

Project Design Phase-II Technology Stack (Architecture & Stack)

Date	06 May 2023
Team ID	NM2023TMID18455
Project Name	CrimeVision: Advanced Crime Classification with Deep Learning

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

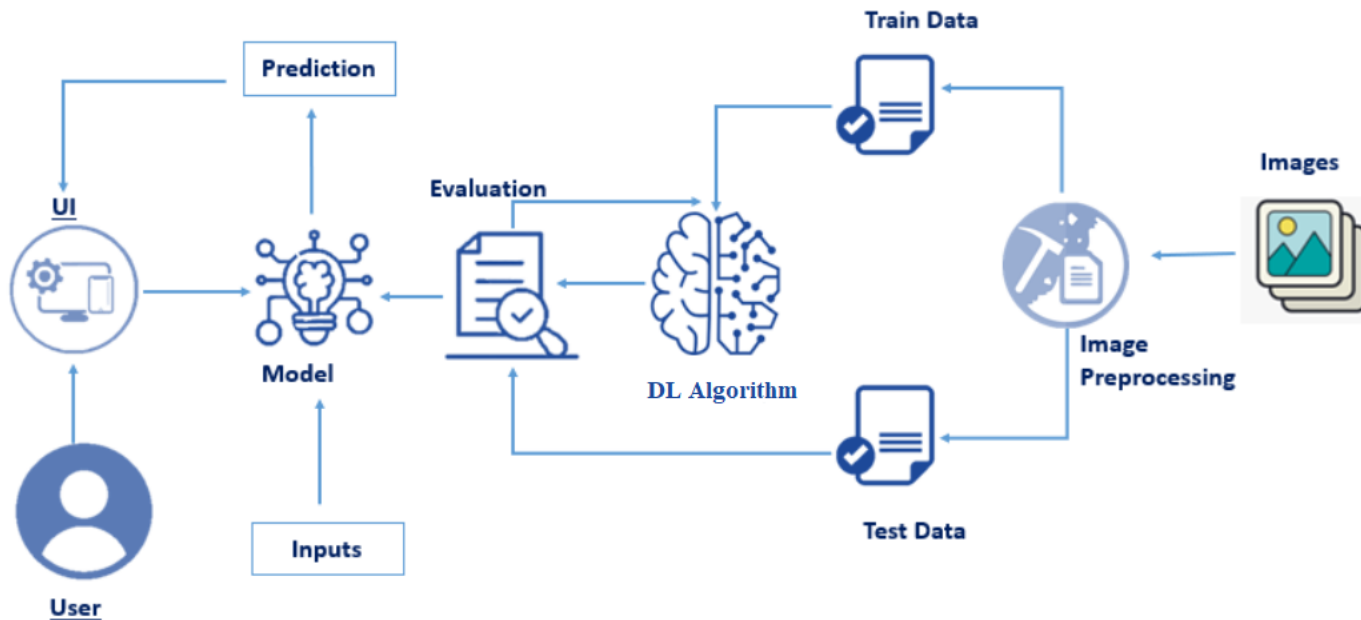


Figure- Components in Technical Architecture in Crime Classification using Deep Learning

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	UI (User Interface)	The visual and interactive elements that allow users to interact with the system.	HTML, CSS, JavaScript, UI frameworks (e.g., React, Angular)
2.	Model	The computational representation or structure that learns from data and makes predictions.	Deep learning frameworks (e.g., TensorFlow, PyTorch), programming languages (e.g., Python), neural network architectures (e.g., CNN, RNN)
3.	Deep Learning Algorithm	The specific algorithm or technique used in the deep learning model to learn patterns and make predictions.	Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), YOLO (You Only Look Once), etc.
4.	Evaluation	The process of assessing the performance and effectiveness of the model.	Metrics (e.g., accuracy, precision, recall, F1-score), programming languages (e.g., Python), data analysis libraries (e.g., NumPy, pandas)
5.	Image Pre-processing	The operations performed on the input images to enhance their quality and facilitate analysis.	Image processing libraries (e.g., OpenCV), programming languages (e.g., Python), image manipulation techniques (e.g., resizing, normalization)
6.	Train Data	The labelled data used to train the deep learning model.	Labelled image datasets, data collection and labelling tools
7.	Test Data	The labelled data used to evaluate the performance of the trained model.	Labelled image datasets, data collection and labelling tools

Table-2: Application Characteristics:

S.no	Characteristic	Description	Technology
1.	Open-Source Frameworks	Utilizing publicly available frameworks with source code accessible to the public, allowing customization and community collaboration.	TensorFlow, PyTorch, Keras, scikit-learn, OpenCV, Django, Flask, Node.js
2.	Security Implementations	Incorporating measures to protect data and ensure the confidentiality, integrity, and availability of the system.	Encryption algorithms (e.g., AES, RSA), secure communication protocols (e.g., SSL/TLS), authentication mechanisms, access control systems
3.	Scalable Architecture	Designing the system to handle increasing workload and user demand by adding resources and expanding functionality.	Cloud platforms (e.g., AWS, Azure, Google Cloud), containerization (e.g., Docker), load balancing, horizontal scaling
4.	Availability	Ensuring the system is accessible and operational, minimizing downtime and maximizing uptime for users.	High availability architectures, redundant infrastructure, fault-tolerant systems
5.	Performance	Optimizing the system to deliver fast and efficient responses, minimizing latency and maximizing throughput.	Caching mechanisms, performance monitoring tools, optimization algorithms, hardware acceleration (e.g., GPUs)