**MINISTRY OF EDUCATION AND SCIENCE OF**

**THE KYRGYZ REPUBLIC**

**INTERNATIONAL ATATURK ALATOO UNIVERSITY**

**FACULTY OF NEW TECHNOLOGIES**

**DEPARTMENT OF APPLIED MATHEMATICS AND INFORMATICS**

**BUILDING OPTICAL TEST GRADING SYSTEM USING IMAGE PROCESSING**

**Diploma Paper**

**By Gulbustan Baudunova**

**Thesis Advisor: Marlen Akimaliev**

**Bishkek – 2017**

**THE MINISTRY OF EDUCATION AND SCIENCE OF**

**THE KYRGYZ REPUBLIC**

**INTERNATIONAL ATATURK ALATOO UNIVERSITY**

**FACULTY OF NEW TECHNOLOGIES**

**DEPARTMENT OF APPLIED MATHEMATICS AND INFORMATICS**

** BUILDING OPTICAL TEST GRADING SYSTEM USING IMAGE PROCESSING**

**Diploma paper**

**By Gulbustan Baudunova**

|  |  |
| --- | --- |
| **Thesis Advisor:**  **Marlen Akimaliev** | **Date:**  **05.06.2017** |
| **Head of Department:**  **Dr. Musa Abdujabbarov** | **Date:**  **05.06.2017** |

**Bishkek – 2017**

**INTERNATIONAL ATATURK-ALATOO UNIVERSITY**

**NEW TECHNOLOGIES FACULTY**

**APPLIED MATHEMATICS AND INFORMATICS DEPARTMENT**

# ABSTRACT

**BUILDING OPTICAL TEST GRADING SYSTEM USING IMAGE PROCESSING**

**By Gulbustan Baudunova**

**Diploma Paper**

|  |
| --- |
| The main objective of my diploma thesis is to provide a supportive community for teachers in Bishkek. This thesis describes the development of an optical test grading software that can be used for counting the examination score from the multiple-choice answer sheets. Thesis will be focused on individual approach with completely new product for teachers in local schools, universities and etc. Nowadays there are many teachers, professors generally make their grade calculations of quizzes or test questions by hand, rather than considering on program that will automatically make calculation of sheets. In this program answer sheet will be scanned and the scanned image of the answer sheet will be given as input to the software system. Using Image processing we will find the answers marked to each of the 105 questions by finding the region of interest and applying several algorithm. Optical Test Grading system was implemented using the Matlab programming language.  **Keywords:** Optical Test Grading, Optical Mark Recognition, Matlab |

**Thesis Supervisor:** Marlen Akimaliev **Date:** 05.06.2017

**ЭЛ АРАЛЫК АТАТҮРК-АЛАТОО УНИВЕРСИТЕТИ**

**ЖАҢЫ ТЕХНОЛОГИЯЛАР ФАКУЛЬТЕТИ**

**КОЛДОНМО МАТЕМАТИКА ЖАНА ИНФОРМАТИКА БӨЛҮМҮ**

# АННОТАЦИЯ

**СҮРӨТТҮН ЖАРДАМЫ МЕНЕН БААЛАРДЫН ЖЫЙЫНТЫГЫН ОПТИКАЛЫК СИСТЕМА АРКЫЛУУ ИШТЕП ЧЫГАРУУ**

**Аткарган Гульбустан Баудунова**

**Дипломдук иш**

|  |
| --- |
| Бул дипломдук иштин негизги максаты мобилдик телефондорго “ Андроид түзмөктөрүнө ылайык түзүлгөн англисче-кыргызча, кыргызча-англисче математикалык cөздүк” аттуу сөздүк иштеп чыгуу болуп саналат. Математикалык cөздүктүн идеясы Эл аралык Ататүрк-Алатоо окуу жайынын математика сабагынын терминдерин англис тилинде үйрөнүп жаткан студенттер үчүн.Сөздүктө 2563 төн ашуун математикалык терминдер камтылган. Бул дипломдук иштин өзгөчүлүгү Кыргызстанда эн биринчи жолу англисче-кыргызча, кыргызча-англисче тилдеринде математикалык сөздүк Андроид уюлдук телефондоруна ылайыкташып түзүлгөн түзмөк болугондугунда. Сөздүктүн калыптанышы жана жасалышы Андроид Студио аттуу программанын жардамы аркылуу ишке ашты. Китепте Андроид Студио туурасындагы сөздүк үчүн колдонула турган башкы функциялары жана программалоо ишинин негиздери түшүндүрүлгөн.Андроид операциондук системасына сөздүк жасоого айрым маалыматтарды алууңуздар үчүн китепти баштан-аяк окуп чыгууңуздарга туура келет.  **Ачкычкы сɵздɵр:** Optical Test Grading, Optical Mark Recognition, Matlab |

**Илимий жетекчиси**: Марлен Акималиев **Датасы:**05.06.2017

**МЕЖДУНАРОДНЫЙ УНИВЕРСИТЕТ АТАТЮРК-АЛАТОО**

**ФАКУЛЬТЕТ НОВЫХ ТЕХНОЛОГИЙ**

**ОТДЕЛЕНИЕ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

# АННОТАЦИЯ

**РАЗРАБОТКА ОПТИЧЕСКОЙ СИСТЕМЫ ОЦЕНКИ РЕЗУЛЬТАТОВ, ИСПОЛЬЗУЯ ОБРАБОТКУ ИЗОБРАЖЕНИЙ**

**Выполнила** **Гульбустан Баудунова**

**Дипломная Работа**

|  |
| --- |
| Цель этого проекта заключается в разработке приложения под названием "Английско-Кыргызский и Кыргызско-Английский математический Словарь на платформе Андроид", которое поддерживается Андроид смартфонами. Цель создания математического словаря это, активное использование этого приложения студентами Международного университета Ататюрк-Алатоо, изучающие английские математические термины. Словарь содержит более 2563 математических терминов. Уникальность данной дипломной работы в том, что впервые в Кыргызстане будет развиваться мобильный математический английско-кыргызский, кыргызско-английский словарь для Андроид смартфонов. Цель этого исследования проектирование и строительство словаря в Android Studio, который является одним из важных программ для разработки функций для Андроид и других платформ. Кроме того, в этом проекте есть ссылки и рекомендации для развития Андроид словаря, что делает эту дипломную работу очень ценной и чтобы получить дополнительную информацию вам следует прочесть до конца.  **Ключевые слова:** Optical Test Grading, Optical Mark Recognition, Matlab |

**Научный руководитель:** Марлен Акималиев  **Дата:** 05.06.2017

**YENİ TEKNOLOJİLER FAKÜLTESİ**

**UYGULAMALI MATEMATİK VE BİLGİSAYAR BӦLÜMÜ**

# ӦZET

**BUILDING OPTICAL TEST GRADING SYSTEM USING IMAGE PROCESSING**

**Hazırlayan Gulbustan Baudunova**

**Diploma iş**

|  |
| --- |
| Bu projenin amacı, gerçek bir Android akıllı telefonlar üzerinde çalışabılecek "Android platformu üzerinde İngilizce-Kırgızça ve Kırgızça-İngilizce matematik Sözlük" adındaki bir sözlük uygulaması geliştirmektir. Matimatik sözlüğünün tasavvuru Uluslararası Atatürk-Alatoo Universitesinin Matimatik dersin ingilizce terminin gören öğrenciler için. Bu sözlükte 2563’ten fazla matematik terimi vardır. Bu Tezin farklılığı Kırgızistanta ilk defa Ingilizce-Kırgızca, Kıgızca-Ingilizce dilinde Matemetik sözlüğünde Android cep telefonlarına uygun olarak yapıldığı cıhat olmasındadır. Sözlüğün ortaya çıkması ve yaılışı düzenlenmesi Android Studiyo ile bu programın yardımıyla düzenlendi. Kitapta Android Studiyo hakkında sözlük için kullanılacak olan önemli fonksiyon ve programdır. Bu işin püf noktası olduğunu belirtmektedir. Android operyasyon sistemine sözlük yapmaya bazı malümatları almak için kitabın baştan sona kadar okumanız gerekmektedir.  **Anahtar kelimeler:** Optical Test Grading, Optical Mark Recognition, Matlab |

**Tez Danışmanı:** Marlen Akimaliev **Tarih:** 05.06.2017

# LIST OF ABBREVIATIONS

IDE Integrated Development Environment

OTG Optical Test Grading

OMR Optical Mark Recognition

IPT Image Processing Toolbox

NCC Normalized Cross Correlation

CHT Circle Hough Transform

SHT Standard Hough Transform

# LIST OF FIGURES

Figure 1.1-Most popular Google Play app categories by device installs…………………………..4

Figure 1.2 Comparison of native apps, mobile websites and hybrid app architectures…………….6

Figure 3.11 Interface of apps or Activity Navigation View……………………………………….44

Figure 3.12 Setting Activity interface……………………………………………………………45

Figure 3.13 Setting Activity is settings……………………………………………………………45

Figure 3.14 Interface is Chinese language………………………………………………………...45

Figure 3.15 Share Activity share link of MATH dictionary………………………………………46

Figure 3.16 Share Activity share favorite words to someone……………………………………..47

Figure 3.17 Short Information of dictionary in English-Kyrgyz languages……………………….47

# LIST OF TABLES

Table 1 Device Setup for Development (based on “Using Hardware Devices,” 2012…………..27

Table 2 Hardware and Software details about Notebook………………...………………………..48

Table 3 Hardware and Software details about Android phone…………………………………….51

# TABLE OF CONTENTS

Contents of Table

Cover page.........................................................................................................................................i

[ABSTRACT ii](#_Toc480886532)

[АННОТАЦИЯ iii](#_Toc480886533)

[АННОТАЦИЯ iv](#_Toc480886534)

[ӦZET v](#_Toc480886535)

[LIST OF ABBREVIATIONS vi](#_Toc480886536)

[LIST OF FIGURES vii](#_Toc480886537)

[LIST OF TABLES viii](#_Toc480886538)

[TABLE OF CONTENTS ix](#_Toc480886539)

[ACKNOWLEDGEMENTS xiv](#_Toc480886540)

[INTRODUCTION 1](#_Toc480886541)

[CHAPTER 1 2](#_Toc480886542)

[RELATED WORK 2](#_Toc480886543)

[1.1 Briefly about optical test grading system 2](#_Toc480886544)

[1.2 Optical Test Grading Based Matlab Image Processing Toolbox 3](#_Toc480886545)

[1.2.1 OTG Based Image Segmentation and Thresholding Technique 3](#_Toc480886546)

[1.2.2 OTG Based Image Segmentation and Template Matching Technigue 5](#_Toc480886547)

[1.3 Problem Statement 6](#_Toc480886548)

[1.3.3 The Agile Model 7](#_Toc480886549)

[1.3 Approach 7](#_Toc480886550)

[CHAPTER 2 9](#_Toc480886551)

[ANDROID AND WORKING PRINCIPLE OF THE DICTIONARY 9](#_Toc480886552)

[2.1 ANDROID 9](#_Toc480886553)

[2.1.1 Android SDK 11](#_Toc480886554)

[2.1.2 Android Emulators 12](#_Toc480886555)

[2.1.2.1 AMIDuOS Emulator 13](#_Toc480886556)

[2.1.2.2 Andy Emulator 13](#_Toc480886557)

[2.1.2.3 BlueStacks Emulator 14](#_Toc480886558)

[2.1.2.4 Droid4x Emulator 14](#_Toc480886559)

[2.1.2.5 Nox Emulator 14](#_Toc480886560)

[2.1.2.6 Genymotion Emulator 15](#_Toc480886561)

[2.2 Java language in Android 16](#_Toc480886562)

[2.3 Microsoft SQL Server 18](#_Toc480886563)

[2.4 Android Studio 19](#_Toc480886564)

[2.4.1. Android Environment Setup instructions: 21](#_Toc480886565)

[2.4.2 Set up testing environment. 23](#_Toc480886566)

[2.4.3 Android Application file structure 26](#_Toc480886567)

[2.4.4 Main application components 27](#_Toc480886568)

[2.4.5 Process handling 29](#_Toc480886569)

[2.5 Android Layouts and Types of Android Layouts 31](#_Toc480886570)

[2.5.1 Change the application language settings 33](#_Toc480886571)

[CHAPTER 3 37](#_Toc480886572)

[IMPLEMENTATION AND PROGRAM TESTS 37](#_Toc480886573)

[3.1 Logo of application 37](#_Toc480886574)

[3.2 Main Activity 38](#_Toc480886575)

[3.3 Drawer Layout 41](#_Toc480886576)

[3.3.1 Settings languages 42](#_Toc480886577)

[3.3.2 Share Activity 44](#_Toc480886578)

[3.3.3 About Activity 45](#_Toc480886579)

[3.4 Program Test 46](#_Toc480886580)

[3.4.1 The Editor View Editor Test 46](#_Toc480886581)

[3.4.2 Export to Google Play Market 48](#_Toc480886582)

[3.4.3 Practical Mobile Phone Test 49](#_Toc480886583)

[CONCLUSION 51](#_Toc480886584)

[REFERENCES 52](#_Toc480886585)

[APPENDIX 54](#_Toc480886586)

# ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Marlen Akimaliev for the useful comments, his support, motivation, overall help and engagement through the learning process and by providing me all necessary material of this diploma thesis. Moreover I would like to express the deepest appreciation to Dr. Musa Abdujabbarov for his help and support.

I also must express my sincere heartfelt gratitude to all the staff members, who helped me directly or indirectly during this course of work.

At last, I am profoundly grateful to my family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

May, 2017

**Gulbustan Baudunova**

# INTRODUCTION

With increasing knowledge and technological progress of society, our country requires to provide a supportive community for teachers in local educational institutions. Nowadays, there are no tools to automate grade calculation of quizzes or test questions. I want to develop a software program that makes grading effortless while capturing and reporting with useful data on assessments.

As a student who graduating a New Technology Faculty, I explore an image-processing algorithm for auto grading of answers sheets. Multiple-choice questions are widespread mechanism used by schools and universities to test student performance. They are common on standardized tests. The relevance of the problem is the fact that big organizations use auto Evaluation of OMR software and dedicated scanner for grading multiple choice questions. But small institutes and individual teachers cannot afford this convenient method of grading and have to do time consuming manual grading. So my aim is to develop a suitable software that would detect marks and hence prepare their results according to the needs.

One of the most important usages of program is checking multiple choices question exam students choose the answer by filling square choice on a printed sheet. Then after scanning, these papers will be checked by special software automatically. This article is written based on Matlab algorithm which is included: scanning, preprocess, steps, basic identify, checked steps, and conclusion of different tests.

As a conclusion, developing and implementing most effective and efficient optical test grading system is a critical issue for Kyrgyzstan. May be the most true way is the individual approach with completely new product.

# CHAPTER 1

# RELATED WORK

# 1.1 Briefly about optical test grading system

Optical Test Grading (OTG) is a technique to sense the presence of marks by recognizing their depth on sheet. A mark is a response on the questionnaire sheet that is filled with pencil. The way of marking is simple to everyone and OTG device can process mark information on sheets rapidly. Thus, OTG has been widely used as a direct input device for data of censuses and surveys and is fit for handling discrete data, whose values fall into a limited number of values.

OTG is a technology that detects the absence or presence of a mark, but not the shape of the mark. OTG software interprets the output from the scanner, and translates it into the desired output. Forms are scanned through a scanner. The forms contain small circles, referred to as bubbles, or boxes that are filled in by the respondent. Optical Test Grading (OTG) reads marks written by pencil in the pre-defined positions on the multiple-choice exams sheet. The OTG can judge the existence of written marks by recognizing their darkness on the sheet. [1]

Nowadays in schools, colleges and classes OTG technology is used. Exams are conducted using OTG system because by using this technology the conduction of exam is getting much easier, powerful, and cheap [2].

# 1.2 Data Science

# 1.2.1 Briefly about Data Science

There's no denying that 'data scientist' is a hot job title to have right now, and for good reason. It's a tremendously fun and challenging field to be in, and despite all of the often undeserved hoopla that surrounds it, data scientists are doing some pretty amazing things. So it's no surprise that many people are clamoring to find out how to become data scientists.

    Data science is not simply a series of programs and tutorials that automatically make inferences from your data. Often times, what isn't in your data has significant implications for inference. Your software package isn't going to tell you what they are. Software is just an instrument which will help to achieve some goals. It is like cello, trumpets are make a music, but people are controlling the instruments. In the world we have same experience, there are a lot of software, but all of them are needed a people who will control them and understand what kind of variables enter.

The question facing every company today, every startup that wants to attract a community, is how to use the data effectively says Mike Loukides in his report. He said that, the web is full of “data-driven apps.” Nowadays almost any e-commerce application is a data-driven application. There’s a database behind a web front end, and middleware that talks to a number of other databases and data services. But merely using data isn’t really what we mean by “data science.” A data application acquires its value from the data itself, and creates more data as a result. It’s not just an application with data; it’s a data product. Data science enables the creation of data products. Most successful data used companies are Google, Facebook, LinkedIn, etc. Google have biggest amount of data and it found the way to use it for instance spell checking algorithm makes our search easier. Facebook suggests our people that we know, sometimes it becomes frightening accurate. LinkedIn manage our professional identity, build and engage with your professional network, access knowledge, insights and opportunities.

# 1.2.1 Types of Data Analysis

Big data streaming analysis nowadays has become one of the most important topic in the list of data analysts since enormous amount of data are produced daily by the numerous smart devices. The analysis of such data is very important and the detection of frequent or even non-frequent patterns can be critical for many aspects of our lives. Data analysis is not only sitting in front of the huge amount of data. In an Interview Study about Enterprise Data Analysis and Visualization data analysts was divided into 3 archetypes: hackers, scripters, and application user.

Hackers are the most proficient programmers of the three groups and they feel themselves most comfortable with manipulating data. In addition to working with an analysis package like R or Matlab, they frequently used a scripting language like Python, and Perl. Because of their skill set, hackers often completed flexible workloads without assistance from coworkers such as IT staff. Because of their knowledge of query languages such as SQL or Pig, they could also typically run jobs at scale on their own. One data analyst said that he is not a DBA, but he is good at SQL. He is not a programmer, but he is good at programming. He is not a statistician but he is good at applying statistical techniques. According to the words hacker data analyst are the combination of programmer, high level skilled programmer and statistician. Because of high level skills in programming, they reported that working with larger data sets limited the types of statistical routines they could run on the data. Also, because this group relied less on IT staff for completing certain tasks, they spent more time in early-stage analytic activities prior to modeling.

Scripters performed most of their analysis within a software package such as R or Matlab. They were able to perform simple manipulations such as filtering and aggregating data, but typically could not perform custom operations such as parsing log files or scraping data off the web. They generally operated on data that had been pulled from the data warehouse by IT staff and stored in an expected format. Some of these analysts could write simple SQL queries to pull data into their analytic tool of choice. In some cases, they were comfortable writing scripts in a scripting language, but typically do not know how to create scripts that run at scale. Scripters often produced visualizations using the statistical package during exploratory analysis. In some cases scripters used a separate tool, such as Tableau, to create interactive dashboards for reporting after the significant insights had been discovered.

    The last set of analysts performed almost all operations in a spreadsheet or other dedicated analysis applications like SPSS, Minitab, Excel and etc. Application users typically worked on smaller data sets than the other groups and generally did not export data from the spreadsheet except for building reports. In some cases, advanced application users wrote scripts using an embedded language such as Visual Basic. To produce visualizations they typically created charts in Excel or exported data to a reporting tool such as Crystal Reports.

# 1.2 Optical Test Grading Based Matlab Image Processing Toolbox

OTG is an optical test grading system which describes the development of an optical test grading software that can be used for counting the examination score from the multiple-choice answer sheets. OTG will be focused on individual approach with completely new product for teachers in local schools, universities and etc. This kind of system will automatically make calculation of sheets. Unfortunately most of teachers, professors in Bishkek generally make their grade calculations of quizzes or test questions by hand. It allows them to take changes without much cost and efficient.

Below subsections provide a survey of the literature related to Optical Test Grading system. That were developed to improve Optical Mark Reader. Below lets discuss about related works about Optical Mark Recognition system.

# 1.2.1 OMR Based Image Segmentation and Thresholding Technique

Optical Mark Reader (OMR), also called “mark sensing”, is a method of scanning technology in which data is input through marks made in predefined positions on a form and entering data into a computer system. Therefore, OMR is best for handling  
discrete data, where values fall into a limited number of values. For examples, gender, occupation, religion, etc.

The OMR scanners began to be used excessively in the early 1950’s, which used a series of sensing brushes in detecting graphite particles on a document that is passed through the machine []. Image-based OMR studies by Chinnasarn et al. presented PC-type microcomputer and image scanner. The system operation could be distinguished into two modes: learning mode and recognition mode. Data extraction from each area can be performed based on the horizontal and vertical projections. For the purpose of checking answer, the number of black pixels in each answer block is counted, and the difference between those numbers in the input and its corresponding model is used as decision criterion. This was a transition between punch cards and barcodes.

Chinnasarn et al, presented a system which was based on Personal Computer-type microcontroller and image scanner. The system operations can be distinguished in two modes: learning mode and operation mode. The data extraction from each area can be performed based on the horizontal and vertical projections of the histogram. For the answer checking purpose, the number of black pixels in each answer block is counted, and the difference of those numbers between the input and its corresponding model is used as a decision criterion. This is the first image-based OMR technique [].

Andrea Spadaccini described JECT-OMR, a system that analyses digital images representing scans of multiple-choice tests compiled by students. The system performs a structural analysis of the document in order to get the chosen answer for each question, and it also contains a bar-code decoder, used for the identification for additional information encoded in the document. JECT-OMR was implemented using the Python programming language, and leverages the power of the Gamera framework in order to accomplish its task. [].

Tien Dzung Nguyen et al. proposes grading multiple choice test which is based on a camera with reliability and efficiency. The bounds of the answer sheet image captured by the camera are first allocated using Hough transform and then skew-corrected into the proper orientation, followed by the normalization to a given size. The tick mark corresponding to the answer for each question can be recognized by allocation of the mask which wraps the answer area [].

Nutchanat Sattayakawee proposes the algorithm of test scoring for grid answer sheets. The method used is based on projection profile and thresholding techniques [].

Rakesh S et al. proposed system consists of an ordinary printer, scanner and a computer to perform computation and is assisted with a graphical user interface. Users can design forms of their choice and use it for survey or other related activities. The filled forms are scanned and scanned images are given as input to a computer, which does the computation and stores the result in a user understandable spreadsheet. The system is independent of hardware and system platform, thereby making it platform independent [9].

A. AL-Marake by presents a low cost and fast solution for optical mark recognition system working in multi-core processor system. The answer sheet is captured using a digital camera and the image is processed. Initially the borders of the sheet are located then the bubbles are detected. Fast techniques are used to detect the bubbles without a rotation correction. An adaptive binarization has been used to overcome the lighting effects of the camera based images [10].

# 1.2.2 OTG Based Image Segmentation and Template Matching Technique

Francisco de Assis Zampirolli et al. presents a simple and innovative method to transform captured images of answer sheets into reduced binary matrices containing answers to the questions plus some control elements, using simple morphological operations for segmentation [].

Azman Talib et al. proposes shape-based vision algorithm, a hierarchical template-matching approach that implemented in this system to verify the imaging and inspecting the correct answer of the Optical Mark Recognition (OMR) sheet form. An OMR answer sheet scheme with all correct answers are marked on the paper and will be used as a template for object recognition during the matching process. Region of interest (ROI) is selected and filtered into grey level to extract the contour of the object. The image is then pre-processed and trained using image processing technique. A low-cost 1.3 MP web camera is used to acquire the marked OMR, image for all questions together with the sequence number; this is to ensure the system can distinguish between different questions having the same answer [].

Ms.Sumitra B. Gaikwad aims to develop Image processing based Optical Mark Recognition sheet scanning system. Find that lot of competitive exams are being conducted as entrance exams. These exams consist of MCQs. The students have to fill the right box or circle in the appropriate answer to the respective questions. During the inspection or examining phase normally a stencil is provided to the examiner to determine the right answer to the questions. This is a manual process and a lot of errors can occur in the manual process such as counting mistake and many more. To avoid this mistakes OMR system is used. In this system OMR answer sheet will be scanned and the scanned image of the answer sheet will be given as input to the software system. Using Image processing [].

Lopresti et al explained the process of optical mark recognition with reference to Remark Office OMR 3.0, made by Principia Products. Also he reported that, for years people who do statistical analysis have been designing questionnaires, and getting them filled out by respondents or interviewers, and then somehow wrestling the data into a computer. With the first computers, much of the data was input by creating decks of punched cards. While this process allowed one to create those necessary computer files, it was subject to input error and thus had to be verified. At the very same time also it allows multiple-choice forms to be read.[ijarece]

Kia mentioned that Optical Mark Recognition (OMR) is used for standardization testing as well as course enrollment and attendance in education [ijarece].

Dillman studied the impact of OMR forms on which it responded it to be relevant issue. One possible disadvantage with OMR surveys is that they may suppress response rates. OMR surveys are often combined with other cost-cutting measures, so their low response rates may simply be an artifact of other choices about survey administration. Generally OMR forms have one standard ink color that provides limited visual appeal creating a disincentive for their response. Moreover these forms are also more tedious to fill out. Rather than simply reading through the survey and checking off or circling responses, the respondent must carefully fill in a circle or “bubble” for each question answer[ijarece]

# 1.3 Problem Statement

This thesis devoted to develop Image processing Optical Test Grading system. Today multiple-choice questions have become an integral part of the educational system. Standardized tests also use in Bishkek private schools and universities to judge students’ academic performance. The students have to fill the right box or circle for the appropriate answer to the respective questions. During the inspection or examining phase normally an examiner spend one week to determine the right answer to the questions. This is a manual process and a lot of errors can occur in the manual process such as counting mistake and many more. Actually in Bishkek educational institutions answer sheets are checked by optical machine, but checkup process takes minimum three weeks. To avoid this mistakes OTG system is used.

# 1.3.1 Objectives

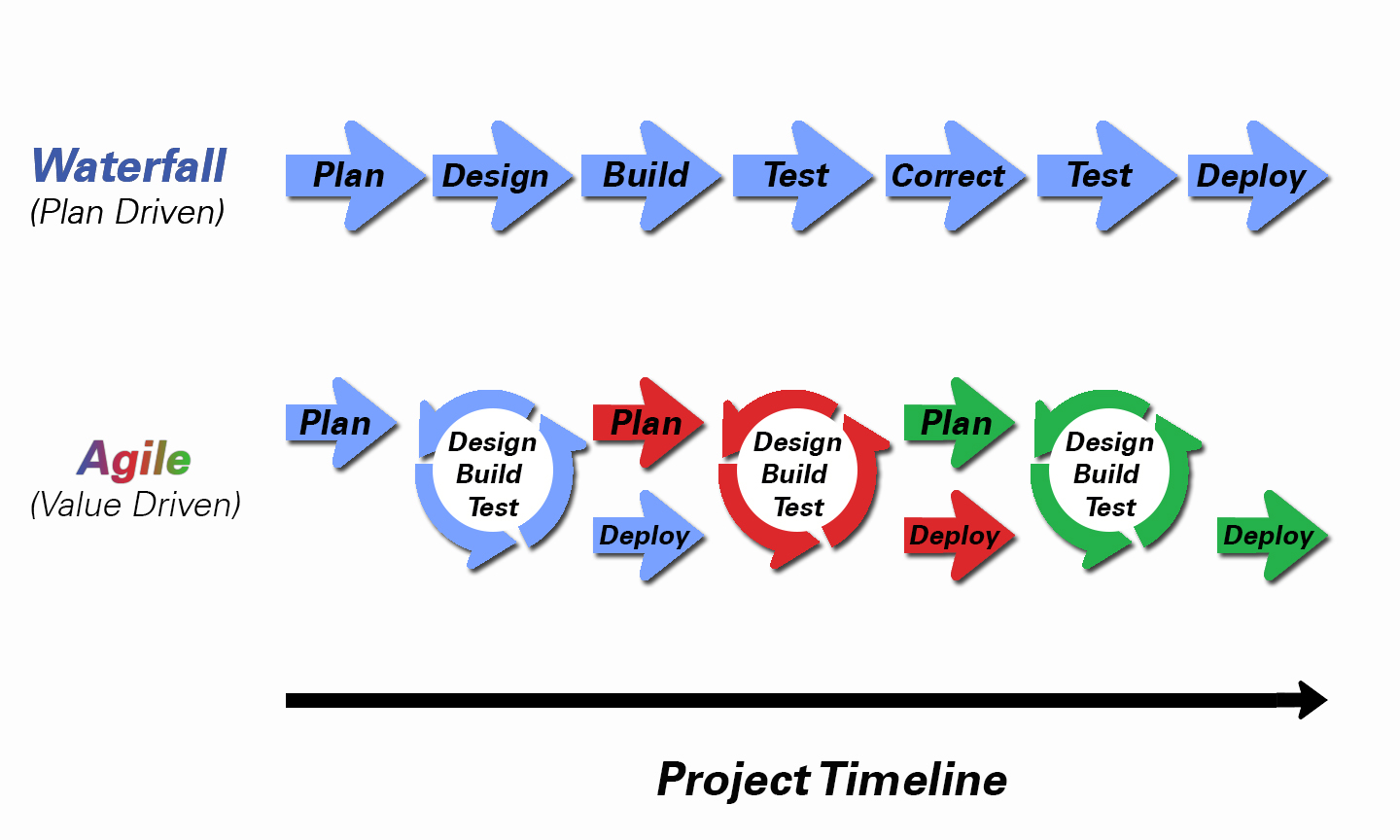
* First, out project aim is to make a supportive community service for staff members in local educational institutions.
* Second, Optical Test Grading system makes grading effortless while capturing, storing, and reporting with useful data on assessments, no need to wait for the optical machine or manual check to grade papers quickly. The user can only make marks and cannot write any information.
* Third, this application can be implemented at institute’s level, which will bring a tremendous change in the trend of Educational system management.
* Fourth, another main objective of OTG system is to make the process of grading more speedly, more accurate and cost efficiency.

Teachers can have all of their multiple-choice grading done before the next class starts. Using Image processing we will find the answers marked to each of the 105 questions by finding the region of interest and applying template matching, circular Hough transform and etc.

# 1.4 Approach

Agile model believes that every project needs to be handled differently and the existing methods need to be with considering to best suit the project requirements. In Agile the tasks are divided to time boxes to deliver specific features for a release. Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer. What is Agile???This project is done using agile methodology, let us refer to the official definition from web-source.

Agile modeling is a methodology, which makes use of practice for modeling and documentation of software based systems. Traditional modeling methods in software development projects have given way to these practices which are applied in a more flexible manner. It supplements the different agile methodologies like extreme programming, agile unified process and scrum models.



*Figure 1: Waterfall and Agile methods*

Why Agile?

From picture above we can see that agile are more flexible than waterfall, therefore agile methods gives following advantages. Agile methodology allows you to make changes, easy to add new features to requirements that can be incorporated at any point of the process. Testing at the end of every sprint ensures that bugs are caught. The active involvement, cooperation and collaboration make agile development a much more enjoyable for most people. Instead of big specs, we can discuss requirements in workshops. Instead of lengthy status reports, we can collaborate around a task-board discussing progress. Instead of long project plans and change management committees, we can discuss what’s right for the product, etc. In my experience this makes it a much more rewarding approach for everyone. In future work OTG system will work with the help of android platform, that’s why we choose agile methodology in our system. To be always with customer everywhere we decide to make android application.

* Users can access from everywhere
* With or without internet access
* Makes bug fix easier
* Platform independent
* Free

# 1.5 Related work by other researches

Finding suitable software for staff members in Bishkek educational institutes can be not difficult, because there are few of mobile applications in Google store. But they are not open source software’s. Let us discuss about one of them called Zipgrade.

Zipgrade is a grading application founded by John Viebach. This application allows teachers to easily and quickly grade multiple choice questions by simply taking a picture of it with your iPhone or iPad. However you cannot simplify download and deploy this app, because it not free and you will need purchase it.

Another similar grading application, which is also gained popularity is Quick Key grading application. Let’s discuss the prehistory of this app.

Quick Key is a suite of mobile teacher tools created by Walter Duncan, a 15-year classroom teacher. Using daily short quizzes, graded instantly, Walter was able to identify students and subjects that needed to be re-taught before moving on to the next lesson. So Walter teamed up with Isaac D. Van Wesep, to build a prototype of Quick Key, a teacher tool for your phone that scans and grades quizzes and tests anywhere, with or without the Internet, and sends the results to your school's electronic grade book. Soon after so many works, the prototype of application was bring to the market. The first version of Quick Key mobile teacher tool launched on the Apple App Store in September 2013. Quick Key is perfect for quizzes, unit tests, midterms and finals. But like Zipgrade, you cannot simply download this app, you will need to customize it, in any case there are a lot of grading application and all of them are paid.

# 1.6 Thesis Outline

This application was implemented on the Matlab platform. The rest of this thesis is laid out as follows. Chapter 2 is theoretical part, to afford you some content and explain the specific problems which were set out to solve, and describe the algorithm of methods that have been used. Chapter 3 considers methodology in development and software development process. Chapter 4 below describes the algorithm of the Optical Test Grading system algorithm and result of the project. The captured images are passed into stitching pipeline which applies various image processing techniques such as feature detection, feature matching, and image warping to rotate, translate, align and merge the capture portions into one single image. Some post processing is then often to ensure seamless joints and uniform intensity level across the document.

# CHAPTER 2

# THEORY AND METHODOLOGY OF OTG SYSTEM

In this section, we propose the novel technique of Optical Test Grading system. These have been explained in detail in the following sub-sections.

# 2.1 Why technology is important?

The world is moving towards technology at a breakneck pace. In 2017 the IT is dynamic more than anything else. There is no universal technology to build a software, every programming language has its strengths and weakness in a specific area. Before coding, we must think! Wrong choice of technology may lead to slower development and budget overrun. The right choice will help to make high quality product.

# 2.2 Image processing in Future?

What if we are in a foreign country where we do not speak the language and do not know their language? Or while driving on the highway, you lose your way and can't understand the street signs? Many tourists and travelers find themselves in similar situations all the time. As Peter Linsley, product manager of Google image search, says Google could come to the rescue again. Instead of typing text, simply take a picture from your mobile phone, do a search by image and the street signs or whatever text is in it will be translated for you. Awesome!

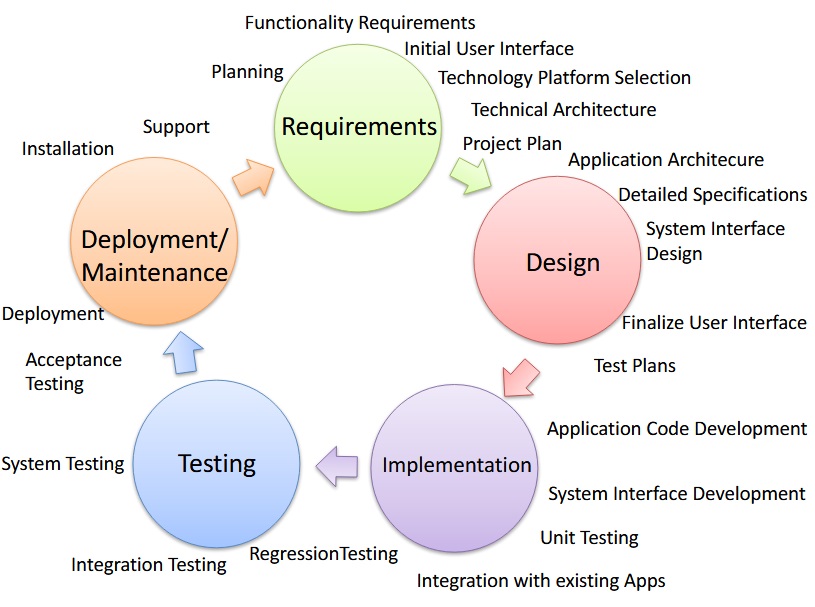
It will changes behavior much like the way Internet text search changed our lives. And as we increasingly replace traditional text with images, 3-D images and real time video in the next few years. Simply put, image processing is a technical analysis of the complex aspects of an image. The field occupies its power from the many recent advances in artificial intelligence and machine learning. "With image processing today, we are going beyond the two-dimensional and going deeper to see what is actually in the image. It is real computer vision," says Peter Linsley.

# 2.3 System Development life cycle

A software development lifecycle is a structure imposed on the development of a software product. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process.

The main reason why we made Optical system is the usability, it is more pleasant and more comfortable to use system, for all educational staff in Bishkek. Benefits are the accuracy, cost efficiency and speed.

Most of the programs is typically done in a prescribed fashion. The following phases were undergone to develop OTG system: System requirements, system design, implementation, testing and maintenance. Shown in Figure 2.



*Figure 2: Software Development life cycle*

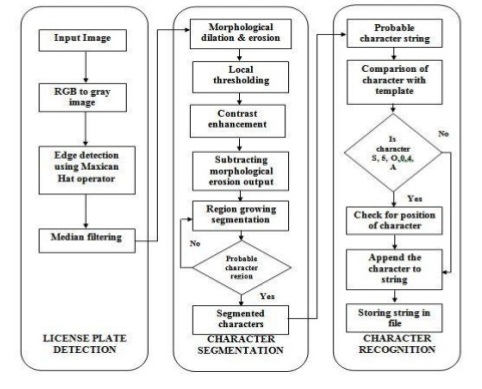
# 2.4 System Requirements and analysis

System requirement phase will explore and define problems, actually any software system is supposed to facilitate some process which was made with another approach. Thus we need to define system problems in educational institutes to make a system which will later work using android platform. But it is another history, another project. Optical Test Grading system will automate these processes, problems are the following: paper based quizzes, test, and exams. Second thing is we need to determine system requirements like: Who is going to use the system? How? These general questions that get answered during system requirement phase. During investigation in educational institutes we have found a ways to solve problems listed above. To make system completely automated for paper based answer sheets we will need to find the best method which will make bubble recognition accurately.

The purpose of the Optical Test Grading system analysis is to collect issues, to determine what to do with existing problem and what a system must do to solve these issues. These is three solutions to our problem ,they are as following do nothing with existing problem and continue use previous applications, modify existing system or develop new one. In Bishkek there are not ready system to modify therefore we will to create a new system, a base of mobile application. The main objective of the project thesis is to provide a supportive community for teachers in Bishkek.

# 2.5 System Design

The result of this phase is a technical design that specifies following issues inputs, outputs. The system can be deployed on any operating system since whole system will be done in Matlab (platform independent).In the design stage, the programming language and the hardware and software platform in which the new system will run are also decided. This system is programmed using Matlab Image processing toolbox. Here is algorithm sequence process (Figure 3).

****

*Figure 3: Algorithm sequence process*

# 2.6 Implementation

Project implementation is the phase where visions and plans are turned into reality. This is the logical conclusion, after evaluating, deciding, visioning, planning, applying for funds and finding the financial resources of a project. Technical implementation is one part of executing a project.

|  |  |
| --- | --- |
| **FUNCTION NAME** | **FUNCTION DESCRIPTION** |
| **imread** | Read image from graphics file |
| **imshow** | Display image |
| **Medfilt2** | 2-D median filtering |
| **edge** | Find edges in grayscale image |
| **graytresh** | Computes a global threshold |
| **Im2bw** | Converts grayscale to the binary |
| **strel** | Represents a flat morphological structuring element |
| **imclose** | Morphologically close image |
| **Imfill** | Fill image region and holes |
| **Bwareaopen** | Remove small objects from binary image |
| **Bwlabel** | Label connected components in 2-D binary image |
| **Bwboundaries** | Trace region boundaries in binary image |
| **Regionprops** | Measure properties of image regions |
| **Imrotate** | Rotate grayscale image |
| **Imcrop** | Crop image |
| **Rgb2gray** | Convert RGB image or color map to grayscale |

*Table 1: Matlab functions*

The Matlab functions that we selected (shown in Table 2) from the Image Processing Toolbox (IPT) execute very frequently used media processing operations, from image enhancement, restoration to image transforms, filtering, image deblurring, analysis, morphology, etc. These functions serve as building blocks to many other algorithms and thus have a strong impact in the scientific computing community.

# 2.7 Perform Image processing, analysis, and algorithm development

Image processing toolbox provides a comprehensive set of reference-standard algorithms and workflow aps for image processing, analysis, visualization, and algorithm development. We can perform image segmentation, image enhancement, noise reduction, geometric transformations, image registration, and 3D image processing.

Image processing toolbox apps let you automate common image processing workflows. We can interactively segment image data, compare image registration techniques and batch-process large data sets. Visualization functions and apps let you explore images, 3D volumes, and videos, adjust contrast, create histograms, and manipulate regions of interest (ROIs).

We can accelerate our algorithms by running them on multicore processors and GPUs. Many toolbox functions support C/C ++ code generation for desktop prototyping and embedded vision system deployment. [1]

# 2.7 Image preprocessing

This is the step taken before the "major" image processing task. The problem here is to perform some basic tasks in order to render the resulting image more suitable for the job to follow. In this case it may involve enhancing the contrast, removing noise, or identifying regions likely to contain the postcode.

In the next subsections we will discuss the methodology of system algorithm, but later we will discuss about image preprocessing in detail.

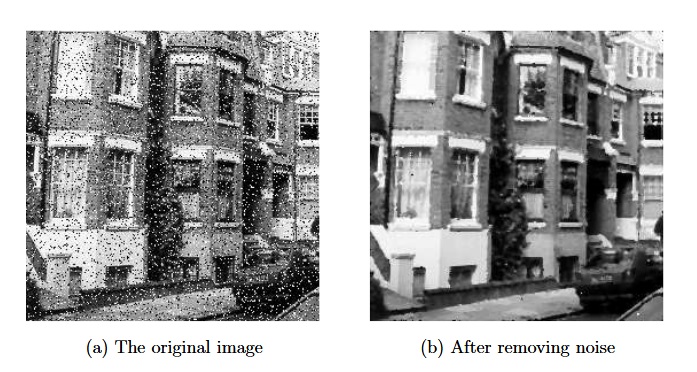
# 2.7.1 Image Enhancement

Image Enhancement techniques in Image Processing Toolbox enable you to increase the signal-to-noise ratio and accentuate image features by modifying the colors or intensities of an image. [1]

This relates to processing an image so that the result is more appropriate for an application. Example include:

* Sharpening or de-blurring an out of focus image.
* Highlighting edges
* Improving image contrast or brightening an image
* Removing noise

Removing noise from an image: noise being random errors in the image. An example is given in Figure 4.Noise is a very common problem in data transmission: all sorts of electronic components may affect data passing through them, and the results may be undesirable. Each type of noise requiring a different method of removal.



*Figure 4: Removing noise from an image*

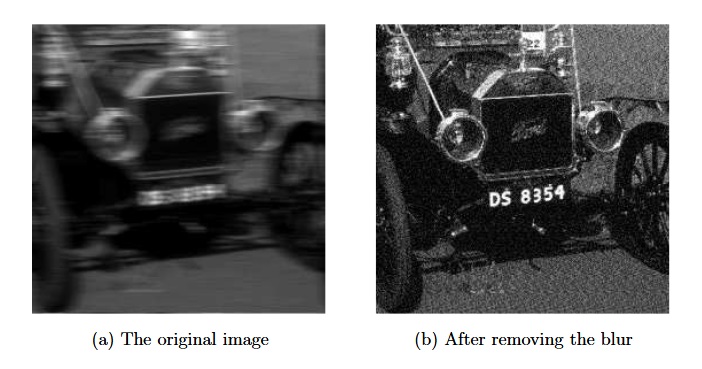
# 2.7.2 Morphological Operations

Morphological operators enable us to enhance contrast, thin regions, or perform skeletonization on regions. Toolbox includes a comprehensive set of morphological operations. [1]

Dilation and erosion are two common morphological operations used in various computer vision and image processing tasks such as blob analysis and text detection. Both dilation and erosion are binary operations. They take an image and a binary kernel matrix as inputs and produce an image as the output. Dilation can be defined as follows: pixel (x, y) of the output image is the maximum pixel value of the neighborhood defined by the kernel centered at (x, y) of the input image. Similarly erosion takes the minimum of the neighborhood.

# 2.7.3 Image Deblurring

Image deblurring algorithms in Image Processing Toolbox include blind, and regularized filter convolution, as well as conversions between point spread and optical transfer functions. These functions help correct blurring caused by out-of-focus optics, movement by the camera or the subject during image capture, atmospheric conditions, short exposure time, and other factors. An example is shown in Figure 5.



*Figure 5: Image deblurring*

# 2.8 Image Analysis

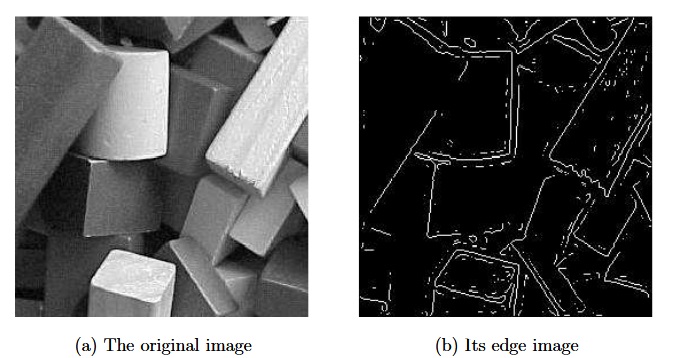
# 2.8.1 Image Region Analysis

We can calculate the properties of the regions in images, such as area, centroid, bounding box, and orientation. Use the Image region analysis application to automatically count, sort, and remove regions based on properties. [1]

# 2.8.2 Edge Detection

Edge detection algorithms let you identify object boundaries in an image. These algorithms include the Sobel, Prewitt, Roberts, Canny, and Laplacian of Gaussian methods. The Canny method can detect true weak edges without being fooled by noise. [1]

Obtaining the edges of an image. This may be necessary for the measurement of objects, an example is shown in Figure 6.Once we have the edges we can measure their spread, and the area contained within them. We can also use edge detection algorithms as a first step in edge enhancement.



*Figure 6: Finding edges in an image*

# 2.8.3 Hough Transform

The Hough transform is designed to identify lines and curves within an image. [1]Using the Hough transform we can:

* Find line segments and endpoints
* Measure angles
* Find circles based on size

# 2.9 Types of images

We shall consider four types of images in image processing:

1. BINARY. Each pixel is just black or white. Since there are only two possible values for each pixel, we only need one bit per pixel. Such images can therefore be very efficient in terms of storage. Images for which a binary representation may be suitable include text (printed or handwriting), fingerprints, or architectural plans.
2. GRASCALE. Each pixel is a shade of grey, normally from 0 to 255, This Range means that each pixel can be represented by eight bits, or exactly one byte. This is a very natural range for image file handling. Other grayscale ranges are used, but generally they are a power of 2. Such images arise in medicine (X-rays), images of printed works, and indeed 256 different grey levels is sufficient for the recognition of most natural objects.
3. RGB. Here each pixel has a particular color, that color being described by the amount of red, green and blue in it. If each of these components has a range 0-255, this gives a total of 253^3=16777216 different possible colors in the image. This is enough colors for any image. Since the total number of bits required for each pixel is 24, such images are also called 24-bit color images. Such an image may be considered as consisting of a stack of three matrices, representing the red, green and blue values for each pixel. This means that for every pixel there correspond three values.
4. INDEXED. Most color images only have a small subset of the more than sixteen million possible colors. For convenience of storage and le handling, the image has an associated color map, or color palette, which is simply a list of all the colors used in that image. Each pixel has a value which does not give its color (as for an RGB image), but an index to the color in the map. It is convenient if an image has 256 color or less, for then the index values will only require one byte each to store. Some image file formats (for example, CompuServe GIF), allow only 256 colors or fewer in an image.

# 2.10 Testing

For testing we did not use any automated frameworks. All testing was done by using Matlab Run icon. View the commands available for running the function by clicking Run on the Editor tab. The command at the top of the list is the command that the Editor uses by default when we click the Run icon.

# 2.11 Deployment

The purpose of deployment phase is to run project in production mode in real life. This ensure that the product was tested successfully and ready to run in production, but it needs android platform. We have another project, which is only works on QR code detection on android platform. In our plans will join two project together and run in production mode in real life.

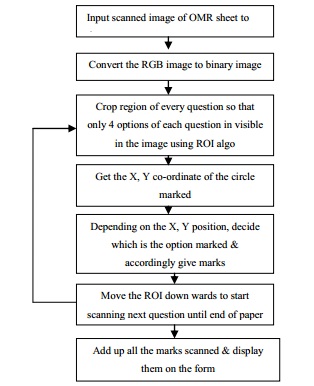
# CHAPTER 3

# ALGORITHM AND OVERVIEW OF THE SYSTEM

As described in the previous section, after taking a picture of answer sheet, our main algorithm is applied to the picture to extract the answer location. This section aims to describe system algorithm details.

# 3.1 The recognition system algorithm

The success of Optical Test Grading recognition system needs combination of various techniques and algorithms, each of which performs a specific task for achieving the main goal of the system. Thus the combination of related algorithms improves the accuracy or the recognition rate of such applications. Figure 7 shows the overview of the proposed system. Image processing of OTG system contains two phases: pre-processing and post-processing phases. Pre-processing prepares how to computer understand an information in each answer form. The other phase is post-processing that deals how to retrieve score from the input sheet.



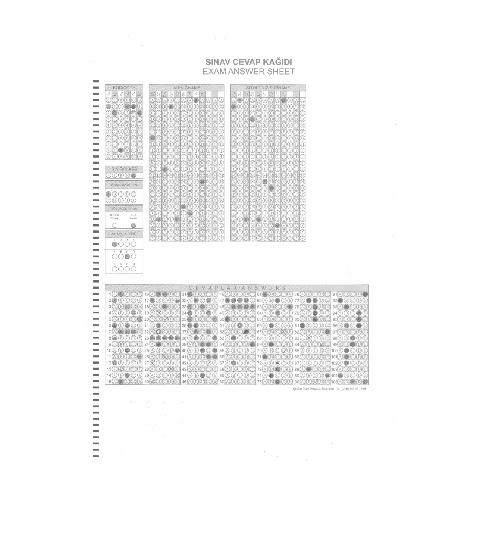
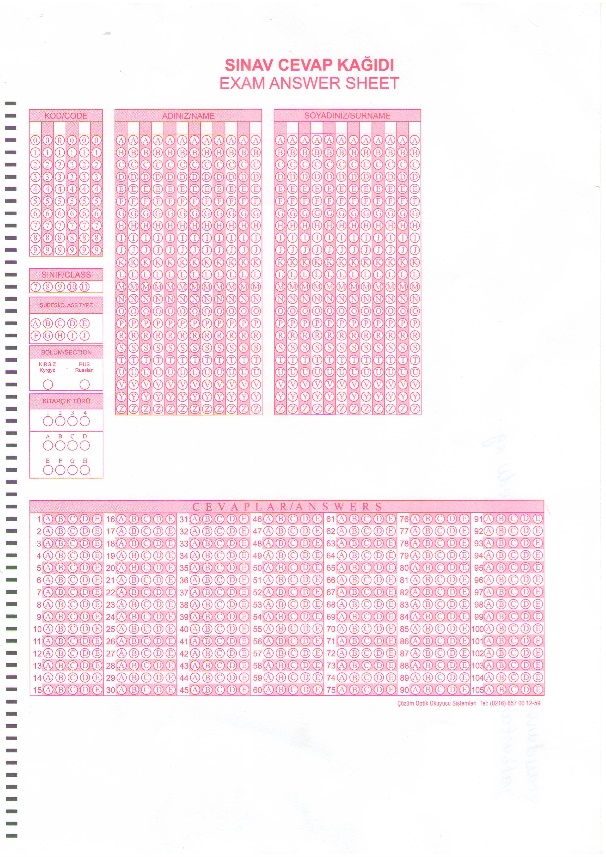
*Figure 7: Proposed System*

# 3.2 Pre-processing phase

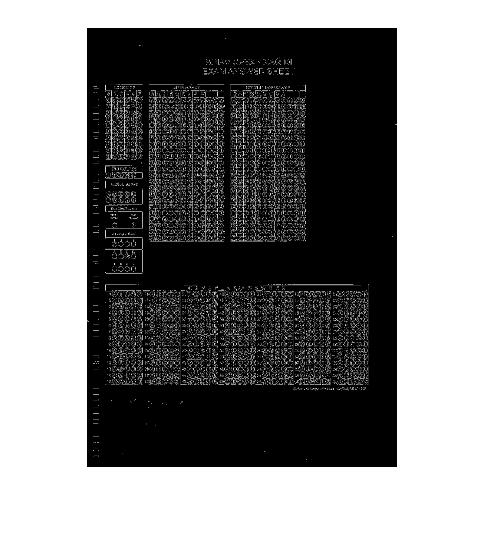
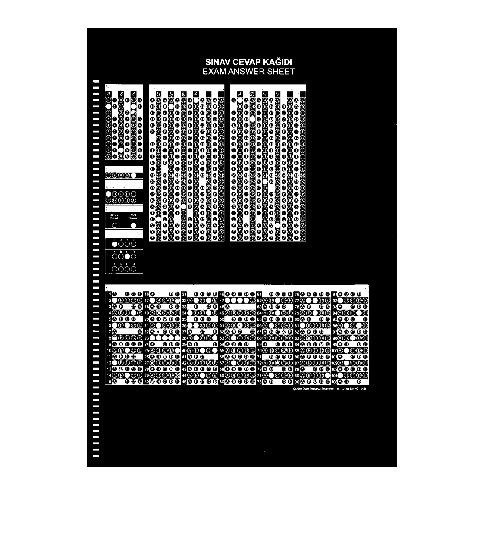
The pre-processing phase consists a set of operations that make the scanned image more suitable for the further phases.

An original empty answer sheet form is entered to the system (Figure 8). The first operation performed to the image is the conversion to grayscale (Figure 9); then the image is converted into binary format, we employ Otsu`s method within each window at each step to obtain the local optimal threshold, which is Camera`s default binarization method (Figure 10). Next, we binarize the grayscale image using filtering followed by 2-D median filtering (Figure 11). For the rank filtering approach, we use a 2-D median filter to reduce “salt and pepper” noise. From experiment, median filtering is more effective than convolution when the goal is to simultaneously reduce noise and preserve edges. The resulting background is then subtracted from the grayscale for global thresholding using Otsu's Method.

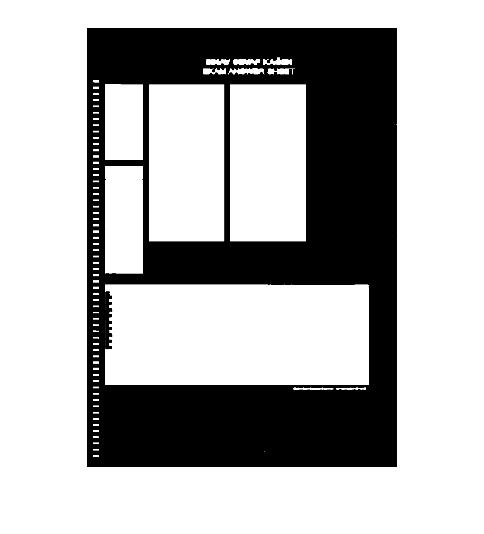
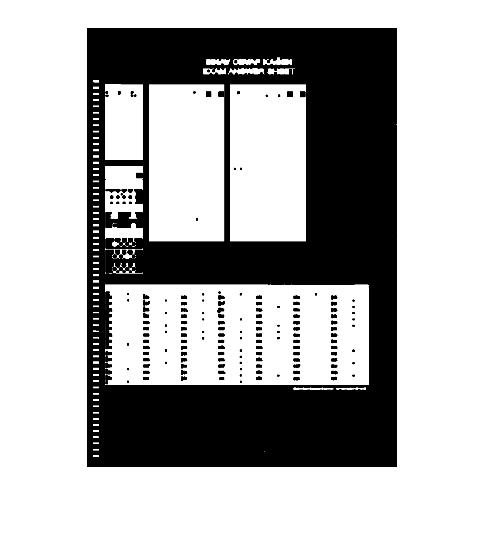
To improve image quality for line detection, we perform morphological operations to estimate the background, and create a copy of the binary image with the detected background subtracted. We use a *strel* method that represents a flat morphological *structuring element*, which is an essential part of morphological dilation and erosion operations. SE = strel('disk',R,N) creates a disk-shaped structuring element, where R specifies the radius. N specifies the number of line structuring elements used to approximate the disk shape. Morphological operations using disk approximations run much faster when the structuring element uses approximations (Figure 12).Then, regions whose areas span less than 2000 pixels are removed from the background-subtracted binary image to generate the improved binary image (Figure 13).



*Figure 8: Original Captured RGB Image Figure 9: Grayscale Image*

**

*Figure 10: Binary Image Figure 11: Filtered Image*

**

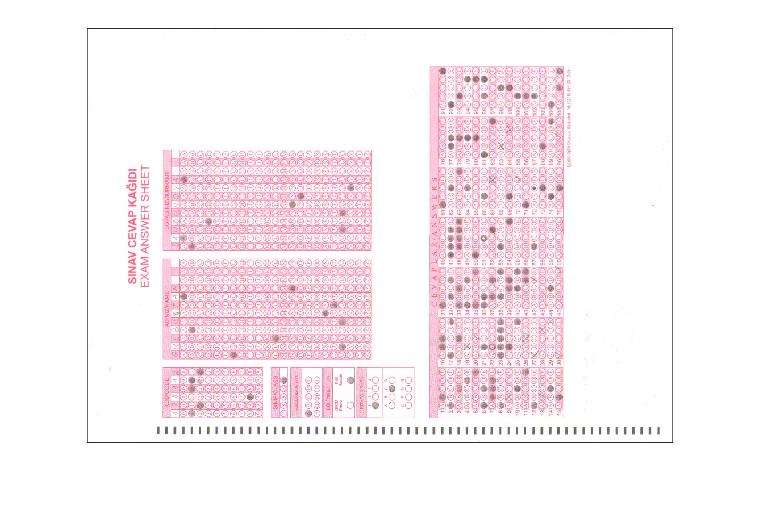
*Figure 12: Erosion and dilation of the Image Figure 13: Filled holes of the Image*

# 3.2.1 Borders extraction

In the design of the answer sheet, borders and lines have been added to facilitate the bubble detection process. The design of the template with a thick line borders solves many problems and increase the reliability and speed. These lines are insensitive to noise and help us to solve the rotation and perspective distortion problems. There are many line detection algorithms with different complexities and robustness. Hough transform is robust for noise and occlusion, but the calculation and/or memory costs are very high. Also digitizing and quantization errors sometimes influence to the accumulation of the peaks in the parameter space. Projection is the fastest way in finding horizontal and vertical lines in an assigned image, because such lines will produce peaks in projection profiles. This method doesn't work with rotated lines or slightly curved line. Line tracking algorithm has been used, with the addition of many heuristics to increase the speed and robustness. Without using thinning or morphological processing, the thick line is tracked using edge tracking. If the line is broken a connection algorithm is used to connect both segments of the line. The detection of each line in the border (Right, Left, Top, Bottom, Etc.) is assigned to a single core in the multi-core processor. There are no dependencies between different lines detection and the processed are parallelized easily.

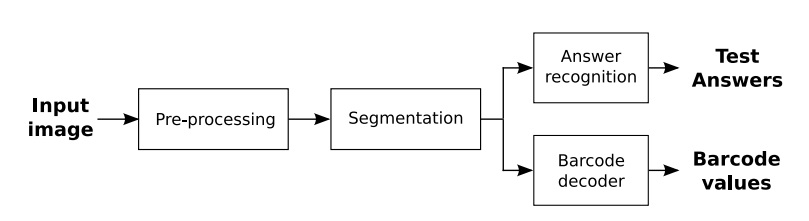
# 3.2.2 Rotation of image

Before the actual note recognition task, it is essential to rotate the answer sheet to an upright position. Next the system does a compensation of rotation effects induced by the ‘imrotate’ operation. The goal of this step is rotate of image answer sheet at a calculated angle to restore it to its normal rectangle. To do that, at first we must calculate the correct angle by using Hough transform method, and then apply bilinear interpolation method with correct angle to rotate all image answer sheet pixels to normal location. Figure 14 shows image answer sheet before rotation operation.



*Figure 14: Rotate Image answer sheet*

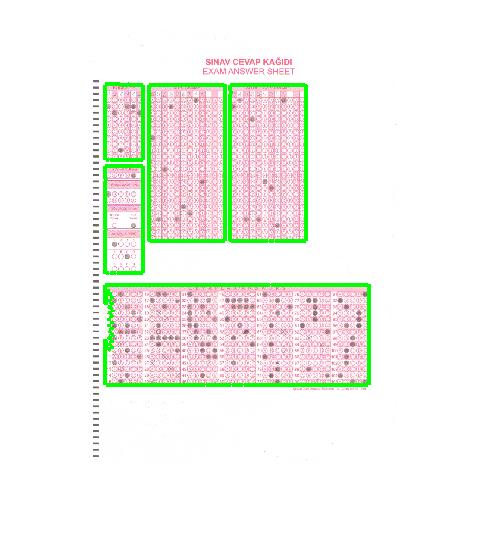
Full block diagram of Optical Test Grading android application is shown in Figure 15



*Figure 15: Block diagram of the application.*

# 3.2.3 Label each connected component

Each form contains 5 rectangular fields: Student Name area, Student Surname area, Answer Area and unused areas. By using ‘bwlabel’ and ‘bwboundaries’ functions we cut the sheet into sections (Figure 16).

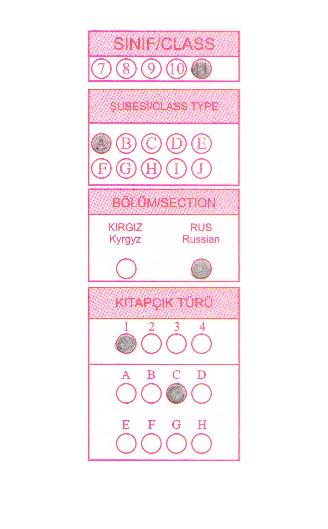
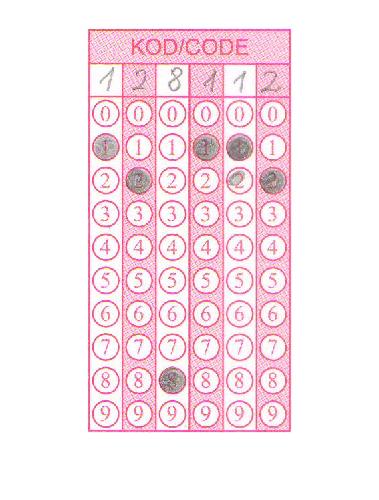


*Figure 16: Label each component of the Image*

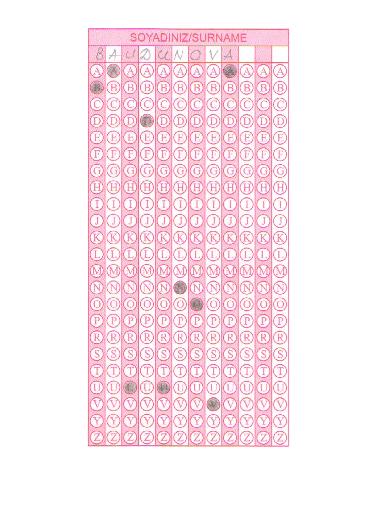
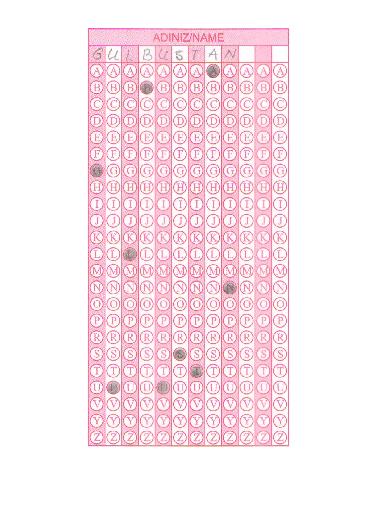
Following figures shows that areas of answer sheet image.

# 

*Figure 17: Answer key area*

** 

*Figure 18: Student Information Area Figure 19: Student Code Area*



*Figure 20: Student Name Area Figure 21: Student Surname Area*

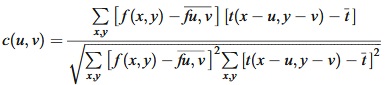
The transformed original image is cropped with each of these 5 bounding boxes to extract the date and columns of numbers. The extracted images are once again binarized using the same functions that was used in stage 1.

# 3.3 Post-processing phase

In the following paragraph the methods of answer recognition will be described, and its basic blocks will be analyzed in detail.

# 3.3.1 Template matching method

Template matching is a method for identifying features in a source image that match a smaller sub-image called the template image [Automated inspection of PCB components using a genetic algorithm template-matching approach]. It is commonly used in object-recognition applications. The basic template-matching algorithm involves sliding the template image over the source image and at each position calculating a grey-scale correlation measure using pixel intensities to estimate the degree of similarity between the template and source image region. Typically, the normalized cross correlation (NCC) is used in template-matching algorithms and is given by:



*Figure 22: Normalized Cross Correlation*

where

*f(x,y)* is the matrix of grey-level pixel intensities in the source image.

*fu,v* is the average grey-level intensity value of the source image in the region coincident with the template  image.

*t* is the matrix of grey-level pixel intensities in the template image.

*t* is the average grey-level intensity value of the template image.

The value of c(u,v) ranges from −1 to 1 and is independent of scale changes of the source and template images. The maximum value of c(u,v) indicates a position where the template best matches the source image. To extend normalised cross correlation to detect patterns that are rotated requires a new template-matching search for each angle, increasing the computational cost. The standard grey-level template-matching approach uses a single template to search for an individual component. The problem of locating and identifying similar components that exhibit variations of grey-level appearance requires a template-modelling approach [].

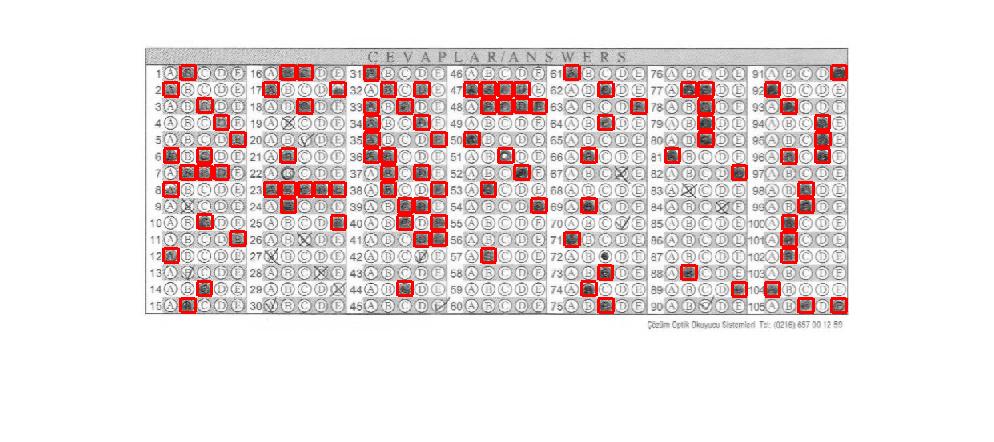
# 3.3.1.1 Template matching algorithm

To find out which bubbles are filled in, we need to convert the original image into the grayscale**,** then use normxcorr2function. MATLAB command ‘corr2’ is used to find the correlation coefficient. The Target Image is placed over the template image and correlation coefficient for each pixel in the template image is found to construct the correlation map. After sliding through all the pixels in the template image, the maximum coefficient is obtained from the map. The pixel position with maximum value is the starting point of the target image. In the above example, maximum value is 0.579 .Loop through all of the bubbles in a column, figure out where they are in the image, and then look at the pixels in that location and see if the bubble looks filled in (Figure 25).

The generalised template be can used to produce the maximum likelihood image of the search space as shown in Fig. 23.



*Figure 23: Generalized Template*



*Figure 24: Template matching method*

Finally, using this method, we get the result by 95 %.Not a bad result! Let’s see how other methods works.

# 3.3.2 Circle finding using the Hough transform method

The circle Hough Transform (CHT) is a basic technique used in Digital Image Processing, for detecting circular objects in a digital image.

The circle Hough Transform (CHT) is a feature extraction technique for detecting circles. It is a specialization of Hough Transform. The purpose of the technique is to find circles in imperfect image inputs. The circle candidates are produced by “voting” in the Hough parameter space and then select the local maxima in a so-called accumulator matrix.

If a circle in the image is described as

(x-a)^2 + (y-b) ^2 = r^2,

where (a, b) are the coordinate of the circle center and r is its radius, then an arbitrary edge point (xi, yi,) will be transformed into a right circular cone in the (a, b ,r) parameter space [A COMPARATIVE STUDY OF HOUGH TRANSFORM METHODS FOR CIRCLE FINDING]. If all the image points lie on a circle then the cones will intersect at a single point in (a, b, r) corresponding to the parameters of the circle. Kimme et al [] give probably the first known application of the Hough Transform to detecting circles in real images. In their work, they have made use of the direction of the gradient at each edge point. The center of a circle must lie on the normal at the edge point. As a result instead of incrementing the whole circular cone, only segments of the cone need be incremented. The size the region which is incremented depends on the accuracy of the edge direction estimation.

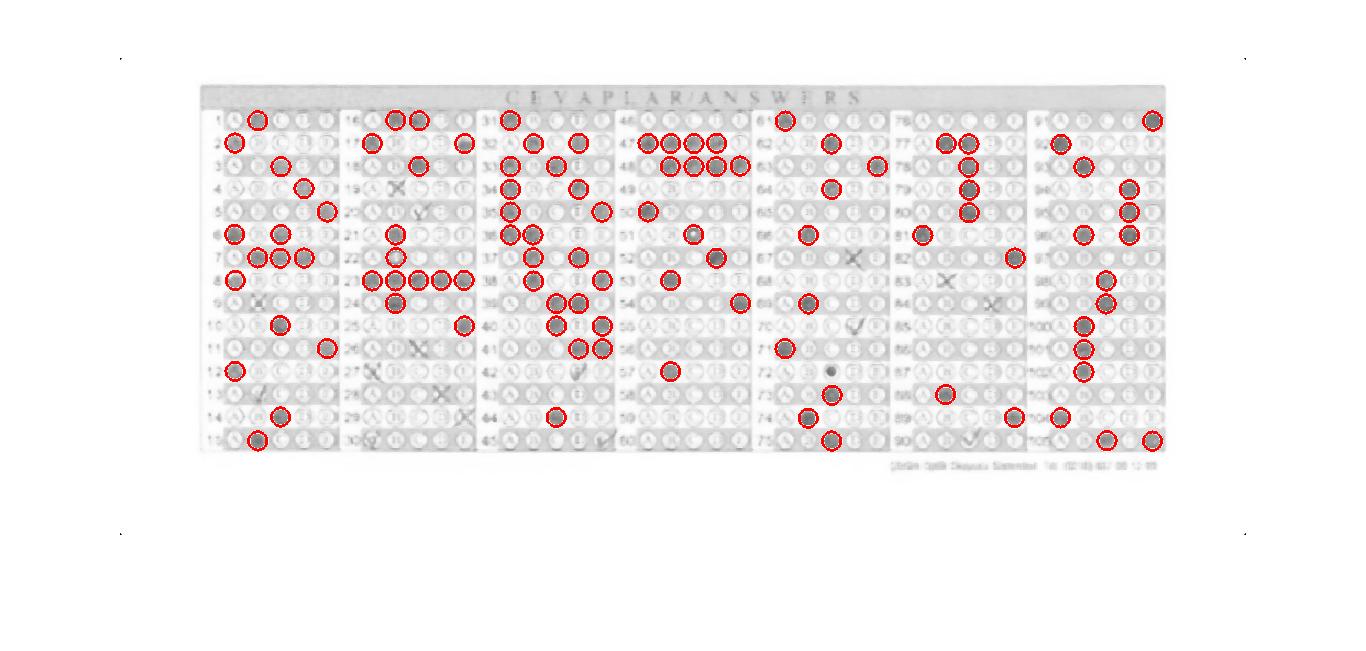
An important part of the complete Hough Transform process is peak detection. An extremely useful technique which we have found eases the peak finding problem considerably is the post-processing method proposed by Gerig and Klein [].It consists of a second data pass which takes each edge point and identifies the maximum value in the accumulator array out of all parameter values voted for by the point. The edge point is labelled with this location. In all the methods considered, this technique is used to detect the final peaks.

# 3.3.2.1 The standard Hough transform

The Standard Hough Transform (SHT) in this study follows the basic idea outlined in the previous section. A 3-D accumulator array is employed and edge direction information is used to limit voting to a section of the cone. In an ideal situation, the center of the circle must lie on a line oriented normal to the edge direction. Therefore we only have to move along the normal of every edge point to find the possible locations of centers. The distance between each edge point and the estimated center is a candidate for radius of the corresponding circle. However, in practice, the edge direction is usually estimated inaccurately. As a result, the detection of the true local maximum in the accumulator array could be difficult if this simple accumulation strategy is used. If the direction error is known to be within a range of ±△[Greek phi Didot.svg](https://en.wikipedia.org/wiki/File:Greek_phi_Didot.svg), then we may say that the center of the circle for the point (xi, yi) is within a certain region. This region diverges as the radius increases. To increment this region exactly in the accumulator is very difficult. The next section explains the algorithm which we use to find circles.

# 3.3.2.2 The circle Hough transform algorithm

The radius is fixed, then the parameter space would be reduced to 2D (the position of the circle center). For each point (x, y) on the original circle, it can define a circle centered at (x, y) with radius R according to (1). The intersection point of all such circles in the parameter space would be corresponding to the center point of the original circle. [wiki]



*Figure 25: The circle Hough Transform method*

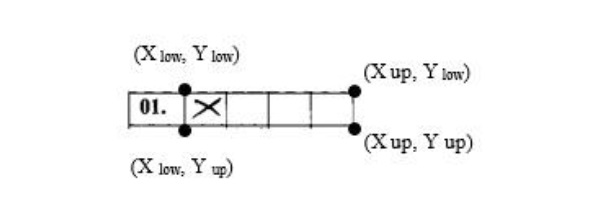
Statistics show that this method lags behind the template matching and outputs a result by 70%.

# 3.3.1 Using coordinate`s method

In this step method to achieve Answer Area Allocation, the following steps must be applied sequentially:

* Invert image answer sheet to binary image.
* Get the coordinates of the two red squares
* Cut the sheet into seat number, section one and section two sections
* Scan each row of the seat number for identifying each number of the seat number
* Using the moments in x and  y direction and the spatial moments, calculate the x and y coordinates
* Map the found coordinates for finding the respective marked seat number
* Scan each row of the first section for identifying answer marked for each section
* Using the moments in x and y direction and the spatial moments, calculate the x and y coordinates
* Map the found coordinates for finding the respective marked answer
* Follow the same steps to identify the marked answers for section two
* Compare the stored scanned answers in the array with actual answers in the database
* Depending on the range of marks defined, declare the class awarded to the student.
* Stop

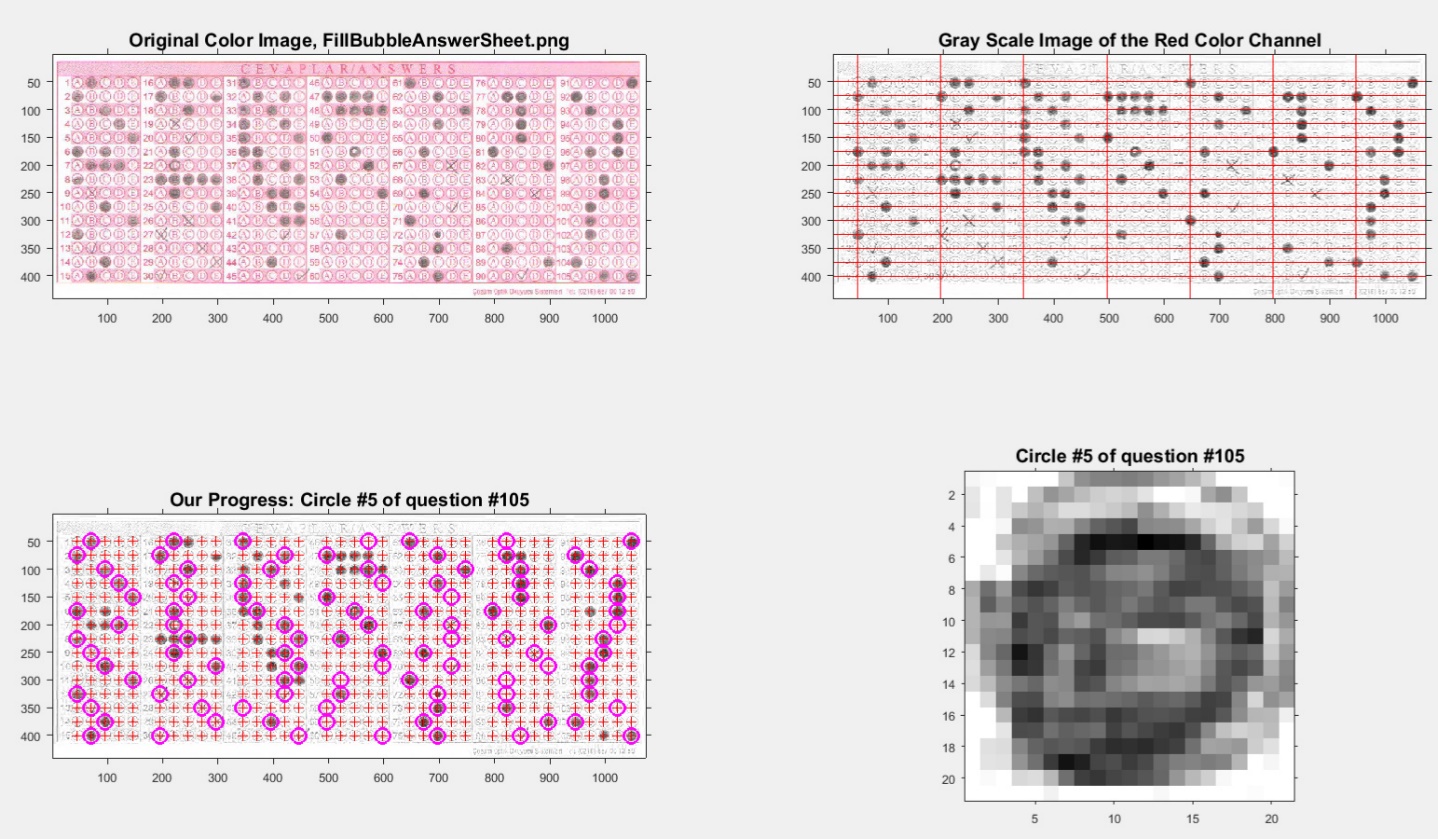
Then each question zone must be determined in order to score the response of that question. Question number is excluded from question segment. Each question comprises 4 positions (X up, X low, Y up, Y low) which are lower and upper bounds of the question answer zone as shown in Figure.



*Figure 26: X,Y algorithm detecting*

The recognized bubbles are stored in the spreadsheet, and are compared against the correct answer keys provided by the administrator. The evaluated results are also stored in the same spreadsheet and made available to the administrator for further grading.

Following figure shows output image after using x, y coordinates method. The proposed system has been tested and some of the bubbles was wrongly detected. Nevertheless, some of the ticks were ambiguously identificated by bubbles as shown in Fig. 27 and the output of the system for such bubbles was ‘detected’ instead of classifying them as filled or unfilled, which is desirable in real life examination.



*Figure 27: Coordinate`s method*

Statistics show that this method lags behind the other methods and outputs a result by 50%.

# REFERENCES

[1] <http://www.statista.com/statistics/232786/forecast-of-andrioid-users-in-the-us/>.

[2]<http://www.androidheadlines.com/2016/03/featured-top-10-learn-a-language-apps-for-android-march-2016.html>.

[3] <http://www.qrcodescanning.com/mobile-technology-classroom-learning.html>.

[4] [www.technopedia.com](http://www.technopedia.com)

# APPENDIX

**App>Manifest>Android.Manifest.xml**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
package="com.math.raiana.dictionary">  
  
<application  
 android:allowBackup="false"  
 android:icon="@drawable/logo\_main"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme">  
 <activity  
 android:configChanges="locale"  
 android:name=".MainActivity"  
 android:label="@string/app\_name"  
 android:theme="@style/AppTheme.NoActionBar">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <activity  
 android:configChanges="locale"  
 android:name=".DefinitionActivity"  
 android:label="@string/title\_activity\_definition"  
 android:theme="@style/AppTheme.NoActionBar" />  
 <activity  
 android:configChanges="locale"  
 android:name=".About"  
 android:label="@string/about\_title"  
 android:theme="@style/AboutStyle" />  
 <activity  
 android:configChanges="locale"  
 android:name=".SettingsActivity"  
 android:label="@string/title\_activity\_settings"/>  
</application>  
  
</manifest>

**Java>com.math.raiana.dictionary>abaut.java**

package com.math.raiana.dictionary;  
  
import android.app.Activity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
  
public class About extends Activity {  
  
 @Override  
 public void onCreate(Bundle savedInstance)  
 {  
 setContentView(R.layout.about\_dialog);  
 super.onCreate(savedInstance);  
 Button close=(Button)findViewById(R.id.close);  
 close.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 onBackPressed();  
 }  
 });  
 }  
}

**AppCompatPreferenceActivity.java**

package com.math.raiana.dictionary;  
  
import android.content.res.Configuration;  
import android.os.Bundle;  
import android.preference.PreferenceActivity;  
import android.support.annotation.LayoutRes;  
import android.support.annotation.NonNull;  
import android.support.v7.app.ActionBar;  
import android.support.v7.app.AppCompatDelegate;  
import android.view.MenuInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
public abstract class AppCompatPreferenceActivity extends PreferenceActivity {  
  
 private AppCompatDelegate mDelegate;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 getDelegate().installViewFactory();  
 getDelegate().onCreate(savedInstanceState);  
 super.onCreate(savedInstanceState);  
 }  
  
 @Override  
 protected void onPostCreate(Bundle savedInstanceState) {  
 super.onPostCreate(savedInstanceState);  
 getDelegate().onPostCreate(savedInstanceState);  
 }  
  
 public ActionBar getSupportActionBar() {  
 return getDelegate().getSupportActionBar();  
 }  
  
 @Override  
 @NonNull  
 public MenuInflater getMenuInflater() {  
 return getDelegate().getMenuInflater();  
 }  
  
 @Override  
 public void setContentView(@LayoutRes int layoutResID) {  
 getDelegate().setContentView(layoutResID);  
 }  
  
 @Override  
 public void setContentView(View view) {  
 getDelegate().setContentView(view);  
 }  
  
 @Override  
 public void setContentView(View view, ViewGroup.LayoutParams params) {  
 getDelegate().setContentView(view, params);  
 }  
  
 @Override  
 public void addContentView(View view, ViewGroup.LayoutParams params) {  
 getDelegate().addContentView(view, params);  
 }  
  
 @Override  
 protected void onPostResume() {  
 super.onPostResume();  
 getDelegate().onPostResume();  
 }  
  
 @Override  
 protected void onTitleChanged(CharSequence title, int color) {  
 super.onTitleChanged(title, color);  
 getDelegate().setTitle(title);  
 }  
  
 @Override  
 public void onConfigurationChanged(Configuration newConfig) {  
 super.onConfigurationChanged(newConfig);  
 getDelegate().onConfigurationChanged(newConfig);  
 }  
  
 @Override  
 protected void onStop() {  
 super.onStop();  
 getDelegate().onStop();  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 getDelegate().onDestroy();  
 }  
  
 public void invalidateOptionsMenu() {  
 getDelegate().invalidateOptionsMenu();  
 }  
  
 private AppCompatDelegate getDelegate() {  
 if (mDelegate == null) {  
 mDelegate = AppCompatDelegate.create(this, null);  
 }  
 return mDelegate;  
 }  
}

**DefinitionActivity.java**

package com.math.raiana.dictionary;  
  
import android.content.Intent;  
import android.support.v7.app.ActionBar;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
  
import android.support.v4.app.Fragment;  
import android.support.v4.app.FragmentManager;  
import android.support.v4.app.FragmentPagerAdapter;  
import android.support.v4.view.ViewPager;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
  
import android.widget.TextView;  
  
import java.util.ArrayList;  
  
public class DefinitionActivity extends AppCompatActivity {  
  
 public static final String KEY\_POSITION = "position";  
 public static final String KEY\_WORD = "word";  
 public static final String KEY\_DEFINITION="definition";  
 public static ArrayList<String> words;  
 public static ArrayList<String> defis;  
 int position;  
 public static String currentWords;  
 public static String currentDefis;  
 public static WordDbAdapter wordDbAdapter;  
 public static MenuItem menuItem;  
 public static boolean ready=false;  
 /\*\*  
 \* The {@link ViewPager} that will host the section contents.  
 \*/  
 private static ViewPager mViewPager;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_definition);  
 defis=getIntent().getStringArrayListExtra(KEY\_DEFINITION);  
 words=getIntent().getStringArrayListExtra(KEY\_WORD);  
 position=getIntent().getIntExtra(KEY\_POSITION,0);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
 setSupportActionBar(toolbar);  
 setupActionBar();  
 // Create the adapter that will return a fragment for each of the three  
 // primary sections of the activity.  
 SectionsPagerAdapter mSectionsPagerAdapter = new SectionsPagerAdapter(getSupportFragmentManager());  
 // Set up the ViewPager with the sections adapter.  
 mViewPager = (ViewPager) findViewById(R.id.container);  
 mViewPager.setAdapter(mSectionsPagerAdapter);  
 mViewPager.setCurrentItem(position);  
 mViewPager.addOnPageChangeListener(new ViewPager.SimpleOnPageChangeListener(){  
 @Override  
 public void onPageSelected(int position)  
 {  
 currentWords = words.get(mViewPager.getCurrentItem());  
 currentDefis=defis.get(mViewPager.getCurrentItem());  
 checkState(ready);  
 }  
 });  
 currentWords=words.get(position);  
 currentDefis=defis.get(position);  
 wordDbAdapter=new WordDbAdapter(getApplicationContext());  
 wordDbAdapter.open();  
 }  
  
 private void setupActionBar() {  
 ActionBar actionBar = getSupportActionBar();  
 if (actionBar != null) {  
 // Show the Up button in the action bar.  
 actionBar.setDisplayHomeAsUpEnabled(true);  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 // Inflate the menu; this adds items to the action bar if it is present.  
 getMenuInflater().inflate(R.menu.menu\_definition, menu);  
 menuItem=menu.findItem(R.id.favorite\_menu);  
 checkState(true);  
 ready=true;  
 return true;  
 }  
 public static void checkState(boolean ready)  
 {  
 if(ready) {  
 if (wordDbAdapter.isFavorite(currentWords)) {  
 menuItem.setIcon(R.drawable.loveon);  
 } else {  
 menuItem.setIcon(R.drawable.loveoff);  
 }  
 }  
 }  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
   
 int id = item.getItemId();  
  
 //noinspection SimplifiableIfStatement  
 if (id == R.id.favorite\_menu){  
 if(wordDbAdapter.isFavorite(currentWords)) {  
 wordDbAdapter.removeFavorite(currentWords);  
 item.setIcon(R.drawable.loveoff);  
 }  
 else {  
 wordDbAdapter.createFavorite(currentWords, currentDefis);  
 item.setIcon(R.drawable.loveon);  
 }  
 return true;  
 }  
 else if(id==R.id.share\_def)  
 {  
  
 Intent shareIntent=new Intent(Intent.ACTION\_SEND);  
 shareIntent.setType("text/plain");  
 shareIntent.putExtra(Intent.EXTRA\_TEXT, currentWords + "-" + currentDefis);  
 startActivity(Intent.createChooser(shareIntent,"Share with friends!"));  
 return true;  
 }  
 else if(id==android.R.id.home)  
 {  
 onBackPressed();  
 }  
 return super.onOptionsItemSelected(item);  
 }  
  
   
 public static class PlaceholderFragment extends Fragment {  
   
 private static final String ARG\_SECTION\_NUMBER = "section\_number";  
 public PlaceholderFragment() {  
  
 }  
  
   
 public static PlaceholderFragment newInstance(int sectionNumber) {  
 PlaceholderFragment fragment = new PlaceholderFragment();  
 Bundle args = new Bundle();  
 args.putInt(ARG\_SECTION\_NUMBER, sectionNumber);  
 fragment.setArguments(args);  
 return fragment;  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 int page=getArguments().getInt(ARG\_SECTION\_NUMBER);  
 View rootView = inflater.inflate(R.layout.fragment\_definition, container, false);  
 TextView textWord = (TextView) rootView.findViewById(R.id.word\_label);  
 TextView textDefi = (TextView) rootView.findViewById(R.id.definition\_label);  
 textWord.setText(words.get(page));  
 textDefi.setText(defis.get(page));  
 return rootView;  
 }  
 }  
  
  
 public class SectionsPagerAdapter extends FragmentPagerAdapter {  
  
 public SectionsPagerAdapter(FragmentManager fm) {  
 super(fm);  
 }  
  
 @Override  
 public Fragment getItem(int position) {  
 // getItem is called to instantiate the fragment for the given page.  
 // Return a PlaceholderFragment (defined as a static inner class below).  
 return PlaceholderFragment.newInstance(position);  
 }  
  
 @Override  
 public int getCount() {  
 // Show 3 total pages.  
 return words.size();  
 }  
  
 }  
}

**FavoriteAdapter.java**

package com.math.raiana.dictionary;  
  
import android.content.Context;  
import android.content.Intent;  
import android.database.Cursor;  
import android.support.v7.widget.RecyclerView;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import java.util.ArrayList;  
  
public class FavoriteAdapter extends RecyclerView.Adapter<FavoriteAdapter.FavoriteHolder> {  
 public static final String KEY\_WORD = "word";  
 public static final String KEY\_DEFINITION="definition";  
 public static final String KEY\_POSITION = "position";  
 public static ArrayList<String> words;  
 public static ArrayList<String> definitions;  
 public FavoriteAdapter(Cursor cursor) {  
 words=convert(cursor, KEY\_WORD);  
 definitions=convert(cursor,KEY\_DEFINITION);  
 }  
 @Override  
 public FavoriteAdapter.FavoriteHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
 Context context=parent.getContext();  
 View view= LayoutInflater.from(context).inflate(R.layout.result, parent, false);  
 return new FavoriteHolder(context,view);  
 }  
  
 @Override  
 public void onBindViewHolder(FavoriteAdapter.FavoriteHolder holder, int position) {  
  
 holder.mItemTextView.setText(words.get(position));  
 }  
  
 @Override  
 public int getItemCount() {  
 return words.size();  
 }  
  
 public ArrayList<String> convert(Cursor cursor,String key)  
 {  
 ArrayList<String> list=new ArrayList<>();  
 if(cursor.moveToFirst())  
 {  
 do {  
 list.add(cursor.getString(cursor.getColumnIndex(key)));  
 }  
 while (cursor.moveToNext());  
 }  
 else {  
 cursor.close();  
 return list;  
 }  
 return list;  
 }  
  
 //View Holder class to hold view  
 public static class FavoriteHolder extends RecyclerView.ViewHolder implements View.OnClickListener{  
  
 private TextView mItemTextView;  
 private Context mContext;  
  
 public FavoriteHolder(Context context,View view)  
 {  
 super(view);  
 mContext=context;  
 mItemTextView=(TextView)view.findViewById(R.id.resultList);  
 view.setOnClickListener(this);  
 }  
 @Override  
 public void onClick(View v) {  
 int position =getLayoutPosition();  
 Intent intent=new Intent(mContext,DefinitionActivity.class);  
 intent.putExtra(KEY\_POSITION, position);  
 intent.putStringArrayListExtra(KEY\_WORD,words);  
 intent.putStringArrayListExtra(KEY\_DEFINITION,definitions);  
 mContext.startActivity(intent);  
 }  
 }  
}

**FavoriteFragment.java**

package com.math.raiana.dictionary;  
  
import android.app.AlertDialog;  
import android.content.DialogInterface;  
import android.database.Cursor;  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.support.v7.widget.LinearLayoutManager;  
import android.support.v7.widget.RecyclerView;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuInflater;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
public class FavoriteFragment extends Fragment implements DialogInterface.OnClickListener {  
 RecyclerView recyclerView;  
 WordDbAdapter wordDbAdapter;  
 TextView textView;  
  
 @Override  
 public void onCreate(Bundle savedInstance)  
 {  
 setHasOptionsMenu(true);  
 super.onCreate(savedInstance);  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater,ViewGroup container,Bundle savedInstance)  
 {  
 View root=inflater.inflate(R.layout.fragment\_favorite, container,false);  
 recyclerView =(RecyclerView)root.findViewById(R.id.listFavorite);  
 recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));  
 textView=(TextView)root.findViewById(R.id.favorite\_empty);  
 wordDbAdapter=new WordDbAdapter(getContext());  
 wordDbAdapter.open();  
 return root;  
 }  
 @Override  
 public void onCreateOptionsMenu(Menu menu,MenuInflater inflater)  
 {  
 inflater.inflate(R.menu.menu\_favorite, menu);  
 }  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 if(item.getItemId()==R.id.delete\_favorite)  
 {  
 AlertDialog.Builder builder=new AlertDialog.Builder(getContext());  
 builder.setTitle(getString(R.string.delete\_dialog));  
 builder.setMessage(getString(R.string.delete\_all));  
 builder.setPositiveButton(getString(R.string.positive\_dialog), this);  
 builder.setNegativeButton(getString(R.string.negative\_dialog), this);  
 builder.setIcon(R.drawable.logo42);  
 builder.show();  
 }  
 return super.onOptionsItemSelected(item);  
 }  
  
 @Override  
 public void onResume()  
 {  
 showResult();  
 super.onResume();  
 }  
  
  
  
 public void showResult()  
 {  
 Cursor cursor=wordDbAdapter.getFavorite();  
 if(cursor!=null)  
 {  
 recyclerView.setVisibility(View.VISIBLE);  
 textView.setVisibility(View.GONE);  
 FavoriteAdapter hAdapter=new FavoriteAdapter(cursor);  
 recyclerView.setAdapter(hAdapter);  
 }  
 else{  
 textView.setVisibility(View.VISIBLE);  
 recyclerView.setVisibility(View.GONE);  
 }  
 }  
  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 switch (which)  
 {  
 case DialogInterface.BUTTON\_NEGATIVE:  
 break;  
 case DialogInterface.BUTTON\_POSITIVE:  
 wordDbAdapter.deleteAllFavorite();  
 showResult();  
 break;  
 }  
 }  
}

**HistoryAdapter.java**

package com.math.raiana.dictionary;  
  
import android.content.Context;  
import android.content.Intent;  
import android.database.Cursor;  
import android.support.v7.widget.RecyclerView;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageButton;  
import android.widget.TextView;  
  
import java.util.ArrayList;  
  
public class HistoryAdapter extends RecyclerView.Adapter<HistoryAdapter.HistoryHolder> {  
 public static final String KEY\_WORD = "word";  
 public static final String KEY\_DEFINITION="definition";  
 public static final String KEY\_POSITION = "position";  
 public static ArrayList<String> words;  
 public static ArrayList<String> definitions;  
 public WordDbAdapter wordDbAdapter;  
 public HistoryFragment fragment;  
 public HistoryAdapter(Cursor cursor,HistoryFragment frag)  
 {  
 fragment=frag;  
 wordDbAdapter=fragment.wordDbAdapter;  
 words=convert(cursor, KEY\_WORD);  
 definitions=convert(cursor,KEY\_DEFINITION);  
 }  
 @Override  
 public HistoryAdapter.HistoryHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
 Context context=parent.getContext();  
 View view= LayoutInflater.from(context).inflate(R.layout.history\_adapter, parent, false);  
 return new HistoryHolder(context,view);  
 }  
  
 @Override  
 public void onBindViewHolder(final HistoryAdapter.HistoryHolder holder, final int position) {  
  
 holder.mItemTextView.setText(words.get(position));  
 holder.mItemDelete.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 wordDbAdapter.removeHistory(words.get(position));  
 fragment.showResult();  
 }  
 });  
 }  
  
 @Override  
 public int getItemCount() {  
 return words.size();  
 }  
  
 public ArrayList<String> convert(Cursor cursor,String key)  
 {  
 ArrayList<String> list=new ArrayList<>();  
 if(cursor.moveToFirst())  
 {  
 do {  
 list.add(cursor.getString(cursor.getColumnIndex(key)));  
 }  
 while (cursor.moveToNext());  
 }  
 else {  
 cursor.close();  
 return list;  
 }  
 return list;  
 }  
  
 //View Holder class to hold view  
 public static class HistoryHolder extends RecyclerView.ViewHolder implements View.OnClickListener{  
  
 private TextView mItemTextView;  
 private ImageButton mItemDelete;  
 private Context mContext;  
 public HistoryHolder(Context context,View view)  
 {  
 super(view);  
 mContext=context;  
 mItemTextView=(TextView)view.findViewById(R.id.results);  
 mItemDelete=(ImageButton)view.findViewById(R.id.delete\_word);  
 view.setOnClickListener(this);  
  
 }  
 @Override  
 public void onClick(View v) {  
 int position =getLayoutPosition();  
 Intent intent=new Intent(mContext,DefinitionActivity.class);  
 intent.putExtra(KEY\_POSITION, position);  
 intent.putStringArrayListExtra(KEY\_WORD,words);  
 intent.putStringArrayListExtra(KEY\_DEFINITION,definitions);  
 mContext.startActivity(intent);  
 }  
 }  
}

**HistoryFragment.java**

package com.math.raiana.dictionary;  
  
import android.app.AlertDialog;  
import android.content.DialogInterface;  
import android.database.Cursor;  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.support.v7.widget.LinearLayoutManager;  
import android.support.v7.widget.RecyclerView;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuInflater;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
public class HistoryFragment extends Fragment implements DialogInterface.OnClickListener {  
 RecyclerView recyclerView;  
 WordDbAdapter wordDbAdapter;  
 TextView textView;  
  
 @Override  
 public void onCreate(Bundle savedInstance)  
 {  
 setHasOptionsMenu(true);  
 super.onCreate(savedInstance);  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater,ViewGroup container,Bundle savedInstance)  
 {  
 View root=inflater.inflate(R.layout.fragment\_history, container,false);  
 recyclerView =(RecyclerView)root.findViewById(R.id.listHistory);  
 recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));  
  
 textView=(TextView)root.findViewById(R.id.history\_empty);  
 wordDbAdapter=new WordDbAdapter(getContext());  
 wordDbAdapter.open();  
 return root;  
 }  
 @Override  
 public void onCreateOptionsMenu(Menu menu,MenuInflater inflater)  
 {  
 inflater.inflate(R.menu.menu\_history, menu);  
 }  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 if(item.getItemId()==R.id.delete\_favorite)  
 {  
 AlertDialog.Builder builder=new AlertDialog.Builder(getContext());  
 builder.setTitle(getString(R.string.delete\_dialog));  
 builder.setMessage(getString(R.string.delete\_all));  
 builder.setPositiveButton(getString(R.string.positive\_dialog), this);  
 builder.setNegativeButton(getString(R.string.negative\_dialog), this);  
 builder.setIcon(R.drawable.logo42);  
 builder.show();  
 }  
 return super.onOptionsItemSelected(item);  
 }  
  
 @Override  
 public void onResume()  
 {  
 showResult();  
 super.onResume();  
 }  
  
  
  
 public void showResult()  
 {  
 Cursor cursor=wordDbAdapter.getHistory();  
 if(cursor!=null)  
 {  
 recyclerView.setVisibility(View.VISIBLE);  
 textView.setVisibility(View.GONE);  
 HistoryAdapter hAdapter=new HistoryAdapter(cursor,this);  
 recyclerView.setAdapter(hAdapter);  
 }  
 else{  
 textView.setVisibility(View.VISIBLE);  
 recyclerView.setVisibility(View.GONE);  
 }  
 }  
  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 switch (which)  
 {  
 case DialogInterface.BUTTON\_NEGATIVE:  
 break;  
 case DialogInterface.BUTTON\_POSITIVE:  
 wordDbAdapter.deleteAllHistory();  
 showResult();  
 break;  
 }  
 }  
}

**MainActivity.java**

package com.math.raiana.dictionary;  
  
import android.content.Context;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.content.res.Configuration;  
import android.preference.PreferenceManager;  
import android.support.design.widget.NavigationView;  
import android.support.design.widget.TabLayout;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.support.v4.app.Fragment;  
import android.support.v4.app.FragmentManager;  
import android.support.v4.app.FragmentPagerAdapter;  
import android.support.v4.view.ViewPager;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.MenuItem;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.OutputStream;  
import java.util.Locale;  
  
  
public class MainActivity extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener {  
 public WordDbAdapter wordDbAdapter;  
 ViewPager mViewPager;  
 HistoryFragment historyFragment;  
 SearchFragment searchFragment;  
 FavoriteFragment favoriteFragment;  
 Locale locale;  
 Locale currentLocale;  
 SharedPreferences preferences;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 setLocale();  
 currentLocale=getResources().getConfiguration().locale;  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.drawer\_layout);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
 setSupportActionBar(toolbar);  
 // Create the adapter that will return a fragment for each of the three  
 // primary sections of the activity.  
 SectionsPagerAdapter mSectionsPagerAdapter = new SectionsPagerAdapter(getSupportFragmentManager());  
 searchFragment=new SearchFragment();  
 historyFragment=new HistoryFragment();  
 favoriteFragment=new FavoriteFragment();  
 // Set up the ViewPager with the sections adapter.  
 mViewPager = (ViewPager) findViewById(R.id.container);  
 mViewPager.setAdapter(mSectionsPagerAdapter);  
 TabLayout tabLayout = (TabLayout) findViewById(R.id.tabs);  
 tabLayout.setupWithViewPager(mViewPager);  
 DrawerLayout drawerLayout=(DrawerLayout)findViewById(R.id.drawer\_layout);  
 ActionBarDrawerToggle toggle=new ActionBarDrawerToggle(this,drawerLayout,toolbar,R.string.navigation\_drawer\_open,R.string.navigation\_drawer\_close);  
 drawerLayout.setDrawerListener(toggle);  
 toggle.syncState();  
 final NavigationView navigationView=(NavigationView)findViewById(R.id.nav\_view);  
 navigationView.setNavigationItemSelectedListener(this);  
 mViewPager.addOnPageChangeListener(new ViewPager.SimpleOnPageChangeListener() {  
 @Override  
 public void onPageSelected(int position) {  
 switch (position) {  
 case 0:  
 navigationView.setCheckedItem(R.id.search);  
 break;  
 case 1:  
 navigationView.setCheckedItem(R.id.history);  
 break;  
 case 2:  
 navigationView.setCheckedItem(R.id.favorite);  
 break;  
 }  
 }  
 });  
 wordDbAdapter=new WordDbAdapter(getApplicationContext());  
 wordDbAdapter.open();  
 copyData(getApplicationContext());  
 }  
 private void setLocale()  
 {  
 preferences= PreferenceManager.getDefaultSharedPreferences(getApplicationContext());  
 String lang=preferences.getString("example\_list", "default");  
 if (lang.equals("default"))  
 {  
 lang=getResources().getConfiguration().locale.getCountry();  
 }  
 locale=new Locale(lang);  
 Locale.setDefault(locale);  
 Configuration configuration=new Configuration();  
 configuration.locale=locale;  
 getBaseContext().getResources().updateConfiguration(configuration, getBaseContext().getResources().getDisplayMetrics());  
 }  
 @Override  
 public void onRestart()  
 {  
 super.onRestart();  
 Locale locale=getLocale(this);  
 if(!locale.equals(currentLocale))  
 {  
 currentLocale=locale;  
 recreate();  
 }  
  
 }  
 public static Locale getLocale(Context context){  
 SharedPreferences sharedPreferences = PreferenceManager.getDefaultSharedPreferences(context);  
 String lang = sharedPreferences.getString("example\_list", "en");  
 return new Locale(lang);  
 }  
 public void copyData(final Context context)  
 {  
 if (!wordDbAdapter.empty()) {  
 Thread thread=new Thread(new Runnable() {  
 @Override  
 public void run() {  
 String DB\_PATH = context.getFilesDir().getParent();  
 String DB\_NAME = "/databases/data.db";  
 String outFilename = DB\_PATH + DB\_NAME;  
 Log.d("BUG:",outFilename);  
 byte[] buffer = new byte[1024];  
 try {  
 OutputStream outputStream = new FileOutputStream(outFilename);  
 InputStream inputStream = getResources().openRawResource(R.raw.words);  
 int length;  
 while ((length = inputStream.read(buffer)) > 0) {  
 outputStream.write(buffer, 0, length);  
 }  
 inputStream.close();  
 outputStream.flush();  
 outputStream.close();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 });  
 thread.start();  
 }  
 wordDbAdapter.close();  
 }  
 @Override  
 public void onConfigurationChanged(Configuration configuration)  
 {  
 super.onConfigurationChanged(configuration);  
 setLocale();  
 }  
 @SuppressWarnings("StatementWithEmptyBody")  
 @Override  
 public boolean onNavigationItemSelected(MenuItem item) {  
 // Handle navigation view item clicks here.  
 int id = item.getItemId();  
  
 if (id == R.id.search) {  
 mViewPager.setCurrentItem(0,true);  
 }  
 else if (id==R.id.history)  
 {  
 mViewPager.setCurrentItem(1,true);  
 }  
 else if (id==R.id.favorite)  
 {  
 mViewPager.setCurrentItem(2,true);  
 }  
 else if (id==R.id.settings)  
 {  
 startActivity(new Intent(getApplicationContext(),SettingsActivity.class));  
 }  
 else if (id == R.id.nav\_share) {  
 Intent shareInt=new Intent(Intent.ACTION\_SEND);  
 shareInt.setType("text/plain");  
 shareInt.putExtra(Intent.EXTRA\_TEXT, "http://google.market.kg");  
 startActivity(Intent.createChooser(shareInt,getString(R.string.share\_friend)));  
 }  
 else if (id==R.id.about)  
 {  
 Intent intent=new Intent(getApplicationContext(),About.class);  
 startActivity(intent);  
 }  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer\_layout);  
 drawer.closeDrawer(GravityCompat.START);  
 return true;  
 }  
 public class SectionsPagerAdapter extends FragmentPagerAdapter {  
  
 public SectionsPagerAdapter(FragmentManager fm) {  
 super(fm);  
 }  
  
 @Override  
 public Fragment getItem(int position) {  
   
 switch (position)  
 {  
 case 0:  
 return searchFragment;  
 case 1:  
 return historyFragment;  
 case 2:  
 return favoriteFragment;  
 default:  
 return searchFragment;  
 }  
 }  
  
 @Override  
 public int getCount() {  
 // Show 3 total pages.  
 return 3;  
 }  
 @Override  
 public CharSequence getPageTitle(int position) {  
 switch (position) {  
 case 0:  
 return getString(R.string.tab\_search);  
 case 1:  
 return getString(R.string.tab\_history);  
 case 2:  
 return getString(R.string.tab\_favorite);  
 }  
 return null;  
 }  
 }  
  
}

**RecycleViewAdapter.java**

package com.math.raiana.dictionary;  
  
import android.content.Context;  
import android.content.Intent;  
import android.database.Cursor;  
import android.support.v7.widget.RecyclerView;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
import java.util.ArrayList;  
  
  
public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.RecyclerItemViewHolder> {  
 public static final String KEY\_WORD = "word";  
 public static final String KEY\_DEFINITION="definition";  
 public static final String KEY\_POSITION = "position";  
 public static ArrayList<String> words;  
 public static ArrayList<String> definitions;  
 public static String mKey;  
 public static WordDbAdapter wordDbAdapter;  
 public RecyclerViewAdapter(Cursor cursor,String key,WordDbAdapter adapter) {  
 mKey=key;  
 switch (key)  
 {  
 case KEY\_WORD:  
 words=convert(cursor, KEY\_WORD);  
 definitions=convert(cursor,KEY\_DEFINITION);  
 break;  
 case KEY\_DEFINITION:  
 words=convert(cursor, KEY\_DEFINITION);  
 definitions=convert(cursor,KEY\_WORD);  
 break;  
 default:  
 words=convert(cursor, KEY\_WORD);  
 definitions=convert(cursor,KEY\_DEFINITION);  
  
 }  
  
 for(int i=0; i<words.size();i++)  
 {  
 Log.d("word:"+i,words.get(i));  
 Log.d("definitions:" + i, definitions.get(i));  
 }  
 wordDbAdapter=adapter;  
 }  
 @Override  
 public RecyclerViewAdapter.RecyclerItemViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
 Context context=parent.getContext();  
 View view= LayoutInflater.from(context).inflate(R.layout.result, parent, false);  
 return new RecyclerItemViewHolder(context,view);  
 }  
  
  
 @Override  
 public void onBindViewHolder(RecyclerViewAdapter.RecyclerItemViewHolder holder, int position) {  
 holder.mItemTextView.setText(words.get(position));  
 }  
  
 @Override  
 public int getItemCount() {  
 return words.size();  
 }  
  
 public ArrayList<String> convert(Cursor cursor,String key)  
 {  
 ArrayList<String> list=new ArrayList<>();  
 if(cursor.moveToFirst())  
 {  
 do {  
 list.add(cursor.getString(cursor.getColumnIndex(key)));  
 }  
 while (cursor.moveToNext());  
 }  
 else {  
 cursor.close();  
 return list;  
 }  
 return list;  
 }  
  
 //View Holder class to hold view  
 public static class RecyclerItemViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener{  
 private TextView mItemTextView;  
 private Context mContext;  
  
 public RecyclerItemViewHolder(Context context,View view)  
 {  
 super(view);  
 mContext=context;  
 mItemTextView=(TextView)view.findViewById(R.id.resultList);  
 view.setOnClickListener(this);  
 }  
 @Override  
 public void onClick(View v) {  
 int position =getLayoutPosition();  
 Intent intent=new Intent(mContext,DefinitionActivity.class);  
 intent.putExtra(KEY\_POSITION, position);  
 intent.putStringArrayListExtra(KEY\_WORD,words);  
 intent.putStringArrayListExtra(KEY\_DEFINITION,definitions);  
 wordDbAdapter.createHistory(words.get(position),definitions.get(position));  
 mContext.startActivity(intent);  
 }  
 }  
}

**SearchFragment.java**

package com.math.raiana.dictionary;  
  
import android.database.Cursor;  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.support.v7.widget.LinearLayoutManager;  
import android.support.v7.widget.RecyclerView;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuInflater;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.SearchView;  
  
public class SearchFragment extends Fragment implements SearchView.OnQueryTextListener,SearchView.OnCloseListener{  
 public static final String KEY\_WORD = "word";  
 public static final String KEY\_DEFINITION="definition";  
 RecyclerView recyclerView;  
 SearchView searchView;  
 WordDbAdapter wordDbAdapter;  
 @Override  
 public void onCreate(Bundle savedInstance)  
 {  
 setHasOptionsMenu(true);  
 super.onCreate(savedInstance);  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater,ViewGroup container,Bundle savedInstance)  
 {  
 View root=inflater.inflate(R.layout.fragment\_search, container, false);  
 searchView=(SearchView)root.findViewById(R.id.searchView);  
 searchView.setOnQueryTextListener(this);  
 searchView.setOnCloseListener(this);  
 recyclerView=(RecyclerView)root.findViewById(R.id.listView);  
 recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));  
 wordDbAdapter=new WordDbAdapter(getContext());  
 wordDbAdapter.open();  
 return root;  
 }  
 @Override  
 public void onCreateOptionsMenu(Menu menu,MenuInflater inflater)  
 {  
 inflater.inflate(R.menu.menu\_search, menu);  
 }  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id= item.getItemId();  
 switch (id)  
 {  
 case R.id.item\_o:  
 searchView.setQuery(searchView.getQuery()+getString(R.string.string\_o),true);  
 break;  
 case R.id.item\_n:  
 searchView.setQuery(searchView.getQuery()+getString(R.string.string\_n),true);  
 break;  
 case R.id.item\_u:  
 searchView.setQuery(searchView.getQuery()+getString(R.string.string\_u),true);  
 break;  
 }  
 return super.onOptionsItemSelected(item);  
 }  
  
 @Override  
 public boolean onClose() {  
 showResult("");  
 wordDbAdapter.close();  
 return false;  
 }  
  
 @Override  
 public boolean onQueryTextSubmit(String query) {  
 showResult(query);  
 return false;  
 }  
  
 @Override  
 public boolean onQueryTextChange(String newText) {  
 showResult(newText);  
 return false;  
 }  
  
 public void showResult(String q)  
 {  
 if(q.equals(""))  
 {  
 recyclerView.setVisibility(View.GONE);  
 }  
 else {  
 Cursor cursor=wordDbAdapter.searchWord(q,0);  
 RecyclerViewAdapter adapter;  
 if(cursor!=null)  
 {  
 recyclerView.setVisibility(View.VISIBLE);  
 adapter=new RecyclerViewAdapter(cursor,KEY\_WORD,wordDbAdapter);  
 recyclerView.setAdapter(adapter);  
 }  
 else  
 {  
 cursor=wordDbAdapter.searchWord(q,1);  
 if(cursor != null)  
 {  
 recyclerView.setVisibility(View.VISIBLE);  
 adapter=new RecyclerViewAdapter(cursor,KEY\_DEFINITION,wordDbAdapter);  
 recyclerView.setAdapter(adapter);  
 }  
 else {  
 recyclerView.setVisibility(View.GONE);  
 }  
 }  
 }  
 }  
}

**SettingActivity.java**

package com.math.raiana.dictionary;  
  
import android.os.Bundle;  
import android.preference.ListPreference;  
import android.preference.Preference;  
import android.support.v7.app.ActionBar;  
import android.preference.PreferenceManager;  
import android.view.MenuItem;  
  
  
public class SettingsActivity extends AppCompatPreferenceActivity {  
  
   
 private static Preference.OnPreferenceChangeListener sBindPreferenceSummaryToValueListener = new Preference.OnPreferenceChangeListener()  
 {  
 @Override  
 public boolean onPreferenceChange(Preference preference, Object value) {  
 String stringValue = value.toString();  
 if (preference instanceof ListPreference) {  
 // For list preferences, look up the correct display value in  
 // the preference's 'entries' list.  
 ListPreference listPreference = (ListPreference) preference;  
 int index = listPreference.findIndexOfValue(stringValue);  
 // Set the summary to reflect the new value.  
 preference.setSummary(  
 index >= 0  
 ? listPreference.getEntries()[index]  
 : null);  
  
 }else {  
 preference.setSummary(stringValue);  
 }  
 return true;  
 }  
 };  
  
 private static void bindPreferenceSummaryToValue(Preference preference) {  
 preference.setOnPreferenceChangeListener(sBindPreferenceSummaryToValueListener);  
  
 sBindPreferenceSummaryToValueListener.onPreferenceChange(preference,  
 PreferenceManager  
 .getDefaultSharedPreferences(preference.getContext())  
 .getString(preference.getKey(), ""));  
 }  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 addPreferencesFromResource(R.xml.pref\_general);  
 bindPreferenceSummaryToValue(findPreference("example\_list"));  
 setupActionBar();  
 }  
  
 private void setupActionBar() {  
 ActionBar actionBar = getSupportActionBar();  
 if (actionBar != null) {  
 // Show the Up button in the action bar.  
 actionBar.setDisplayHomeAsUpEnabled(true);  
 }  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
 if (id == android.R.id.home) {  
 onBackPressed();  
 return true;  
 }  
 return super.onOptionsItemSelected(item);  
 }  
}

**WordAdapter.java**

package com.math.raiana.dictionary;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.SQLException;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
import android.database.sqlite.SQLiteQueryBuilder;  
import android.util.Log;  
  
public class WordDbAdapter {  
 public static final String KEY\_ID="\_id";  
 public static final String KEY\_WORD = "word";  
 public static final String KEY\_DEFINITION="definition";  
 public static final String TAG = "CustomersDbAdapter";  
 public DatabaseHelper mDbHelper;  
 public SQLiteDatabase mDb;  
 private static final String DATABASE\_NAME = "data.db";  
 public static final String DATA\_HISTORY="history";  
 public static final String DATA\_FAVORITE="favorite";  
 private static final String FTS\_VIRTUAL\_TABLE = "dictionary";  
 private static final int DATABASE\_VERSION = 1;  
 //Create a FTS3 Virtual Table for fast searches  
 private static final String DATABASE\_CREATE =  
 "CREATE VIRTUAL TABLE " + FTS\_VIRTUAL\_TABLE + " USING fts3(" +  
 KEY\_ID+" INTEGER PRIMARY KEY AUTOINCREMENT,"+  
 KEY\_WORD + "," +  
 KEY\_DEFINITION +");";  
  
 private static final String HISTORY\_CREATE =  
 "CREATE VIRTUAL TABLE " + DATA\_HISTORY + " USING fts3(" +  
 KEY\_ID+" INTEGER PRIMARY KEY AUTOINCREMENT,"+  
 KEY\_WORD + "," +  
 KEY\_DEFINITION +","+");";  
 private static final String FAVORITE\_CREATE =  
 "CREATE VIRTUAL TABLE " + DATA\_FAVORITE + " USING fts3(" +  
 KEY\_ID+" INTEGER PRIMARY KEY AUTOINCREMENT,"+  
 KEY\_WORD + "," +  
 KEY\_DEFINITION +","+");";  
 private final Context mCtx;  
  
 private static class DatabaseHelper extends SQLiteOpenHelper {  
 DatabaseHelper(Context context)  
 {  
 super(context, DATABASE\_NAME, null, DATABASE\_VERSION);  
 }  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 db.execSQL(DATABASE\_CREATE);  
 db.execSQL(HISTORY\_CREATE);  
 db.execSQL(FAVORITE\_CREATE);  
 }  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 Log.w(TAG, "Upgrading database from version " + oldVersion + " to "  
 + newVersion + ", which will destroy all old data");  
 db.execSQL("DROP TABLE IF EXISTS " + FTS\_VIRTUAL\_TABLE);  
 db.execSQL("DROP TABLE IF EXISTS " + DATA\_HISTORY);  
 db.execSQL("DROP TABLE IF EXISTS " + DATA\_FAVORITE);  
 onCreate(db);  
 }  
 }  
 public WordDbAdapter(Context ctx) {  
 this.mCtx = ctx;  
 }  
 public WordDbAdapter open() throws SQLException {  
 mDbHelper = new DatabaseHelper(mCtx);  
 mDb = mDbHelper.getWritableDatabase();  
 return this;  
 }  
 public void close() {  
 if (mDbHelper != null) {  
 mDbHelper.close();  
 }  
 }  
  
 public long createHistory(String word, String definition){  
  
 ContentValues initialValues = new ContentValues();  
 initialValues.put(KEY\_WORD, word);  
 initialValues.put(KEY\_DEFINITION, definition);  
 return mDb.insert(DATA\_HISTORY, null, initialValues);  
 }  
 public long createFavorite(String word, String definition){  
 ContentValues initialValues = new ContentValues();  
 initialValues.put(KEY\_WORD, word);  
 initialValues.put(KEY\_DEFINITION, definition);  
 return mDb.insert(DATA\_FAVORITE, null, initialValues);  
 }  
  
 public int removeFavorite(String word)  
 {  
 String[] args={word};  
 return mDb.delete(DATA\_FAVORITE, KEY\_WORD + " = ?", args);  
 }  
 public int removeHistory(String word)  
 {  
 String[] args={word};  
 return mDb.delete(DATA\_HISTORY, KEY\_WORD + " = ?", args);  
 }  
 public Cursor searchWord(String query,int dic){  
 String search\_word=KEY\_WORD;  
 if(dic==1) search\_word=KEY\_DEFINITION;  
 Log.w(TAG, query + "\*");  
 String[] column={KEY\_ID,KEY\_WORD,KEY\_DEFINITION};  
 String selection=search\_word+" LIKE ?";  
 String[] selectionArgs=new String[]{query+"%"};  
 return query(selection, selectionArgs, column, search\_word);  
 }  
  
  
 private Cursor query(String selection, String[] selectionArgs, String[] columns, String search\_word)  
 {  
 SQLiteQueryBuilder builder=new SQLiteQueryBuilder();  
 builder.setTables(FTS\_VIRTUAL\_TABLE);  
 Cursor cursor = builder.query(mDbHelper.getReadableDatabase(), columns, selection, selectionArgs, search\_word, null, KEY\_WORD);  
 if (!cursor.moveToFirst()) {  
 cursor.close();  
 return null;  
 }  
 return cursor;  
 }  
  
 public boolean empty()  
 {  
 String q="SELECT COUNT(\*) FROM "+FTS\_VIRTUAL\_TABLE+";";  
 Cursor cursor=mDb.rawQuery(q, null);  
 cursor.moveToFirst();  
 int i = cursor.getInt(cursor.getColumnIndex("COUNT(\*)"));  
 cursor.close();  
 Log.d(TAG,"size: "+i);  
 return i>=2594;  
 }  
  
 public Cursor getHistory()  
 {  
 String[] col={KEY\_ID,KEY\_WORD,KEY\_DEFINITION};  
 Cursor cursor=mDb.query(DATA\_HISTORY, col, null, null, KEY\_WORD, null, KEY\_ID);  
 if (!cursor.moveToFirst()) {  
 cursor.close();  
 return null;  
 }  
 return cursor;  
 }  
 public Cursor getFavorite()  
 {  
 String[] col={KEY\_ID,KEY\_WORD,KEY\_DEFINITION};  
 Cursor cursor=mDb.query(DATA\_FAVORITE,col,null,null,KEY\_WORD,null,null);  
 if (!cursor.moveToFirst()) {  
 cursor.close();  
 return null;  
 }  
 return cursor;  
 }  
 public boolean isFavorite(String word)  
 {  
 String[] col={KEY\_WORD};  
 String selection=KEY\_WORD+" = ?";  
 String[] args={word};  
 Cursor cursor=mDb.query(DATA\_FAVORITE,col,selection,args,null,null,null);  
 boolean res=cursor.moveToFirst();  
 cursor.close();  
 return res;  
 }  
 public boolean deleteAllHistory()  
 {  
 int doneDelete;  
 doneDelete = mDb.delete(DATA\_HISTORY, null , null);  
 Log.w(TAG, Integer.toString(doneDelete));  
 return doneDelete > 0;  
 }  
 public boolean deleteAllFavorite()  
 {  
 int doneDelete;  
 doneDelete = mDb.delete(DATA\_FAVORITE, null , null);  
 Log.w(TAG, Integer.toString(doneDelete));  
 return doneDelete > 0;