Revolutionizing Creativity and Communication: Introducing Air Canvas

Souryadip Ghosh[1], Tanishka Chakraborty [2], Dhrubajyoti Ghosh[3], Anita Pal[4]

[1]Student, Dept. of CSE, OmDayal Group of Institutions, West Bengal, India
[2]Student, Dept. of CSE, OmDayal Group of Institutions, West Bengal, India
[3]Assistant Professor, Dept. of CSE, OmDayal Group of Institutions, West Bengal, India
[3]Associate Professor, Dept. of Mathematics, National Institute of Technology, Durgapur, India

---***---

Abstract - *In today's ever-evolving landscape of human-computer interaction, there is an increasingly pressing need for innovative interfaces that can seamlessly blend the digital and physical worlds. This need stems from the growing demand for more intuitive, immersive, and engaging ways to interact with technology. Traditional input methods, such as keyboards and mice, while reliable, often fall short in meeting the expectations of modern users who crave more natural and expressive means of engagement. The demand for innovative interfaces has escalated, giving rise to the significance of the Air Canvas work. This work involves cutting-edge endeavor seamlessly fuses the capabilities of OpenCV and MediaPipe to craft an interactive drawing platform steps in to monitor real-time hand movements with precision. Within the realm of this virtual canvas, users are afforded the remarkable ability to paint in mid-air, with their hand gestures translated into vibrant strokes on a digital canvas. This work underscores the potential of computer vision and gesture recognition for immersive, creative applications that bridge the gap between technology and art.*

Key Words: Machine learning, MediaPipe, Hand Gesture, Hand Tracking, nonrigid motion analysis, human-computer interaction

1.INTRODUCTION

In today's era of technological innovation, the Air Canvas work stands at the intersection of art, computer vision, and human-computer interaction. It harnesses the power of Python, OpenCV, MediaPipe, and NumPy to create a captivating and interactive digital canvas that allows users to paint in the air using hand gestures. Python serves as the project's foundation, known for its versatility and extensive libraries. When combined with machine learning and computer vision libraries such as OpenCV and MediaPipe, Python empowers us to explore new horizons in creative expression and user interaction. The central theme of the Air Canvas work revolves around Hand Gesture Recognition and Hand Tracking. MediaPipe, a versatile framework, offers pre-trained models that excel at hand tracking. By integrating this technology, we can precisely detect and track hand movements in real-time, creating a bridge between the physical and digital worlds. The canvas is no longer confined to a tangible surface; it's now a virtual playground where users can unleash their creativity by their hands. This work interprets these hand movements as brush strokes on a digital canvas, offering a dynamic and colourful painting experience. Beyond the technical marvel, this work redefines how we engage with technology. It demonstrates how natural human gestures can be seamlessly integrated into digital environments, offering a unique and immersive way to express artistic ideas. This concept transcends conventional boundaries, inspiring creativity and showcasing the potential of technology to bring innovation and joy to our lives. However, it has only been in recent years that there has been an increased interest in trying to introduce other human-to- human communication modalities into HCI. This includes a class of techniques based on the movement of the human arm and hand, or hand gestures[1]. In this project, we will delve into the intricacies of Python, OpenCV, MediaPipe, and NumPy to create an Air Canvas that not only showcases the technical prowess of these tools but also highlights the incredible possibilities at the intersection of art and technology. It's an exploration of how a simple hand gesture can transform into a stroke of digital art, paving the way for future innovations in human-computer interaction.

This paper organised in seven sections. Section one contains introduction. Section two describes the literature survey. Section three represents the basic preliminaries for this work. Section four gives the methodology. Section five describes implementations. Section six includes observations. Finally, Section seven draws conclusions based on our study.

2. LITERATURE REVIEW

This literature review encompasses a diverse array of seminal papers in the field of gesture recognition, computer vision, and human-computer interaction. In their 1997 review, "Visual Interpretation of Hand Gestures for Human-Computer Interaction," Pavlovic, Sharma, and Huang [1] offer a comprehensive overview of gesture recognition methodologies. Cipolla and Hollinghurst's "Human-Robot Interface by Pointing With Uncalibrated Stereo Vision" (1996) [2] explores uncalibrated stereo vision for human-robot communication via pointing gestures. "Active Shape Models—Their Training and Application" (1995) [3] by Cootes et al. introduces Active Shape Models (ASMs) for object recognition. Fukumoto, Suenaga, and Mase's "Finger-Pointer" (1994) [4] investigates image processing techniques for recognizing pointing gestures. Ju, Black, and oob's "Cardboard People" (1996) [5] presents a parameterized model for articulated image motion, potentially applicable to gesture modeling. Lanitis et al. (1995) [6] discuss automatic interpretation of human faces and hand gestures using flexible models. Huang et al. (2016) [7] introduce a pointing gesture-based egocentric interaction system. Betancourt et al. (2015) [8] focus on filtering SVM frame-by-frame binary classification in a detection framework. Ren et al. (2015) [9] propose Faster R-CNN for real-time object detection with region proposal networks. In 2015, Kervrann and Heitz[10] discuss learning structure and deformation modes of nonrigid objects in long image sequences. Lastly, "Deepfinger" (2015) by Huang et al. [11] presents a cascade convolutional neuron network approach to finger key point detection in egocentric vision with a mobile camera. These foundational works collectively contribute to the evolving landscape of gesture-based human-computer interaction.

3. PRELIMINARIES

The MediaPipe framework handles hand gesture identification and tracking, and the OpenCV library handles computer vision. The application uses machine learning principles to track and identify hand movements and hand tips.

3.1 MediaPipe

The user experience can be significantly enhanced acrossa range of technological domains and platforms by being able to recognise the shape and motion of hands. For instance, it can provide as the foundation for hand gesture control and sign language comprehension. It can also makeit possible for digital information and material to be superimposed on top of the real world in augmented reality. Although it comes effortlessly to individuals, robust real- time hand perception is an extremely difficult computer vision problem due to the fact that hands frequently occludethemselves or each other and lack high contrast patterns. A high-fidelity hand and finger tracking solution is MediaPipe Hands. It uses machine learning (ML) to deduce 21 3D hand landmarks from a single image. Our solution delivers real- time performance on a cell phone, and even scales to several hands, unlike existing state -of-the-art systems, which mostly rely on powerful desktop environments for inference. We anticipate that making this hand perception functionality, will lead to the creation of innovative use cases, igniting new research directions. MediaPipe is a powerful framework for building various AI and computer vision applications, including the "Air Canvas" project. In this context, MediaPipe offers a pre- built solution for hand tracking, which simplifies the implementation significantly. Applications that are concerned with deictic gestures usually use only a single (index) fingertip and some other reference point on the hand or body [4]. MediaPipe provides real-time hand tracking and gesture recognition capabilities, allowing the system to precisely locate and track the user's hand movements. This information can be seamlessly integrated into the canvas interaction logic, enabling users to draw or interact with the canvas using hand gestures. Motion cue is also commonly applied for gesturer localization and is used in conjunction with certain assumptions about the gesturer. For example, in the HCI context, it is usually the case that only one person gestures at any given time. Moreover, the gesturer is usually stationary with respect to the (also stationary) background. Hence, the main component of motion in the visual image is usually the motion of the arm/hand of the gesturer and can thus be used to localize her/him. The disadvantage of the motion cue approach is in its assumptions. While the assumptions hold over a wide spectrum of cases, there are occasions when more than one gesturer is active at a time (active role transition periods) or the background is not stationary. By leveraging MediaPipe, the Air Canvas work benefits from a robust and efficient hand tracking solution, reducing development time and complexity while enhancing the user's ability to interact naturally with the virtual canvas.

3.2 OpenCv

Object detection image processing methods are included in the OpenCV computer vision library. Real-time computer vision applications can be created by utilising the OpenCV library for the Python programming language. The processing of images and videos as well as analytical techniques like face and object detection use the OpenCV library. OpenCV (Open Source Computer Vision Library) is a cornerstone of the Air Canvas work. Its image processing and computer vision capabilities are essential for real-time hand tracking and canvas interaction. OpenCV enables tasks like hand detection, gesture recognition, and tracking hand movements. It aids in drawing on the canvas and providing visual feedback. Additionally, it can handle video input from cameras, making it suitable for interactive applications. OpenCV's robustness and versatility are invaluable in creating a seamless and responsive Air Canvas experience, enhancing user engagement and interaction with the virtual canvas.

3.3 NumPy

The N-dimensional array type known as ndarray is the most significant object defined in NumPy. The collection of identically categorised things is described. A zero-based index can be used to access items in the collection. A ndarray's items all take up the same amount of space as a memory block. Every item in ndarray is a data-type object (called dtype). A Python object of one of the array scalar type s represents each item that is retrieved from an ndarray object (via slicing). NumPy plays a vital role in a Air Canvas work, enabling efficient image processing and data manipulation. It allows for tasks like image loading, resizing, and canvas creation. NumPy's array operations facilitate drawing and real-time updates on the canvas. It's crucial for processing hand-tracking data, removing noise, and smoothing movements. Integration with other libraries like

OpenCV or machine learning tools can enhance functionality. NumPy's versatility and speed make it an indispensable tool for creating a responsive and interactive Air Canvas system.

3.2 Hand Landmark Model

After detecting the palm over the whole image, our subsequent hand landmark model uses regression, or direct coordinate prediction, to accomplish precise keypoint localization of 21 3D hand -knuckle coordinates inside the detected hand regions. The model acquires a reliable internal hand posture representation and is unaffected by self-occlusions or partially visible hands. We manually added 21 3D coordinates to around 30K real-world photos to obtain ground truth data, as shown below (we take Z- value from image depth map, if it exists per corresponding coordinate). We additionally render a high-quality synthetic hand model over a variety of backgrounds and map it to the associated 3D coordinates in order to better cover the range of possible hand poses and provide additional supervision on the nature of hand geometry. MediaPipe's hand landmark model is at the core of the Interactive Air Canvas work, enabling accurate and real- time hand tracking. This model identifies and tracks 21 key points on the user's hand, including fingertips, knuckles, and palm landmarks. These hand landmarks serve as critical spatial references, allowing for precise capture of the hand's position and movement in three-dimensional space.The hand landmark model employs deep learning techniques, particularly convolutional neural networks (CNNs), to localize these landmarks in real-time video frames. This approach ensures that the system can accurately and rapidly track the hand's movement, making it responsive and natural for users. Canvas interaction heavily relies on these hand landmarks. By mapping the coordinates of these landmarks to the canvas space, users can draw lines, shapes, and images with their hand movements. This mapping process, combined with NumPy and OpenCV for efficient data manipulation and canvas rendering, results in a seamless and interactive canvas experience. While the utilization of hand landmarks greatly enhances the project's usability, it also comes with challenges, such as handling occlusion, adapting to varying lighting conditions, and optimizing real-time performance. These challenges were addressed through a combination of software enhancements and hardware considerations to ensure the system's reliability and robustness in various usage scenarios. This category encompasses a diverse range of models, some of which are built upon deformable 2D templates representing human hands, arms, or even the entire body [2], [3], [5], [6], [7]. These deformable 2D templates consist of sets of points that define the object's outline and are utilized as interpolation nodes to approximate the object's contour. The primary interpolation function employed is typically a piecewise linear function. These templates are composed of three key components: average point sets, point variability parameters, and external deformations. The average point sets define the "typical" shape within a specific group of shapes, while point variability parameters describe the permissible variations or deformations within the same group of shapes. Both of these parameter types are commonly referred to as internal parameters. For instance, when considering the human hand in an open position, there exists an average shape that represents this configuration, and all other instances of an open hand posture can be generated by making slight adjustments to this average shape. Internal parameters are derived through principal component analysis (PCA) applied to a substantial set of training data.

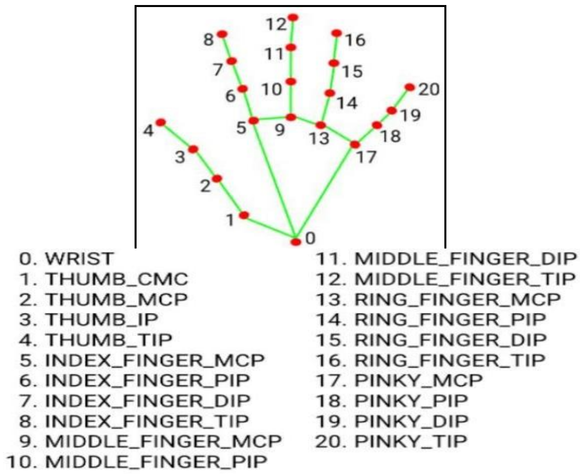

Fig -1: Hand Landmark Model

4. METHODOLOGY

Based on the web camera frames that were captured, a virtual paint programme was offered. The web camera sends the system the frames that it has received. Until the application is finished, the approach uses a web camera to collect each frame. This leveraged the Mediapipe framework to develop a hand landmark estimation model. This model was trained using the prepared dataset, incorporating convolutional neural networks (CNNs) for real-time landmark detection. To enable real-time hand tracking and detection, we implemented OpenCV to capture video frames from a webcam or camera feed. The trained MediaPipe model was used to detect and track hand landmarks in each frame. For canvas creation and interaction, we created a virtual canvas using NumPy, defining its size and initial settings. We mapped the coordinates of hand landmarks to the canvas space, allowing users to draw lines, shapes, and images through hand movements. To enhance the user experience, we developed a graphical user interface (GUI) to display the canvas and provide user controls. We also implemented visual feedback mechanisms. Throughout the development process, we optimized the code for real-time performance, considering factors like latency and frame rate. Rigorous testing and debugging were conducted to ensure robustness, addressing issues such as occlusion and lighting variations.

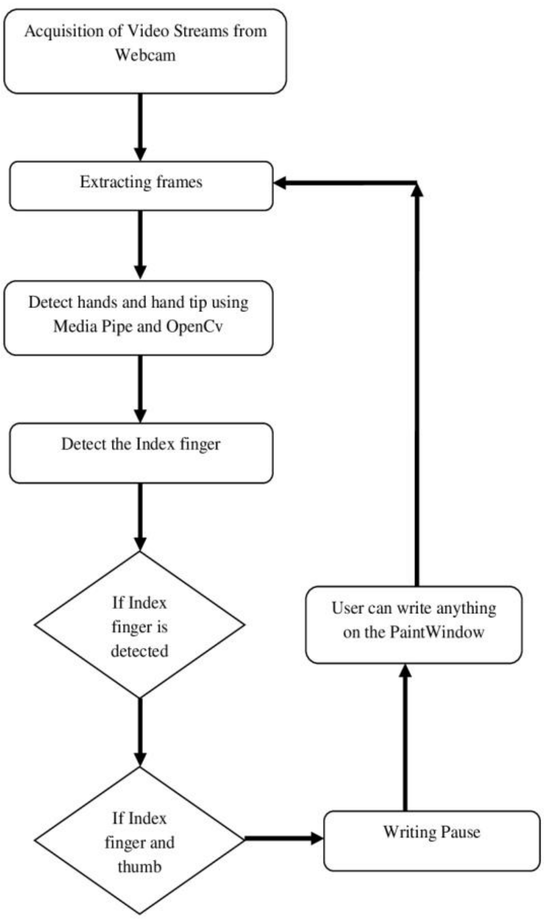

Fig -2: Flow chart of air canvas application

5. IMPLEMENTATION

5.1 Detection of Index Finger

The detection of the index finger using MediaPipe relies on its real-time hand tracking pipeline. MediaPipe employs a deep neural network-based model to perform key point estimation, precisely locating 21 hand landmarks, including fingertips. These landmarks are detected by processing the input image or video frame with a multi-stage convolutional neural network (CNN), a two CNN-based stages hand and fingertip detection framework is proposed but it is not good enough for real-world applications. Currently, in this field, we need a large benchmark dataset to train and evaluate the performance on hand detection and fingertip detection[8]. The network has been trained on a vast dataset to recognize and locate hand key points accurately. The index finger's tip corresponds to one of these landmarks, and its position is extracted from the model's output, allowing for real-time tracking and interaction with the index finger's movement in applications like gesture recognition, virtual painting, and more. In [9], the detector only reports the existence of the hand without its location. Faster R-CNN [10] is the most recent general object detection algorithm with good performance. A two stages CNN-based hand and fingertip detection framework is proposed in [11]. By extracting the position of this specific landmark from the model's output, applications can accurately track the index finger's movement and orientation. This tracking capability opens up a wide range of possibilities, from creating virtual painting interfaces to enabling gesture-based interactions in various digital applications, making it a powerful and versatile concept in the realm of human-computer interaction and computer vision.

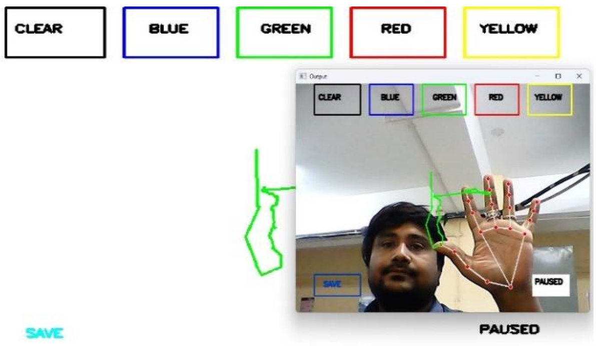

Fig -3: Detection of Index Finger

5.2 Save File

Incorporating the ability to save and pause the paint window adds valuable functionality to our digital canvas. Users can pause their work at any point, preserving their creative progress for future reference. This feature not only facilitates convenience but also encourages experimentation and creativity without time constraints. Additionally, the option to save artwork allows users to keep a digital record of their creations, fostering a sense of accomplishment and providing a means for sharing or editing their work at a later time. Together, these capabilities enhance the overall user experience and make the Air Canvas work a versatile and user-friendly artistic tool.

```
current_time = datetime.datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
        image_filename = f"painting_{current_time}.png"
        cv2.imwrite(image_filename, paintWindow)
        print(f"Saved image as {image_filename}")
```

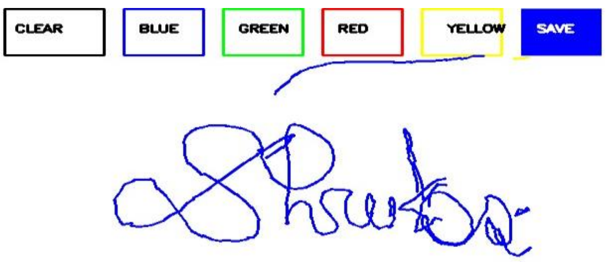

Fig -4: Saved Picture as PNG file and file name as per our saved date

Building on the user-centric design, the GUI also includes a save button on the frame screen. Clicking this button enables users to save their artwork or content created on the virtual canvas. Furthermore, users can utilize gesture-based save functionality by bringing their index finger and thumb together while forming save "S". This intuitive gesture recognition simplifies the saving process, allowing users to preserve their work with a natural hand movement.

5.3 Pause PaintWindow

This work is an innovative digital platform that redefines creativity and user interaction. Leveraging cutting-edge technologies, including real-time hand tracking and drawing capabilities, this project offers users a unique and immersive canvas experience.

Users can express their creativity by simply moving their hands in the air, thanks to advanced hand tracking powered by the MediaPipe library. The system precisely captures hand movements and translates them into digital strokes, allowing for the creation of stunning artwork, dynamic presentations, or engaging educational content. With a user-friendly graphical interface and the flexibility to pause the canvas flow with a click or by typing "p," the Interactive Air Canvas ensures an intuitive and enjoyable experience. Whether you're an artist, educator, or professional, this wok opens up a world of possibilities for interactive digital expression. Designed with performance and user feedback in mind, this work combines the power of NumPy, OpenCV, and MediaPipe to provide a responsive and seamless canvas interaction platform. Explore your creativity in the digital realm with the "Interactive Air Canvas" and revolutionize the way you engage with your imagination.

Fig -5: Index finger and thumb Unite them to make Pause

This work incorporated an intuitive pause feature that allows users to temporarily halt the continuous flow of writing by bringing their index finger and thumb together. This innovative functionality is achieved through the precise tracking of hand landmarks. The system continuously monitors the positions of hand landmarks, particularly those corresponding to the index finger and thumb. When these landmarks converge or come into close proximity, the system detects this gesture. This detection serves as a trigger for the pause action. Upon recognizing the gesture, the system responds by suspending the drawing process, giving users a moment to reposition or take a break. This dynamic pause mechanism not only adds a natural and interactive element to the canvas but also enhances the user experience by seamlessly aligning with users' hand movements and intentions. This feature exemplifies how hand landmarks, precisely tracked through the MediaPipe framework, can be leveraged to create responsive and user-centric interactions within the Interactive Air Canvas, making the creative process more fluid and intuitive for users.

6. OBSERVATIONS

The Interactive Air Canvas work delivers an engaging and innovative platform for creative expression through hand tracking and virtual painting.

6.1 Accurate Hand Tracking

This work employs the precise hand tracking capabilities of the MediaPipe library. It captures and follows 21 hand landmarks, including fingertips and knuckles, enabling a responsive and accurate tracking of hand movements.

6.2 Gesture-Based Pause

A standout feature is the gesture-based pause functionality. Users can effortlessly pause the drawing process by bringing their index finger and thumb together. This intuitive gesture recognition offers users the flexibility to take breaks or make adjustments without the need for physical buttons.

6.3 Real-Time Canvas Interaction

The PaintWindow provides users with a virtual canvas to draw or write in the air. Hand movements are translated into digital strokes on the canvas in real-time, delivering a fluid and immersive interaction experience.

6.4 User-Friendly Interface

The graphical user interface (GUI) enhances usability, making it accessible to users of all levels. The GUI includes a pause button on the frame screen, ensuring easy access to the pause feature. Additionally, users can manually pause by typing the "p" key, adding an extra layer of control.

6.5 Seamless Integration

The integration of NumPy, OpenCV, and MediaPipe ensures a seamless and efficient system. NumPy and OpenCV facilitate canvas creation, rendering, and image processing, while MediaPipe handles the hand tracking with remarkable accuracy.

6.6 Creative Freedom

The work empowers users to unleash their creativity. Whether for artistic expression, educational purposes, or professional presentations, the "Interactive Air Canvas" offers a versatile and engaging platform for digital creation.

Overall, this work represents a successful fusion of technology and creativity, providing users with a dynamic canvas interaction experience that is both user-friendly and feature-rich.

6. CONCLUSION

In conclusion, the Air Canvas work showcases the convergence of OpenCV and MediaPipe to deliver an interactive and artistic experience. It highlights the potential of computer vision and gesture recognition technologies in creating engaging applications that transcend traditional boundaries. By enabling users to paint in the air and see their creations come to life on a digital canvas, this project not only sparks creativity but also demonstrates the exciting possibilities at the intersection of technology and art. It underscores how also demonstrates the exciting possibilities at the intersection of technology and art. It underscores how innovative solutions can bring us closer to a future where human-machine interactions are both intuitive and inspiring. This work represents an innovative leap in how we interface with technology. Hand gestures, a natural form of expression, are harnessed to create digital art. The impact is profound, reaching artists, educators, individuals with disabilities, and anyone seeking an unconventional yet engaging means of artistic expression. The motivation behind the project lies in the quest to push the boundaries of creativity and user interaction. Traditional art tools are static and require physical contact with surfaces, limiting the expressive freedom of artists. By harnessing the capabilities of Python, OpenCV, MediaPipe, and NumPy, we aim to empower individuals to create art in a more intuitive and immersive manner. The Air Canvas work presents a unique set of technical challenges. Precise hand tracking and gesture recognition demand intricate computer vision algorithms, which OpenCV and MediaPipe provide. Ensuring a seamless and real-time drawing experience while managing hardware limitations is also a significant technical feat that we'll tackle.

REFERENCES

[1] D. Kornack and P. Rakic, “Cell Proliferation without Neurogenesis in Adult Primate Neocortex,” Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.

[2] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[3] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] V. I. Pavlovic, Rajeev Sharma, T. S. Huang, Fellow, IEEE “Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review” vol. 19, no. 7, 1997

[6] R. Cipolla and N.J. Hollinghurst, “Human-Robot Interface by Pointing With Uncalibrated Stereo Vision,” Image and Vision Computing, vol. 14, pp. 171-178, Mar. 1996

[7] T. F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, “Active Shape Models—Their Training and Application,” Computer Vision and Image Understanding, vol. 61, pp. 38-59, 1995.

[8] S.X. Ju, M.J. Black, and Y.Y. oob, “Cardboard People: A Parameterized Model of Articulated Image Motion,” Proc. Int’l Conf. Automatic Face and Gesture Recognition, Killington., pp. 38-43,1996

[9] C. Kervrann and F. Heitz, “Learning Structure and Deformation Modes of Nonrigid Objects in Long Image Sequences,” Proc. Int’l Workshop on Automatic Face and Gesture Recognition, 1995

[10] A. Lanitis, C.J. Taylor, T.F. Cootes, and T. Ahmed, “Automatic Interpretation of Human Faces and Hand Gestures Using Flexible Models,” Proc. Int’l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp. 98-103, 1995.

[11] Y. Huang, X. Liu, X. Zhang, and L. Jin, "A Pointing Gesture Based Egocentric Interaction System: Dataset, Approach, and Application," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, pp. 370-377, 2016.

[12] A. Betancourt, P. Morerio, L. Marcenaro, M. Rauterberg, and C. Regazzoni. Filtering svm frame-by-frame binary classification in a detection framework. In Image Processing (ICIP), 2015 IEEE International Conference on, pp. 2552–2556, 2015

[13] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Neural Information Processing Systems (NIPS), 2015.

[14] Y. Huang, X. Liu, X. Zhang, and L. Jin. Deepfinger: A cascade convolutional neuron network approach to finger key point detection in egocentric vision with mobile camera. In The IEEE Conference on System, Man and Cybernetics (SMC), pp. 2944–2949, 2015.

BIOGRAPHIES

Souryadip Ghosh,
Student of Computer Science & Engineering Department,
OmDayal Group of Institutions,
Maulana Abul Kalam Azad University of Technology, West Bengal, India

Tanishka Chakraborty
Student of Computer Science & Engineering Department,
OmDayal Group of Institutions,
Maulana Abul Kalam Azad University of Technology, West Bengal, India

Dr. Dhrubajyoti Ghosh,
Assistant Professor,
Computer Science & Engineering Department,
OmDayal Group of Institutions,
Maulana Abul Kalam Azad University of Technology, West Bengal, India

Dr. Anita Pal,
Associate Professor,
Mathematics Department,
National Institute of Technology,
Durgapur, India.