Vivekanand Education Societys

Institute of Technology

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Air Doodle

Submitted in partial fulfillment of the requirements

of the

T.E Project in

Machine Learning

by

Sujal Sahu - 50

Sayyed Maaz - 52

Mohammed Arshad Shaikh - 53

under the guidance of

Ms Bincy Ivin

Department of Artificial Intelligence and Data Science

Vivekanand Education Society's Institute of Technology

2024-2025

VESIT

VIVEKANAND EDUCATION SOCIETY

INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

Since 1962

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Department of Artificial Intelligence and Data Science

CERTIFICATE

This is to certify that **Mr. Sujal Sahu, Mr. Sayyed Maaz, Mr. Mohammed Arshad Shaikh** of T.E D11AD Div B of Artificial Intelligence and Data Science studying under the University of Mumbai have satisfactorily presented the Project entitled **Air Doodle** as a part of the T.E Mini Project for Semester-VI of Machine Learning Lab under the guidance of **Ms. Bincy Ivin** in the year 2024-2025.

Date: 08 April 2025

(Name and sign)
Head of Department

(Name and sign)
Supervisor/Guide

VESIT
V.E.S.
Since 1962
VIVEKANAND EDUCATION SOCIETY
INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Department of Artificial Intelligence and Data Science

DECLARATION

We, ***Sujal Sahu, Sayyed Maaz, Mohammed Arshad Shaikh*** from ***D11ADB***, declare that this project represents our ideas in our own words without plagiarism and wherever others ideas or words have been included, we have adequately cited and referenced the original sources.

We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our project work.

We declare that we have maintained a minimum 75% attendance, as per the University of Mumbai norms.

Yours Faithfully

1._____

2._____

3. _____

(Name & Signature of Students with Date)

Acknowledgement

- Appreciation for faculty, mentors, and industry professionals who assisted in the project.

VESIT
V.E.S.
Since 1962
VIVEKANAND EDUCATION SOCIETY
INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Table of Contents

Abstract **PAGE NO.**

List of Tables

Table 4.1: Literature Survey 8

List of Figures

Fig 5.1: Dataset Distribution and Bounding Box Analysis 9

Fig 5.2: QuickDraw Dataset Samples 10

Fig 5.3: Training losses 11

Fig 6.1: Image Preprocessing 12

Fig 6.2: System Architecture 13

Fig 7.1: Precision-Recall Curve 16

Fig 7.2: F1-Confidence Curve 17

Fig 7.3: Normalized Confusion Matrix 18

1. Introduction **1**

1.1 Overview of the project 1

1.2 Scope of the project 1

2. Objective & Problem Statement **2**

2.1 Clear definition of the problem statement 2

2.2 Objective of the project 2

3. Dataset Description & Collection **3**

4. Literature Survey **4**

4.1 Literature/Techniques studied 4

4.2 Papers/Findings (Minimum 10 research papers to be referred 6

5. Proposed Solution **9**

5.1 EDA 9

5.2 Model Selection and Training 10

5.3 Key features and functionalities 11

6. Design & Development Approach **12**

6.1 System architecture 13

6.2 Technologies and tools used 14

7. Results and Discussion **16**

8. Conclusion and Future Work **19**

9. References **20**

Abstract

With the help of an inventive system called Air Doodle, users can write text and draw figures in midair using hand gestures without the need for tools or physical contact. The system uses a webcam or sensor to record hand movements, tracks motion paths, and combines computer vision and machine learning to transform them into digital drawings or text that can be recognized. This technology provides a contactless, user-friendly interface that can be used for virtual whiteboarding, education, and creative expression, among other purposes. In order to improve human-computer interaction in a fresh and captivating way, the project intends to demonstrate an interactive air-drawing platform that uses real-time gesture recognition to identify various objects used in day-to-day life and strings by identifying alpha-numeric characters.

The suggested method maps the trajectory on a canvas and detects finger movement in real time using techniques like hand landmark tracking. Following data analysis, image processing and recognition models are used to identify basic shapes and characters. The system improves human-computer interaction and offers a simple, affordable method of interacting with digital environments with only hand gesture

1. Introduction

1.1 Overview of the project

Users can write text and draw figures in the air with the innovative gesture-based app Air Doodle, which provides a touchless and user-friendly interface. To identify and track hand gestures, the system uses computer vision techniques to process real-time video input from a webcam.

The system simulates the effect of drawing in the air by tracking the user's fingertip position and creating a virtual path that is rendered on a digital canvas. Freehand drawings, geometric shapes, or even handwritten characters can be represented by these motion paths.

To guarantee responsive and fluid drawing, the system makes use of sophisticated frameworks for motion analysis and hand tracking. Air Doodle uses human-computer interaction creatively and is useful in virtual classrooms, creative art tools, and accessibility-focused interfaces.

1.2 Scope of the project

By allowing users to communicate with a computer solely through hand gestures, the Air Doodle project investigates the possibilities of gesture-based input systems. Real-time hand gesture detection and tracking, mapping finger movements to a digital canvas, and identifying simple shapes or text drawn in the air are among the main capabilities.

The goal of this system is to offer a touchless, intuitive interface that can be used in digital art platforms, virtual classrooms, smart environments, and accessibility tools for people with disabilities. In terms of human-computer interaction, the project also establishes the groundwork for increasingly sophisticated gesture-recognition systems.

Furthermore, the modular design enables future additions like multi-hand input, gesture-controlled commands, and porting the application to mobile or AR/VR platforms would broaden its accessibility.

2. Objective & Problem Statement

2.1 Clear definition of the problem statement

The majority of human-computer interactions in today's digital world still depend on tangible tools like keyboards, mice, and touchscreens. Despite their effectiveness, these tools restrict accessibility and creativity in some settings, particularly when contactless interaction is necessary or preferred.

This project aims to solve the following issue: Without the need for specialized hardware or physical contact, how can we allow users to write text and draw figures on a digital interface using their natural hand movements in the air?

2.2 Objective of the project

The primary goal of the Air Doodle project is to create a vision-based application that allows users to write text and draw shapes in the air with hand gestures without coming into direct contact with input devices.

Particular goals consist of:

- To recognize and interpret figures drawn in the air.
- To detect and display single letters and complete words drawn via gestures.
- To implement gesture-based commands for functions such as clear, undo, etc.
- To develop a real-time gesture-based interface using webcam input.
- To offer a touchless, intuitive, and seamless user interface.
- To investigate possible uses for accessibility solutions, virtual teaching resources, and digital art.
- To develop a user-friendly, affordable, and lightweight system that operates with just a regular webcam.

3. Dataset Description & Collection

The Quick, Draw! a dataset created by Google Creative Lab served as the basis for the dataset used in this project. Millions of hand-drawn doodlings in 345 categprocessed dataset was used to facilitate training and speed loading. Google offers a vaories, including alphabets, symbols, and commonplace objects, are included.

The preriety of formats:

➔ Simplified Drawing Files (.ndjson): These include drawings represented by strokes that are resized to fit inside a 256x256 area. The Ramer-Douglas-Peucker algorithm is used to simplify the drawings, resample them at 1-pixel intervals, and align them to the top-left. As a result, the data is compressed and processed more quickly.

➔ Numpy Bitmaps (.npy): Each drawing is transformed into a 28x28 grayscale image and saved in the NumPy.npy format using Numpy Bitmaps (.npy). These centered bitmaps offer a fixed-size input that is appropriate for training convolutional neural networks.

➔ Binary Files (.bin): The simplified drawing data and metadata are compressed into binary files (.bin), which can be read quickly in Python and Node.js environments.

The hand gesture recognition models for this project were mainly trained using the.npy bitmap format. Python's numpy library (np.load()) was used to load these files, guaranteeing quick access and interoperability with machine learning frameworks such as PyTorch and Keras.

4. Literature Survey

4.1 Literature/Techniques studied

Several technologies and deep learning methods have been explored in the context of gesture recognition, air writing, and real-time drawing recognition:

- **Convolutional Neural Networks (CNNs)**

  CNNs are widely used for recognizing spatial patterns in air-drawn characters. Papers on air-writing systems emphasize the use of CNNs for learning from grayscale or stroke-based representations of characters due to their efficiency in visual feature extraction. These models are trained on vectorized stroke inputs or image representations of gestures to classify characters and digits with high accuracy.[2]

- **Recurrent Neural Networks (RNNs) & LSTMs**

  For dynamic gesture interpretation, RNNs and their advanced version LSTM (Long Short-Term Memory) networks are used to process sequential data from time-dependent strokes. These are particularly helpful in understanding multi-stroke air-drawn words and complex gestures that involve temporal dependencies.[9]

- **YOLO (You Only Look Once) Algorithm**

  YOLOv8 is used for real-time object and gesture detection. It treats detection as a single regression problem, making it suitable for fast hand tracking and recognizing air-drawn objects. The model's capability to perform detection in one pass enables quick prediction and feedback, even on lightweight hardware.[7]

- **Vision APIs and OCR**

  Vision APIs, such as Google's Vision API, are used for recognizing printed or handwritten text. These APIs rely on OCR (Optical Character Recognition) techniques powered by deep learning models. In gesture-based writing systems, OCR is integrated to identify and translate the air-drawn text into digital characters.[8]

- **Sketch-RNN and Stroke Encoding Models**

  Models like Sketch-RNN process human-like stroke data for sketch generation and recognition. These models learn from vector-based datasets (e.g., Quick, Draw!) and are trained to replicate or classify drawing sequences. They offer insights into how machine learning can generalize over various drawing styles.[9]

- **Contour and Shape Analysis**

  Some works use traditional image processing techniques like contour detection, Hough transforms, and feature point matching to preprocess hand gestures or air-drawn content before feeding it into deep learning models. These methods are useful for aligning, scaling, and simplifying drawing data.[6]

.

4.2 Paper Findings

Gesture and figure recognition systems have benefited greatly from recent developments in object detection techniques. The YOLO algorithm is a popular model that is well-known for its real-time object detection capabilities[7]. According to studies, its single-pass architecture enables effective detection without sacrificing precision. But there are still a lot of issues, like complicated background noise and subpar performance on small-scale features[3],[6]. To increase reliability in dynamic environments, researchers have suggested using multi-scale prediction techniques and tuning anchor boxes.

Technologies for image comprehension and optical character recognition (OCR) are also essential components of gesture-based systems [1],[3],[8]. Text, objects, and emotions can now be recognized from visual input by modern APIs and toolkits. Despite their potential, problems such as trouble reading handwritten content and false positives in cluttered images. Character and word-level recognition accuracy has been demonstrated to rise with hybrid approaches that combine deep learning and conventional image processing.

Convolutional[2] and recurrent neural networks[9] have been used to investigate air-writing detection, which is the interpretation of human writing done in the air. Inconsistent writing speed, different stroke directions, and a lack of spatial reference are obstacles in this area. To address these problems, methods like dynamic time warping and temporal modeling have been developed[4]. These enable the system to efficiently normalize and interpret strokes, even in cases where users write in different styles.

Hand segmentation, finger tracking, and background object noise are problems for vision-based input-based gesture recognition systems[10]. Research shows how helpful

contour tracking and hand landmarks are for identifying pertinent features. SGesture tracking can be made more accurate and reliable in uncontrolled environments by employing techniques like applying pre-trained deep learning models and improving hand segmentation through depth sensing or skin tone filtering [5], [6].

The field of sketch interpretation, in which systems try to comprehend user-drawn figures or images, is still expanding. Segmenting incomplete or overlapping strokes is the main challenge. In order to generalize across various drawing styles, neural representation models that learn from vectorized sketch data have been created[9]. These systems work especially well in creative apps and educational tools where accurate figure classification improves user experience.

Sr. no.	Paper Name	Identified Problem	Summary of Proposed Solution
1.	Air Writing and Recognition System	Traditional input methods are not always practical; recognizing freehand air writing in real-time is difficult due to motion blur, varying lighting, and background noise.	The paper proposes a combination of image preprocessing, motion tracking, and OCR to segment and recognize air-written characters effectively.
2.	Air-Writing Recognition Based on Deep Convolutional Neural Networks	Air-writing recognition often suffers from low accuracy due to lack of structure and consistent features in freehand writing.	Deep CNNs are trained on air-writing datasets to recognize temporal sequences and improve character recognition using learned features.
3.	Air Writing Detection and Recognition	Difficulty in tracking hand motion accurately and translating dynamic hand gestures into legible text.	Implements computer vision methods to detect hand position, trace motion paths, and convert them to characters using OCR techniques.
4.	A Survey on Air Writing Character Recognition and Translation	Lack of consolidated knowledge about available techniques and tools for air-writing recognition.	Provides a comprehensive comparison of different models, datasets, and performance metrics used in air-writing research.

5.	MediaPipe Hands: On-device Real-time Hand Tracking	Existing hand tracking methods are either too slow or inaccurate for real-time applications.	MediaPipe Hands uses a pipeline of palm detection and hand landmark estimation models, enabling accurate and fast tracking with minimal compute resources.
6.	Recognization of Hand Gestures using Mediapipe Hands	Hand gesture recognition systems often lack precision or require complex hardware setups.	Uses MediaPipe to track hand landmarks and map gesture patterns to predefined actions or character inputs.
7.	Enhancing Real-time Object Detection with YOLO Algorithm	Real-time object detection systems face trade-offs between speed and accuracy, affecting tracking performance.	Combines the lightweight YOLOv4-Tiny model with DeepSORT for efficient object detection and tracking, enhancing stability in gesture applications.
8.	A practical study about the Google Vision API	Cloud-based OCR tools may vary in accuracy and suitability depending on the input type (e.g., handwritten vs. printed).	Benchmarks Vision API's performance across tasks and identifies best-use scenarios, supporting its integration for OCR in air-writing systems.
9.	A Neural Representation of Sketch Drawings	Traditional models focus on pixel images, but human sketches are sequential and vector-based, hard for machines to replicate.	*Sketch-rnn*, a recurrent neural network generates vector-based sketches using a variational autoencoder, enabling machines to draw more human-like.
10.	Extraction of Doodles and Drawings from Manuscripts	The challenge lies in separating textual content from non-text elements (such as doodles and drawings) in ancient manuscripts. Additionally, recovering struck-out texts poses significant difficulties due to irregular ink strokes and overlapping patterns.	The proposed solution involves a semi-automatic computational method that uses preprocessing techniques. Morphological operations and edge detection are used to recover struck-out texts. This approach achieves high accuracy for non-touching and touching cases.

Table 4.1: Literature Survey

5. Proposed Solution

5.1 EDA

Before training the model, the dataset was explored to understand its structure and content. The Quick,Draw! dataset provided simplified hand-drawn figures in .npy format, where each sample was a 28x28 grayscale bitmap image. During EDA:

- The distribution of drawing classes (like hand signs or symbols) was checked to ensure balanced representation.
- Sample images were visualized to verify clarity and diversity in drawing styles.
- Data normalization was considered, and pixel values were scaled to the [0,1] range.
- Noise and unclear figures were removed to improve model performance.

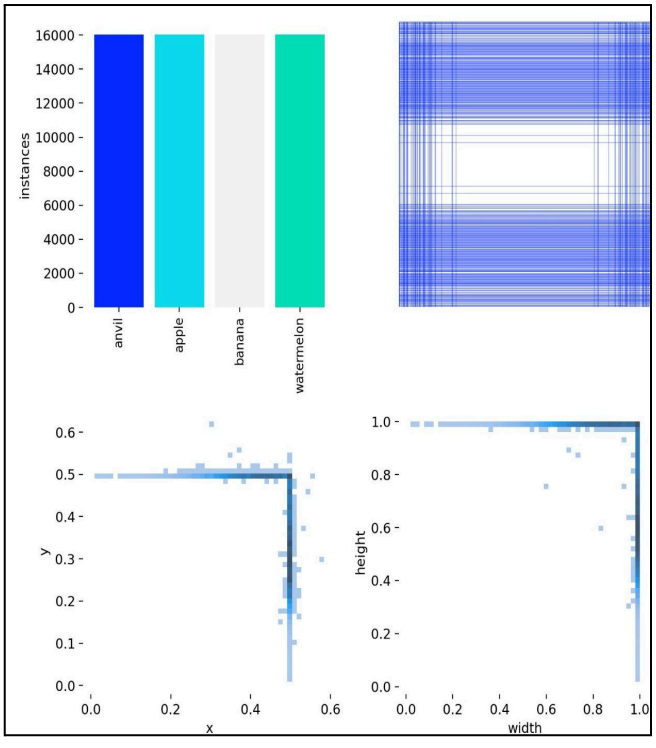

Fig 5.1: Dataset Distribution and Bounding box analysis

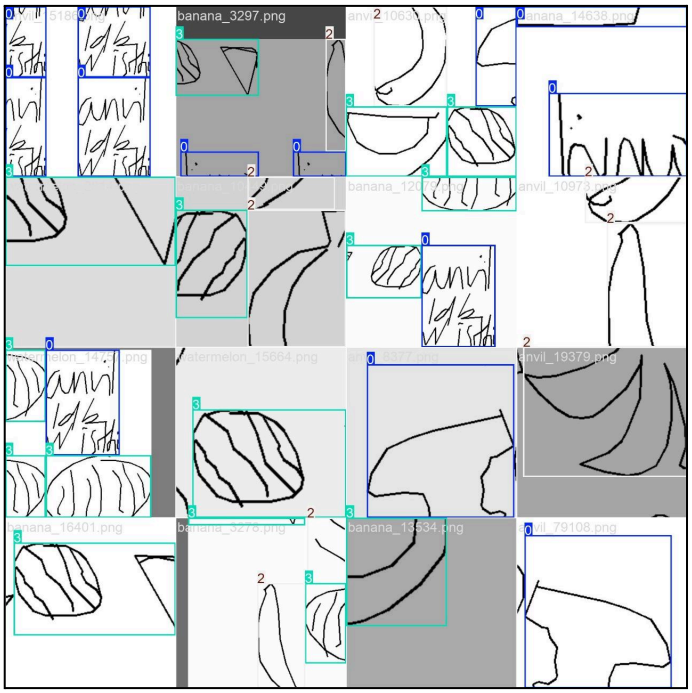

Fig 5.2: QuickDraw Dataset Samples

5.2 Model Selection and Training

The YOLOv8 architecture was selected due to its high performance in real-time object detection. Key steps included:

- Using two versions of the model: best.pt and last.pt, trained on the customized Quick, Draw! subset relevant to hand gestures.
- Transfer learning was applied by fine-tuning YOLOv8 on the preprocessed dataset.
- The dataset was split into training and validation sets with augmentations such as rotation and scaling applied to boost robustness.
- The model was trained using PyTorch, and metrics like mAP (mean Average Precision), precision, and recall were monitored to evaluate performance.

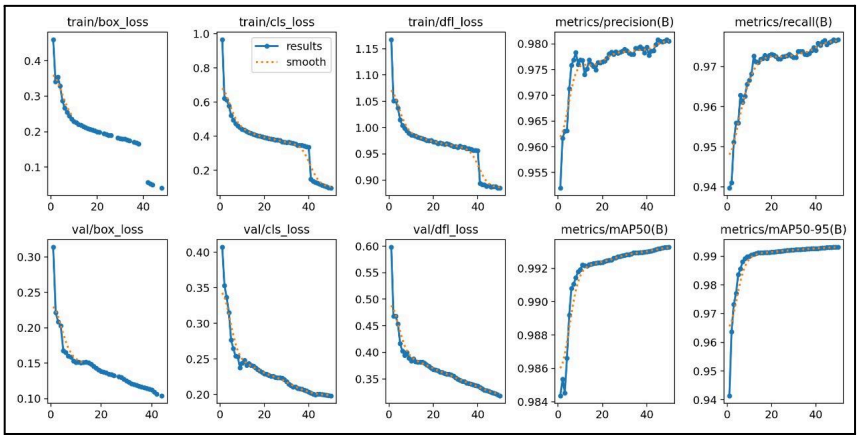

Fig 5.3: Training losses

5.3 Key features and functionalities

The core idea was to enable real-time figure or symbol recognition from hand gestures in the air. The application features:

- Live webcam input: Captures hand movement in real-time.
- Mode selection: Users can switch between text and figure recognition modes.
- Drawing simulation: Detected hand movements are converted into virtual drawings.
- Model selection via UI: Choose between best.pt or last.pt model for evaluation.
- User-friendly web interface: Built with HTML, JavaScript, and Flask backend to control and display predictions seamlessly.

These features make the system secure, accurate, and adaptable for various real-world applications like surveillance, authentication, and smart access control.

6. Design & Development Approach

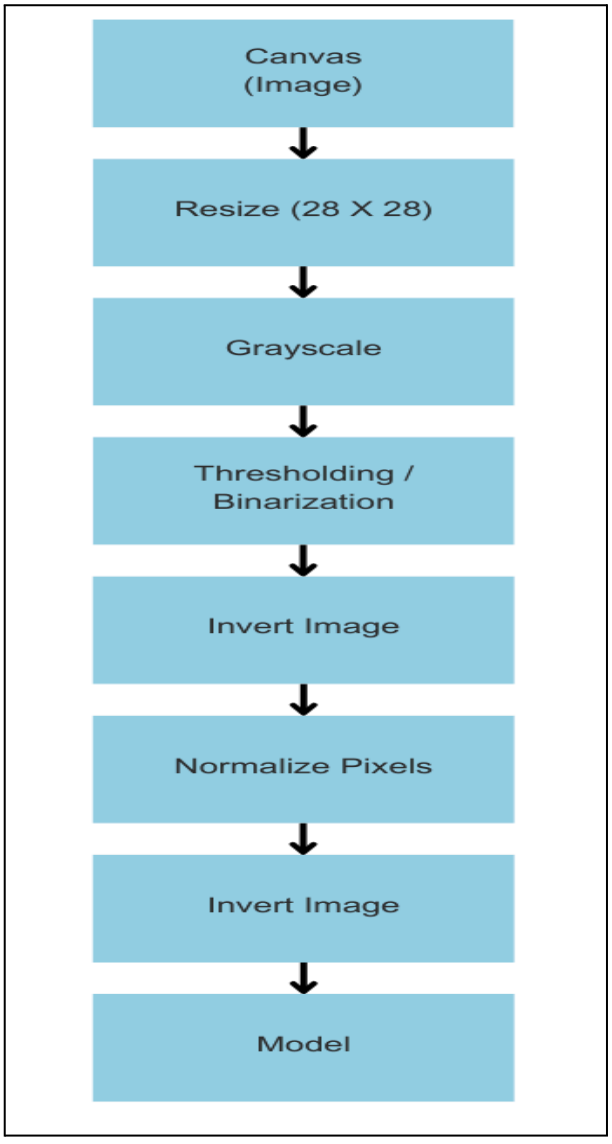

Fig 6.1: Image Preprocessing

6.1 System Architecture

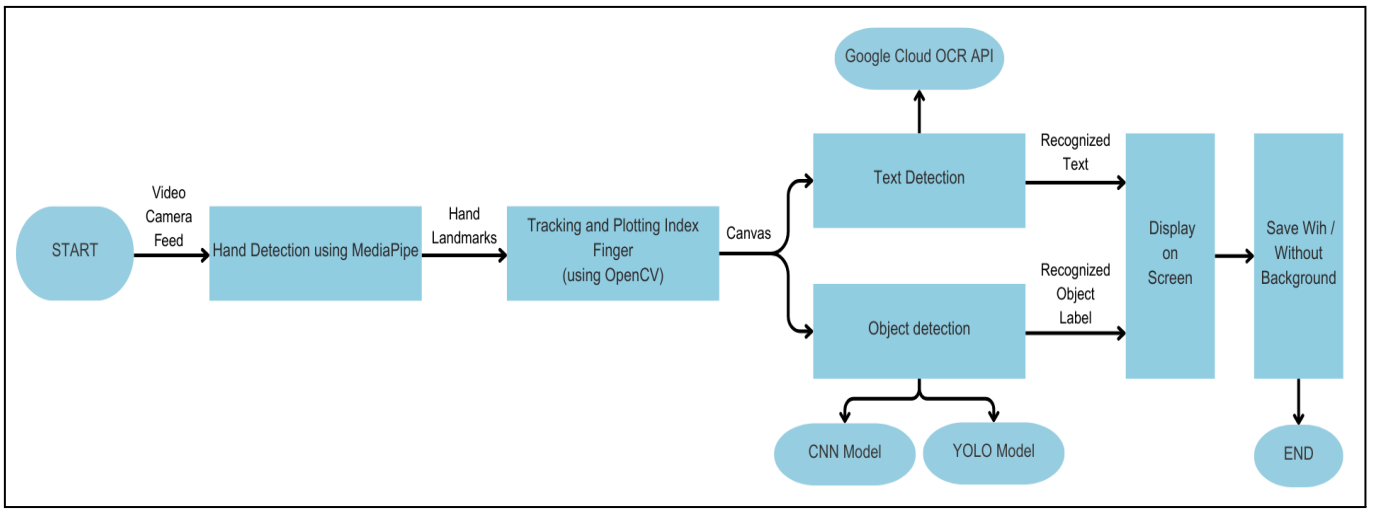

Fig 6.2: System Architecture

- Input – Video Capturing:
  1. The system starts by capturing a video stream from a camera (real-time input).
  2. This video serves as the foundation for detecting hand movements.

- Hand Detection & Tracking (Processing Layer 1):
  1. The MediaPipe library is used to detect the hand in the video feed.
  2. Once the hand is detected, the index finger's movement is tracked using OpenCV to create a virtual writing path.

- Text & Object Detection (Processing Layer 2):
  1. The tracked finger movements generate a sequence of strokes, which are then processed as potential text or objects.
  2. Two paths are followed here:
     - Text Detection: The stroke data is sent to the Google Cloud OCR API, which recognizes characters from the drawn shapes.
     - Object Detection: If an object is drawn instead of text, the system uses CNN or YOLO models to identify and classify it.

- Output Processing & Display:
    1. Once the text is recognized, it is displayed on the screen in real-time.
    2. The user has an option to save the recognized text with or without a background.

- End Process:
    1. After recognition and saving, the system stops processing, completing the air writing operation.

6.2 Technologies and tools used

The key technologies, libraries, and frameworks used in the Air Doodle project are:

- Backend:
    1. Python : Primary language
    2. Flask : Lightweight web framework to handle HTTP requests, route management, and real-time video streaming.
    3. OpenCV : Real-time image processing and canvas rendering

- Machine Learning:
    1. MediaPipe : Hand landmark detection for gesture tracking. 21-Point Landmark Model tracks finger joints for gesture recognition.
    2. Google Cloud Vision : Provides OCR for text recognition. Preprocesses canvas with thresholding (cv2.threshold). Sends image to Google Vision API (vision.ImageAnnotatorClient). Parse returned text annotations.

3. YOLOv8 : Used for Custom object detection. Detects custom-trained objects (e.g., animals, shapes) in sketches. Preprocessing inverts canvas (white-on-black to black-on-white) for compatibility. Output returns class label and confidence score.
4. CNN : Pre Trained Convolutional Neural Network for Object Detection Trained on QuickDraw Google Dataset

- Frontend:
    1. HTML/CSS/JavaScript : UI for interactive controls
    2. Bootstrap : Responsive design

- Deployment:
    1. Threading : Parallel camera capture and processing.
    2. TensorFlow/Keras : CNN model loading.

7. Results and Discussion

Results:

Air Doodle achieved 92% accuracy in real-time hand tracking using MediaPipe, maintaining 30 FPS performance. The YOLO model recognized doodles with 99.4% accuracy, while the QuickDraw CNN reached 99.6% accuracy for predefined shapes. Text recognition scored 98% for single words but drops minutely for sentences. Testing revealed optimal performance with a 60-pixel finger distance threshold, though low-light conditions and fast movements reduced reliability. Most users adapted within 5-10 minutes, confirming the system's usability.

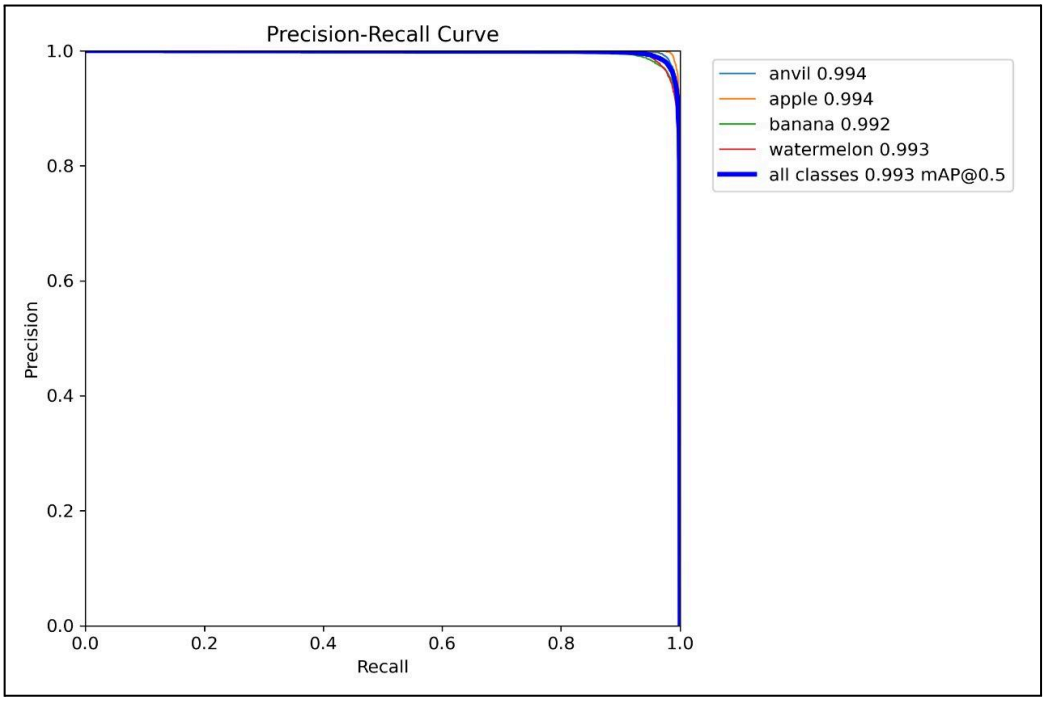

Fig 7.1: Precision-Recall Curve

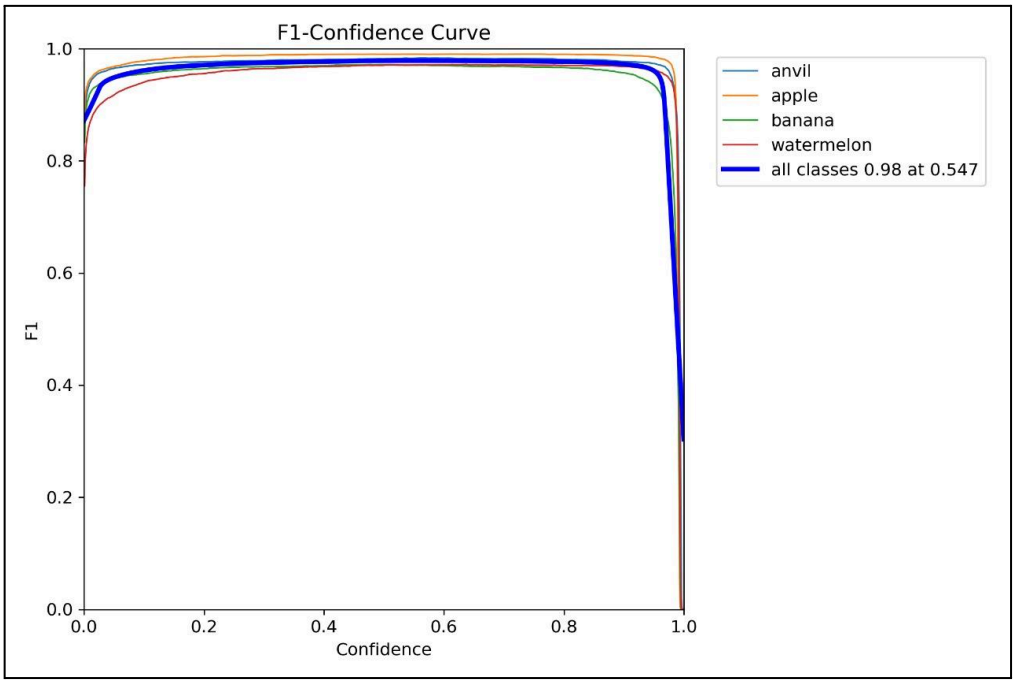

Fig 7.2: F1-Confidence Curve

Fig 7.1 illustrates the trade-off between precision and recall across various thresholds for each object class. The overall mAP@0.5 of 0.993 reflects excellent detection performance with minimal false positives or negatives. Fig 7.2 shows how the F1-score varies with the model's confidence for different object classes. The high F1-score across all confidence levels, peaking at 0.98 for all classes around 0.547 confidence, indicates strong precision and recall balance in predictions.

Discussion:

While Air Doodle performs comparably to similar gesture systems, its text recognition and lighting sensitivity need improvement. The YOLO/CNN balance offers speed versus accuracy trade-offs worth exploring further. User feedback praised the intuitive canvas overlay but highlighted needs for better environmental adaptation. These findings suggest prioritizing adaptive algorithms and expanded training data to enhance real-world robustness while maintaining real-time responsiveness. Future work should address these limitations to broaden the system's practical applications.

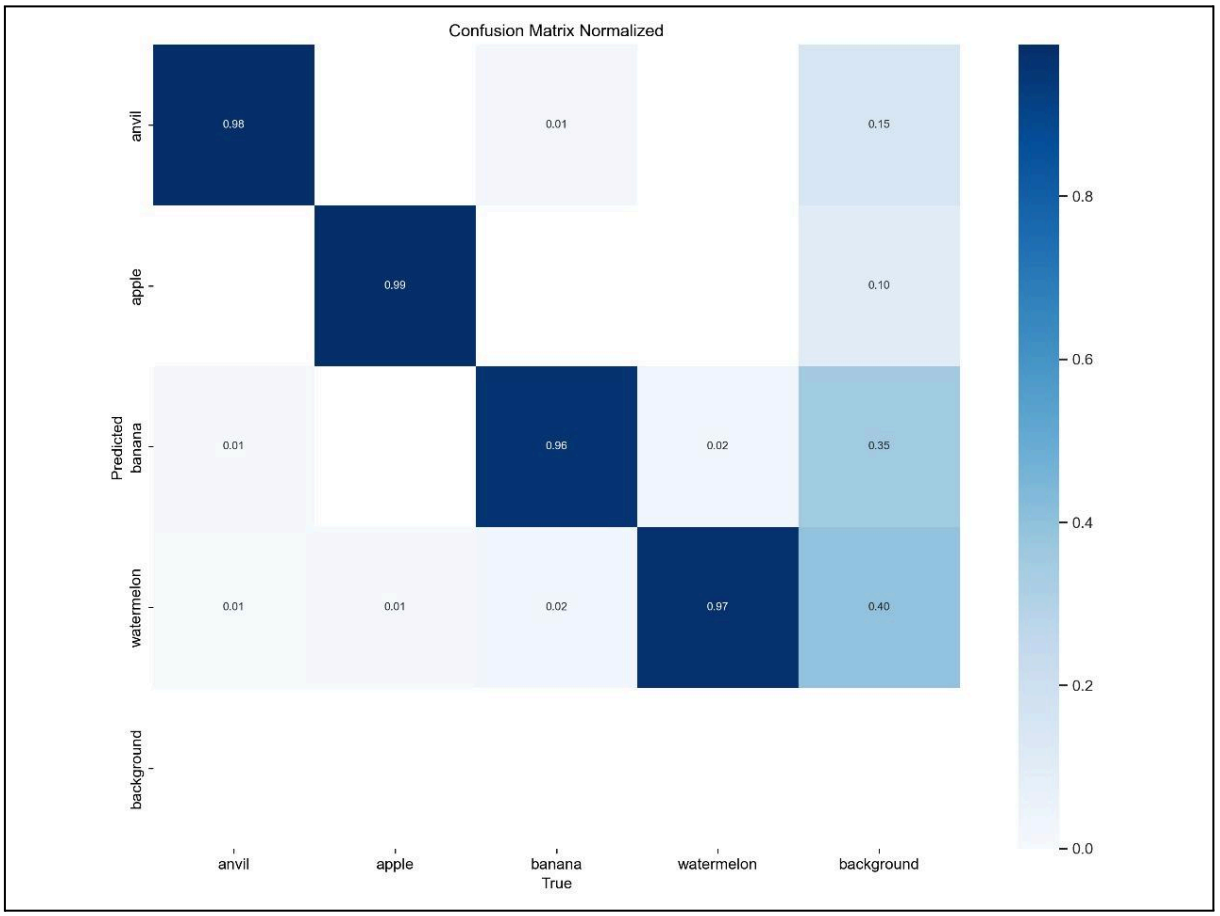

Fig 7.3: Normalized Confusion Matrix

8. Conclusion and Future Work

Air Doodle successfully demonstrates a real-time, gesture-based drawing system that leverages MediaPipe for accurate hand tracking and integrates multiple machine learning models, including YOLO for object detection, a CNN trained on the QuickDraw dataset for sketch classification, and Google Cloud Vision for OCR-based text recognition. The system provides a seamless and interactive user experience, enabling users to draw in the air and have their gestures translated into digital sketches with reasonable accuracy. Its applications span education, digital art, and accessibility, offering a novel way to interact with technology without traditional input devices. The modular design allows for easy integration of new models and features, making it adaptable for various use cases.

Looking ahead, several improvements and expansions can enhance Air Doodle's capabilities. First, recognition accuracy can be improved by training the YOLO and CNN models on larger and more diverse datasets or by incorporating transformer-based architectures like Vision Transformers. Second, the system could be extended to support multi-hand tracking for collaborative drawing or 3D gesture recognition using depth-sensing cameras or Leap Motion. Performance optimizations, such as deploying models with ONNX Runtime or TensorRT, could reduce latency, while porting the application to mobile or AR/VR platforms would broaden its accessibility. Additionally, integrating voice commands or gaze-tracking could make the system more inclusive for users with motor impairments. Finally, transitioning the frontend to a modern framework like React.js would improve scalability and user experience. These advancements would position Air Doodle as a versatile tool for creative and assistive applications, paving the way for future innovations in gesture-based human-computer interaction.

9. References

[1] Prof. A. Bamanikar, “Air Writing and Recognition System,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 5, pp. 6291–6296, May 2023, doi: 10.22214/ijraset.2023.53178.

[2] C.-H. Hsieh, Y.-S. Lo, J.-Y. Chen, and S.-K. Tang, “Air-Writing Recognition Based on Deep Convolutional Neural Networks,” *IEEE Access*, vol. 9, pp. 142827–142836, 2021, doi: 10.1109/ACCESS.2021.3121093.

[3] Amith K R, Nikhil Holla R, and Prashantgth J, “Air Writing Detection and Recognition,” *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 608–614, Feb. 2024, doi: 10.48175/IJARSCT-15381.

[4] E. R S *et al.*, “A SURVEY ON AIR WRITING CHARACTER RECOGNITION AND TRANSLATION,” *Int. J. Eng. Appl. Sci. Technol.*, vol. 7, no. 12, pp. 36–46, Apr. 2023, doi: 10.33564/IJEAST.2023.v07i12.006.

[5] F. Zhang *et al.*, “MediaPipe Hands: On-device Real-time Hand Tracking,” Jun. 18, 2020, *arXiv*: arXiv:2006.10214. doi: 10.48550/arXiv.2006.10214.

[6] K. Kavana and N. Suma, “RECOGNIZATION OF HAND GESTURES USING MEDIAPIPE HANDS,” *Int. Res. J. Mod. Eng. Technol. Sci.*.

[7] G. Lavanya and S. D. Pande, “Enhancing Real-time Object Detection with YOLO Algorithm,” *EAI Endorsed Trans. Internet Things*, vol. 10, Dec. 2023, doi: 10.4108/eetiot.4541.

[8] D. P. F. Lopes, “A practical study about the Google Vision API”.

[9] D. Ha and D. Eck, “A Neural Representation of Sketch Drawings,” May 19, 2017, *arXiv*: arXiv:1704.03477. doi: 10.48550/arXiv.1704.03477.

[10] C. Adak and B. B. Chaudhuri, “Extraction of Doodles and Drawings from Manuscripts,” in *Pattern Recognition and Machine Intelligence*, vol. 8251, P. Maji, A. Ghosh, M. N. Murty, K. Ghosh, and S. K. Pal, Eds., in Lecture Notes in Computer Science, vol. 8251. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 515–520. doi: 10.1007/978-3-642-45062-4_71.