ISSN NUMBER (ONLINE) –2368-1209&(PRINT)–2691-5944

Smart Mirror with Artificial Intelligence

Mayukh Mukherjee, Soumik Podder

Abstract--Technology in the 21st century has been steadily revolving around the word "smart". In technological parlance, "smart" indicates any device that would be capable of performing some actions based on decisions of its own, without user intervention. The trend of late, has been to make every item of necessity "smart", with the focus of tightly integrating many such items into a common ecosystem where each device would be able function without much explicit user interaction, which is what the end goal of the field of IoT (Internet of Things) is. Artificial Intelligence (AI) has its basis in gadgets interacting with the environment and users, by detecting certain triggers and learning from those interactions to build up a knowledge model, which would shape its future decisions. It has been under consideration by innovators and technology enthusiasts, to integrate "smart" functions into a mirror, where it would give us information such as the news, time and weather, while also serving its core purpose, which obviously is to show one's reflection. This would, in essence, make the mirror an IoT device. However, it is the introduction of AI driven features within the mirror which would make it qualify as a truly "smart" device.

Index Terms--API; Haar-Classifiers, OpenCV; NLG/NLU; Speech Recognition

I. INTRODUCTION

We were certainly not the first ones to have tried to make the common mirror smart, and we certainly will not be the last to do so. The features incorporated in the smart mirror that would differentiate it from a plain mirror can be classified into two categories: one being the basic features that aim to enhance and enrich the user experience, with the second category being catered towards it's AI based functioning. The heart of the device is a Raspberry Pi, with a computer monitor being its display output. The latter acts like a mirror when set to a lower brightness when a one-way reflective acrylic sheet affixed on the front. The first category of features consists of fetching from the Internet the time, news, weather updates, stock information and so on for display on the mirror dashboard.

Mayukh Mukherjee, Department of Electronics and Communication Engineering, Guru Nanak Institute of Technology, Kolkata, West Bengal-700114 (Email: mayukhmukherjee95@gmail.com)
Dr. Soumik Podder, School of Computer Science & Artificial Intelligence, SR University, Warangal-506371(soumik.podder@sru.edu.in)

The second category leans towards Artificial Intelligence-based activation of the mirror, for providing additional functionality such as voice-based web searches. Facial detection plays the prime role in this sort of functionality. Also, another major feature is the availability of voice detection-based command, which can be used to perform virtually any action that can be done while browsing the Internet.[1]

The core objectives of our project can be summarized as below: The mirror is rather commonplace in our everyday life, and aiming to make it smart is not just a forced necessity but rather a refreshing addition that the majority of people would welcome.Most people spend considerable time on their smartphones immediately after waking up, and eventually still use the mirror while getting themselves ready. We want to incorporate several smartphone capabilities into the mirror, so that individuals can utilize them concurrently as they go about their everyday routine.The mirror is usually a centrally placed object in living rooms and common areas. Allowing the mirror to function as a news bulletin and a wall clock, by providing real-time news headlines and Internet timing from the web, presents an ingenious way of keeping people updated, from a very viewable and easily noticeable area of a room.The Smart Mirror can essentially be transformed into a full-fledged Virtual Assistant, much like Amazon Alexa or Apple's Siri, and can provide companionship to individuals living alone, while also being a source of entertainment in similar situations. [2]

II. METHODOLOGY

The first step is the physical construction of the smart mirror. A four-piece wooden frame is constructed, and the monitor is encased in the frame in a similar fashion to the front-glass in a photo frame. The Raspberry Pi is affixed to the back of the monitor and connected to the monitor using an HDMI cable. The webcam (with inbuilt microphone) is placed on top of the wooden frame, according to the preferred orientation of the monitor, while the speakers are kept concealed in the rear of the monitor to simulate the effect of auditory output coming from inside the mirror.

- Raspberry Pi OS or any other supported OS is installed onto the PI through the NOOBS boot loader or alternatively the OS can be directly flashed onto the SD Card, which serves as the boot device.
- We install Python after booting into the OS, followed by installing PIP, which is a package installer for

https:// doi.org/10.15864/ajac.22014

ISSN NUMBER (ONLINE) –2368-1209&(PRINT)–2691-5944

Python (Chethan K. et al., 2019), through command line.

- Once this is done, we clone the repository of our Smart Mirror code from GitHub, via command line.
- We then run the Smart Mirror code using the command python smart mirror code.py., where "smart mirror code" is the python file name containing the code.

A. Common functionality

As mentioned previously, common functionality includes displaying the news headlines, weather and so on, by utilising various online weather and news services. API tokens from the websites of such services are embedded in the code, providing results similar to what a web search would yield. The fetched data from these services are displayed on the mirror dashboard. We use WeatherAPI.com, which is a free weather and geo-location API provider that provides information including, but not limited to weather forecast, historical weather, future weather, air quality data, etc. etc. Put simply, whatever a browser-based web search result would yield from WeatherAPI.com would be fetched and directly displayed on the Smart Mirror. The process is identical for displaying news headlines, stock market information or whatever other functionality is put in.

B. Artificial Intelligence Implementation

The AI comes into play upon facial detection and/or recognition of voice-based command. The Smart Mirror's webcam actively monitors if any person is to stand and face it. Activation of the smart functions commences with the user facing the monitor leading to successful upright frontal face detection. We use modules from the OpenCV (Open-Source Computer Vision) library, in our python code to perform facial detection. OpenCV is a huge open-source functions library for implementation in computer programming languages, to aid in facial detection, recognition and image processing. It is based on Haar Cascade Classifiers. [3,4]

Haar-Like Features and Haar Cascade Classifiers:

Historically, object detection was performed by computing pixel by pixel RGB values of a test image with sample images to decide what the object within the test image is. This process being computationally expensive, had long called for a more efficient solution. This solution came in the form of a novel object detection framework, now popularly known as Haar Cascade Classifiers, which was first proposed by Paul Viola and Michael Jones in their 2001 research paper, titled Robust real-time face detection. This technique is one of the oldest and possibly the first object detection algorithm for real-time video footage, long antedating the modern deep learning era that was to follow. Haar Cascade Classifiers are based on Haar Features, which represent the pixel intensity patterns observed in various regions of a human face. The three originally proposed features were (a) Edge Features, (b) Line Features and (c) Four Rectangle Features which are shown in Figure 1 below:

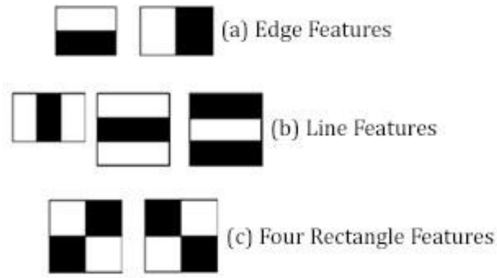

Fig. 1.The original set of Haar Features

Ed The Edge features represent a sudden and sharp shift in pixel intensities between two adjacent regions. In terms of the human face, these represent the regions like the junction of the forehead hairline and the forehead itself, where there exists a large shift in color intensities by virtue of the hair being darker in color as compared to the skin complexion in the forehead region. Line features represent a region of smaller pixel intensity encapsulated between two regions of significantly higher pixel intensities or a lighter region encased between two similarly dark regions. In reference to the human face, line features would translate to the region encompassing the lower and upper lips, where the point of contact of the two lips forms the darker region that is shadowed by the much lighter upper and lower lip surfaces. Finally, the four-rectangle features represent a mesh of diagonally varying intensities, much representative of the jawline and cheekbone areas. Each of the aforementioned Haar Features are constrained in a 24x24 pixels window. The window is made to traverse the entirety of the image, with the image being a grayscale still photo of the live object facing the camera. [3.4]The traversal starts from the top left of the image and with each pass, moves on towards the right, until it reaches the bottom right corner of the image. The traversal flow is demonstrated in the diagram Figure 2.

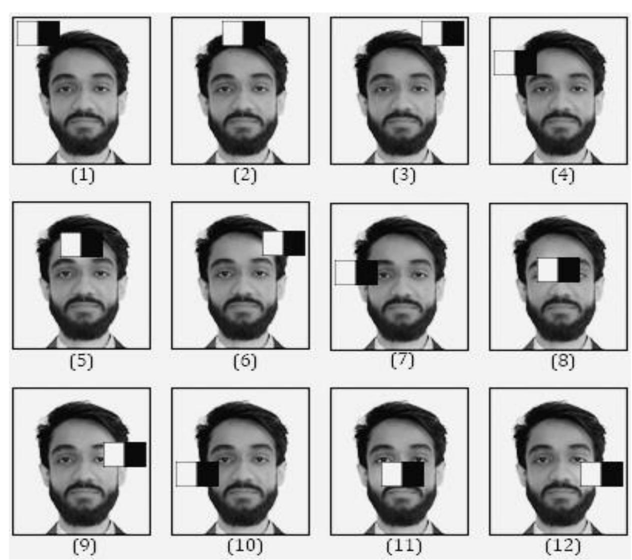

Fig. 2 Traversal of the edge-feature window over the object. Each pass of the traversal is exhibited serially from labels (1) to (12)

https:// doi.org/10.15864/ajac.22014

ISSN NUMBER (ONLINE) –2368-1209&(PRINT)–2691-5944

The calculation performed on each traversal is the difference between the total sum of the pixel intensities of all pixels superimposed by the black part of the detection window and the total sum of the pixel intensities of all pixels lying below the white part of the detection window. If the difference is close to a specific learned threshold, an edge is perceived on that portion of the image. This calculation can be demonstrated with a simpler example. Consider Figure 3 below which shows a gradient, which, say, is a part of an image. We want to detect whether an edge exists in the said region.

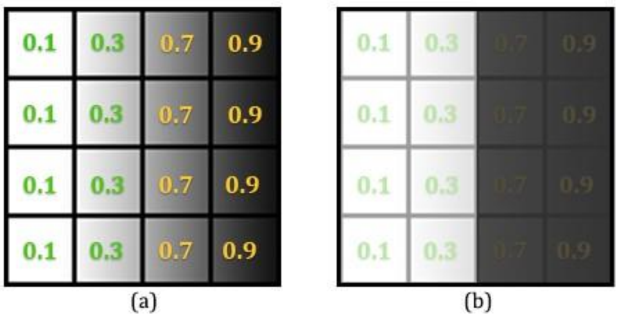

Fig. 3 Here box (a) shows a sample region representing an edge. Box (b) shows the edge feature detection window superimposed on the sample region. Each of the pixel values in green is considered to lie in the white region, and the pixel values in orange in the black region.

The calculation involves subtracting the sum of the pixels lying in the brighter region from the sum of pixels in the darker region and for Figure 3 above, is equal to:

$$(\mathbf{0.7} + \mathbf{0.7} = \mathbf{0.7} + \mathbf{0.7} + \mathbf{0.9} + \mathbf{0.9} + \mathbf{0.9} + \mathbf{0.9}) - (\mathbf{0.1} + \mathbf{0.1} + \mathbf{0.1} + \mathbf{0.1} + \mathbf{0.3} + \mathbf{0.3} + \mathbf{0.3} + \mathbf{0.3}) = \mathbf{4.8}$$

If the expected threshold is between, say, 4.5 to 5.0, a result of 4.8 is considered positive and an edge is assumed to exist. Similarly, for the line feature, the calculation involves the subtraction of the total sum of pixel intensities in the two dark regions from the pixel intensities under the white region sandwiched in between. Nonetheless, such a method of calculation would result in a huge number of mathematical operations in a realistic situation, where one run of a single Haar Feature window over the entire image area would involve thousands of traversals. This issue is addressed via the introduction of Integral Images, where the value of the pixel intensity of any pixel within the integral image is equal to the sum of pixel intensities of all pixels lying on top and at left of the corresponding pixel in the original image (Paul Viola et al., 2020). The elucidate this concept, consider the example in Figure 4:

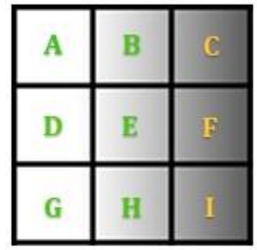

Fig. 4 Portion of a sample image containing a total of 9 pixels labeled A to I.

The pixel in the integral image corresponding to the pixel with intensity "E", will have a pixel intensity value equivalent to A+B+C+D, i.e., the pixel intensities to the left and above of pixel "E" plus the pixel intensity of "E" itself. For pixel "H", the corresponding integral image's pixel will have intensity equivalent to A+B+D+E+G+H. The computation of an integral from the sample object followed by the Haar-Feature traversal, reduces the number of addition operations required, by almost 1/4th of the original technique.

However, unfortunately, the three Haar features mentioned previously are not sufficient to perform detection with high rates of accuracy and as such, a whopping 1,60,000 similar features were initially proposed, with the ideal target being the traversal of all the 1,60,000 features across the entire integral image, one feature at a time. This dilemma is remediated by AdaBoost, short for Adaptive Boosting, an Ensemble learning method that has its roots in Machine Learning. AdaBoost is fed with a gargantuan amount of training data, comprising of positive images (images of faces) and negative images (images without faces) from which it separates the positive images according to the results obtained from running the complete set of Haar feature calculations on the training images. The results are tallied with a predefined error rate before segregating the positive and negative images and the process is repeated with new error rates being computed at every run. This results in a feature reduction to almost 6000 features. The functioning of AdaBoost in this case is highly dependent on the type of training images fed to it and is expected to vary according to the ethnic and visual variation of the people within the training images. Another improvement was in the form of the introduction of a cascading technique, where, instead of applying all the 6,000 features, the features were divided into groups, referred to as stages. These stages are applied one after the other and if a particular region of an image qualifies as negative for multiple successive stages, the entire region is discarded as a non-object (rest of the features are not run on it). [5]

There exists a training method, as well as pre-trained models in OpenCV. We use the latter, which are available as .xml files in the OpenCV installation folder. The first step is the creation of a CV::Cascade Classifier function, followed by the loading of the necessary XML files using the cv::Cascade Classifier::load method. Subsequently, the detection is done using the cv::Cascade Classifier::detect Multi Scale method, which returns boundary rectangles for the detected faces or eyes, conforming facial detection

ISSN NUMBER (ONLINE) –2368-1209&(PRINT)–2691-5944

III. Working

The smart mirror will invariably display information viz. time, date, weather, news, etc. on its dashboard, with an active internet connection. Activation of AI functionality of the smart mirror can be either through voice command or through facial recognition. Upon utterance of any of the launch phrases of the smart mirror such as "Hey mirror!" or "Mirror, mirror, on the wall!" (We have included many popular launch phrases in the code), within the pick-up range of the microphone, the AI functionality is activated, causing the mirror to produce a human-like audio output from the speakers, asking the user what it needs to do. An example of the audio output is: "Hey Mayukh! How may I help you?". From this point onwards, further interaction can be continued using further voice commands, such as asking the mirror phrases like "show me a map of India" and so on. The verbal launch phrase can be used while being physically away from the mirror, as long as the microphone picks up our voice. The second mode of activation relies on facial detection to activate the mirror, which will then actively listen for any verbal command. [5,6]
The flow of the working process of the smart mirror can be laid out as demonstrated in Figure 5.

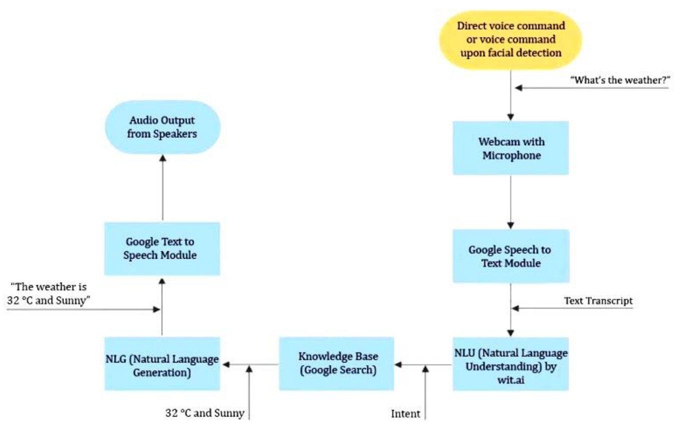

Fig. 5 A flowchart demonstrating the working of the AI in the Smart Mirror.

Explanation of the flowchart:

AI can be implemented by introducing a simple webcam (with microphone) and a speaker to the mirror.

- Upon successful objection detection on the webcam, the mirror is activated and will listen for voice commands.
- Google Speech to Text API will be implemented in the code which will help in Speech Recognition to give us the audio transcript of the input voice (Deep Mehta et al., 2021).
- This audio transcript is fed to an online open-source Natural Language Understanding (NLU) platform provided by wit.ai
- The NLU gives the actual intent of the audio transcript, which is then fed to a knowledge base, which in our case is a simple Google Search.
- The Google Search would give us a corresponding search result in the form of a textual output based on the Intent of the Audio Transcript.
- This textual output is fed to a Natural Language Generator (NLG), which performs essentially the opposite function of an NLU, giving us a full-fledged human-like textual output.
- This full textual output is fed to Google's Text to Speech Module yielding an audio version of the text, which is then played on the Speakers.[5,6]

IV. Future Scope

While IoT has almost reached its peak, AI is still a developing field. There is much to be understood and done, as far AI in technology is concerned. At present AI is not fully self-sufficient in that, there still exists some degree of human intervention in the form of having to make the devices learn through supervised/unsupervised learning. Also, there is a certain saturation point to which devices can be trained through learning data as there will always exists innumerable situations where the trained data would not suffice for correct decision making. Such is the case even for our smart mirror, where there may arise situations where the mirror fails to interpret the "intent" of the user. It is here where most of the scope for future work lies. Future work on the project is intended to be focused on the following:

- Provide the smart mirror with more and more training data, thereby enabling it to accurately and precisely perform intended actions in any number of unique user inputs.
- The current mirror is based on facial detection, wherein upon detection of any human face the mirror can be interacted with. We intend to incorporate facial recognition, wherein the mirror can be activated only by a valid group of users with whom a facial match is found.
- Incorporate facial expression detection, where landmark features (Nirmala H. *et al*., 2020) of the face are used to determine the mood of the person and provide personalized suggestions based on the same.
- Integrate other IoT devices, such as room lights, fans & television with the mirror, enabling the mirror to act as a centralized control dashboard, from where all the appliances can be controlled through voice commands.

V. CONCLUSION

In the present age of the Internet, we aim to integrate technology into the simplest of objects that come to our day to day to use, with the intention of enriching our usage experience while also increasing functionality of the object. We performed the same with our Smart Mirror, where we incorporated technology into one of the simplest household

ISSN NUMBER (ONLINE) –2368-1209&(PRINT)–2691-5944

objects, that has been of use to humans since time immemorial. The smart mirror provides a lot of functions that other devices can provide, all in one place, all accessible while utilising the mirror for general tasks. Implementation of AI further expands its functionality creating the pathway for future extensibility in its features, with suitable groundwork.

VI. References

[1] D. Mehta, M.Jain ,P. Thakkar.Personalized Smart Mirror with User Detection. International Research Journal of Engineering and Technology (IRJET), Volume: 08 Issue: 03., Mar. 2021DOI:https://www.irjet.net/archives/V8/i3/IRJET-V8I3476.pdf

[2]N..H,ArjunPrakashRajaput, R. V. Topannavar,S. R G.An artificially intelligent iot based smart mirror. International Journal of Scientific Research in Engineering and Management (IJSREM), Volume: 04 Issue: 09,Sep 2020. DOI:https://ijsrem.com/volume-04-issue-09-september-2020/

[3]C. K, Adnan Ahmed, .N.Ganapathy, P. N Simha, S. Kothari. REFLECTA - Artificial Intelligence Based Smart Mirror. International Journal of Innovative Technology and Exploring Engineering (IJITEE). Volume-8, Issue-6S4, Apr 2019.
https://www.ijitee.org/wp-content/uploads/papers/v8i6s4/F10970486S419.pdf

[4]J. Zeng, C.Li ,L.-Jie Zhang. A Face Recognition System Based on Cloud Computing and AI Edge for IOT. Edge Computing – EDGE 2018, Lecture Notes in Computer Science,vol 10973. 2018. Springer,.https://doi.org/10.1007/978-3-319-94340-4_7

[5]P.Viola , M. J. Jones, Robust real-time face detection. International Journal of Computer Vision, 57(2):137–154, May 2004. https://doi.org/10.1023/B:VISI.0000013087.49260.fb

[6]Intel Corporation. Cascade Classifier Tutorial. OpenCV Modules. https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

VII. Biographies

Mayukh Mukherjee was born in Dumdum, Kolkata and subsequently raised in the town of Barrackpore, Kolkata. He completed his schooling from Modern English Academy, Barrackpore and then went on to pursue a diploma in Electronics and Telecommunication Engineering from JIS School of Polytechnic, Kalyani, Nadia in the year 2017. He completed his graduation in Electronics and Communication Engineering(Bachelor of Technology) from Guru Nanak Institute of Technology, Kolkata in the year 2023 and is currently employed at Capgemini Technology Services India Limited as an Analyst. His study interests include the fields of Artificial Intelligence and Machine learning and their implementation in modern data science and analytics. He is also a keen follower and avid enthusiast of evolving computing technologies and advances in modern computing hardware.

Dr. Soumik Podder is Assistant Professor of School of Computer Science and Artificial Intelligence of SR University, Warangal, Telengana State, India. He was previously working as Assistant Professor in Electronics and Communication Engineering of Guru Nanak Institute of Technology, Kolkata and Department of Electronics and Telecommunication Engineering, C V Raman Global University, Odisha. He obtained his PhD and M.Tech degree in Nanotechnology from Jadavpur University. He obtained his B.Tech in Electronics and Telecommunication Engineering from University of Kalyani, West Bengal. He is Gold Medalist in M.Tech and awarded DST-INSPIRE Fellowship from DST, Govt. of India for his doctoral studies. He was also recipient of Newton-Bhaba Fellowship during his PhD. He has teaching experience of 6 years in reputed engineering colleges in India. His research interests are Nanobiotechnology, Nanoelectronics, Biosensor, Artificial Intelligence, IoT and Block Chain. He has published more than 20 research papers in highly reputed internal journals and conferences such as American Chemical Society, Royal Society of Chemistry, IEEE etc. He is regular reviewer of RSC Advances, Scientific Report (Springer Nature), Springer Discover Nano, RSC Materials Chemistry B, Taylor's & Francis, Wiley etc. He is premium member of American Chemical Society and IEEE, IEEE Electron Device Society, IEEE Photonics Society, American Physical Society , Science AAAS and FOSET.. He is elected as Editor in IIP Proceedings. He is also certified reviewer of ACS journals.