

DRUG INFORMATION EXTRACTION USING IMAGE PROCESSING

¹G.Lavanya, ²K.V.S.S.Srinija, ³J.Paavani, ⁴Alim Sofia, ⁵D.Indu

¹ student, ²student, ³student, ⁴student, ⁵Assistant Professor

Department of Computer Science and Engineering,
Gayatri Vidya Parishad College of Engineering for Women, Visakhapatnam, India.

Abstract- People find many types of drugs for medication but not all of them are safe for usage. It should be checked if the drug they are using does not cause any side effects and provides speedy recovery. The information about the drug should be clear and accurate so that there are no side effects after usage. In order to provide accurate information, all the data should be collected. This issue can be solved using an application that reads the information about these drugs and states whether it is safe to consume that drug or know if it causes any side effects. This project uses image processing and what kind of medication it is just by capturing an image and extracts the text on the image and reviewing the information about that drug. This method also helps decide a proper medication for the illiterate.

Keywords: Medication, Image processing.

1. INTRODUCTION:

Taking the right medication with fewer side-effects is very important for patient health. If we take the wrong medication instead of the right medication it will affect that patient. In this app, we will give a clear description of medication uses, side-effects and ratings. Drug information extraction is an image processing technique. This technique helps to know about medication side effects and uses. Capture an image from the camera or Upload an image from the gallery and then give input to the Mediinfo application. The Internal process contains 3 steps. The first one is pre-processing of an image and the second one is Extraction and the final one is matching. Pre-processing removes noise from an image. And then find counters from the pre-processed image. Draw a Bounding rectangle based on counters. Extract text from the cropped image and compare the medication name with the dataset. If the match found it will display results in the results screen otherwise it will display error.

App building: There are different methods to build an app. Android studio is one of the ways to build an app. We used HTML and PHP for checking whether they are giving correct input format or not and for submitting the selecting image. For backend, python is used. After creating layouts in-app and connecting java, PHP, and HTML files. we build apk using BUILD APK in android studio[5].

2. PROPOSED ALGORITHM

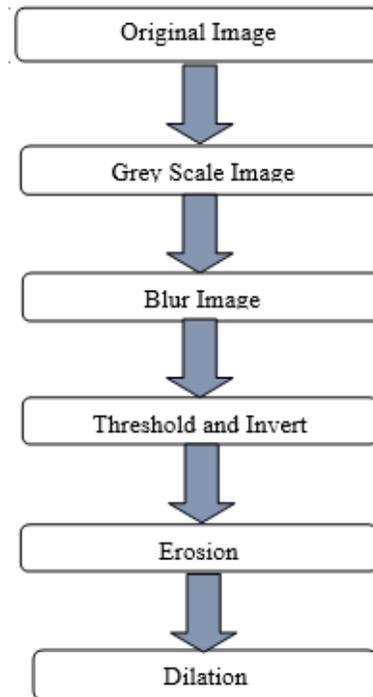
Basically to complete the task of displaying output to the user about medication can be done using 3 steps.

2.1. Preprocessing

Preprocessing of an image removes noise from the given image and results are also accurate. Preprocessing technique done in 5 steps. Those are gray scaling, blurring, Thresholding, Erosion, and dilation[1]. In every step, the image will become less noisy. You should do above 5 steps in an order.

Preprocessing of an Image:

There are mainly five steps involved in preprocessing of an image, they are explained below



First, take an image from either camera or gallery. Then convert that original image to grayscale image. The grayscale image consists of only shades of gray. This process leaves only the luminance of an image and removes all colors in an image. Take the output image of gray scaling then apply the blurring technique to it. Blurring means smoothing of an image. Blurring is nothing but applying a Low pass filter to each and every pixel in an image. Low pass filter is a filter that allows low-frequency pixels and it neglects the high-frequency pixels. After applying a low pass filter filtered image will be obtained as output. Then apply Thresholding to the filtered image. Thresholding is a method of Binarization. Binarization means segmenting an image into 2 colored images[3]. Those colors are black and white. The binarized image will be obtained as output. Apply morphological technique to the binarized image to structure an image. There are 2 morphological operations present. The first one is erosion and the second one is Dilation. Erosion removes unwanted pixels in an image. Dilation adds pixels that will fill holes in an image. Till now pre-processing is over.

The final output of the pre-processing image contains only black and white pixels. And then find counters from the pre-processed image. Draw a Bounding rectangle based on counters. Take the second-highest bounded rectangle and crop that image.

2.2. Extraction

The second step is extraction, exacting text from an image using OCR (optical character recognition) technique[2]. It is one of the best techniques to extract text from an image.



Extract text from cropped image using pytesseract module. Pytesseract is a module that helps to convert image format to string format. That string contains some special symbols also. You have to remove that special symbol from a string. After removing special characters from the string one new string will be obtained as an output. Output stored in a text file.

2.3. Matching

The third step is matching, Matches the content in the dataset with newly obtained string. If the match is found then the output will store in the previous text file. The later text file content will display in the application. The text file contains uses, rating and side effects of medication.

3. APP EXPLANATION

In this project, every picture represents an activity. Each page contains different pictures. In this explanation, each page is explained along with screenshots. The usage and interaction with the app will be explained here.

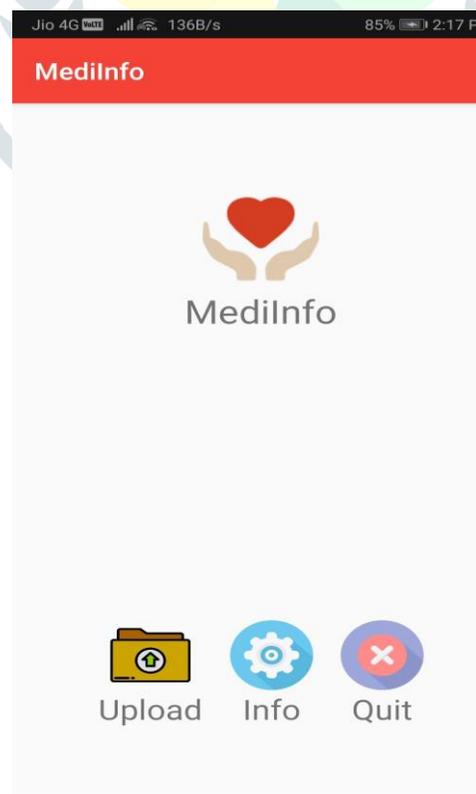


Fig 3.1: Front page of app

The above figure represents the front page of an app. On this page each button will have different functionality. The Upload button is used to load a medication image through either camera or files in the phone. The info button contains useful information about app. The quit button is used to exit from the app.

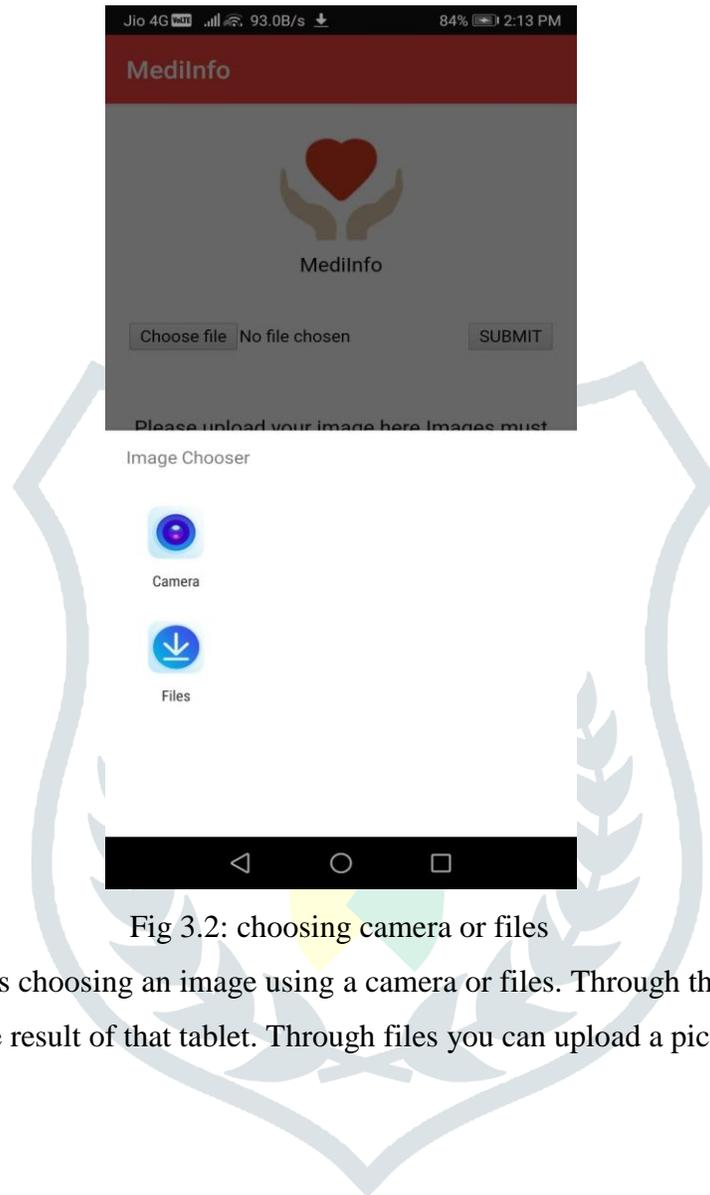


Fig 3.2: choosing camera or files

Above the figure represents choosing an image using a camera or files. Through the camera, you can capture a picture and you can see the result of that tablet. Through files you can upload a picture to see the results.



Fig 3.3: camera

The above figure represents camera of an app. By clicking on the tick mark present in the camera you can capture that picture. After That you can view the results in results page.

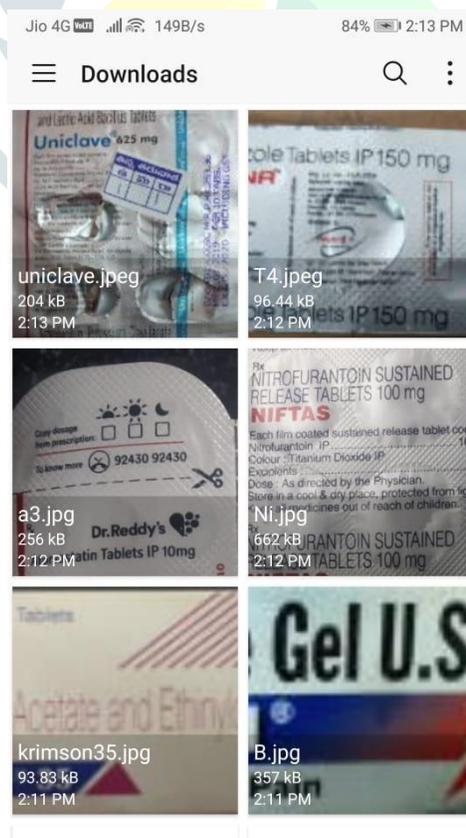


Fig 3.4: uploading an image

The above figure represents the uploading of an image. To upload an image you have to choose any image first. For that, you have to click an image. It will load an image from the gallery.



Fig 3.5: uploaded an image from gallery

Above figure represents the upload page of app. Here you can see image name after uploading completes. By clicking on submit button you can view results. If you uploaded other than jpg, jpeg, png then it will show an error message to user.

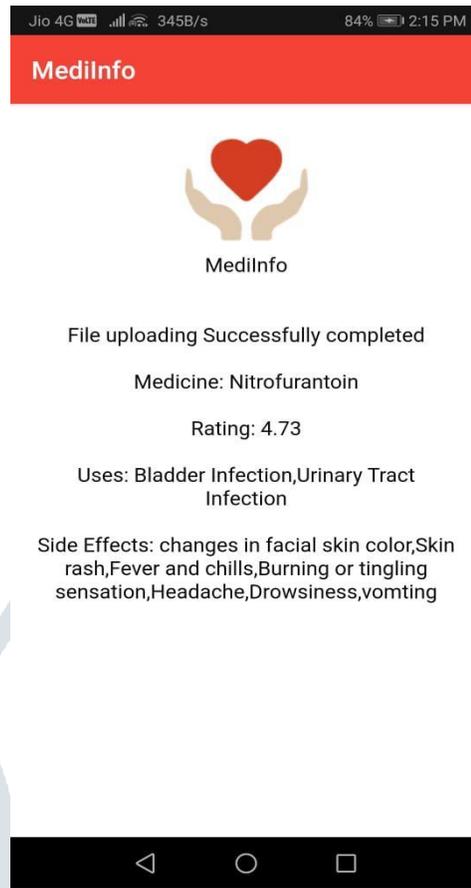
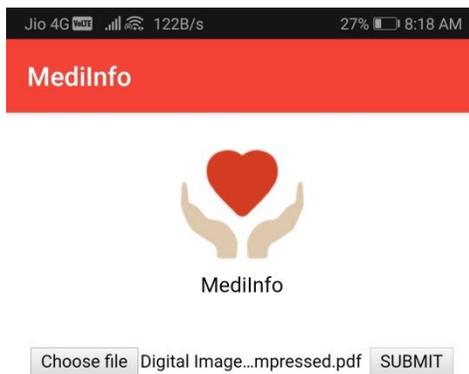


Fig 3.6: Results page

The above figure represents results of an captured or uploaded image. You can see the output on the results page. Results page contains medication details like medication name, rating, uses and side effects.

4. EXPERIMENTAL RESULTS AND ANALYSIS:

Analysis part consists of testing various files. Fig 4.1 shows uploading a pdf as input to Mediinfo app. App only take images as input but here we upload a pdf .



Please upload your image here. Images must be of png, jpg or of jpeg format. It may take a few minutes based on your file size and internet connection!!

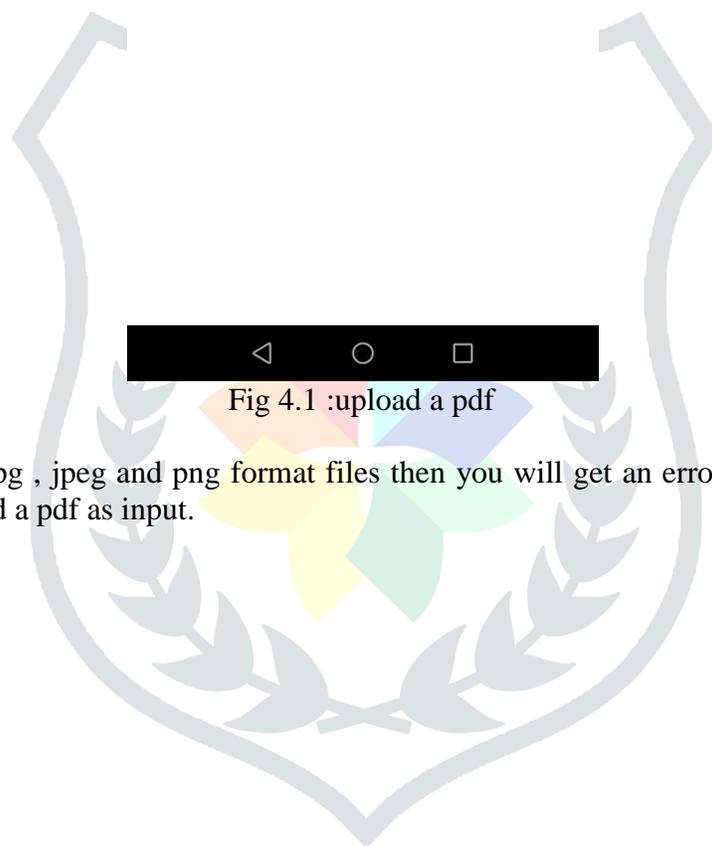


Fig 4.1 :upload a pdf

If you upload other than jpg , jpeg and png format files then you will get an error message.fig 4.2 shows an error message while upload a pdf as input.

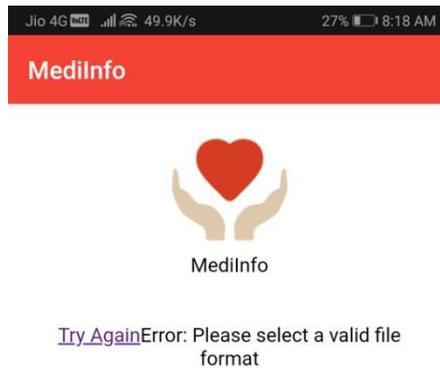


Fig 4.2 : error message

5. CONCLUSION:

In this paper, we develop an app to provide medical help to the people who are illiterate and wants to have knowledge about the medication they are prescribed by their doctors. This app is user friendly and easy to use.

6. ACKNOWLEDGMENT:

The project team members would like to express our thanks to our guide D. Indu Assistant professor of Computer Science and Engineering Department, GVPCEW for her valuable suggestions and guidance in completing out project model.

7. REFERENCES:

- [1]Image edge detection based on OpenCV; Guobo Xie, Wen lu ,\emph{International Journal of Electronics and Electrical Engineering Vol. 1, No. 2, June 2013}
- [2] Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study; Chirag Patel, Atul Patel,Dharmendra Patel , \emph{International Journal of Computer Applications (0975 – 8887) Volume 55– No.10, October 2012}
- [3] Jufeng Yang, Kai Wang, Jiaofeng Li, Jiao Jiao, Jing Xu, "A fast adaptive binarization method for complex scene images", 19th IEEE International Conference on Image Processing (ICIP), 2012.
- [4] Bradski, Gary; Kaehler, Adrian (2008). Learning OpenCV: Computer vision with the OpenCV library. O'ReillyMedia,Inc.p.6.
- [5] <https://developer.android.com/>
- [6] Rafael C. Gonzalez and Richard E. Woods. Digital Image Processing, 3rd edition, Pearson publication.
- [7] <https://craftofcoding.wordpress.com/2017/02/13/image-binarization-1-introduction/>