Hand Joint Detection from X - Ray Images using Computer Vision and Deep Learning Approach

Anusha Guruprasad

Department of Computer Science
Pace University New York, USA
ag84104n@pace.edu

Mihika Mishra

Department of Computer Science Pace University New York, USA
mihika.mishra@pace.edu

Darsh Joshi

Department of Computer Science
Pace University
New York, USA
dj97498n@pace.edu

Abstract

Osteoarthritis (OA), the most prevalent form of arthritis in humans, primarily affects the knee, hip, and hand joints. This paper focuses on the development of an automated system for the detection and classification of hand OA. The current standard for diagnosing hand OA involves manual inspection by radiologists, who mark and measure specific joints in X-ray images of the hand. This process, particularly in identifying the twelve key joints in the fingers, is labor-intensive and time-consuming. Our proposed system aims to automate this procedure by identifying the Metacarpophalangeal (MCP), Proximal Interphalangeal (PIP), and Distal Interphalangeal (DIP) joints, excluding the thumb, as these are crucial for hand OA diagnosis. The system aligns bounding boxes centered on these joints, following the natural angle of each finger, to facilitate further analysis.

Keywords: Hand OA, Automated System, Metacarpophalangeal (MCP), Proximal Interphalangeal (PIP), Distal Interphalangeal (DIP), Thumb, Deep Learning Method, Data Preprocessing, Computer Vision Method

Anusha Guruprasad : Deep Learning Method
Mihika Mishra: Data Preprocessing
Darsh Joshi: Computer Vision Method

Introduction

Osteoarthritis (OA) is a degenerative joint disease that represents a significant health concern due to its prevalence and the pain and disability it causes. In the realm of hand OA, early detection and accurate classification are vital for effective management and treatment. Traditionally, the evaluation of hand OA has been reliant on radiologists performing manual assessments of X-ray images, a process that is not only time-consuming but also prone to human error.

The primary objective of our research is to introduce and validate an automated, computer-aided method for the identification and classification of hand OA. This paper presents a novel approach that focuses on the automatic detection of twelve critical joints in the fingers: the Metacarpophalangeal (MCP), Proximal Interphalangeal (PIP), and Distal Interphalangeal (DIP) joints, with the exclusion of the thumb. The methodology involves utilizing centered bounding boxes, aligned with the direction of each finger, to accurately identify these joints on X-ray images.

This automated approach aims to revolutionize the screening process for hand OA. By leveraging advanced image processing techniques and machine learning algorithms, our system is designed to facilitate the rapid and precise assessment of OA severity. Metrics such as the Kellgren and Lawrence (KL) grade and Joint Space Width (JSW) are used to evaluate the degree of joint degradation, offering a more objective and consistent basis for diagnosis compared to manual methods.

The subsequent sections of this paper will detail the methodology employed in developing this automated system, the validation process, and a comparison of its performance against traditional manual techniques. Through this research, we aim to contribute significantly to the field of rheumatology and radiology, offering a more efficient, accurate, and accessible tool for the diagnosis and classification of hand OA.

I. Data

The dataset central to our research comprises a comprehensive collection of 3588 X-ray images, specifically curated for the study of hand Osteoarthritis (OA). Of these images, 3577 have undergone detailed labeling to facilitate the development and validation of our automated hand joint identification system.

Data Availability

The dataset is segregated into two distinct downloadable zipped files:

Raw X-ray Images: This file contains the entire set of X-ray images in DICOM (Digital Imaging and Communications in Medicine) format. DICOM is a widely used standard for storing and transmitting medical imaging information. These images form the basis of our automated analysis and classification of hand OA.

Labeling Information Files: Accompanying the raw images are text (TXT) files, each corresponding to an individual X-ray image. These files contain critical labeling information, delineating the precise location and characteristics of the twelve key joints in the fingers, which are vital for the diagnosis of hand OA. The labeling format and details will be comprehensively explained during the course.

Loading and Viewing Images

The DICOM format of the X-ray images necessitates specific software or programming functions for proper viewing and analysis. In our research, we utilized Matlab, a high-level programming language and interactive environment widely used in scientific computing. The function dicomread in Matlab is employed to load and visualize these DICOM images effectively.

Data Preprocessing

In the data preprocessing phase of our project, team implemented a MATLAB script to efficiently prepare our X-ray image dataset for analysis. The primary goals were to convert DICOM files to the more accessible JPEG format and systematically organize the corresponding labeling information.

DICOM to JPEG Conversion: The first script, convertDicomInFoldersStartingWith9, targets specific folders (starting with '9') in the source directory to locate DICOM files, particularly in the 'v06' subdirectories. It then uses convertDicomToJpgInFolder and convertDicomToJpg functions to convert these files into JPEG format. This step is essential for making the images compatible with a wide range of image processing and machine learning tools.

Label File Management: The enhanced version of the script adds label file management. After converting DICOM files to JPEG, it moves and renames the associated label files to align with the new JPEG files using the renameLabelFile function. This alignment is crucial for integrating the images with their diagnostic data for automated analysis.

These preprocessing steps are vital in ensuring the dataset's compatibility with diverse analytical tools and maintaining an organized, efficient workflow for processing our large-scale medical image dataset.

II. Algorithms

1. Computer Vision Method

Here, we report a complete computer vision technique that can reliably identify finger joints in X-ray pictures. With potential applications in orthopedics, biomechanics, and medical diagnostics, our approach aims to precisely detect and assess the location of finger joints inside X-ray scans by utilizing modern image processing and machine learning techniques. X-ray image acquisition is the first step in our process. Next, user-guided region of interest (ROI) selection is performed, with a focus on the joints of interest. To improve image quality and emphasize joint areas, we employ a series of image processing techniques, such as gamma correction, CLAHE (Contrast Limited Adaptive Histogram Equalization), Gaussian blur, and Canny edge detection. The contours are then located, and the centers are computed. We use KMeans clustering to identify the joint centers. Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) metrics are used to evaluate the accuracy of our joint detection, and comparison analysis may be done with the help of the given visualizations. Our approach provides a reliable and efficient way to locate finger joints precisely in X-ray pictures, which may have consequences for medical diagnosis and therapy planning.

1.1. Methodology

1. Region of Interest (ROI) Selection

We allow the user to direct the selection of regions of interest (ROIs) within the X-ray images in order to concentrate the analysis on the pertinent anatomical features. Users can manually identify the regions that contain the finger joints in this stage. ROI selection is a crucial component of our methodology since it reduces computational overhead and improves accuracy by focusing processing on the regions of key interest.

2. Image Processing Techniques

a. Gaussian Blur

To minimize image noise and fine details while keeping structural characteristics, Gaussian blur is used. To achieve the required smoothing effect, a Gaussian kernel with a size of (9, 9) is convolved with the ROI.

b. Gamma Correction

Gamma correction is used to modify the image's luminance. This step is critical for attaining the best possible image brightness and contrast. To rectify the image, a lookup table is built that maps pixel values to modified gamma values.

c. CLAHE (Contrast Limited Adaptive Histogram Equalization)

CLAHE is used to improve the contrast of X-ray pictures, making minor anatomical details like joints more visible. CLAHE's adaptive nature ensures that local contrast is improved, especially in regions of varied intensity.

d. Canny Edge Detection

Canny edge detection is used to detect edges and boundaries in processed images. This approach is essential for drawing attention to the outlines of the finger joints. Before applying the Canny edge detector, adjusted images are normalized to an 8-bit unsigned integer format, allowing for exact edge recognition.

3. Contour Analysis and Joint Detection

Following image processing, contour analysis is used to detect the contours of the finger joints inside the processed ROIs. Following the identification of contours, their centers are estimated using typical picture moment analysis techniques. These contours' centers are thought to be possible joint centers.

KMeans clustering is applied to the estimated centers to further refine the detection of joint centers. The number of clusters is set to 12, which corresponds to the number of finger joints expected. As a result, a collection of cluster centers representing the most likely locations of the finger joints in the X-ray images is produced.

fig 2.1 Resultant image with detected visual points

1.2. Evaluation

1. Accuracy Matrics

Quantitative indicators are used to objectively assess the accuracy of our finger joint detection approach. We compare our methodology's projected joint centers to a collection of reference coordinates indicating ground truth values. There are two primary accuracy metrics used:

Mean Squared Error (MSE): The Mean Squared Error (MSE) is the average squared difference between the projected joint centers and their reference coordinates. MSE measures the overall difference between the identified joints and the ground truth, with lower values indicating more accuracy.

Root Mean Squared Error (RMSE): The MSE is used to calculate the Root Mean Squared Error (RMSE), which is the square root of the average squared deviations between predicted and reference coordinates. RMSE is a more interpretable measure of accuracy that takes into consideration the size of errors. A smaller RMSE indicates that the anticipated and true joint centers are more aligned.

2. Visualization

We provide a visual representation of the projected joint centers alongside their closest matches from the reference set, in addition to quantitative measures. This graphic facilitates in the qualitative evaluation of our method's accuracy. Scatter plots are constructed to show the predicted coordinates (in blue) and their reference coordinates (in red, indicated with a 'x'). This comparative image enables a rapid and straightforward comprehension of the alignment of expected and true joint locations.

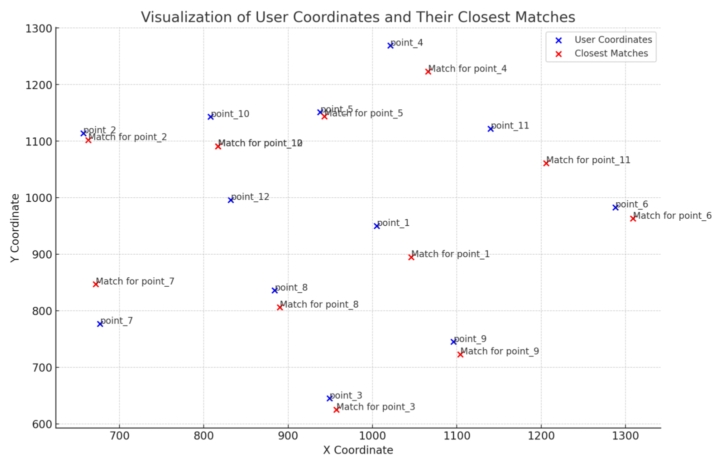

fig 2.2 visual representation of actual and predicted point coordinates

2. Deep Learning Method

Detectron, developed by Facebook AI Research (FAIR), has emerged as a pivotal open-source software system designed explicitly for object detection tasks within image data. Evolving from its predecessor, Detectron2 represents a significant overhaul, leveraging the robustness of PyTorch to offer a more adaptable and modular framework for various computer vision applications, prominently object detection, instance segmentation, and keypoint detection.

The framework's notable advancement lies in its adaptability, allowing users to implement state-of-the-art object detection algorithms efficiently. Detectron2's architecture fosters a flexible environment, providing researchers and practitioners with an array of cutting-edge model architectures, including Faster R-CNN, Mask R-CNN, RetinaNet, among others. These models, rooted in deep learning, predominantly harness convolutional neural networks (CNNs) as their backbone, enabling effective feature extraction and object classification.
Detectron2's ascendancy in the computer vision research landscape can be attributed to its accessibility, versatility, and impressive performance capabilities. Its ability to handle various computer vision tasks has found applications across diverse domains, ranging from autonomous vehicles to medical imaging, surveillance systems, and beyond. This paper delves into the intricate workings of Detectron models, elucidating their fundamental architecture, training methodologies, and their applicability in addressing complex object detection challenges.

2.1. Methodology

The algorithm aims to train a custom object detection model using the Detectron2 framework, focusing on the detection of X-ray hand key points within images.

Data Preparation:

- Annotation classes for hand keypoints are defined, such as "mcp2," "pip2," "dip2," and others.
- A custom dataset loader is implemented to read image files (.jpg) and corresponding label files (.txt) with hand keypoints information.

Dataset Registration and Visualization:

- The custom dataset is registered using Detectron2's DatasetCatalog and MetadataCatalog.
- A visualization function is created to display image samples with annotated hand keypoints for debugging purposes.

Model Configuration and Training:

- The algorithm configures the Detectron2 model by loading a predefined YAML configuration file (Base-RCNN-FPN.yaml) and setting training parameters.
- The training process involves setting up the model, defining datasets for training and testing, initializing weights, and configuring data loaders.
- The model is trained using the DefaultTrainer from the Detectron2 engine, which iterates through the dataset and optimizes the model to detect hand keypoints.
- Metrics such as accuracy, loss values, and training progress are monitored and stored in the metrics.json file for performance evaluation.

Conclusion:

- Upon completion, the algorithm successfully trains a custom object detection model for hand keypoints using the Detectron2 framework, enabling accurate detection and localization of specific key points within images.
- This description outlines the steps involved in training a custom object detection model for hand keypoints, incorporating data loading, model configuration, training, and performance monitoring.

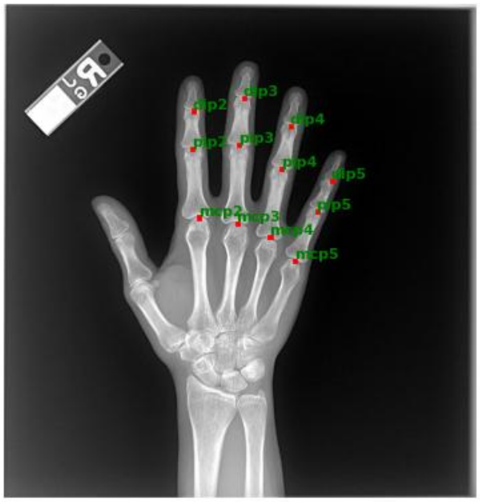

Fig 2.1.1 visualization of the x-ray hand on test data

2.2. Metrics

During the training iterations, the code computes and stores various metrics, including accuracy, loss values (such as classification and bounding box regression losses), data loading time, and learning rate, in the metrics.json file.

This file acts as a log, recording key performance indicators of the model's training progress over successive iterations.

These records are instrumental in analyzing and assessing the model's performance trends and behavior, aiding in iterative model improvement and optimization for object detection tasks.

Metrics	Average Value
Classification Accuracy	96.4%
Loss for Box Regression	0.00025
Total Loss	0.7
Total time taken for the model	26 mins 47 seconds
Number of epochs	500
Batch size	10

Table 2.2.1 Metrics

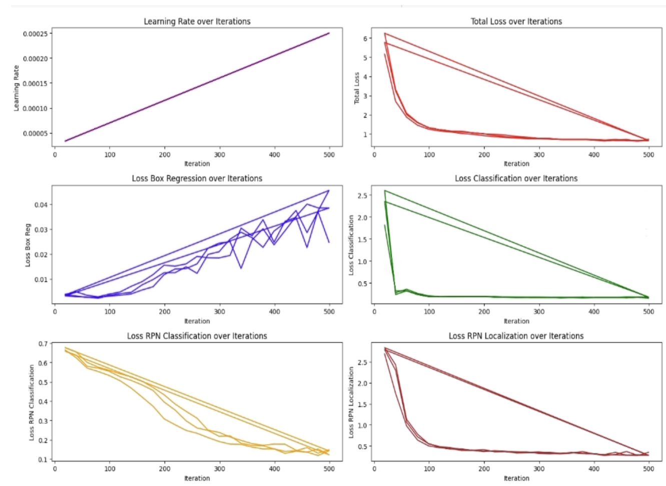

Figure 2.2.1 Visual representation of metric analysis

The visual depiction presented herein illustrates the comprehensive metric analysis conducted, showcasing the fluctuations observed in the box plot loss, regression loss, and accuracy metrics across successive iterations.

III. Conclusion

In conclusion, our comparative analysis between Computer Vision (CV) and Deep Learning (DL) methodologies for keypoint localization in images revealed significant performance differences. In regard to the CV method, our experimentation revealed that by strategically defining Regions of Interest (ROIs) and applying computer vision techniques exclusively within these localized areas, we observed notable improvements in keypoint localization accuracy and label derivation. Even with this, the DL approach demonstrated far better results in plotting keypoints and deriving associated labels compared to the CV method.

IV. References

Peteya, Jennifer Anita. "Resolving details of the nonbiomineralized anatomy of trilobites using computed tomographic imaging techniques." PhD diss., The Ohio State University, 2013.

Mery, Domingo, and Carlos Arteta. "Automatic defect recognition in x-ray testing using computer vision." In 2017 IEEE winter conference on applications of computer vision (WACV), pp. 1026-1035. IEEE, 2017.

Shamir, Lior, Shari M. Ling, William Scott, Marc Hochberg, Luigi Ferrucci, and Ilya G. Goldberg. "Early detection of radiographic knee osteoarthritis using computer-aided analysis." Osteoarthritis and Cartilage 17, no. 10 (2009): 1307-1312.

Üreten, Kemal, and Hadi Hakan Maraş. "Automated classification of rheumatoid arthritis, osteoarthritis, and normal hand radiographs with deep learning methods." Journal of Digital Imaging 35, no. 2 (2022): 193-199.

Cai, Zhaowei, and Nuno Vasconcelos. "Cascade R-CNN: High quality object detection and instance segmentation." IEEE transactions on pattern analysis and machine intelligence 43, no. 5 (2019): 1483-1498.

Chen, Peng, and Qingwei Shen. "Research on Table Tennis Swing Recognition Based on Lightweight OpenPose." In 2023 16th International Conference on Advanced Computer Theory and Engineering (ICACTE), pp. 207-212. IEEE, 2023.