**MY-RES PROJECT 2024**

**MAGIC MIRROR**

**INTRODUCTION**

This document is about the Magic Mirror application, specifically what it does and how it works. The Magic Mirror is designed to identify objects in real-time and provide the user with the respective AUSLAN sign, with the aim to help users to learn AUSLAN. The application opens with a ‘Welcome’ page, where the user can see themselves and then they can move into either ‘Learning’ or ‘Game’ mode by pressing on the corresponding button. In ‘Learning’ mode, the user can hold up an object to the camera, it will be identified, and then a video of the AUSLAN sign for the object will overlay on the live feed. Each object that is learnt will be listed on the left-hand side of the screen. In ‘Game’ mode, the AUSLAN sign will again be played over the live feed, and then the user can hold up the correct object to win points. The user can toggle ‘Hints on’ or ‘Hints off’ depending on whether they want the word of the object to be displayed or not, and there is also a ‘Skip’ button that allows the user to skip that object. The signs used in this prototype are in the ‘videos\_anz’ folder.

**UNDERSTANDING THE COMPONENTS**

**IMPORTS:**

* ‘tkinter’: Provides the GUI framework
* ‘PIL’ (Pillow): Used for image processing (converting OpenCV images to a format that ‘tkinter’ can display)
* ‘cv2’ (OpenCV): Handles video capture and image processing
* ‘numpy’: Used for numerical operations (e.g. choosing random objects)
* ‘threading’: Manage threads, allowing video playback in parallel with the main program
* ‘YOLO’ (Ultralytics): The pretrained model for real-time object detection. This program uses the ‘yolov8n.pt’ model

**MEMORYGAME CLASS:**

* The ‘MemoryGame’ class initialises the YOLO model for object detection and sets the confidence threshold, this being the parameter that determines the minimum confidence level required for the model to consider a detection valid. This threshold helps filter out low-confidence detections, reducing false positives and ensuring that only more certain detections are used. The initialisation also opens a video capture using the webcam on the device (laptop/PC) and sets up the GUI window with control buttons

**GUI COMPONENTS:**

* ‘self.root’: Main application window
* ‘self.video\_frame’: A ‘tkinter’ frame used to display the video feed
* ‘self.control\_frame’: Contains control buttons for interacting with the game modes

**MODES AND ACTIONS:**

* Modes:
  + Welcome: Default mode when the program starts, waiting for the user to select a mode
  + Learning Mode: The program detects and learns objects from the webcam feed
  + Game Mode: The user must identify previously learned objects
* Buttons:
  + Learn Button: Switches to learning mode
  + Game Button: Switches to game mode if there are learned objects
  + Quit Button: Exits the application
  + Skip Button: Skips the current object in game mode and selects a new one
  + Show Text Button: Toggles hints on and off during game mode

**OBJECT LEARNING AND GAME LOGIC:**

* Object Detection (‘predict’ method): Uses YOLO to detect objects in the video feed
* Learning Objects: Detected objects (excludes ‘person’) are added to ‘learned\_objects’
* Video Playback (‘play\_video’ method): Plays a video associated with a learned object if it exists
* Game Mode: The program randomly selects a learned object for the user to identify, and if the object is correct the score increases

**HELPER METHODS:**

* ‘resize\_window\_proportionately’: Resizes the window based on the aspect ratio
* ‘toggle\_show\_text’ and ‘update\_hint\_button\_text’: Manage hint visibility
* ‘choose\_next\_object’: Selects the next object to be identified in game mode
* ‘skip\_object’: Allows the user to skip the current object

**SUGGESTED IMPROVEMENTS**

This prototype was tested and the following improvements were suggested:

* Reduce the lag of the live feed/video playback (this could be due to the way the videos are loaded in
* Blur background on live feed to ensure the camera only detects objects in close proximity
* Print out images during testing (however this is not ideal as users will be holding real objects to the camera)
* Introduce more objects
  + Utensils, laptop, chair, etc. Doesn’t need to only be handheld objects (in future)
* Refine database library to improve the object identification (sometimes a fan in the background is identified as a plane)

**CONCLUSION**

This program creates an interactive application where users can teach the system to recognize objects and later test their memory by identifying these objects. The program utilizes object detection through YOLO, handles real-time video processing, and provides a user-friendly interface for interaction.

**INSTALLING THE LIBRARIES**

Here is a guide on how to download and install popular Python libraries used in this application, including Tkinter, Pillow, OpenCV, and YOLO with pip, Pythons’ package installer. This should provide the general gist of how and what to download, but there may be some troubleshooting required based on your machine.

Firstly, ensure that Python and pip are installed on your system. Python 3.x is recommended as it's the latest and supported version. You can verify their installation by using:

python --version

pip --version

If these commands don't work, you may need to use python3 and pip3 instead.

**Step-by-Step Installation:**

1. Installing Tkinter

Tkinter is included with Python installations by default. If it’s not available, you can install it using your system’s package manager:

sudo apt-get install python3-tk # For Debian/Ubuntu

brew install python-tk # For macOS using Homebrew

2. Installing Pillow (PIL)

Pillow is the Python Imaging Library, used for opening, manipulating, and saving many different image file formats.

pip install pillow

3. Installing OpenCV

OpenCV is used for computer vision tasks like capturing video from the camera.

pip install opencv-python-headless

If you need GUI features to display images or videos, install with:

pip install opencv-python

4. Installing NumPy

NumPy is a fundamental package for numerical computation in Python.

pip install numpy

5. Installing YOLO

YOLO can be installed via pip, depending on the version you need. For YOLOv8, you can install it directly from the Ultralytics package:

pip install ultralytics

**Additional Tips**

Virtual Environments: It is highly recommended to use Python virtual environments to manage dependencies and avoid conflicts between projects.

python -m venv env

source env/bin/activate # On Unix/macOS

.\env\Scripts\activate # On Windows

Checking Installation: After installation, check that everything is working by importing the libraries in Python:

import tkinter as tk

from PIL import Image

import cv2

import numpy as np

from ultralytics import YOLO

**Note:** Before starting the installation, it's good practice to update pip to its latest version:

pip install --upgrade pip

**AUSLAN VIDEOS**

This project relies on videos from <https://www.nzsl.nz/>. These were compared to signs from <https://auslan.org.au/> to ensure that the signs are used in AUSLAN. These videos can be saved into a file called ‘video\_anz’ and ensure the name of the videos are the name of the object that they are connected to. For example, if the object is a bottle the name of the corresponding video would be ‘bottle’.

**RESEARCH PAPERS/SOURCES**

Alexander Kim – ‘Magic Mirror: Assessing Social Validity of a AI-Powered Smart Mirror for People with Intellectual Disabilities to Improve Independence in Daily Tasks’

Yu-Tseng Wu – ‘A robust framework for recognizing the meaning of objects in video streams’

Van Vu Kieu – ‘A real time hama object detection API using classification method and transfer learning approach’

Chih-Hsueh Liao – ‘A solid real-time API for customized dataset based on YOLOv5’

Weihao Yuan – ‘The role image repair plays in visual recognition technology.’

Eleni Orfanidou – ‘Research methods in sign language studies : a practical guide’ (ISBN : 1-118-34596-7)

Annelies Kusters, Mara Green. Erin Moriarty and Kristin Snoddon – ‘Chapter Sign language ideologies: Practices and politics’ (ISBN: 9781501510090)

Akshay Kore – ‘Designing Human-Centric AI Experiences : Applied UX Design for Artificial Intelligence’ (ISBN : 1-4842-8088-1)

Annelies Kusters – ‘Sign language ideologies in practice’ (ISBN : 1-5015-1002-9)