FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TURKEY

Visualizing The Path of a Photo Taker From Image Metadata

Rasty MAJEED

Master's Thesis

DEPARTMENT OF SOFTWARE ENGINEERING

JANUARY 2021

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TURKEY

Department of Software Engineering

Master's Thesis

Visualizing The Path of a Photo Taker From Image Metadata

Author
Rasty MAJEED

Supervisor
Assoc. Prof. Cihan VAROL

JANUARY 2021
ELAZIG

DECLARATION

First, I want to thank my Allah who helped me and gave me the ability and patience to complete this humble work. I would like to thank my Dad (Shakhawan). May he rest in peace, and especially my Mom (Payman), who supported me in all the phases of my life. She always has been the source of my strength, trust, and the rock I can lean on. I would also like to thank wife (Shayma), my brother (Muhammed) and sisters (Rawa, Roza) for everything they did for me, and for always being there throughout my journey. The duty of gratitude calls on me to extend my thanks and appreciation to the best guide and my supervisor, Dr. Cihan VAROL, who I was privileged to have supervised this study with. Furthermore, I would like to thank my brother and life-long friend (Mustafa) who supported me and was the sound of the reason in many cases in my life. Finally, I would like to extend my thanks and gratitude to all my colleagues and friends who have contributed with even just one word or one piece of scientific advice in the support of this work.10 January 2020

03 January 2021

Rasty MAJEED

PREFACE

One of the most challenging investigation in Forensic Analysis is the work conducted on smartphones. Specifically using smartphone data to provide evidence for a case is becoming a norm nowadays. Many Android device applications use and work with location services that assist investigators in working with log files that store valuable information.

This thesis's some problems were found, which are Some libraries had issues with Google Maps. A better processor was needed to run the tool faster, as it sometimes jammed. While running the codes, an exception occurred when obtaining the image metadata, due to the connection between the tool and the Google Maps API. I would like to thanks my supervisor Dr. Cihan Varol and appreciate his support for helping me in these problems

Rasty MAJEED
ELAZIG, 2021

TABLE OF CONTENTS

Pages

ABSTRACT

Visualizing the Path of a Photo Taker from Image Metadata

Rasty MAJEED

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Software Engineering

January 2021, Page: [xi+32]

Nowadays, mobile forensics is one of the most challenging aspects of the digital forensic analysis field. Smartphones provide a wealth of interesting data for forensic analysis and investigators. As many applications use location data to enhance or provide their services, the log files of Android devices can contain valuable information and geodata. Based on the geodata (longitude and latitude), it is possible to recreate the whereabouts of the smartphone and its user at specific points and locations in time. Image metadata is an interesting and important part of the information in any digital forensic investigation. It will show and provide information about the exact time and location where the picture was taken, the device that took the picture and much more. Currently, there are many commercial tools available that help investigators of computer forensic analysis to cover and work with a wide range of geographic information that is connected to criminal activities and crime scenes. However, to the best of our knowledge, there are no specific forensic tools available to work with the geolocation of pictures that have been taken by Android devices. Photos captured on Android devices that utilise Google Photos store the date/time and geolocation of the pictures in their properties for security reasons. This developed application depends on these photos to track the user's journey and find all the details of this journey. Photos taken in different places and at different times during a journey can be used in forensic analysis to locate routes between captured photos and show it on Google Maps based on the geolocation details.

Keywords: Mobile, Mobile Forensics, Android, Forensic analysis, Geo-location, Security, Google map, Metadata, Google API, Image.

ÖZET

Visualizing The Path of a Photo Taker from Image Metadata

Rasty MAJEED

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Yazılım Mühendisliği Anabilim Dalı

Ocak 2021, Sayfa: [xi+32]

Bugünlerde, mobil cihazların adli bilişim çerçevesinde incelenmesi önemli ve aynı zamanda zorlayıcı bir hal almıştır. Mobil cihazlar hem adli bilişim hem de uzmanlar için önemli veriler sunmaktadır. Bir çok uygulama konum verisini kullanarak servislerini verirken, Android cihazlarda ki log dataları konum bilgisi ve diğer hususlarda önemli bilgi barındırmaktadırlar. Konum bilgisi (Enlem-Boylam) kullanılarak, cihazın ve kullanıcının nerelerde bulunduğu öğrenilebilir. Resim meta verisi adli bilişim araştırmalarında önemli bir yer tutmaktadır. Bu meta veri herhangi bir resimin nerede, ne zaman, hangi makine ile çekildiği ve daha fazla bilgiye erişimi sağlar. Her ne kadar literatürde kriminal bilgisini ve bölgesiyle ilgili araştırmalara yardımcı olan uygulamalar mevcut olsa da, bildiğimiz kadarıyla Android cihazlardan çekilmiş fotoğrafların konum bilgilerinden faydalanıp görsel olarak bir harita çizen bir uygulama mevcut değil. Google Photos uygulaması kullanılarak Android cihazları ile çekilen resimlerin gün, saat, konum bilgisi gibi bilgileri güvenlik nedeniyle hafızada saklanmaktadır. Geliştirdiğimiz uygulama, çekilen fotoğraflardan gelen bilgi doğrultusunda kullanıcının güzergahını belirlemektedir. Farklı zaman ve farklı konumlarda çekilen fotoğralar, adli bilişim incelemelerinde kullanılabilirken, geliştirilen bu uygulama Google Harita özelliği ile kullanıcının hareketlerini görselleştirmektedir.

Anahtar Kelimeler: Mobil, Mobil Adli Bilişim, Android, Adli Bilişim Analizi, Konum Bilgisi, Güvenlik, Google Harita, Meta veri, Google API, Resim.

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABBREVIATIONS

Abbreviations

CDMA: Code Division Multiple Access.

GSM: Global System for Mobile Communication.

DFIs: Digital Forensics Investigators.

SDK: Software Development Kit.

IDE: Integrated Development Environment.

API: Application Programing Interfaces.

GPS: Global Positioning System.

1. INTRODUCTION

With more services and functionalities to offer every user, smartphones have become more important than ever before. The market for smartphones has continually increased in the last few years, and in the fourth quarter of 2010, 401 million units of cell phones were shipped [1].
The uses of smartphones have gone beyond telephony to a broad range of applications. The user interface evolved and now typically consists of a touchscreen with an alphanumeric keyboard. Some smartphones run an operating system with an open application programming interfaces, allowing users to install, play and delete applications that were developed by third parties (Developers) [2].

Smartphones offer a variety of connectivity methods, for example Bluetooth, WLAN, GSM, HSDPA, UMTS, (for wide area mobile access), USB (for connecting and transferring data with a local computer), etc.

Smartphones offer a variety of connectivity methods, such as Bluetooth, WLAN, GSM, HSDPA, UMTS (for wide-area mobile access) and USB (for connecting and transferring data with a local computer). It is obvious that they have become part of our daily lives because they hold both corporate and personal information and can collect and save substantial amounts of data, including SMS text messages, GPS location, E-mails, multimedia and other personal/public information [3].

Digital forensics is a fast-paced and exciting field that greatly influences almost all fields, such as civil litigation, corporate investigation and national security. Mobile forensic counts as one of the fastest evolving and growing digital forensic disciplines, which provides many opportunities and challenges. Android forensics is one of the main challenges that involve the analysis and acquisition of data from devices. Understanding both the tools and the platform is essential for effective investigation [3].

Every day, nearly 40 million photos are uploaded to the internet, with just about 136,000 being uploaded to Facebook alone. As smartphone cameras started becoming more powerful and common, people always try to take photos of what they like. Moreover, the ease of using smartphones has a positive influence on the number of photos taken.

In this thesis, a tool is developed for use in forensic analysis to obtain hidden data from photos captured by an Android mobile device. The data and information stored within a taken picture are called metadata. This metadata will help determine a wealth of information like location, date and the type of the device that captured the photo [2, 3]. This software is built to help obtain the metadata, which would be helpful for investigation and display it in an easy and comprehensible manner. This will show the insights of the pictures and the events surrounding them. It should be noted that the device will store the date, time and geolocation of each captured photo only if the device's location services are enabled [4]

2. LITERATURE REVIEW AND RELATED WORKS

In general, many websites (like Google Images, Photo Location or Metapicz) and software (Geotagging [Android], My photo location [Android], Photo GPS Extract [Windows], GeoSetter [Windows] and onAround Immageo [Windows]) can find the location of photos based on their geolocations [5]. These websites and software depend on the similarity of the shape of the photo to give the approximate geolocation. Furthermore, the approach will also incorporate other ways of guessing the location of taken images by trying to find the geolocation from other applications [5].

Google Images accomplishes this by analysing the submitted picture and constructing a mathematical model of it using advanced algorithms. It is then compared with billions of other images in Google's databases before returning matching and similar results. When available, not only can Google find the similarity of the same location as in the picture, but it can also use the metadata about the image to find the location [6].

Other research has been conducted with the same idea, but different approaches and results were obtained. One of the tools, ExifTool, works with metadata in a wide variety of files, not only images. The biggest challenge of ExifTool is that it is built to run and work as a command line. Thus, working on it will not be that easy, and it will be difficult to interpret the results. Our tool is developed with a simple GUI that helps the user to use and conduct a search on the file [7]. In ExifTool, the user has to manually filter the information about the location and other data needed for investigations to work on the presented data and visualise it. Once the user has filtered this data, then the output needs to be saved to a file, which can only be read by another tool to display the information in an understandable format. To understand the geodata, location and map tools will be required. The developed product in this study will obtain the images and then extract and convert the metadata of each image in a simple way. It will then work on the geodata, which is stored in the metadata, to display the result on the map depending on longitude, latitude, date and time.

Specifically, the benefits of this developed tool over ExifTool are the ease of use and speed. ExifTool requires the completion of several manual steps to extract the metadata and show the results in a visualised map. On the other hand, in our software, all the processes (uploading the image, extracting, converting and displaying metadata information in GUI windows format, getting the location on the map for each image, finding routes between images and calculating the distance of the trip) will be carried out on the click of a button [6,7]. Overall, our tool is designed to be more automated.

2.1. Similar Developed Applications

2.1.1. Forensic Analysis of Geodata in Android Smartphones

The authors worked to develop a tool to analyse geodata on a smartphone. The approach works with a simple description of app-specific geodata information that has been stored in a smartphone that enables the presentation and extraction of all relevant geodata. This tool was developed to be compatible with an Android device [8].

The focus of the author's work is the use of geodata on an Android device to provide forensic analysis of a person's information to investigators by capturing photos at specific times. The more geodata available, the better a location history can be reconstructed. The authors implemented a tool, called Android Forensic Toolkit that analyses and identifies the location of an Android device based on apps that work with GPS.

The developed tool was implemented with a data acquisition module that has been implemented in C#, which works with Android 2.2 on the HTC device. The tool will extract, convert and show modules of the geodata of each app that is connected to the GPS of the device [8].

Primarily, this tool will obtain geo-information of each installed app connected to the GPS in the device and then convert the geodata stored in the apps to show the exact location on the map.

A disadvantage of this tool is that it only works with Android version 2.2 and HTC devices, leaving out the many other available Android versions, operating systems and devices that use GPS-connected apps. However, the developed tool can work with any Android systems and devices that can connect to GPS because the photos, which will be captured in Android devices with GPS capability, will store geodata information inside the metadata of the pictures [8].

2.1.2. Image Mapping through Metadata (Caleb Riggs, Tanner Douglas and Kanwalinderjit Gagneja)

In this paper, the authors worked on analysing the images. In any investigation, the image metadata is the most important. Therefore, the authors developed a tool to extract metadata from images into a folder to help the forensic analysis community [9]. The developed tool uses Java libraries and Google Maps API. Upon retrieval of the metadata, the tool displays the information in a user-friendly manner to support investigators.

The authors discovered that the tool used various methods to obtain information, including people and object recognition. One other method discovered was the use of the object size in the images to discover information (metadata) [9]. All the data inside metadata will help to find useful information, such as geolocation, to help investigators to analyse and understand the events in the photo.

In developed software, Java acts as a backend to work with the metadata in the image. The software will then extract the metadata inside the image such as geolocation, data taken, the device took the pictures, time the picture was taken and where it was taken. This data is then displayed clearly on the map using Google Maps API [9].

However, our tool goes further than the developed tool by enabling the device to also find the routes between each image from first (earlier date) to the last image [9].

2.1.3. Location Identification of the Individual Based on Image Metadata (Ramesh Kumar P, Ch Srikanth b and KL Sailaja C)

In this article, the authors used a methodology to locate an individual from a video and images using metadata. The aim was to identify the longitude and latitude of the video and images to obtain the location where the images and video were uploaded using a device that has a location GPS service such as an Android device [10]. The output of the experiment was a map displaying the exact location of the captured images with zooming options to help find the location [10].

The authors used ExifTool, which is a Perl library with command-line CMD application to write, read and edit the information of the metadata. This tool ExifTool supports many formats of metadata, including GPS, XMP, EXIF, JPEG, Photoshop and AFCP.

They adopted python's geolocation API to work on the extracted metadata to find the location based on the longitude and latitude values. The module consists of many codes, such as geocode, that help the user to find country, street and states [10].

The disadvantage of this method is that it uses ExifTool, a free tool developed by Phil Harvey, to extract metadata from many different images formats, and it then uses the Geo Location API as a separate tool to find the location of the geodata extracted with ExifTool.

In our tool, C# codes are used to get, extract and convert metadata to a readable format that will work directly with Google Maps API in one software to show the exact location and date of each image on the map with every details [10].

2.1.4. Assessment of Metadata Associated with Geotag Pictures (Anand Nandipati)

The main goal of this research was to collect and study information of metadata with geotag pictures uploaded to websites like social networks. The author worked on the tagging service with photos, which become a wide adaptation among Web 2.0 services. Keywords (tags) help to mark ownerships, retrieve resources and browse to create a rich network, which is good for investigations [11].

The author conducted a survey of people who use geotags on pictures on social networks and found that around 25% (1 in 4) pictures show recognisable geographic features and will be found by a geotag to give a better result [11].

The main challenge in this method is finding and understanding the semantic content of each captured photo and determining the possibility that they have a recognisable geographic in it to find the exact location.

In our tool, as soon as an image has been uploaded, the tool will extract the exact geolocation inside the metadata of the image, which makes it easier to show the point on the map [11].

2.1.5. MapExif: An Image Scanning and Mapping Tool for investigators (Lionel Prat, NhienAn LeKhac and Cheryl Baker)

In this paper, the authors developed a mapping and image scanning tool to aid investigations. The tool scans all files in the directory and displays photos based on Google Maps filters. It will also show the user similar photos with the same GPS coordinates and time to inform the user about photos that have no GPS coordinates.

The developed tool can run on any operating system without installation, making it portable. It also works with read-only environments, which is good for forensic results [12].

MapExif works with SQL and PHP codes and can work on a single computer with local data to extract the metadata inside every photo to find the geolocation of each image.

While both this tool and our tool find the exact location on each photo, our tool will find the routes between photos at the same time [12].

2.1.6. Sharing, Discovering and Browsing Geotagged Pictures on the Web (Carlo Torniai1, Steve Battle and Steve Cayzer)

In this paper, the authors discovered a way to use geotagging features available on websites, depending on applications that use geotagged pictures and then implemented a prototype system to help and provide a way to:

- Use geotagging metadata to discover pictures
- Publish geotagged pictures
- Show a spatial relationship between geotagged pictures

By using the compass heading information and location embedded in the RDF description, an algorithm was discovered to identify the spatial relation between pictures [13].

The test of metadata of picture discovery and sharing has been implemented, which allows the user's photo to be combined with other users' photos. They have shown that the expressed RDF based on geographical metadata will enable a server to find, discover, browse and link geographically related images effectively [13].

Their approach finds the geotagging images using different websites and applications (such as Flickr, Picasa, Zoomr and GeoRadar), which can share photos and organise them by geotagging

features depending on other services to display them on online maps (such as Yahoo Maps, Google Maps/Earth and Greasemonkey). The advantage is that it can find relations between tested photos. Other advantages are:

- Simple expression of country, city, area or place order
- An easy way to create tag-like format
- Their prototype was useful to prove the concept, but it was not suitable for real deployment.

In this paper, the authors used websites and other online tools to find the location of each image, which has tagging features depending on some information in the metadata and places of each images which have been captured in that location. However, the tool in our research will extract the longitude and latitude inside the metadata of each image and then convert them to a readable format, which makes C# codes work with them easily. It will then present the exact location with the help of Google Maps API on the map [13].

2.1.7. Image Mapping through Metadata (Caleb Riggs, Tanner Douglas and Kanwalinderjit Gagneja)

In this paper, the authors worked on analysing the images. In any investigation, the image metadata is the most important. Therefore, the authors developed a tool to extract metadata from images into a folder to help the forensic analysis community [9]. The developed tool uses Java libraries and Google Maps API. Upon retrieval of the metadata, the tool displays the information in a user-friendly manner to support investigators.

The authors discovered that the tool used various methods to obtain information, including people and object recognition. One other method discovered was the use of the object size in the images to discover information (metadata) [9]. All the data inside metadata will help to find useful information, such as geolocation, to help investigators to analyse and understand the events in the photo.

In developed software, Java acts as a backend to work with the metadata in the image. The software will then extract the metadata inside the image such as geolocation, data taken, the device took the pictures, time the picture was taken and where it was taken. This data is then displayed clearly on the map using Google Maps API [9].

However, our tool goes further than the developed tool by enabling the device to also find the routes between each image from first (earlier date) to the last image [9].

3. ANDROID PLATFORM

Android has an open-source code for smartphone devices that are based in and managed by Linux 2.6 and Open Handset Alliance, a group of mobile device software vendors, carriers and component manufacturers. Android has made great changes and improved smartphones, and in the area of forensics in the last two years, this system has become the second-largest smartphone system in the world, capturing 26% (61.5 million) devices subscribing to Android.

Android is a feature-rich, fast-growing, and exciting mobile platform. The combination of the connectivity, features, and popularity of using Android will lead to a growing need for Android forensics, while the value and difficulty of mobile forensics are increasing. It is obvious that Android is an open source that is an ideal platform for working on forensic analysis [14].

3.1. Android Features

- The first core of Android was designed to be online from the start, whether through wireless networks or cellular networks such as Code Division Multiple Access and Global System for Mobile Communication (CDMA/GSM). Smartphone devices that work on Android operating systems allow the sending and receiving of phone calls and messages. Many other services are used with the cellular network, typically using a touch screen.
- The second core of Android devices is the ability to download and install software from any open-source market, and specifically, the Android play store. This counts as one of the main features for Android users because using this feature can extend the functionality of the device and is useful for forensic analysis. The downloaded applications can be a rich source of information [15].
- The third core is that the user can store and save personal data on the user device, which will be the basis for forensic work. Android-based devices prefer to use NAND memory and SD card (FAT32 file system) as their default storage for external memory, so it can store a large amount of data [15].

3.2. Android Architecture

The Android operating system is a pyramid of software elements divided across five areas with four main layers as shown in Figure 1.

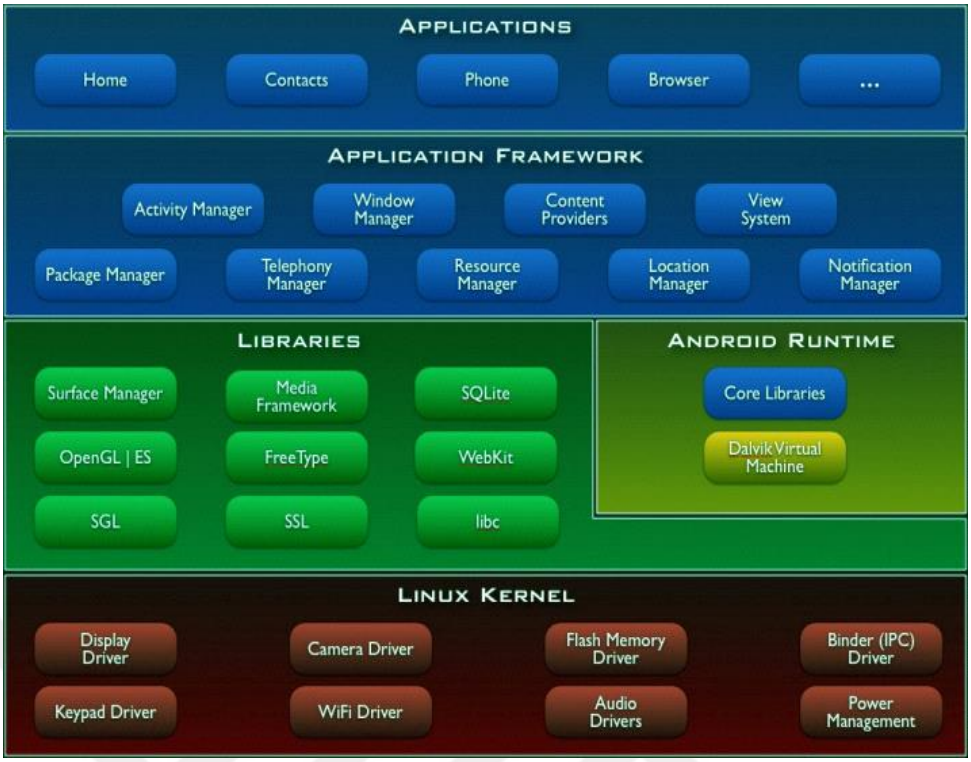

Figure 3.1. Android Architecture

a) Linux Kernel: At the bottom of the Android architecture design is Linux; Linux 3.6 has approximately 115 patches. This shows the level of abstraction between hardware devices and gathers important hardware drives such as keypads, cameras, and display. The Kernel manages everything in Linux, and it is effective in the vast array and networking of device drivers [16].

b) Libraries: Above Linux Kernel, there is a collection of libraries such as the SQLite database and the open-source web browser engine Web Kit, which is useful for sharing and storage application data. The libraries are used for recording and playing audio and video and SSL libraries are handling the security of the internet. These libraries, specific to Android development are called Java-based libraries. Android library categories include the application framework, user interface building, database, and graphics [14].

The following are the key core of Android libraries available for the Android developer:

- Android.app: the cornerstone of every application and provides access to the application model

- Android.content: simplifies content access, messaging, and publishing between application components and applications
- Android.database: used to gain access to published data by providers and contains SQLite database management classes
- Android.os: helps applications get to standard operating system services, which contain system services, messages, and interposes communication.
- Android.text: used to manipulate and render text on every device.
- Android.view: builds the fundamental blocks of the software user interface.
- Android.widge: a collection of pre-built user interface components such as labels, radio buttons, buttons, and layout managers
- Android.webkit: a collection of classes, which intend to give access to web-browsing capabilities to build applications.

c) Android Runtime: The third part of the architecture, also found in the second layer is called Android runtime. Android runtime sets a key complex called the Dalvik Virtual Machine, which is a type of Java Virtual Machine designed specifically and optimized for Android. Dalvik VM uses core Linux features such as multi-threading and memory management, which is fundamental in Java. Also, it allows all Android applications to be run in their instance and process. Android runtime tools are a collection of core libraries that enable the developers of Android applications to write Android code using Java programming language [14].

d) Application Framework: This layer prepares a higher-level service for every application in a form that is coded by java classes. Developers can use this service in the applications they have already developed.

The key services of the Android framework:

- Active Manager: controls all parts of the application activity task and lifecycle
- Content Providers: allow all applications to share and publish data with other apps
- Resource Manager: provides access to every embedded resource that are non-code applications such as strings, user interface layouts, and color settings
- Notifications Manager: allows all the applications to show notifications and alerts to the user
- View System: a set of extensible views used to develop the application user interface.

e) Applications: in the top layer, all Android applications can be found; the user writes the application and installs it in the application layer, i.e., the installed application: browser, games, contact book, and so on [14].

3.3. Android Security

Android has industry-leading security and the availability to work with developers and the implementers of the device to keep the ecosystem and platform safe. A robust security model is fundamental to enable a strong and high-level ecosystem of applications and devices [3], which are built and supported by cloud services [16]. Android has been subjected to a strong security platform as shown in Figure 2

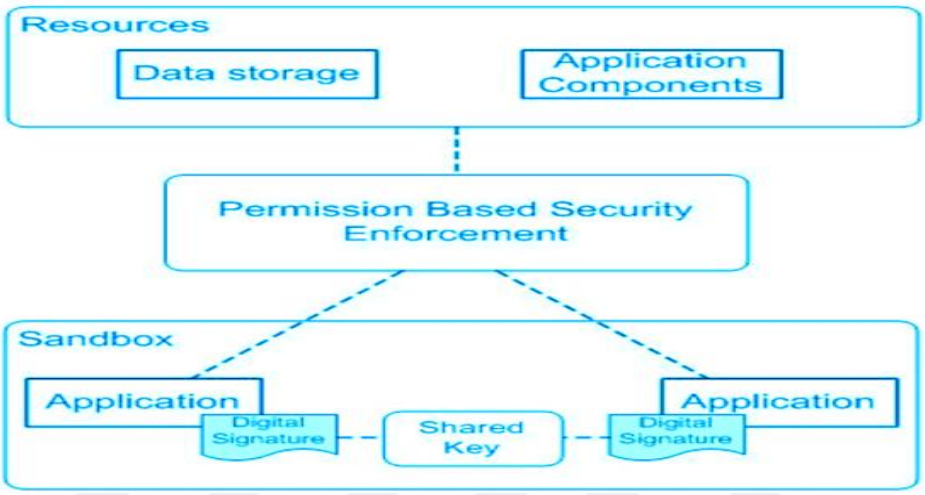

Figure 3.2. Android Security Process

The key components of Android security include:

- Design review: after the developer creates an Android application with a rich and configurable security design and model, every major feature of the Android platform is reviewed by security resources with the right security system in the architecture of the Android system [16].

- Code review and penetration test: during development, the Android platform has a strong, secure open-source component. With this vigorous security, possible susceptibilities and weaknesses are shown. The developed system will be performed by external and high-level security experts upon release.

- Community review and open source: Android's open source enables any interested party to provide a security review on the applications. In addition, use the technologies of open source to significant high-level security review such as Linux kernel.

- Monthly security update: the team of Android security provides updates monthly to Google Nexus and every partner device [16]

4. FORENSICS ANALYSIS

The Android operating system (OS) is one of the most-used OS in mobile devices around the world. Digital forensic investigations can start when a crime has occurred. Mobile devices likely need to be examined to work on case data. The role of digital forensic investigators (DFIs) involves learning techniques and opportunities to proliferate and expand into every corner of businesses, entertainment and communications in which it deals with many devices, such as tablets or cell phones, with a common OS like Android. Digital forensics investigators work on Android devices during many investigations and have several models and techniques with which to acquire data from Androids [17].

Forensic techniques used on Android devices are either physical or logical in nature. Physical techniques target and then locate the physical storage directly, which means that this technique does not depend on the file system to gain access to data. Importantly, physical techniques provide access to much important deleted data because the file system can make data obsolete or deleted but does not delete the stored medium unless it is necessary. These techniques directly work with the storage medium, and it is possible to recover both obsolete and deleted data. Logical techniques extract allocated data and gain access to files. Allocated data is the wanted and necessary data that is not deleted and can be found on the file system. An exception to the logical definition is that some files, such as the SQLite database, can be found and allocated along with the deleted records still in the database [18].

In Android forensics, logical techniques gain direct access to the OS and file system at a less effective and more abstract level than traditional logical techniques. These two techniques depend on the content providers that are built into the software development kit (SDK) and Android platform.

In this research, we developed and explained a tool that is used for forensics work to get data/time and geolocation from Android platforms. The forensics tasks that are applied to the Android platforms are described in the next section [18].

4.1. Comprehensive of Geodata for Digital Forensics:

Location information with metadata can be useful in forensic investigations. The use of smartphones has increased in the last few years and has led to the development of applications that are aware of the location. Some of the applications became more popular because of these features with geolocation.

In a recent study on smartphone platforms, the target of 50,000 apps showed that 25% of them would request access to the phone for its location and 15% of other applications. They even request fine-grained access to location [8].

Some example of apps that are widely used for geodata include the following:

• New generation smartphones will be developed with high-quality cameras with geotagging features. Those cameras store the information of geographical position (longitude and latitude) in the image's metadata.

• Smartphones' OS comes with several pre-installed applications, which are related to the navigation and map system.

• Third-party applications, which are based on location features, become more popular as the app allows the user to share location and cities, for example. Twitter works with microblogging services to get location while using the application. Similarly, Facebook, which is the leading social media platform, had 6.5 million users in February 2011 that worked with location geotagging.

The log of those applications that are based on geolocation will be stored with some information and geodata in smartphones. Working and analysing recorded log and information will be very useful for investigating in forensic analysis [8].

To analyse an Android phone's geodata, a modular diagram of geodata is shown in Figure3. However, the diagram focuses on smartphones based on Android OS and can be extended to different platforms.

All data from the Android smartphone (apps data, images and videos) need to be transferred to any type of PC, which makes all the applications under the administrator user. To gain access to these applications, different processes and tools needed [8].

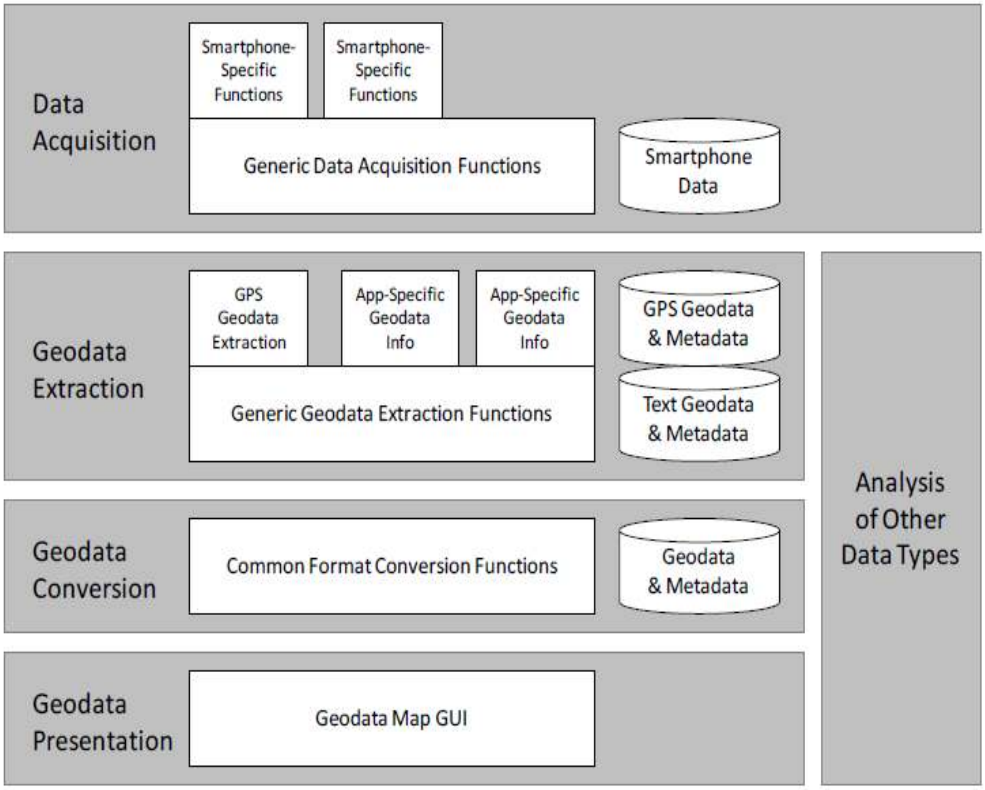

Figure 4.1. Android Geodata Architecture in Forensic Analysis

5. VISUALIZING THE PATH OF A PHOTO TAKER FROM IMAGE METADATA

This tool aims to obtain data from photos taken in Android mobiles and then show the metadata, such as date, time and geolocation, in a specific tool that will be developed with C# and connected to Google Maps to display on the map and get the routes between the captured photos from the first photo to the last one depending on the date and time of the captured photo. It will also present metadata, such as geolocation and captured date, of the picture in the tool in a separate part of the windows form. Moreover, it will find the distance (in kilometres) between the routes that the owner of the device took to different places to capture photos. This tool will be developed for use in forensic analysis to obtain invisible data from photos captured by an Android mobile device [19]. However, the device will store the data, time and geolocation of each captured photo only if the device's location services are enabled.

5.1. Images and Formats

Images in our society have become so commonplace that people use smartphones, which have better cameras. Moreover, the number of pictures taken has increased in the last few years. With the availability of all these pictures, it was a good point to target and work with image formats in our program. The most popular image formats are JPG, PNG and GIF [20].

JPGs are the most important format in our case because they are the default and main format for most cameras, as they work accurately with other types. PNG images are defined and known for being very high in quality with transparency features that are used by many image programs [21]. GIF pictures have transparency features. Moreover, the images can be reduced in size and their colors can be customized, making GIF suitable for online graphics.

5.2. Metadata

Any data that can be described by data is called metadata. In documents, the metadata will store and save information, such as who created the document, when the document was created and saved and the last date of modification, as well as other details [22].

In images, the metadata will also store information, which can be categorised into three different types:

- Technical
- Descriptive
- Administrative

Technical data include the information about the type and the model of the camera that took the photo and the settings used in that camera, such as ISO settings that work with environment lighting, which can be used to draw conclusions on the environment around the captured photo. Technical data also store other information such as aperture speed, shutter, focal depth and more, although these may not be valuable information in forensic analysis.

The most important technical data is GPS coordinator data. Nowadays, camera phones have become more usable as primary cameras because the main cameras of new generation phones have more features and sensors. At the same time, GPS data will be stored in each image captured with a smartphone [23].

The GPS data is useful in forensic analysis investigations for different cases, and GPS data was the main point focused on in our project. On cameras of smartphones, the model of the phone will be stored in the metadata, which will be helpful because it can verify who took the photo. For example, a photo taken with SAMSUNG SM G950U is S8 SAMSUNG GALAXY.

The administrative data includes the information added by the administrator, such as the owner of the file, and the permissions that it has. This type of data was not useful for our case in smartphone investigation. The benefits of this tool will include the following advantages for digital forensics:

- The metadata, including geolocation and capture date, aids in locating criminals using a phone.
- Obtaining the location of the captured photo on the map allows investigators to find the location of the crime.
- Tracing the routes between photos may be used to investigate the location of the mobile user.
- The dates and times stored in the photo properties will allow investigators to know the exact date of a crime.
- The metadata obtained from the photos is presented in a separate window, and the tool calculates the distance (in kilometres) of the route taken by the user.

5.3. Material and Methodology

Microsoft Visual Studio C# 2015 is used to write codes to obtain metadata from photos. It is a tool developed by Microsoft, called an integrated development environment (IDE), and for developing computer programs, web service and mobile apps, among others.

C# uses Microsoft software development platforms, such as Windows form, Microsoft Silverlight, Windows API, Windows Store and Windows Presentation Foundation, which can produce and manage code and native code [24].

Google Maps Platform API is used to show Google Maps on the tool for displaying the location of each photo and tracing the routes between them [25].

Google Maps Platform API is a set of SDK and API that helps developers to link Google Maps into a windows application. Web pages and mobile apps are used to obtain and collect data from Google Maps that give the best option to get routes and direction with high-quality and real-time traffic updates to the user. An Android mobile device with the GPS (location service) enabled is used to record the photos' geolocation data. Also, a Windows 10 computer with Intel Core i5 processor is used.

The method used is divided into three categories:

- Obtaining metadata from the captured photos.
- Displaying the location of the captured photos on Google Maps using the Google Maps API.
- Tracing the route between the captured images from the first to the last photo.

When the GPS location service is enabled, Android devices store details in the photo properties, which the program can then use to display the location and routes on Google Maps [26].

It should be noted that metadata cannot be obtained from photos if:

- The metadata is removed from photos by another application or user.
- Photos are transferred via instant messaging applications in which the metadata is deleted automatically.
- The metadata of the photos that have been zipped will be removed.

Once you capture or record the picture with a specific device or camera, the camera stores the data and create a path that is linked to the camera and the photo. Once that photo is transferred by an application (instant message) to another device, the link between the photo and the camera of the original device is lost. The camera from the new device is unable to link the photo to the original point of view of the original camera that has captured the photo. Furthermore, if the metadata are deleted, there is no way to retrieve the metadata of the photo. Typically, an EXIF stripper does remove the actual bytes containing the data. If metadata is lost through an EXIF stripper, the file size usually becomes smaller, so there is no way that they can be recovered [20].

5.4. Algorithm and Implementation Details

5.4.1. Overview

In this thesis, many functions/methods have been used and written to show all that was composed by visual studio C#, which should manage and control the link between components. In this tool, many existing libraries, such as Exiflib, ExifNET, Geolocation, GMap.Net.Windows,

GPS.Net, Measurements, MetadataExtractor, photo.exif, System.Data.SQLite and XMPCore, were used to get and extract metadata and display it.

Visual studio C# works as an object-oriented and powerful language, which is built on the .NET platform that provides APIs and code analysis. The program will find and search on images in windows browser and filter them to upload the images with all their information to the windows form by using specific code, as shown in Table 5.1. The images will be stored in the temporary memory of the computer where the uploaded images are linked with C# codes.

Table 5.1. Codes of Uploading and Filtering Images

Steps	Uploading and filtering images
1	foreach (string img in files)
2	uploading picture as {PictureBox pic = new PictureBox();
3	pic.SizeMode = PictureBoxSizeMode.StretchImage;
4	display on windows form pic.Image = Image.FromFile(img);
5	if (x > this.ClientSize.Width - 100)
6	{ x = 20;
7	y += maxHight + 10; }}

Libraries, like Exiflib, ExifNET, MetadataExtractor, photo.exif, System.Data.SQLite and XMPCore, will work with images stored in the temporary memory to get and extract the information in each image and convert them to readable text that will make it easy for C# to connect the geolocation information with Google Maps API to display them on the map in the windows form (GUI).

Definition of some used Libraries:

- Exiflib: 313,672 downloads, used for reading JPG Exif data in a simple way without unnecessary and heavyweight installation.
- ExifNET: 3,356 downloads, used to read exif metadata of any type file of images [26].
- GPS.Net: 2, 039 total downloads, used to provide simple objects on the map.
- Geolocation: total downloads of 102,012. A library used in C# to calculate the cardinal direction and distance between two different points of coordinates and then provide that longitude and latitude around the exact coordinate to allow simple LINQ or SQL selection of any point of locations in the given radius [27].
- GMap.Net: a powerful open source platform used in visual studio C# to enable the use of geocoding, map and routing from Yahoo, Google, ArcGIS, CloudMade, WikiMapia, Bing and MapQuest in presentation and windows form. Many other map applications and websites also support caching and run on mobile operating systems.

- Metadata-Extractor: A Java library used in visual studio C# for reading metadata in all types of images [28].
- Photo.exif: a library used to show metadata in different types of images like TIFF, JPEG and WAV.

5.4.2. Activity Diagram

The main flowchart of the developed tool is shown in Figure 4. The user simply clicks on the upload button to select the direction of the folder, which contains the images for investigation. It will then extract the metadata of each image to add them to an array to work on it later for geolocation. Once all the metadata have been extracted and converted, the C# codes will render the information in the windows form. It will then connect each location details (longitude and latitude) based on date and time and display on Google Maps with all information like date, time, longitude and latitude.

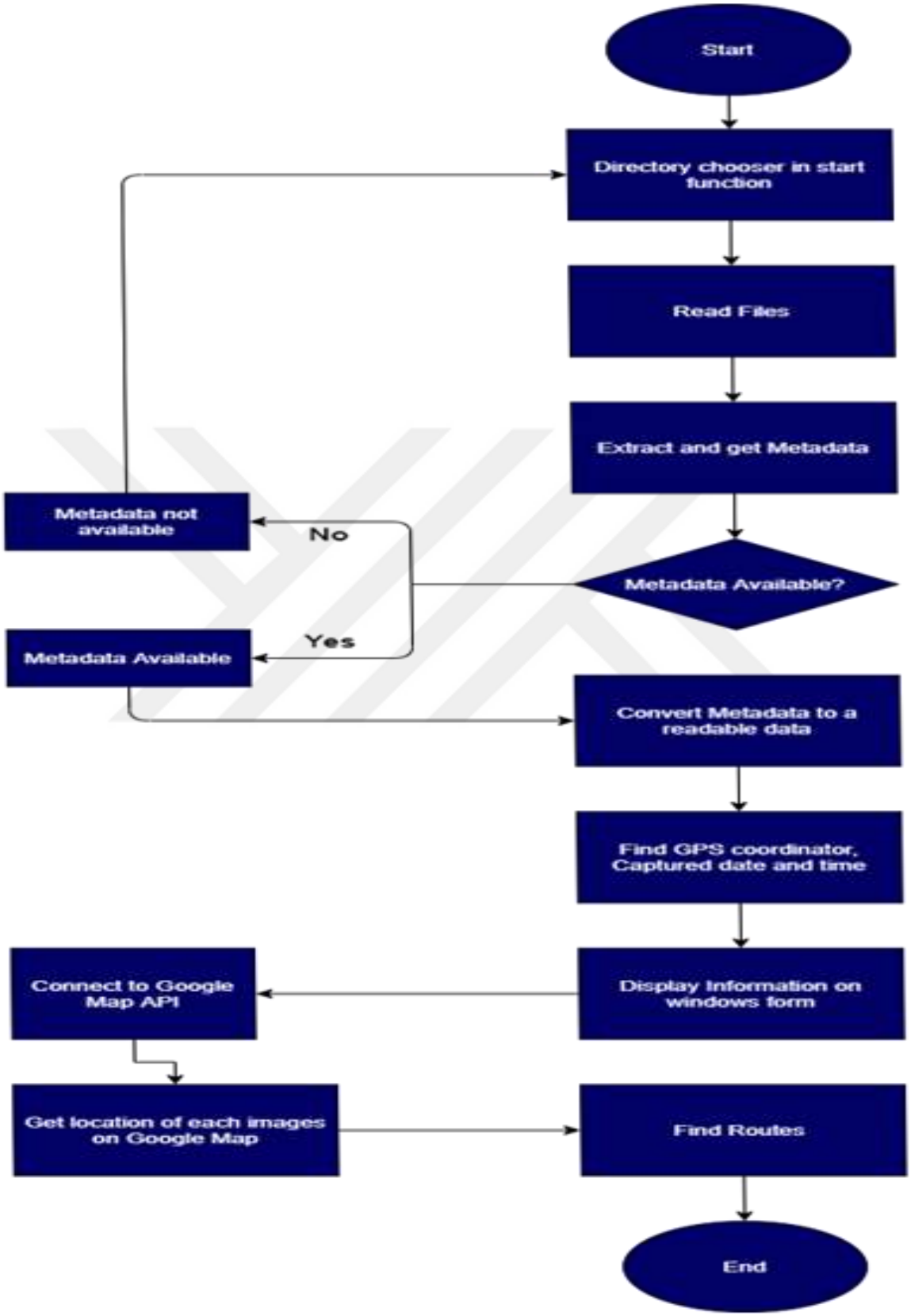

Figure 5.1. Forensic Tool Activity Diagram

5.4.3. UML Function Diagram

For illustrating the tool component, we will use the UML diagram, as shown in Figure 5. In the main process, the tool will try to filter the image and at the same time extract the metadata in the image to make it easier for the compiler to convert the metadata to an understandable text, which will make it faster for the compiler to connect the geolocation information with Google Maps API depending on the date and time and display the result on the map in the windows form of the tool.

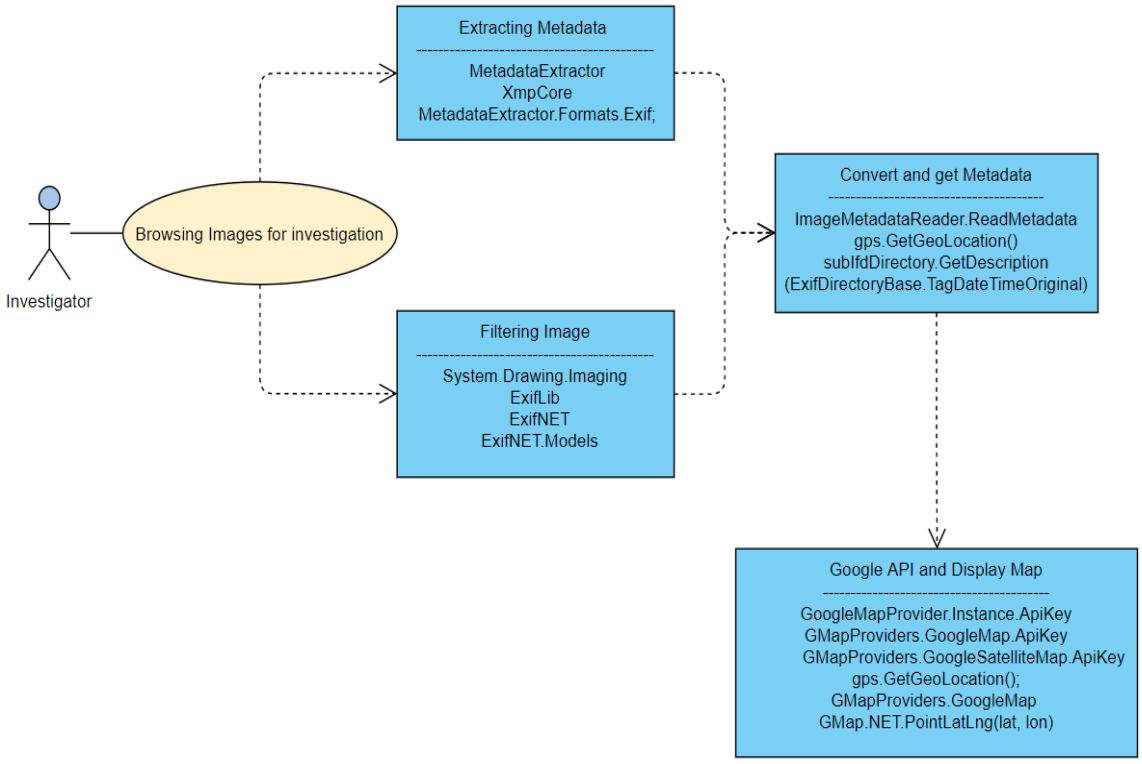

Figure 5.2. Forensic Tool UML Function Diagram

5.5. Finding Metadata

For First, we found a way to find and extract the metadata from the image. To do that, we used and called libraries called ImageMetadataReader and MetadataExtractor. These two libraries were used to find and pull metadata from many different types of images and videos such as JPG, PNG, TIFF, MP4 and PSD.

In our tool, we focused on extracting data from JPG files, which is the default file type of smartphone cameras. Using JPG alone will help to take many photos in one camera.

By using our main function, (OpenFileDialog()), to upload images to our tool and then filtering the files with (openFileDialog.Filter = "JPG files (*.jpg*)|*.jpg"), we will be able extract the metadata from uploaded images by using the (MetadataExtractor) function for investigating the information.

Table 5.2 shows the codes for running the metadata extractor function. The (array) will store the information of each metadata to work on them one by one. Then using the (GetGeoLocation and TagDateTimeOrginal) functions, it will convert all the metadata, which have been stored in our array, to be visible on the windows form.

Table 5.2. Metadata Extractor Code

Steps	Metadata Extractor
1	string[] array = data.FileNames;
2	foreach (string Ar in array)
3	defin GPS { var gps = ImageMetadataReader.ReadMetadata(Ar).OfType<GpsDirectory>().
4	FirstOrDefault();
	get Location var location = gps.GetGeoLocation();
5	varsubIfdDirectory=ImageMetadataReader.ReadMetadata(Ar).
6	OfType<ExifSubIfdDirectory>().FirstOrDefault();
	var dateTime = subIfdDirectory.
7	GetDescription(ExifDirectoryBase.
	TagDateTimeOriginal); } }

With the above codes in the table, we were able to get the exact data on (geolocation, date and time) of each image that was captured with the Android mobile, which will be useful information for investigators to know the exact date and time of the taken image, as well as the location.

Figure 5.3. shows the information obtained from printing the extracted metadata for each uploaded image in a directory.

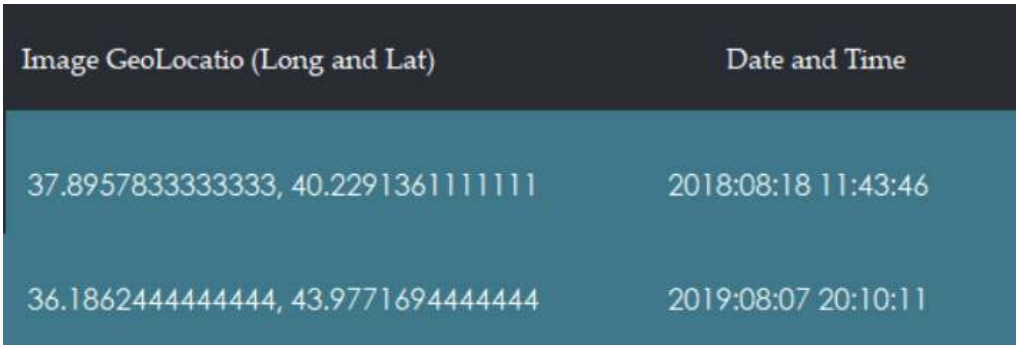

Figure 5.3. Metadata Extraction of Images

We also found a way to upload more images to the tool at the same time. We used (OpenFileDialog. Multiselect = true) because it will use the device's original file browser to select the folder. Once the directory has been selected, the user will choose multiple images at the same time to upload them in an array in the tool.

5.6. Visualize Results

5.6.1. Google Maps API

After we uploaded all the images and metadata have been extracted from them, the next step will be visualising the map result. The main idea was to use Google Maps API to help work on the Google Maps data. Google Maps API has no official visual studio C# libraries, so we obtained and worked with libraries from GitHub to run and compile the program, as the code shown in Table 5.3. Figure 5.4 shows the initial render map view of each image with added pins on the map for each one of them. The pins will go through the uploaded images and get the information from extracted metadata to find the exact location on the map.

Table 5.3. Codes to Find Location on Map

Steps	Find Location on Map
1	Define points on the map Var gps = ImageMetadataReader.ReadMetadata(Ar).
2	OfType<GpsDirectory>().FirstOrDefault();
	Get location in metadata
3	var location = gps.GetGeoLocation();
4	find Latitud double lat = Convert.ToDouble(location.Latitude.ToString());
5	find Logitude double lon = Convert.ToDouble(location.Longitude.ToString());

Another step is to make the windows format show the images where the locations were marked on the map. To display these images, we need to create some function in C#, such as file dialog, picture box and dialog result. With these functions, the uploaded images will be added to the windows format with all the information of metadata.

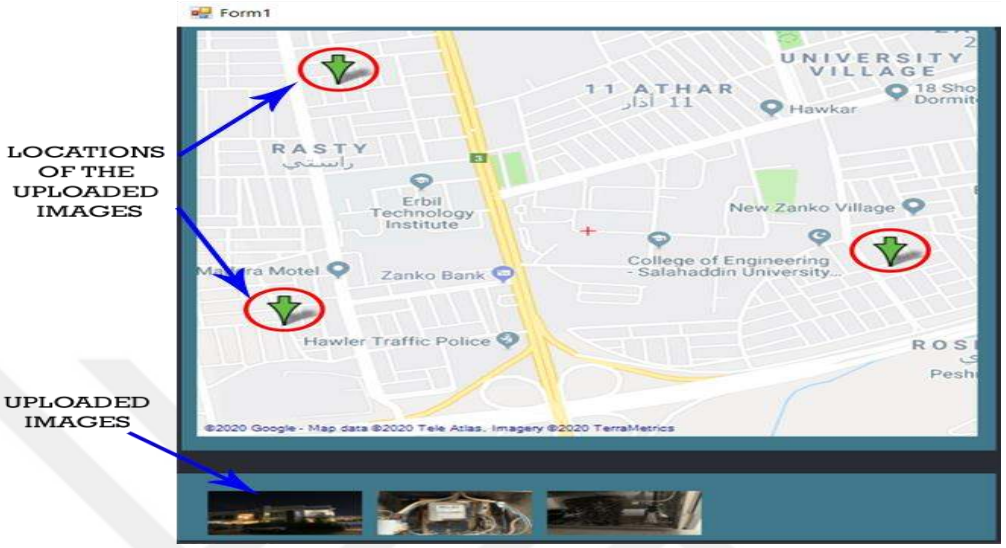

Figure 5.4. Initial Render Map View

Now that these steps have been accomplished for each location, a basic HTML file is set up. This HTML file is called the API of Google Maps and is shown in the windows format. To communicate with the map file, C# functions were created and called from GMap.NET.MapProviders. This made the C# implementation available since it will need only one step to call the library (GMap), which was previously installed in the visual studio. Three main libraries were called to add the pins as locations on Google Maps: CacheProviders, Markers and Projection. Moreover, some functions, such as zooming, initial zoon and button click, were created to manipulate the map.

5.6.2. Google Map Provider

Using Google Maps Provider functions will help to get and find the routes between images' locations easily on the map. The next step is to make the C# application show the exact routes between images to show a good result on the map. We started the process by trying to work on libraries like GpsDirectory, GetGeoLocation and GetRoute to find the direction between images from the earliest date and time to the last one. Sample code for these libraries is shown in Table 5.4. Additionally, one more function (Rout.Distance and GMap Internals) were used to find the whole distance for all the trip.

After these processes, to display the result on the map, as shown in Figures 5.5 and 5.6 many different functions and libraries were used to make the process better and faster for investigations.

Table 5.4. Codes to Find Routes between Images

Steps	Find Routes between Images
1	define route as (GoogleMapProvider.Instance.GetRoute)
2	var route = GoogleMapProvider.Instance.GetRoute (_points[i], _points[i + 1], false, false, 14);
3	put points on map ((_points[i], _points[i + 1])
4	define r as new routes var r = new GMapRoute(route.Points, "My Route") var routes = new GMapOverlay("routes");
5	routes equal to new GMapOverlay("routes");
6	routes as map.Overlays.Add

Below are the best libraries that were used for the result:

- GmapOverlay
- GoogleMapProvider
- GMapRoute
- System.Drawing
- Globalization
- Distance

Figure 5.5. Finding Routes between Different images (1)

Figure 5.6 Finding Routes between Different images (2)

5.7. Device Specification

The tool was developed on a computer with the following specifications:

- Dell Laptop Intel CORE i58250U CPU @ 1.60GHz (8 CPU)
- RAM Memory of the laptop: 8GB
- Graphic (Intel UHD) graphic family: 4GB
- Windows 10

Specs of the Android mobile used to capture are listed below:

- Huawei nova 3i Octa core (2.2 GHz, Quad core, Cortex A73 + 1.7 GHz, Quad core, Cortex A53).
- Graphic Mali-G51 MP4
- RAM: 4 GB
- GPS specs: A-GPS, GLONASS
- Android 8.1 (Oreo)

5.8. Results

5.8.1. Time Efficiency

In terms of time, our tool is very efficient. However, only the written codes were utilised, and we did not record the Google Maps API or libraries. We timed our tool many times at different numbers of images to observe the time it takes to load, extract metadata and find the location on Google Maps, as well as the routes between the images. In Table 5, a basic sample has been recorded.

Table 5.5. Time Efficiency whole Process

NO of Images	Uploading Images	Extract Metadata	Load Map	Load Routes	Total
1	0.30 MSec	0.10 MSec	0.20 MSec	null	0.60 MSec
2	0.70 MSec	0.10 MSec	0.35 MSec	0.45 MSec	1.6 Sec
3	0.80 MSec	0.10 MSec	1.20 Sec	2 Sec	4.1 Sec
4	0.90 MSec	0.10 MSec	1.45 Sec	2 Sec	4.45 Sec

The following Figures (5.8, 5.9 and 5.10) show the amount of time taken to upload images, display the location of each image on the map and then find the routes. The figures clearly show the charts' pattern referring to the time complexity in our tool.

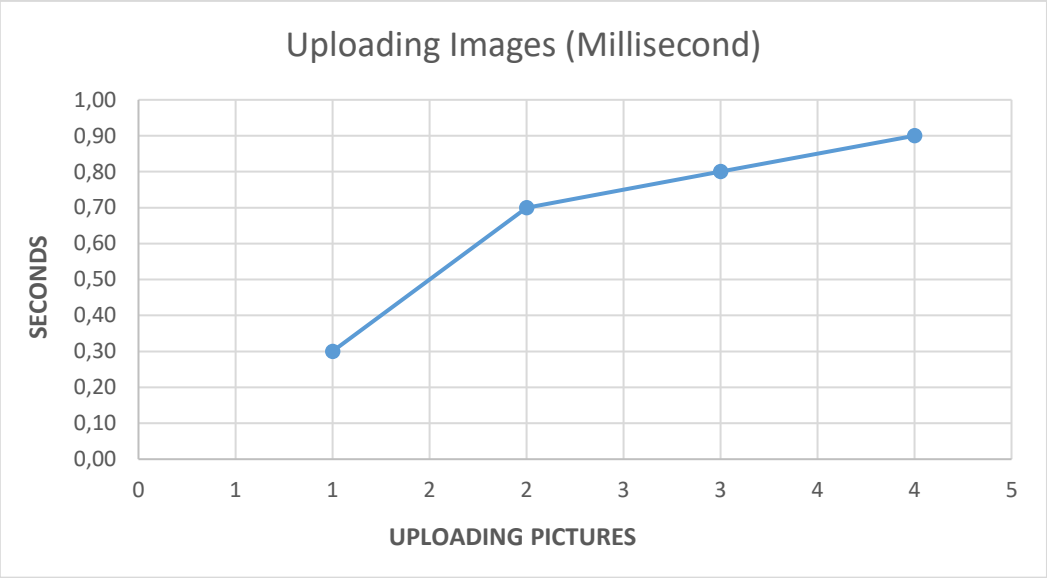

Figure 5.7. Uploading Images Time Efficiency

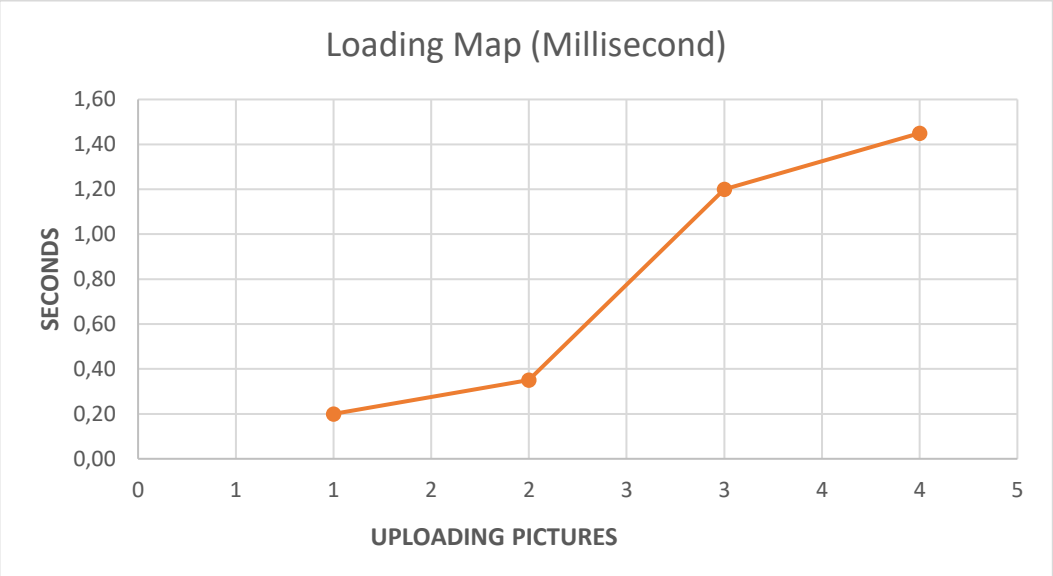

Figure 5.8. Loading Map Time Efficiency

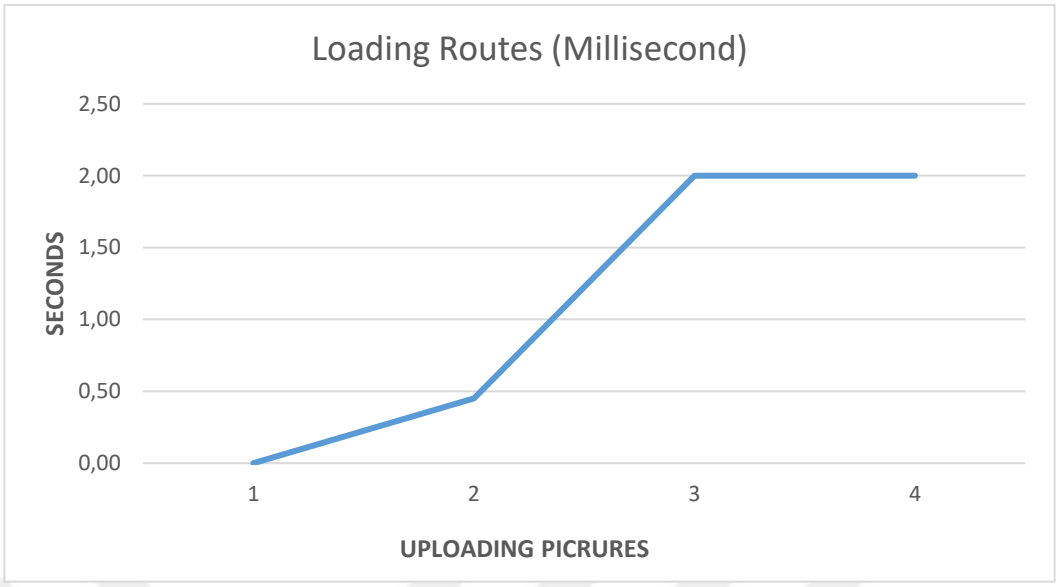

Figure 5.9 Loading Routes Time Efficiency

The above charts represent the process of loading time, which depends on processor speed, memory speed and RAM. The map loading time is limited on the network speed since it relates to Google Maps API. The following Figure 5.4 (Initial Render Map View) can be changed from time to time, which would be inferred by some outliers. These outliers can change based on the bandwidth of the networks and cache version.

The combined average load time of images, upload time, map load time and routes time are shown in Figure 5.10, in which the pattern of the chart has the same complexity as Figure 5.9 (Loading Routes).

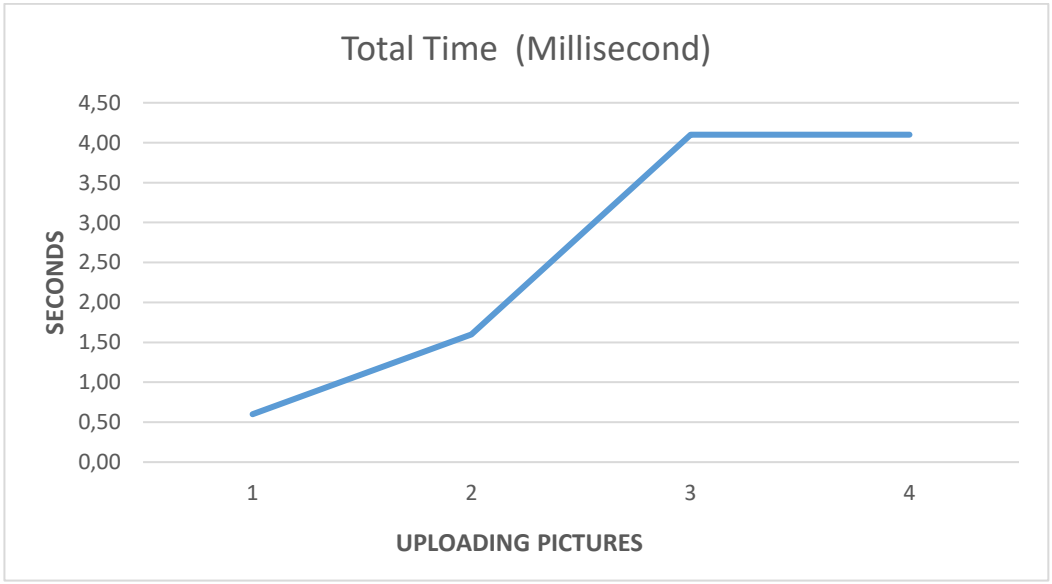

Figure 5.10. Total Time Efficiency

5.8.2. Result Analysis

The result of our tool makes us understand the information on each image that have been taken and how much the camera technology has improved over past years, making photos an important part of investigations.

In our tool, we found that images are not only important for forensic investigations, like crimes, but also useful for accident investigations because of the information that can be gained from images such as the date and the time of the taken image and the device that took the image. This makes it an available object in which many details can be found inside (persons, objects and many other things) that will be needed for forensic investigations. Metadata was another way to investigate images.

While searching and extracting metadata, we discovered that it can be classified into different types such as descriptive, technical and administrative. Our tool used C# functions and libraries to extract metadata from wanted images, and the extracted metadata of each image contains the taken date, device type, geolocation taken time, place of the taken image, etc.

The result of this work is the ability to find the routes between images needed for investigations, as shown in Figure 5.11 In the first step, the investigator adds the images to the tool to extract all the information in the image's metadata. Then with the help of this information that contains geolocation, date and time, the location of the taken images can be found. Moreover, based on the earliest time of the images, the routes between images from the first to last one can be found.

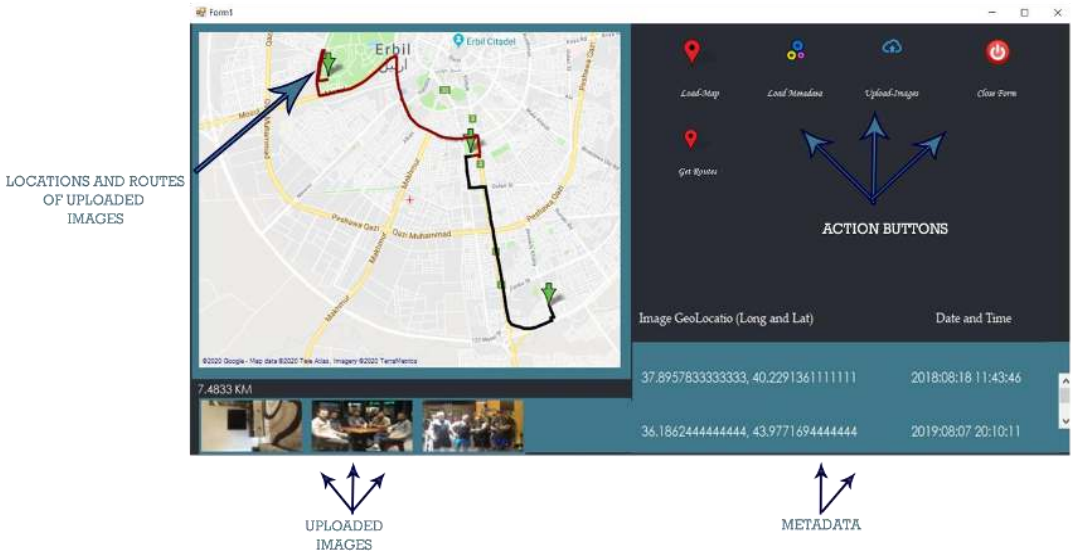

Figure 5.11. Final Result of Developed Tool

6. CONCLUSIONS AND FUTURE WORK

In many digital forensic investigations, many investigators have worked on images. Image metadata is one of the most important parts of images that contain many different types of information about the image. Photos captured on Android devices that utilize Google Images store the date/time and geolocation of the pictures in their properties for security reasons. This application depends on these images to track the user's journey. In this work, a tool has been developed to extract all the information from the image's metadata, which helps investigators to analyze every case on the images. Photos taken in different places and at different times during a journey can be used in forensic analysis. The developed tool can track the user's travel route, locate it and show it on Google Maps based on the date/time and geolocation details saved in the photo's properties.

The tool, developed in C# and connected with Google Maps API and many other libraries in the visual studio of C#, can be used to find the routes for a journey using photos taken by an Android mobile device. It shows the details (date/time and geolocation) of the captured photos on Google Maps and obtains the route between the captured photos from the first to the last photo taken. Once the software obtains the metadata from images, investigators can use this information for their work. This tool is used for analyzing and the decrypting the metadata inside captured images and then identifying the location and direction of the images. Unfortunately, the following problems were encountered:

- Some libraries had issues with Google Maps.
- A better processor was needed to run the tool faster, as it sometimes jammed.
- While running the codes, an exception occurred when obtaining the image metadata, due to the connection between the tool and the Google Maps API.

In the future, this tool can be used as an online tool to gain access to it faster and easier. The addition of features, such as the type of mobile device that captured the photo, and connecting it to other tools to obtain details from instant messaging applications, such as chats and calls, will enhance the tool's ability to find more details for investigations and forensic analysis work.

REFERENCES

[1] Kratzke, Cynthia, and Carolyn Cox. "Smartphone technology and apps: rapidly changing health promotion." Global Journal of Health Education and Promotion 15, no. 1 (2012).

[2] Aldhaban, Fahad. "Exploring the adoption of Smartphone technology: Literature review." In 2012 Proceedings of PICMET'12: Technology Management for Emerging Technologies, pp. 2758-2770. IEEE, 2012.

[3] Al-Hadadi, Mubarak, and Ali AlShidhani. "Smartphone forensics analysis: A case study." International Journal of Computer and Electrical Engineering 5, no. 6 (2013): 576.

[4] Villalba, Luis Javier García, Ana Lucila Sandoval Orozco, and Jocelin Rosales Corripio. "Smartphone image clustering." Expert Systems with Applications 42, no. 4 (2015): 1927-1940.

[5] https://superuser.com/questions/1198836/recover-metadata-information-from-photo.

[6] https://www.forensicfocus.com/Forums/viewtopic/t=12397/

[7] https://www.forensicfocus.com/Forums/viewtopic/t=9071/postdays=0/postorder=asc/start=0/

[8] Maus, Stefan, Hans Höfken, and Marko Schuba. "Forensic analysis of geodata in android smartphones." In International Conference on Cybercrime, Security and Digital Forensics, http://www. schuba. fh-aachen. de/papers/11-cyberforensics. pdf. 2011.

[9] Riggs, Caleb, Tanner Douglas, and Kanwalinderjit Gagneja. "Image Mapping through Metadata." In 2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC), pp. 1-8. IEEE, 2018.

[10] Ramesh Kumar, P., Ch Srikanth, and K. L. Sailaja. "Location Identification of the Individual based on Image Metadata." Procedia Computer Science 85 (2016): 451-454.

[11] Nandipati, Anand. "Assessment of metadata associated with geotag pictures." PhD diss., 2011.

[12] Prat, Lionel, Cheryl Baker, and Nhien An Le-Khac. "MapExif: an image scanning and mapping tool for investigators." International Journal of Digital Crime and Forensics (IJDCF) 7, no. 2 (2015): 53-78.

[13] Torniai, Carlo, Steve Battle, and Steve Cayzer. "Sharing, discovering and browsing geotagged pictures on the world wide web." In The Geospatial Web, pp. 159-170. Springer, London, 2007.

[14] A. Hoog, A.f.i., analysis and mobile security for Google Android: Elsevier, 2011.

[15] A. Developers, W.i.a., " ed: Android Developers, http://developer. android. com/guide/basics/what-is …, 2011.

[16] W. Enck, M.O., and P. McDaniel, "Understanding android security," IEEE security & privacy, vol. 7, pp. 50-57, 2009.

[17] Palmer, Gary L. "Forensic analysis in the digital world." International Journal of Digital Evidence 1, no. 1 (2002): 1-6.

[18] Curran, Kevin, Andrew Robinson, Stephen Peacocke, and Sean Cassidy. "Mobile phone forensic analysis." In Crime Prevention Technologies and Applications for Advancing Criminal Investigation, pp. 250-262. IGI Global, 2012.

[19] Diakopoulos, N., and Irfan Essa. "Mediating photo collage authoring." In Proceedings of the 18th annual ACM symposium on User interface software and technology, pp. 183-186. ACM, 2005.

[20] Anderson, Eric C., and Patricia Scardino. "Method and system for extending the available image file formats in an image capture device." U.S. Patent 6,493,028, issued December 10, 2002.

[21] Kee, E., Micah K. Johnson, and Hany Farid. "Digital image authentication from JPEG headers." IEEE transactions on information forensics and security 6, no. 3 (2011): 1066-1075.

[22] Ford, B.D., Johnson, G.M., Bray, C., Cieplinski, A., Khoe, M.L., Victor, B.M., Costanzo, B.C. and Bernstein, J.T., Apple Inc, 2014. Metadata-assisted image filters. U.S. Patent 8,854,491.

[23] Dasgupta, S., Verizon Data Services LLC, 2010. System and method for providing image geo-metadata mapping. U.S. Patent Application 12/336,606.

[24] Hejlsberg, A., Scott Wiltamuth, and Peter Golde. C# language specification. Addison-Wesley Longman Publishing Co., Inc., 2003.

[25] Kobayashi, Shinji, Tetsushi Fujioka, Yuji Tanaka, Michiyoshi Inoue, Yoshiyuki Niho, and Akira Miyoshi. "A geographical information system using the Google Map API for guidance to referral hospitals." Journal of medical systems 34, no. 6 (2010): 1157-1160.

[26] Hu, Shunfu, and Ting Dai. "Online map application development using Google maps API, SQL database, and ASP .NET." International Journal of Information and Communication Technology Research 3, no. 3 (2013).

[27] Torniai, C., Steve Battle, and Steve Cayzer. "Sharing, Discovering and Browsing Photo Collections through RDF geo-metadata." In SWAP, vol. 201. 2006.

[28] Kee, E., Micah K. Johnson, and Hany Farid. "Digital image authentication from JPEG headers." IEEE transactions on information forensics and security 6, no. 3 (2011): 1066-1075.

CURRICULUM VITAE

[redacted]

PERSONAL INFORMATION

RESEARCHER INFORMATION

[redacted]

EDUCATION

Master Degree **:** (Visualizing The Path of a Photo Taker from Image Metadata)
Firat University, Faculty of Graduate School, Department of Software Engineering, Year 2021
Supervisor: Dr. Cihan Varol

Bachelor **:** Erbil Polytechnic University, Engineering Faculty, Department of Information System Engineering, Year 2014

High School **:** Kurdistan Typical School, Erbil City, 2010

RESEARCH EXPERIENCES

- ✓ Computer Programming languages available (Visual Studio, C#)
- ✓ Computer programs you can use (Windows Server, Linux, MS Office, Adobe programs for Graphic design)

WORK EXPERIENCE

2019 – 2020 --Administrator and logistic in Oilserv company
2016 – Currently -- ID card engineer in Iraq National Government
2015 – 2016 --Lecturer in Lavin institute for computer
2014 – Currently -- Lecturer in Erbil technical collage
2013 -- Sales Employee in Soz for Mobile
2013 – 2016 -- Revenue Assurance Engineer in Korek Telecom
2012 – 2013 -- WI-FI Engineer in Tarinnet company
2011 -- Database Designer in Avary company

ACADEMIC ACTIVITIES

Paper:

1. **Database system security: Case Study** (TBD 35. Ulusal Bilişim Kurultayı -TBD 35th National Informatics Congress)

Proceedings:

1. **Team Building - Korek Telecom 2015**
2. **Team Leader Skills - Korek Academy 2015**
3. **NLP (Neuro – Linguistic programming) - TIPCONS 2017**
4. **TOT (Train of Trainer) - TIPCONS 2018**
5. **Safety and Health – Oilserv 2019**