

بسمه تعالی



دانشگاه صنعتی امیرکبیر  
دانشکده مهندسی کامپیوتر و فناوری اطلاعات

گزارش پروژه درس محاسبات فراگیر

تشخیص زمینه با استفاده از گوشی همراه

عاطفه زارع

۹۲۱۳۱۰۰۶

تابستان ۱۳۹۳

## فهرست مطالب

۱-	هدف پروژه.....	۳
۲-	الگوریتم KNN.....	۳
۳-	توصیف پروژه.....	۴
۳-۱-	CLASS DIAGRAM پروژه.....	۵
۴-	اجرای پروژه.....	۶

## ۱- هدف پروژه

هدف از انجام این پروژه تشخیص زمینه (context) با استفاده از تلفن همراه (mobile) بوده است. در این راستا از اطلاعات دریافتی از حسگرهای (sensor) موجود در تلفن همراه کاربر برای تشخیص اطلاعات سطح بالاتر استفاده می شود. پروژه ویژه تلفن های همراه اندروید (android) نوشته شده است و اطلاعات حرکتی کاربر شناسایی می شود و ویژگی بارز آن کارایی بالای آن در حافظه و محاسبات می باشد. برای این منظور سعی شده است از کمینه پردازش در سمت کاربر استفاده شود و با انتخاب الگوریتم ها و ساختار های داده ای (data structure) مناسب به این هدف دست یابیم.

از میان اطلاعات حسگر های مختلف که در فاز یک پروژه شناسایی و نمایش داده شد تنها از حسگر شتاب سنج (accelerometer) استفاده شده است و از آنجایی که دریافت مداوم اطلاعات حسگر ها بار پردازشی برای کاربر به وجود می آورد و همچنین حسگر شتاب سنج با دقت بسیار بالایی حرکت کاربر را توصیف می کند تنها به اطلاعات این حسگر بسنده شده است.

## ۲- الگوریتم Knn

با استفاده از الگوریتم k نزدیکترین همسایه (k nearest neighborhood) و یا به اختصار knn از اطلاعات دریافتی از حسگر شتاب سنج کاربر، نوع حرکت کاربر شناسایی می گردد. انواع حرکات کاربر که در این جا تشخیص داده می شوند عبارتند از:

CLASSIFIED/IDLE/SITTING  
CLASSIFIED/IDLE/STANDING  
CLASSIFIED/WALKING  
CLASSIFIED/WALKING/STAIRS/DOWN  
CLASSIFIED/WALKING/STAIRS/UP  
CLASSIFIED/VEHICLE/BUS  
CLASSIFIED/VEHICLE/CAR  
CLASSIFIED/DANCING

این الگوریتم که در فایل ضمیمه (KNNclassify) به تفصیل شرح داده شده است. نقاط ویژگی (features point) در این پروژه دارای ۴ بعد می باشند که تشکیل یافته اند از متوسط مقدار شتاب سنج در محور Y، متوسط مقدار شتاب سنج در محور Z، فاصله مقدار بیشینه و کینه شتاب سنج در محور Y و فاصله مقدار بیشینه و کینه شتاب سنج در محور Z.

در مدل ضمیمه my model.xlsx مقادیر چهار بعد از نقاط و دسته درست آنها آورده شده اند. این مقادیر با کمک نرم افزار ایجاد شده در فاز یک پروژه با نام fantastic.apk که در ضمیمه آورده شده است تهیه شده است. سپس این مدل با استفاده از یک ابزار جاوایی desktop در یک ساختار داده ای map ریخته شده و سپس serialize شده و در یک فایل نوشته شده است. این فایل با نام basic\_model در پروژه ActivityRecorder وجود دارد که توسط classifier مورد استفاده قرار می گیرد. این الگوریتم به صورت بسیار کارا در کلاس classier پیاده سازی شده است.

اگر n تعداد درایه های model باشد و p تعداد ابعاد هر نقطه باشد (در این جا  $p=4$  می باشد)، مرتبه محاسباتی این الگوریتم  $O(pn)$  می باشد و از نظر حافظه به غیر از مدل داده ی قابل توجه دیگری ذخیره سازی نمی شود. کد classifier را در ذیل مشاهده می نمایید:

```
public class Classifier {  
  
    private final Set<Map.Entry<Float[], String>> model;
```

```

public Classifier(final Set<Entry<Float[], String>> model) {
    this.model = model;
}

public String classify(final float[] data) {
    final float oddTotal = data[5], evenTotal = data[2];
    final float oddMin = data[3], oddMax = data[4];
    final float evenMin = data[0], evenMax = data[1];

    final float[] points = {
        Math.abs(evenTotal / 128),
        Math.abs(oddTotal / 128),
        evenMax - evenMin,
        oddMax - oddMin
    };

    float bestDistance = Float.MAX_VALUE;
    String bestActivity = "UNCLASSIFIED/UNKNOWN";

    for (Map.Entry<Float[], String> entry : model) {
        float distance = 0;

        for (int i = 0; i < points.length; i++) {
            distance += Math.pow(points[i] - entry.getKey()[i], 2);
        }

        if (distance < bestDistance) {
            bestDistance = distance;
            bestActivity = entry.getValue();
        }
    }

    return bestActivity;
}
}

```

### ۳- توصیف پروژه

بخش برنامه نویسی در قالب دو پروژه انجام شده است. بخش اول یک پروژه با ساختار android و با نام ActivityRecorder می باشد و بخش دوم آن در قالب یک پروژه جاوایی با نام common می باشد. ActivityRecorder دارای وابستگی (dependency) به common می باشد و هر دو آنها در قالب پروژه های eclipse IDE می باشند و با استفاده از ابزار apache ant نیز build می شوند.

نام برخی از کلاس ها و وظایف آنها در ذیل آورده شده اند:

AccelerReader: interface خواننده اطلاعات شتاب سنج.

RealAccelerReader: یکی از کلاس های پیاده سازی کننده AccelerReader و دریافت کننده داده های محور Y و Z از شتاب سنج.

AccelerReaderFactory: ایجاد کننده نمونه مناسب از AccelerReader.

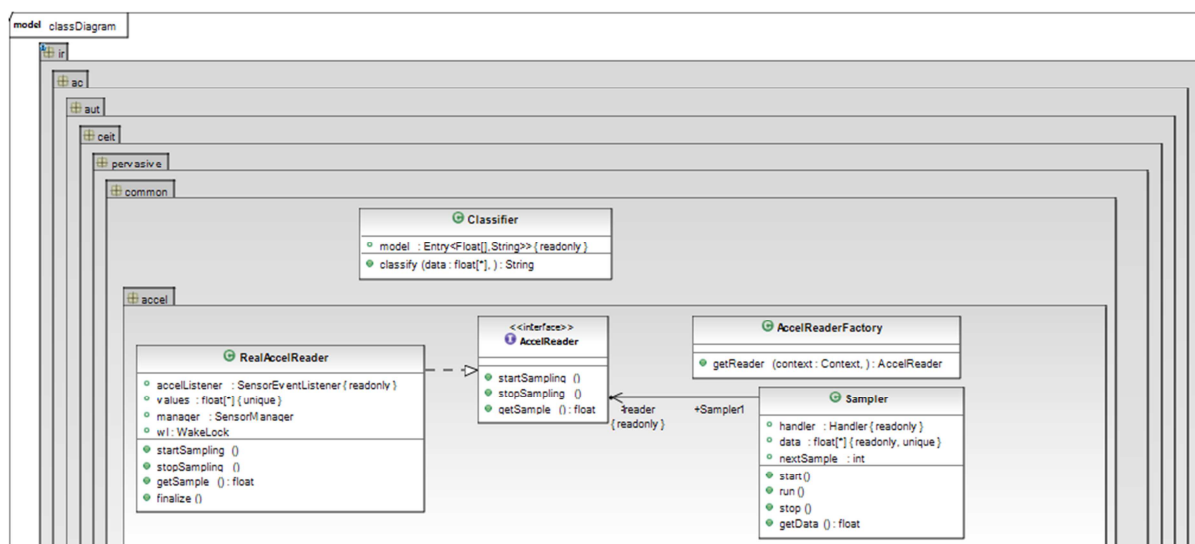
RecorderService: یک نوع سرویس اندروید می باشد که از thread های مختلف نمونه ایجاد کرده و با استفاده از آنها از کلاس sampler نمونه مناسب ساخته و آن را اجرا می کند. همچنین مدل را با استفاده از ModelReader خوانده و به صورت static ذخیره و در اختیار کلاس های دیگر قرار می دهد.

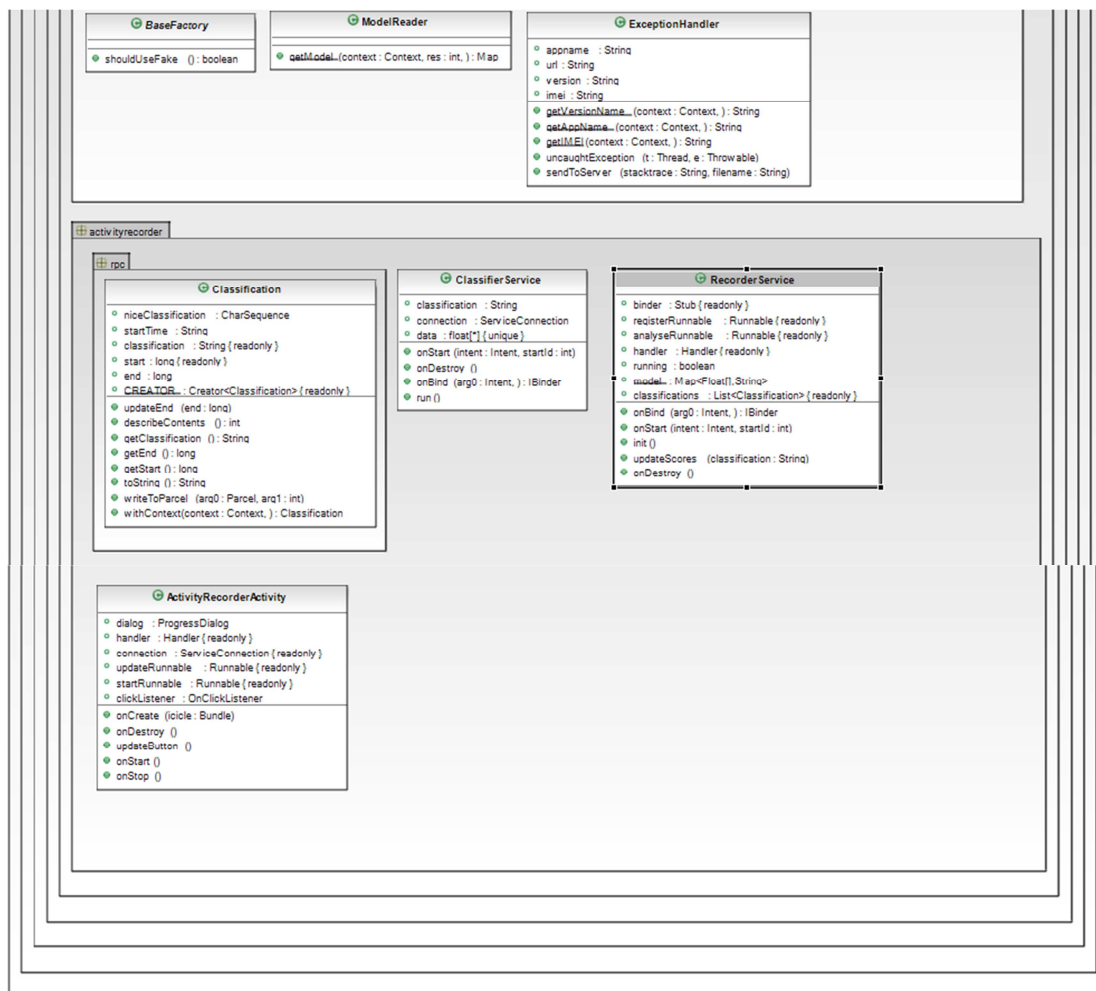
Sampler: با شروع این thread، ۱۲۸ نمونه از حسگر شتاب سنج از AccelReader دریافت می کند و مقدار کمینه، بیشینه، مجموع مقادیر را محاسبه می کند سپس وقفه زمانی ۵۰ میلی ثانیه ای داده و مجدداً این کار را انجام می دهد. با هر بار وقفه زمانی thread مربوط به RecorderService فراخوانی می شود که در نتیجه آن ClassifierService اجرا شده و داده های جمع آوری شده به آن پاس می شود.

ClassifierService: داده های دریافتی از حسگر و داده های مدل را به classifier می دهد.  
Classifier.java: اجرا کننده الگوریتم knn بر روی داده های خوانده شده از sensor ها و نتیجه گیری نوع رسته فعالیت کاربر می باشد. به این صورت که از داده های دریافتی از حسگر ها چهار نقطه یاد شده برای الگوریتم را ایجاد می نماید و سپس با جستجوی مدل نزدیکترین نقطه از مدل به این نقطه حسگرها را محاسبه می نماید و رسته مربوط به نقطه یافت شده به عنوان رسته نقطه حسگرها اعلام می شود که با عنوان bestActivity در کد یاد شده است.

Class diagram این پروژه با استفاده از myEclipse رسم شده و با نام classDiagram.uml در پروژه موجود است که با ابزار های UML قابل مشاهده است همچنین نسخه pdf آن نیز در ضمیمه وجود دارد.  
بخش هایی از پروژه common جهت استخراج زمینه مکان کاربر می باشد که به صورت نهایی نشده وجود دارد.

### ۳-۱- Class Diagram پروژه

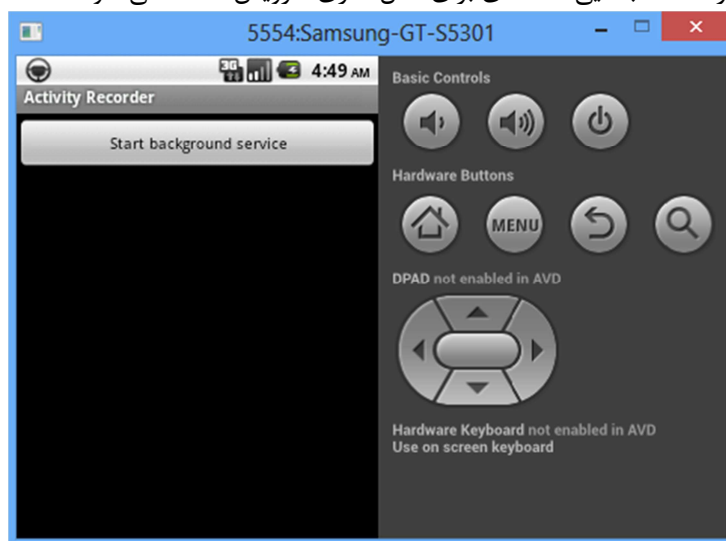




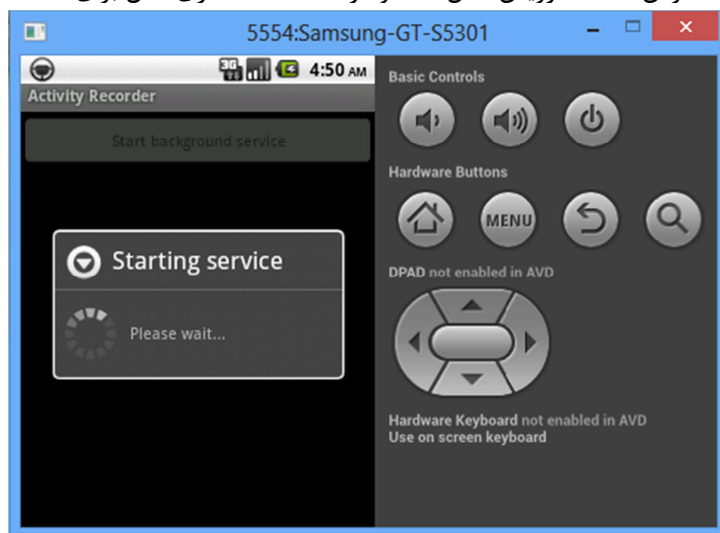
#### ۴- اجرای پروژه

نمونه ای از اجرای پروژه ActivityReader را در شبیه ساز Eclipse با پیکربندی Samsung-GT-S5301 در شکل های ذیل مشاهده می نمایید.

در صفحه ابتدایی دگمه ای برای فعال سازی سرویس مشاهده می شود:



با فشردن دگمه سرویس فعال شده و در مدت آماده سازی مدل برای استفاده ProgressDialog به کاربر نشان داده شده است:



با شروع سرویس و دریافت اطلاعات شتاب سنج و با استفاده از فرایندی که توضیح داده شد فعالیت کاربر تشخیص داده شده و با برچسب زمانی به صورت لیستوار نمایش داده می شود. در صورتی که کاربر برای مدتی در یک وضعیت قرار داشته باشد داده های آن جمع شده و تنها یک درایه نمایش داده می شود:

