



DIGITAL IMAGE PROCESSING AND RECOGNITION USING PYTHON

D. Sri Shreya

Department of Electronics and Communication
Vellore institute of Technology, Chennai, Tamil Nadu, India

Abstract— In this project, the primary aim will be the conversion of images into Grayscale in which conversion of pixels to array takes place and apply Blur effect using The Gaussian blur which is a type of image-blurring filter that uses a Gaussian function which also expresses the normal distribution in statistics for calculating the transformation to apply to each pixel in the image. The above two processes are applied to the input images.

These two above mentioned processes can be achieved by utilizing the most relevant python libraries and functions, followed by conversion of the digital image to numerical data and then, applying the effects to the image to get back the image with applied effects in it.

Face recognition refers to matching a face present in an input image from the training/pre-saved dataset and by applying Deep Learning Concept. This will be achieved by defining a function to read and convert images to data, apply the python function, and then, recreating the image with results.

Keywords— Gray Scale, Blur effect, Gaussian blur, Python

I. INTRODUCTION

Digital image processing is the method of a digital computer to process digital images through the use of a relevant algorithm. It is a subcategory or field of digital signal processing, digital image processing contains a lot of advantages over analog image processing but digital image processing allows the use of much more complex algorithms, and hence, can offer both more sophisticated performance at simple tasks, and the implementation of methods which would be impossible by analog means. It provides a much wider range of algorithms that can be applied to the input data and can avoid problems such as the build-up of noise and distortion during processing. Since images are defined over two dimensions or more than just 2-dimensions digital image processing may be modeled in the form of multidimensional systems. The generation and development of digital image processing are mainly affected by three factors: the development of computers, the development of mathematics especially the creation and improvement of discrete mathematics theory, and lastly the demand for a wider range of applications in environmental, agricultural, military, industry and medical science increasing over-time.

Digital image processing deals with the manipulation of digital images through a digital computer. It is also a subfield of signals and systems but it majorly focuses on images. The input of this system processing is a digital image and the system process that image using efficient algorithms, and gives an image as an output with its corresponding description.

These processed images can now either be directly used or we can use them further for developing higher-order models used in Artificial Intelligence and Deep Learning. Image Recognition is one such concept that will be applied to differentiate and recognize faces in digital images that can be taken use of in various technical fields like user entry security systems, face unlock system, blind people support, etc. Moreover, this can be used in digital camera images, digital cameras generally include specialized digital image processing hardware – either dedicated chips or added circuitry on other chips – to convert the raw data from their image sensor into a color-corrected image in a standard image file format. A film like “Westworld (1973)” was the first feature film to use digital image processing to pixelate photography to simulate an android's point of view.

The proposed model uses conversion images to grayscale and apply face recognition to lessen the time complexity of face detection and recognition. Hence, it detects faces faster and with an accuracy of more than 90%.

1) THE GAUSSIAN BLUR

The Gaussian blur is a type of image-blurring filter that uses a Gaussian function (which also expresses the normal distribution in statistics) for calculating the transformation to apply to each pixel in the image. The formula of a Gaussian function in one dimension is

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

In two dimensions, it is the product of two such Gaussian functions, one in each dimension:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.

When applied in two dimensions, this formula produces a surface whose contours are concentric circles with a Gaussian distribution from the center point. Values from this distribution are used to build a convolution matrix that is applied to the original image.

Open CV provides faster in-built libraries to compute these

equations and apply them to the image pixels.

$$Y' = 0.299R' + 0.587G' + 0.114B'$$

2) GRAYSCALE

For grayscale images, the result is a two-dimensional array with the number of rows and columns equal to the number of pixel rows and columns in the image. Low numeric values indicate darker shades and higher values of lighter shades. The range of pixel values is often 0 to 255. We divide by 255 to get a range of 0 to 1.

Color images are represented as three-dimensional NumPy arrays - a collection of three two-dimensional arrays, one each for red, green, and blue channels. Each one, like grayscale arrays, has one value per pixel and their ranges are identical.

$$Y_{\text{linear}} = 0.2126R_{\text{linear}} + 0.7152G_{\text{linear}} + 0.0722B_{\text{linear}} \quad [5]$$

$$C_{\text{linear}} = \begin{cases} \frac{C_{\text{srgb}}}{12.92}, & C_{\text{srgb}} \leq 0.04045 \\ \left(\frac{C_{\text{srgb}} + 0.055}{1.055}\right)^{2.4}, & C_{\text{srgb}} > 0.04045 \end{cases}$$

To undo the effects of gamma compression before calculating the grayscale luminance, it's necessary to apply the inverse operation, gamma expansion

II. PROPOSED ALGORITHM

READING IMAGE:

Input image

Read image using cv2

Convert image to NumPy array

Each array element represents the pixel and its color in RED, BLUE, and GREEN

GRAY SCALE:

$$\text{Gray} = 0.299R + 0.587G + 0.114B$$

(OR)

$$\text{Gray} = (\text{Red} + \text{Green} + \text{Blue}) / 3$$

For each pixel:

Red = Pixel.Red

Green = Pixel.Green

Blue = Pixel.Blue

$$\text{Gray} = (\text{Red} + \text{Green} + \text{Blue}) / 3$$

Pixel.Red = Gray

Pixel.Green = Gray

Pixel.Blue = Gray

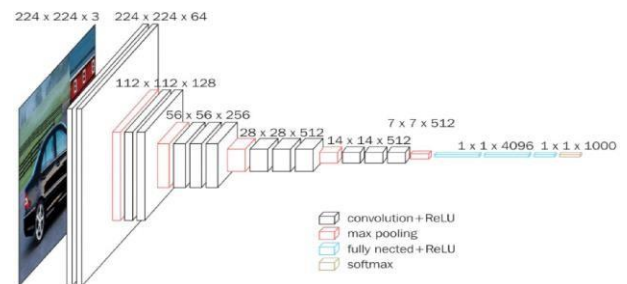
Recognition Using CNN and Face Recognition:

Construct Sequential Convolution Neural Network Layer for extracting key features from the pixels and learn the characteristics for recognition. Use feature extractor and apply max pooling to each pixel group to determine the position of key feature extractor. Use Keras weight as well as 64 layered ImageNet CNN for training the images. This trained model will give accuracy between 75% to 96% depending on the data set size used for training. Alternatively use a face recognition library to obtain the same neural network result with 80% accuracy and almost 1/5th of the time compared to the standard CNN model.

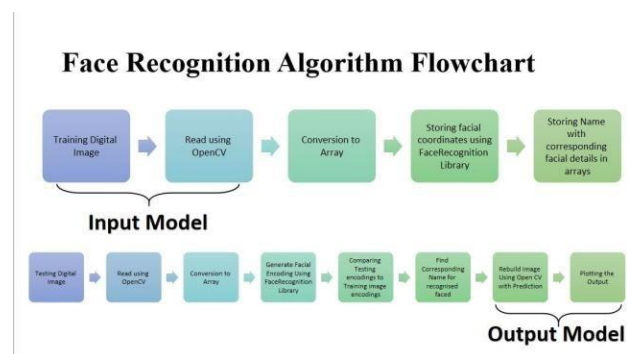
Image Reconstruction:

Use cv2 to rebuild the image with all RGB pixels with Grayscale values. Find the region of the image with the face and its corresponding name and accuracy. Print the image and change pixels near the face with a box to plot the name prediction obtained. If the prediction value doesn't cross the threshold, it doesn't contain any recognizable face.

Deep Neural Network



FLOWCHART

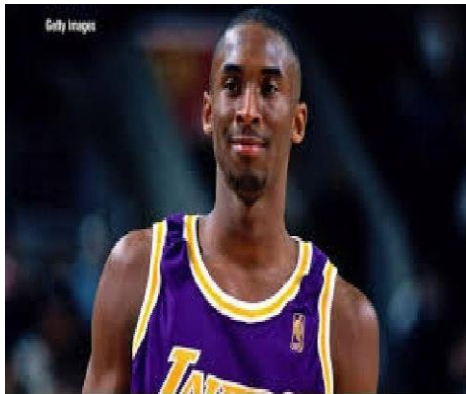


III. EXPERIMENT AND RESULT

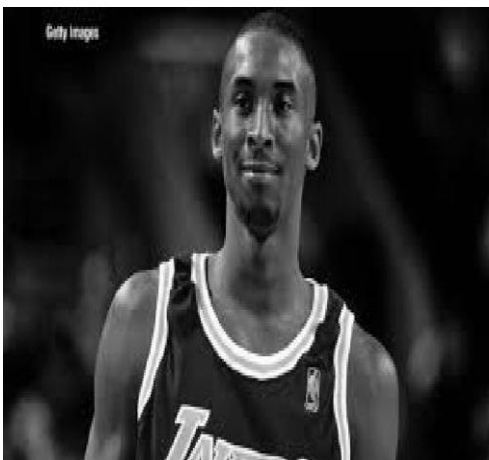
Input 1

The input image has the face of Kobe Bryant.

The model is trained to identify Kobe Bryant

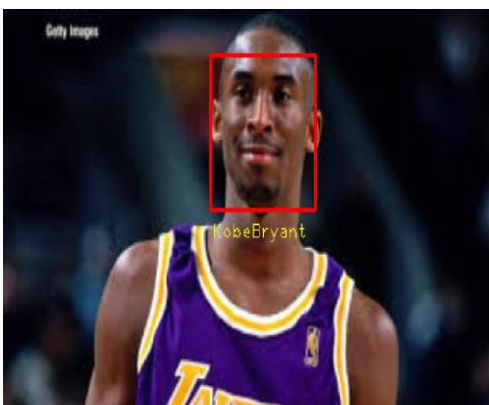


(a) Original image

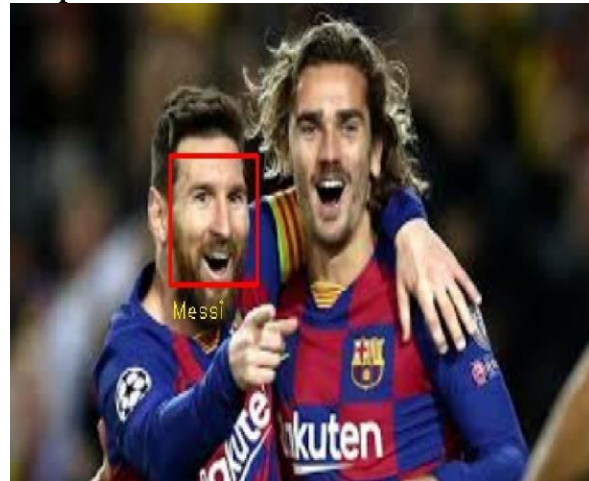


(b) Gray scale image

Output: Final recognition



Output:



IV. CONCLUSION

Input 2:
The input image has the faces of Messi and Griezmann.
The model was trained to recognize Messi and not identify Griezmann.

Hence, the Image Processing and Recognition model was implemented by the deep learning concept and implemented. It successfully converts the images into grayscale and then uses it for detection. By converting the image into grayscale, we reduce the magnitude of pixel values, and then images are used for face recognition. This process made the face recognition faster as compared to the traditional model and detects faces



with high accuracy, as evident from the output. The accuracy achieved was 80% using libraries and up to 97% using the CNN model from Deep Learning. This can be used in the same way to detect faces as used in modern security systems, face unlocks monitoring systems, etc.

V. REFERENCE

1. C. Chang and P. J. Chou. Taipei, Taiwan - Face Detection in Real Time Based on HOG. N. J. Wang, S.: IEEE, DOI:10.1109/ISPACS.2012.6473506, 2012. International Symposium on Intelligent Signal Processing and Communications Systems. pp. 333- 337. (ISBN: 978-1-4673- 5081-5)
2. S.V.Viraktamath, Mukund Katti, Aditya Khatawkar, Pavan Kulkarni. 3- Face Detection and Tracking using OpenCV., s.l.: SIJ, July-August 2013, The Standard International Journals (The SIJ) , (Vol. 1, pp. 45-50. ISSN: 2321 – 2403)
3. <https://www.irjet.net/archives/V7/i10/IRJET-V7I10219.pdf>