

Synergistic Integration of Advanced X-Ray Techniques for Enhanced Fracture Detection

Anusha Adhikari¹, Dipannita Dey², Dipyaman Goswami³, Shounak Chattopadhyay⁴, Asima Adak⁵, Anindya Sen⁶

^{1,2,3,4,5,6}Department of Electronics and Communication Engineering, Heritage Institute of Technology, Kolkata – 700107

Abstract

Deep learning models including VGG-16, VGG-19, YOLO v8, and a customized model are employed for fracture detection on the processed images and further utilizing Streamlit's user-friendly interface, users can easily input X-ray images for pre-processing and fracture detection. Our study addresses the imperative role of X-ray imaging in medical diagnosis, emphasizing the significance of pre-processing techniques to enhance image clarity. This work developed an interactive web application to access X-ray based medical images for diagnostic purpose using Streamlit, a Python library. Incorporating a bounding box feature, our system enables fracture localization using the pretrained VGG-19 model. Additionally, users can download the processed images, facilitating efficient fracture identification and seamless pre-processing. This integrated approach enhances the diagnostic capability of X-ray imaging, enabling accurate fracture localization and fostering improved patient care through the amalgamation of state-of-the-art deep learning models and advanced image processing techniques.

Keywords: *X-ray Image, Preprocessing, Morphological gradient, Sobel, edge, Canny, edge detection, Fracture Detection, VGG-19, VGG-16, YOLOv8, Deep Learning models, Streamlit, App, Site.*

1. INTRODUCTION

X-ray imaging is essential for diagnosing medical conditions, but image complexity frequently reduces clarity and necessitates sophisticated pre-processing methods. The advanced pre-processing techniques used in this work to prepare X ray pictures for medical analysis are the main topic. Our project explores strategies for Morphological Gradient, Sobel Edge Detection, and Canny Edge Detection in light of the difficulties presented by the intricate nature of X-ray images. Our goal in carefully investigating and applying these techniques is to guarantee accurate and dependable outcomes in the pre-processing of X-ray images.

Our effort intended to improve fracture detection methods in medical imaging, drawing inspiration from the paper "A Comparative Study of Multiple Deep Learning Algorithms for Efficient Localization of Bone Joints in the Upper Limbs of Human Body" [1]. Understanding how image quality affects detection accuracy [3], we concentrated on pre-processing each unique X-ray image to create a uniform dataset. By resolving problems caused by different colour scale,

sharpness, noise intensity, and image size, our method enhanced detection outcomes. Beyond technological advancement, our X-ray pre-processing initiative aims to support the health care community with comparable fracture diagnosis difficulties [2]. Our goal is to improve patient outcomes and diagnostic accuracy in medical imaging by revolutionising the field through image quality optimisation. This work ensures responsible innovation, to abide by regulatory compliance, and guarantees the ethical use of technology in healthcare. In the end, our initiative highlights the benefits of combining technology innovation with healthcare by providing away to use cutting-edge diagnostic technologies to improve patient care and overall health.

Further augmenting this endeavour is the creation of a comprehensive dataset comprising of pre-processed X-Ray images, designed to serve as a valuable resource for advanced clinical research and contributions. The primary objective was to establish a diverse dataset for potential future research endeavours. Extensive efforts were undertaken to source data from various hospitals, albeit facing numerous rejections initially. Ultimately, assistance was procured from Techno India DAMA Healthcare and Medical Centre. Despite persistent endeavours spanning several months, permissions from other hospitals and clinics remained elusive. As a recourse, a selection of images was acquired from the FracAtlas dataset [9] available online. This resultant dataset amalgamates high-quality DICOM images obtained from DAMA Hospital with those from the FracAtlas dataset. Each image underwent meticulous manual pre-processing to eliminate noise and redundancies within the dataset. This dataset is meticulously structured into two distinct components: fractured and non-fractured sub-datasets within ‘Canny’(Canny Edge Detection applied) and ‘Morph’(Morphological Gradient applied) datasets respectively, diligently tailored for training deep learning models. Additionally, the integration of a pre-trained VGG19 [11] model for fracture detection significantly enhances the project’s analytical capabilities. This model is seamlessly integrated into an interactive user interface, empowering users to selectively classify regions of interest as fractured or non-fractured.

We have extensively evaluated the three image processing methods—Morphological Gradient, Sobel Edge Detection, and Canny Edge Detection—that have been tested in this project. Moreover, this project encompasses a rigorous comparative analysis of various deep learning architectures, including VGG16, VGG19, YOLOv8, [7,8] and a custom model. These architectures have been rigorously trained and evaluated to deliver comprehensive comparative statistics, showcasing their respective capabilities in fracture detection. The user interface serves as a structured pathway, offering users a seamless and user-friendly experience in applying the image processing techniques and thus in detecting fractures from those processed X-ray images.

Our overarching goal is to deliver a robust and efficient platform for the automated pre-processing of X-ray images, thereby significantly enhancing diagnostic and therapeutic capabilities within the medical realm. By amalgamating cutting-edge expertise in image processing with the practical demands of clinical practice, this project aspires to make a substantial contribution to the continuous advancement of healthcare through innovative technological solutions and have a significant impact in the research domain. Leveraging image processing techniques allows for the extraction of critical information from X-ray images, aiding in accurate diagnosis and treatment planning. Utilizing advanced models and algorithms facilitates the automated detection of fractures and abnormalities, thereby enhancing doctors’ ability to accurately recognize and diagnose such conditions, ultimately improving overall diagnostic accuracy.

2. EXPERIMENTAL DETAILS

2.1 IMAGE PRE PROCESSING

Image processing refers to the manipulation and analysis of digital images using computer algorithms. In the context of fracture detection, image processing plays a crucial role in enhancing medical imaging techniques for identifying and analysing fractures in X-ray images. [4] indicates that undesired elements in the image, such as noise, are eliminated, and adjustments like enhancing contrast are made to produce an image suitable for subsequent feature extraction.

The benefits of image processing in fracture detection are numerous. Firstly, it allows for the enhancement of X-ray images, improving the clarity and visibility of fractures, which can be particularly helpful in detecting subtle or complex fractures that may be missed by the naked eye. Secondly, image processing techniques can automate the process of fracture detection, reducing the time and effort required by healthcare professionals to identify and diagnose fractures accurately. This automation also helps in standardizing the diagnostic process, leading to more consistent and reliable results.

Furthermore, image processing algorithms can be trained using machine learning techniques to recognize patterns indicative of fractures, contributing to the development of intelligent diagnostic systems. These systems can assist radiologists and clinicians by providing them with automated fracture detection tools, improving diagnostic accuracy and efficiency.

2.1.1 MORPHOLOGICAL GRADIENT

In image processing, the morphological gradient (Morph) is the difference between a binary picture's eroded and dilated versions [5]. Every pixel in the binary input image at first indicates either the foreground (white) or the background (black). A structural element is defined for the erosion and dilation processes; this is usually a small binary matrix, known as a kernel. Erosion shrinks the edges of the foreground pixels, whereas dilation enlarges them. Pixel by pixel computation of the difference between the dilated and eroded pictures produces the gradient image. It has the following mathematical representation:

$$\text{Gradient} = \text{DilatedImg} - \text{ErodedImg}$$

By emphasising regions with significant intensity fluctuations, which usually correspond to the edges of objects in the original image, this gradient image highlights object boundaries. It's a helpful tool for image analysis tasks, including feature extraction and edge identification.

Changes made include that if "Morphological" is selected, contrast is increased initially, and then morphological gradient processing is applied twice, using a 2x2 kernel the first time and a 3x3 kernel the second. Lastly, adjustments are made to contrast and brightness. Furthermore, bounding boxes are created on the processed photos by setting "draw_bbox" to True. With the option to add bounding boxes to the processed photos, this user interface also allows for parameter adjustments for a clearer picture.

2.1.2 SOBEL EDGE DETECTION

The Sobel operator, named in honour of Irwin Sobel and Gary M. Feldman, uses pixel-by-pixel analysis of image intensity gradients to find edges. It finds areas of high spatial frequency, suggestive of edges, by calculating the direction and rate of change from light to dark through the use of 2-D spatial gradient assessment [4]. This provides information about edge likelihood and direction by revealing how abrupt or smooth transitions are. It convolves them with the input image using different 3x3 kernels for changes in the horizontal and vertical directions to approximate derivatives. G_x and G_y as images representing the horizontal and vertical derivative approximations, respectively. The calculations are represented as follows respectively:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \quad (1)$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \quad (2)$$

Here, A represents the original input image. The gradient magnitude, G , is calculated as the square root of the sum of squares of G_x and G_y :

$$G = (G_x^2 + G_y^2)^{1/2} \quad (3)$$

The direction of the gradient is calculated using:

$$\theta = \arctan2(G_y/G_x) \quad (4)$$

A value of 0 would indicate a vertical edge that is darker on the left side.

The changes are made with respect to the Sobel class is a subclass of `nn.Module` in PyTorch. It defines a 2D convolutional layer (`nn.Conv2d`) for both horizontal and vertical gradients, with 1 input channel, 2 output channels, and a 3x3 kernel size. The weight of the convolutional filter is a single tensor G that is formed by concatenating the modified Sobel kernels for G_x and G_y . After applying the Sobel filter to the input image, the forward function squares gradients, adds up all the channels, and takes the square root to determine the gradient magnitude. This forms Sobel Type 2. The original sobel filter is referred to as Sobel Type1. Sobel Type 2 uses G_x and G_y matrices different from those in Type 1. Depending on the type, the edge detection method uses OpenCV's Sobel (Type1) or PyTorch's Sobel class (Type 2, also referred to as 'Sobel' here) filtering to apply Sobel filtering on RGB pictures before returning the treated image. If `draw_bbox` is True, it additionally potentially draws a bounding box on the image. This gives you the freedom to experiment with various edge detection strategies and tactics.

2.1.3 CANNY EDGE DETECTION

John F. Canny developed the well-known Canny edge detection (Canny) method in 1986. It is recognised for its precision in edge recognition while reducing noise and false positives. It generates a binary image [14]. There are five major steps involved: To reduce noise, the image is first smoothed using a **Gaussian filter** [3]. To find variations in pixel intensity, **intensity gradients**

are then computed throughout the image. **Gradient magnitude thresholding** [6] is then used to eliminate erroneous answers. Potential edges are identified by **double thresholding** according to the gradient magnitudes of such edges. Lastly, **hysteresis-based edge tracking** ensures coherent edge detection by suppressing weak edges that are unconnected to strong ones. The algorithm's mathematical steps are as follows: it computes intensity gradients, applies gradient magnitude thresholding, double thresholds, and uses hysteresis for edge tracking. Robust edge identification over an extensive range of images is made possible by this all-encompassing method. The equation for a Gaussian filter kernel of size $(2k+1) \times (2k+1)$ is defined by:

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2}\right) \quad (5)$$

where, $1 \leq i, j \leq (2k+1)$

This feature enhances capabilities for image processing tasks like feature extraction and object detection by enabling specific threshold changes for Canny edge detection and enabling additional processing like morphological gradient processing and bounding box visualisation. If `draw_bbox` is True, it additionally potentially draws a bounding box on the image.

2.2 MODELS USED

2.2.1 VGG-16

Working with RGB images of 224x224 pixel dimensions, the VGG16 model is a well-known convolutional neural network (CNN) utilised in computer vision tasks [7]. In order to extract complicated features while maintaining spatial information, it uses 13 convolutional layers with Rectified Linear Unit (ReLU) activation, 3x3 filters [12], a stride of 1, and same-padding. Every two convolutional layers are followed by a stride of two downsample feature maps and max pooling layers with 2x2 windows. Then, three completely connected layers with 4096 neurons each and ReLU activation function as feature detectors. The last fully connected layer has 1000 neurons to represent the 1000 classes in ImageNet. Probabilities for multi-class classification are computed by the softmax layer at the output. VGG16's vast design, with almost 138 million parameters, allows it to recognise intricate patterns in images.

Three more Dense layers were added to the top three levels of the VGG16¹ architecture to customize it. These comprised a layer of 256 neurons and a layer of 128 neurons, each activated by ReLU. One neuron in the final layer—with binary classification—therefore had sigmoid activation. For integrity's sake, existing layers were set as non-trainable. The custom layers were trained to maximise performance by transfer learning after initial weights were established using ImageNet dataset weights.

2.2.2 VGG-19

Deep convolutional neural network (CNN) architecture VGG-19, an extension of VGG-16

¹For code, visit [this site](#).

developed by the University of Oxford's Visual Geometry Group, is widely used in computer vision tasks. It uses 16 convolutional layers to process RGB images with 224×224 pixels [15], and then activates each layer with a Rectified Linear Unit (ReLU). To maintain spatial dimensions, these layers use 3×3 filters with a stride of 1 [7]. They also use same-padding. Following every two convolutional layers, feature map sizes are decreased by max pooling layers with 2×2 windows and a stride of 2. After that, three completely connected layers with 4096 neurons each with ReLU activation function as feature extractors, and for each of ImageNet's 1000 classes, there is a final fully connected layer with 1000 neurons. Multi-class categorization is made possible by the soft max output layer. With over 143 million parameters, VGG-19's depth allows it to efficiently catch detailed patterns in images, which makes it appropriate for applications involving complex image identification.

Three more Dense layers were added to the top layers of the VGG-19² architecture to customise it. These consisted of a layer of 128 neurons with ReLU activation and a layer of 256 neurons with ReLU activation each. One neuron in the final layer—with binary classification—had sigmoid activation. For integrity's sake, existing layers were set as non-trainable. The custom layers were trained to maximise performance by transfer learning after initial weights were established using ImageNet dataset weights.

2.2.3 YOLO-v8

The core architecture of YOLOv8, an improved Darknet53 variant with Cross-Stage Partial Connections (CSP) [8] that directly connect early and later layers to improve gradient flow, feature learning, and information flow, is called CSPDarknet53. With this approach, details in input photos can be gradually captured, ranging from simple edges to intricate shapes. Although it isn't mentioned clearly, YOLOv8 is probably built like architectures like the Feature Pyramid Network (FPN) with a "neck" component to combine feature maps from different depths, increasing feature representation that's essential for accurate object detection. YOLOv8's "head" manages final maps for bounding box predictions and object categorization utilising anchor boxes. To guarantee robustness and accuracy in object identification tasks, it incorporates training approaches like AutoAnchor and multiscale training.

A code is initiated to alter the backbone, CSPDarknet53 (of YOLOv8³ architecture), which is well-known for its effectiveness and improved data transfer via CSP.

2.2.4 CUSTOM MODEL

The input layer of the network starts with 100×100 grayscale images processed by Conv2D utilising 64 3×3 filters and ReLU activation, keeping spatial dimensions with 'same' padding and (1,1) strides. After downsampling dimensions by half with MaxPooling2D, there is another layer of Conv2D with 128 filters. A second Max-Pooling2D layer replaces this, cutting dimensions in half. After MaxPooling2D, the dimensions of a third Conv2D layer with 256 filters are lowered to 12×12 pixels while maintaining spatial information .

²For code, visit [this site](#).

³For code, visit [this site](#).

A final MaxPooling2D layer reduces size to 6x6 pixels after the fourth Conv2D layer with 128 filters refines features. Data is converted to 1D for Dense layers (256 and 128 neurons with ReLU activation for feature extraction) by the Flatten layer. For regularisation, a Dropout layer with a 20% rate is included. The output layer predicts class probabilities using sigmoid activation, which is suitable for binary classification⁴.

It performs hierarchical feature extraction (strides) and capable of high-level feature learning after being flattened into a 1D vector by several convolutional and max-pooling layers. To reduce overfitting, dropout regularisation is used, and a sigmoid activation function is used for binary classification in the final output layer.

2.3 STREAMLIT

Streamlit [10] is specifically designed for machine learning engineers and data scientists who want to create web applications to showcase their work or share insights. Some of the features of Streamlit⁵ are the following: **(i) Simple API:** Build an app in a few lines of Python code. **(ii) Widgets:** Adding interactive elements (like sliders, date pickers, and radio buttons) is straightforward. **(iii) Automatic Updates:** As you iteratively save your source file, your app automatically updates. **(iv) Deployment:** Effortlessly share, manage, and deploy your apps directly from Streamlit.

We developed a web application using Streamlit, a Python library that enables the creation of interactive web interfaces with minimal coding effort. The main components include: **(i) Interface Setup:** Streamlit simplifies the creation of web apps. The `st.set_page_config (layout='wide')` configures a spacious layout, optimizing the display of elements like images and controls. **(ii) Header Section:** The header employs a two-column layout for balanced visual hierarchy and organized element access. **(iii) Title and Instructions:** The `st.title` function prominently displays the main title, while `st.caption` provides comprehensive user guidance and instructions, facilitating onboarding and workflow understanding. **(iv) Sidebar and User Inputs:** The sidebar functions as a control panel with inputs such as file uploads (`st.sidebar.file_uploader`), dropdown menus (`st.sidebar.selectbox`), and sliders (`st.sidebar.slider`), enabling customized image processing. **(v) Main Image Display:** Uploaded X-ray images are displayed in one column, with processed images in another, allowing real-time observation of modifications. An interactive zoom feature enhances analysis by allowing exploration of specific image areas. **(vi) Image Processing Techniques:** Techniques such as Morph, Sobel, and Canny are supported. Users can select methods and adjust settings via the sidebar. A bounding box feature enables detailed region analysis. **(vii) Interactive Features:** Action buttons (`st.sidebar.button`) initiate tasks, and feedback messages inform users of the results. **(viii) Downloading Modified Image:** Users can download processed images via a button, with base64 encoding facilitating further analysis or sharing.

This design provides an intuitive and user-friendly experience for processing and analysing X-ray images of bones.

⁴For code, visit [this site](#).

⁵Complete code (along with image processing) can be found [here](#).

3. RESULTS AND DISCUSSIONS

3.1 PREPROCESSING TECHNIQUES COMPARISON

For edge identification applications, the **morphological gradient (Morph)** approach is straight forward and simple to use. It does this by calculating the difference between eroded and dilated versions of a picture, which successfully identifies probable edges by highlighting areas of notable intensity change. For basic edge detection tasks, its simple algorithmic processes involving gradient computation, erosion, and dilation make it appropriate. Its limited capacity to handle intricate edge structures, however, can provide problems in situations when precise edge localization is essential.

Sobel edge detection (Sobel) is a flexible option for a range of image processing jobs because it achieves a balance between sensitivity and complexity. It retains computational efficiency while providing useful edge information by determining the gradient’s magnitude and direction. With the help of its pre-programmed Sobel kernels, edge features in grayscale or RGB images can be captured in an organised manner for edge detection. This method is well-suited for fracture detection applications, where accurate edge localization is crucial, due to its moderate sensitivity and capacity to provide edge direction information.

Canny edge detection (Canny) is a potent tool for accurate edge detection jobs because of its great sensitivity and efficacy in noise reduction. Accurate edge localization in intricate image structures is ensured by its multi-stage approach, which combines edge tracking, thresholding, gradient calculation, and Gaussian smoothing. Canny edge detection has superior results in identifying edges with different intensities and orientations, despite being susceptible to parameter adjustment. This makes it especially useful for fracture identification, where precise diagnosis depends on the ability to discern minute edge features among noise.

In Figure 1, we can see the effect of the 3 image processing techniques namely Morph (1(b)), Sobel type 2 (1(c)), and Canny(1(c)) on the original DICOM X-ray image 1(a).

3.2 MODELS COMPARISON

From Figure 2 & 3 (where, TP - True Positive, TN - True Negative, FP - False Positive, FN-False Negative), we can calculate the performance metrics⁶ of the models' accuracy, precision and the recall for testing data undergoing Morphological gradient transformation and Canny edge transformation as given in Tables 1 & 2 respectively.

Table 1: Performance metrics for models under test (Morph)

Models	Accuracy	Precision	Recall ¹
VGG19	0.644	0.622	0.622
VGG16	0.683	0.698	0.644
CM ²	0.639	0.638	0.644
YOLOv8	0.851	0.847	0.854

Table 2: Performance metrics for models under test (Canny)

Models	Accuracy	Precision	Recall ¹
VGG19	0.857	0.831	0.764
VGG16	0.824	0.813	0.733
CM ²	0.665	0.666	0.594
YOLOv8	0.602	0.33	0.91

¹ F1 Score can be calculated using recall and precision, as it is the harmonic mean of the two.

²CM stands for Custom Model.

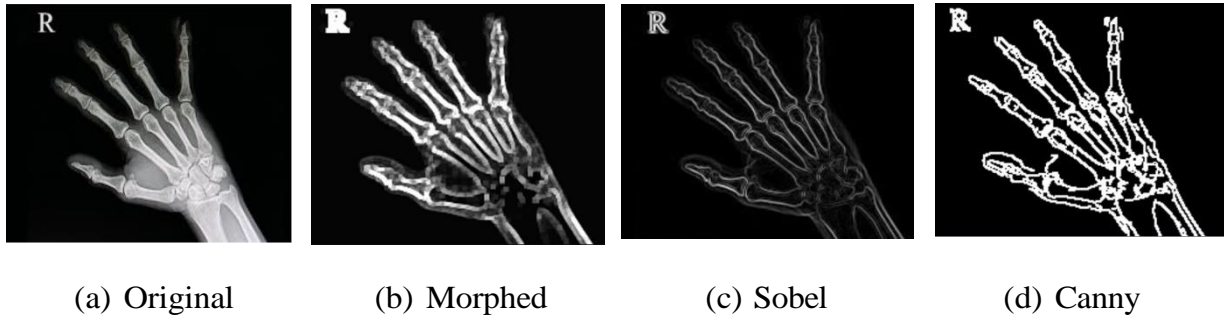


Figure 1: Image Processing

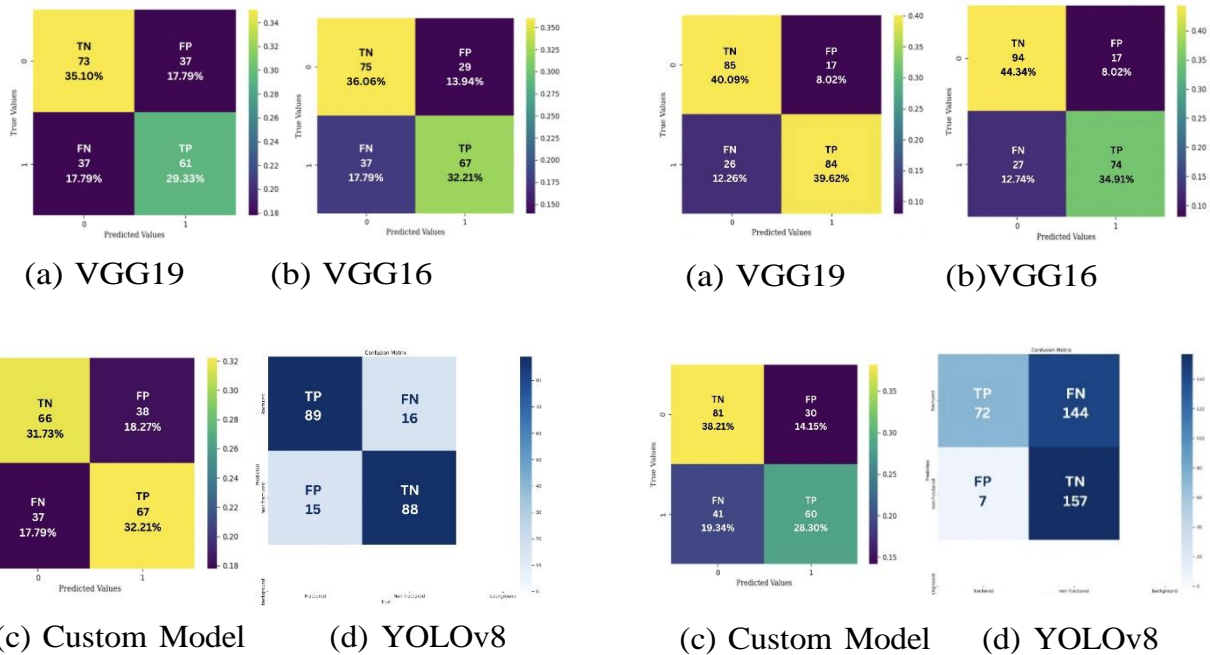


Figure 2: Confusion Matrices for the models under test for Morph dataset

Figure 3: Confusion Matrices for the models under test for Canny dataset

The study evaluates the effectiveness of machine learning models for fracture diagnosis that were trained on constrained X-ray image datasets, with an emphasis on recognising and resolving overfitting issues. The Morph and Canny datasets are used in the analysis.

As seen in Figure 2 and Table 1, YOLO v8 beats the Custom model in terms of accuracy⁷ and loss measures, however the graphs of VGG-19 and VGG-16 show plateau regions, which suggest learning saturation. All models for the Morph dataset exhibit overfitting, with VGG-19 displaying more prominent indications. The confusion matrices demonstrates the superior performance of YOLO v8 over the Custom model and also the overfitting problems with VGG-19.

Among the Custom model, VGG-16, YOLO v8, and VGG-19, VGG-19 performs the best; YOLO v8 shows uneven accuracy. The loss graphs show similar trends, with VGG-19 showing the least loss and VGG-16, Custom model, and YOLO v8 following closely behind. Confusion matrices verify that VGG-19 outperforms YOLO v8 in terms of accuracy, precision, recall, and F1 score. However, from Figure 3, Table 2 and the plots for the Canny dataset clearly demonstrate

⁷Accuracy and loss graphs of the models for both morphed and canny images can be found [here](#).

overfitting in all models, with training accuracy being higher than validation accuracy.

3.2.1 APPLICATION OVERVIEW

Users who upload X-ray photos of their bones for assessment can take advantage of the user-friendly interface that this Streamlit-based web application⁸ provides for image processing and analysis. For more precise control, users can choose from a variety of edge detection methods, including Canny Edge Detection, Morphological Gradient, and Sobel Edge Detection. They can also modify parameters like thresholds, contrast, brightness, and zoom factor. Clear headings, instructions, and control panels make for an easy-to-navigate hierarchical layout. Users may observe the results instantly thanks to real-time processing of uploaded photographs, while interactive zoom and bounding box options let them concentrate on particular regions. Horizontal and vertical adjustments enable accurate bounding box placement, facilitating in-depth examination of specific regions. The application uses algorithms or pre-trained models to process photographs inside the bounding box and extract pertinent information, of fracture detection—a crucial feature made possible by pixel analysis. In general, the application provides users with an intuitive interface and extensive image processing capabilities for medical picture analysis and diagnosis.

4. CONCLUSION

4.1 EFFECT OF IMAGE PROCESSING

Morphological gradient limits its use in advanced research and industrial applications, particularly for bone X-rays, by producing fragmented pictures that may cover fractures and struggle with complicated edges.

Despite slight brightness problems, Sobel Edge Detection is effective for most edge types, providing crisper lines and broad applicability. Custom neural network Sobel Type2 outperforms Type1 but may still have brightness problems.

Although it is widely used and produces clear outlines, Canny Edge Detection may produce too many or misdraw lines.

Due to its sharp, clear lines, Sobel Edge Detection—specifically Type 2—remains dependable for medical imaging activities, notably X-ray analysis. This makes it perfect for fracture diagnosis in spite of brightness variations. Although Canny Edge Detection provides comparable advantages, its use in medical imaging tasks may be limited by problems with excess lines and misdrawn edges. All things considered, SobelEdgeDetectionType2 meets the necessary criteria and provides a strong basis for additional study and model training in medical imaging applications.

4.2 MODELS

As we can see, in case of the Morph dataset, YOLO v8 performs best, and in case of Canny Edge Detected dataset VGG19 performs best. The short dataset size—just 1056 pictures for the Canny dataset and 1036 for the Morphological Gradient Detected Dataset—is the primary cause of overfitting in X-ray fracture detection models. This small dataset size makes it difficult to train complicated models effectively, and larger datasets with more than 6,000

⁸Visit our site at [here](#) or [here](#).

images are required for best results. Furthermore, the intrinsic variability found in X-ray images—such as variations in fracture features and bone structure—adds still another level of complexity to the process of generalising models. Because they had fewer characteristics than other edge detection techniques, models trained on the Canny dataset performed better overall. In order to assess model generalisation, the research highlights the significance of accuracy, loss graphs, and confusion matrices. It also recommends investigating techniques like as data augmentation and regularisation in addition to acquiring larger and more varied datasets in order to improve the models' robustness.

4.3 STREAMLIT-BASED APP

Due to resource constraints, the app has been deployed via 2 separate links to accommodate twice the amount of users than allowed in 1 link.

ACKNOWLEDGMENTS

We sincerely thank Prof. (Dr.) Asima Adak for her consistent leadership throughout our study, "Optimising X-Ray Images for Enhanced Fracture Recognition." and Prof. (Dr.) Anindya Sen for his insightful ideas and suggestions. Special thanks to Saumodip Das, Soham Basu, Soumalya Bose, and the helpful juniors. We also thank the director of Techno India DAMA Healthcare & Medical Centre for his cooperation and for granting dataset access. This research gave us a platform to improve our knowledge of medicine, neural networks, and image processing. We are grateful for the department of Electronics and Communication Engineering's assistance. The experience enhanced our education and gave us the chance to investigate different approaches to enhance fracture detection in medical research.

REFERENCES

- [1] Bose S., Basu S., Bera I., Mallick S., Paul S., Das S., Sil S., Ghosh S., Sen A. (2023). A Comparative Study of Multiple Deep Learning Algorithms for Efficient Localization of Bone Joints in the Upper Limbs of Human Body. In *Computational Vision and Bio-Inspired Computing* (pp.637-658). doi: 10.1007/978-981-19-9819-5_46.
- [2] Wei Z., Na M., Huisheng S., Hongqi F. (2009). Feature Extraction of X-ray Fracture Image and Fracture Classification. *International Conference on Artificial Intelligence and Computational Intelligence*. Shanghai, China. doi: 10.1109/AICI.2009.40
- [3] Kurniawan S. F., Putra I. K. G. D., Sudana A.A. K. O. (10th June 2014). Bone Fracture Detection Using OpenCV. *Journal of Theoretical and Applied Information Technology* Vol. 64 No.1.
- [4] Tripathi A. M., Upadhyay A., Rajput A. S., Singh A. P., Kumar B. (2017). Automatic Detection of fracture in femur bones using Image Processing. *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. doi:10.1109/ICIIECS.2017.8275843
- [5] Swathika B., Kamalakannan A., Baskaran B., Rajamanickam G. (2015). Radius Bone Fracture Detection Using Morphological Gradient Based Image Segmentation Technique. *International Journal of Computer Science and Information Technologies*. ISSN: 0975-9676 (Online). 6. 1616-1619.
- [6] Sahin M. E. (January 2023). Image processing and machine learning-based bone fracture detection and classification using X-ray images. *International Journal of Imaging Systems and Technology* 33(3). doi:10.1002/ima.22849

- [7] Simonyan K., Zisserman A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv, ICLR 2015. Available: [here](#)
- [8] Ju R.-Y., Cai W. (2023). Fracture Detection in Pediatric Wrist Trauma X-ray Images Using YOLOv8 Algorithm. arXiv. Available: [here](#)
- [9] Abedeen I., Rahman Md. A., Protyasha F. Z., Ahmed T., Chowdhury M. T., Shatabda S. (2023). FracAtlas: A Data-set for Fracture Classification, Localization and Segmentation of Musculoskeletal Radiographs. Figshare Dataset Available: [here](#)
- [10] (2024). Streamlit documentation. streamlit.io, [Online]. Available: [here](#)
- [11] Dey N., Zhang Y.-D., Rajinikanth V., Pugalenth R., Raja N. S. M. (2021). CustomizedVGG19 Architecture for Pneumonia Detection in Chest X-Rays. Pattern Recognition Letters, Volume 143, Pages 67-74. Available: [here](#)
- [12] El-Saadawy H., Tantawi M. M., Shedeed H. A., Tolba M. F. (2021). A hybrid Two-Stage GNG Modified VGG method for bone X-Rays classification and abnormality detection. IEEE Access. PP. 1-1. Available: [here](#)
- [13] (13 May, 2024). Confusion Matrix in Machine Learning. www.geeksforgeeks.org, [Online]. Available: [here](#)
- [14] Yang A. Y., Cheng L., Shimaponda-Nawa M., Zhu H. -Y. (2019). Long-Bone Fracture Detection using Artificial Neural Networks based on Line Features of Xray Images. IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, pp. 2595-2602. doi:10.1109/SSCI44817.2019.9002664
- [15] Jabbar J., Hussain M., Malik H., Gani A., Khan A. H., Shiraz M. (2022). Deep Learning Based Classification of Wrist Cracks from X-ray Imaging. Computers, Materials & Continua, 73(1), 1827-1844. doi:10.32604/cmc.2022.024965