GSoC 2016 Proposal
Blood Sample Image Analysis

Abdul Fatir Ansari

Department of Civil Engineering, IIT Roorkee, India, PIN 247667
abdulfatirs@gmail.com,abdulfatir@outlook.com
http://abdulfatir.com

Interests: Computer Vision, Image Processing, Machine Learning, Cryptography, Bioinformatics

Abstract. The aim of this project is to use the camera and processing power of modern day cell phones to develop an intuitive and user-friendly application for the detection of cancer biomarkers from a small drop of blood. This will serve as a screening test for the same. The application will allow the user to take images of the blood samples in a set format. The image will then be segmented using a background subtraction algorithm to detect the regions of interest. After noise removal, the intensity of each individual blob will be calculated. A linear curve will be fit through the intensity and known concentration data and the concentrations of the unknown samples will be estimated from the standard curve which will quantify the various molecules present in the sample.

Keywords: Image Analysis, Segmentation, Cancer, Biomarker, Android, Java, OpenCV

1 Background

I'm currently in junior year of civil engineering at *Indian Institute of Technology, Roorkee*. I've been involved in programming for 7 years now, of which the last 2 have been in the fields of Computer Vision & Machine Learning. I'm an ardent follower of the advancements in the field of Artificial Intelligence and am always eager to contribute to the field. Image Processing & Computer Vision form a very important part of AI Research and hence this project interests me.

1.1 Image Processing & Computer Vision

I've worked on various projects involving computer vision and image processing some of which include the following.

1. **Posture Recognition in HINE Exercises** – A multi-class classification task which involved segmentation of infant's body from videos of Hammersmith Infant Neurological Examinations, post-processing the segmented

videos, skeletonization, and then feature extraction to train a Hidden Markov Model for broad level exercise classification.
Languages: C++ (OpenCV), Python, MATLAB
Publication: A. F. Ansari, P. P. Roy, and D. P. Dogra. Posture recognition in HINE exercises. *Proceedings of International Conference on Computer Vision and Image Processing*, 2016 (in press).
Current Work: Finer level sequence labelling using Bag of Words approach and training Recurrent Neural Networks (LSTM) for the problem of classification.

2. **Vehicle Detection & Tracking** – A vehicle detection and tracking task for estimation of traffic flow parameters on an unsignalized intersection.
Languages: C++ (OpenCV)

1.2 Android

I've been working on Android platform for 3 years now. The following two applications among others showcase my work in this field.

1. **Sanibridge Health App** (*ongoing*) – An Android health application for monitoring various biomarkers. I worked on the full stack development of the application including the backend under the supervision of Dr. Tomas Helikar.
Languages: Java (Android), PHP+MySQL
2. **Dodgy Bat** – An obstacle avoidance game for Android platform using LibGDX.
Languages: Java (Android)

2 Programming Skills

- Java, Python, C++, MATLAB are the languages I'm most experienced with. I'm also well acquainted with web development both backend (PHP, Python, NodeJS), DBMS (MySQL), and frontend (HTML+CSS+JS and their various frameworks).
- I am also familiar with C#, Perl and LaTeX.
- I've been working on Open Source projects since 2010. Some of my projects are also on GitHub.
- I want to learn and build new image processing and computer vision algorithms, and learn about the applications of machine learning in computer vision this summer.

3 Biology

I'd studied biology at high school level. I am interested in learning biology this summer because I've been working in the field of medical computer vision of late and the knowledge of biology related to the work I am doing will be a plus.

4 Details of Proposal Challenge

I used the following algorithm for the proposal challenge in Python & C++.

4.1 Methodology

1. Convert the image into HSV color space and threshold the image based on the presence of red hue. (Circular Hough Transform after finding edges (Canny) of the image can also be used for blob detection)
2. Remove noise and false detections using 5x5 median filter and morphological operations.
3. Detect the contours of the segmented image and keep the largest ones (which represent the samples) and discard small blobs (if present) as spurious detections.
4. For each blob, calculate the average pixel intensity value after converting the image into grayscale.
5. Input the concentration of known blobs and fit a one-degree curve through the data (since the Standard Curve is linear).
6. Find the values of concentrations of unknown sample and quality control samples from the plot.

4.2 Results

The results of calculations on PC and Android platform are given in tables 1 and 2 respectively.

Sample	Actual Value	Calculated Value	Error
QC1	156 ng/ml	149.82 ng/ml	3.9 %
QC2	750 ng/ml	843.15 ng/ml	12.4 %
Unknown Sample	NA	273.41 ng/ml	NA

Table 1. Results on PC

$$\mathbf{D = 0.05081\ C + 75.55}$$

D = Darkness (255-Brightness)
C = Concentration in ng/ml

4.3 Android Prototype

I also developed an Android Prototype (code on GitHub repository and fully functional code and binaries here) for the above algorithm. The video of the

Sample	Actual Value	Calculated Value	Error
QC1	156 ng/ml	151.82 ng/ml	2.68 %
QC2	750 ng/ml	847.86 ng/ml	13.05 %
Unknown Sample	NA	270.80 ng/ml	NA

Table 2. Results on Android Device

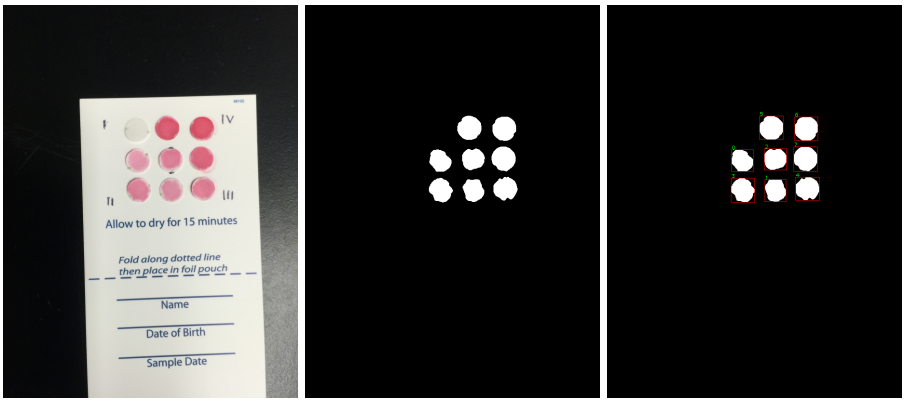

Fig. 1. Input → Segmentation & Noise Removal → Blob Detection

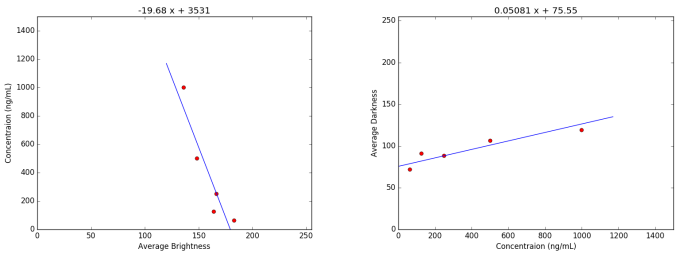

Fig. 2. Raw Curve & Standard Curve

workflow can be found here. The special feature of this prototype application is that I've made it interactive *i.e.* one can tap on the detected blobs and edit the properties of that very sample.

Usage

1. Tap Load Image and look for the image in the gallery.
2. Tap Analyze and wait. The app will look for 8 largest red hue blobs and will outline them with red color and label them with a number.
3. Tap on each rectangle to input values for that blob.
4. An input dialog will open which will ask for Sample Type and Concentration.
   - For Known samples, choose Known and enter concentration.
   - For Quality Control samples, choose QualityControl and enter concentration.
   - For Unknown sample, choose Unknown and leave concentration as it is.
5. Tap Results and a dialog box with results and equation will be shown.

The screenshots of the Android application are shown in figures 3, 4 and 5.

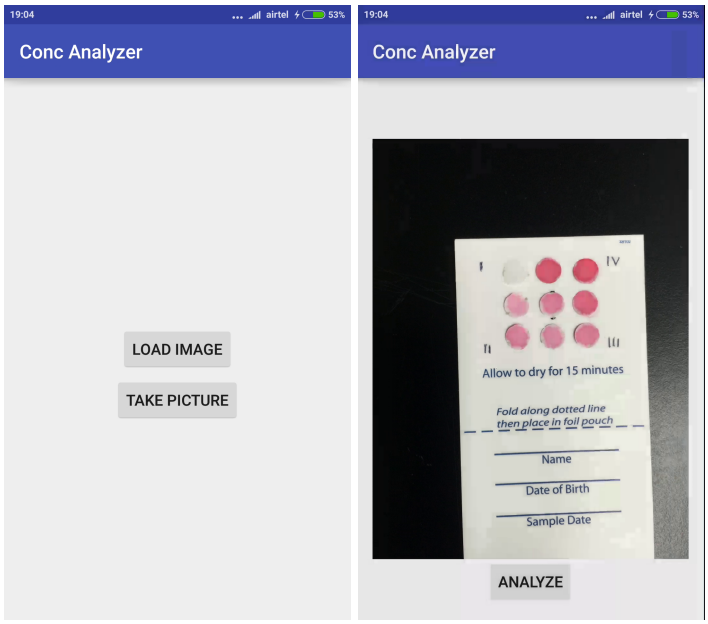

Fig. 3. Home Screen & Analyze Screen

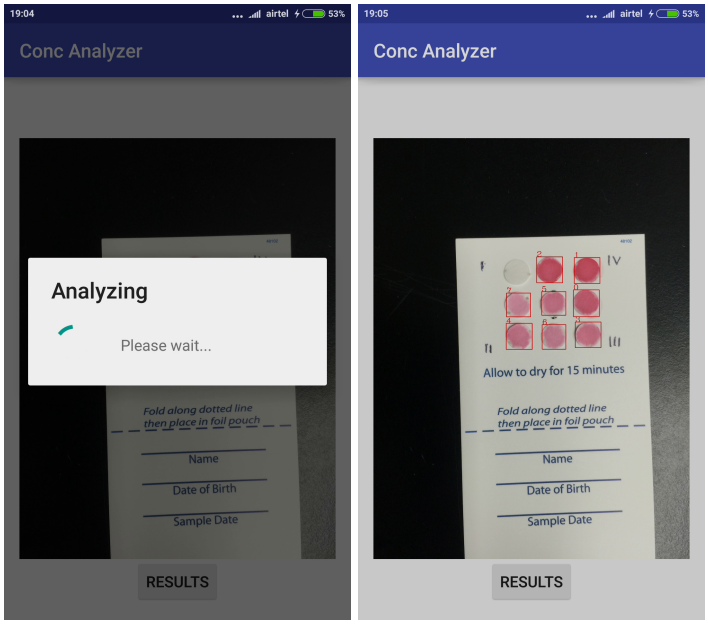

Fig. 4. Analyzing & Detected Blobs

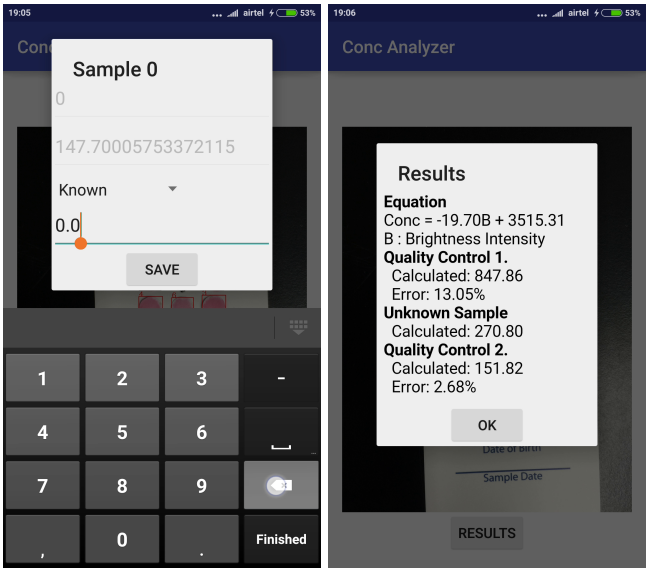

Fig. 5. Enter Sample Details & Results

5 Project Details

5.1 Mobile Based Blood Analysis (Required)

Skills: Java (Android), Image Processing

1. **Image Capture** – The image containing the blood samples will be captured using the camera package of Android SDK. A guideline image will be overlayed on the camera view to guide the user into taking good quality pictures.
2. **Segmentation** – The captured picture will then be segmented using any of the suitable methods (depending on the resultant accuracy of the computation) which include *i) Hue based segmentation*: The image will be converted to HSV color space and segmented based on color bounds of the Hue channel. (This can also be accomplished by converting the image in YC_rC_b color space and segmenting using C_r channel) and *ii) Circle detection based approach*: The edges of input image will be detected using suitable filters (Canny, Sobel etc.) and the circular Hough Transform will be applied on the edge detected image.
3. **Noise Reduction** – The image output from previous step will contain noise and spurious detections. This noise can be completely removed by filtering the image using a 5x5 median filter and then applying morphological operations.
4. **Detection** – The blobs which contain the sample can then be detected by finding the contours of the binary image from the previous step and then choosing the largest ones. Since the variation of size of the blobs will be bimodal, the regions of interest can easily be found out by Otsu like method even if the number of samples is unknown. If the number of samples is known, say N, the N largest blobs (area wise) will be chosen. The image processing tasks can either be done using OpenCV + JNI as I have done in my prototype or be implemented from scratch, whatever suits the project best. I understand the underlying algorithms perfectly and can efficiently implement them in Java.
5. **Measurement** – The intensities of the detected blobs will be calculated by averaging the intensities of pixels it contains. This can be done by converting the image into grayscale (as I've done in the proposal challenge) or YC_rC_b or HSV whichever gives the best results.
6. **Standard Curve Generation & Estimation** – A linear curve will be fit through the data using a suitable curve fitting method (least squared error linear regression etc.) and the equation relating the intensity to concentration will be found out. The standard curve will be displayed using suitable graphing API (as I've already done in the *Sanibridge* application) after suitable change of axes along with the equation. The values of unknown samples will be then evaluated from the standard curve and displayed accordingly.

The application will have two modes of processing, *a) Manual Processing* – where the user will tap on the blobs to add information about them, *b) Batch*

Processing – where the properties of blobs which occur at specific positions will be entered just once and any subsequent images will be processed automatically.

5.2 REST API & Web Based Blood Analysis (Optional)

Skills: PHP/Python, C++, HTML+CSS+JS

I believe the Android based project won't take much of the time, therefore, I am proposing the following additions to it.

- A REST API for performing the same image processing and standard curve estimation tasks online. Since analysis on mobile devices for large images might be a bit time consuming, user will be given a choice to analyze the image online instead (if connected to the internet). The algorithms for this will be rewritten for efficiency and a REST API will be created.
- A complete Web Application in HTML5 and CSS3 which will use the same backend API. User will be able to upload the image and analyze them interactively. The web application can have two modes namely *a) Automatic* – where user will upload the image and the analysis will be done automatically, and *b) Expert* – where user can interact with analysis procedure and choose different methods.

6 Timeline

- 30 April - 22 May: Familiarize myself with the project problem and prototype different algorithms in Python and C++ to analyze which one gives the best results.
- 23 May - 3 June: Code the best algorithm from the previous step in Android platform and make the first build of application with basic UI structure.
- 3 June - 13 June: Develop the Results section (Standard Curve and Equation) of the application.
- 13 June - 21 June: Make improvements to the user interface and make it intuitive and user friendly. Fix bugs.
- 21 June - 28 June: Mid Term Evaluation Period. Fix Bugs and make general improvements to the application and make the first release build.
- 28 June - 8 July: Develop server backend for allowing user to choose server analysis as an option.
- 8 July - 28 July: Develop the REST API for the backend server. Develop front end for web application in HTML5, CSS3 and JavaScript.
- 28 July - 8 August: Make changes to the Android application to incorporate server analysis option.
- 8 August - 16 August: Test the Android and Web applications rigorously to fix bugs, if any.
- 16 August - 24 August: Code submission.

7 Questionnaire

– Classes & exams end by 30th April 2016.
– I don't have any school related activities or internship during summer.
– I'll be able to work 40 hrs per week on my GSoC project.

8 Why?

Why saturation based measurement is best for the task at hand and still I chose not to use it?
When the image is converted into HSV color space and then segmented based on red hue, the next logical step is to measure the intensity by averaging the Saturation of each blob which represents how saturated the blob is with the color red. As the concentration increases, the reddishness i.e. saturation will increase and hence a curve of positive slope will be obtained which will serve as the standard curve. However, in the sample the average saturation value of either C3 or QC1 does not comply with this logic, but this may be due to camera or some other error. Table 3 shows the average saturation values of all samples. The ones which do not comply are in red. Curve fitting using this data produced results with one calculation breaching the error limit of 20% so I discarded this method.

Sample	Concentration	Avg. Saturation
C2	62.5 ng/ml	31.44
C3	125 ng/ml	38.6
QC1	156 ng/ml	33.9
C4	250 ng/ml	40.5
C5	500 ng/ml	49.5
QC2	750 ng/ml	55.11
C6	1000 ng/ml	56.59

Table 3. Saturation based measurement

Why is this project important for me?
I'll be going for further studies in the field of Computer Vision. Given my background (Civil Engineering), such a project will be a strong point in my profile and hence I'm passionate about it.

Why should this project be given to me?
I'm well-versed in the field of Image Analysis and understand the algorithms to be used perfectly. I'm a research oriented person who is always eager to learn new things, and the project itself aims to provide optimal image analysis results which requires research and trials using various different techniques.