.

C. Face Recognition

In the paper [17], the authors discussed Smart attendance System using facial recognition technique. To develop the system, authors used different algorithms. Viola-Jones was used for face detection. To cutoff part of the images, RoI (Region of Interest) was used. PCA (Principal Component Analysis) was used for feature selection. SVM (Support Vector Machine) was used for the classification. During this research, pictures of each individual were captured and implementation was performed on images in different situations, angles and lightings [17].

In the paper [18], the authors proposed technique which consists of two steps. In the first step, Dataset of faces is created and the second step is to convert color space in image to Hardware Security Integration (HIS) using saturation layer. Image fragmentation is performed using Curvelet transform; feature extraction takes place using PCA. Face recognition is performed using Elman neural network. After applying proposed algorithm, result shows 94% accuracy.

D. Emotion Recognition

S. Divvala, J. Redmon, A. Farhadi and R. Girshick [20], have proposed in their paper, The YOLO architecture for object detection. They redefine the object detection problem as regression problem instead classification. This helped to improve speed and optimization performance. Due to this change, the results were better than other object detection methods such as Region-based Convolutional Neural Networks. Fast YOLO processes per second, 155 frames [20].

In proposed research [1], the researchers obtained a YOLO compact network. This is exclusively used for single category object detection. The AP result of YOLO-compact is 86.85%. The authors also discussed the methodologies to convert a huge & heavy network to a tight and efficient network. They have performed number of experiments on YOLOv3 network and the YOLOcompact network's infrastructure was obtained. They have used the pedestrian category from the VOC2007 test set as their single category to detect and show the result for the same [1].

Yan W.Q. and Q. Zhang [7], have designed a deep neural network to identify currency notes. The Single Shot MultiBox Detector (SSD) model gives 96.6% accuracy for identifying the correct denomination of the paper note [7].

III. Proposed System

As discussed earlier, proposed system is a wearable device. The device is mainly composed of two parts: The first one is a headset with a camera, microphone, and earphones and the second is a processing device that can easily be worn or carried in a bag and contains the battery and the processor. One cable will help to connect the headset with processing device.

The system consists of five component modules of the device which can be accessed via voice over commands and the architecture is developed using a python interpreter. The deep learning state of the arts such as image captioning, object detection, and OCR (optical character recognition) which are used to develop the systems functionalities. AIML (artificial intelligence markup language) is used to develop the assistive chatbot. The state of art is developed in python.

Following Fig.1 describes the system architecture of the device which explains the interfacing of different modules of the system with camera interface and headset for voice over conversations.

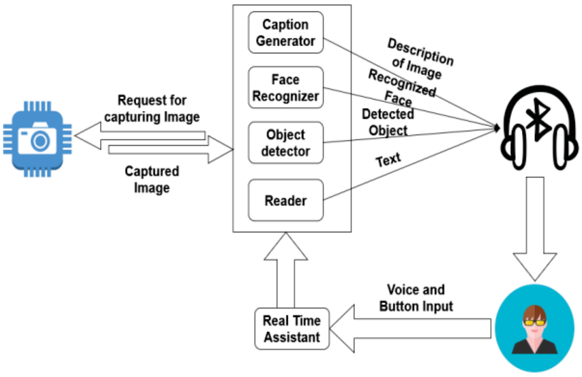

Fig. 1. System architecture of device

Proposed System follows some commands to function well and to interact with the user. Fig. 2 describes the flow of commands and user activity with the system.

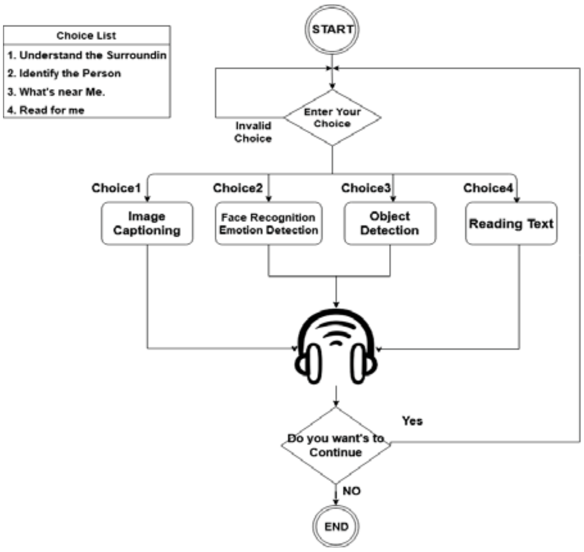

Fig. 2. Flow of commands in system

As we can see in the Fig. 2, there are four different sets of commands divided with respect to modules. Whenever the system will ask the user for the choice, user will give the choice. The given choice is compared with existing available choices and then the task associated with the specific command will be performed. If the command is not recognized by the system then again user is given a chance to choose another command. The cycle continues until the user chooses to exit the system.

The modules in the system are image captioning, object detection, face-emotion detection, face recognition, voice-over chat bot and reading. A voice-over chat-bot takes voice command as input. The voice command is then processed and the respective module is executed which are mentioned above. After execution of module output generated by module will be sent back to voice-over chat-bot which will be further audible to blind people using Bluetooth devices. This cycle will continue until the user chooses to exit. The detailed explanation of each module is given below.

A. Voice-over Chatbot

Voice-over Chatbot module is used to communicate with user via text or text-to-speech. The chat-bot module is the backbone of the system as it will be interacting with the user over voice commands. The module is inspired by Jarvis which is a fictional character from the marvel world. To make the chat-bot interactive and realistic AIML is used as the backend to write dialogues and replies. pyTTSX (Python) and gTTS (Google text to speech) are used for audio output and python as interpreting language. The Module will be responsible for voice-based conversation between user and system. The flow of chat-bot modules is divided into four parts.

a. ***Taking input:*** The voice-over input is taken using Pyaudio and speech_recognition which are python modules. The dialogue script of this chat-bot is written in dialogue and category pair. For example, if the dialogue is “detect the objects” then its reply will be “object detection”.

b. ***Categorization:*** Here we are processing the recognized text to obtain desired category of the dialogue to perform task. If the dialogue is “Detect the objects”, then by using pattern matching of AIML the category for object detection module is obtain i.e. “object”.

c. ***Collection:*** After retrieving a category of text, the system will perform the task with respect to category, like in example object detection category was chosen. So the task will get performed by module and data will get sent back to chat-bot. Then the collected data is again processed by the chat-bot module to convert data to meaningful sentences. For example, “Detected objects are table and fruit basket”.

d. ***Reply to the user:*** By using gTTS (google text to speech) the output dialogue is audible to the user. It will be a reply from the system.

```
<aiml version=”1.0.1” encoding=”UTF-s”?>
  <category>
     <pattern>HELLO ALICE</Pattern>
       <template>
                 Hello User!
       </template>
  </category>
</aiml>
```

Fig. 3. Algorithm for Basic HTML chat-bot Script

Fig. 3 represents the basic AIML script which recognizes the pattern “HELLO ALICE” and in reply to the text enclosed in <template> is sent to the system i.e. “HELLO USER”. So to make AIML compatible with our system we wrote our own

AIML script which matches the pattern with commands like “explain surrounding to me” and in reply to this the category of dialogue is sent to the system i.e. “image captioning”. We will use a <random> tag here, so that every dialogue will have more than one response which can make the chat-bot more realistic. The chat-bot also performs as a personal assistant and can carry out certain tasks like indicating time, giving information about topics from Wikipedia, meeting notes. Python modules like time and Wikipedia are used.

No.	*Tags used for AIML Categories*
1	*<topic></topic>*
2	*<category></category>*
3	*<pattern></pattern>*
4	*<template></template>*
5	*<srai></srai>*
6	*<random></random> with <li></li>*
7	*<set></set>*
8	*<get></get>*
9	*<that></that>*
10	*<think></think>*
11	*<condition></condition>*

Fig. 4. AIML tags used in chat-bot

If any command or voice-input is not understood by the chatbot then it will ask the user what task the user wants to perform over that command and store it as AIML conversation for future reference.

B. Image Captioning

The image captioning model is used to describe the image in natural language. It consists of two sub modules such as image processing and natural language processing. It is a process of retrieving semantics from image and presenting it clearly so as human shall understand it. Natural Language Processing and Computer Vision are used here. To generate captions for a given image deep learning model is used. It is trained on 30K images from Flickr and build over pretrained zInceptionV3 model and GloVe. Word Embedding with Convolutional Neural Net and Recurrent Neural Net are used to generate results. Fig.5 explains the architecture of image caption generator. Image Vector and Partial caption is given as input to the feed forward network and the predicted word will be generated as output in the sequence of partial caption. This module will be used to give the user an idea about the surrounding. The result is in the form of natural language.

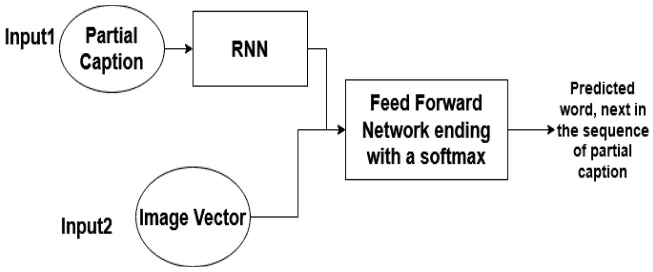
Input1
Partial Caption
RNN
Feed Forward Network ending with a softmax
Predicted word, next in the sequence of partial caption
Input2
Image Vector

Fig. 5. Architecture of Image Cation Generator

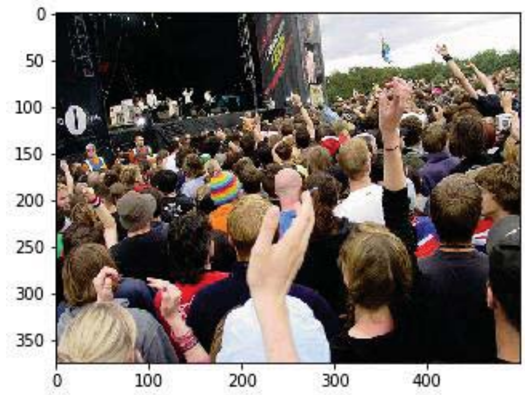

Fig. 6. Image with generated cation as “A group of men dancing in front of crowd”

Most of the time it works fine but sometimes the generated caption may not make any sense so the generated caption will be passed through a data filter module.

C. Face Recognition Module

One of the challenges that a blind person are facing is recognizing the faces of individuals. The proposed face-recognition module will help blind person to recognize the face of friend or family member and other people whose database is already stored. It will be useful for the person to identify people in front of him.

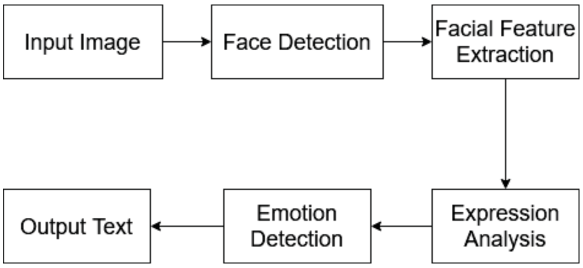

Fig. 7. Flow of operation in face recognition

The face recognition module is divided into following three steps.

a. Detecting face in the image: The detection of a face in the image is done by using OpenCV[6] (Computer Vision Library) and haar_cascade [8]. haar_cascade is a binary file used to detect specific objects available in the given image. The location of the face is mapped over the image.

b. Extracting the features: After retrieving the area of interest the feature extraction takes place. The features like width /height of a face, color, width of lips, nose etc. are extracted. The features are then scaled so that size of face in image does not affect the ratio of feature stored in database. All features will get stored in XML form of data.

c. Comparing the database: The extracted features from step 2 are then compared with the existing data of faces along with the name of the face. If the match is found then the labeled name is forwarded to emotion detection. If the face doesn't exist then the user is asked to provide the name of the face and then it gets stored in the database.

D. *Emotion Detection Module*

Recognizing the emotions of people around can be possible with this module. In this module, we trained a CV model that was built using Keras and VGG16 – a variant of Convolutional Neural Network. We are using this model to check the emotions in real-time using OpenCV and webcam. The model trained on a fer2013 dataset containing 35k face features with labeled 7 emotions which are Angry, Feared, Disgust, Happy, Sad, Surprise, and Neutral. a.

The extracted data of face from face detection module is used for the prediction of emotion. The output is forwarded to chat-bot including name of face and emotion on face.

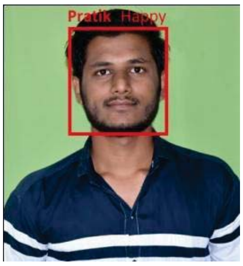

Fig. 8. Person's image with recognized face and emotion

As we can see in Fig. 8, the person's face is identified as stored in the database and the emotion is also recognized as happy.

E. *Object Detection Module*

This module is intended to help the user to recognize the objects in front of him. We are going to use YOLO (You Only Look Once) [1][20] algorithm to detect the objects. YOLO was introduced to produce a single-step procedure which includes classification and object detection. YOLO is successful to gain 45 FPS and further a lite version named Tiny-YOLO, achieves around 244 FPS on GPU computers. The model used in the state of art is resnet [2]. Inception-ResNet-v2 is a CNN that is trained on more than a million images from the ImageNet [3] dataset. 164 layered deep networks can classify images into 1000 object categories. These categories help the system to notify the user about the surrounding objects. The objects with the most prediction accuracy are transferred to the chat-bot. The process of bounding boxes is dropped because visual confirmation is of no use in the system.

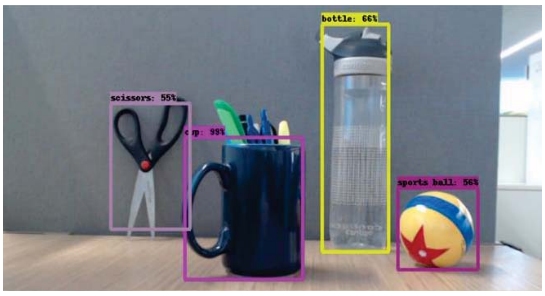

Fig. 9. Detected Objects with accuracy

F. *Reading module*

Reading is one of the key factors for the system. The thought of a complete assistant is incomplete without reading for a blind person. The camera module captures the image text. The Tesseract software is used to extract the text from the image. Tesseract software converts the image to text using following techniques.

a. Preprocessing: For better character recognition, image is preprocessed. In pre-processing, the technique of binarization is carried out by separating the text from its background. In this, a properly aligned image is converted into binary image. Binary image has two colors such as black and white. Before binarization, each image undergoes the process of alignment.

a. Character recognition: A two-stage approach was used for character recognition. Adoptive recognition is the second stage in which the Tesseract predicts the text from the stage 1 output. Based on the shape of the recognized data, the text is predicted. This predicted text is transferred to a chatbot for text to speech reading.

IV. IMPLEMENTATION DETAILS

This section of the paper explores some deep learning frameworks, APIs and modules to implement the project as follows:

TensorFlow: The proposed system includes deep learning models like object detection and image captioning. To train these models, the tensorflow framework is used.

Opencv: The tasks of image processing are performed using opencv library.

Keras: various keras processes are used in this system for deep learning methodology.
Tesseract: tesseract used to extract text from image for reading.

gTTS (Google Text-to-Speech): text to speech api is used in chatbot conversation for audio reply.

Pyttsx: text to speech api is used in chat-bot conversation for audio reply.

Speech recognition: used to recognize text from audio input from the user.

AIML: used for pattern recognition for categorization of command.

FUTURE WORK

The Visual navigation module can be include as the future scope of the proposed work which will help the blind person to navigate through surrounding using ultrasonic sensors and a camera. In the voice-over assistant module without limiting the English language, the commands can be given in Indian regional languages like Hindi, Marathi, etc. A personalized voice recognition module can be included to detect the voice of the user who is using the system.

CONCLUSION

In this paper we presented a wearable device for visually impaired people which will help them to do the tasks such as reading, recognizing the person etc. Voice-over Chat-bot is the backbone of the system as it will be interacting with the user over voice commands. The image captioning model is used to describe the image in natural language. Face detection modules will help recognize friends and family members. Reading module will extract text from the image and last one object detection will recognize the object in front of the person. Although the system is very effective and useful for visually impaired people, It will be hard for chat-bot to recognize the command in noisy environment, chat-bot may get confused between voice of an user and person nearby it. The system will need high processing power to run the computer vision deep learning models.

ACKNOWLEDGMENT

This work is supported by Sanjivani College of Engineering, Kopargaon. We are thankful to our management for providing us a platform to complete our research work. We are thankful to our beloved Director for his continuous support and encouragement to climb up new heights. We are very much thankful to our Head of Department, Dr. M. A. Jawale for her guidance and support. We are thankful to our colleagues and friends, without their support it would not be possible to complete this work. Love and blessing showered by our parents kept us motivated to complete this work. We would like to say thanks to our reviewers for their insights and comments. Lastly, we are thankful to each and every one who has directly or indirectly helped us to make this research successful.

REFERENCES

[1] Y. Lu, L. Zhang and W. Xie, "YOLO-compact: An Efficient YOLO Network for Single Category Real-time Object Detection," 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 2020, pp. 1931-1936, doi: 10.1109/CCDC49329.2020.9164580.

[2] X. Lu, X. Kang, S. Nishide and F. Ren, "Object detection based on SSD-ResNet," 2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS), Singapore, 2019, pp. 89-92, doi: 10.1109/CCIS48116.2019.9073753.

[3] J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[4] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier and S. Lazebnik, "Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models," IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 2641-2649, doi: 10.1109/ICCV.2015.303.

[5] R. Sangpal, T. Gawand, S. Vaykar and N. Madhavi, "JARVIS: An interpretation of AIML with integration of gTTS and Python," 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur,Kerala, India, 2019, pp. 486-489, doi: 10.1109/ICICICT46008.2019.8993344.

[6] M. Khan, S. Chakraborty, R. Astya and S. Khepra, "Face Detection and Recognition Using OpenCV," 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2019, pp. 116-119, doi: 10.1109/ICCCIS48478.2019.8974493.

[7] Q. Zhang and W. Q. Yan, "Currency Detection and Recognition Based on Deep Learning," 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Nov. 2018.

[8] Ning Jiang, Wenxin Yu, Shaopeng Tang and S. Goto, "A cascade detector for rapid face detection," IEEE 7th International Colloquium on Signal Processing and its Applications, Penang, 2011, pp. 155-158, doi: 10.1109/CSPA.2011.5759863.

[9] Y. Wei, X. Zhu, B. Sun and B. Sun, "Comparative studies of AIML," 2016 3rd International Conference on Systems and Informatics (ICSAI), Shanghai, 2016, pp. 344-349, doi: 10.1109/ICSAI.2016.7810979.

[10] V. K. Shukla and A. Verma, "Enhancing LMS Experience through AIML Base and Retrieval Base Chatbot using R Language," 2019 International Conference on Automation, Computational and Technology Management (ICACTM), London, United Kingdom, 2019, pp. 561-567, doi: 10.1109/ICACTM.2019.8776684.

[11] N. N. Khin and K. M. Soe, "University Chatbot using Artificial Intelligence Markup Language," 2020 IEEE Conference on Computer Applications(ICCA), Yangon, Myanmar, 2020, pp. 1-5, doi: 10.1109/ICCA49400.2020.9022814.

[12] Jyoti Aneja, Aditya Deshpande, Alexander G Schwing, "Convolutional image captioning" IEEE Conference on Computer Vision and Pattern Recognition.5561–5570.

[13] Qingz hong Wang and Antoni B Chan. 2018. CNN+ CNN: Convolutional Decoders for Image Captioning. arXiv preprint arXiv:1805.09019

[14] Jiuxiang Gu, Gang Wang, Jianfei Cai, and Tsuhan Chen, "An empirical study of language cnn for image captioning" International Conference on Computer Vision (ICCV).1231–1240, 2017.

[15] Subhashin Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, Raymond Mooney, Trevor Darrell, and Kate Saenko, "Captioning images with diverse objects" IEEE conference on computer vision and pattern recognition.1170–1178, 2017.

[16] Y. Huang, J. Chen, W. Ouyang, W. Wan and Y. Xue, "Image Captioning with End-to-End Attribute Detection and Subsequent Attributes Prediction" IEEE Transactions on Image Processing, vol. 29, pp. 4013-4026, 2020, doi: 10.1109/TIP.2020.2969330.

[17] Abdullah, Ahmed SS, Majida Ali Abed, and Israa Al Barazanchi, "Improving face recognition by elman neural network using curvelet transform and HSI color space." Periodicals of Engineering and Natural Sciences 7.2 (2019): 430-437.

[18] Kumar, Neela Ashish, et al. "Smart Attendance Marking System using Facial Recognition." Research Journal of Science and Technology 11.2 (2019): 101-108.

[19] A. Jaiswal, A. Krishnama Raju and S. Deb, "Facial Emotion Detection Using Deep Learning," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-5, doi: 10.1109/INCET49848.2020.9154121.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016.