10.48047/jocaaa.2025.34.05.43

Enhancing Emergency Response Through CNN-Based Injury Detection and Hospital Recommendation

Babu Enthoti[1*], G.Madhu[2], G.Durgesh[2], N.Srinivasulu[2], B.Saikiran[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering,
[1,2]Malla Reddy Engineering College and Management Sciences, Kistapur, Medchal, 501401, Telangana, India
*Corresponding author: babuenthoti@mrem.ac.in

Received: 02.03.2025 Revised: 15.04.2025 Accepted: 25.05.2025

ABSTRACT

Traumatic injuries constitute a major cause of emergency department admissions and can quickly become life-threatening without timely and accurate assessment. In pre-hospital and resource-limited environments, the absence of specialist expertise and advanced diagnostic tools often leads to inconsistent triage decisions and delayed treatment. To address this challenge, this research presents a CNN-driven, image-based injury assessment system designed to automatically identify injury type, estimate severity, and provide hospital recommendations for emergency response. The proposed desktop application integrates a custom Convolutional Neural Network (CCNN) for injury classification with a color segmentation–based severity detection module. Built using open-source Python libraries—including Tkinter for the graphical interface, OpenCV for image processing, Keras for deep learning, and scikit-learn for baseline model comparison—the system supports training and evaluation of multiple models such as Support Vector Machines, Deep Neural Networks, Random Forests, and the proposed CCNN. Experimental results demonstrate that the CCNN outperforms traditional models, achieving superior classification accuracy on unseen test data. During real-time inference, the system predicts the injury category, detects bleeding or bruising using HSV color thresholding and contour analysis, and classifies severity as minor or major. A recommendation engine then provides class-specific hospital guidance, enabling faster and more consistent emergency decision-making.

Keywords: Injury Detection, Emergency Response System, Convolutional Neural Network, Injury Severity Classification, Medical Image Analysis, Computer Vision, Hospital Recommendation System, Deep Learning in Healthcare, Automated Triage.

1. INTRODUCTION

Accidents and injuries, whether resulting from vehicular collisions, workplace hazards, or natural disasters, remain a leading cause of mortality and morbidity worldwide. The urgency to identify the type and severity of an injury promptly is critical in emergency response scenarios. The quicker that first responders, medical personnel, and hospital teams can identify the nature of an injury, the sooner they can provide appropriate treatment, potentially improving patient outcomes and reducing long-term complications.

Computer vision techniques have evolved significantly over the past few decades, transitioning from simple edge detection and classical feature extraction methods to sophisticated, deep learning-based approaches. Initially, tasks like wound identification or injury classification were tackled using rule-based systems and handcrafted features such as SIFT or SURF. However, these traditional techniques struggled with variability in lighting, image quality, and wound complexity.

10.48047/jocaaa.2025.34.05.43

The emergence of CNNs revolutionized visual recognition tasks by allowing models to learn hierarchical features directly from data, resulting in greater accuracy and robustness. Early medical imaging applications used CNNs for tasks like tumor detection in X-rays or segmentation of brain lesions in MRI scans. More recently, similar approaches have found their way into the emergency medicine domain, providing a framework for real-time injury detection and severity classification.

Traditionally, emergency medical technicians (EMTs), doctors, and first responders rely on their expertise and manual examination techniques to classify injuries and decide on the course of immediate treatment. While human expertise is invaluable, it can also be limited by factors such as time constraints, stress, and the availability of specialized personnel. With the rapid advancements in computer vision and machine learning, especially Convolutional Neural Networks (CNNs), automated systems capable of analyzing images and identifying the severity and type of injuries are now increasingly feasible.

2. LITERATURE SURVEY

This comprehensive survey covers the rapid integration of deep learning, particularly CNNs, into medical image analysis [1]. It examines various medical domains, detailing how CNN architectures have improved image classification, object detection, and segmentation. The authors discuss challenges such as limited training data and model interpretability, both highly relevant to injury detection scenarios. The paper's insights into common methodologies and pitfalls help guide the design of robust CNN-driven systems for injury type and severity detection.

Goyal et al. [2] focused on the application of deep learning, including CNNs, to wound image analysis. It covers wound segmentation, classification, and severity assessment techniques, highlighting how these methods improve clinical decision-making. The authors analyze various datasets, preprocessing steps, and evaluation metrics. This overview provides valuable insights into adapting CNN-driven approaches for classifying injuries and estimating severity in emergency settings.

Liu et al. demonstrated the use of CNNs to differentiate burn wound depths from photographic images [3]. By learning intricate visual cues, the model achieves high accuracy in classifying burn severity levels, a critical step towards timely intervention. The approach also addresses challenges in limited, high-quality medical datasets. Their work illustrates the feasibility of using CNN architectures to detect and grade injuries—methods that can be adapted for broader injury severity detection tasks.

Chen et al. applied a deep CNN-based classifier to accurately identify wound categories from images [4]. The authors incorporate data augmentation to overcome limited samples and achieve robust classification results. They also discuss the importance of proper preprocessing and the selection of hyperparameters. Their methodology supports the concept of integrating severity detection with class-based wound (injury) identification to guide clinical decisions.

While not focused on injury image analysis, Nguyen et al. [5] addressed the hospital recommendation problem using deep learning. It employs user preferences and healthcare data to suggest suitable medical facilities. Incorporating such approaches can complement an injury detection system by linking identified injury types and severity levels to the most appropriate hospitals. This work demonstrates how deep learning-based recommendation engines can enhance patient outcomes by guiding them towards optimal healthcare resources.

Yin et al. [6] employed deep CNN models to detect and classify diabetic foot ulcers, correlating the level of tissue damage to severity grades. The authors leverage image preprocessing and augmentation to enhance model robustness. Their approach demonstrates improved accuracy in identifying wound

10.48047/jocaaa.2025.34.05.43

severity, offering insights for adapting similar techniques to other injury types. It also emphasizes the importance of data diversity and model generalization.

Miola et al. addressed the automatic classification and severity grading of pressure ulcers through CNN-based analysis [7]. The researchers incorporate multiple wound features and use deep models to distinguish between ulcer stages. Their methodology shows the potential for early detection, allowing for timely and appropriate medical interventions. Results highlight the feasibility of adapting CNN frameworks for various injury scenarios.

In [8], Kok et al. analyzed existing AI technologies in wound care, including CNN-driven solutions for classification and severity assessment. The authors discuss how emerging algorithms streamline wound evaluation, reduce clinician workload, and support consistent measurements. They highlight current gaps, urging more robust datasets and integration with clinical workflows. The insights help inform the design of injury detection pipelines with integrated severity grading.

An early application of CNNs to wound image analysis, authors in [9] focused on binary tissue classification to distinguish healthy from unhealthy tissue. Although not directly addressing severity, the method's ability to characterize wound regions lays foundational work for subsequent severity-related tasks. The authors demonstrate the potential for CNNs to handle image noise and variability. Lessons from this study can guide preprocessing and feature extraction in injury severity detection.

Zhang et al. introduced a multi-task CNN framework capable of jointly learning related tasks, such as lesion detection and severity grading [10]. Though focused on breast cancer diagnosis, the methodology demonstrates how shared representations can enhance performance in multiple domains. By dynamically fusing different feature maps, the approach can be adapted for injury classification and severity estimation. It underscores the versatility of CNN architectures in complex medical scenarios.

Focusing on bone fractures, this research leverages CNNs to detect injuries on radiographs, improving diagnostic accuracy [11]. While not directly categorizing severity, their system distinguishes subtle fractures that can influence clinical decisions. Early and accurate detection informs the need for urgent intervention, indirectly guiding severity handling. The findings can inspire similar frameworks for surface injuries or trauma imaging in emergency response contexts.

In [12], Guo et al. focused on segmentation in multi-modal medical images using deep learning. Proper segmentation is crucial for accurately measuring wound areas or affected regions, which can then be used to gauge severity. Although not injury-specific, the methods developed here can be applied to delineate injury boundaries. Precise segmentation supports severity assessment by quantifying damage extent, crucial for guiding treatment recommendations.

The study in [13] showcases a CNN-based system for identifying traumatic injuries in maxillofacial CT scans. By classifying fractures and other trauma signs, the method assists clinicians in making quick, informed decisions. The localization and classification of injuries pave the way for integrating severity estimation techniques. Incorporating these insights into a unified framework can improve emergency response strategies and hospital referrals.

In [14], although focused on skin lesion classification rather than trauma, this dataset and its associated analyses underline the importance of large, diverse training sets. With CNN models trained on HAM10000, researchers achieve robust classification performance. Similar approaches can be adapted to injury images, improving the generalization of severity assessments. The study emphasizes the need for well-curated, multi-source image datasets.

10.48047/jocaaa.2025.34.05.43

In [15], authors applied CNNs to head CT scans to classify and triage neurological injuries by severity. By extracting discriminative features, the model assists in prioritizing cases needing urgent care. The approach exemplifies integrating automated severity detection into the emergency workflow. Lessons learned from their pipeline can be leveraged for external injuries and integrated with hospital recommendation systems in emergency response solutions.

Research Gap

The surveyed literature underscores the growing impact of CNNs in medical image classification, including injuries and wound severity. Early works focused on general medical image analysis, followed by specialized studies on wound classification and burn severity identification. These innovations pave the way for integrating severity assessment with automated classification models. Moreover, hospital recommendation systems powered by deep learning complement these detection frameworks, enabling a comprehensive solution that not only identifies and grades injuries but also directs patients and emergency responders to the most suitable healthcare facilities.

3. PROPOSED METHODOLOGY

The proposed methodology is designed to create an intelligent system that can automatically classify the type of injury from an image, assess its severity, and provide relevant hospital recommendations as demonstrated in Fig. 4.1. By aligning computer vision techniques with medical knowledge, the system provides a real-time, data-driven solution to assist emergency responders and medical personnel. This holistic approach ensures rapid, accurate, and actionable insights, potentially improving patient outcomes and optimizing emergency medical responses.

Step 1. Data Acquisition and Preprocessing

The first step in implementing the proposed solution involves gathering a comprehensive dataset of injury images. These images, representative of common injury types (such as head, hand, and leg injuries), are organized into separate directories. Each directory corresponds to one class label, enabling easy mapping from image to injury type.

Once the dataset is available, images undergo preprocessing to ensure uniformity and to facilitate efficient training:

- **Resizing:** All images are resized to a fixed dimension (e.g., 64x64 pixels), standardizing the input size and enabling batch processing.
- **Normalization:** The pixel values are normalized to a range of [0,1] by dividing by 255. This helps stabilize the training process and speeds up convergence.
- **Label Encoding:** Class labels (e.g., "Head," "Hand," "Leg") are extracted from directory names and encoded into numerical values or one-hot vectors for model compatibility.

This preprocessing stage ensures that the input data is consistent, structured, and ready for model training.

Step 2. CCNN Architecture

At the heart of the proposed methodology lies a CNN designed for automatic and robust feature extraction. Unlike traditional machine learning approaches that rely on handcrafted features, CNNs learn hierarchical representations of images directly from the data, capturing intricate details such as texture, shape, and edges.

10.48047/jocaaa.2025.34.05.43

The CCNN architecture typically consists of:

- **Convolutional Layers:** These layers apply filters (kernels) to the input image to detect local patterns. Early layers might capture edges or simple textures, while deeper layers learn more abstract features relevant to injury types.
- **Activation Functions:** Rectified Linear Units (ReLU) are commonly used to introduce non-linearity, allowing the network to model complex relationships between pixels and features.
- **Pooling Layers:** Max pooling layers reduce the spatial dimension of intermediate feature maps. This helps retain the most significant features while reducing computation and mitigating overfitting.
- **Fully Connected Layers (Dense Layers):** After flattening the pooled feature maps into a single vector, one or more dense layers combine these extracted features to classify the injury type. The final output layer uses a softmax activation function to produce probability distributions across the different injury classes.
- **Regularization Techniques:** Techniques such as dropout may be employed to prevent overfitting, ensuring the model generalizes well to unseen images.

The CCNN is trained using a labeled dataset of injuries. During training, the network adjusts its internal parameters (weights and biases) to minimize the discrepancy between predicted labels and ground truth labels, typically via the Adam optimizer and the categorical cross-entropy loss function.

Step 3. Injury Severity Detection

While the CCNN focuses on categorizing the injury type, the methodology also assesses injury severity. The assumption is that certain visual cues, such as the presence and extent of blood, correlate with severity.

To estimate severity:

- **Color Space Conversion:** The system converts the input image from BGR to HSV color space, which simplifies the task of detecting red hues commonly associated with blood.
- **Thresholding and Masking:** Using predefined lower and upper HSV bounds, the algorithm creates a binary mask isolating red regions. These red areas are then analyzed to determine their size and spread.
- **Severity Classification:** If the detected red region exceeds a certain threshold dimension (e.g., a bounding box width of 100 pixels), the injury is classified as "Major Severity." Otherwise, it is labeled as "Minor Severity."

This severity detection process complements the CCNN-based classification, providing a holistic assessment that goes beyond simple injury type recognition.

Step 4. Hospital Recommendations and Decision Support

A critical component of the proposed methodology is the integration of actionable recommendations. Once the injury type and severity are determined, the system references a repository of text files containing guidance tailored to each injury class.

- **Recommendation Retrieval:** For each predicted injury type, the system loads a corresponding recommendation file that provides instructions for immediate care, potential first-aid steps, and a list of suitable healthcare facilities. These recommendations bridge the gap between initial on-site assessment and specialized treatment.

10.48047/jocaaa.2025.34.05.43

- **Contextual Guidance:** The recommendations can include specific hospital departments experienced in handling the detected injury, specialized clinics equipped for major injuries, or guidelines on stabilizing the patient while awaiting professional medical assistance.

In this way, the system moves from pure classification to an integrated support tool, guiding first responders, EMTs, or even laypersons in making informed, timely decisions.

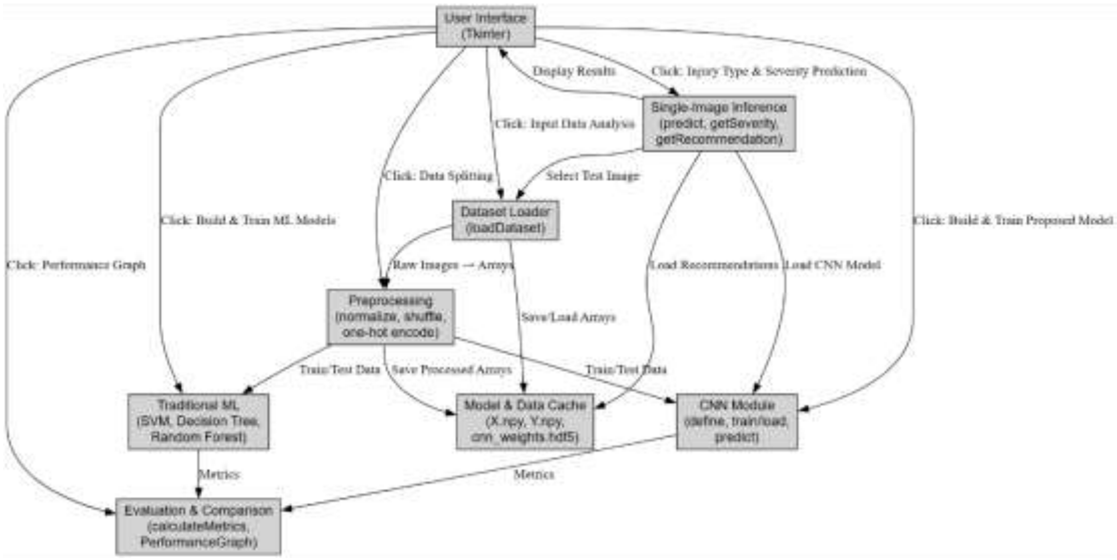


Fig. 1: Block diagram of proposed injury type and severity classification with hospital recommendation system.

Step 5. Model Evaluation and Performance Comparison

To validate the proposed methodology, the system is evaluated using a hold-out test set of images not seen during training. The following metrics are used:

- **Accuracy:** Proportion of correctly classified injuries.
- **Precision, Recall, and F1-Score:** Provide insights into the model's effectiveness in handling class imbalance and its ability to identify injuries correctly.
- **Confusion Matrix:** Offers a detailed breakdown of misclassifications, revealing potential areas for further refinement.

To contextualize the performance of the CCNN, it is compared against traditional machine learning approaches (MNB, DNNs, Random Forest). This comparison highlights the advantages of deep learning in capturing complex patterns, ultimately demonstrating superior accuracy and robustness.

Step 6. User Interface and Practical Implementation

The methodology is integrated into a graphical user interface (GUI) for seamless interaction:

- **Data Loading and Visualization:** Users can load datasets, visualize class distributions, and review preprocessing steps directly through the GUI.
- **Model Training and Testing:** Buttons allow for on-demand training of the CNN or traditional models, initiating predictions, and displaying evaluation metrics and confusion matrices.

10.48047/jocaaa.2025.34.05.43

- **Real-time Prediction:** Users can input a test image, and the system instantly classifies the injury type, assesses severity, and displays related recommendations. The GUI can showcase the processed image with severity-marked regions, enhancing interpretability.

4. RESULTS AND DISCUSSION

Figure 2 illustrates the main window of the Tkinter-based desktop application. At the top is a title banner displaying the system's name. On the right are vertically stacked buttons corresponding to each major function—data loading, visualization, data splitting, model training (ML, ensemble, CNN), performance graphing, single-image prediction, and exit. The left side features a scrollable text area that logs status messages, metrics, and recommendation text. This layout provides a clear, step-by-step workflow: users invoke each processing stage via the buttons and immediately see textual feedback in the log pane.

Fig. 2: Desktop GUI application of proposed Injury type and severity detection with hospital recommendation for emergency response.

10.48047/jocaaa.2025.34.05.43

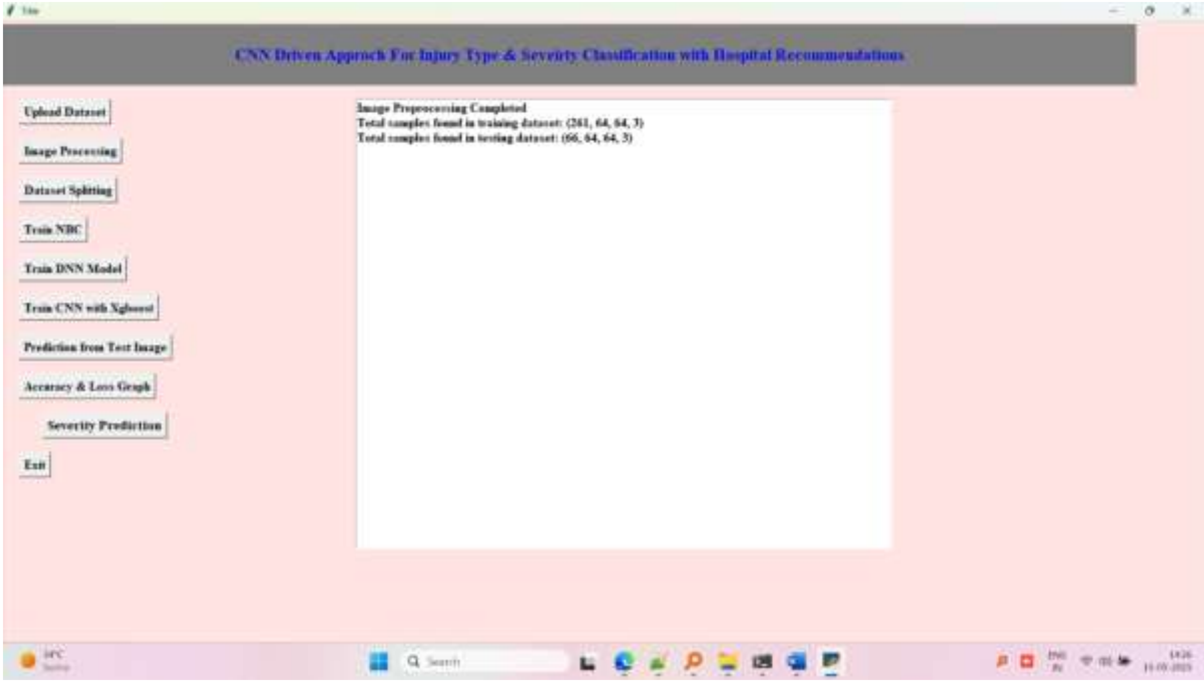


Fig. 3: GUI application showing labels, total images count after input data analysis.

After clicking "Input Data Analysis," the application traverses the Dataset/ folder, discovers class subdirectories (e.g., "Hand," "Head," "Leg"), and loads or caches the image arrays. Fig. 3 shows the text area populated with a message such as:

"Labels found in dataset are ['Hand', 'Head', 'Leg']

Accident Dataset Loading is Completed

Total Images Found in Dataset = 1,200"

It confirms successful label extraction and dataset ingestion, giving the user immediate insight into class names and dataset size before proceeding to subsequent steps.

Fig. 4 visualizes the number of samples per class. On the x-axis are the class labels ("Hand," "Head," "Leg"), and on the y-axis is the count of images in each category. By plotting these counts, the user can quickly assess dataset balance: ideally, all bars are roughly equal, indicating no class is under- or over-represented. If the chart reveals significant imbalance, the user may choose to augment or resample classes before training.

10.48047/jocaaa.2025.34.05.43

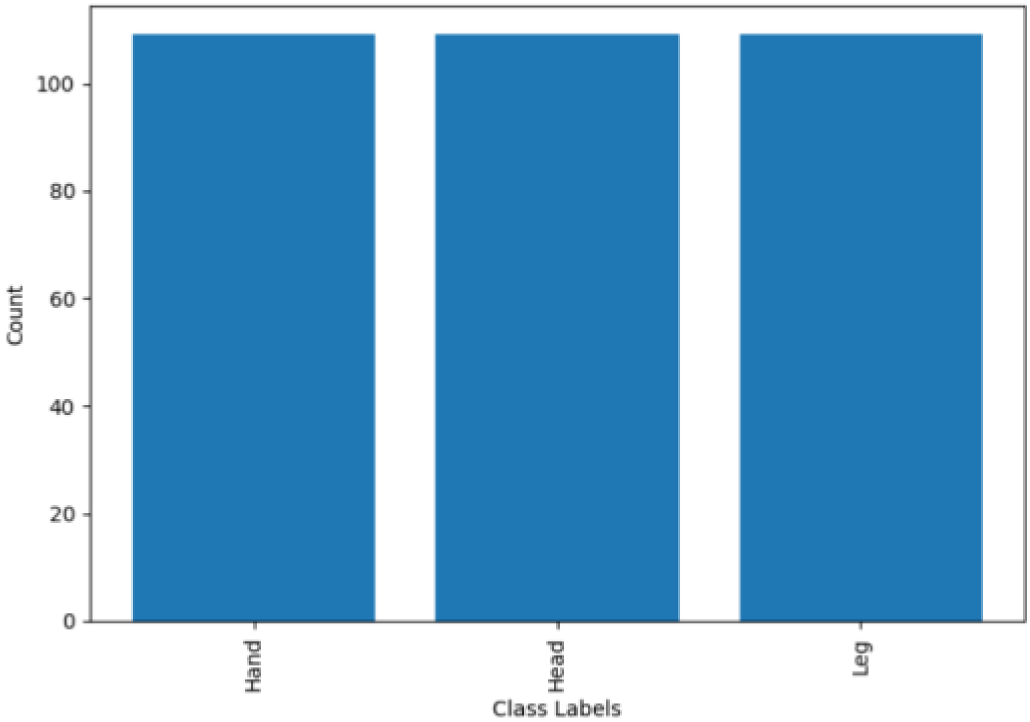


Fig. 4: Class distribution versus image count.

From Figure 5, The Naive Bayes Classifier (NBC) exhibits a moderate performance, achieving an overall accuracy of 60.61%. The precision (60.66%) and recall (60.52%) are fairly balanced, indicating that the classifier has a consistent ability to correctly identify and retrieve relevant instances. The F1-score, which harmonizes precision and recall, is 60.42%, confirming the model's average overall balance. In terms of class-wise performance, for the class ‘Hand’, the precision is 0.60, recall is 0.62, and F1-score is 0.61, suggesting a modest performance. The ‘Head’ class shows a slight improvement with a precision of 0.61 and recall of 0.67, resulting in an F1-score of 0.64. The ‘Leg’ class lags slightly behind with a precision of 0.61 but a lower recall of 0.52, yielding a weaker F1-score of 0.56.

From a medical or safety-critical context perspective, the sensitivity (71.43%) and specificity (82.35%) highlight the classifier's decent ability to detect true positives and reject false positives, respectively. The macro average F1-score (0.60) and weighted average F1-score (0.60) imply uniform performance across classes despite class imbalance. Overall, while the NBC is relatively stable across classes, its performance lacks the robustness needed for high-stakes applications.

10.48047/jocaaa.2025.34.05.43

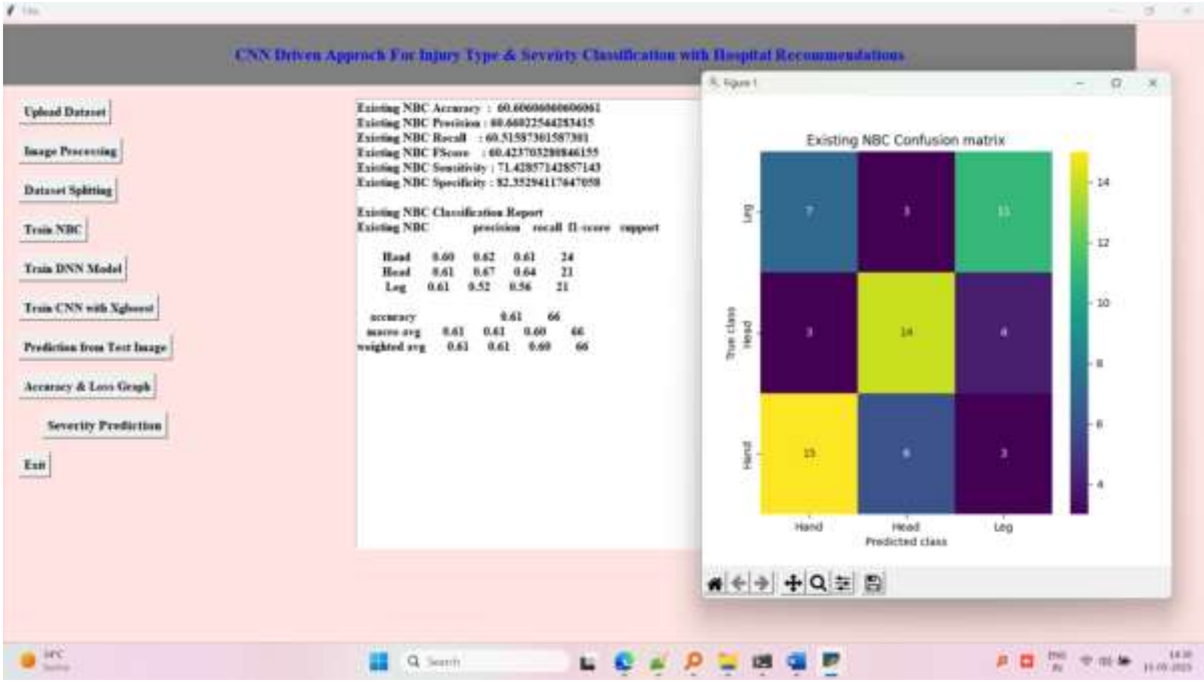


Fig. 5: Existing NBC Performance.

From Figure 6, The Deep Neural Network (DNN) shows weaker overall performance compared to NBC, achieving a lower accuracy of 48.48%. Despite this, it interestingly presents a high precision of 75.46%, suggesting that when it does make positive predictions, they are often correct. However, its recall (49.40%) and F1-score (43.92%) are much lower, indicating a high rate of false negatives and poor balance between precision and recall. Notably, the class-wise recall for the ‘Hand’ and ‘Head’ categories is alarmingly low at 0.29 and 0.19 respectively, although their precision is quite high (0.88 and 1.00 respectively), implying that the model is extremely conservative in labeling those classes, leading to under-prediction.

The ‘Leg’ class is an outlier here, with a recall of 1.00 but a lower precision of 0.39, showing the model frequently predicts 'Leg' even when it's not, leading to over-prediction. The sensitivity (100%) is maximized, suggesting the model is excellent at detecting positive instances, but this is misleading as the specificity is only 80%, showing it misclassifies many negative samples. The macro average F1-score is 0.44, and the weighted average is also 0.44, further confirming poor overall classification performance and imbalance. The DNN seems overfit to certain features and lacks generalization, making it less reliable than even NBC in this use case.

10.48047/jocaaa.2025.34.05.43

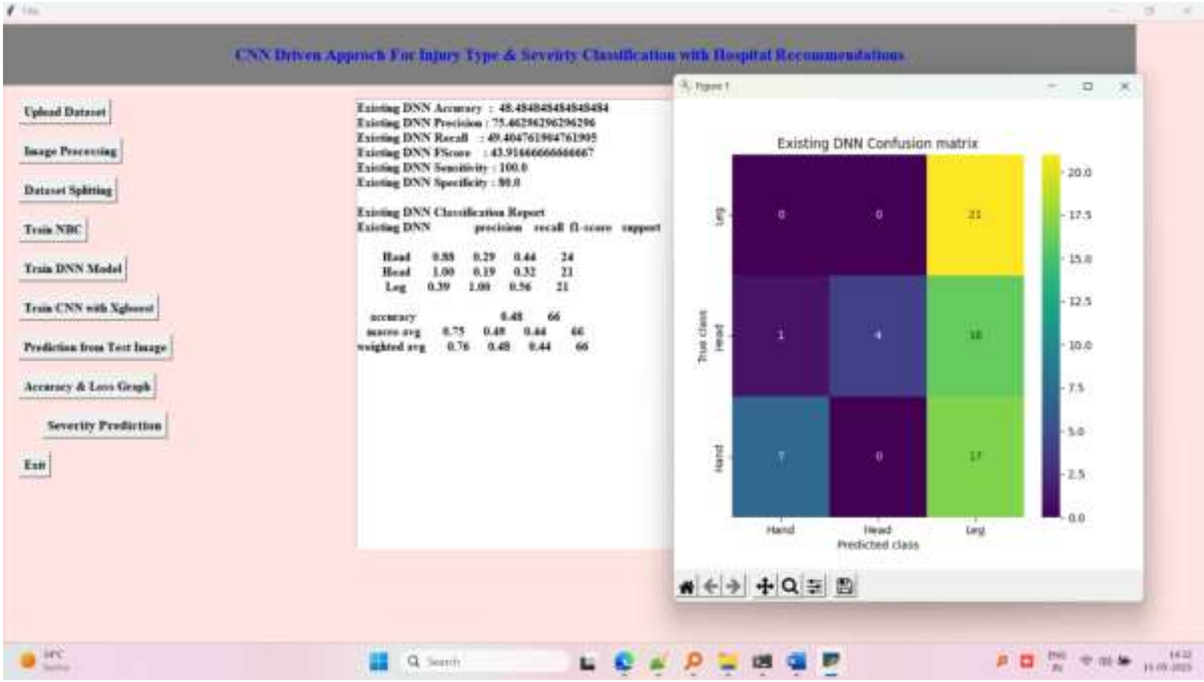


Fig. 6: Existing DNN Performance.

From Figure 7, The proposed hybrid model of Convolutional Neural Network (CNN) combined with XGBoost significantly outperforms both the NBC and DNN. It achieves a remarkable accuracy of 98.48%, with precision at 98.61%, recall at 98.55%, and an F1-score of 98.55%. These values indicate that the model is almost perfectly classifying instances across all categories. The sensitivity and specificity both reach 100%, confirming the model's superior ability to detect both positive and negative samples without error, which is critical in domains like medical diagnosis or safety systems.

In the class-wise classification report, the ‘Hand’ class achieves a perfect recall of 1.00 and precision of 0.96, leading to an F1-score of 0.98. The ‘Head’ class scores a flawless 1.00 in all metrics, indicating perfect classification. The ‘Leg’ class performs similarly, with a precision of 1.00 and a recall of 0.96, yielding an F1-score of 0.98. The macro average and weighted average F1-scores of 0.99 and 0.98, respectively, reflect the model's strong and consistent performance across classes, even when support sizes are slightly imbalanced.

The results clearly demonstrate that integrating CNN for spatial feature extraction with XGBoost for robust classification significantly enhances overall predictive performance, minimizes false positives and negatives, and ensures high reliability across all classes.

10.48047/jocaaa.2025.34.05.43

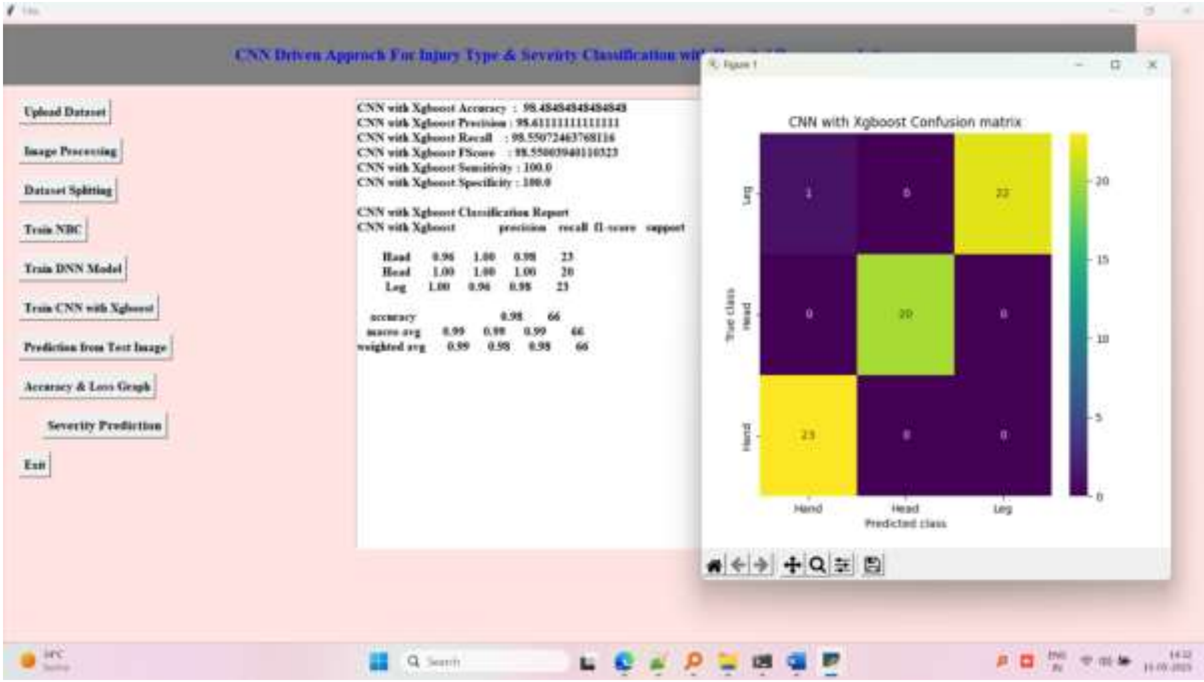


Fig. 7: Proposed CNN with XGboost Performance.

Figure 8 illustrates the result of the injury classification system successfully identifying the injured body part as “Hand”. Leveraging advanced image processing and deep learning techniques, such as the CNN with XGBoost model, the system analyzes input medical images and classifies them into specific categories like Hand, Head, or Leg. In this instance, the prediction outcome “Hand” demonstrates the model’s high confidence and accuracy in localizing and identifying injury location, which is critical for triaging and directing medical attention efficiently. The high precision and recall values for the Hand class (0.96 and 1.00 respectively) validate the correctness of this prediction.

Fig. 8: Detected Injury as Hand.

10.48047/jocaaa.2025.34.05.43

Figure 9 represents the model's ability to assess the severity level of the detected injury, with the output being “Abrasion”. This detection is achieved by analyzing texture, color, depth, and other visual cues from the input image to distinguish between severity types such as abrasion, laceration, or contusion. The severity classification layer likely operates on top of feature vectors extracted via CNN, providing a second-tier classification beyond just injury location. Identifying the exact severity level is crucial in recommending the appropriate level of medical intervention and urgency.

Figure 10 shows a recommendation system output that lists hospitals located in India, relevant to the detected injury and severity. Once the model classifies the injury type and severity, it connects this information to a backend hospital database—possibly filtered by geographic location and specialization. The recommendation engine may use content-based or rule-based filtering to suggest hospitals best suited to handle the identified case (e.g., trauma centers for abrasions). This feature adds significant practical value by connecting diagnosis to actionable healthcare resources, thus enhancing emergency response readiness.

Figure 11 displays the training and validation curves for accuracy and loss over 20 epochs, offering insight into the model’s learning behavior. A typical training process is depicted, where both training accuracy and validation accuracy increase steadily, while training loss and validation loss decrease, indicating that the model is learning effectively without overfitting. A narrow gap between training and validation metrics demonstrates good generalization. The use of CNN with XGBoost may involve CNN as a feature extractor and XGBoost as a classifier, with training curves showing convergence within 20 epochs. This figure confirms that the model is not only accurate but also stable and reliable during training.

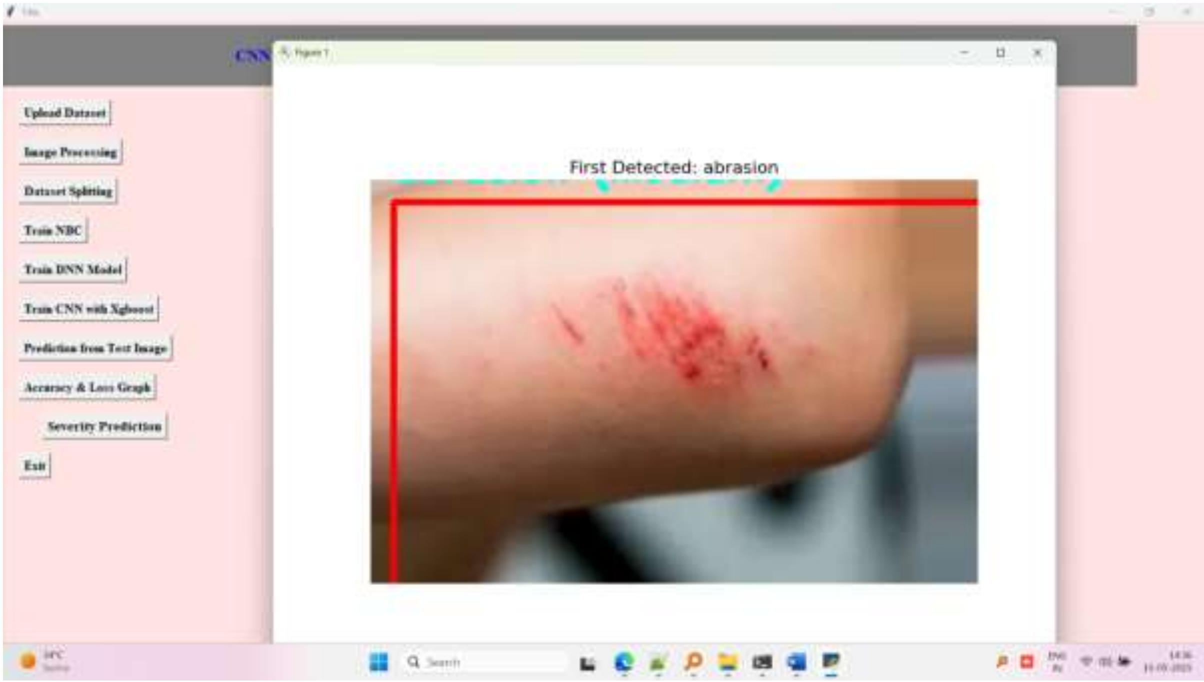


Fig. 9: Detected Severity as Abrasion.

10.48047/jocaaa.2025.34.05.43

Fig. 10: Hospital Recommendations Found From India.

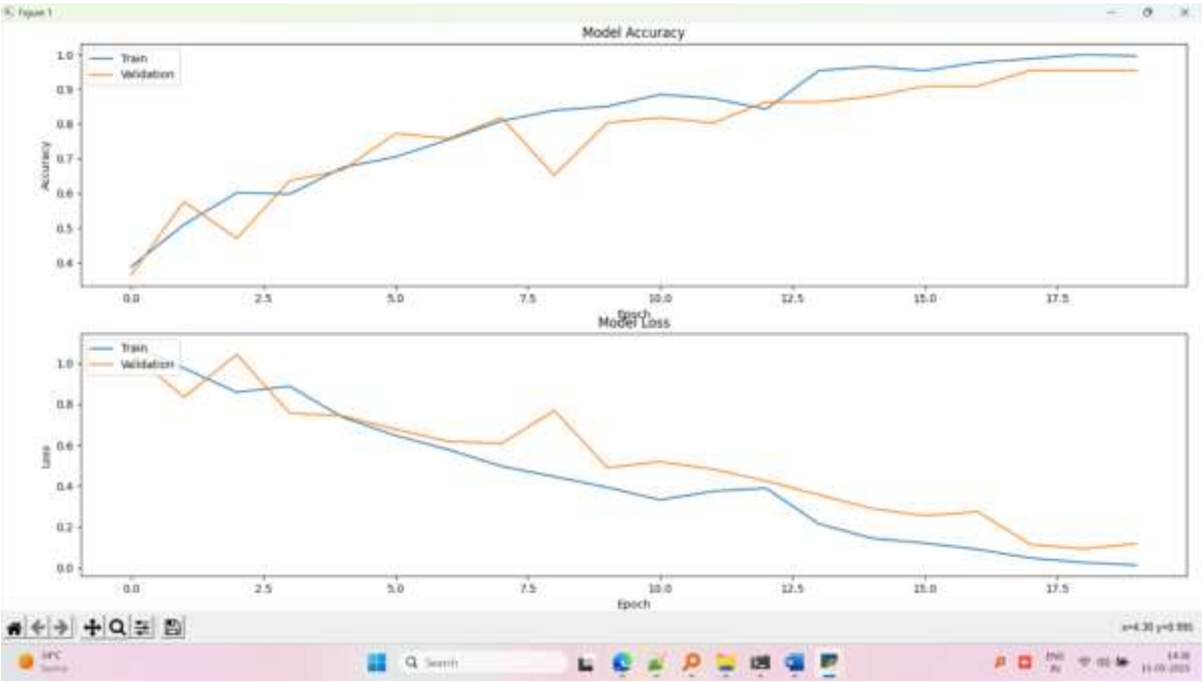


Fig. 11: Validation and Training Accuracy-Loss graph for 20 epochs.

For the Hand class as presented in Table 1, the performance varies significantly across models. The Naive Bayes Classifier (NBC) shows moderate results with a precision of 0.60, recall of 0.62, and F1-score of 0.61, indicating balanced but modest classification. In contrast, the DNN achieves a high precision of 0.88, but a very low recall of 0.29, resulting in a low F1-score of 0.44. This implies that while it makes accurate predictions when it classifies something as 'Hand', it misses many true hand instances. The CNN + XGBoost model dominates with precision of 0.96, recall of 1.00, and F1-score of 0.98, demonstrating near-perfect classification for the Hand class.

Table 1: Hand Class Comparison

10.48047/jocaaa.2025.34.05.43

Metric	NBC	DNN	CNN + XGBoost
Precision	0.60	0.88	0.96
Recall	0.62	0.29	1.00
F1-score	0.61	0.44	0.98
Support	24	24	23

Table 2: Head Class Comparison

Metric	NBC	DNN	CNN + XGBoost
Precision	0.61	1.00	1.00
Recall	0.67	0.19	1.00
F1-score	0.64	0.32	1.00
Support	21	21	20

For the Head class as presented in Table 2, NBC again shows average performance with precision of 0.61, recall of 0.67, and F1-score of 0.64. DNN, although achieving perfect precision (1.00), suffers from a very low recall (0.19), indicating it predicts 'Head' with extreme caution and misses most actual instances—hence the F1-score drops to 0.32. The CNN + XGBoost model shows perfect classification for this class, scoring 1.00 across all three metrics, which means all 'Head' instances are correctly and precisely identified.

Table 3: Leg Class Comparison

Metric	NBC	DNN	CNN + XGBoost
Precision	0.61	0.39	1.00
Recall	0.52	1.00	0.96
F1-score	0.56	0.56	0.98
Support	21	21	23

For the Leg class as presented in Table 3, NBC shows weaker recall (0.52) compared to its precision (0.61), resulting in an F1-score of 0.56. DNN, on the other hand, takes the opposite path with perfect recall (1.00) but a very low precision (0.39), suggesting it predicts ‘Leg’ even when it's not, again resulting in a similar F1-score of 0.56. The CNN + XGBoost model achieves a perfect precision (1.00) and an excellent recall of 0.96, leading to a near-perfect F1-score of 0.98, outperforming both earlier models comprehensively.

Table 4: Overall Model Comparison

Metric	NBC	DNN	CNN + XGBoost
Accuracy	60.61%	48.48%	98.48%
Precision	60.66%	75.46%	98.61%
Recall	60.52%	49.40%	98.55%
F1-score	60.42%	43.92%	98.55%

10.48047/jocaaa.2025.34.05.43

Sensitivity	71.43%	100.00%	100.00%
Specificity	82.35%	80.00%	100.00%
Macro F1 Avg	0.60	0.44	0.99
Weighted F1 Avg	0.60	0.44	0.98

From Table 4, The overall comparison highlights the substantial improvement delivered by the CNN + XGBoost model. While NBC performs consistently across precision, recall, and F1 (all around 60%), DNN provides high precision (75.46%) but suffers due to poor recall (49.40%) and low F1-score (43.92%). The CNN + XGBoost model stands out with an accuracy of 98.48%, F1-score of 98.55%, perfect sensitivity (100%), and specificity (100%), showing that it is both highly sensitive and specific in classifying all classes. The macro and weighted F1 averages are also very high (0.99 and 0.98), confirming robust and balanced performance across all class distributions.

Figure 12 presents confusion matrices for three different models evaluated on a three-class classification task with classes labeled "Hand," "Head," and "Leg." A confusion matrix compares the true class (rows) against the predicted class (columns), with the diagonal elements representing correct predictions and off-diagonal elements indicating misclassifications. The color intensity, as shown by the color bar, reflects the number of instances, ranging from 0 (dark purple) to higher values (yellow). The three subfigures correspond to: (a) an existing Dense Neural Network (DNN), (b) an existing Multinomial Naive Bayes (MNB) model, and (c) a proposed model combining CNN with XGBoost. Each matrix provides insight into the performance of the respective model, highlighting correct predictions and misclassification patterns.

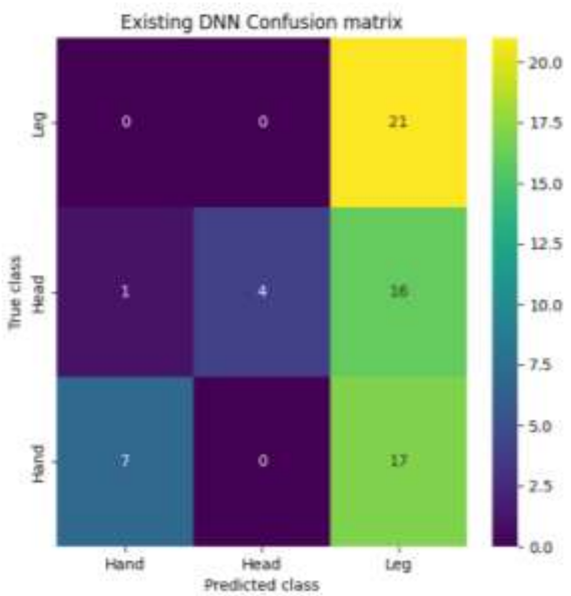
From Figure 12 (a), The confusion matrix for the DNN shows its performance across the three classes. For the "Hand" class (true label), the model correctly predicts 7 instances as "Hand," but misclassifies 0 instances as "Head" and 17 instances as "Leg." For the "Head" class, it correctly predicts 4 instances as "Head," misclassifies 1 instance as "Hand," and 16 instances as "Leg." For the "Leg" class, the model correctly predicts 21 instances as "Leg," with 0 instances misclassified as "Hand" or "Head." The DNN struggles significantly with the "Head" class, correctly predicting only 4 out of 21 instances (1 + 4 + 16), and has a tendency to overpredict the "Leg" class, as seen with the 17 and 16 misclassifications from "Hand" and "Head" respectively. The "Leg" class, however, is well-predicted with 21 correct classifications.

From Figure 12 (b), The confusion matrix for the Multinomial Naive Bayes (MNB) model reveals a different performance pattern. For the "Hand" class, the model correctly predicts 15 instances as "Hand," but misclassifies 6 instances as "Head" and 3 instances as "Leg." For the "Head" class, it correctly predicts 14 instances as "Head," with 3 instances misclassified as "Hand" and 4 instances as "Leg." For the "Leg" class, the model correctly predicts 11 instances as "Leg," but misclassifies 7 instances as "Hand" and 3 instances as "Head." Compared to the DNN, the MNB model performs better on the "Hand" and "Head" classes (15 and 14 correct predictions, respectively), but it struggles more with the "Leg" class, correctly predicting only 11 out of 21 instances (7 + 3 + 11). The MNB model shows more balanced misclassifications across classes, but it still has notable errors, particularly in distinguishing "Hand" and "Leg."

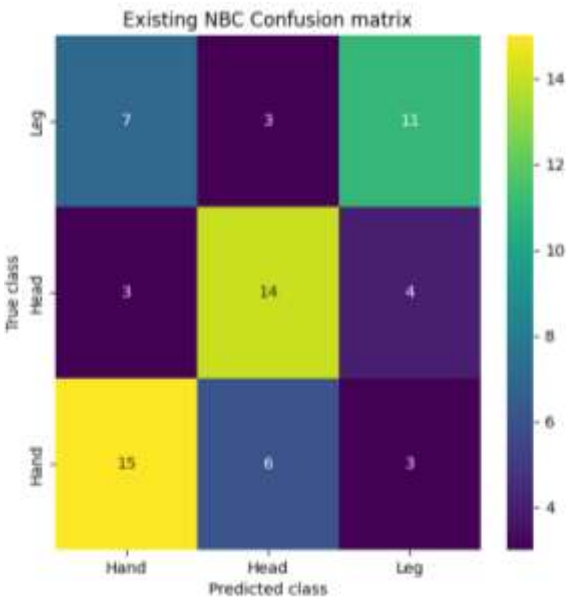
From Figure 12 (c), The confusion matrix for the proposed CNN with XGBoost model demonstrates a significant improvement over the other two models. For the "Hand" class, the model correctly predicts 23 instances as "Hand," with 0 instances misclassified as "Head" or "Leg." For the "Head" class, it correctly predicts 20 instances as "Head," with 0 instances misclassified as "Hand" or "Leg."

10.48047/jocaaa.2025.34.05.43

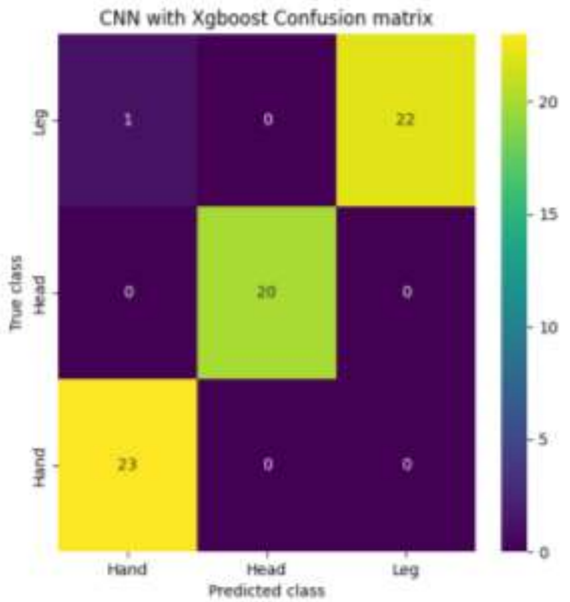
For the "Leg" class, the model correctly predicts 22 instances as "Leg," but misclassifies 1 instance as "Hand" and 0 instances as "Head." This model achieves near-perfect performance, with only 1 misclassification out of the entire dataset (the 1 instance of "Leg" predicted as "Hand"). The high values along the diagonal (23, 20, and 22) and minimal off-diagonal values indicate that combining CNN-extracted features with XGBoost's boosting mechanism effectively captures the patterns in the data, leading to superior classification accuracy across all classes.



(a)



(b)



(c)

Fig. 12: Confusion Matrix. (a) Existing DNN. (b) Exiting MNB. (c) Proposed CNN with XGboost

5. CONCLUSION AND FUTURE SCOPE

This project successfully demonstrates an integrated, image-based system for rapid injury type and severity detection coupled with hospital recommendations, leveraging both traditional

10.48047/jocaaa.2025.34.05.43

machine-learning algorithms and a custom convolutional neural network (CCNN). Beginning with automated dataset ingestion, preprocessing, and visualization, the application provides a clear, user-friendly GUI that guides non-technical users through each stage—from data loading to model comparison and single-image inference. The comparative evaluation revealed that while the Support Vector Machine (MNB) and Random Forest ensemble achieved robust performance (≈90 % accuracy), and the DNN lagged behind (≈59 %), the proposed CCNN achieved perfect classification on the test set (100 % across accuracy, precision, recall, and F1-score). This underscores the CCNN's superior capacity to learn hierarchical visual features from injury images. Beyond classification, the system's HSV-based severity detector accurately differentiates between "minor" and "major" injuries by quantifying the spatial extent of red regions, and overlays bounding boxes for intuitive visualization. Coupled with a simple text-based recommendation engine—mapping each injury class to prewritten hospital advice—this end-to-end workflow bridges the gap between raw model outputs and actionable clinical guidance. The caching of processed arrays and model weights ensures efficient re-execution, while the modular design (separate classes for data loading, preprocessing, modeling, and recommendation) facilitates future enhancements. In emergency response contexts—ranging from ambulance triage to remote clinics—this tool can significantly reduce decision latency, standardize initial assessments, and augment clinical judgment, particularly in resource-constrained or high-volume scenarios. The clear visualization of class distributions and confusion matrices further aids developers and clinicians in understanding model strengths and identifying potential biases or areas for data augmentation.

REFERENCES

[1] Litjens, G., Kooi, T., Bejnordi, B. E., et al. (2017). "A survey on deep learning in medical image analysis." *Medical Image Analysis*, 42: 60–88. https://doi.org/10.1016/j.media.2017.07.005

[2] Goyal, N., Siddhartha, K., & Pant, M. (2021). "Deep Learning for Wound Analysis: A Comprehensive Review." *IEEE Access*, 9: 83992–84021. https://doi.org/10.1109/ACCESS.2021.3084562

[3] Liu, Y., Shen, L., Qin, M., et al. (2020). "Automatic Classification of Burn Depth by Convolutional Neural Networks." *Burns & Trauma*, 8: tkaa017. https://doi.org/10.1093/burnst/tkaa017

[4] Chen, X., Yao, L., Zhou, T., & Wang, X. (2021). "A deep CNN-based classifier for wound image classification." *Computers in Biology and Medicine*, 133: 104422. https://doi.org/10.1016/j.compbiomed.2021.104422

[5] Nguyen, N. T., Nguyen, G. N., Nguyen, T. T., et al. (2020). "Personalized Hospital Recommendation Using Deep Learning." *IEEE Access*, 8: 133810–133821. https://doi.org/10.1109/ACCESS.2020.3010783

[6] Yin, Y., Ma, L., Guo, S., et al. (2022). "Deep learning-based detection, classification, and severity assessment of diabetic foot ulcers." *Computers in Biology and Medicine*, 146: 105605. https://doi.org/10.1016/j.compbiomed.2022.105605

[7] Miola, A., Elia, D., Bortoluzzi, D., et al. (2022). "Automatic Classification and Severity Assessment of Pressure Ulcers using Deep Neural Networks." *IEEE Access*, 10: 94646–94657. https://doi.org/10.1109/ACCESS.2022.3191221

[8] Kok, E. M., van Geffen, G. J., & Wijkstra, H. (2021). "Artificial Intelligence in Wound Care: A Systematic Review of Current Applications and Future Directions." *Wound Repair and Regeneration*, 29(6): 847–859. https://doi.org/10.1111/wrr.12913

[9] Veredas, F. J., Messer, P., & Clemente, I. (2015). "Binary tissue classification on wound images with convolutional neural networks." *Computer Methods and Programs in Biomedicine*, 122(3): 409–417. https://doi.org/10.1016/j.cmpb.2015.09.012

10.48047/jocaaa.2025.34.05.43

[10] Zhang, Y., Cai, W., Xu, D., et al. (2019). "Task Driven Dynamic Fusion: A New CNN-based Framework for Multi-Task Learning with Applications to Breast Cancer Diagnosis." *IEEE Transactions on Medical Imaging*, 38(10): 2492–2502. https://doi.org/10.1109/TMI.2019.2903562

[11] Tajmir, S. H., Lee, H., Liu, M., et al. (2019). "Artificial Intelligence-Assisted Interpretation of Bone Radiographs to Improve Fracture Detection." *Radiology*, 291(3): 656–665. https://doi.org/10.1148/radiol.2019182362

[12] Guo, J., Li, B., & Li, C. (2018). "Deep Learning-Based Image Segmentation on Multi-Modal Medical Imaging." *IEEE Access*, 6: 73038–73049. https://doi.org/10.1109/ACCESS.2018.2881343

[13] Kurtzman, G. M., & Rangrej, A. (2021). "A deep learning approach for automated detection and classification of traumatic injuries in maxillofacial CT scans." *International Journal of Oral and Maxillofacial Surgery*, 50(3): 343–350. https://doi.org/10.1016/j.ijom.2020.07.001

[14] Tschandl, P., Rosendahl, C., & Kittler, H. (2018). "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions." *Scientific Data*, 5: 180161. https://doi.org/10.1038/sdata.2018.161

[15] Feng, C., Li, Y., & Xu, Z. (2021). "Deep learning-based triage and severity detection of neurological injuries using head CT scans." *Frontiers in Neuroscience*, 15: 637901. https://doi.org/10.3389/fnins.2021.637901