Fazeel Asghar

Date of birth: 19/08/2001

Nationality: Pakistani

Gender: Male

CONTACT

📍 Basti Mian Mohammad Islam, Chak 4/p, Khanpur, 64100 Khanpur, Pakistan **(Home)**

📍 House Opposite to Ghausia Masjid, Ghausia Colony, One unit, Bahawalpur, Pakistan, 63100 Bahawalpur, Pakistan **(Hostel)**

✉️ fasghar40@gmail.com

🏠 (+92) 3126690033

ABOUT ME

AI Engineer | ML & DL Scientist | Computer Vision & IoT Developer

I'm an AI Engineer passionate about crafting intelligent solutions. With expertise in **Machine Learning (ML)** and **Deep Learning (DL)**, I specialize in developing algorithms that extract insights from data. I also thrive in Computer Vision, creating solutions for image analysis and object recognition. As an IoT Developer, I connect physical devices to the digital world, enhancing efficiency across various domains. With a focus on innovation and a knack for problem-solving, I'm excited to contribute to groundbreaking projects.

EDUCATION AND TRAINING

20/10/2021 – CURRENT Bahawalpur

Bachelors Of Sciences in IT. Islamia University of Bahawalpur

Website https://www.iub.edu.pk/

14/07/2019 – 28/11/2021 Khanpur, Pakistan

Fsc Pre-Engineering Punjab Group of Collges, Khanpur Campus

Website https://pgc.edu/campus/khanpur/

17/04/2017 – 15/07/2019 Khanpur, Pakistan

Matriculation Government High School Colony Khanpur

LANGUAGE SKILLS

MOTHER TONGUE(S): Urdu | Saraiki

Other language(s):

English

Listening C1	**Spoken production** B2
Reading C1	**Spoken interaction** B2
Writing B2	

Levels: A1 and A2: Basic user; B1 and B2: Independent user; C1 and C2: Proficient user

DIGITAL SKILLS

OpenCV | Python | Computer Vision | Deep Learning | Keras | IoT Developer | Data Science | Matplotlib | Tensorflow | Numpy | Machine Learning | C++

ADDITIONAL INFORMATION

Projects

26/01/2024 – CURRENT

Urine Sediment Particles Detection: I classified 7 urine sediment particles from microscopic slides dataset using a **Deep Learning** approach, achieving an **85%** accuracy rate. Later, I employed **YOLOv5** for particle detection.

The entire process involved:

- **Matplotlib**: Utilized for plotting various statistical graphs, such as accuracy graphs and confusion matrices.
- **Numpy**: Provided support for numerical computation.

- **Cv2**: Used for image resizing and detecting urine particles.
- **TensorFlow**: Employed for deep learning using CNN, incorporating cross-validation techniques.
- **TensorFlow.keras**: Utilized the Keras API for model building and training.

Following this, **YOLOv5** was employed to handle the entire training and detection process.

18/05/2023 – CURRENT

Wheel Chair Automation (Car Model): This model will detect object in front of it and tries to avoid it using **Computer vision** with the help of **Arduino** Uno using different sensors using **Python's pyserial** library.
Following are the dependent devices:
- Arduino Uno
- 2x Infrared Sensors
- 1x Ultrasonic Sensors.
- Breadboard
- Jumper wires.
- 2x Batteries

Further on, we can embed different **vital signs** monitoring devices in it for monitor critical patients.

08/07/2023 – 08/09/2023

Skin Disease Classifcation: I've classified binary skin disease classification **(Melanoma and Seborrheic)** using Deep Learning and achieved 85%+ accuracy and integrated to Django.

I employed following libraries and techniques:
- **Opencv (Cv2)**: for image processing:- Image-resizing and for detection and displaying.
- **Seabourn and Matplotlib**: for data visualization-:- accuracy graphs, evaluation etc.
- **Pandas:** to fetch label from files and integrate labels to each image.
- **Tensorflow and Keras:** for training model using **CNN algorithm.**
- **Django:** for **webservice** integration

04/10/2023 – 09/12/2023

Bulb Automation: Developed Bulb ON and OFF projects by **Chatbot**, **Computer Vision** and by **Sound Sensor**.
By Sound Sensor:
- Arduino Uno
- Python
- Python's Pyserial library
- Breadboard
- Jumper wires
- Relay Module

By Computer Vision:
- **Mediapipe** to get the face mesh with it's coordinates to access.
- **Opencv** to access webcam and do basic cv tasks.

By Chatbot:
- **Pyttsx3** for bot to speak.
- **SpeechRecognition** for speech recognition as we speak.

05/02/2024 – 03/05/2024

Voice Assistant: I've developed a **Voice Assistant** using Python, tailored to perform various internal OS tasks such as **opening drives**, **browsers**, **playing** music, and accessing the **webcam**. Here's how it's built using Python and its libraries:

- **Pyttsx3**: This library is utilized for speech synthesis, allowing the assistant to speak.
- **OpenCV2**: Used for accessing Computer Vision capabilities, enabling tasks like accessing the webcam.
- **Wikipedia**: Integrated to access relevant information.
- **Playsound**: This library facilitates playing music and sounds.
- **SpeechRecognition**: Essential for enabling the assistant to recognize and process speech in real-time as we speak.

Text Sentiment Detection: The Sentiment Detection project, dataset downloaded through Kaggle, is trained using three different models:

1. 1D - CNN
2. BI-LSTM
3. Bert

To accomplish this, we've employed several libraries and techniques:

- **Pandas**: This is used for accessing, organizing, and preparing the dataset.
- **Numpy**: It's utilized for numerical computations.
- **NLTK**: This library helps with text preprocessing tasks like tokenization and removing stop-words.
- **Scikit-learn** (**Sklearn**): We've used this for measuring metrics and for evaluation purposes.
- **Keras** and **TensorFlow**: These are essential for implementing different algorithms and techniques within the project.

18/07/2023 – 14/08/2023

Indian Car License Plate Recogntion This project focuses on **recognizing license plates** using advanced techniques for analyzing images. We've built it using Python and several helpful libraries. Our process involves **finding** license plates in images and then **blurring** out those areas. The entire project comes with a **user**-**friendly web interface** that's easy to interact with, thanks to Django.

We've used the following libraries:

- **Matplotlib**: for displaying images.
- **NumPy**: for doing math stuff.
- **cv2**: for working with images.
- **TensorFlow**: for the advanced learning part using Sequential approach.
- **TensorFlow**.**keras**: for building and training models.
- **Django**: for making the project accessible through the web.