ABSTRACT
As more people around the world gain access to the Internet, the user base of online social networks such as Facebook, Twitter, Google Plus, LinkedIn, and Instagram will continue to grow. Due to the prevalence and ubiquity of social media, more personal content is being migrated online and more people are using online data as a way to access others both personally and professionally. Therefore, it has become increasingly important to be able to manage the quality and quantity of personal content that is available on the Internet. Unfortunately, the sheer amount of data uploaded to social networks makes this an extremely difficult goal to realize. For a typical online user, the amount of personal content continues to accumulate on the Internet, raising a significant amount of privacy concerns commonly discussed in the mainstream media. Though many social media sites offer advanced and customized control over the privacy settings of personally uploaded data, the fact that few users utilize these features, coupled with the personal content uploaded by “friends” on social media, lends itself to a wealth of personal content online that people often do not realize exist.

The “Untagged Photo Detector” is a web application that seeks to remedy this issue using a defensive approach which identifies unexpected photos on Facebook and gives users the option to take action against them. Specifically, it helps users discover pictures uploaded by Facebook friends which they are not currently tagged in. The application aims to raise awareness of personal data that are available online and give users a simple interface through which they can better manage their photo privacy. The ultimate goal is to improve the privacy management and awareness of personal data on social networking platforms.

1. INTRODUCTION
As of January 2014, online social network (OSN) use for adults with internet access stands at 74% [14]. To put this into perspective, globally, the number of worldwide internet users is at 2.92 billion [17], approximately 41% of the total world population and is growing at an annual rate of about 7%.

With the proliferation of social networks comes the sharing of personal data, including photos and videos. For example, Facebook, with its 1.28 billion monthly active users [1], sees 350 million photos uploaded daily [16] and Instagram, with 200 million monthly active users, has daily uploads in the 40 million range [3]. Considering the large number of other social networks, such as Twitter, Google Plus, LinkedIn, Pinterest, and YouTube, amongst others, users likely have their personal information exposed across a variety of OSNs.

In the age of the “like”, individuals are becoming their own content creators rather than merely content consumers. However, this shift towards online sharing raises significant privacy concerns for users. For many users, as the number of tweets, pictures, wall posts, and videos grows, it is unclear exactly what information of theirs is accessible online and to what audience. Past studies have found that on Facebook, user expectations of privacy settings and actual privacy settings only aligned 37% of the time [12]. Moreover, when these privacy settings did not match, they tended to give a larger number of users than expected access to that content.

These privacy issues are common among many and are often discussed in the media. In particular, Facebook has undergone significant backlash related to its privacy settings. In 2012, almost 13 million users never altered or knew they could alter their privacy settings [7]. With Facebook announcing the splitting of the Messenger application from the core application a few months ago, the concerns related to privacy have become even more heightened. Even with easier to use controls on social networking sites, it is still no simple task for users to view what personal content of theirs is available to what audience.

Thus, our goal is to make the process of managing personal content easier for social network users. Given the immense amount of personal data, we chose to focus specifically on photographs, which comprise a large portion of the available information on Facebook. In fact, photographs account for 93% of the most engaging posts on social networks. They get 39% more interaction than other types of posts - 53% more likes, 104% more comments, and 84% more click-through as compared to posts that are text-based [2]. The widespread exposure of Facebook pictures leads to some of the biggest privacy concerns. About 36% of Facebook users feel uncomfortable with others posting content about them without asking for their permission in advance [15]. To mitigate this concern, our application will allow users to discover images of themselves uploaded onto Facebook either with or without their permission. As a result, Facebook users will be able to manage and keep track of all of photographic content related to them, even if they are not explicitly tagged in the photographs.

Advisor: Chris Murphy (cdmurphy@seas.upenn.edu)
2. RELATED WORK

The motivation for our project proposal stems from past research that attempts to quantify this privacy problem resulting from increased social sharing online. These studies have examined the prevalence of privacy setting errors in social networks. In 2011, Liu et al. performed analysis on user expectations versus reality related to sharing of Facebook photos. Users on Amazon Mechanical Turk were asked about who the audience for ten uploaded photos was intended to be and compared this with the actual privacy settings of these images. It was found that only 37% of the actual settings and the desired settings matched. In the majority where the actual and desired settings did not align, the actual privacy settings were almost unanimously less private than the user had originally intended [12], demonstrating that there is likely a large discrepancy between what users hope their privacy to be and what their privacy actually is. These studies suggest that photos of a user uploaded by others may be viewable by an alarmingly large audience, which is a source of deep concern for the average user that hopes to limit the exposure of personal content.

In another study analyzing the sharing intentions of Facebook users, Madejski and Johnson found that for all 65 users surveyed, there was at least one privacy inconsistency for every user [13]. For 94% of individuals, the inconsistency was a hide error, where the networks displayed something the user wanted to hide [13]. On the other hand, 85% had a show error, in which the user was hiding something they wanted shown [13]. Clearly there exists a huge inconsistency between what the user of a social network such as Facebook intends to display to their audience and what that user believes is shown. Subsequently this raises concerns for the user’s friends, who although may be knowledgeable about their privacy settings, could have pictures of themselves unknowingly revealed to a significantly large audience.

A study of Facebook use at Carnegie Mellon University (CMU) in 2005 demonstrated the large amount of personal information students tend to share. Through surveys of over 4000 students at CMU, it was noted that 50.8% of users put their address on Facebook, 87.8% list their birth date, and 39.9% of profiles listed at least one phone number [8]. Information such as birthday, address, phone number, and hometown can be used to estimate someone’s social security number. This makes individuals susceptible to identity theft, a problem that goes well beyond certain photos being visible to undesired audiences.

Other work has attempted to decrease the amount of information available to third party developers on Facebook and other similar social networking sites, deemed privacy by proxy [6]. Many social network users quickly bypass the terms and conditions, giving the application access to a large amount of their data. These third party applications pose another privacy concern, as the application is able to gather the individual’s personal data and learn about that user’s network. Components include data hiding and user ID encryption, in which markup languages are used to anonymize data and handle inputs without releasing any private information to third parties.

Even if the user adequately hides their profile and data, it is possible for his or her private information still to be exposed through sensitive attribute inference [19]. This occurs where hidden information of users are exposed due to the social network containing profiles, groups, and links that are publically visible. Particularly, homogeneous groups can leak a significant amount of information. One can essentially exploit a social network with mixed profiles to learn about sensitive attributes. While much more effective with group information than public links, Zheleva et al. were able to fairly accurately determine sensitive information on four different social networks - Facebook, Flickr, Dogster, and Bibsonomy.

3. SYSTEM MODEL

The “Untagged Photo Detector” is a web application that enables users to find photographs on Facebook which they are not currently tagged in. When using the application, the user will be prompted to login to Facebook, which allows the application to search through uploaded photos of the user’s friends for the test set - photos of the user in which he or she is not already tagged in. The user also provides the application a training set, about twenty to thirty photos, which the application uses to train the facial recognition engine.

Untagged photos in which the user is detected will be displayed to the user, ranked in order by the percent likelihood that he or she appears in the photo. This is a measure that combines the facial recognition results and the metadata surrounding a photo. In addition, the user will be shown a risk level associated with each photo, which is based on metadata processing and represents the degree to which the photo may be risky - either due to a large audience or sensitive content. At this point, the user could choose to take action by requesting the removal of this photo from Facebook or simply ignoring the photo, discarding it from the application’s feed.

Figure 1 below is a depiction of the system model design for the application.

4. SYSTEM IMPLEMENTATION

“Untagged Photo Detector” is designed as a stand alone web application using Node.js [11], a runtime environment for networking applications, with Express [4] as the framework and Jade [10] as the templating language for the front end.

Beyond the general technology framework, “Untagged Photo Detector” mainly utilizes two publicly available technologies in order to successfully perform the task of identifying users in untagged Facebook photos. The first piece of technology is a free, open source Face API offered by Lambda Labs, a machine learning and artificial intelligence company that processes images online. The Facial Recognition API is treated like a black box in the application and used to detect and recognize the user’s face in the photos. The other web service is the Facebook graph API which grants access to all of a user’s Facebook data, including but not limited to the user’s profile, friends, past locations, and album information. The application bridges together these two main components in order to recognize users in photos uploaded on Facebook.

The system implementation of the application is broken down into the following steps. First, the Facial Recognition API is trained against the user’s facial features using a high quality training set uploaded by the user to the application. Next, photos are retrieved from Facebook using the Facebook API. The retrieved photos are sent to the Facial
4.1 Training

To be able to recognize the user’s face in photos, the Facial Recognition API must be trained using a set of photos that uniquely identify the user. The application prompts the user to upload at least 20 photos to be used as the training set. Each photo must contain only the user and no other detectable faces. In addition, the user should not be wearing any glasses or facial accessories that block his or her facial features. Uploading photos that adhere to these general guidelines will ensure the best results for facial recognition. These photos are uploaded to an album on the Face API under a unique username which we obtain from the user’s unique Facebook ID.

The Facial Recognition API needs to be trained on more faces in order to be able to uniquely identify the user’s presence in photos. Without other people in the training set there would be no basis of comparison and any detected faces would be trivially recognized as the user. A set of test users consisting of mostly public figures and celebrities were chosen to represent the three major races - Caucasian, Mongolian, and Negroid. For each racial group a total of six users, half male and half female, were selected. Photos from these individuals were downloaded from the Internet and uploaded to the Facial Recognition API beforehand. With the Facial Recognition API properly trained and configured, the application then moves on to the next stage of retrieving photos from Facebook.

The effect that the quality of the training set has on the accuracy of the result is further discussed in the evaluation section.

4.2 Photo Retrieval

To retrieve relevant Facebook photos, the user must first log in to the Facebook application through the web application. By providing the web application with the App ID and App Secret associated with any application registered with the Facebook API, the user can then be prompted to log on through Facebook, giving the necessary permissions - user’s friend list. This log in is done with Passport [9], an authentication middleware for Node.js which is useful for web services such as the Facebook API that require token-based credentials for data access. Upon successful login, the application is given an access token, which uniquely identifies the user to the application.

From there, the application can begin to query the Facebook Graph API for the test set. The Facebook Graph API [5] is integrated into the application using facebook-node-sdk [18], an open source SDK similar to the built-in JavaScript SDK. In Facebook Graph API, nodes and edges can be retrieved with HTTP GET requests. In order to be as efficient as possible when retrieving the large amount of photo data from the Graph API, we have limited the number of requests we make for each user and the amount of information from those requests to a minimum.

The application makes two requests. The first request simply obtains the user’s id and name. The second request retrieves photos uploaded by the user’s friends. This is a much more expensive GET request, given the amount of friends a user likely has and the amount of photos each of his friends has uploaded. Currently we are using the following HTTP request to grab all photos that user’s friends have uploaded.

```
GET graph.facebook.com/me?fields=friends.limit(10)
{albums.limit(10){photos{id, source, link, name, tags, comments, likes}}}
```

From each friend of the user, the albums he owns are retrieved. Then for each photo in those albums, the source of the image, the link of the image, and the people who are tagged in it are fetched. The link field differs from the source of the image, which is simply the path to the JPEG file. In contrast, the link stores the URL to the Facebook page that contains the photo. This URL is fetched to allow the user to directly access the photo from Facebook later on and take any relevant action. Relevant metadata from the photo is also retrieved, including the caption, likes, comments, which are stored in the name, likes, and comments field, respectively. With the JSON response, each photo’s tags can be compared to the user’s id. For each photo in which the user
is not tagged, the source of the image and relevant metadata is passed to the photo processing component. The friends and albums fields are each limited by 10, which means that each page of the GET request will display up to 10 albums or friends. Above this amount of friends or albums, and the Facebook API will sometimes return an error identifying that too much information is being queried for at once. To see the next 10 albums or next 10 friends, the application utilizes Facebook Graph API’s cursor-based pagination. In the JSON response, the “next” field allows the application to access the next part of the relevant data. Once there is no more “next” field, the applications knows that it has reached the end of the data.

It is important to note the limitations that the Facebook Graph API places on the functionality of the “Untagged Photo Detector.” With the Graph API, when the user gives permission for the application to access the “user’s friends” category, the list of user’s friends that can be returned by a query are only those who have previously logged into the application. This is an update in Graph API v2.0 and above; previous versions of the API allowed applications to access a user’s entire friends list. This means that the more users that utilize the application the larger the test set for each user. Unfortunately, this also means that for the first few users, who may have few friends on the application, the potential photos returned are severely reduced.

4.3 Photo Processing

4.3.1 Facial Recognition

Once the test set is retrieved from Facebook, the links to the photos are sent to the Facial Recognition API to be recognized. Due to the large number of photos retrieved from Facebook, the photos are uploaded to the API in smaller batches to prevent overloading the server with too much data. The API returns a JSON object that contains the status and the results. For each photo uploaded to the server, the result contains a list of Euclidean coordinates that represent the locations of the detected faces and the predicted gender of each tag with an associated confidence. When one or more faces are detected in the photo, the result also contains a confidence score on a scale of 0 to 1 for each user in the training set. The scores for all of the users in the set should add up to a total of 1.

Recognizing the photos is performed remotely on the server of the Lambda Labs Facial Recognition API. Since the application treats this service as a black box, the performance of this step is beyond the scope of our control. Unfortunately, the photo recognition step constitutes the bottleneck of the application. The recognizer has fast performance when no faces are detected in the photos. However, when faces are detected in the photo, it can several seconds to recognize the users. The larger the training set, the longer it takes to recognize the user. This trade off is further examined in the performance part of the evaluation section.

4.3.2 Metadata Processing

The metadata associated with a photo is processed in order to improve upon the accuracy of the score returned from the face recognizer. The motivation for using metadata comes from the observation that a user can be present in a photo even when the face recognizer returns a low score. For example, the face recognizer may be unable to recognize the user’s face in photos with bad lighting, poor quality, unusual facial angles, and other unpredictable conditions that hinder the recognition results. In other words, though a high face recognition result typically indicates a user’s presence in the photo (with some level of error), a low face recognition score may not indicate a user’s absence in photos. Thus it is necessary to seek other ways to determine whether or not a user is present in a photo. In particular, the metadata gives contextual clues that may further suggest a user’s presence in a photo.

The metadata provides the context associated with an image including the creation time, the user tags, the caption, the likes, the comments, the location, the album in which the photo appears in, etc. The application processes the captions and the comments in order to help generate the confidence score. In addition, it looks at the likes, comments, and captions to generate the risk level score.

4.4 Results Ranking

4.4.1 Generating Confidence Score

Before displaying the untagged photos to the user, it is necessary to generate a score based on the likelihood of the user being present in the photo. The resulting scores will be used to sort the photos in the next phase and display them in order from highest to lowest probability of match. In addition, a threshold value will be selected in order to filter out photos that the user is unlikely to have appeared in.

A user’s presence in a photo depends on two major factors. A positive test result from the photo recognition algorithm would provide a strong indication of a user’s presence in a photo. However, there are also many cases where user presence in a photo cannot be detected directly through facial recognition. Consider, for example, a case where a photo has poor lighting, the user’s face is turned at an angle, or the user is not facing the camera at all. In these scenarios, a human observer could still be able to deduce the identity of the person through other data present on the page. This motivates the inclusion of both the facial recognition confidence level and the result of processing the metadata in the final score representing the probability of the user appearing in a photo.

What sorts of metadata should we be interested in? At the very least, we would want to consider the caption and comments within the photo that provide context. If a user’s name is mentioned in the text, he is more likely to be in the photo. Another factor to consider is the percentage of photos in the album that the user is tagged in. If the album already contains a large portion of photos of the user, he is more likely to appear in a photo within this album. We will now briefly discuss how to calculate the score.

The photo recognition tool returns an accuracy level in the form of a percentage value. We will take this percentage and scale it to be out of 80 points. The bulk of the weight falls into the facial recognition score because a high confidence level from facial recognition provides a very strong indication of the user’s presence in the photo. The remaining 20 points will come from the metadata associated with the photo. The 20 points come from a binary check of whether the user’s name, first or last, is mentioned in either the caption or any of the comments. If so, the confidence level for the photo is boosted by 20 points.
The score is calculated as follows:

\[ P(X) = (A \times 80 + B \times 20) \]

where \( X \) is a random variable representing the probability that the user appears in the photo, \( A \) is the match percentage returned by the facial recognition algorithm and \( B \) is the binary score representing whether or not the user’s name was mentioned in the photo. The cutoff threshold is currently set at 60%, so any photos with a 60% or higher confidence level are shown to the user. This cutoff threshold is further evaluated in the ‘Evaluation Results’ section.

### 4.4.2 Generating Risk Level

The calculated risk level associated with each photo represents the degree to which the user should be concerned about the photo’s presence on Facebook. A photo is categorized as risky for two main reasons - either the photo has a large audience or the photo may contain some sensitive content. These two aspects are combined into one single measure that is shown to the user.

A large audience is partly determined by the number of likes and comments on a photo. When a photo is liked or commented on, it is often shown at the top of friend’s news feeds, so it is more likely that many people have seen that photo, which makes the photo potentially more risky to the user. To determine whether there is sensitive content, the application once again looks at the photo’s caption and comments, checking to see if any of them contain a bank of negative words. These words include ugly, gross, drunk, fat, etc.

The risk level is calculated as follows:

\[ R(X) = (70 \times A/A' + 20 \times B/B' + 10 \times C/C') \]

where \( R(X) \) is the risk level associated with the photo, \( A \) is the number of likes on the photo, \( A' \) is the maximum number of likes in the photos that will be returned to the user, \( B \) is the maximum number of comments on the photo, \( B' \) is the maximum number of comments in the photos that will be returned to the user, \( C \) is the number of negative words found in the caption or comments, and \( C' \) is the maximum number of negative words out of the photos that will be returned to the user.

As shown in the equation, each value - likes, comments, or negative words is normalized by the maximum in that category. Likes are given the most weight at 70%, then comments at 20%, and negative words at 10%. There does not need to be a photo of 100% risk level returned to the user. The risk level is more of a relative measure rather than an absolute measure. Seeing the risk level may help the user to determine whether or not they should seek removal of the photo. Figure 2 shows how the metadata processing produces the confidence and risk levels.

### 4.5 Photo Display

After the resulting set of photos have been properly sorted by rank, a list of paths to JPEG image files will be returned, along with their corresponding links on Facebook, confidence levels, and risk levels. The images are displayed in the browser by embedding the JPEG links in HTML. The resulting photos will be ranked by the decreasing level of accuracy calculated in the previous phase. For each photo, the user can decide to request the removal of the photo, which links directly to the photo on Facebook, or to ignore the photo, which removes it from the feed such that it will never be displayed again.

The application implements caching on the front end to help improve the speed and to guarantee a better user experience. An open source library for Node.js called “node-cache” was used to cache the results returned from the processing phase so that no additional processing is required to reload the page when the contents have not changed.

## 5. RESULTS

The resulting web application implements the core functionality for detecting potentially risky photos of users. The web application was named “Untagged Photo Detector.” The features implemented include: uploading a training set of photos containing the user’s face, retrieving photos and metadata (likes and comments) from the user’s friends using the Facebook API, performing facial recognition using the facial recognition API provided by Lambda Labs, processing the metadata and generating scores for facial recognition and risk level, ranking and presenting the photos based on the scores generated, and remove and ignore functionality on the detected photos.

The Untagged Photo Detector consists mainly of the login, import, and results page. When the user first opens the application, the login page opens (shown in Figure 3). In the login page, the user can log in to his or her Facebook account.

![Figure 3: Login Page](image)

Once the user is logged in, the user is directed to the import page, which is shown in Figure 4. The import page allows the user to select photos that the application can use
as a training set to train the face recognizer. The user can choose photos locally by clicking the “Choose Files” button and clicking “Submit.” Approximately 20 to 30 photos are recommended for optimal facial recognition.

For each of the photos outputted on the results page, the user can choose to “Remove” or “Ignore” the photos. When the user clicks “Remove,” the application is redirected to the Facebook page where the photo is located, where the user can select “Report photo” (shown in Figure 6). The reason why the application can’t directly select this button is because the URL for the button is same as the URL for the page containing the photo. When the user selects “Ignore,” the photo is removed from the results page.

6. EVALUATION

6.1 Criteria

The application is evaluated with the input provided by twenty users. Each user has a test set that consists of a collection of 20 accurately tagged photos of the user uploaded to Facebook by his or her friends who are already on the application. The test set is then generated by untagging the user from all the photos in the collection. Each user then runs the “Untagged Photo Detector” on his or her Facebook data. The accuracy of the results is calculated by the percentage of photos in the test set that showed up in the result. The precision of the results measures the percentage of the returned photos actually contain the user. In addition, each user reported the percentage of photos that they have not previously seen. This statistic is the most informative in gauging whether the application is true to its mission of raising awareness of personal content. Lastly, the percentage of photos that the user took action against was also calculated in order to assess the usefulness of this application in helping to change the way that people manage their online content.

6.2 Results

The results, aggregated across the twenty users, show that the application is 88% accurate and 81% precise, which shows that in most cases it correctly identifies the presence of a user in photos uploaded by his or her friends. This means that there were about 12% false negatives and 19% false positives. In addition, around 52% of the photos returned to the user are new photos that have not been seen before, and around 21% of the returned photos were subsequently removed from Facebook. These statistics show that the application goes a step beyond simply recognizing faces in photos, but actually impacts the way that people manage their online content by discovering interesting photos that
the user should be concerned about. 85% of the users claim that they would use this application again, which indicates the need for a better privacy management solution than the ones that are currently available on the market.

The precision and accuracy are influenced by the cutoff confidence level chosen for the application - which was set to 60%. This was found to be a confidence level that would have few false positives but would also not cut off many true positive results. To do so, the for the user test set, all photos with 40% or above confidence level were displayed to the user. However, in the final precision, accuracy, percentage of previous unseen photos, and percentage of removed photos statistics, only photos with 60% were considered. Figure 7 shows how the precision changed with the confidence level cutoff. From 40% confidence level or above, there is a certain number of photos containing the user. The second line in the graph shows what percentage of these photos are displayed at each cutoff point. 60% was chosen to be the final cutoff point because it is a medium between false positives and cutting off true positives.

![Figure 7: Amount of time to retrieve photo data with different number of friends](image)

### 6.3 Training Set

It is also worth mentioning that the accuracy of the results depends heavily on the quality of the training set. As previously mentioned in the implementation section, the training set must contain more than one user to guarantee meaningful results during recognition. When the number of users in the training set is just one, the number of false positives is high because the application will recognize any face as the user. In addition, the diversity in the training set also matters. The facial recognizer will most likely not be able to differentiate two people of the same race if there is only one user of that race in the training set. In this case the number of false positives will also be high as the application might incorrectly recognize the user as the person of the same race in the training set. The same argument follows for gender.

These observations led to the decision of training the Facial Recognition API using a diverse set of users before running the application.

It is also interesting to note that the application will perform better if the application is able to train on the people who appear frequently in the user’s photos. Therefore, one way to improve the accuracy of the application is to train the facial recognizer with these people. However, obtaining this set of people is a difficult task because it is impossible to find this set of users before the application finds untagged photos of the user. One way to adapt to this limitation is to use machine learning to learn and predict the users that might appear in the photos next time. Unfortunately, the application does not implement this currently, but it would be a desirable extension to the project.

### 6.4 Performance

Depending on the number of friends and photos that a user has, the Facebook Graph API will take a different amount of time to retrieve all the relevant photos and metadata. See Figure 8 for a graph of the number of friends versus the amount of time it takes to retrieve all relevant photo data. Looking at this data, for a user with over 1000 friends, it will likely take up to about 15 seconds to retrieve all of the relevant photo data for the initial scan. At 20 friends, it already takes approximately 7 seconds. Given that limiting the number of friends to a smaller subset doesn’t drastically decrease the time, e.g. about 600 friends still takes 12 seconds, it is best to just retrieve all the relevant data. In the end, this helped to return the best results for the user.

The mean number of friends adult Facebook users have is about 338 friends. For Facebook users in the 18-29 age bracket, though, this number is likely higher, as 27% of users have over 500 friends [15]. Hence for the average adult Facebook user with 338 friends, he or she will have to wait about ten seconds for just the photo retrieval component of the application.

![Figure 8: Amount of time to retrieve photo data with different number of friends](image)
7. FUTURE WORK

Given that the project is still at its initial stages and that there is a high need for privacy management of photos in the Internet, there is much room for future work. Once of the main area for future work is the maintenance of photo privacy management. Currently, the application only performs a one-time scan for potentially risky photos of the user. However, the application can further be improved in order to conduct periodic scans of the user’s friend’s photos. This could be done in a manner such that the application scans every week only photos that have been uploaded that week.

Another room for development is on the method in which the training set is loaded. Currently, the application makes the user manually upload 20 to 30 photos of the user locally. The reason for doing this was that the profile picture of the user on Facebook might contain faces of other users, thus making the training set inaccurate. However, the application can be improved such that that app automatically goes through the user’s photos and detects the user’s face once the user logs into Facebook. This could be done by using coordinates of faces in photos, and comparing the user’s profile pictures and finding the person whose face is repeated in the photos.

The metadata processing also could be further developed. Currently, the application counts the number of likes and comments in order to detect the exposure level of the photos, and checks whether the comments contain negative words by comparing with a bank of negative words. The metadata processing could be made smarter using algorithms such as natural language processing in order to check for the context of the photos.

Additionally, a history of detected photos for which the user clicked “Remove could be stored, in order to keep track of which photos the user identified as being risky. A statistical metric can be constructed using this history of records, which can be used to make future detection by the application smarter.

Finally, the scope of the project can be enlarged to other social networks and forms of data. Presently, the application only focuses on photos on Facebook. However, the project could be expanded to other social networks such as Instagram, Google, Twitter, and LinkedIn. Also, the idea of the project can be expanded to other forms of private information such as address information, career information, social network interaction information and other personal information.

8. CONCLUSION

The rising popularity of social media networks and the prevalence of content sharing online today raises concerns over the privacy and protection of personal data. In fact, studies have shown that many users of social networks are unaware of what data of theirs is accessible and to whom. This is the primary motivation for building the “Untagged Photo Detector,” which is a web application that helps users gain more awareness and control over photos that they appear in on Facebook. Though there are numerous forms of online personal content, the focus of this study is on photos circulating on Facebook because the vast amount of uploaded photos makes it extremely difficult for users to manage their privacy. In contrast to a preventive approach, the application helps users identify presence in photos only after they have already been uploaded to Facebook. The application detects a users’ unexpected presence in Facebook photos and displays them to the user from highest to lowest probability. The application also provides basic options for removing or ignoring a photo from the results page. It encourages users to take active measures to help protect the privacy of their personal information on Facebook. The evaluation results show that the application has raised user awareness of personal data by displaying content that they have not previously seen before on Facebook. The success indicated by the results motivates further development of the product to provide better privacy control to users over their online personal data.

9. ETHICS

An application that deals so heavily with the issue of privacy and personal information could pose many ethical questions. An area where such concerns could arise is the user’s private Facebook data which could be exposed and vulnerable to malicious attacks as the application is running. By prompting the user to log in to their Facebook, the application holds an important piece of information, the user-specific access token, which is the key to all of the user’s private information on Facebook. In addition, the access token can also be used to query the user’s friends’ information that have been shared. Some of this data might have been intended for a limited audience or unintentionally uploaded to Facebook. Thus, the access token should be protected to prevent against common security issues such as identity theft. The application must take extra measures to securely store the user’s fetched data, which they entrust with the application. Similarly, the cache of results should not expose any of the user’s sensitive information. Without an effective security measure, the process of fetching private information could lead to additional privacy risks which would counteract the original goal of protecting privacy.

10. REFERENCES

problems-are-rise-f749990/.


[16] Cooper Smith. Facebook users are uploading 350 million new photos each day.

