

HOCHSCHULE FLENSBURG

MOBILE COMPUTING

## Alarm-App Dokumentation

*Adrian Kurth, Jannes Peters, Tjark Klüttermann und Jan Peter Lamp*

January 5, 2018

## 1 Gruppenmitglieder

- Adrian Kurth 590289
- Jannes Peters 590252
- Tjark Klüttermann 590236
- Jan Peter Lamp 590235

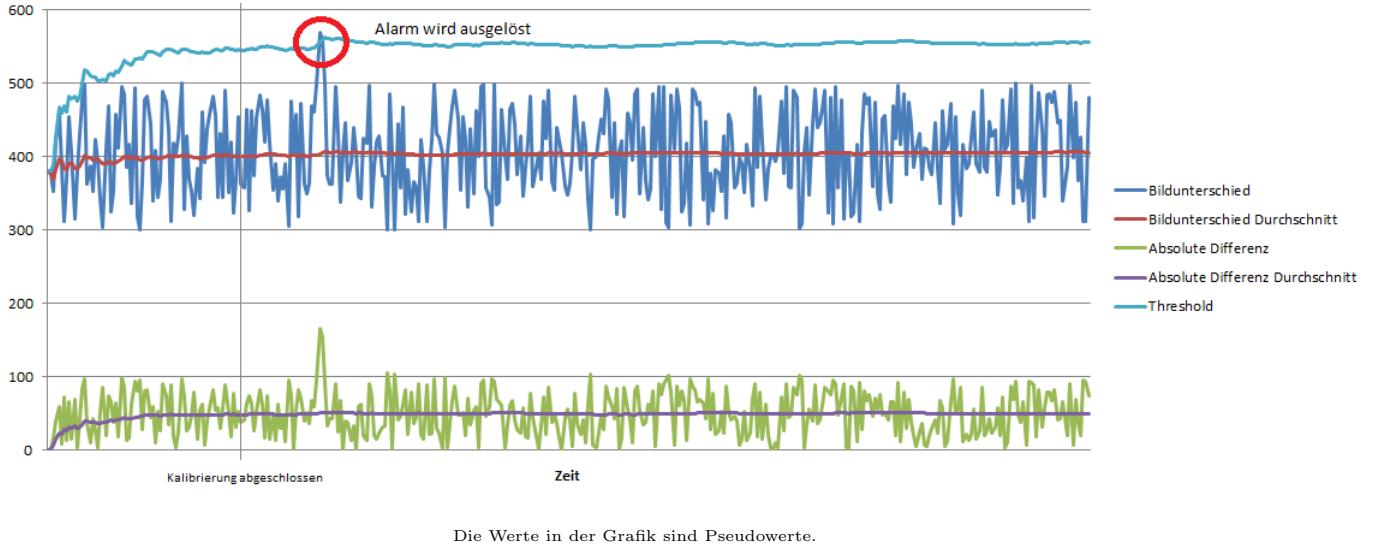
## 2 Aufgabenverteilung

- **UI-Design**  
Jannes Peters
- **Sounddesign**  
Adrian Kurth und Jannes Peters
- **Software-Architektur**  
Jannes Peters
- **Einbinden von OpenCV**  
Jan Peter Lamp
- **Prototyping**  
Adrian Kurth, Jannes Peters, Tjark Klüttermann und Jan Peter Lamp
- **Algorithmus**  
Adrian Kurth, Jannes Peters, Tjark Klüttermann und Jan Peter Lamp
- **Red-Highlighting**  
Jannes Peters
- **Refactoring**  
Adrian Kurth, Jannes Peters, Tjark Klüttermann und Jan Peter Lamp

## 3 Programmstruktur

Das Programm besteht aus einer Activity und einem Custom-View.

## 4 Algorithmus



Wir messen die Differenz der aufeinanderfolgenden Frames(dunkelblau). Daraus wird ein gleitender Mittelwert errechnet(weinrot). Die Differenz zwischen dem absoluten und dem durchschnittlichen Unterschied wird festgestellt(grün). Davon wird der Durchschnitt berechnet(lila). Der Threshold wird aus wie folgt berechnet:

Gleitender Mittelwert des Bildunterschieds(weinrot) + (6 \* Gleitender Mittelwert der Differenz(lila))

wobei die 6 ein durch Testen herausgefunder Faktor ist.

Die Änderung der Bilder wird durch eine Matrize dargestellt. Um die aktuelle Änderung in den Frame einzulegen, wird die aktuelle Änderung in den roten Kanal eines leeren Bildes kopiert. Dieses Bild wird dann mit dem Schwarz-Weiß-Bild addiert und gezeichnet. Somit sieht man die Änderungen in rot auf dem Bild.

## 5 Klassen

### 5.1 MainActivity.java

Die MainActivity ist die View und der Controller. Hier befindet sich die State-Machine der App.

### 5.2 AlarmCameraView.java

Diese Klasse erweitert die JavaCameraView von OpenCV um die Funktionalität des Alarms.

## 6 Materialien

- Android Studio 3
- Asus Zenfone 2 (Android CM 7.1.2)
- OnePlus X (Android 6.0.1)

- Asus Zenfone 2 (Android 5.0)
- Discord (Voicechat)
- Teamviewer 13
- Git

## 7 Testbericht

Die App wurde verschiedenen Smartphones und Android-Versionen getestet (siehe Punkt Materialien). Dabei lief die App problemlos auf allen Geräten.  
Um zu Prüfen, ob die App zum richtigen Zeitpunkt auslöst, haben wir verschiedene Bewegungen und Schattenbilder getestet.

## 8 Probleme

Folgende Probleme traten während der Entwicklung auf:

- **Android Dokumentation für OpenCV nicht vorhanden**

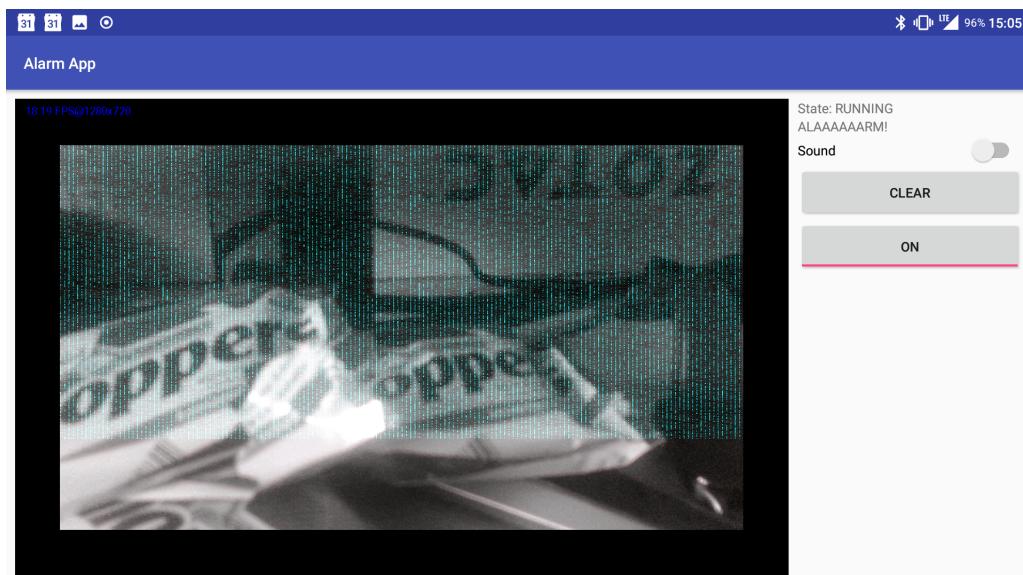
Dadurch war es manchmal schwierig, den Ort von Funktionen oder Konstanten zu finden, selbst wenn man sie in der C-Dokumentation gefunden hat.

- **Nur OpenCV-Version 2.4.2 lief ohne Probleme**

Alle anderen Versionen liefen nur teilweise oder gar nicht (SIGSEV-Fehler bei einigen Funktionen).

- **Renderfehler**

Dies ist ein nicht reproduzierbarer Fehler, der manchmal auftritt und durch einen Neustart behebbar ist.



## 9 Sourcen

### 9.1 MainActivity.java

```
File - MainActivity.java
1 package com.example.alarm.alarmapp;
2
3 import android.Manifest;
4 import android.content.Intent;
5 import android.content.pm.PackageManager;
6 import android.media.MediaPlayer;
7 import android.os.Bundle;
8 import android.os.Handler;
9 import android.os.Message;
10 import android.support.annotation.NonNull;
11 import android.support.v4.app.ActivityCompat;
12 import android.support.v4.content.ContextCompat;
13 import android.support.v7.app.AppCompatActivity;
14 import android.util.Log;
15 import android.view.SurfaceView;
16 import android.view.View;
17 import android.view.WindowManager;
18 import android.widget.Button;
19 import android.widget.Switch;
20 import android.widget.TextView;
21 import android.widget.ToggleButton;
22
23 import com.example.alarm.alarmapp.views.AlarmCameraView;
24
25 import org.opencv.android.BaseLoaderCallback;
26 import org.opencv.android.LoaderCallbackInterface;
27 import org.opencv.android.OpenCVLoader;
28
29 /**
30  * This is the MainActivity. It handles all the ui related stuff like
31  * buttons and controls the AlarmCameraView.
32 */
33 public class MainActivity extends AppCompatActivity implements Handler
34     .Callback, View.OnClickListener, AlarmCameraView.IAlarmCameraListener
35 {
36     private static final String TAG = "MainActivity";
37     private static final int TIMEOUT_START = 10000;
38     private AlarmCameraView mCameraView;
39     private ToggleButton mTbtnStartStop;
40     private Switch mSwSound;
41     private TextView mTvAlarmTriggered, mTvState;
42     private boolean mHasPermission = false;
43
44     private Runnable mRunnableStartTimer = () -> onCommand(Command.
45     START_TIMER);
46
47     private MediaPlayer mAlarmPlayer;
48     private Handler mUiHandler;
49
50     private State mState = State.IDLE;
51
52     @Override
```

File - MainActivity.java

```
51     public void onAlarm() {
52         Log.d(TAG, "onAlarm()");
53         onCommand(Command.ALARM);
54     }
55
56     @Override
57     public void onCalibrating() {
58         Log.d(TAG, "onCalibrating()");
59         onCommand(Command.CALIBRATING);
60     }
61
62     @Override
63     public void onRun() {
64         Log.d(TAG, "onRun()");
65         onCommand(Command.RUNNING);
66     }
67
68     @Override
69     public Handler getHandler() {
70         return mUiHandler;
71     }
72
73     private enum State {
74         IDLE, WAITING_TO_START, CALIBRATING, RUNNING
75     }
76
77     private enum Command {
78         BUTTON_START_STOP, START_TIMER, ALARM, CALIBRATING, RUNNING,
79         ON_PAUSE, RESET_ALARM_TEXT
80     }
81
82     @Override
83     protected void onCreate(Bundle savedInstanceState) {
84         super.onCreate(savedInstanceState);
85
86         //check for camera permission, if it is not granted invoke a
87         //layout that displays an error
88         mHasPermission = ContextCompat.checkSelfPermission(this,
89             Manifest.permission.CAMERA) == PackageManager.PERMISSION_GRANTED;
90         if (!mHasPermission) {
91             setContentView(R.layout.activity_permission_missing)
92             ;
93             ActivityCompat.requestPermissions(this, new String[] {
94                 Manifest.permission.CAMERA}, 0);
95             return; //cancel remaining init. To start the real app we
96             //need permission from the start.
97         }
98
99         setContentView(R.layout.activity_main);
100
101         getWindow().addFlags(WindowManager.LayoutParams.
102             FLAG_KEEP_SCREEN_ON);
103
104         mUiHandler = new Handler(this);
105     }
```

Page 2 of 6

File - MainActivity.java

```
98         mAlarmPlayer = MediaPlayer.create(this, R.raw.sound_alarm);
99
100        mCameraView = findViewById(R.id.cameraView);
101        mTvAlarmTriggered = findViewById(R.id.tvAlarmTriggered);
102        mTvState = findViewById(R.id.tvState);
103        mSwSound = findViewById(R.id.swAlarmSound);
104        mTbtnStartStop = findViewById(R.id.tbtnStartStop);
105
106        mTbtnStartStop.setOnClickListener(this);
107
108        mCameraView.setVisibility(SurfaceView.VISIBLE);
109        mCameraView.setAlarmListener(this);
110    }
111
112    @Override
113    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
114        switch (requestCode) {
115            case 0:
116                //restart app if we got the permission
117                if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
118                    Intent newActivity = new Intent(this,
119                        MainActivity.class);
120                    startActivity(newActivity);
121                    finish();
122                }
123            break;
124        }
125    }
126
127    private void onCommand(Command cmd) {
128        Log.d(TAG, "Command: " + cmd.toString() + " received in state : " + mState.toString());
129        if (cmd == Command.RESET_ALARM_TEXT) mTvAlarmTriggered.setText("false");
130        switch (mState) {
131            case IDLE:
132                switch (cmd) {
133                    case BUTTON_START_STOP:
134                        mState = State.WAITING_TO_START;
135                        mUiHandler.postDelayed(mRunnableStartTimer,
136                                TIMEOUT_START);
137                        break;
138                    default:
139                        Log.e(TAG, "invalid command in " + mState +
140                            ": " + cmd);
141                }
142            break;
143            case WAITING_TO_START:
144                switch (cmd) {
145                    case START_TIMER:
146                        mCameraView.startAlarm();
147                    break;
148                }
149        }
150    }
```

Page 3 of 6

File - MainActivity.java

```
145             case CALIBRATING:
146                 mState = State.CALIBRATING;
147                 break;
148             case BUTTON_START_STOP:
149                 mState = State.IDLE;
150                 mUiHandler.removeCallbacks(
151                     mRunnableStartTimer);
152                 break;
153             case ON_PAUSE:
154                 mState = State.IDLE;
155                 mUiHandler.removeCallbacks(
156                     mRunnableStartTimer);
157                 break;
158             default:
159                 Log.e(TAG, "invalid command in " + mState + ": "
160                     + cmd);
161             }
162             break;
163         case CALIBRATING:
164             switch (cmd) {
165                 case BUTTON_START_STOP:
166                 case ON_PAUSE:
167                     mState = State.IDLE;
168                     mCameraView.stopAlarm();
169                     break;
170                 case RUNNING:
171                     mState = State.RUNNING;
172                     break;
173                 default:
174                     Log.e(TAG, "invalid command in " + mState +
175                         ": " + cmd);
176             }
177             break;
178         case RUNNING:
179             switch (cmd) {
180                 case BUTTON_START_STOP:
181                 case ON_PAUSE:
182                     mState = State.IDLE;
183                     mCameraView.stopAlarm();
184                     break;
185                 case ALARM:
186                     if(mSwSound.isChecked()) playAlarmSound();
187                     mTvAlarmTriggered.setText("ALAAAAAARM!");
188                     mUiHandler.postDelayed(() -> onCommand(
189                         Command.RESET_ALARM_TEXT), 2000);
190                     break;
191                 default:
192                     Log.e(TAG, "invalid command in " + mState +
193                         ": " + cmd);
194             }
195             break;
196         default:
197             Log.e(TAG, "State " + mState + " not implemented.");
198     }
```

Page 4 of 6

File - MainActivity.java

```
193         Log.d(TAG, "State after command execution: " + mState.  
194             toString());  
195             mTvState.setText(String.format(getString(R.string.state_val),  
196                 mState.toString()));  
197             //set start stop button to correct rendering for the current  
198             state  
199             mTbtnStartStop.setChecked(mState == State.RUNNING || mState  
200                 == State.CALIBRATING || mState == State.WAITING_TO_START);  
201         }  
202     }  
203     //region lifecycle  
204  
205     @Override  
206     protected void onResume() {  
207         super.onResume();  
208         if (mHasPermission) {  
209             if (!OpenCVLoader.initDebug()) {  
210                 Log.d(TAG, "Internal OpenCV library not found. Using  
211                     OpenCV Manager for initialization");  
212                 OpenCVLoader.initAsync(OpenCVLoader.  
213                     OPENCV_VERSION_2_4_2, this, mLoaderCallback);  
214             } else {  
215                 Log.d(TAG, "OpenCV library found inside package.  
216                     Using it!");  
217                 mLoaderCallback.onManagerConnected(  
218                     LoaderCallbackInterface.SUCCESS);  
219             }  
220         }  
221     }  
222     @Override  
223     protected void onStart() {  
224         super.onStart();  
225     }  
226     @Override  
227     protected void onPause() {  
228         super.onPause();  
229         if (mHasPermission) {  
230             if (mCameraView != null) {  
231                 mCameraView.disableView();  
232                 onCommand(Command.ON_PAUSE);  
233             }  
234         }  
235     }  
236     //endregion  
237     private void playAlarmSound() {  
238         if (!mAlarmPlayer.isPlaying()) {  
239             mAlarmPlayer.seekTo(0);  
240             mAlarmPlayer.start();  
241         }  
242     }  
243 }
```

File - MainActivity.java

```
239     private BaseLoaderCallback mLoaderCallback = new
240         BaseLoaderCallback(this) {
241             @Override
242             public void onManagerConnected(int status) {
243                 switch (status) {
244                     case LoaderCallbackInterface.SUCCESS:
245                         {
246                             Log.i(TAG, "OpenCV loaded successfully");
247                             mCameraView.enableView();
248
249                         } break;
250                     default:
251                         {
252                             super.onManagerConnected(status);
253                         } break;
254                     }
255                 };
256
257
258             @Override
259             public boolean handleMessage(Message msg) {
260                 return false;
261             }
262
263             @Override
264             public void onClick(View v) {
265                 switch (v.getId()) {
266                     case R.id.tbtnStartStop:
267                         onCommand(Command.BUTTON_START_STOP);
268                         break;
269                     }
270                 }
271             }
272 }
```

## 9.2 AlarmCameraView.java

File - AlarmCameraView.java

```
1 package com.example.alarm.alarmapp.views;
2
3 import android.content.Context;
4 import android.hardware.Camera;
5 import android.os.Handler;
6 import android.util.AttributeSet;
7 import android.util.Log;
8
9 import com.example.alarm.alarmapp.R;
10
11 import org.opencv.android.CameraBridgeViewBase;
12 import org.opencv.android.JavaCameraView;
13 import org.opencv.core.Core;
14 import org.opencv.core.CvType;
15 import org.opencv.core.Mat;
16 import org.opencv.imgproc.Imgproc;
17
18 import java.util.ArrayList;
19 import java.util.Date;
20
21 /**
22 * A camera view that has a movement detection.
23 */
24 public class AlarmCameraView extends JavaCameraView implements
25     CameraBridgeViewBase.CvCameraViewListener2 {
26     private static final String TAG = AlarmCameraView.class.getName();
27
28     private int mTimeToCalibrate = 10000;
29     private double mAlarmThreshold = 6d;
30     private int mProcessFps = 4;
31     private double mAverageOver = 100d / mProcessFps;
32
33     private long mCalibratingStartedAt = 0;
34
35     private boolean mAreCameraParamsSet = false;
36     private State mState = State.IDLE;
37     private IAlarmCameraListener mAlarmListener = null;
38     private Mat mLastMat = null;
39     private Mat mEmptyGrayMap = null;
40
41     private double mMovingAbsDiffAvg = -1d;
42     private double mMovingDiffAvg = -1d;
43     private double mMaxDiff = 0;
44
45     public AlarmCameraView(Context context, int cameraId) {
46         super(context, cameraId);
47         this.setCvCameraViewListener(this);
48     }
49
50     public AlarmCameraView(Context context, AttributeSet attrs) {
51         super(context, attrs);
52         this.setCvCameraViewListener(this);
53     }
```

Page 1 of 5

File - AlarmCameraView.java

```
54     public interface IAlarmCameraListener {
55         void onAlarm();
56         void onCalibrating();
57         void onRun();
58         Handler getHandler();
59     }
60
61     private void onAlarmInternal() {
62         Log.d(TAG, "onAlarmInternal");
63         if (mAlarmListener != null) mAlarmListener.getHandler().post(
64             () -> mAlarmListener.onAlarm());
65     }
66
67     private void onCalibratingInternal() {
68         Log.d(TAG, "onCalibratingInternal");
69         if (mAlarmListener != null) mAlarmListener.getHandler().post(
70             () -> mAlarmListener.onCalibrating());
71     }
72
73     private void onRunInternal() {
74         Log.d(TAG, "onRunInternal");
75         if (mAlarmListener != null) mAlarmListener.getHandler().post(
76             () -> mAlarmListener.onRun());
77     }
78
79     public enum State {
80         IDLE, CALIBRATING, RUNNING
81     }
82
83     public State getCurrState() {
84         return mState;
85     }
86
87     public void startAlarm() {
88         mCalibratingStartedAt = System.currentTimeMillis();
89         mState = State.CALIBRATING;
90         Log.d(TAG, "State: " + mState);
91         onCalibratingInternal();
92     }
93
94     public void stopAlarm() {
95         mCalibratingStartedAt = 0;
96         mMovingAbsDiffAvg = -1;
97         mMaxDiff = 0;
98
99         mState = State.IDLE;
100    }
101
102    public void setAlarmListener(IAlarmCameraListener alarmListener)
103    {
104        this.mAlarmListener = alarmListener;
105    }
106
107    public void removeAlarmListener() {
```

Page 2 of 5

File - AlarmCameraView.java

```
104         this.mAlarmListener = null;
105     }
106
107     @Override
108     public void onPreviewFrame(byte[] frame, Camera arg1) {
109         if (!mAreCameraParamsSet) {
110             mAreCameraParamsSet = true;
111             Camera.Parameters params = mCamera.getParameters();
112             params.setFocusMode(Camera.Parameters.FOCUS_MODE_FIXED);
113             mCamera.setParameters(params);
114         }
115         super.onPreviewFrame(frame, arg1);
116     }
117
118     @Override
119     public void onCameraViewStarted(int width, int height) {
120         mEmptyGrayMap = new Mat(height, width, CvType.CV_8UC1);
121     }
122
123     @Override
124     public void onCameraViewStopped() {
125         if (mEmptyGrayMap != null) mEmptyGrayMap.release();
126         mEmptyGrayMap = null;
127         if (mLastMat != null) mLastMat.release();
128         mLastMat = null;
129     }
130
131     @Override
132     public Mat onCameraFrame(CvCameraViewFrame inputFrame) {
133         Mat screenMat = inputFrame.rgba();
134
135         if (mState == State.RUNNING || mState == State.CALIBRATING) {
136             if (mState == State.CALIBRATING) {
137
138                 if (System.currentTimeMillis() -
139                     mCalibratingStartedAt >= mTimeToCalibrate) {
140                     mState = State.RUNNING;
141                     onRunInternal();
142                     Log.d(TAG, "State: " + mState);
143                 }
144
145             if (mLastMat != null) {
146                 //calculate trigger values
147                 Mat grayMat = inputFrame.gray();
148                 Mat diff = new Mat(grayMat.size(), grayMat.type());
149                 Core.absdiff(grayMat, mLastMat, diff);
150                 final double diffD = Core.mean(diff).val[0];
151                 if (mMovingAbsDiffAvg == -1d) mMovingAbsDiffAvg =
152                     diffD;
153                 else {
154                     mMovingAbsDiffAvg = (mMovingAbsDiffAvg * (
155                         mAvergeOver - 1) + diffD) / mAvergeOver;
156                 }
157             }
158         }
159     }
160 }
```

Page 3 of 5

```
File - AlarmCameraView.java
155             if (mMovingDiffAvg == -1d) mMovingDiffAvg = 0;
156         else {
157             mMovingDiffAvg = (mMovingDiffAvg * (mAverageOver
158 - 1) + Math.abs(diffD - mMovingAbsDiffAvg)) / mAverageOver;
159         }
160         if (mMaxDiff < diffD) {
161             mMaxDiff = diffD;
162         }
163         double absCurrAlarmThreshold = (mMovingDiffAvg *
164             mAlarmThreshold) + mMovingAbsDiffAvg;
165         boolean alarmTriggered = diffD >
166             absCurrAlarmThreshold;
167         if (alarmTriggered) Log.d(TAG, "Alarm Triggered: " +
168             new Date().toGMTString());
169         Log.v(TAG,
170             String.format("onProcessedFrame:\t%s\t%s\t%s\t%s\
171             \t%s",
172                 String.format(getContext().getString(R.string
173                     .curr_alarm_threshold_val), absCurrAlarmThreshold),
174                 String.format(getContext().getString(R.string
175                     .moving_diff_abs_avg_val), mMovingAbsDiffAvg),
176                 String.format(getContext().getString(R.string
177                     .moving_diff_avg_val), mMovingDiffAvg),
178                 String.format(getContext().getString(R.string
179                     .max_diff_val), mMaxDiff),
180                 String.format(getContext().getString(R.string
181                     .AbsDiff_val), diffD)
182             )
183         );
184         if (alarmTriggered && mState == State.RUNNING)
185             onAlarmInternal();
186         //draw changes on screenMat
187         ArrayList<Mat> mergeMats = new ArrayList<>(3);
188         mergeMats.add(diff);
189         mergeMats.add(mEmptyGrayMap);
190         mergeMats.add(mEmptyGrayMap);
191         //create buffer for red diff (rgb mat)
192         Mat matRedDiff = new Mat(diff.size(), screenMat.type(
193             ));
194         //merge two empty single channel mats and the diff
195         //mat as different channels into rgb mat
196         Core.merge(mergeMats, matRedDiff);
197         //create buffer for gray image in rgb mat
198         Mat grayRgbMat = new Mat(screenMat.size(), screenMat.
199             type());
200         //convert gray single channel mat to rgb mat
201         Imgproc.cvtColor(grayMat, grayRgbMat, Imgproc.
202             COLOR_GRAY2RGB);
203         //add the gray and red overlay together
204         Core.add(grayRgbMat, matRedDiff, screenMat);
```

Page 4 of 5

File - AlarmCameraView.java

```
194             //release resources
195             grayRgbMat.release();
196             grayMat.release();
197             matRedDiff.release();
198             diff.release();
199             mLstMat.release();
200         }
201     }
202     mLstMat = inputFrame.gray();
203 }
204     return screenMat;
205 }
206 }
207 }
```

### 9.3 AndroidManifest.xml

File - AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.alarm.alarmapp">
4
5     <uses-permission android:name="android.permission.CAMERA"/>
6
7     <uses-feature android:name="android.hardware.camera" android:
8         required="false"/>
9     <uses-feature android:name="android.hardware.camera.autofocus"
10        android:required="false"/>
11    <uses-feature android:name="android.hardware.camera.front" android
12        :required="false"/>
13    <uses-feature android:name="android.hardware.camera.front.
14        autofocus" android:required="false"/>
15
16    <application
17        android:allowBackup="true"
18        android:icon="@mipmap/ic_launcher"
19        android:label="@string/app_name"
20        android:roundIcon="@mipmap/ic_launcher_round"
21        android:supportsRtl="true"
22        android:theme="@style/AppTheme">
23            <activity android:name=".MainActivity"
24                android:screenOrientation="landscape">
25                <intent-filter>
26                    <action android:name="android.intent.action.MAIN" />
27                </intent-filter>
28            </activity>
29        </application>
30
31 </manifest>
```

## 9.4 activity\_main.xml

File - activity\_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://
schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     xmlns:opencv="http://schemas.android.com/apk/res-auto"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context="com.example.alarm.alarmapp.MainActivity">
9
10    <com.example.alarm.alarmapp.views.AlarmCameraView
11        android:id="@+id/cameraView"
12        android:layout_width="0dp"
13        android:layout_height="0dp"
14        android:layout_marginBottom="8dp"
15        android:layout_marginEnd="8dp"
16        android:layout_marginStart="8dp"
17        android:layout_marginTop="8dp"
18        android:visibility="visible"
19        app:layout_constraintBottom_toBottomOf="parent"
20        app:layout_constraintEnd_toStartOf="@+id/linearLayout"
21        app:layout_constraintStart_toStartOf="parent"
22        app:layout_constraintTop_toTopOf="parent"
23        opencv:camera_id="any"
24        opencv:show_fps="true"/>
25
26    <LinearLayout
27        android:id="@+id/linearLayout"
28        android:layout_width="200dp"
29        android:layout_height="0dp"
30        android:layout_marginBottom="8dp"
31        android:layout_marginRight="8dp"
32        android:layout_marginTop="8dp"
33        android:orientation="vertical"
34        app:layout_constraintBottom_toBottomOf="parent"
35        app:layout_constraintEnd_toEndOf="parent"
36        app:layout_constraintTop_toTopOf="parent"
37        android:layout_marginEnd="8dp">
38
39        <TextView
40            android:id="@+id/tvState"
41            android:layout_width="match_parent"
42            android:layout_height="wrap_content"
43            android:text="@string/idle"/>
44
45        <TextView
46            android:id="@+id/tvAlarmTriggered"
47            android:layout_width="match_parent"
48            android:layout_height="wrap_content"
49            android:text="@string/alarm_triggered_val"/>
50
51        <Switch
52            android:id="@+id/swAlarmSound"
53            android:layout_width="match_parent"
```

File - activity\_main.xml

```
54         android:layout_height="wrap_content"
55         android:checked="false"
56         android:text="Sound"/>
57
58     <ToggleButton
59         android:id="@+id/tbtnStartStop"
60         android:layout_width="match_parent"
61         android:layout_height="wrap_content"
62         android:text="ToggleButton"/>
63
64     </LinearLayout>
65 </android.support.constraint.ConstraintLayout>
66
67
```

## 9.5 activity\_main\_permission\_missing.xml

File - activity\_main\_permission\_missing.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://
  schemas.android.com/apk/res/android"
3                               xmlns:app="http://schemas
  .android.com/apk/res-auto"
4                               xmlns:tools="http://
  schemas.android.com/tools"
5                               android:layout_width="
  match_parent"
6                               android:layout_height="
  match_parent">
7
8   <TextView
9     android:id="@+id/textView"
10    android:layout_width="wrap_content"
11    android:layout_height="wrap_content"
12    android:layout_marginBottom="8dp"
13    android:layout_marginEnd="8dp"
14    android:layout_marginStart="8dp"
15    android:layout_marginTop="8dp"
16    android:text="This app requires the camera permission to run."
17    app:layout_constraintBottom_toBottomOf="parent"
18    app:layout_constraintEnd_toEndOf="parent"
19    app:layout_constraintStart_toStartOf="parent"
20    app:layout_constraintTop_toTopOf="parent"/>
21 </android.support.constraint.ConstraintLayout>
```