Diagnosis of Bone Fissure Fracture detection using OpenCV Tool

Manpreet Kaur
mkaur2967@gmail.com
Ramgarhia Institute of Engineering and Technology, Phagwara, Punjab

Bharti
bnahar9@yahoo.co.in
Ramgarhia Institute of Engineering and Technology, Phagwara, Punjab

Dr. Naveen Dhillon
principal@riet.ac.in
Ramgarhia Institute of Engineering and Technology, Phagwara, Punjab

ABSTRACT

New technologies are emerging daily in different fields, most employed technologies are in medical environment. However, still there are many old techniques being quite popular, very efficient and also very effective in this manner. X-rays are one of these techniques that are used for detection of bone fractures and injuries. But sometimes the size of fractures is not significant and could not be detected easily or when size of fracture is too small. Therefore, effective and intelligent systems should be designed to overcome these problems. This paper aims to develop the classification system that would be capable of detecting and classifying the bone fractures. The developed system comprises of different principal stages. The images of the fractures are processed using different image processing techniques using OpenCV in order to detect their location and shapes.

Keywords— Image Processing, Bone fracture detection, Medical imaging, Computer vision

1. INTRODUCTION

Image processing is a method to perform operations on an image. It is used to extract some useful information from an image. It is a type of signal processing in which input is an image and output may be image or characteristics/features or another useful data associated with that image. Today, image processing is among rapidly growing new technologies. It forms core research area among engineering and computer science disciplines too.

OpenCV is one of the most popular computer vision techniques. It is an open-source library that includes several hundreds or even thousands of computer vision algorithms. It is available for C++ and python platform. In our research we use python because it is available for every platform.

We developed system is designed to detect bone fractures or bone injuries. This can be accomplished by utilizing diverse image processing methods and techniques to find fractured regions. Thus, each supplied images are processed, and then a fracture should be detected on the original image, using one of the algorithm. The proposed system uses bone fracture images obtained from internet.

2. RELATED WORKS

2.1 Color Space Conversion

There are few common color models, specifically RGB, HSV, and YUV. A color model is an abstract mathematical model that describes how colors can be represented as a set of numbers which are further represent bits. In this technique convert them to grayscale. Grayscale is a range of gray shades from white to black, as used in a monochrome display or black and white printout [1]. Grayscale images are most commonly used in image processing because it contain data which is many times smaller than other models. Smaller data enables developers to do more complex operations in a shorter length of time.

2.1.1 Average method: The Average method takes the average value of R, G, and B or Red, Green and blue as the grayscale value.

\[ \text{Grayscale} = \frac{R + G + B}{3}. \]

Theoretically, the formula is 100% correct. But when writing code, you may encounter uint8 overflow error — the sum of R, G, and B is greater than 255 so output is more than 8 bits or a byte. To avoid the exception, R, G, and B should be calculated respectively before this step [2].

2.1.2 The Weighted Method: The weighted method, also called luminosity method, weighs red, green and blue according to their wavelength or spectrum. The improved formula is as follows:

\[ \text{Grayscale} = 0.299R + 0.587G + 0.114B \]

It is most accurate method

2.2 Thresholding

Thresholding is used to separate out regions of an image corresponding to objects which we want to analyze. This separation is based on the variation of intensity between the object pixels and the background pixels. To differentiate the
pixels, we are interested in from the rest (which will eventually be rejected), we perform a comparison of each pixel intensity value with respect to a threshold (determined according to the problem to solve)[3]. Once we have separated properly the important pixels, we can set them with a determined value to identify them (i.e. we can assign them a value of 0 (black), 255 (white) or any value that suits your needs). Different values are chosen for different applications.

3. RESULTS AND DISCUSSIONS

Results are obtained from sample images. Results are obtained on Intel i9-9900K octa-core at 3.5GHz. It shows result in 20 to 100 milli-seconds. Through this application, the robustness, flexibility and speed of this intelligent bone fracture detection system have been demonstrated.

![Canny Image](image1)

Results and time taken

This algorithm, the identification of the images is yielded 91.4% where, 19 images out of the available 20 bone images correctly yielded.

4. CONCLUSIONS

In this paper, we develop an algorithm image processing and neural network approaches for the detection of bone fractures. Image processing techniques are very useful and significant in the medical field such as in X-ray radiographs and MRI.

The motivation of this work was to compare and evaluate the obtained results and also, effectiveness of a back propagation neural network with a suggested algorithm in recognizing different bone fractures. Comparing the identification ratios as discussed in the results. The developed system is a robust system since it was trained to recognize bone fractures. The images used for training and testing the proposed system were collected from the benchmark database, which contains 100 images. The images contain different fractures in size and illumination conditions for each subject, which was an advantage for us to make the system more efficient and robust. Finally, it can be stated that the experimental analysis of the proposed bone fracture detection system showed a great efficiency and an outperforming rate.

5. REFERENCES


