A Novel Approach for Disaster Victim Detection under Debris Environments Using Decision Tree Algorithms with Deep Learning Features

World often suffer from natural calamities like earthquake of building collapse which leads to loss of several lives and properties. Under natural calamities there should be a technology which can detect victims efficiently so on time medical aid will be provided to save lives. So in propose paper author saying Snake like robots can be equipped with Artificial Intelligence algorithm faster detection of victims.

In propose paper author employing fine tune Resnet50 transfer learning algorithm to extract victim learned features from the images, extracted features will be input to J48 based decision tree algorithm to select relevant features. Selected relevant features were trained with several Machine learning algorithms such as MLP, Random Forest, SVM and many other algorithms.

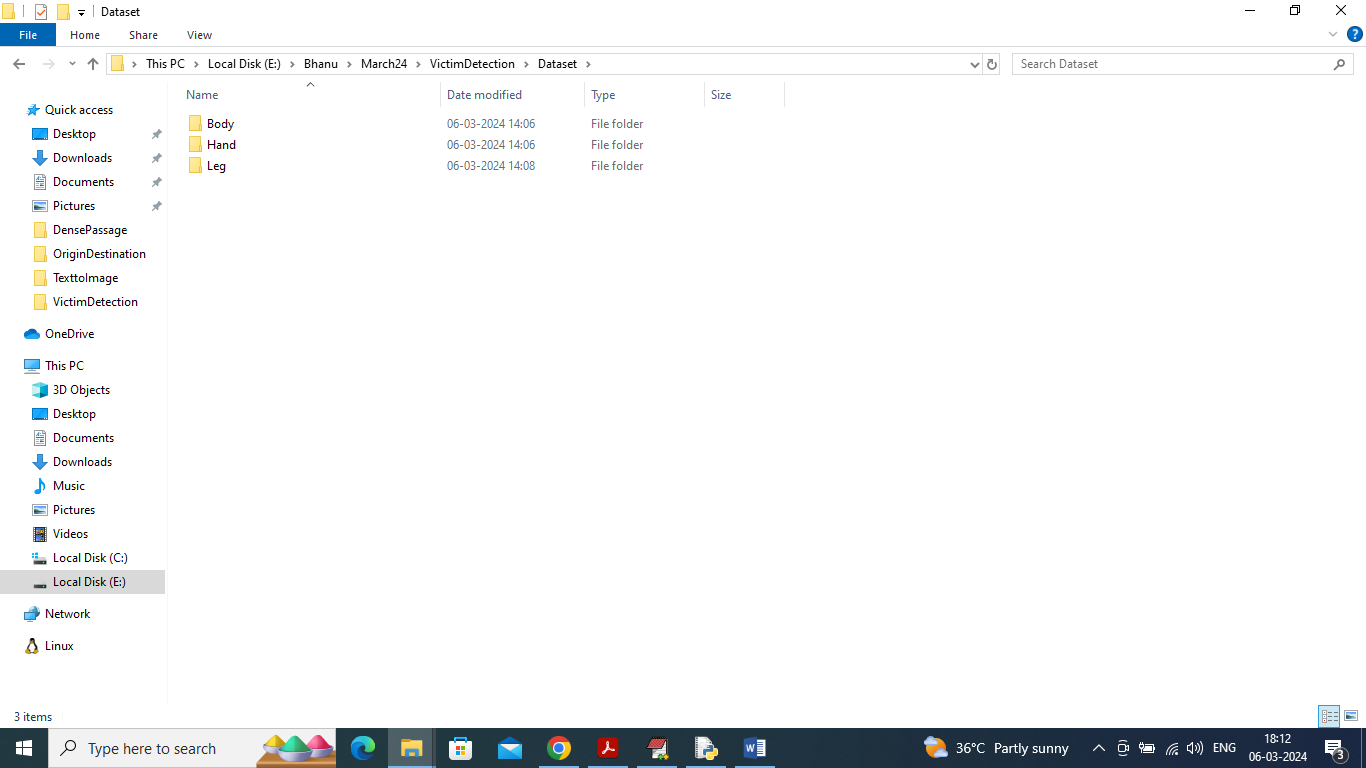
Each algorithm performance is evaluated in terms of accuracy, precision, recall and FSCORE. Among all algorithms Random Forest is giving best detection accuracy.

To train all algorithms author hash utilized Victim Detection Dataset which can be downloaded from below website

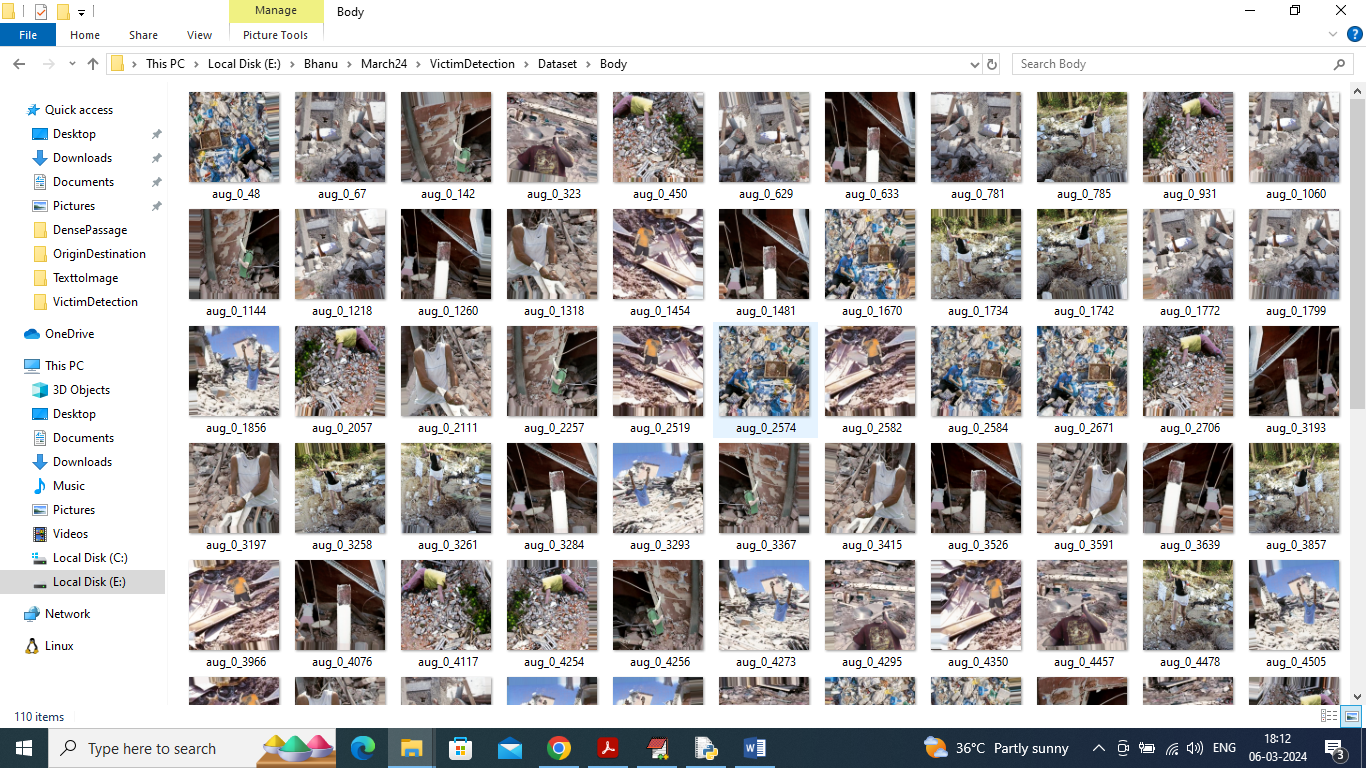
<https://github.com/noahzn/VictimDet/tree/main>

Above dataset contains images of trapped victim body parts such as Hand, Leg and Body. All the above mention algorithms will get trained with above dataset to detect victims.

In below screen showing some images from dataset



In above screen we have 3 different folders and just go inside any folder to view related images like below screen



So by using above dataset images will train Resnet50 to extract learned features and then employ j48 to select features and then train with ML algorithms for victim detection.

Extension Concept

In propose paper author has used all traditional ML algorithms so as extension we have utilized XGBOOST algorithm which is also a tree based algorithm with multiple ESTIMATORS to tuned dataset features which can help in better prediction accuracy.

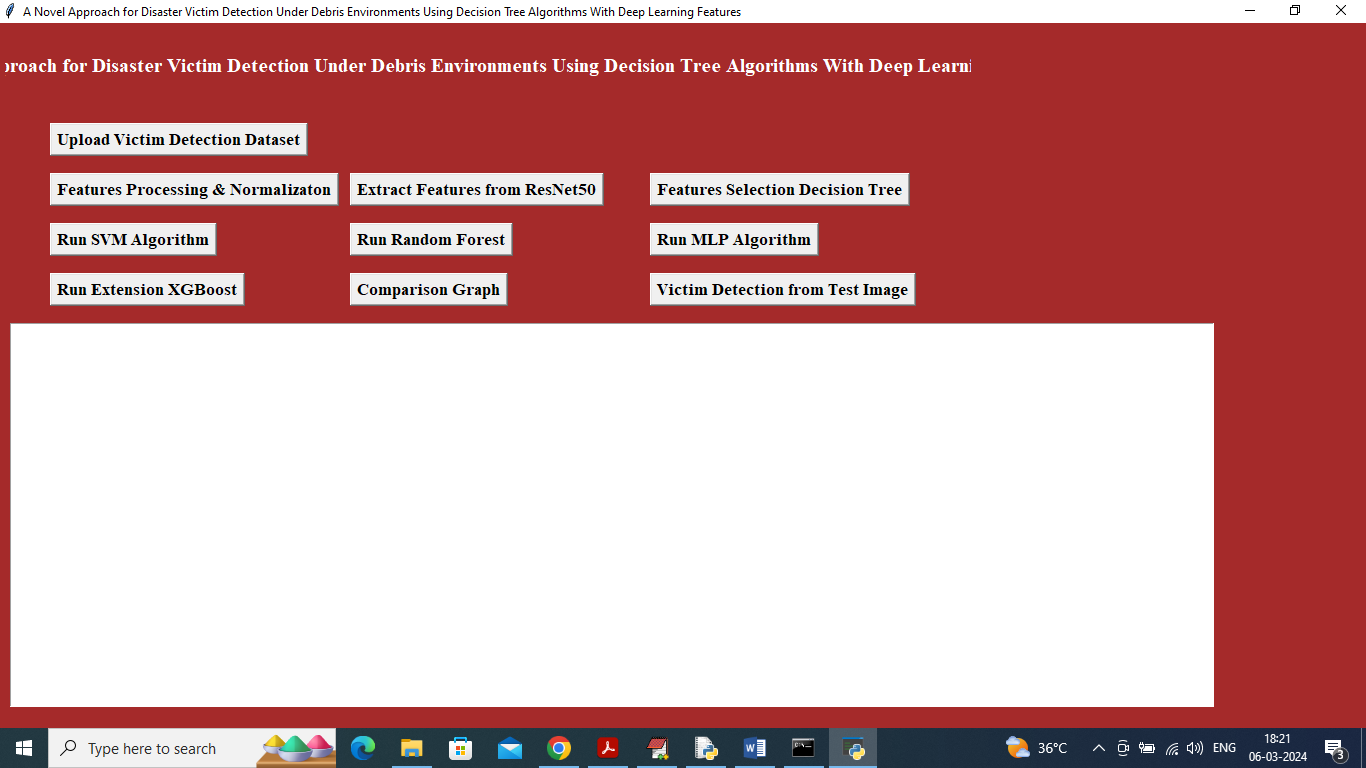
To implement this project we have designed following modules

1. Upload Victim Detection Dataset: using this module will upload dataset and then read and extract features from all images and then plot graph of different class labels found in dataset
2. Features Processing & Normalizaton: will process each images to normalize and shuffle pixels values and then split process images into train and test where application using 80% images for training and 20% for testing
3. Extract Features from ResNet50: processed images will be input to Resnet50 algorithm to extract learned features
4. Features Selection Decision Tree: extracted features will be input to J48 decision tree algorithm to select relevant features
5. Run SVM Algorithm: selected features will be split to train and test and then run with SVM algorithm to calculate victim detection accuracy
6. Run Random Forest: will train random forest algorithm and then calculate prediction accuracy
7. Run MLP Algorithm: will train MLP algorithm and then calculate prediction accuracy
8. Run Extension XGBOOST: will train extension XGBOOST algorithm and then calculate prediction accuracy
9. Comparison Graph: will plot comparison graph between all algorithms
10. Victim Detection from Test Image: will upload test image and then ML algorithm will predict detected victim part from the image

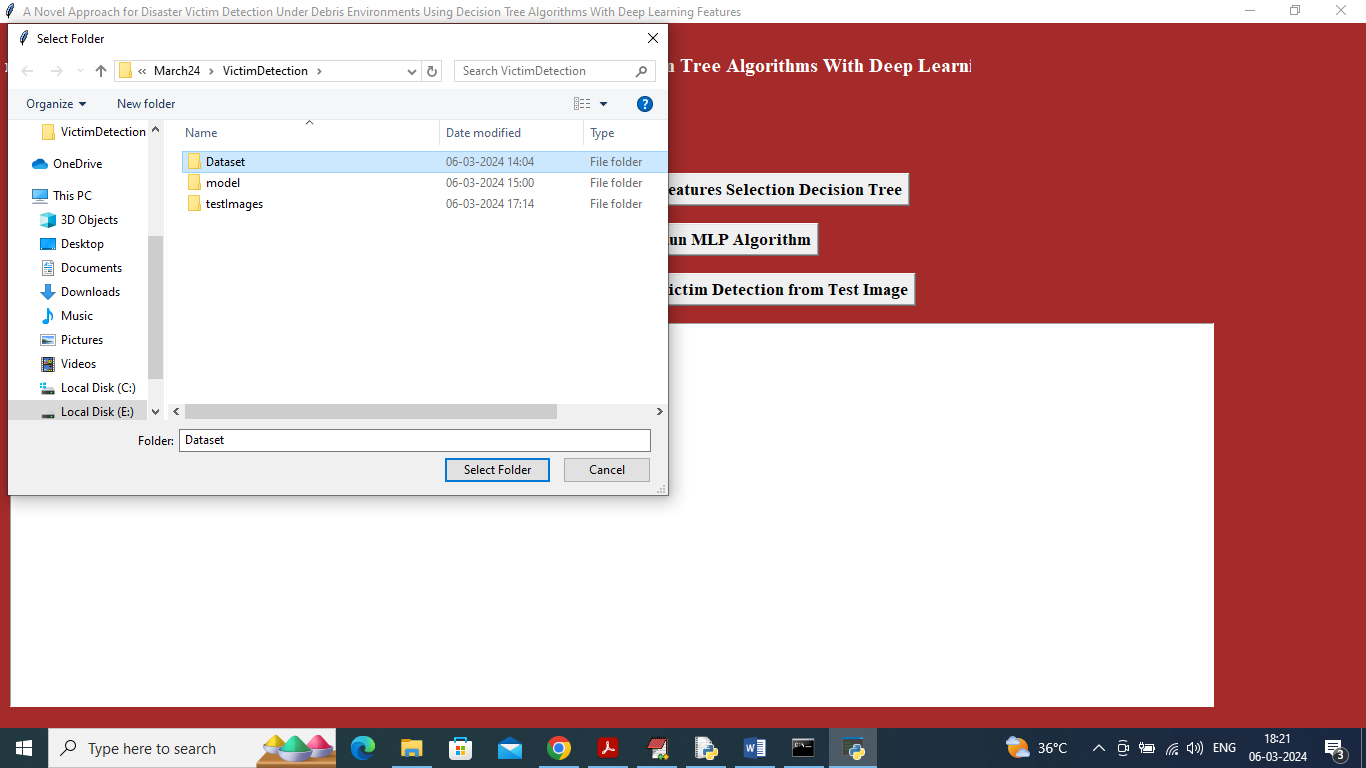
Note: this project has lots of module so training all modules in web will consume lots of time so we develop as TKINTER GUI

SCREEN SHOTS

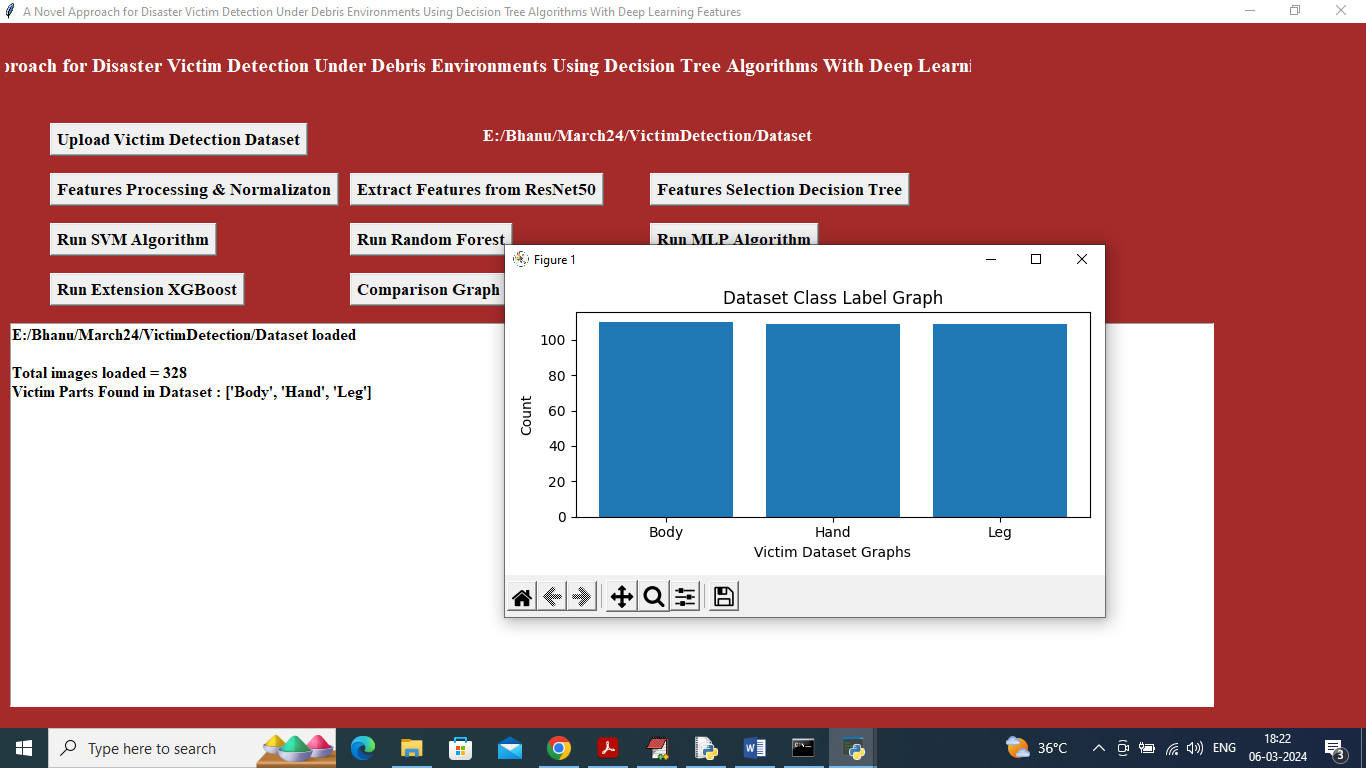
To run project double click on run.bat file to get below screen



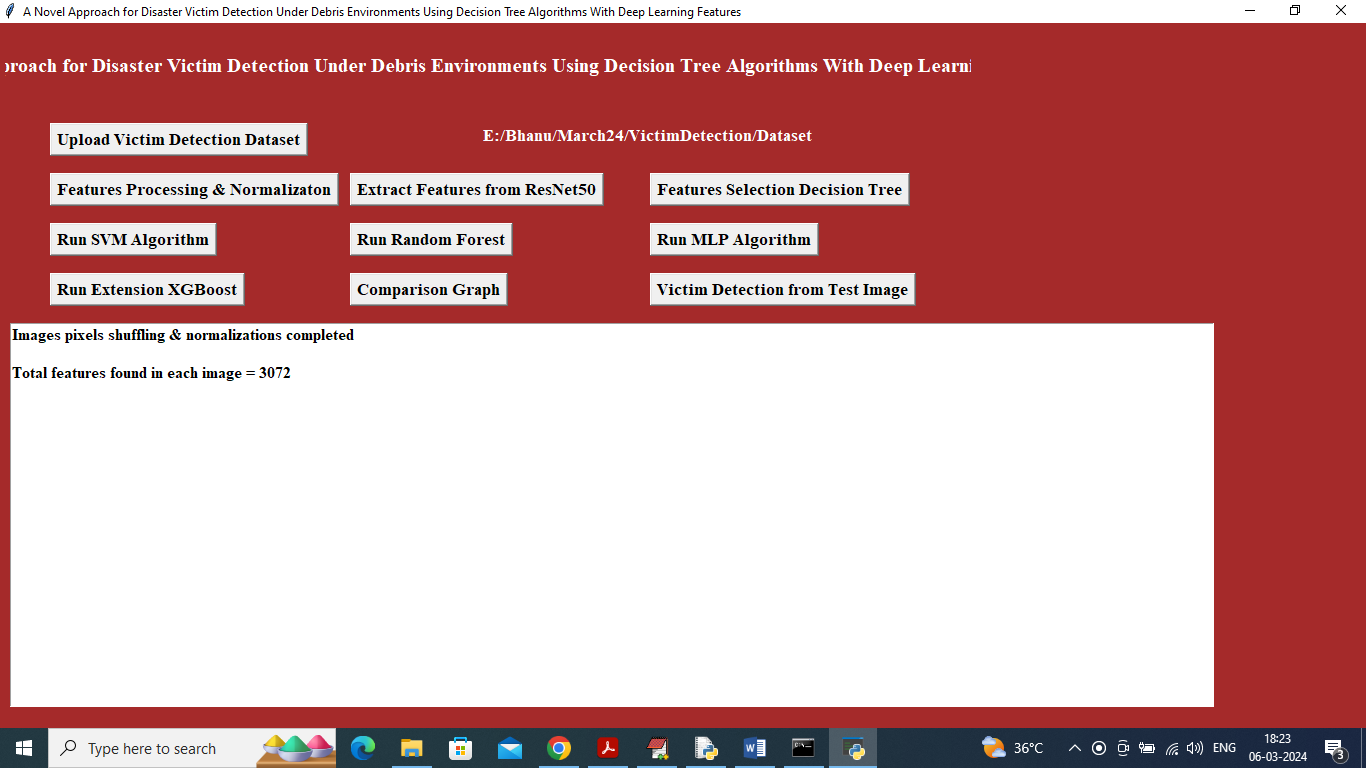
In above screen click on ‘Upload Victim Detection Dataset’ button to upload dataset and get below output



In above screen selecting and uploading Dataset and then click on ‘Select Folder’ button to load dataset and get below page



In above screen can see dataset contains 328 images of various part labels and in graph x-axis represents victim part names and y-axis represents count of images under that category and now close above graph and then click on ‘Features Processing & Normalization’ to process features and get below page



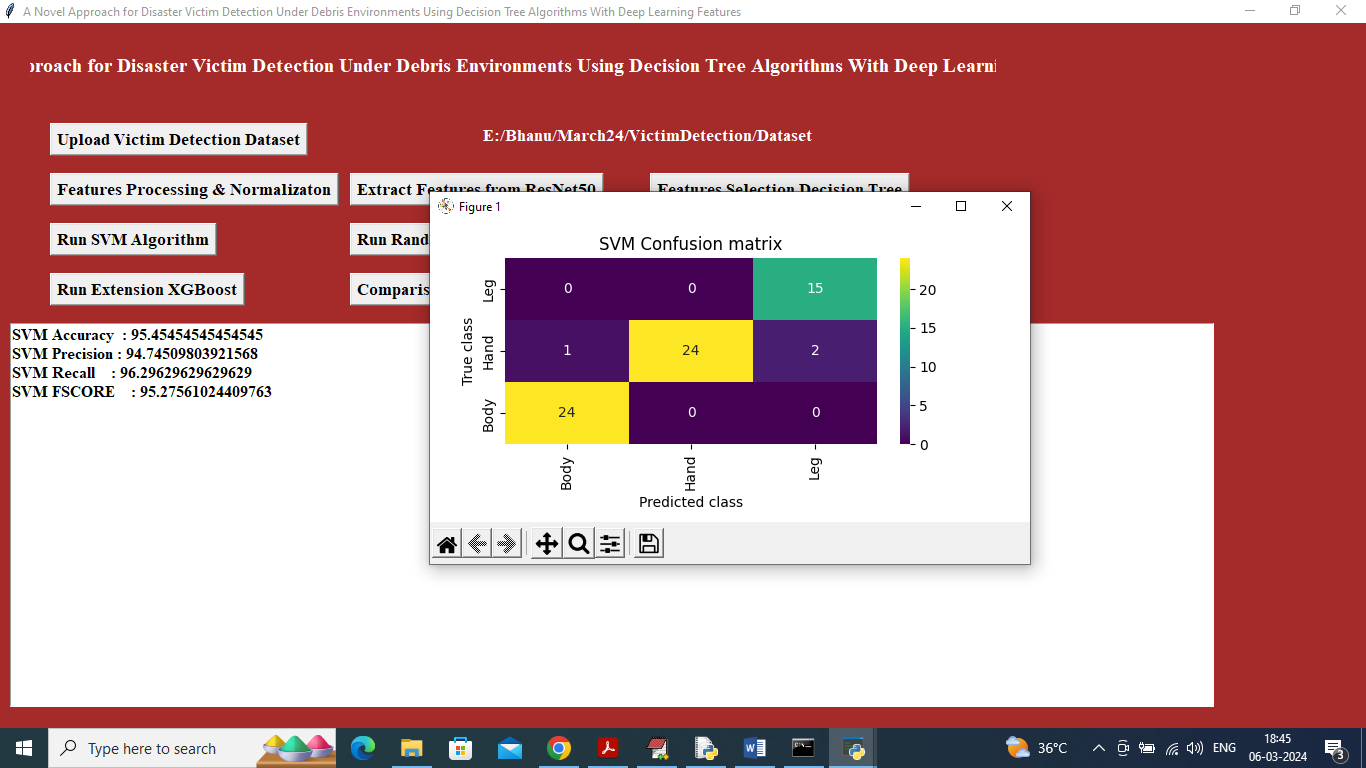
In above screen application read 3072 features from each image and then input this features to ResNet50 algorithm to extract features and get below page



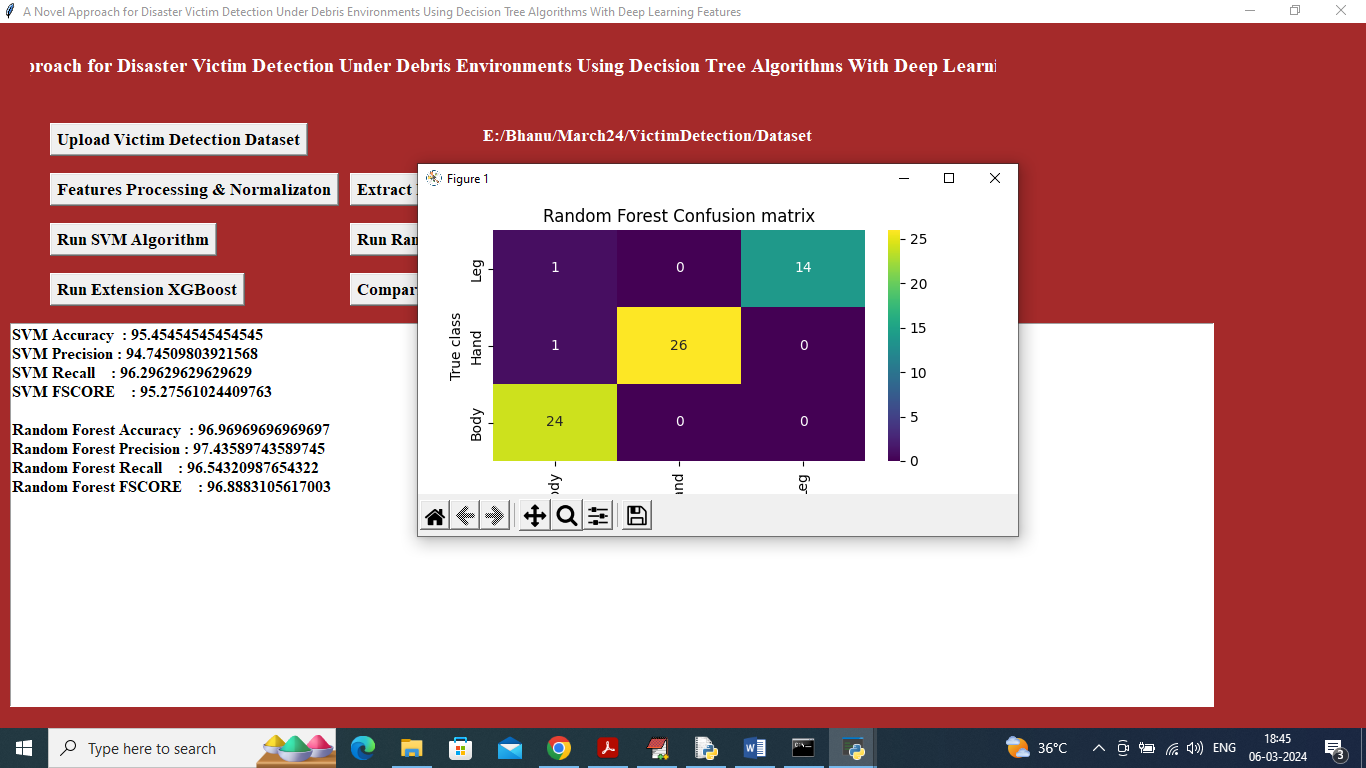
In above screen Resnet50 extracted relevant 256 features from given 3072 original features and now click on ‘Features Selection Decision Tree’ button to select relevant features out of 256



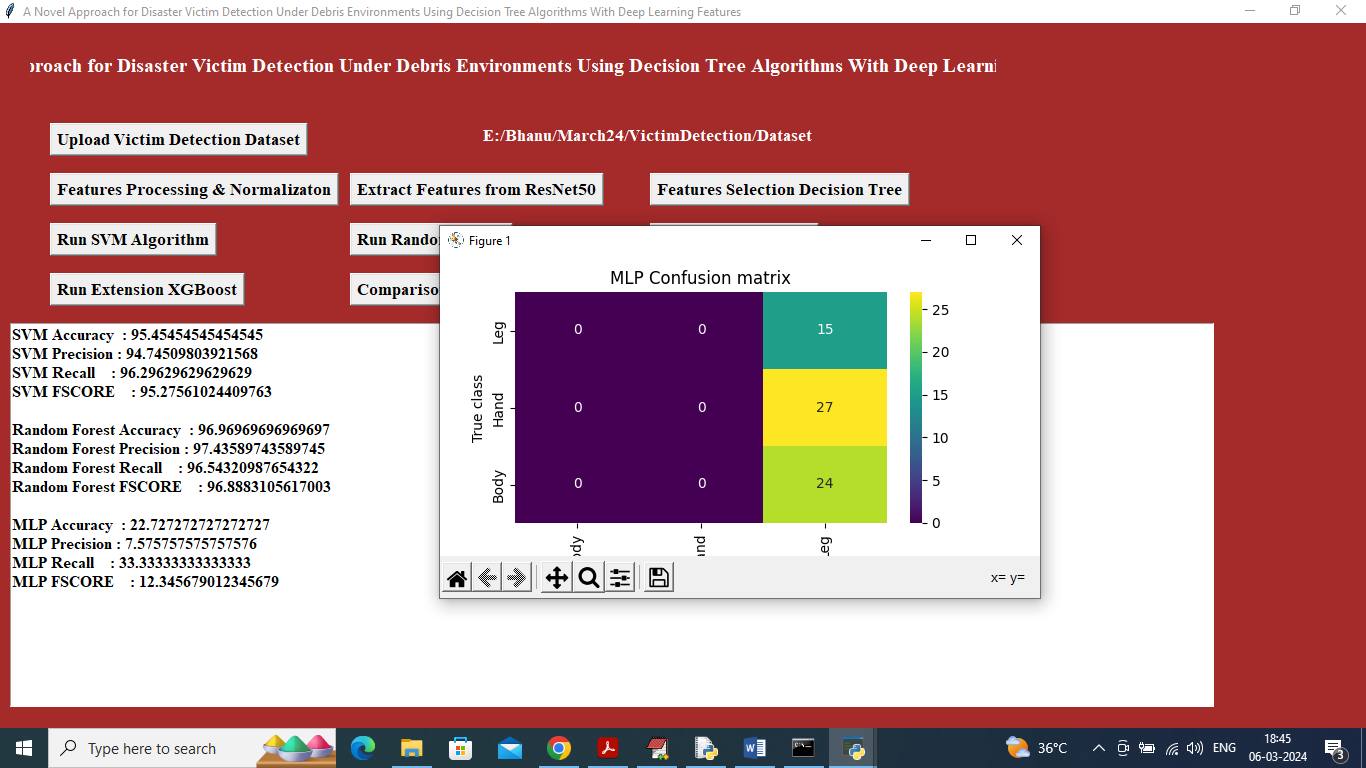
In above screen from 256 J48 Decision Tree selected 14 features and then can see Train and test size and now click on ‘Run SVM Algorithm’ button to get below page



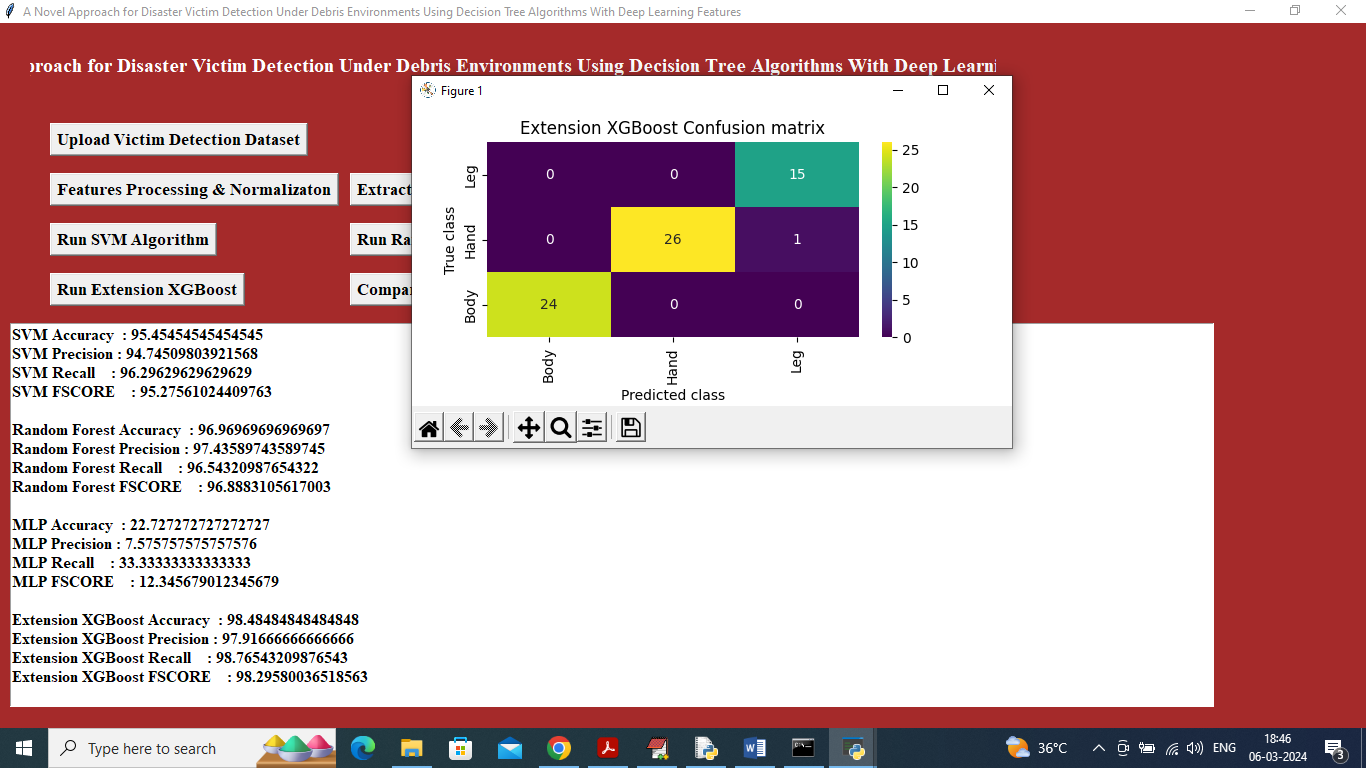
In above screen SVM training completed and it got 95% accuracy on test images and can see other metrics like precision, recall and FSCORE. In confusion matrix graph x-axis represents Predicted Labels and y-axis represents true labels and then all different colour boxes in diagnol represents correct prediction count and remaining all blue boxes contains incorrect prediction count. Now click on ‘Run Random Forest’ button to get below output



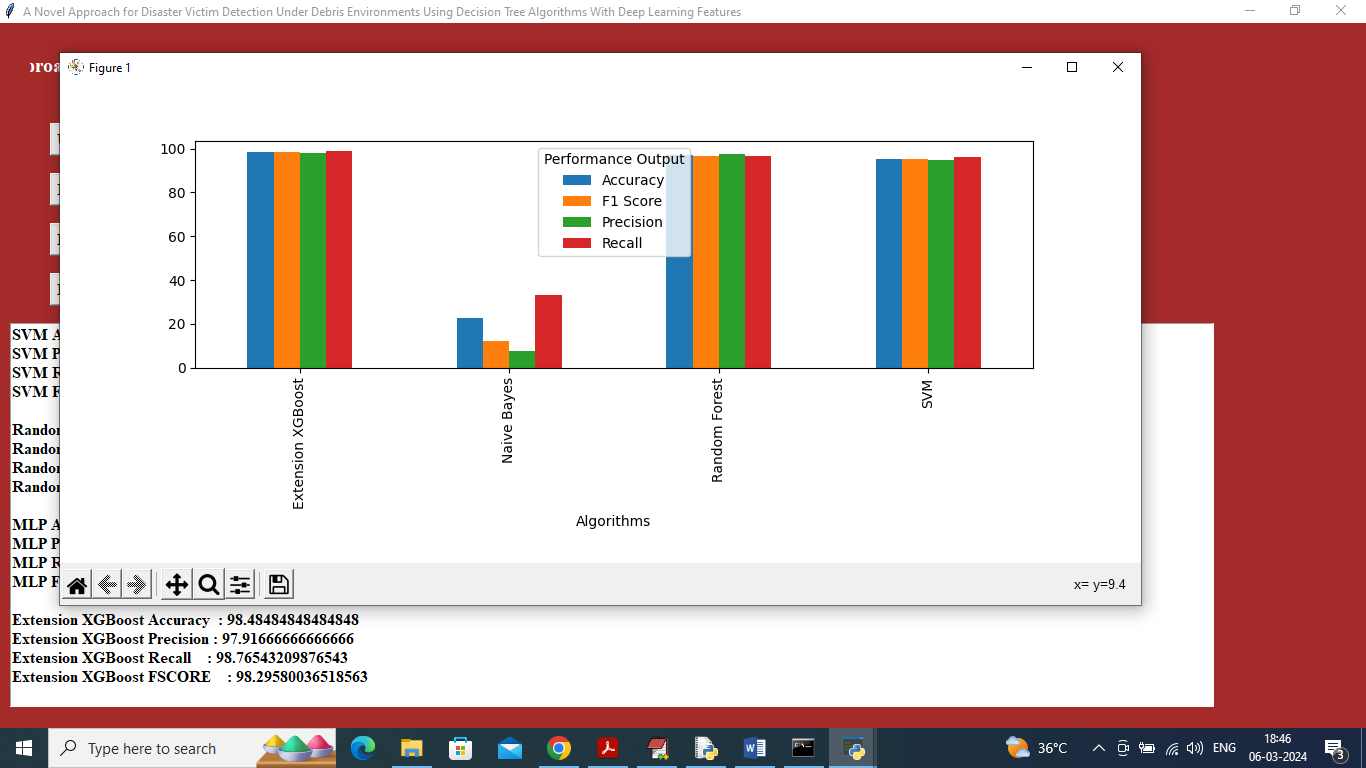
In above screen Random Forest also got 96% accuracy and now click on ‘Run MLP Algorithm’ button to get below output



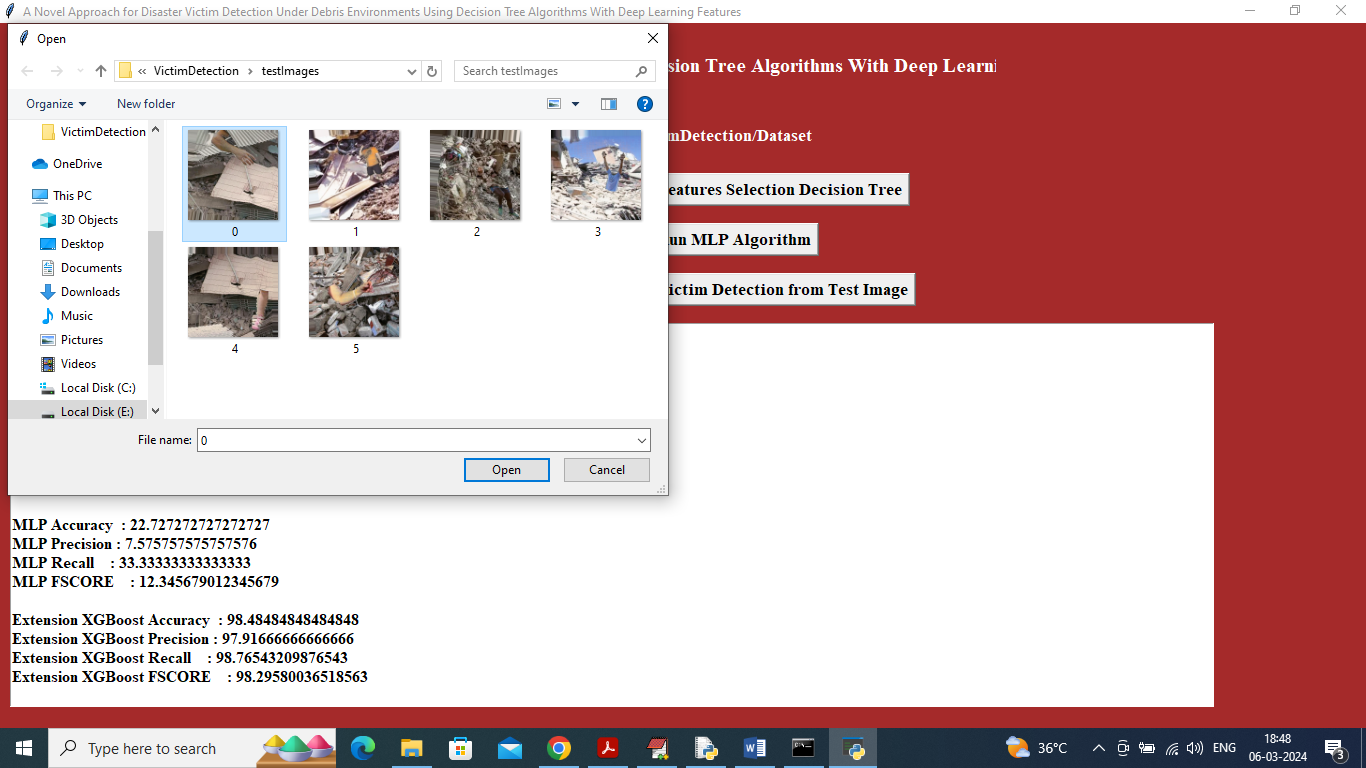
In above screen MLP got 22% accuracy and now click on ‘Run Extension Algorithm’ button to get below output



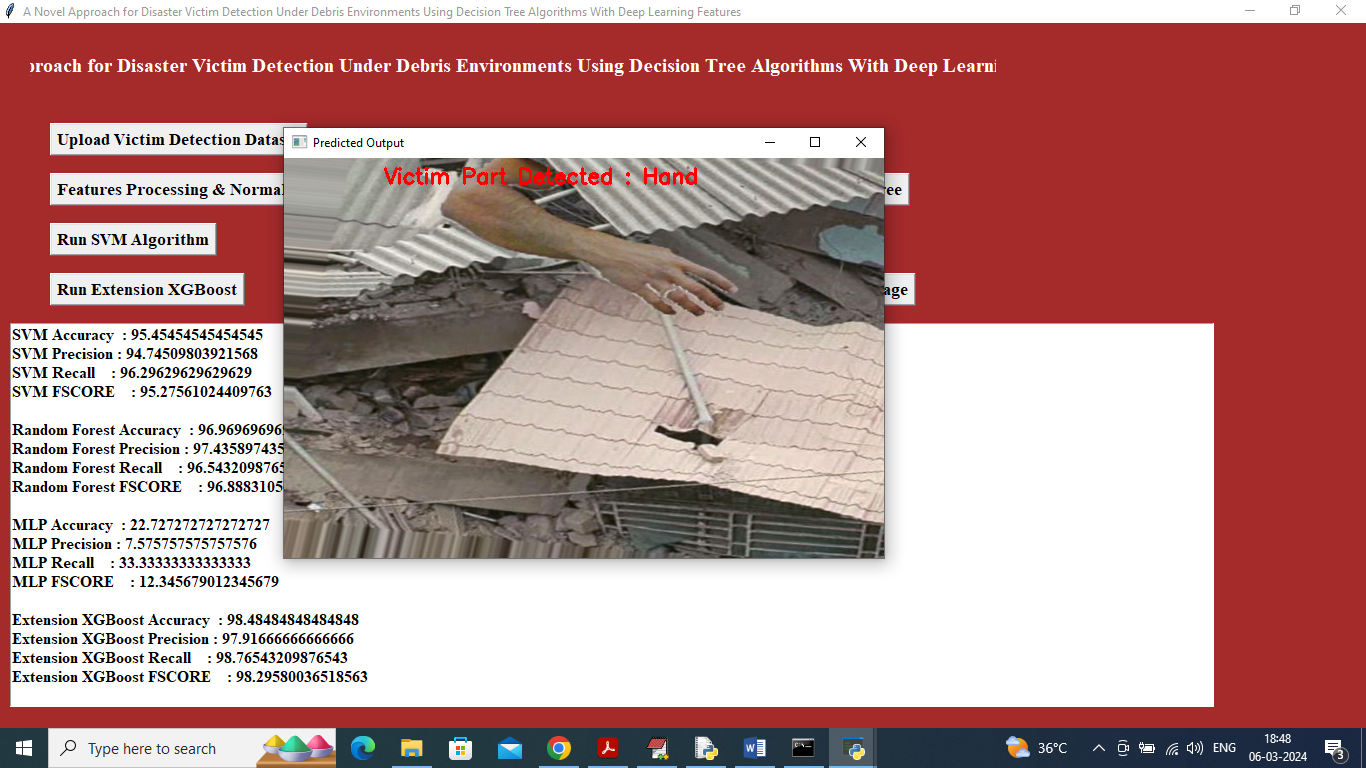
In above screen extension XGBOOST got 98% accuracy and can click on ‘Comparison Graph’ button to get below graph



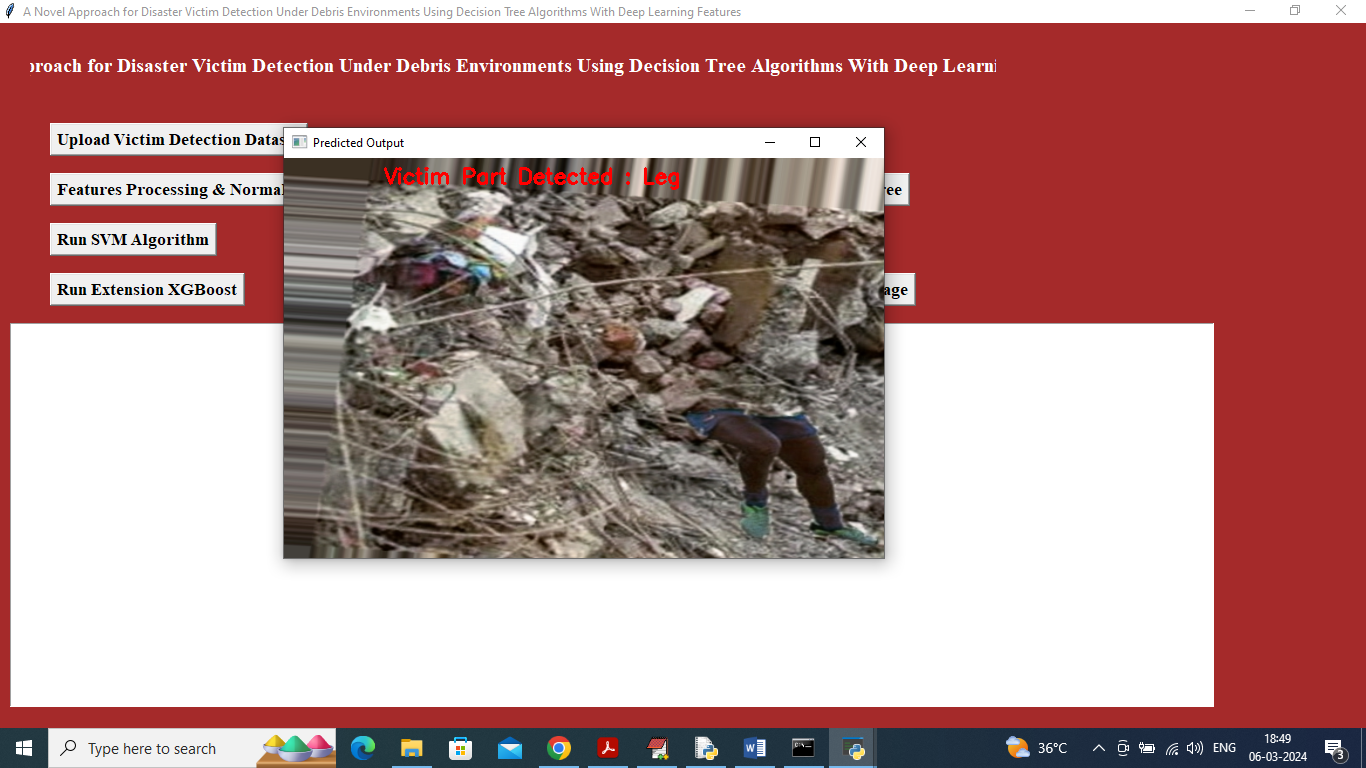
In above screen x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour and in all algorithms XGBOOST got high accuracy and now click on ‘Victim Detection from Test Image’ button to upload test image and get below output



In above screen selecting and uploading 0.jpg file and then click on ‘Open’ button to get below output



In above screen in red colour text can see victim part detected as HAND and similarly you can upload and test other images



In above screen leg is detected