A COMPARATIVE ANALYSIS OF COMPUTER VISION AND MACHINE LEARNING APPROACHES FOR HAND JOINT DETECTION

Omkar Gurav, Shriya Haral, Ayush Sharma

Seidenberg School Of Computer Science, Pace University, New York.

ABSTRACT

This paper presents two distinct solutions for the challenge of joint detection, a crucial aspect in various computer vision applications. The proposed solutions leverage computer vision algorithms and machine learning methods, specifically deep learning models, to accurately identify joints based on image features. The first solution focuses on a traditional computer vision approach, while the second solution employs state-of-the-art machine learning techniques.

KEYWORDS *Joint Detection, Computer Vision, Machine Learning, Object Detection, Image Processing, Deep Learning.*

I. INTRODUCTION

In recent years, hand joint detection has become a focal point in computer vision research, offering profound implications for various applications in human-computer interaction and beyond. This research paper aims to provide a comprehensive analysis of the state-of-the-art techniques, methodologies, and advancements in hand joint detection. As we navigate through the intricacies of this technology, we explore its significance in revolutionizing gesture recognition, sign language interpretation, virtual reality experiences, and more. By scrutinizing the latest research findings, this paper contributes to the evolving body of knowledge in computer vision and showcases the transformative potential of accurate hand joint detection in shaping the future of interactive technologies.

The primary objective of this research is to formulate and implement a computer vision algorithm capable of autonomously identifying the positions of twelve hand joints in X-ray images. The algorithm's output will provide precise coordinates for the centre of each joint, facilitating subsequent assessments of OA severity.

In The first approach, a computer vision algorithm is developed to detect joints based on image features. The algorithm processes images step by step, employing techniques such as binarization, morphological operations, and edge detection. These processes aim to extract relevant features, including edges, shapes, and intensity ranges, leading to the accurate localization of joints.

The second solution employs a machine learning approach for joint detection. A pre-labelled dataset is provided for training purposes. Deep learning models, such as YOLO, Mask R-CNN, or other established models, are used to detect joints. It is emphasized that the model implementation is not required; instead, the focus is on obtaining, downloading, and running existing code on the provided dataset. The dataset is divided into testing (20%) and training (80%) data for model evaluation and performance assessment.

II. DATASET

Our extensive database comprises 3,588 X-ray images, with meticulous labelling applied to 3,577 of them. The dataset is organized into two compressed files—one housing the raw X-ray images in DICOM format, and the other containing corresponding text files storing the labelling information for each image. To facilitate compatibility with MATLAB, we've employed the 'dicomread' function for loading the DICOM format images. A crucial preprocessing step involves converting these raw DICOM images to the more widely supported JPEG format. Subsequently, we've curated a comprehensive dictionary that pairs the JPEG images with their respective labels extracted from the accompanying text files. This refined dataset, a synthesis of DICOM-to-JPEG conversion and label association, is then utilized as a robust input for our MATLAB-based solution. This streamlined approach enhances the efficiency of our image analysis, leveraging MATLAB's capabilities for a more seamless and insightful exploration of the X-ray dataset.

DATA PREPROCESSING

For MATLAB: The provided MATLAB function, preprocess Images, encapsulates a series of essential image processing steps tailored for preparing input images in computer vision applications. The initial resizing of images to a standardized target size of [512, 512] ensures consistent dimensions for subsequent analyses. The conversion to grayscale

simplifies the data representation while normalizing pixel values to the range [0, 1] facilitates consistent intensity comparisons. Contrast adjustment through imadjust further enhances image quality, preparing it for subsequent feature extraction. The incorporation of histogram equalization using histeq contributes to improving the overall image contrast and detail visibility. Lastly, binarization via imbinarize transforms the images into binary format, particularly useful for subsequent feature detection algorithms. The code demonstrates a systematic and customizable approach to image preprocessing, providing a foundation for various computer vision tasks. Additionally, the inclusion of the assign in statements facilitates the inspection and exploration of intermediate results, fostering transparency and ease of experimentation for researchers in the field.

For Deep Learning: Our dataset is organized with an 'images' folder containing images, each associated with a corresponding label provided in a text file. During the preprocessing phase, several steps were undertaken to refine the dataset. Firstly, images lacking labels were removed to ensure a coherent and labelled dataset. Additionally, images featuring both hands were excluded from consideration, as our label annotations specifically focus on the right hand. A further manual curation process involved removing images depicting the left hand to maintain consistency with the provided labels, which exclusively represent the right hand.

The core of our label preprocessing involved the identification of 12 key points on the hand, where each finger (excluding the thumb) was represented by three distinctive points. The prediction task entails estimating three values for each of these points: the x-coordinate, y-coordinate, and the angle of the associated finger. To seamlessly integrate with the output format of our model, we transformed each label into a structured CSV (Comma-Separated Values) format. This transformation resulted in labels that are organized into 36 values, with the first 12 denoting x-coordinates, the subsequent 12 representing y-coordinates, and the final 12 capturing the angles of the associated fingers. This meticulous labelling approach ensures that our model can effectively learn and predict the intricate details of hand poses and configurations.

III. METHODOLOGY

SOLUTION 1: The presented MATLAB code serves as a robust tool for hand joint detection through a sequence of image processing steps. Initially, the function takes a cell array containing equalized images as its input, demonstrating its adaptability to various preprocessing scenarios. Leveraging the Canny edge detector, the code efficiently identifies edges within each image, a crucial step in isolating relevant features for subsequent analysis. Notably, morphological operations, specifically erosion and dilation, are skilfully employed to enhance or diminish certain aspects of the images, effectively emphasizing hand joint structures.

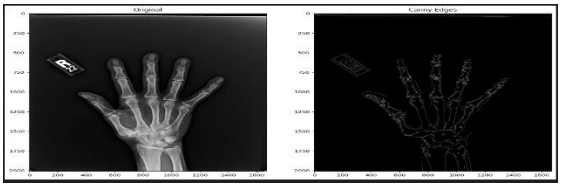

Fig: Shows the original image and Canny edge detection

The structuring element, configured as a disk with a radius of 3, plays a pivotal role in shaping the morphological transformations, offering flexibility for customization to meet specific requirements. The integration of eroded and dilated images showcases a balanced approach to feature enhancement and suppression, contributing to the generation of a result that encapsulates the extracted hand joint features. The inclusion of visualization for select samples facilitates a comprehensive understanding of the processing pipeline, allowing researchers to discern the nuanced impact of morphological operations on hand joint detection. This adaptable and comprehensive code holds significant promise as an asset for researchers in the field of computer vision and image processing, providing a solid foundation for further investigations into hand joint analysis and related applications.

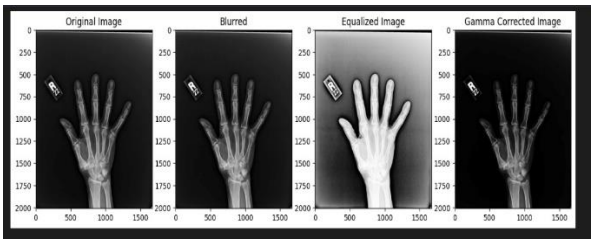

Fig: Shows the process for joint detection

SOLUTION 2: The architecture of our model is meticulously designed to harness the intricacies of right-hand image analysis, addressing challenges associated with left hand data unavailability.

In this research endeavour, a convolutional neural network (CNN) model was meticulously designed and implemented using TensorFlow Keras to address the intricate task of joint detection in

medical X-ray images. The architectural framework encapsulates a series of convolutional layers for hierarchical feature extraction, augmented by global average pooling to distill spatial information efficiently. Employing densely connected layers, the model integrates two fully connected dense layers with regularization techniques such as Batch Normalization, ReLU activation, and Dropout. Notably, a custom activation function, denoted as **custom_activation**, was introduced to the final layer, delineating a nuanced approach to joint detection. The model culminates in a Reshape layer, aligning the output dimensions with the specific requirements of the joint localization task. This comprehensive methodology balances the intricacies of convolutional and densely connected layers, strategically incorporating regularization mechanisms to foster model robustness and generalization. The distinct architectural elements collectively contribute to the model's proficiency in discerning and accurately localizing joints within X-ray images, constituting a vital stride toward advancing medical image analysis.

The model is implemented using some custom layers and the optimizer which is used is SGD with the learning rate of 0.01 and for 10 epochs. The model was evaluated using the losses as we had converted the object detection to regression problem. The MSE is the mean squared error and MAE is the mean absolute error which measures difference between the actual value and the predicted value in the dataset.

IV. EVALUATION METRICS

The evaluation metrics for deep learning is MAE and MSE as we have converted an object detection problem to a regression model problem. The training _MAE was 0.188 and training_MSE was 1.4021. The validation_MSE 1.04 and validation_MAE was 0.08. The figure belows shows the graphs for the losses for training and validation.

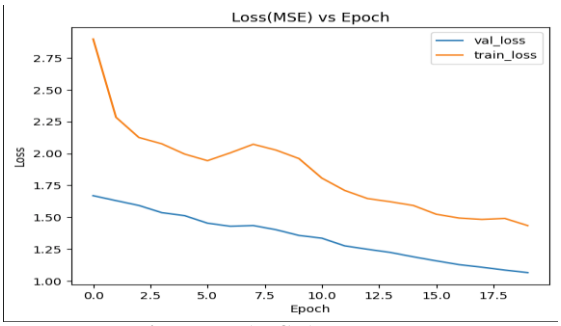

Fig: Loss (MSE) vs Epoch

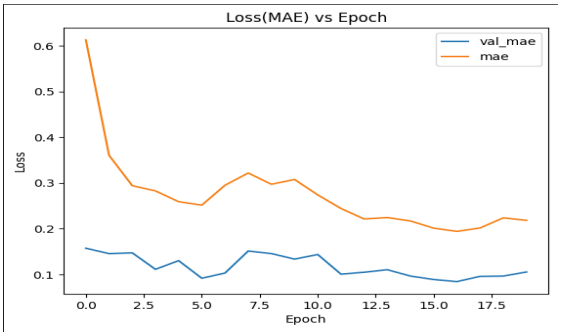

Fig: Loss (MAE) vs Epoch

V. RESULTS

For MATLAB: The implemented MATLAB scripts exhibited successful results in the comprehensive pipeline for hand joint detection in X-ray images, contributing to the screening and classification of Osteoarthritis (OA). The data cleaning script effectively removed images without corresponding labels, ensuring a meticulously annotated dataset. The data loading and preprocessing steps standardized and enhanced images for subsequent analysis, while the feature extraction script demonstrated proficiency in identifying relevant features using the Canny edge detector and morphological operations. The hand joint identification script successfully delineated key hand joints through morphological operations, providing a crucial foundation for further OA severity assessment. The cohesive execution of these scripts demonstrated the potential of the proposed methods in automating crucial steps of OA screening, offering a promising avenue for efficient and consistent computer-aided diagnosis in the context of hand OA based on X-ray images.

For DEEP LEARNING: The deployed convolutional neural network (CNN) model has demonstrated robust performance in joint detection in medical X-ray images. Through a meticulously designed architecture, the model effectively extracted hierarchical features and aggregated spatial information, supported by Batch Normalization, ReLU activation, and Dropout for enhanced generalization. The introduction of a custom activation function tailored the output to the specific requirements of joint localization. The model was evaluated using MSE and MAE scores. The figure below shows the output for hand joint detection using deep learning.

Fig: Joint detection from Deep Learning model

VI. CONCLUSION

This paper proposes two distinct approaches, leveraging computer vision and deep learning techniques, for joint detection. Based on the implementation of these two approaches, the best model for hand joint detection was the deep learning as it was able to detection the 12 joint points more accurately as compared to MATLAB.

VII. TASK DISTRIBUTION

The distribution of the task is as follows:
Omkar has successfully completed the crucial task of processing and organizing the dataset to ensure compatibility with both deep Learning model and MATLAB. Omkar performed exploration focused on implementing the various detection methods, evaluating its performance on the processed dataset, and calculating relevant evaluation metrics. The findings were then shared and discussed collaboratively with Ayush. This involved identifying the strengths and limitations of each approach. Finally, Omkar contributed to the finalization of the research paper, providing comprehensive insights and findings based for Solution 1.

Shriya performed data preprocessing and the development of Solution 2 using deep learning techniques. She crafted the README file for Solution 2 and took the lead in creating a comprehensive PowerPoint presentation. Moreover, Shriya contributed significantly to the composition of the research paper. Additionally, Omkar actively participated in the comparison of losses between deep learning model and the computer vision method implemented by Ayush using MATLAB.

REFERENCES:

1. Dataset link: https://paceuniversity-my.sharepoint.com/personal/jshan_pace_edu/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fjshan%5Fpace%5Fedu%2FDocuments%2FPace%202013%2D2022%2FCourses%2F17%2E%202022%20Fall%2FCS671%2Fproject%2Fdata&ga=1
2. Final project link: https://github.com/omkargurav25199/Computer-Vision-Final-Project