Toward Metadata Removal to Preserve Privacy of Social Media Users

Abigail Week[1], Joshua Yee[1], Mayra Carrera[1], Kuira Edwards[2], Vicki Murray-Garcia[3], Meghann Marchello[3], Shahab Tayeb[4]
[1]AEOP UNITE, [2]UNLV STEM, [3]RET, [4]UNLV
Las Vegas, Nevada

Abstract—Social media site users are constantly uploading images onto social media platforms to document their lives for others. These users may be unaware, though, that the pictures they upload have metadata embedded within them. The metadata includes a wide variety of different types of information about the pictures and the camera used to take them, including Global Positioning System location data. After the picture is uploaded, data is collected by data mining companies that are allowed access to the data by social media sites. What these companies do with this data is uncertain and raises privacy concerns for users. This research provides insight into how much information is stored within pictures and how this may affect users' privacy. We have also supplied a programming example in the Python programming language that performs the removal of all the metadata, or just the Global Positioning System location data, from images before they are uploaded. Instead of completely stripping away the metadata, our solution allows the users to create identical copies of their images that are free of metadata. These images can now be uploaded free of information the user may not be comfortable sharing with data mining companies or other social media users.

Keywords— meta-data, image files, privacy, GPS location

I. Introduction

Social Networking Services (SNS) act as a medium for users to interact and communicate with people all over the world, and have been growing tremendously since the creation of the first SNS website, Six Degrees, in 1999. As these free platforms grow bigger, privacy concerns grow, as well, because when posts are uploaded, metadata, or embedded information, can sometimes be shared as well [1]. Metadata can include sensitive information such as the time the photo was taken, the location it was taken at, and the camera model it was taken with[2]. While certain social networks have technologies which erase the metadata, there are other sites which do not alter it. Leaving it visible allows for others to access and manipulate the data in such a way that the user who posted it is at risk. It only takes 20% of user information to conclude more than what is given [3]. Many sites provide privacy settings to keep users' posts and profiles within their group of friends [4]; however, if friends were to share the picture, the metadata attached could also be sent and endanger the original poster's privacy. For these reasons, we present a program so the user can strip metadata from multiple photos before uploading.

A. Background

The topic of privacy encompasses an enormous number of platforms and issues. This research paper concerns itself with user privacy. Specifically, we focus on online user privacy as it appears within social networks. A breach in user privacy can lead to malicious users obtaining extensive information, including current or common locations. It also can damage the trust between the user and SNS corporations [5]. In current processes, both the user and businesses have certain controls and permissions of information which the user distributes [6]. Social networks already provide options to modify privacy and general settings. Larger sites also strip metadata located in posts to protect privacy concerns. Problems arise when sites do not take precautions to protect user privacy.

They say "an image is worth a thousand words" which is usually taken figuratively. However, this statement could and should be taken literally considering there can be over 800 kilobytes worth of information on a single picture. This information contains metadata. Metadata is defined as a set of data that describes and gives information about the data itself. There are 4 specific types of metadata: descriptive, administrative, structural, and preservation. In our research, we focus on descriptive metadata, specifically EXIF data [7]. EXIF data includes technical information about an image and its capture method, such as the camera model used and Global Positioning System (GPS) location [8]. The majority of pictures uploaded are taken on camera phones which automatically gathers and embeds information, such as camera settings [9]. Sharing pictures with this information available on social media platforms can cause unwanted metadata sharing. A few mature social media sites have made it so that the metadata is stripped from the image once it is uploaded. However, SNS still have access to the metadata and what they do with this information is unknown [8] [10]. This could be a privacy concern for several users.

B. Related Works

According to Henne's research "A photo may include textual descriptions of places (location, street, city, state, country) besides GPS coordinates. Some cameras include the artist's name as well as a unique device identifier into media files [11]." Henne's survey shows that the threat was even greater as friends of friends and strangers uploaded those pictures because as the pictures spread, the embedded location data spreads. The results show that 98% of the participants

were aware of privacy threats while uploading. However, even though the participants were aware of threats while uploading, only 1.4% believed the photo privacy threat was very low.

Furini and Tamanini surveyed social media users on location data privacy [12]. Users of location-aware applications are less worried than those who do not use location-aware applications. Women seem to be more worried than men. When talking about photos, an "angry" reaction is predominant. People do not like to be contacted, but if the service asks for their authorization and provides benefits, users are willing to be contacted. Men find the service "Interesting," whereas women find it "Intrusive." Following the survey, the researchers developed a program meant to show how easily users can be found using publically available location data. They organized their application into four blocks: geocoding, tweets and photo retrieval, Users Filtering, and KML (Keyhole Markup Language) production, and were successful in locating information of a Twitter user.

Toves implemented Exiftool in his research to show the development of metadata and how Exif data from a website can be captured[13]. As his research focused largely on Exiftool, Toves "demonstrated how the ExifTool has become a powerful tool in reading and manipulating this metadata." It is especially important because it is a similar tool which can interact with metadata, including functions which can delete the data. The user has the option to delete all of the metadata, to delete and comment when the metadata was deleted, or to relocate the data to a duplicate file. We can use this approach to model our own program, which focuses on deleting all the metadata or only location data.

C. Motivation

Image protection is extremely important for users of social media; however, many users are unaware that the images they upload can impact their privacy. The motivation for our research is the user privacy risks that come with uploading images. Social media sites currently offer two different types of security. First, the social media sites require authorization from the user. Second, some social media sites have servers provided to store user data on the server in an encrypted manner. According to Henne, Sznogott, and Smith "The amount of media uploaded to the Web is expanding rapidly: For instance, the number of photos uploaded per month on Facebook has risen from 2 billion in 2010 to 6 billion in 2011, and to over 9 billion in 2012 [14]." Privacy is becoming a more important issue, but is complicated because of the vast amount of friends each user has. Many people are concerned with their own posts, but there is also danger when a friend uploads an image that contains friends and the user or just the user. Also, when friends explicitly share images outside of the original poster's circle, it spreads to more people than those who originally had the capability of viewing it. As users cannot monitor their accounts or friends' accounts all the time, this becomes an uncontrollable issue [14].

Our goal is to improve default privacy settings and provide a better tool to ensure user protection when they choose to upload photos since every post has a risk that metadata could be accessible. Many users are unaware of what metadata is included or how to access that metadata to prevent privacy threats. The focus is on the removal of data before images are uploaded. There is a formulation of python code and an algorithm that will show how a user could protect their privacy in a more efficient way.

Our paper will focus on three specific questions:

- What metadata is encrypted in pictures when a social media user uploads information?
- How does that data threaten user privacy?
- How can a single user better protect their privacy?

D. Justification

To remedy the lack of privacy a user could encounter while dealing with certain social media sites, our research focuses on implementing a program which strips the data from multiple photos before the user uploads. The program will be created using the Python programming language and Spyder Integrated Development Environment (IDE). We focus on stripping the metadata because while encryption protects embedded information, it does not prevent someone from accessing the information. Someone can decrypt the image or gain access from a friend within the user's circle.

We also prioritize stripping GPS data in addition to stripping metadata because GPS data poses the biggest risk. If a malicious user obtains the coordinates, they can determine where the image was taken or where the user is often located. In 2013, Smith et. al mentioned that "the privacy issues of location information and other metadata embedded into uploaded media have not yet received much attention [15]." While there has been a shift in attention regarding location, the debate today is more on how location services should be implemented, rather than if they should be implemented.

E. Organization

In this paper, we will explore how metadata is hidden within pictures and how users can access the data to protect themselves. We will discuss the methodology behind our actions, as well as the materials, equipment, and equations used. The results from our experiment are shown in the third section. The fourth section is our discussion where we interpret our results, draw conclusions, and validate our conclusion by defending our interpretation of the results. We also continue conversation from previous research by reflecting on it and discussing the possibilities for future experiments. At the end of the discussion, we conclude by summarizing our experiment and providing ways to create a more efficient experiment to gain better results and draw more accurate conclusions.

II. Methods

As there are potential risks for the user when uploading, our research goal is to ensure privacy through the user's end. Our proposed solution is a program built with the Python programming language which will access a photo's metadata and strip it completely. This solution is meant to be implemented before a picture is uploaded. Our proposed method stems from similar programs accessible to the public,

but will differ in that users are allowed to choose to strip the entire picture, or just the GPS location, which is the easiest type of metadata to glean further information from. We decided to strip the photo of the data rather than encrypt because, using encryption, there are still ways to access the metadata. By completely stripping the photo metadata, however, we eliminate the possibility that others are able to find the information and decrypt it.

A. Materials

After considering other integrated development environments (IDEs) such as NetBeans, we decided to use the Spyder IDE to work with Python 3. Python 3 is more flexible compared to other programming languages. It is actively developed, updated often, supports multiple platforms, and has many libraries available. Python allows for more third-party libraries, which will be used to access the images, as well. The third party Python image library PIL will be used to work with images.

There are several datasets that could be used to implement our approach. The Tiny Images Dataset containing 79,302,017 32x32 color images is an option. This dataset was created by [16] to study scene recognition and benefits of having a large data set. It has been used by [17] to examine the roles of contextual information. It contains metadata for its images, which is beneficial since we will be working with metadata. It requires access to MATLAB and [16] provides code to access the first ten images. Another option is the RAISE dataset [18] containing 8,156 raw images. This data set was created by Dang-Nguyen et al. for digital image forensics and as a shared source for research. It has been used by [19] to research digital camera identification and [20] to explore image steganography with deep learning. Also, it was used by [15] to analyze issues regarding white image correction. The images contain metadata and its settings from the device they were taken with.

There is also the MIRFLICKR dataset [21] containing 25,000 images from Flickr users available under a Creative Commons License. These images do not contain EXIF data, but [22] does.

B. Algorithm

TABLE I. NOTATIONS

Notations	
Symbol	***Meaning***
I	Original image folder, complete dataset
I_x	Image with metadata, part of dataset
F	File directory/location
N	New/duplicate image
I(M)	Image only metadata. Includes dimensions, pixels, what keeps image same
E	All metadata. Includes I(M), GPS, Camera

(Symbols used in our algorithm and their meaning.)

TABLE II. PSEUDOCODE

Steps	
1. Program asks for location of I 2. User chooses if they want to delete all metadata or GPS 3. New folder is created in F, named new_images	
4a	***4b***
4a. I(M) of I_1 is put into list 5a. N is created in new_images, information contained in list is placed in N 6a. Program moves to I_2	4b. E of I_1 is put into list with sections named “0th”, “Exif”, “GPS, “1st” 5b. “GPS” is removed from list 6b. N is created in new_images, information contained in list is placed in N 7b. Program moves to I_2

(*This generalizes our pseudo-code for those not familiar with them*)

Input: User can strip a single file or multiple files.

- JPEG or TIFF files

Flag to strip either:

- All Meta-Data
- GPS only Meta-Data

For each file:

If the file exists:

Open the file

If strip all Meta-Data :

Read image data (only) into list

Create a blank new image (metadata-free)

Write image data from list (read above) to new image

Else : (Strip only the GPS data from the image)

Load image data using piexif

Get the GPS data from the image

Remove GPS data

Save file with new name

Keep track of how many files have changed.

Print # of files that have been “stripped”

Print total size of original files and new files.

Print total time it took to process.

Fig. 1. *Pseudocode displaying how metadata and GPS data is stripped*

Fig. 1 is derived from Fig. 2 and is explained in Table I and Table II. It reflects the process after the user inputs I. The application checks if the F is valid, as well as if I_x in I is in .jpeg or .tiff format. If any of these checks return false, an error appears. Otherwise, the application opens the I_x. Pillow library commands will be used to pull only I(M) into a list. N is then created only containing the dimensions, resolutions, and depth of I_x to keep the format and size the same. This is the method that occurs if the user chooses to remove all the metadata.

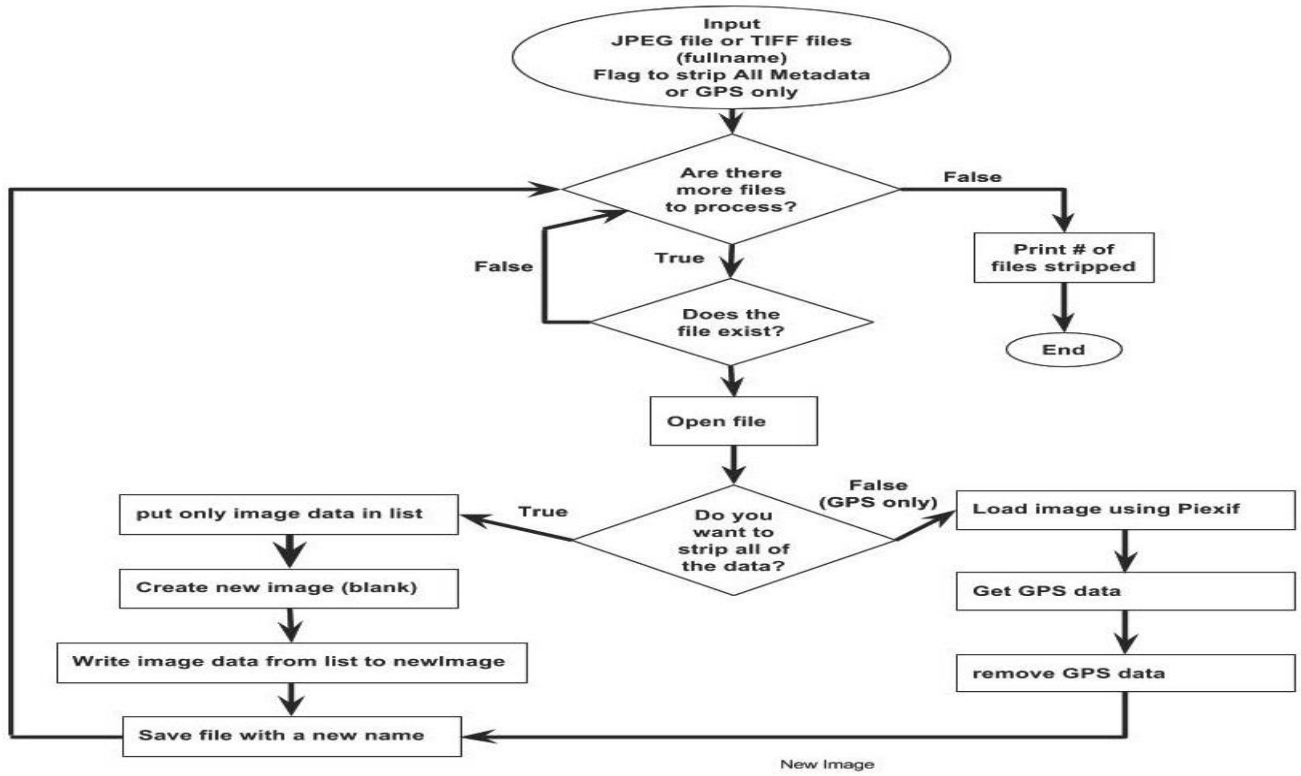

Fig. 2. *Shows the flowchart used to create the algorithm*

Instead of stripping E, there is also the option to only delete the location data embedded in I_x. In this instance, E loads into a dictionary separated into sections "0th," "EXIF," "GPS," and "1st". The location data will be found under the keyword GPS, and deleted. N is created and placed in the new folder.

In the beginning, the process creates a stripped images folder, if it does not already exist. Both processes occur as I_x in I is being stripped. Every time I_x is converted to N, N is placed in a new subfolder to avoid cluttering when removing large amounts of data. This process repeats for all unstrapped (unprocessed) files. Currently, our code supports stripped folders.

III. RESULTS

A. Implementation

TABLE III. IMAGE PROPERTIES

Image Properties						
	Origin	***Image***	***Camera***	***Advanced Photo***	***GPS***	***File***
Normal Image	X	X	X	X	X	X
GPS Data Stripped Image	X	X	X	X		X (size modified)
EXIF Data Stripped Image		X				X (size modified)

*X INDICATES THAT THE INFORMATION IS PROVIDED WITHIN THE IMAGE PROPERTIES.

Each of the five tabs in Table III indicates a specific property found within the details tab in the image's properties. All three images have the exact same appearance, as if nothing is missing from them; however, each has shared and different properties among them, including file size, which is drastically changed.

Normal Image— The origin tab includes the date and time the picture was taken as well as the program name data. The dimensions, width, height, horizontal resolution, vertical resolution, bit depth, resolution unit, and color representation are located within the image tab. The camera tab provides all of the information about the camera used to take the picture. This includes the camera maker and model, f-stop, exposure time, ISO speed, exposure bias, focus length, metering mode, 35mm focal length, and flash mode. The advanced photo property tab incorporates the brightness, exposure program, white balance, and EXIF version data. The GPS tab stores the latitude, longitude, and altitude where the picture was taken.

The file tab has all of the information regarding when the file was created on the computer, the item type and name, the folder path, the date created and modified, the amount of availability, the file's owner, the computer the file was created on, and the size of the file.

GPS Data-Stripped Image— Similar to the normal image, the GPS data-stripped image includes the same properties as the normal image, excluding one of them: all of the information that is under the GPS tab. The GPS data-stripped image does not include any GPS data whatsoever and the file size is significantly smaller than the normal image's file size. The file name is of course different from the normal image's name.

EXIF Data-Stripped Image— The EXIF data-stripped image is similar to the normal image and the GPS data-stripped image. The EXIF data-stripped program is supposed to

remove all the metadata inside the image or hidden in the image when going to the properties of the image of picture. The program strips everything from the image except the image itself and the file with the size already modified in the program. The change to the GPS data-stripped image and the EXIF data-stripped image is that the EXIF data-stripped image does not have the origin, the camera, the advanced photo, or the GPS property data.

We proposed an algorithm to strip all the metadata or GPS location data. From the data set [23], we implemented one approach, removing all metadata. The researchers used 27 raw images to test the program and successfully removed all metadata. We hypothesized that all the metadata included in the images would be removed. We were able to remove all metadata successfully from the sections circled in fig. 3, fig. 4, fig 5, and fig. 6.

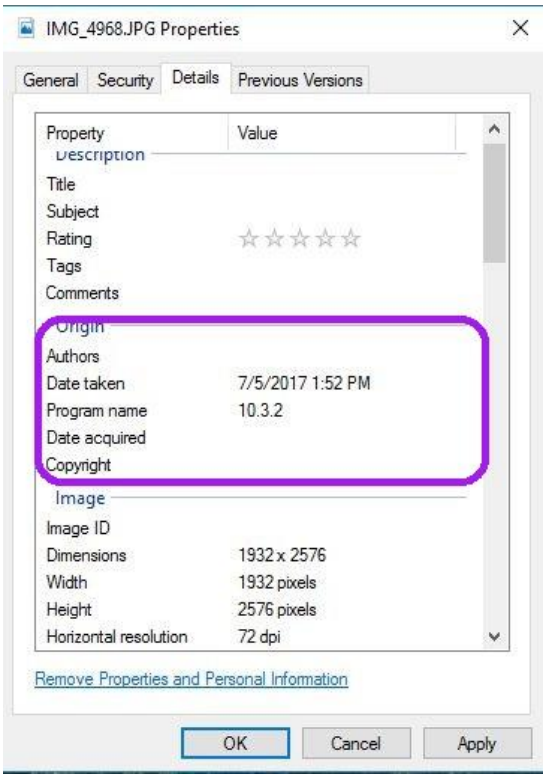

Fig. 3. (*Screenshot of the details from the images used. It shows origin metadata, which include date taken and program name*)

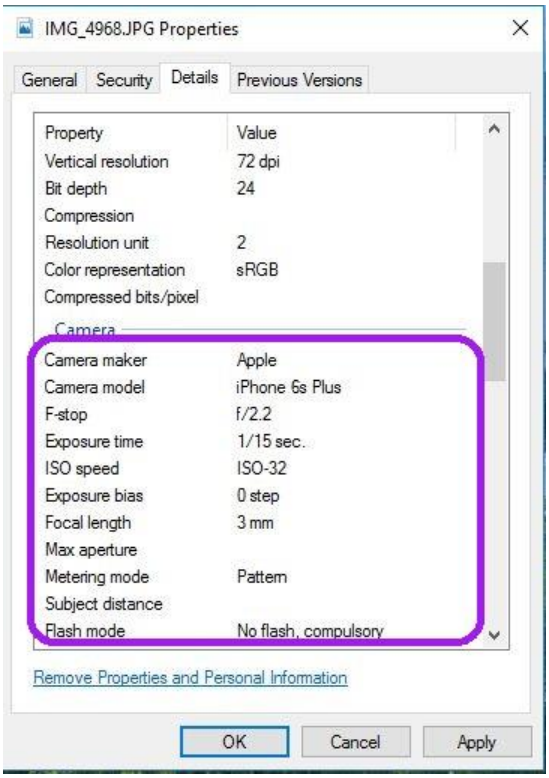

Fig. 4. (*Screenshot of the details from the images used. It shows camera metadata, which include camera maker, camera model, f-stop, exposure time, ISO speed, exposure bias, focal lenght metering mode, and flash mode*)

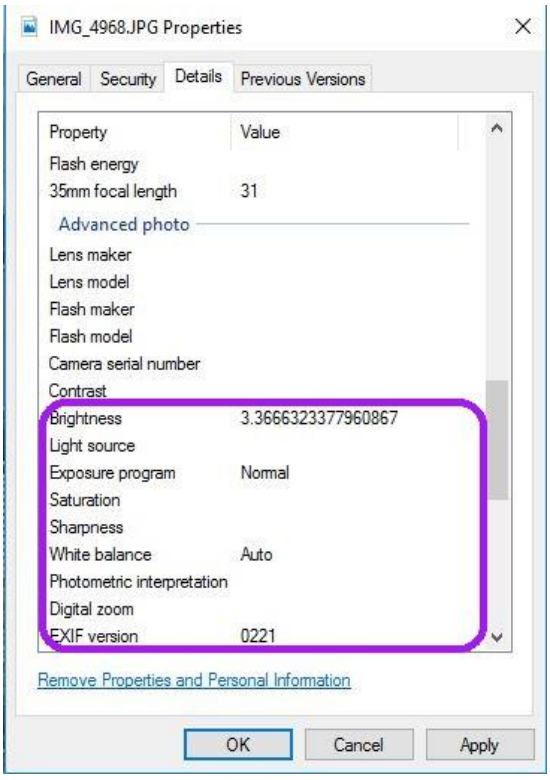

Fig. 5. (*Screenshot of the details from the images used. It shows advanced photo metadata, which include brightness, exposure program, white balance, and Exif version*)

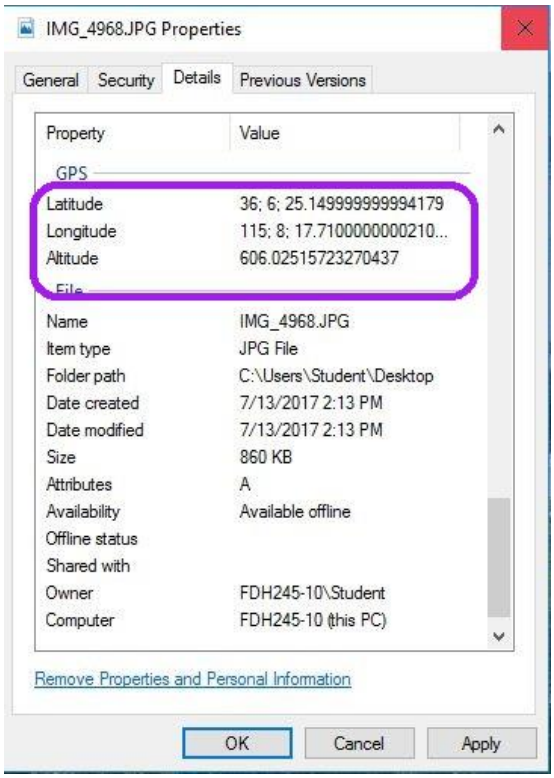

Fig. 6. (*Screenshot of details from one of the images used. It shows GPS metadata, which include latitude, longitude, and altitude*)

To study our program, we recorded the time it took to run and the size of stripped files shown on Table IV. We used the device described in Table VI. It took a long time, but that does not take into consideration the device used. There are supercomputers that can do it in seconds. While studying the images, the researchers noticed the difference in size of the original images with metadata and the stripped images without metadata. We conducted another test to record the size of the images before and after running the program as shown in Table V. The size decreased. One explanation of the increasing in file size is that the original files are compressed using the Lempel-Ziv-Welch (LZW) compression and the stripped files are uncompressed.

TABLE IV. TIMING AND SIZE

Removing all metadata		
Number of Images	***Size after (bytes)***	***Time (hh:mm:ss)***
27	613,347,056	0:03:22.388576

TABLE V. SIZE BEFORE AND AFTER

Removing all metadata			
Number of Images	***Size before (bytes)***	***Size after (bytes)***	***Time (hh:mm:ss)***
27	883915967	613,347,056	0:04:18.749799
462	413575645	804,741,931	3:04:50.572980

TABLE VI. DEVICE USED

Device information		
Processor	***RAM***	***System Type***
Intel ® Pentium ® CPU P6200 @ 2.13 GHz 2.13 GHZ	4:00 GB	64-bit

To implement the second approach, removing only GPS metadata, we used our own set of images. In this case, the image size increased instead of decreasing. It could be that this approach only stripped the GPS metadata and left the rest of the metadata. It could also be that in the original file, the metadata was compressed, and in the new image it is not. We also tested our own images combined with some of the dataset images. We wanted to see if it made a difference if the dataset contained images with GPS data while stripping all metadata. The results are shown in Table V. The size increased.

TABLE VII. GPS ONLY

Removing GPS Only			
Number of Images	***Size before (bytes)***	***Size after (bytes)***	***Time (hh:mm:ss)***
433	373,360,282	749,885,180	0:04:56.809941

B. Limitations

We proposed the use of several datasets. It was difficult finding datasets containing metadata. Those that contained metadata did not include GPS metadata or had it in a separate file [16]. Reference [18] had images with metadata in the same file, but did not include GPS. With it we were able to implement one approach, which was removing all metadata. Another limitation was time, as this research was conducted in six weeks. Due to time constraints, the researchers were not able to use [16], since getting access to, downloading, and combining the MATLAB files would take too much time. In addition, the files were in binary, and our program only worked on .tiff and .jpg files, which means the dataset would need to be converted. If we did combine the images with the metadata, conversion of the files would take time. We would have to create a conversion tool that would keep the metadata in the images, since we were not able to find one. Out of the 8,156 raw images in [18], we used 27 images because of the lack of time. They did not contain GPS data so the researchers were only able to test removing all the metadata from these images.

IV. DISCUSSION

There are other tools that have been created to edit, encrypt, and delete metadata, such as Exiftool, which encompasses all three options. It was a helpful tool to refer to, but ultimately was not our end goal as we wanted our program to be easily accessible to social media users. Exiftool uses the console and coding, but our audience may not be knowledgeable about file extensions or python commands, and so we built our program wanting to simplify it as much as possible. In the future, this program could be available for SNS users on both desktop and mobile platforms. We can also extend the type of files that can

be stripped and give a choice between stripping a folder and stripping a single file.

For further research, we need to figure out why the image grows in size after the metadata is removed. We can also find ways to decrease the time it takes to strip large folders. More research must go to whether or not the size and strip time were issues of compression. During testing, we can also test multiple devices and OS' to check if they are compatible and what, if anything, increases the program's runtime. By offering encryption or editing, we can allow the user more choice about what happens to the metadata, which can be helpful for future reference or analysis. Future research also entails a larger dataset to check for times and irregularities as files and folders grow larger.

V. Conclusion

We have created a way to protect users' privacy by introducing a program that allows users to strip either the metadata or location data from a batch of .jpg and .tiff files. This was done by storing only the image information in a duplicate file. The paper's focus was user privacy and metadata because as the internet and social media become more commonplace, there is more room for everything on SNS to be exploited [24]. Metadata, if shared, can find its way into the hands of malicious people. Although some SNS remove the metadata stored on a post, it is unknown if this information is completely deleted. By implementing our program, users can have access to a simple application which strips metadata so that user information will not be collected for any unknown purposes.

Acknowledgment

This material is based upon work supported in part by the Department of Defense under Army Educational Outreach Program (AEOP) and the National Science Foundation under Grant No. 1710716. The authors thank the UNLV writing center for helping with revising the manuscript.

References

[1] Lepsoy, J., Kim, S., Atnafu, D., & Kim, H. J. (2015, December). Metadata protection scheme for JPEG privacy & security using hierarchical and group-based models. In Information & Communication Technology and Accessibility (ICTA), 2015 5th International Conference on (pp. 1-5). IEEE

[2] Arjun, K. P., Achuthshankar, A., Soumya, M. K., Sreenarayanan, N. M., Priya, V. V., & Faby, K. A. (2016, January). PROvacy: Protecting image privacy in social networking sites using reversible data hiding. In Intelligent Systems and Control (ISCO), 2016 10th International Conference on (pp. 1-4). IEEE.

[3] Agrawal, D., El Abbadi, A., Arora, V., Budak, C., Georgiou, T., Mahmoud, H. A., ... & Wang, S. (2015, February). Mind your Ps and Vs: A perspective on the challenges of big data management and privacy concerns. In Big Data and Smart Computing (BigComp), 2015 International Conference on (pp. 1-6). IEEE.

[4] Alan Mislove, Bimal Viswanath, Krishna P. Gummadi, and Peter Druschel. 2010. You are who you know: inferring user profiles in online social networks. In Proceedings of the third ACM international conference on Web search and data mining (WSDM '10). ACM, New York, NY, USA, 251-260. DOI=http://dx.doi.org/10.1145/1718487.1718519

[5] Chen, K., & Rea Jr, A. I. (2004). Protecting personal information online: A survey of user privacy concerns and control techniques. Journal of Computer Information Systems, 44(4), 85-92.

[6] Leary, D. E. (2015). Big data and privacy: Emerging issues. IEEE Intelligent Systems, 30(6), 92-96.

[7] Pomerantz, J. (2015). Metadata (The MIT Press Essential Knowledge series). Cambridge, MA: The MIT Press, 16-19.

[8] Yoo, G., Sun, D., Lyu, K., & Kim, H. (2012, October). Data hiding technique for digital images tracing system on the Web. In Information Science and Service Science and Data Mining (ISSDM), 2012 6th International Conference on New Trends in (pp. 293-296). IEEE.

[9] Davis, M., & Sarvas, R. (2004, June). Mobile media metadata for mobile imaging. In Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on (Vol. 3, pp. 1707-1710). IEEE.

[10] McAuley, J., & Leskovec, J. (2012). Image labeling on a network: using social-network metadata for image classification. Computer Vision–ECCV 2012, 828-841.

[11] Benjamin Henne, Christian Szongott, and Matthew Smith. 2013. SnapMe if you can: privacy threats of other peoples; geo-tagged media and what we can do about it. In Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks (WiSec '13). ACM, New York, NY, USA, 95-106. DOI=http://dx.doi.org/10.1145/2462096.2462113

[12] Furini, M., & Tamanini, V. (2015). Location privacy and public metadata in social media platforms attitudes, behaviors and opinions. Multimedia Tools and Applications, 74(21), 9795-9825.

[13] Toevs, B. (2015, November). Processing of Metadata on Multimedia Using ExifTool: A Programming Approach in Python. In Information and Computer Technology (GOCICT), 2015 Annual Global Online Conference on (pp. 26-30). IEEE.

[14] Chin, E., Felt, A. P., Sekar, V., & Wagner, D. (2012, July). Measuring user confidence in smartphone security and privacy. In Proceedings of the eighth symposium on usable privacy and security (p. 1). ACM.

[15] Smith, M., Szongott, C., Henne, B., & Von Voigt, G. (2012, June). Big data privacy issues in public social media. In Digital Ecosystems Technologies (DEST), 2012 6th IEEE International Conference on (pp. 1-6). IEEE.

[16] Torralba, A., Fergus, R., & Freeman, W. T. (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE transactions on pattern analysis and machine intelligence, 30(11), 1958-1970.

[17] Parikh, D., Zitnick, C. L., & Chen, T. (2012). Exploring tiny images: The roles of appearance and contextual information for machine and human object recognition. IEEE transactions on pattern analysis and machine intelligence, 34(10), 1978-1991.

[18] "D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, G. Boato, RAISE - A Raw Images Dataset for Digital Image Forensics, ACM Multimedia Systems, Portland, Oregon, March 18-20, 2015"

[19] Quiring, E., & Kirchner, M. (2015, November). Fragile sensor fingerprint camera identification. In Information Forensics and Security (WIFS), 2015 IEEE International Workshop on (pp. 1-6). IEEE.

[20] Couchot, J. F., Couturier, R., Guyeux, C., & Salomon, M. (2016). Steganalysis via a Convolutional Neural Network using Large Convolution Filters. arXiv preprint arXiv:1605.07946.

[21] Huiskes, M. J., & Lew, M. S. (2008, October). The MIR flickr retrieval evaluation. In Proceedings of the 1st ACM international conference on Multimedia information retrieval (pp. 39-43). ACM.

[22] Huiskes, M. J., Thomee, B., & Lew, M. S. (2010, March). New trends and ideas in visual concept detection: the MIR flickr retrieval evaluation initiative. In Proceedings of the international conference on Multimedia information retrieval (pp. 527-536). ACM.

[23] Smith, M., Szongott, C., Henne, B., & Von Voigt, G. (2012, June). Big data privacy issues in public social media. In Digital Ecosystem Technologies (DEST), 2012 6th IEEE International Conference on (pp.1-6). IEEE.

[24] Lindsay, B. R. (2011). Social media and disasters: Current uses, future options, and policy considerations.