Mahmoud Saeed Mansour

☎ +20 106 376 1694 ✉ mahmoudsaidmansour1000@gmail.com in mahmoud-mansour

Education

Pharos University 2022 – 2026
Bachelor of Science in Computer Science & Artificial Intelligence, GPA:3.74 *Alexandria , Egypt*

Projects

Image Viewer with Filters and Classification | *Python, PyQt5, OpenCV, TensorFlow, CNN, Image Processing*

- Developed a desktop GUI application using PyQt5 for interactive image loading and visualization.
- Implemented multiple image processing techniques including Canny edge detection, Median filtering, Otsu thresholding, Harris corner detection, and SIFT feature extraction.
- Trained and integrated a CNN model to classify images into six scene categories: buildings, forest, glacier, mountain, sea, and street.
- Enabled real-time side-by-side visualization of original and processed images within the same interface.
- Packaged a pre-trained model and dataset files to allow immediate usage without retraining.

Tic Tac Toe Game | *Dart, Flutter*

- Developed a mobile application using the Flutter framework with two play modes: against the bot or against a friend. Implemented an AI algorithm for the bot and integrated win/lose detection with options to replay or exit the game.

Face Mask Segmentation during COVID-19 | *Python, CNN, TensorFlow, Keras, Flask, HTML, CSS*

- Built a binary semantic segmentation model for face mask detection using ResNet50 and VGG16 as feature extractors. Implemented custom upsampling layers with ReLU and a sigmoid-activated output for accurate segmentation.
- Trained the model on COVID-19 image datasets (train/validation/test splits), saved as `model.h5`, and deployed via a Flask web app with a custom HTML/CSS interface for real-time mask detection.

Tic Tac Toe Game with AI | *Python, PyQt5, OOP, Minimax Algorithm*

- Developed a single-player Tic Tac Toe game in Python with an interactive PyQt5 GUI. Integrated the Minimax algorithm for a smart AI opponent that makes optimal moves.
- Added symbol selection, dynamic board updates, gradient UI elements, and message box alerts for game results.

Real-time Sign Language Recognition | *Python, OpenCV, MediaPipe, Flask, scikit-learn*

- Built a real-time ASL recognition system using MediaPipe Hands and OpenCV for gesture detection and visualization.
- Extracted and normalized hand landmarks, stored in Pickle, and trained a Random Forest model to classify 14 signs with high accuracy.
- Deployed the model in a Flask web app with live webcam feed, intuitive overlays, and custom-colored landmarks for user-friendly feedback.

Real-time Object Detection | *Python, YOLOv8, OpenCV, Tkinter*

- Developed a YOLOv8-based object detection system for self-driving applications with support for webcam and video input.
- Created a Tkinter GUI with FPS display, bounding boxes, class labels, and efficient threaded video streaming for real-time performance.

Real-time Lane Detection | *Python, ResNet50, Kalman Filter, Hough Transform, OpenCV, Tkinter*

- Implemented multiple approaches for lane detection: segmentation with ResNet50, Kalman filter tracking, and Hough transform.
- Developed a Tkinter GUI with OpenCV integration for real-time lane detection, FPS monitoring, and annotated video streaming.

Real-time Traffic Light Detection | *Python, YOLOv8, PyTorch, OpenCV*

- Trained a custom YOLOv8 model on a traffic light dataset to classify signals as Green, Yellow, or Red. Saved the model in ONNX format for cross-platform deployment.
- Implemented real-time detection with bounding boxes, color mapping, and decision-making logic for traffic signals.

Real-time Traffic Signs Detection | *Python, TensorFlow, Keras, OpenCV, scikit-learn*

- Built and trained a CNN model to classify 43 classes of traffic signs with a Softmax-activated output layer.
- Saved the trained model as `.h5` and integrated it with OpenCV for real-time classification from video streams.

Travel Planner Chatbot | *Python, Transformers, HuggingFace, FAISS, Streamlit, SpeechRecognition*

- Developed an AI-powered chatbot for travel planning using Retrieval-Augmented Generation (RAG), combining speech-to-text, semantic search, and large language models.
- Integrated **Qwen2.5 LLM** fine-tuned on the OSU TravelPlanner dataset with a **FAISS** vector database for efficient information retrieval.
- Implemented voice input with SpeechRecognition API and built an interactive Streamlit UI for real-time queries.
- Enabled the chatbot to generate multi-day itineraries with explanations (justifications) for each recommendation.

EmpathyBot: Sentiment-Driven Chat | *Python, HuggingFace Transformers, FAISS, Streamlit, TweetEval*

- Built a sentiment-aware chatbot that classifies user emotions into six categories (joy, sadness, anger, love, surprise, fear) using a pretrained transformer (TweetEval).
- Integrated RAG to retrieve semantically similar tweets with SentenceTransformers and FAISS, enriching responses with context.
- Applied few-shot prompting by augmenting model inputs with (tweet, label) pairs to improve explainability.
- Created a Streamlit web UI displaying emotion prediction, retrieved examples, and chatbot explanations; deployable via Colab/ngrok.

Smart License Plate Detection with OCR | *Python, YOLOv8, Haar Cascade, OpenCV, EasyOCR, PyQt5*

- Developed a hybrid system to detect and recognize Arabic license plates using YOLOv8 and Haar Cascade.
- Applied preprocessing and EasyOCR to extract Arabic text and map it to Latin equivalents for broader usability.
- Integrated edge detection (Canny, Sobel) for enhanced visualization and robustness under noise.
- Built a PyQt5 GUI for users to upload images and select YOLO or Haar-based detection with real-time annotated results.
- Achieved strong accuracy with YOLO across diverse conditions, while Haar Cascade provided faster but less robust detection.

Smart Car Parking Monitoring System | *Python, OpenCV, YOLOv8, DeepSORT, Object Detection*

- Built a real-time vehicle detection and tracking system for monitoring parking lot occupancy using YOLOv8 and DeepSORT.
- Defined polygonal regions to accurately detect and count vehicles inside designated parking zones.
- Displayed bounding boxes, object IDs, and occupancy counts directly on the video stream.
- Generated annotated output videos for analysis and reporting purposes.
- Designed the system to work with recorded surveillance videos or live camera feeds for automated parking management.

Bitcoin Price Prediction Web Application | *Python, PyTorch, LSTM, GRU, Flask, Time-Series Forecasting*

- Developed deep learning models (LSTM and GRU) to predict future Bitcoin prices for multi-day horizons including 7, 15, 30, 60, and 90 days.
- Implemented full data preprocessing pipeline including scaling, sliding window creation, and inverse transformation of predictions.
- Built a Flask-based web application and REST API to serve real-time predictions and statistical insights such as min, max, and expected price change.
- Integrated automatic model loading with a fallback demo mode when trained weights are unavailable.
- Designed the system with modular separation between data processing, model inference, and web interface for scalability.

Technical Skills

Programming Languages: Python, Java , Dart , C#
Software: Dart , Flutter , PyQt5 , tinkerkad , Flask , Java-GUI
AI/ML& python-libraries:CNN , tensorflow , pytorch , open-cv , sklearn , scipy , MediaPipe , scikit-learn , numpy , pandas , ultralytics(yolo)
Concepts:Artificial Intelligence, Machine Learning, Neural Networks, DeepLearning , NLP , Data Science
Operating Systems:: Windows, Linux (Ubuntu)

Training

Amit January 2024 – August 2024
Machine Learning & AI Diploma *Alexandria, Egypt*

New Horizons September 2023 – January 2024
Flutter Application Development *Alexandria, Egypt*

European Educational Group July 2024 – September 2024
Deep Learning *Alexandria, Egypt*

National Telecommunication Institute(NTI) August 2025 – September 2025
Natural Language Processing *Online*

Languages

Arabic: Mother Tongue
English: Intermediate