Varshith Batti

Vijayawada battivarshith2003@gmail.com 7702246996 linkedin.com/batti-varshith
https://github.com/battivarshith

PROFILE

Dedicated Computer Science & Engineering individual with strong technical acumen, seeking a full-time or internship role focused on solving high-complexity real-time problems using state-of-the-art technology stacks. Experienced in team-based projects, enhancing leadership skills, and adept at adapting architectures to specific use cases through continuous learning and technical stack enhancement.

Education

Woxsen University *2021 – 2025*
Computer Science Engineering
- GPA: 6.72/10.0

Graviity Junior College *2019 – 2021*
MPC
- Percentage: 70.4%

NSM *2007 – 2019*
Secondary School of Education
- Percentage: 70.2%

Experience

Research Assistant *Hyderabad*
CHSS *Feb 2024 – June 2024*
- Worked as a Research Intern at CHSS(Centre for Human Security Studies) on AI and developed a tracking system using Embedded C language.

Projects

Limited Training Approach To Model Latent Fingerprint Data For Time-Constrained Solutions github.com/latentfingerprint
- **Tech Stack Used:** One-Shot Learning, Few shot Learning, Siamese Network, Prototypical Network, Densenet121, VGG19, ResNet50.
- This project develops an efficient methodology for latent fingerprint recognition, addressing the challenge of high processing times in existing methods. By integrating one-shot and few-shot learning techniques within a lightweight architecture, the approach utilizes a prototypical network based on DenseNet121, achieving a test accuracy of 91.66% on the IIIT-D latent fingerprint dataset. The method demonstrates high performance with an F1 score of 93.32% and precision of 93.93%, achieving accurate and computationally efficient latent fingerprint recognition. This novel approach balances high accuracy with streamlined processing for effective latent fingerprint detection.

Lung Cancer Classification github.com/lungcancer
- **Tech Stack Used:** CNN & Its architectures (VGG19, DenseNet121, ResNet50, InceptionV3).
- This project, **Lung Cancer Classification Using Deep Learning**, involved classifying lung cancer images obtained from The Cancer Imaging Archive. The DICOM images were converted to JPG format for model training, which used various CNN architectures, including VGG19, DenseNet121, ResNet50, and InceptionV3. Among these, DenseNet121 achieved the highest performance, with an accuracy of 74.40%, an F1 score of 68.88%, precision of 72.29%, and recall of 73.91%, indicating its effectiveness for lung cancer classification in this study.

Abnormal Human Activity Detection github.com/abnormalactivity

- **Tech Stack Used:** Roboflow for Annotations, Object Detection Frameworks (YOLOv5, YOLOv8, YOLOv11)
- The project began with data collection and labeling, followed by model training and fine-tuning. Each YOLO model version was tested for performance metrics, including mean Average Precision (mAP), precision, and recall. YOLOv5 emerged as the top performer with a mAP score of 73.9%, precision of 70.4%, and recall of 67.4%. YOLOv11 also showed high recall at 68.5% with a mAP of 72.6%. These results demonstrate the effectiveness of the system in accurately detecting abnormal activities in real time, making it highly valuable for enhancing automated surveillance and security systems.

Smart Parking System github.com/smartparking

- **Tech Stack Used:** Python, Flask, OpenCV, Tesseract, Pandas, NumPy, HTML, CSV.
- Smart Parking Registration system developed by students at Woxsen University that uses Automatic Number Plate Recognition (ANPR) technology. The core workflow is: The system uses webcams at parking entry/exit points to capture vehicle images. These images are processed using OpenCV for image enhancement and edge detection, followed by OCR (Optical Character Recognition) to extract the license plate numbers. The extracted data is stored in CSV files tracking check-ins and check-outs. The system has a web interface built with Flask that includes admin login functionality and a dashboard showing real-time parking status. When a vehicle enters, the system captures its plate, records the entry time in 'check-in.csv', and when it exits, records it in 'check-out.csv'. The dashboard displays current occupancy and available slots (maximum 20 spaces). If the parking is full, it displays a waiting message; otherwise, it shows the number of available spots. The solution incorporates security measures like password protection for admin access, though the report notes this could be enhanced with proper password hashing in future iterations.

Technologies

Technical Skills: Python, C, Java, SQL, JavaScript, HTML,CSS, Machine Learning, Deep Learning

Publications

Limited Training Approach To Model Latent Fingerprint Data For Time-Constrained Solutions February 2025

10.1109/COMSNETS63942.2025.10885690

Language

- English
- Telugu