**Smart Teaching System**

**GITHUB: https://github.com/AkshatSingal/Smart-Teaching-System.git**

**SEMINAR REPORT**

*Submitted in partial fulfillment of the requirements for the degree of*

**Bachelor of Technology**

*in*

**Computer Science Engineering and Information Technology**

*by*

**Amiya Trehan (20csu137)**

**Akshat Singal (20csu006)**

**Under the guidance of**

**Dr. Swati Gupta**

Department of Computer Science and Engineering

School of Engineering and Technology

The NorthCap University, Gurugram- 122001, India

Session 2024-25

**DECLARATION**

I hereby declare that the thesis entitled “Smart Teaching System" submitted by me, for the award of the degree of Bachelor of Technology in Computer Science Engineering and Information Technology to NCU is a record of bonafide work carried out by me under the guidance of Dr. Swati Gupta.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Haryana

**Signature of the Candidate**

**CERTIFICATE**

**This is to certify that the Project Report entitled, “Smart Teaching System” submitted by Amiya Trehan (20csu137) and Akshat Singal (20csu006)to The NorthCap University, Gurgaon, India, is a record of bonafide Project work carried out by him/her under my/our supervision and guidance and is worthy of consideration for the award of the degree of Bachelor of Technology in Computer Science Engineering of the University.**

**Supervisor Name**

# ACKNOWLEDGEMENTS

I would like to thank Northcap University for giving me the golden opportunity to showcase the skills I have acquired throughout the course of my degree. I would like to extend my appreciation to the entire faculty and staff of the School of Computer Science Engineering for providing me with a conducive environment for learning and nurturing my academic growth throughout our college years.

I would also like to express my deepest appreciation to Dr. Yogita Gigras for her invaluable support and guidance. She has provided during the completion of this project. I thank her for the patience and constructive feedback in shaping the outcome of this project.

I would also like to extend my sincere gratitude to the external guides who have provided me with constructive criticism and a path to follow in my project. Their guidance has helped me in refining my project and understanding the new prospects and possibilities in my topic of research.

**Akshat Singal**

**Amiya Trehan**

# EXECUTIVE SUMMARY

In the digital age, harnessing the power of technology can revolutionize the educational sector by making learning more engaging for students. This project aims to develop a smart teaching system tailored for primary school teachers and students, comprising two components: an alphabet recognition system utilizing motion gesture and a smart image generator.

The alphabet recognition system will enable students to learn and practice the alphabet through gestures. By utilizing MNIST dataset training, the Keras library and OpenCV, this component offers a splendid educational experience that captivates young minds.

Complementing the alphabet recognition system, the smart image generator component empowers teachers with a powerful tool to generate visually compelling and customized educational graphics. By using ControlNet, a neural network architecture and an interactive whiteboard, this component can generate a wide range of high-quality images based on custom input to pique the curiosity of young learners.

The integration of these two components creates a cohesive smart teaching system that takes educating youngsters to another level. This system not only enhances teaching efficiency but also nurtures creativity, visualization, and a lifelong love for learning among children. The technical highlights of this project include motion gesture capture, API interaction and user interaction with graphical elements.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  |  | **Page No.** |
|  | **ACKNOWLEDGEMENTS** | i |
|  | **EXECUTIVE SUMMARY** | ii |
|  | **TABLE OF CONTENTS** | iii |
|  | **LIST OF FIGURES** | v |
|  | **LIST OF TABLES** | vi |
|  | **LIST OF ABBREVIATIONS** | vii |
|  |  |  |
| **1.** | **INTRODUCTION** | 1 |
|  | 1.1 OBJECTIVE | 1 |
|  | 1.2 MOTIVATION | 2 |
|  | 1.3 BACKGROUND | 3 |
|  | 1.4 LITERATURE SURVEY | 4 |
|  |  |  |
| **2.** | **PROJECT DESCRIPTION AND GOALS** | 10 |
|  | 2.1 MODULE DESCRIPTION | 10 |
|  | 2.2 GOALS | 15 |
|  |  |  |
| **3.** | **TECHNICAL SPECIFICATIONS** | 16 |
|  | 3.1 HARDWARE REQUIREMENTS | 16 |
|  | 3.2 SOFTWARE REQUIREMENTS | 16 |
|  |  |  |
| **4.** | **DESIGN APPROACH AND DETAILS** | 19 |
|  | 4.1 DESIGN APPROACH | 19 |

|  |  |  |
| --- | --- | --- |
|  | 4.2 CODES AND STANDARDS | 21 |
|  | 4.3 CONSTRAINTS, ALTERNATIVES AND TRADEOFFS | 34 |
|  |  |  |
| **5.** | **SCHEDULE, TASKS AND MILESTONES** | 36 |
|  |  |  |
| **6.** | **PROJECT DEMONSTRATION** | 39 |
|  |  |  |
| **7.** | **COST ANALYSIS AND DISCUSSION** | 44 |
|  | 7.1 COST ANALYSIS | 44 |
|  | 7.2 RESULTS AND PERFORMANCE | 45 |
|  | 7.3 DISCUSSION | 45 |
|  |  |  |
| **8.** | **FUTURE SCOPE** | 47 |
|  |  |  |
| **9.** | **SUMMARY** | 49 |
|  |  |  |
| **10.** | **REFERENCES** | 50 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page no.** |
| 1. | Project architecture for Motion Recognition system | 19 |
| 2. | Project architecture for Smart Image generator | 20 |
| 3. | In the process of drawing the alphabet “c” | 39 |
| 4. | In the process of drawing the alphabet “a” and the correct prediction of the alphabet “c” | 39 |
| 5. | In the process of drawing the alphabet “t” and the correct prediction of the alphabet “a” | 40 |
| 6. | Result displaying the final word “cat” | 40 |
| 7. |  |  |
| 8. |  |  |
| 9. |  |  |
| 10. |  |  |
| 11. |  |  |
| 12. |  |  |
| 13. |  |  |
| 14. |  |  |
| 15. |  |  |
| 16. |  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page no.** |
| 1 | Hardware Requirements | 18 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ML** | Machine Learning |
| **AI** | Artificial Intelligence |
| **LoRa** | Low Rank Adaptation |
| **MiDaS** | Mixed Data Sampling |
| **API** | Application Programming Interface |
| **LSTM** | Long Short-Term Memory |
| **HED** | Holistically-Nested Edge Detection |
| **GPU** | Graphics Processing Unit |
| **OCR** | Optical Character Recognition |
| **CLIP** | Contrastive Language-Image Pre-training |
| **FID** | Fréchet Inception Distance |
| **URI** | Uniform Resource Identifier |
| **JS** | JavaScript |
| **ReFL** | Reward Feedback Learning |
| **GB** | gigabyte |
| **Mbps** | Megabits per second |
| **PC** | Personal Computer |
| **MNIST** | Modified National Institute of Standards and Technology |
| **HSV** | Hue, Saturation, Value |

|  |  |
| --- | --- |
| **ROI** | Region of Interest |
| **CPU** | Central Processing Unit |
| **AR** | Augmented Reality |
| **VR** | Virtual Reality |
| **CSS** | Cascading Style Sheets |
| **png** | Portable Network Graphics |
| **MOOCs** | Massive Open Online Courses |
| **GenAI** | Generative Artificial Intelligence |

1. **INTRODUCTION**

This section covers the objective of this project, the motivation to complete the project, a background of the techniques involved and a comprehensive literature survey.

## OBJECTIVE

The primary objective of this project is to apply the knowledge and skills I gained throughout my academic journey to create a smart learning system specifically designed to revolutionize the educational experience for children as well as empower teachers to teach more effectively.

The purpose is to make a system for teachers that functions as a comprehensive support tool, streamlining the teaching process. By incorporating the features of motion gesture recognition and smart image generation, the system will make it easier for the teacher to help students understand and memorize the alphabet better.

For students, the intent is to create an interactive and engaging learning system. Digital learning, integrated with the school curriculum, will ignite curiosity and a love for technology. This personalized approach will cater to each child's unique learning pace and style, ensuring no student gets left behind.

The target for my smart learning system is to bridge the gap between traditional education and the captivating world of technology. By leveraging the power of Machine Learning, this project aspires to empower educators, spark a passion for learning in young minds, and ultimately, pave the way for a more effective and enjoyable educational experiences.

## MOTIVATION

Currently, teaching young children can be a static experience. While traditional methods like flashcards and worksheets have their place, they come with limitations. They can become repetitive for young learners, hindering their motivation and focus. They often don't provide opportunities for children to actively participate and learn through exploration. Since children learn best through a combination of visual, auditory, and kinesthetic experiences, technology can provide innovative solutions to help cater to this multi-sensory approach.

Recent breakthroughs in ML techniques offer exciting possibilities for interactive learning. Motion gesture recognition allows devices to recognize and interpret physical movements. This can be harnessed to create an interactive alphabet learning system. In a class of multiple students, the teacher can simply capture the movement in air for presentation. As children explore letter shapes through physical motions, they develop a kinesthetic connection to the alphabetic symbols, enhancing their understanding and retention. This multisensory approach caters to diverse learning styles, ensuring an effective educational experience for all students. An intelligent image generator, that uses ControlNet, can take a child's chosen letter, along with a desired theme or layout, and generate a unique and engaging image. By combining these two components, this project aims to create a dynamic and engaging way for young children to learn the alphabet.

## BACKGROUND

In the current digital world, technology has become an integral part of life. It has become necessary for professionals harness technology effectively to become more effective and produce better work. The educational sector is not going to be left behind in this digital revolution. The educational landscape is undergoing a rapid transformation which is driven by the rapid advancement of technology and changing societal needs. Smart classrooms have been around since a while, making it easier for teachers to teach complex concepts, encourage kinesthetic learning and save time.

1. Motion Gesture Technology

Gesture Technology leverages computer vision to interpret the movements done by users without the need for physical contact with the system. Motion recognition has greatly improved since its onset in the 1980s with advancements in AI and Ml, which have improved the responsiveness and overall accuracy of such models. This technology continues to evolve and it is going to become increasingly common in our daily lives, making it easier for humans to interact naturally with the world around them. Gesture-based technology represents a significant step towards more human-centric computing, where the barriers between users and devices are reduced, and interactions become more seamless and intuitive.

1. Generative AI

Generative AI is a frontier in AI and has seen sincere development in past decade with the advent of models like GPT, DALL-E, Midjourney etc. GenAI can create original content based on previously fed patterns and data. Generative AI is pushing the boundaries of creativity and innovation by generating texts, images, video clips and even music.

Adopting GenAI in classrooms can help optimize teaching and learning for teachers and students respectively. AI-generated resources can aid the pre-existing educational material and offer diverse perspectives and approaches to tackling complex academic scenarios. GenAI is a

fascinating technology for young children and it can intrigue them to pursue the field of AI which is growing at a rapid pace.

## LITERATURE SURVEY

1. **Adding Conditional Control to Text-to-Image Diffusion Models**

**Authors - Lvmin Zhang, Anyi Rao, and Maneesh Agrawala Year - 2023**

This paper discusses ControlNet, a neural network architecture designed to add spatial conditioning controls to large, pre-trained text-to-image diffusion models like Stable Diffusion. ControlNet utilizes a novel approach of "zero convolutions" to connect a trainable copy of the model's encoding layers to the original, locked model, ensuring no harmful noise disrupts the fine-tuning process. This allows for the integration of various conditioning controls such as edges, depth, and structure, enabling a more precise image generation that aligns with users' spatial composition requirements. The experiments done by the authors demonstrate that ControlNet can facilitate broader applications in controlling image diffusion models, leading to more personalized and accurate image generation based on user inputs.

1. **Advancing AI Image Generation: Unveiling the Mechanisms and Innovations of Stable Diffusion Technology**

**Authors - Charles Anderson, Thomas Taylor, Christopher Moore Year - 2023**

The authors explore the advancements in AI image generation, particularly focusing on Stable Diffusion technology. The paper highlights ControlNet and T2I-Adapter control modules, which enhance image controllability and are changing industry production modes. The study delves into the technical aspects of Stable Diffusion, including its operating mechanism and the role of latent vector space in speeding up the process. This new-age technology can handle various tasks

such as text-to-image, image-to-image, and super-resolution. There is an emphasis on image information generators as a key performance improvement factor. Additionally, the paper provides insights into the computational methods and analyzes important modules of Stable Diffusion, aiming to assist in understanding its principles and applications in AI painting.

1. **Diffusion Self-Guidance for Controllable Image Generation**

**Authors - Dave Epstein, Allan Jabri1, Ben Poole, Alexei A. Efros, Aleksander Holynski Year - 2023**

This paper presents Self-Guidance, a novel method for controllable image generation using diffusion models, enabling precise manipulation of image attributes such as shape, position, and appearance without additional training or models. The technique leverages internal representations of pre-trained text-to-image diffusion models to guide the sampling process, allowing for complex image edits like resizing objects, altering their positions, or merging elements from different images. The study demonstrates Self-Guidance's ability to perform granular, zero-shot control over generated images, showcasing its potential for editing real images as well. The method's key contributions include disentangled control over the generative process and the ability to extract and utilize properties from model representations for meaningful guidance. There is an emphasis on the need for safeguards against harmful manipulations using this Technology. The results put forth the scope of Self-Guidance to extend the capabilities of generative models, offering a new level of creative freedom in image generation and editing.

1. **Empowering Local Image Generation: Harnessing Stable Diffusion for Machine Learning and AI**

**Authors - Ahmed Imran KABIR, Limon MAHOMUD, Abdullah Al FAHAD, Ridwan AHMED**

**Year - 2024**

This paper explores the use of Stable Diffusion's diffusion models for advanced image synthesis, focusing on local hardware implementation for personal or commercial use. It delves into the capabilities of deep learning models to generate images from text descriptions, emphasizing the importance of the initial noise in determining image elements. The experiments carried out by the authors show that diffusion models enhance image detail and quality when combined with model architecture, like LoRA and XL-XDXL 1.0 models. Stable Diffusion and related models enable the use of local computer GPUs for deep learning and AI image generation, offering an open-source alternative to paid applications like Midjourney.

1. **Controllable text-to-image generation**

**Authors - Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, Philip H. S. Torr Year-2019**

This work presents ControlGAN, a novel text-to-image generative adversarial network designed to synthesize high-quality images based on natural language descriptions and enable user control over specific parts of the image generation. The authors introduce a word-level spatial and channel-wise attention-driven generator that disentangles visual attributes, allowing the model to focus on generating sub regions corresponding to relevant words. Additionally, a word-level discriminator provides fine-grained feedback by correlating words with image regions, aiding in the manipulation of specific visual attributes without altering other content. The use of perceptual loss further reduces randomness in image generation, ensuring consistency with unmodified text. Extensive experiments on benchmark datasets demonstrate ControlGAN's ability to outperform existing methods, effectively manipulate synthetic images using text descriptions, and maintain image diversity. The paper's contributions include the attention-driven generator, word-level discriminator, and perceptual loss adoption, which collectively enhance the controllability and quality of text-to-image synthesis.

1. **Image Generation from Layout**

**Authors - Bo Zhao, Lili Meng, Weidong Yin, Leonid Sigal**

**Year - 2019**

This paper introduces Layout2Im, a novel approach for generating images from layouts, which includes bounding boxes and object categories. The authors address the challenges of image generation from layout by disentangling object representation into category and appearance, using convolutional LSTM for composition, and decoding to an image. They find that Layout2Im significantly outperforms previous methods on COCO-Stuff and Visual Genome datasets, demonstrating the ability to generate complex and diverse images with accurate object placement. The study explores the scope of this research in practical applications of automatic high-resolution image generation, potentially replacing visual search engines by further controlling fine-grained attributes.

1. **ImageReward: Learning and Evaluating Human Preferences for Text-to-Image Generation**

**Authors - Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, Yuxiao Dong**

**Year - 2024**

This research presents ImageReward, a comprehensive solution for learning and improving text- to-image models based on human preference feedback. It introduces a general-purpose text-to- image human preference reward model trained on 137k expert comparisons, outperforming existing scoring models in human evaluation. The paper also proposes ReFL, an algorithm for optimizing diffusion models against a scorer, which shows advantages over other methods in both automatic and human evaluations. The authors highlight challenges in generative models, such as text-image alignment, body problems, human aesthetic, and biases, by integrating human preferences into the model training process. The solution to tackling this problem could be a systematic annotation pipeline, extensive analysis of human preferences, and a direct tuning method for diffusion models.

1. **Uni-ControlNet: All-in-One Control to Text-to-Image Diffusion Models**

**Authors - Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, Kwan-Yee K. Wong[**

**Year - 2024**

This paper introduces a unified framework, Uni-ControlNet, which allows the simultaneous use of various local and global controls within a single text-to-image diffusion model. The aim of the study is to enhance the controllability and quality of image generation from text descriptions by incorporating additional control modes like edge maps, depth maps, and CLIP image embeddings. Uni-ControlNet, by fine-tuning only two adapters, can efficiently handle multiple control signals and outperforms existing methods in terms of controllability, generation quality, and composability. The findings indicate that Uni-ControlNet requires less fine-tuning cost and model size, making it more practical for real-world applications.

1. **ControlStyle: Text-Driven Stylized Image Generation Using Diffusion Priors**

**Authors - Jingwen Chen, Yingwei Pan, Ting Yao, Tao Mei Year - 2024**

This work introduces a new diffusion model, ControlStyle, which aims to improve editability in content creation by generating images that align semantically with text prompts and stylistically with a style image. It upgrades a pre-trained text-to-image diffusion model with a trainable modulation network, allowing for additional conditions of text prompts and style images. Extensive experiments carried out by the authors demonstrate that ControlStyle produces visually pleasing and artistic results, surpassing simple combinations of text-to-image models and conventional style transfer techniques.

1. **GlyphControl: Glyph Conditional Control for Visual Text Generation**

**Authors - Yukang Yang, Dongnan Gui2, Yuhui Yuan, Weicong Liang, Haisong Ding, Han Hu, Kai Chen**

**Year - 2024**

This work is on GlyphControl, a novel approach for visual text generation that enhances the performance of the Stable-Diffusion model by using glyph conditional information. The authors developed a training benchmark dataset called LAION-Glyph and evaluated the effectiveness of GlyphControl using OCR-based metrics, CLIP score, and FID. Empirical evaluations revealed that GlyphControl outperforms the recent DeepFloyd IF approach in terms of OCR accuracy, CLIP score, and FID. The paper suggested that exploration in advanced font rendering integration and color control will lead to the more robust visual text generation models.

1. **Characterizing Children’s Motion Qualities: Implications for the Design of Motion Applications for Children**

**Authors: Aishat Aloba, Lisa Anthony Year: 2021**

This research focuses on understanding how children's motions differ from adults' to improve motion recognition algorithms for children's applications. It utilizes 24 articulation features (11 newly developed) describing motions quantitatively, using the Kinder-Gator dataset which represents motions as postures defined by 3D positions of 20 joints. The key result was that children's motions are faster, more intense, less smooth, and less coordinated compared to adults. Additionally, the type of motion also affects variations in children's performances. The findings suggest a framework for enhancing motion recognition algorithms and designing motion applications tailored for children to improve their interactive experiences with motion-based applications.

1. **Liu, H., Yao, C., Zhang, Y., & Ban, X. (2024). GestureTeach: A gesture guided online teaching interactive model. Computer Animation and Virtual Worlds, 35(1), e2218.**

**Authors: Hongjun Liu, Chao Yao, Yalan Zhang, Xiaojuan Ban Year: 2024**

This paper discusses GestureTeach, an innovative online teaching model that leverages natural gestures for interaction, enhancing the teaching and learning experience. The model uses a multi- stage pipeline for gesture recognition, making it easy for teachers to engage students effectively. It consists of a gesture recognition module (GRM) and an animation generation module (AGM), which work together to transform hand gestures into animated educational content. The GRM employs a palm detection block (PDB) and a landmark regression block (LRB) to track hand movements and recognize gestures. The AGM then uses these gestures to create text and image animations, making teaching interactive and fun. The authors evaluated the system through a two-stage study involving 15 teachers and 90 students, demonstrating a preference for GestureTeach over traditional methods. GestureTeach has the potential to revolutionize online education by providing seamless interaction and improving knowledge comprehension through animated displays. Their findings imply that integrating intuitive gesture-based interactions can significantly enhance online teaching platforms, making them more engaging and effective for both educators and learners.

1. **Education in the Era of Generative Artificial Intelligence (AI): Understanding the Potential Benefits of ChatGPT in Promoting Teaching and Learning**

**Authors: David Baidoo-Anu, Leticia Owusu Ansah Year: 2023**

This article discusses the impact of generative AI, specifically ChatGPT, on education. It explores the potential benefits like personalized learning and interactive teaching, and drawbacks, like misinformation and privacy concerns. The paper emphasizes the need for collaboration among policymakers, educators, and technologists to safely integrate AI tools in education. It also discusses the implications for practice, suggesting that generative AI could enhance assessment practices and prepare students for an AI-driven work environment.

1. **Empowering education with generative artificial intelligence tools: Approach with an instructional design matrix**

**Authors: Lena Ivannova Ruiz-Rojas, Patricia Acosta-Vargas, Javier De-Moreta-Llovet and Mario Gonzalez-Rodriguez**

**Year: 2023**

This piece of research explores the integration of GenAI technologies with the 4PADAFE instructional design matrix to improve education, particularly in MOOCs. The study was conducted with teachers from the University of ESPE Armed Forces who joined a MOOC course and evaluated the impact of GenAI tools on the teaching-learning process. The findings indicate that AI tools combined with the 4PADAFE matrix can personalize education, adapt content, and promote key skills development to prepare students for 21st-century demands. The results showed a high utilization rate of tools like ChatGPT, with positive impacts on student engagement and motivation. However, the study had limitations like sample size and focus on specific tools.

### ARTIFICIAL INTELLIGENCE (AI) IN EDUCATION: USING AI TOOLS FOR TEACHING AND LEARNING PROCESS

**Author: Tira Nur Fitria Year: 2021**

The paper highlights various AI tools and platforms that have been integrated into educational technology, such as Virtual Mentors, Voice Assistants, Smart Content, Presentation Translators, Global Courses, Automatic Assessment, Personalized Learning, Educational Games, and Intelligent Tutoring System designed to facilitate daily educational activities and make the learning process more practical and effective. The author emphasizes that while AI can handle systemic tasks like grading and administrative reporting, it cannot replace the unique qualities of human teachers, such as providing inspiration and developing character. The author highlights the collaborative potential of AI with human intelligence, with teachers focusing more on non- systemic work and AI for systemic tasks.

# PROJECT DESCRIPTION AND GOALS

This section covers a detailed description of the project including an elaborate description of each module and the goals of the system.

## MODULE DESCRIPTION

The project consists of two subcomponents – a motion recognition system and a smart image generator.

### MOTION GESTURE RECOGNITION

This component uses machine learning techniques to recognize characters drawn in the air. It is composed of the following modules:

**Module 1: Building the Model and Calculating Accuracy**

* + 1. Data Handling

The MNIST dataset is imported and reshaped into 28x28 pixels. Specifically, the EMNIST (Extended MNIST) dataset for letters is selected. The dataset is loaded into two arrays: X (containing the images) and y (containing the corresponding labels).

* + 1. Data Split

The data is split into training (80%) and testing (20%) sets.

* + 1. Model Definition

By utilizing Keras, a neural network model is trained with Flatten, Dense and Dropout layers. The model is compiled with categorical cross-entropy loss and the Adam optimizer.

* + 1. Accuracy Evaluation

Before training, the model’s accuracy is evaluated on the test set. The model is then trained using the training data for 10 epochs. A ModelCheckpoint callback saves the best model during training based on validation performance. Finally, the trained model weights are loaded from the saved file 'the\_best\_model.keras'.

**Module 2: Alphabet Recognizer**

1. Color Detection

The color blue is defined in HSV format to detect the object used to draw the character. blueLower and blueUpper represent the lower and upper bounds for detecting blue colors. These values are used later for color filtering.

1. Kernel, Blackboard and Deque Creation

A kernel of size 5 x 5 is defined to be used in image processing operations like erosion, dilation and morphological transformations.

A 480 x 640 x 3 array blackboard is initialized with zeros. This is the canvas where letters or drawings can be displayed. The alphabet of a 200 x 200 x 3 array is also initialized. These arrays will be used to store drawn letters.

A deque (double-ended queue) named points created with a maximum length of 512 points will store the coordinates of points drawn on the screen.

**Module 3: Final Prediction**

The code opens the webcam using OpenCV.

1. Character Processing

Captured frames are flipped horizontally and the frames are converted to HSV and gray scale. Blue pixels falling within the specified range are detected. The blue mask is processed by applying morphological operations of erosion, opening, and dilation.

1. Contour Search (blue colored object detection)

The program looks for contours in the image. If contours are found, the largest contour is selected based on area, a circle is drawn around the contour, the centroid of the contour is calculated and it is added to the points deque. When no contours are detected, the blackboard is converted to gray scale, median blur is applied to reduce noise, Gaussian blur is applied for further smoothing and the blurred image is thresholded. Finally the contours are found in the thresholded image.

1. Contour processing

If there is at least one contour detected, the contours are sorted by area in descending order. It is checked if the area of the largest contour is greater than a threshold. If this condition is met, the bounding rectangle around the contour is calculated and the ROI containing the drawn alphabet from the grayscale blackboard is extracted. Then the alphabet image is resized to a fixed size and pixel values are normalized.

1. Connecting points with lines

The code then iterates through the stored points and connects them with lines. If both the current and previous points exist , a black line is drawn on the original frame connecting the points. A white line is drawn on the blackboard connecting the points.

1. Prediction

Finally the model predicts the character based on the processed input by finding the index of the maximum value in the prediction array.

1. Resetting and Drawing

After processing the alphabet gesture, the points deque is emptied and the blackboard is reset to an empty canvas.

### SMART IMAGE GENERATOR

This component takes input in the form of an image drawn by the user on the whiteboard and a prompt specified to generate an image.

**Module 1: Install and setup the API**

1. Installation of the Replicate package First, the Replicate package is installed.
2. Set the Replicate API environment variable

The OS library is imported. The user is asked to input their unique token number which is authenticated by the Replicate service. This sets up the environment for the user.

**Module 2 : Design the Whiteboard**

1. Initialization

The size of the whiteboard is defined. Then, the scaling factor is determined for display. The cellSize is then calculated based on the scaling factor and the canvas size.

1. Whiteboard table creation

A table element is created to represent the whiteboard, with styling applied for appearance and functionality. The table is set to have a white background, a crosshair cursor, and a light gray border.

1. Addition of drawing functionality

The variables isDrawing and selectedPixel are used to track the drawing state and the currently used pixel. Event listeners are added to handle mouse clicks, movements, and context menu events. The functions: handleMouseClick, handleMouseMove, handleMouseDown, handleMouseUp, and handleContextMenu, are defined to manage drawing interactions.

1. Addition of buttons

Two buttons - Save Drawing and Delete Drawing are created to allow users to save their drawings as a png or clear the whiteboard if they are not satisfied. Event listeners are added to the buttons to invoke the respective functions when clicked.

1. Final Setup

A loop is used to create rows and cells for the table based on the canvas size.

Each cell is styled to represent a pixel on the whiteboard and is given event listeners for drawing functionality. A style element is created to apply additional CSS rules to the table cells. The table and buttons are appended to a container, which is then added to the document body.

1. Execution

The entire JavaScript code is executed to display the whiteboard in the notebook.

**Module 3: Image processing**

The program reads an image file and encodes it in base64 format. Then it constructs a data URI using the encoded image string.

**Module 4: ControlNet Invocation**

1. Input Preparation

The input parameters for the AI model, including seed, image, prompt, structure, and resolution are set up. The user is asked to input a prompt.

1. Model Execution

The replicate.run function is called and an image is generated by ControlNet based on the image and the prompt. Finally, the output link to the generated image is displayed.

# GOALS

This project focuses on creating a fun and engaging system to teach young children the alphabet.

* + 1. Develop a motion gesture recognition system that allows children to learn alphabet structure by physically forming them in the air and also provides them with real-time feedback and recognition of the formed letters.
    2. Ensure the system provides as accurate results as possible.
    3. Ensure the system can run repeatedly and maintain responsiveness.
    4. Ensure the system is easy-to-use.
    5. Make the system run and respond as quickly as possible.

# TECHNICAL SPECIFICATIONS

This section consists of the hardware and software constraints of building and using this smart teaching system effectively.

## HARDWARE REQUIREMENTS

**Table 1. Hardware Requirements**

|  |  |
| --- | --- |
| **Type of Hardware** | **Minimum Requirements** |
| Processor | Intel Core i7 7th Gen Processor (or equivalent) |
| RAM | 8 GB |
| Graphics Card | Nvidia GeForce Graphics Card (compatible with DirectX 12 or later with WDDM 2.0 driver) |
| NIC | Integrated Wi-Fi 802.11ac |
| Internet connection | At least 4 Mbps stable internet connection |

## SOFTWARE REQUIREMENTS

**Notebooks used**

* + 1. **Google Colab:** Used to execute the Smart Image Generator component on Google's cloud servers. This has been done to utilize computing resources including GPUs and TPUs with ease.
    2. **Jupyter Notebook:** This open-source web application has been used to execute the motion gesture recognition component.

**Languages used**

1. **Python:** This high-level programming language has been used to code both the components. In the Motion gesture recognition component, handling the data, building the model, calculating accuracy, carrying out OCR and predicting the alphabet is carried out through Python. In the Smart Image Generation component, it is responsible for downloading the relevant packages and libraries, setting up an environment variable, encode image data, send a request to the diffusion model and integrate a JS code to make the model work.
2. **JavaScript:** This scripting language has been used to create the Whiteboard application within the model. The JS code enables users to interactively draw on a virtual whiteboard, save the drawing as a png file on their PC and delete the current drawing if they are not satisfied.

**API used**

1. **rossjillian/controlnet:** This diffusion model is used to generate an image for the user based on the input image and prompt.

**Dataset used**

1. **MNIST :** A popular database of handwritten digits that is commonly used for training image processing systems

**Libraries and Frameworks used**

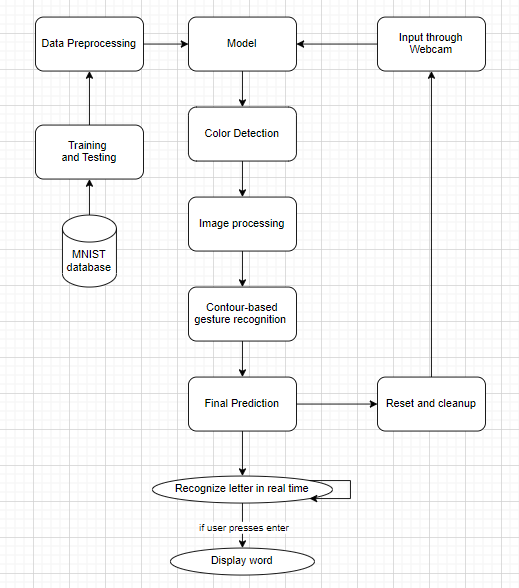
1. **OpenCV:** An open-source library used to process images and videos to identify objects, faces,the handwriting of a human etc.
2. **Keras**: A popular deep learning framework
3. **Numpy**: A Python library used for working with arrays
4. **Sklearn:** Python library to implement machine learning models and statistical modeling
5. **Replicate**: This library allows users to run open-source machine learning models with a cloud API.
6. **OS**: This module in Python provides functions for interacting with the operating system.
7. **IPython** : This library provides an interactive command-line terminal for Python and an interactive command shell for interactive computing in multiple programming languages.

# DESIGN APPROACH AND DETAILS

This section covers the project design and codes.

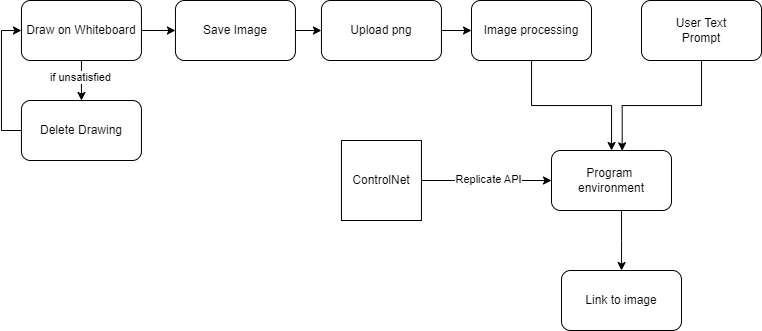
## DESIGN APPROACH

**MOTION RECOGNITION SYSTEM - Project Architecture**

****

**Fig 1. Project architecture for Motion Recognition System**

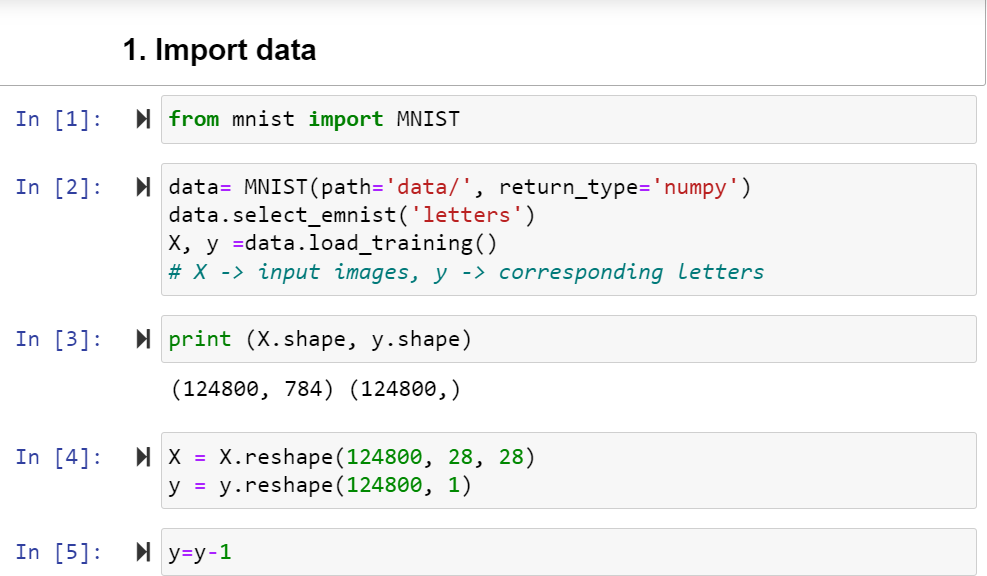
**SMART IMAGE GENERATOR - Project Architecture**

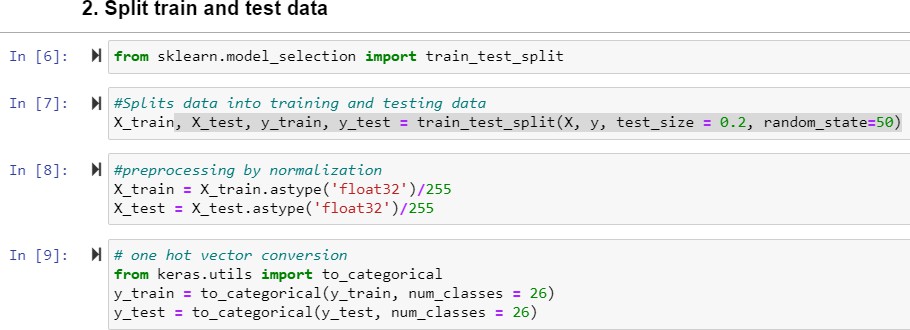
****

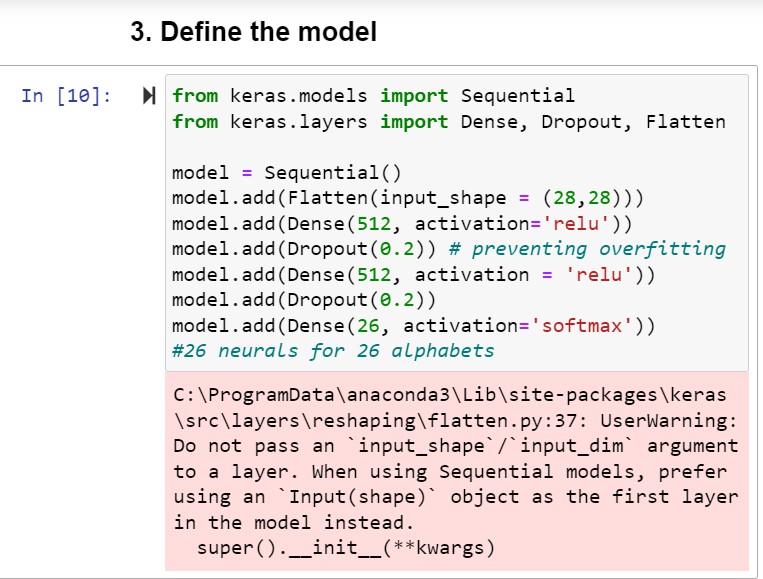
**Fig 2. Project architecture for Smart Image generator**

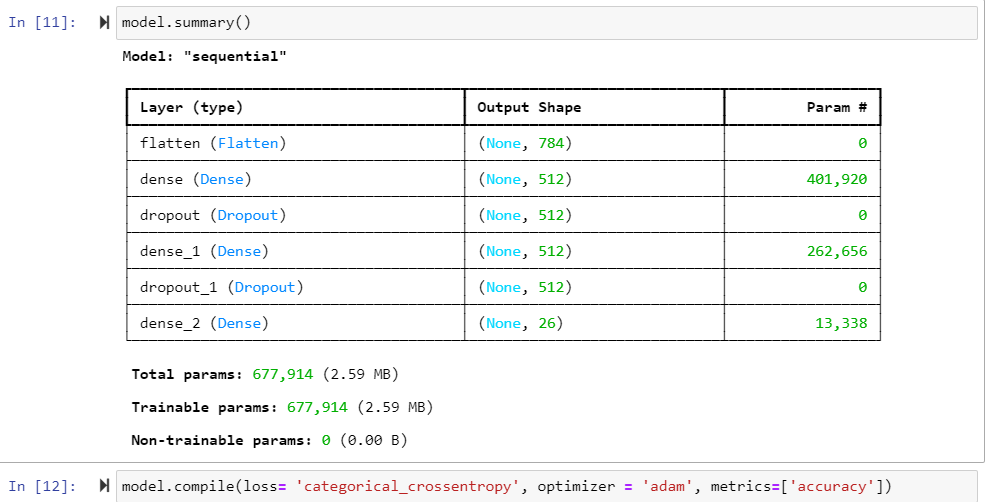
# CODES AND STANDARDS

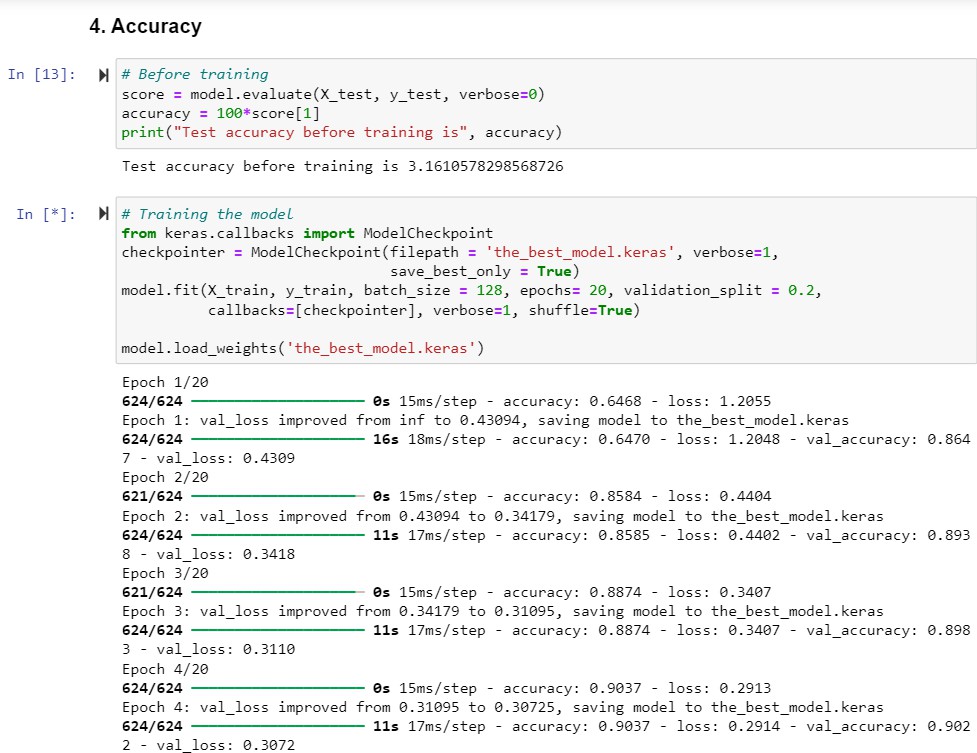
**MOTION RECOGNITION SYSTEM**

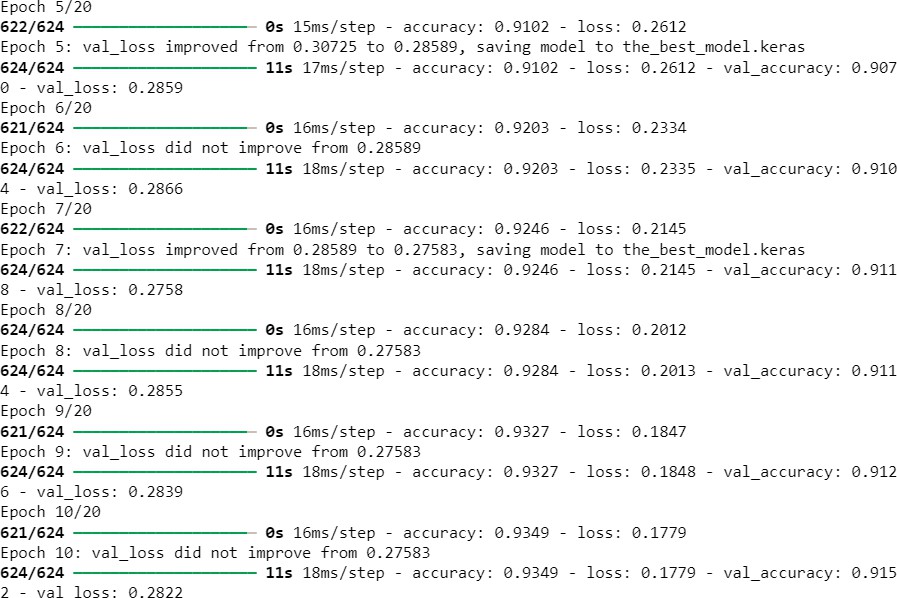
****

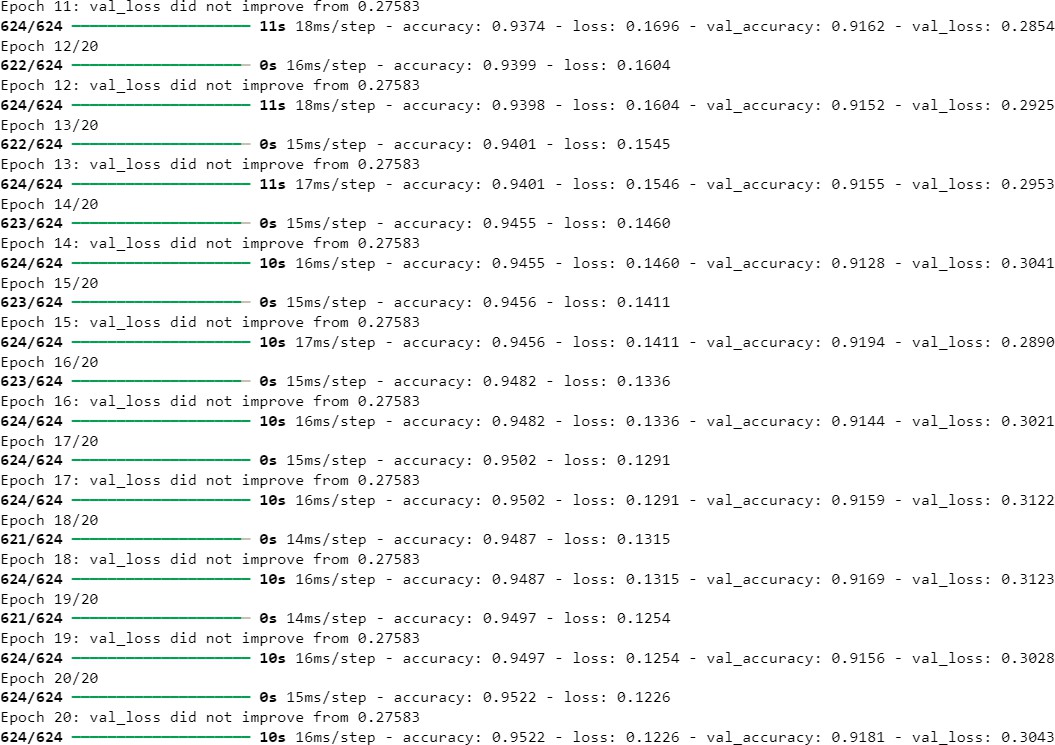


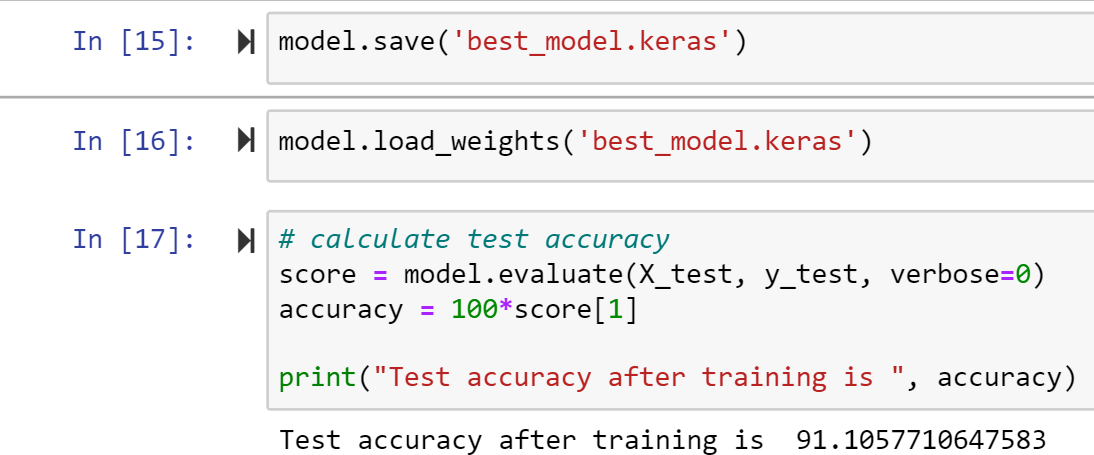
****

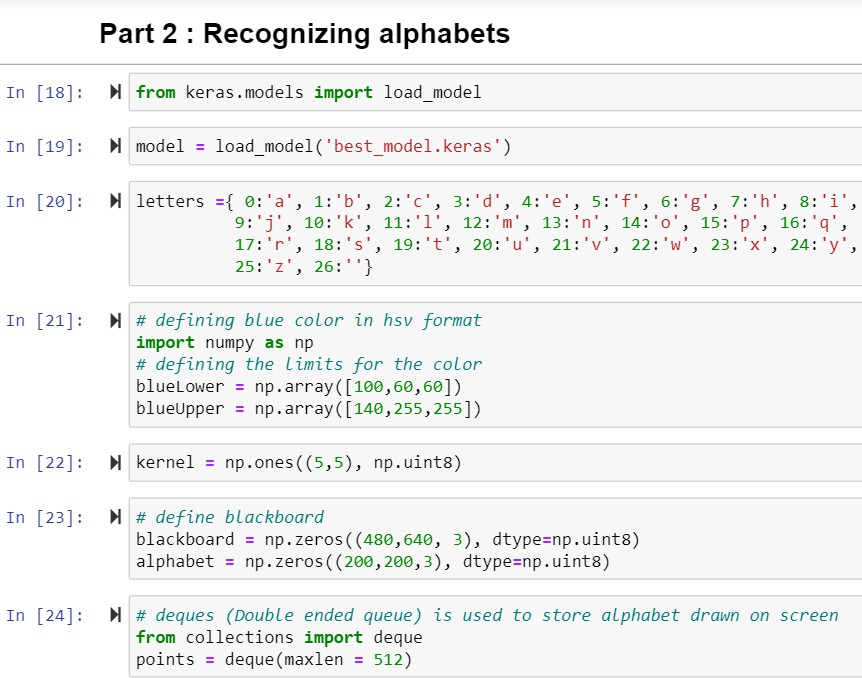


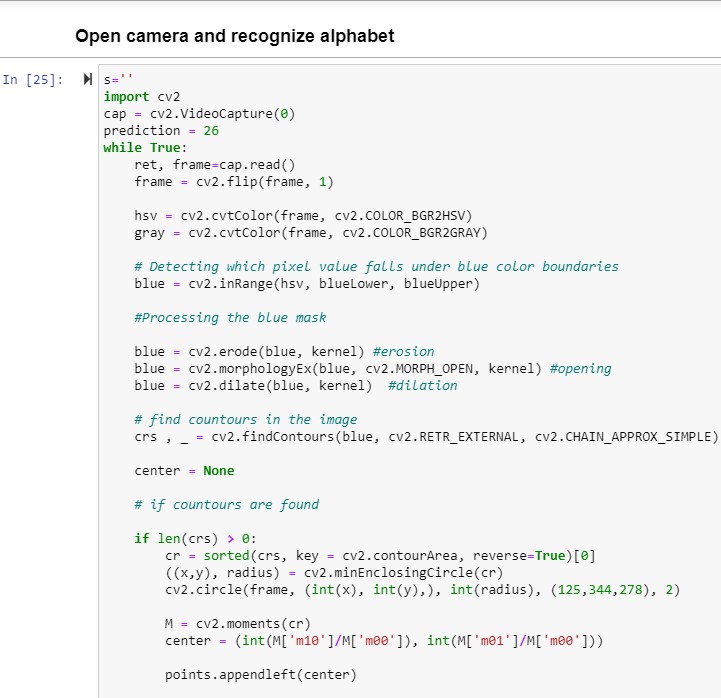
****

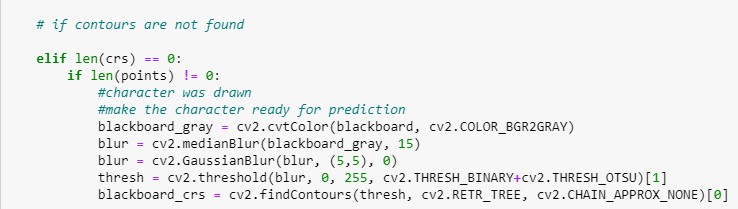


****



****

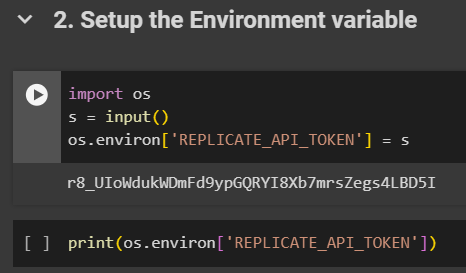
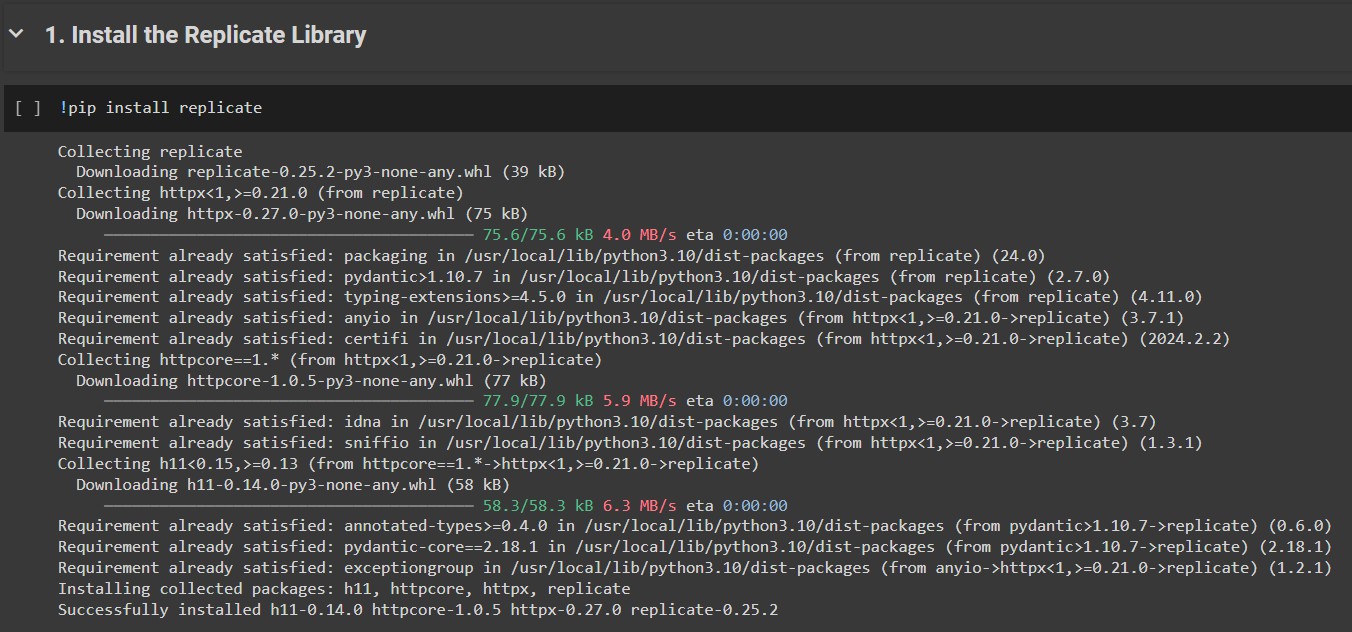


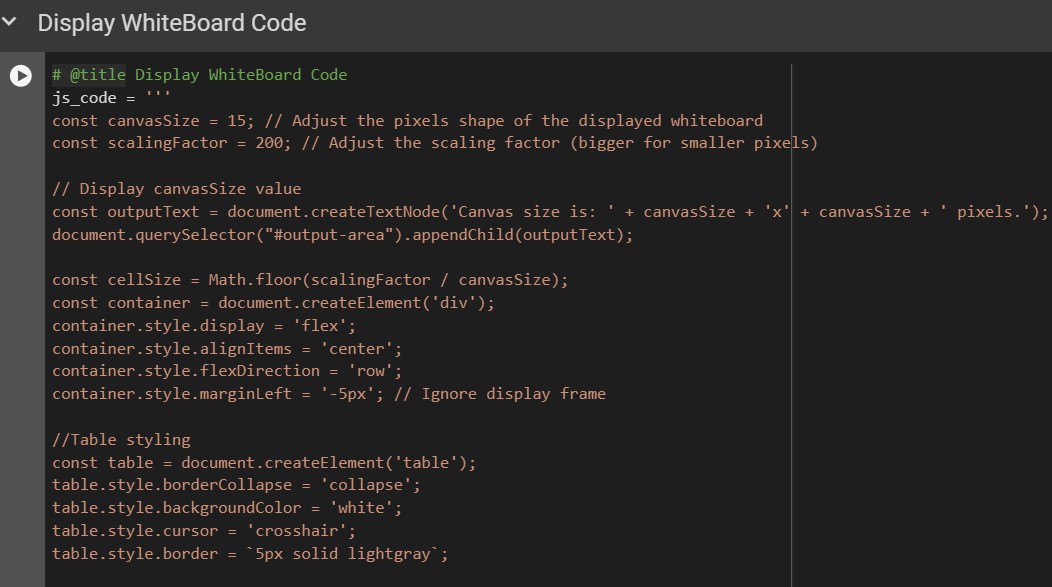
****

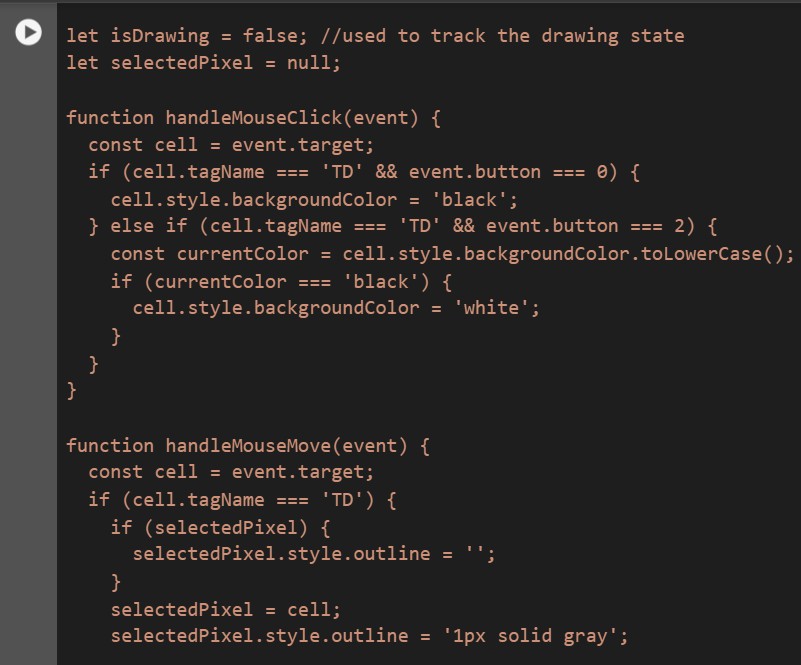


****

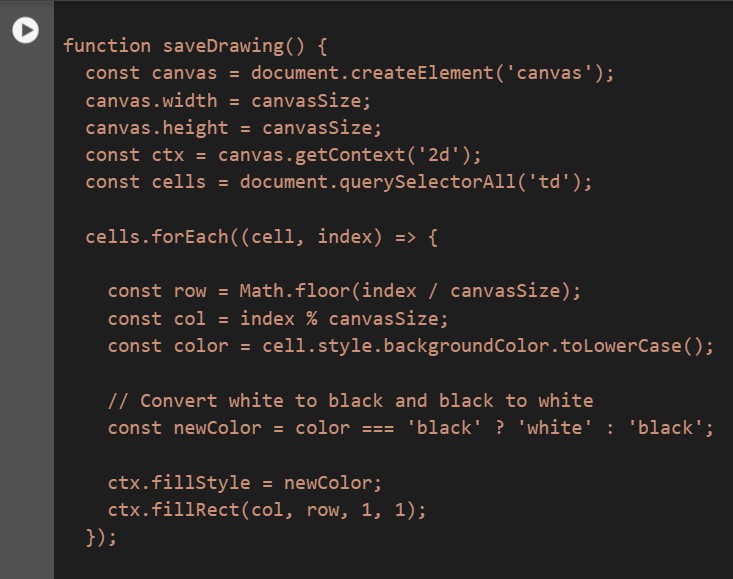
**SMART IMAGE GENERATOR**

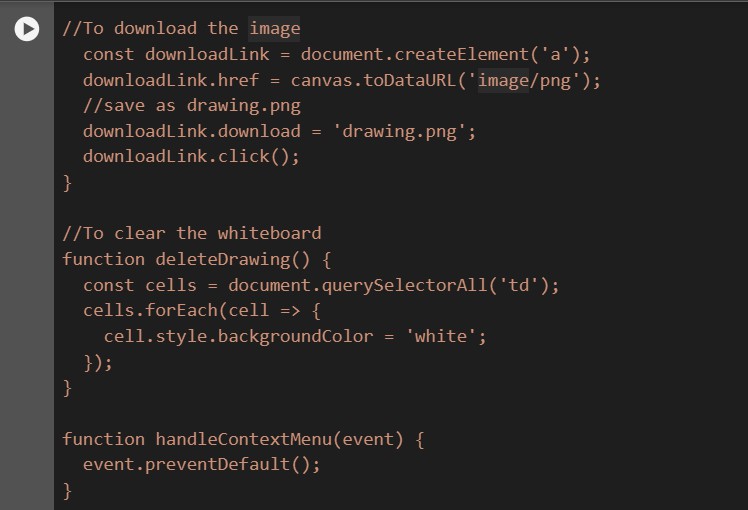
****

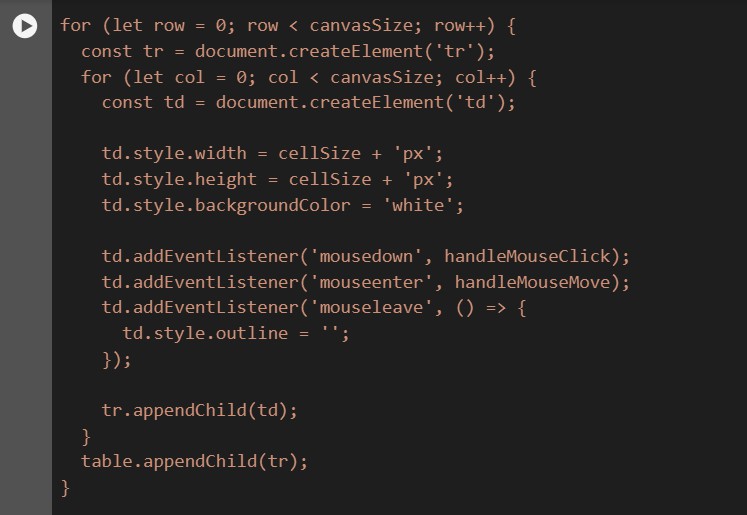


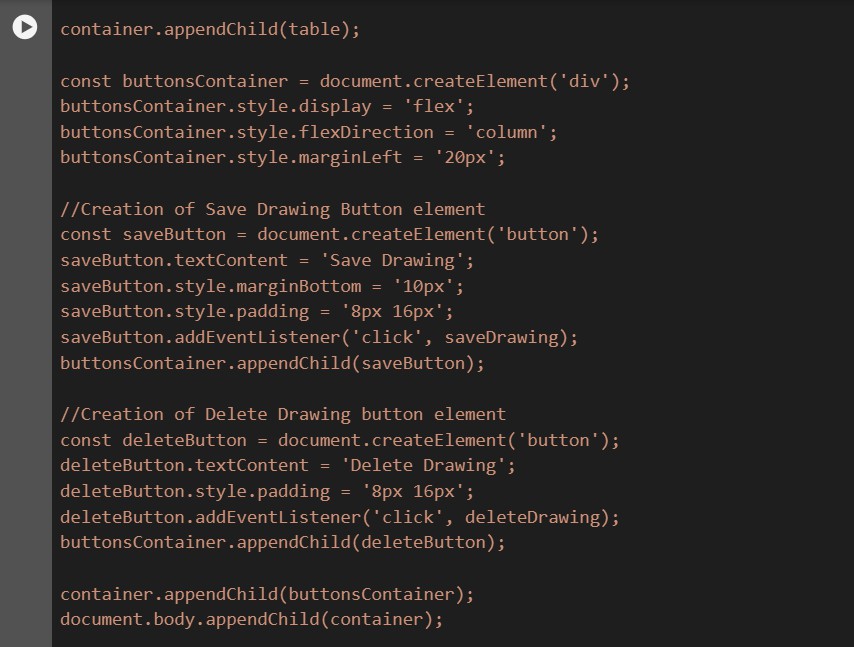
****

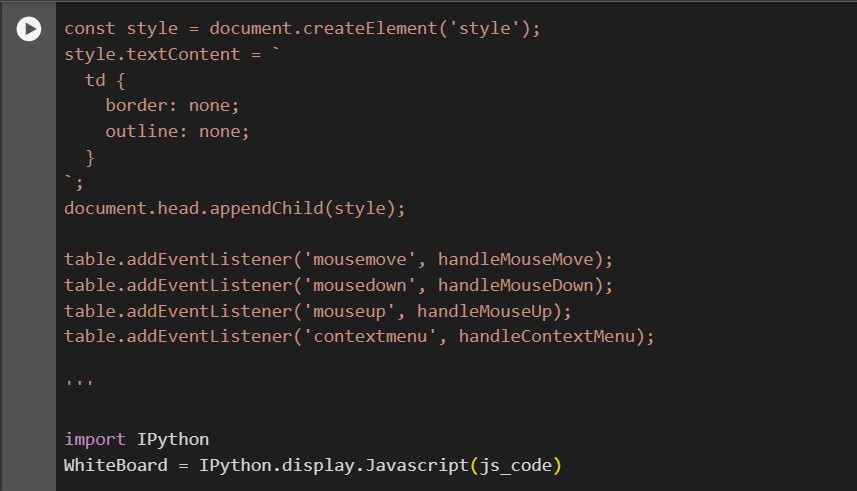


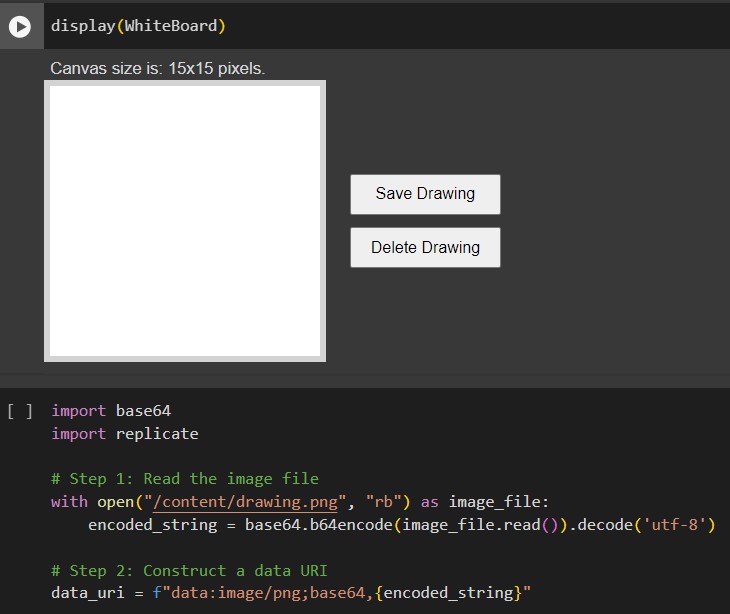
****



****



****



****

## CONSTRAINTS, ALTERNATIVES AND TRADEOFFS

1. **Constraints**
2. **Hardware Constraints:**
   * Computational Power: Training and deploying deep learning models, especially for real- time video analysis, is computationally intensive and requires powerful hardware resources (e.g., GPUs, high-end CPUs).
3. **Software Constraints:**
   * Real-time Performance: The system must process and analyze real-time gestures, imposing stringent performance requirements on the deep learning models and associated software components.
4. **Environmental Constraints:**
   * Lighting Conditions: The system's performance may be affected by varying lighting conditions in different environments, such as low-light scenarios or extreme brightness.
   * Camera Angles and Occlusions: If the blue object is partially or fully occluded in certain camera angles, detection and recognition could be more challenging.
   * Background setup: If multiple blue-colored objects are present in the environment, recognition might get erred.

## Alternatives

**1. Other generative AI programs for image generation:** Services such as MidJourney, Dall-E, Adobe Firefly etc. can be used in place of this Stable Diffusion model. However, these generalized models don't offer enough personalization options. Developing a model of one's own can prove to be extremely resource-intensive. If the budget of an institute is

low, they can go for cheaper image generation models to be integrated with their learning platform.

## Trade Offs

**Real-time Performance vs. Computational Complexity**

Enabling real-time video analysis is a key requirement for the system, but this comes at the cost of increased computational complexity, especially when dealing with high-resolution video streams or large-scale deployments. Optimizing the ML model and leveraging hardware acceleration including GPUs will be necessary to maintain performance without compromising accuracy.

**High Initial Investment vs. Better Education**

The development and deployment of this Smart Teaching System will require substantial upfront investments in hardware, software and training resources. However it comes with the long-term benefits of increased student retention, parent satisfaction and educator satisfaction which could potentially offset these costs and lead to an overall benefit of an educational institution.

**Traditional teaching vs. Modern methods**

It is possible that teachers/parents argue against the use of technology in the favor of traditional teaching. Here, it is important to understand that this system aims to aid traditional methods of teaching and learning such as writing on paper, using flashcards, scribbling on the whiteboard, not replace them.

# SCHEDULE, TASKS AND MILESTONES

## SCHEDULE

**Phase 1: Project Planning and Requirements Gathering (Week 1-3)**

1. Define project scope, objectives, and goals.
2. Conduct literature surveys on existing solutions and identify gaps.

**Phase 2: System Architecture and Design for Component 1 (Week 4-6)**

1. Design the system architecture.
2. Select appropriate technologies and frameworks to be used in the component.
3. Work towards the feedback provided by the examiner

**Phase 3: Motion Gesture Recognition Component Completion (Week 7-9**

1. Collect and preprocess the dataset of motion gestures for alphabet recognition.
2. Develop and train machine learning models for gesture recognition.
3. Code and finish the real-time motion recognition model.

**Phase 4: Smart Image Generation Module planning (Week 10-12)**

1. Look for pre-trained models to integrate with the existing component.
2. Test out models and choose one.
3. Integrate with the help of API

**Phase5 : Whiteboard completion (Week 12-14)**

1. Code the whiteboard and integrate it with the smart image generator
2. Work towards feedback provided by the examiner

**Phase 6 : Project Completion (Week 14-16)**

1. Optimize performance and address any issues identified.
2. Consolidate all research and prepare the report

## TASKS

Task 1: Read relevant literary sources to understand the current work. Task 2: Design the overall system architecture and select technologies.

Task 3: Collect and preprocess datasets of motion gestures for motion recognition. Task 4: Develop and train machine learning models for motion gesture recognition.

Task 5: Implement the motion gesture recognition module for real-time motion recognition. Task 6: Look for relevant models for the smart image generation component.

Task 7: Test out the models and select the most relevant one. Task 8: Prepare an interactive whiteboard on Javascript

Task 9: Using Replicate, integrate the model with the whiteboard. Task 10: Address any issues identified.

Task 11: Complete the final project and report.

## MILESTONES

Milestone 1: Project scope, objectives, and requirements defined.

Milestone 2: System architecture and technology selection for the Motion Recognition system completed.

Milestone 3: Dataset collected and preprocessed for Component 1.

Milestone 4: Machine learning models for motion gesture recognition developed and trained for Component 1.

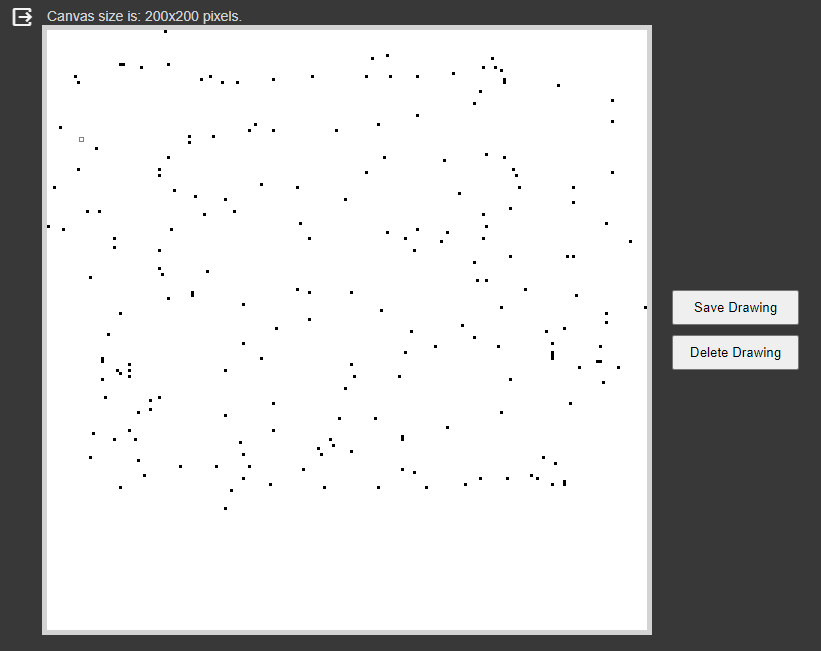
Milestone 5: Motion gesture recognition module implemented and integrated. Milestone 6: Technology selection for Smart Image generator completed.

Milestone 7: Interactive whiteboard implemented.

Milestone 8: Thorough self-testing, performance optimization and issue resolution completed.

### SMART IMAGE GENERATOR

**Trial 1 : Snowfall in Delhi**

****

**Fig 9. Drawing scattered snowflakes on a whiteboard o200 x 200 pixel whiteboard for Trial 1**

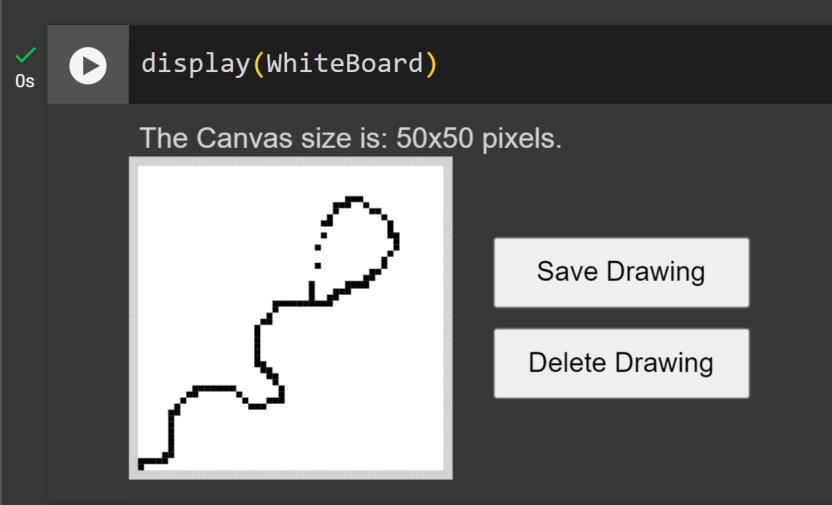
****

**Fig 10. Giving the prompt and receiving the output link for Trial 1**



**Fig 11. Output image for Trial 1**

**Trial 2 : A balloon floating through space**

****

**Fig 12. Drawing a floating balloon on a 50 x 50 pixel whiteboard for Trial 2**

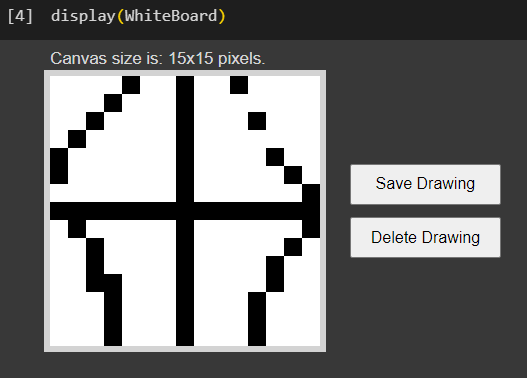


**Fig 13. Giving the prompt and receiving the output link for Trial 2**

****

**Fig 14. Output image for Trial 2**

**Trial 3: A window with red curtains**

****

**Fig 15. Drawing a window with curtains on a 15 x 15 pixel whiteboard for Trial 3**

****

**Fig 16. Giving the prompt and receiving the output link for Trial 3**



**Fig 17. Output image for Trial 3**

# COST ANALYSIS AND DISCUSSION

## COST ANALYSIS

Here is a cost analysis for building and deploying the proposed smart teaching system in the educational sector:

* + 1. **Development Costs**

Software Development: Minimal, using open-source libraries and frameworks.

Data Collection and Preprocessing: Approximately $130 (₹10,000) for data annotation and preprocessing.

Model Development and Training: Approximately $195 (₹15,000) for cloud GPU instances or workstations.

* + 1. **Deployment Costs**:

Cloud Computing Resources: Approximately $65 - $130 per month (₹5,000 - ₹10,000) depending on usage.

On-Premises Server: Approximately $260 - $390 (₹20,000 - ₹30,000) for a mid-range server.

Integration with Webcam: Approximately $130 - $260 (₹10,000 - ₹20,000) for integration and hardware upgrades.

* + 1. **Maintenance and Operational Costs:**

Model Retraining and Updates: Approximately $65 per year (₹5,000) for periodic retraining and updates.

Data Storage and Management: Approximately $13 - $26 per month (₹1,000 - ₹2,000) for secure storage.

* + 1. API usage cost

For small-scale development and deployment, the total costs could range between approximately

$390 - $650 (₹30,000 - ₹50,000), including initial development, deployment, and the first year of maintenance and operational expenses.

\*\*ControlNet runs on Nvidia A100 (40GB) GPU hardware. The predictions typically complete within 5 seconds. The prediction time for the model varies significantly based on the inputs.

## RESULTS AND PERFORMANCE

### MOTION RECOGNITION SYSTEM

The accuracy is calculated using the following formula:

Accuracy = correct predictions X 100

total predictions

Where:

"correct predictions" refers to the number of predictions made by the model that match the true labels in the test set.

"total predictions" refers to the total number of predictions made by the model on the test set.

* + 1. Tested on batch\_size =64, epochs=30

accuracy = 90.70112109184265

* + 1. Tested on batch\_size = 64, epochs =10

accuracy = 90.91346263885498

* + 1. Tested on batch\_size = 128, epochs =10

accuracy = 91.14983677864075

* + 1. Tested on batch\_size =128, epochs =20

accuracy = 91.34615659713745

Therefore, a batch size of 128 and 20 epochs was chosen for this project.

## DISCUSSION

The project is still on a primary level and there are multiple improvements that could be made to it if provided with additional resources and time.

Due to system constraints, the motion recognition system could not be run on more epochs and vast batch sizes. Therefore, the accuracy was not able to cross a certain limit. Currently, the motion gesture is captured in continuation and does not allow for breaks. Therefore it gets difficult to draw the letter 'i'. The program is only accustomed to identify the color blue as a cursor. This could pose real life difficulties in the classroom since there will probably be multiple blue colored objects present in the environment. Therefore, this is okay for demo purposes, but for real life, the color needs to be made very specific and unique and a specialized device could be given to the teacher for the same. Clearly, with more data and better testing, the accuracy of the model can improve.

The whiteboard for the smart image generator could be improved many folds with more research and expertise. Currently, the whiteboard is a little difficult to work with. I have made the whiteboard to be 15 x 15 pixels. An eraser, multiple colors and a larger space could take the image generation to another level. The reason for adding the whiteboard to this project was just to make it a one-stop application for teachers.

The Motion Recognition System code was more feasible to run on Anaconda whereas the Smart image generator ran better on Google Colab. Therefore, it was difficult to integrate both the programs. Ideally, a smart learning interface can be made for educational institutes which can include other functionalities as well.

# FUTURE SCOPE

This project has exciting future prospects and a promising potential to expand and improve.

1. Advanced Motion Recognition

The overall motion gesture recognition technology can be improved to make the system smoother. After mastering the alphabet, the system could be extended to teach letter sounds, forming simple words, and eventually progressing to basic reading skills.

1. Incorporation of phonics

By integrating phonics, the system could teach the relationship between letters and sounds, promoting better pronunciation and reading comprehension. By integrating sounds, music, or sound effects, engagement can be enhanced.

1. Customization for Different Learning Styles

The system could offer various difficulty levels and learning pathways to cater to visual, auditory, and kinesthetic learning styles.

1. Multilingual support

With the right model training and testing, the program can be expanded to support multiple languages for learning the alphabet.

1. Engaging user interface

By creating a smart learning portal, the models can be integrated to work together, simultaneously and seamlessly, greatly enhancing the learning experience.

1. Multi-platform Accessibility

By making the system accessible on various devices like smartphones, tablets, and smart speakers, the reach and convenience can be increased. In the digital age, e-learning is gaining limelight. Soon this platform can be made available for children to learn at home.

1. Integration with emerging technologies

Imagine children interacting with letters and words in an AR environment, creating a more immersive learning experience. VR could potentially create interactive 3D alphabet worlds for children to explore and learn in, further enhancing engagement. In the future, the system could leverage machine learning to adapt to a child's learning pace and adjust difficulty levels or content in real-time.

# SUMMARY

The Smart Teaching System is an innovative educational tool designed to enhance the learning experience for primary school students by integrating technology into the classroom. The system comprises two main components: a Motion Recognition System and a Smart Image Generator. The Motion Recognition System utilizes motion gesture technology and the MNIST dataset to create an interactive platform where students can learn and practice the alphabet through gestures, receiving instant feedback. This system is built using Keras, OpenCV, and other machine learning techniques, ensuring an engaging and multisensory learning experience. The smart image generator, powered by ControlNet, a neural network architecture, enables teachers to produce customized educational graphics, enriching the teaching material with high-quality images that capture students' attention. Both components work together to provide a dynamic and cohesive teaching environment that not only improves teaching efficiency but also fosters creativity, visualization, and a lifelong love for learning among children.

The system has been built after a rigorous literature survey. The design approach details the project architecture, codes and standards, and addresses constraints, alternatives, and tradeoffs.

Future enhancements could include advanced motion recognition, phonics integration, customization for different learning styles, multilingual support, and the use of emerging technologies like AR and VR to further revolutionize the educational landscape. The project demonstrates significant potential for scalability and adaptation to various educational settings, promising to make learning more accessible, personalized, and enjoyable for young students.

# REFERENCES

1. Zhang, L., Rao, A., & Agrawala, M. (2023). Adding conditional control to text-to-image diffusion models. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 3836-3847).
2. Anderson, C., Taylor, T., & Moore, C. Advancing AI Image Generation: Unveiling the Mechanisms and Innovations of Stable Diffusion Technology.
3. Epstein, D., Jabri, A., Poole, B., Efros, A., & Holynski, A. (2023). Diffusion self-guidance for controllable image generation. Advances in Neural Information Processing Systems, 36, 16222-16239.
4. KABIR, A. I., MAHOMUD, L., Al FAHAD, A., & AHMED, R. Empowering Local Image

Generation: Harnessing Stable Diffusion for Machine Learning and AI.

1. Li, B., Qi, X., Lukasiewicz, T., & Torr, P. (2019). Controllable text-to-image generation. Advances in neural information processing systems, 32.
2. Zhao, B., Meng, L., Yin, W., & Sigal, L. (2019). Image generation from layout. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8584-8593).
3. Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., ... & Dong, Y. (2024). Imagereward: Learning and evaluating human preferences for text-to-image generation. Advances in Neural Information Processing Systems, 36.
4. Zhao, S., Chen, D., Chen, Y. C., Bao, J., Hao, S., Yuan, L., & Wong, K. Y. K. (2024). Uni-

controlnet: All-in-one control to text-to-image diffusion models. Advances in Neural Information Processing Systems, 36.

1. Chen, J., Pan, Y., Yao, T., & Mei, T. (2023, October). Controlstyle: Text-driven stylized image generation using diffusion priors. In Proceedings of the 31st ACM International Conference on Multimedia (pp. 7540-7548).
2. Yang, Y., Gui, D., Yuan, Y., Liang, W., Ding, H., Hu, H., & Chen, K. (2024). GlyphControl: Glyph Conditional Control for Visual Text Generation. Advances in Neural Information Processing Systems, 36.
3. Aloba, A., & Anthony, L. (2021, October). Characterizing Children's Motion Qualities: Implications for the Design of Motion Applications for Children. In Proceedings of the 2021 International Conference on Multimodal Interaction (pp. 229-238).
4. Liu, H., Yao, C., Zhang, Y., & Ban, X. (2024). GestureTeach: A gesture guided online teaching interactive model. Computer Animation and Virtual Worlds, 35(1), e2218.
5. Baidoo-Anu, D., & Ansah, L. O. (2023). Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning. Journal of AI, 7(1), 52-62.
6. Ruiz-Rojas, L. I., Acosta-Vargas, P., De-Moreta-Llovet, J., & Gonzalez-Rodriguez, M. (2023). Empowering education with generative artificial intelligence tools: Approach with an instructional design matrix. Sustainability, 15(15), 11524.
7. Fitria, T. N. (2021, December). Artificial intelligence (AI) in education: Using AI tools for teaching and learning process. In Prosiding Seminar Nasional & Call for Paper STIE AAS (pp. 134-147).