MAHESH RAMESH DESAI

LinkedIn GitHub mdesai3@buffalo.edu +1 7162928577

EDUCATION

University at Buffalo, The State University of New York — **August 2021 – February 2023**
Master of Science in Computer Science and Engineering — Buffalo, NY

University of Mumbai — **August 2017 – May 2021**
Bachelors of Computer Engineering — Mumbai, India

TECHNICAL SKILLS

Programming – C, C++, Python, JavaScript, Java, React JS, HTML, CSS, SQL, Typescript, JSON, MATLAB
Databases and Framework - Flask, Django, NodeJS, MySQL, Mongo DB, Hadoop, Digital Ocean Cloud
Software and Tools– Docker, MapReduce, Shell script, AutoCAD, Raspberry PI, VS Code, ETL, REST API, GitHub

EXPERIENCE

Institute for Artificial Intelligence and Data Science-SUNY Buffalo — Buffalo, NY
Graduate Research Assistant — **August 2022 – Present**

- Performed junction point elimination and employed **Otsu thresholding** to extract skeleton of Org Chart figure, facilitating the implementation of connected component algorithm to find relation between nodes
- Performing **Name Entity Recognition** using **Spacy** on **OCR** data, to identify Job Title and find relation

Graduate Researcher Intern — **June 2022 - August 2022**

- Built **pipeline** to extract graph data and tested it on **Linux server** by creating environment using **PyTorch** and **CUDA**
- Utilized **Pillow** and **OpenCV** to calculate dominant pixel color of a legend patch and enhanced detection by identifying pixels within a similar **HSV value** range
- Achieved recognizing all lines and value points in line graph image and related it to labels in legend
- Identified symbols in legend using morphological process, detected similar symbols in line graph image by feature extraction and matching using **FAM-Net** model giving **75.6%** accuracy

Cyberace Infovision Private Limited — Mumbai, MH
Software Engineering Intern — **June 2019 - July 2019**

- Collaborated with team to program web application, designed to comprehend all data of an app on Google Play store to generate insights pertaining to market performance of a newly launched app
- Deployed a web application on a **Flask server**, seamlessly storing all newly collected data in a **MongoDB** database
- Employed **Python's NumPy, Pandas, Matplotlib, and Scikit-learn** modules for accurate and reliable data processing
- Utilized ML algorithms, including **time series forecasting** and **sentiment analysis**, to evaluate performance of apps

RELEVANT PROJECTS

Criminal Clothes Detection *(Python, React.js, Typescript, Flask, YoloV3, Mobile Net, OpenCV, Digital Ocean Cloud)*

- Engineered a **Flask-based web application** utilizing **REST API** to detect criminal attire from CCTV camera footage.
- Detected people from video frames using **YoloV3**, type of cloths using **Mobile-Net** and clothes color using **OpenCV**
- Utilized **MongoDB** to capture data from video frames and visually present real-time data through **dynamic charts** in **React.js** web pages, by making use of **Chart.js** and **Plotly.js** and hosting the project on **Digital Ocean Cloud.**

RAFT Web Application *(Python, Docker, Flask)*

- Developed a **distributed Flask web application** with 5 nodes, deployed on **Docker** for **scalability**
- Implemented leader election and safe log replication using **RAFT**, ensuring fault tolerance and reliability

Acuity Eye Test and Disease Detection Application *(Python, JavaScript, Flask, Tkinter, Keras, Beautiful Soup)*

- **Led a team** in developing a web application that assesses visual acuity using a Log MAR chart, incorporating **Google Speech to Text** for voice input within **Flask backend**
- Trained and integrated **Keras** eye disease detection module with web application with **79%** accuracy
- Displayed useful information regarding selected disease on web page by **Web Scrapping** using **Beautiful Soup**

KNN Classification models and AutoEncoding and Decoding *(Python, Keras)*

- Build four KNN classification models such as **VGG16, VGG19, XCEPTION** and **INCEPTIONV3** from **Keras**, trained and tested them on **CISFR-10 dataset** , and noted which model performs best by calculating **Silhouette Score**
- Build **AutoEncoding** and **Decoding** model and tested optimizers and loss function and Evaluated image quality

Song Playlist (*C++)*

- Developed an interactive Song Playlist Application enabling users to add, remove, shuffle, save liked songs, and efficiently search tracks by names and singers.