RespiTrack
version 1.0

Respiratory patterns tracking and analysis
TOC

Overview
Understanding the problems
Project objective
Application flow

Project milestones
What’s next?
The team
Overview

A person’s breathing pattern abnormalities are serving a major role in detecting many human diseases like asthma, acute respiratory failure, heart failure and more.

The detection of such abnormalities is primarily done by human diagnostic.

As part of a research being done in GIP lab, Technion, we aim to automate the detection and analysis process using tools of image processing.

The project is implemented as an Android app.
Understanding the problems

01 **Invasiveness** - the examination of a patient’s breathing is done by a medical staff member and requires the patient to physically be present during the entire process.

02 **Human Error** - since the examination is done by humans it is prone to human error. It can either be misread of data or a bad decision by the medical staff.

03 **Time consuming** - The amount of patients to be examined in the current process is slow due to physical limitations and therefore collecting large data from patients is limited.
Project objective

RespiTrack objective is to provide automated recording of breathing patterns and presenting the breathing rate and patterns in an easy to use method for comparing with abnormal patterns which are typical to some diseases.

We aim to solve the problems described:

- No physical attendance required - simple use of an Android App.
- Reduce human error with advanced algorithms and method.
- No time wasted - send results any time.
Application flow

User Configuration: Customize stickers amount and color
Recognition: Filtering noises to detect stickers
Tracking: Recording samples taken from stickers positions through time
Results: Generate results graph
BPM: Calculate breaths per minute (BPM) and save results
To track movement of the chest during breathing we use same color stickers placed on points of interest on the body.

The homepage screen allows the user to choose the color of stickers as well as the number of stickers to perform the examination.
We use OpenCV methods to extract the stickers’ color specific range from an image taken by the phone’s camera. We then use the connected components algorithm to label the connected areas which suppose to represent the stickers. Finally, we filter out noise with our own algorithm that considers area size and shape to determine a sticker.
We use OpenCV methods to extract the stickers’ color specific range from an image taken by the phone’s camera.

We then use the connected components algorithm to label the connected areas which suppose to represent the stickers.

Finally, we filter out noise with our own algorithm that considers area size and shape to determine a sticker.
We use OpenCV methods to extract the stickers’ color specific range from an image taken by the phone’s camera.

We then use the connected components algorithm to label the connected areas which suppose to represent the stickers.

Finally, we filter out noise with our own algorithm that considers area size and shape to determine a sticker.
Following the method (developed in the GIP lab) we take the sum of the distances between each 2 points on each frame.

We plot the sum of every sample on a graph to provide the user a nice interface to see what the breathing looks like through time.

Normal breathing rate will be presented as a smooth sine-like wave.
Results

When the user hits the stop button, we take the recorded samples and present 2 graphs:

- Normalized sums graph
- Fourier graph

Normalized sums

We perform a normalization of the samples recorded to present how the chest expands and contracts during the breathing.

This graph varies between breathing rates, the cycles become shorter when breathing rapidly and longer when breathing slowly which is represented in the graph by the distance between peaks.
Results

When the user hits the stop button, we take the recorded samples and present 2 graphs:

- Normalized sums graph
- Fourier graph

Fourier graph

We use the Cooley-Tukey algorithm (radix-2 form) to perform Fast Fourier Transform on the normalized samples.

The FFT transfers the data collected from the time domain to the frequency domain.

A “normal” breathing has a single dominant magnitude as shown in the example.

The dominant frequency (x-axis) will be greater for faster breathing.
BPM

(Breaths Per Minute)

By multiplying the dominant frequency by 60 we get the BPM.

Reset to start over the process

Save a pdf file containing the data recorded (can be used for offline analysis)
Recap

By examining the resulting graphs and comparing them to the ones we receive from disease carrying persons we can advance in these diseases’ detection using this simple method.

Using the app is simple and require no invasive procedures or special tools.
Project milestones

Environment
Setup Android Studio IDE and OpenCV image processing library

Research
Understanding the purpose and designing workflow to suit the methodology and solutions

Stickers tracking
Using OpenCV algorithms to detect specific colors and shapes from a camera frame

Data and UI
Collecting and presenting data in a fast way with minimum loss and crafting easy to use and understand UI
What’s next?

- Recording the real-time analysis frames.
- Offline analysis (from a file).
- Machine learning tools to replace the need of stickers.
- Collect big data from the app of sick and not sick users to extend the research.
The Team

Students

Guy Berger  
guyber1992@gmail.com  Linkedin

May Schwartz  
mayschwartz1@gmail.com  Linkedin

Supervisors

Alon Zvirin

Yaron Honen